Technische Universität Hamburg-Harburg

Bachelor Thesis

Institut für Thermofluiddynamik

TECHNISCHE THERMODYNAMIK

Prof. Dr.-Ing. G. Schmitz

Modelling and Control of a Cooling System for Power Electronics on a Commercial Aircraft

Author
cand. BSc. Christian Hoffmann

Supervising Tutor
Dipl.-Ing. Karin Dietl

September, 15^{th} 2008

Ich versichere, diese Arbeit im Rahmen der am Arbeitsbereich üblichen Betreuung selbstständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben.

I hereby affirm the sole authorship of this thesis, that it has been completed under the supervision common to the institute and that no sources or auxiliary means have been used other than specified.

Hamburg-Harburg, September, 15^{th} 2008

TECHNISCHE THERMODYNAMIK

Technische Universität Hamburg - Harburg

Mr. cand. BSc. Christian Hoffmann
Student ID: 20522587

Task description for a Bachelor Thesis

Subject:

Modelling and Control of a Cooling System
for Power Electronics on a Commercial Aircraft

In the framework of the European research program "More Open Electrical Technologies (MOET)" cooling architectures are developed for a bleedless aircraft by the Institute of Thermo-Fluid Dynamics together with Airbus Germany. The cooling architectures, which provide cooling for cabin air, power electronics and commercial systems, are optimised regarding drag, weight and power consumption in order to minimise the fuel required. For the purpose of optimising power consumption an elaborated control strategy is also needed. Within this bachelor thesis different controllers shall be developed and assessed for the dedicated cooling system. The system under consideration consists of one simple liquid cooling cycle where the heat acquired in the aircraft is rejected to ambience via a so-called ram air channel and a second cooling cycle, which is linked to the same ram air channel, via a vapour cycle, such providing cooling below ambient. For each cold consumer there exist minimum and maximum temperatures at in– and outlet which have to be kept through all flight conditions. The designed controller shall not only be able to assure these temperatures for all operating conditions but also to minimise the system energy consumption.

In more detail the following items shall be investigated:

- Development of a suitable model in order to develop and test different controllers
- Description of the relevant theoretical background
- Development of different controllers in order to obtain the required temperatures with the lowest energy consumption possible
- Definition of a suitable set of test cases in order to assess the controllers' performance
- Assessment of the different controllers

Hamburg-Harburg, den 15. September 2008

Prof. Dr.-Ing. G. Schmitz

Abstract

In this thesis different controller synthesis methods and control strategies are analysed with respect to their applicability to a newly developed cooling cycle architecture, which is designed for usage in commercial aircrafts. The novelty of the new cooling system consists in the utilisation of a ram air channel instead of drawing air in from the turbine engines. An increased demand for control effort and electronics is a trade off for higher efficiency in turbine operation. To determine the controllability of the architecture under development, a non-final version is the subject of this thesis. Physical modelling of the plant, as well as the design of a heuristically tuned PI controller structure, $\mathcal{H}_2$ *norm based LQG controller synthesis, LQG gain scheduling and a brief consideration of* $\mathcal{H}_\infty$ *norm based robust LQG controller synthesis is covered. A theoretical chapter on fuzzy control offers a different view on heuristic controller synthesis and a possible steering level gain scheduling solution.*

Contents

Nomenclature

Classical Control Theory

$\boldsymbol{A}$	A state space model's system matrix	
$a^0, \boldsymbol{a^0}, \boldsymbol{A^0}$	Scalar of nominal values, vector of nominal scalar entries, Matrix of nominal scalar entries	
$\boldsymbol{B}$	A state space model's input gain matrix	
$\boldsymbol{C}$	A state space model's output gain matrix	
$\boldsymbol{D}$	A state space model's disturbance gain matrix	
$\boldsymbol{d}$	Disturbance vector, $\boldsymbol{\Delta d}$ denotes small deviations from equilibrium position $\boldsymbol{d^0}$	
$\boldsymbol{K}(s)$	Controller	
$\boldsymbol{P}(s)$	Generalised Plant	
τ	Time constant, typically defined for first order systems	[s]
τ^*	Normalised time constant	[kg]
$\boldsymbol{u}$	Input vector, $\boldsymbol{\Delta u}$ denotes small deviations from equilibrium position $\boldsymbol{u^0}$	
V	Cost Function	
$\boldsymbol{w}$	Fictitious input to generalised plant.	
$\boldsymbol{x}$	State vector, $\boldsymbol{\Delta x}$ denotes small deviations from equilibrium position $\boldsymbol{x^0}$	
$\boldsymbol{y}$	Output vector, $\boldsymbol{\Delta y}$ denotes small deviations from equilibrium position $\boldsymbol{y^0}$	
$\boldsymbol{z}$	Fictitious output of generalised plant.	

Fuzzy Control Theory

A_i^j	Fuzzy set for i^{th} input associated with j^{th} linguistic value
$\tilde{A}_i^j, \tilde{B}_i^p$	Linguistic value

$\hat{A}_i^j$	Fuzzy set with dynamically changing membership function	
B_i^p	Fuzzy set for i^{th} output associated with p^{th} linguistic value	
$\mathcal{F}(u_i)$	Fuzzification operator for input u_i	
$\mathcal{F}^s(u_i)$	Singleton fuzzification operator	
$\mu_{A_i^j}(u_i)$	Membership function associated with fuzzy set A_i^j	
$\mathcal{U}_i$	Universe of discourse of the i^{th} controller input	
u_i	i^{th} controller input	
$\tilde{u}_i, \tilde{y}_i$	Linguistic variable	
$\mathcal{Y}_i$	Universe of discourse of the i^{th} controller output	
y_i	i^{th} controller output	

Hydraulics

A	Cross section area	$[\mathrm{m}^2]$
d	Characteristic length/diameter	$[\mathrm{m}]$
H	Pump head	$[\mathrm{m}]$
H^*	Normalised pump head	
H_R	Rated pump head	$[\mathrm{m}]$
$K_v(v)$	A control valves resistance coefficient dependent on the opening ratio v ranging from 0...1.	
λ	Pipe resistance coefficient	
L	Hydraulic inductance/Fluid inertia	$[\frac{1}{\mathrm{m}}]$
$\dot{m}$	Mass flow rate	$[\frac{\mathrm{kg}}{\mathrm{s}}]$
μ	Absolute roughness	$[\mathrm{m}]$
ν	Kinematic viscosity	$[\frac{\mathrm{m}^2}{\mathrm{s}}]$
n	Rotational speed	$[\frac{1}{\mathrm{s}}]$
n_R	Rated pump speed	$[\frac{1}{\mathrm{s}}]$
n^*	Normalised pump speed	

Δp	Pressure difference	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
Re	REYNOLDS number	
ϱ	Density	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
R	Hydraulic resistance	
R_{lam}	Hydraulic resistance under laminar flow conditions	$[\frac{\mathrm{m}}{\mathrm{s}}]$
R_{turb}	Hydraulic resistance under turbulent flow conditions	$[\frac{1}{\mathrm{kg}\cdot\mathrm{m}}]$
$\dot{V}$	Volume flow rate	$[\frac{\mathrm{L}}{\mathrm{s}}]$
$\dot{V}^*$	Normalised pump volume flow rate	
v	Opening ratio of a control valve. v may range from 0...1	
v_{ref}	Reference opening ratio of a control valve, taken as an input signal. v_{ref} may range from 0...1	
$\dot{V}_R$	Rated pump volume flow rate	$[\frac{\mathrm{L}}{\mathrm{s}}]$
w	Fluid flow velocity	$[\frac{\mathrm{m}}{\mathrm{s}}]$
ζ	Hydraulic resistance coefficient	

Thermodynamics

α	Temperature gradient	$[\frac{\mathrm{K}}{\mathrm{m}}]$
c_p	Heat capacitance ats constant pressure	$[\frac{\mathrm{J}}{\mathrm{kg\,K}}]$
c_v	Heat capacitance at constant volume	$[\frac{\mathrm{J}}{\mathrm{kg\,K}}]$
$\Delta\vartheta_m$	Mean logarithmic temperature difference	$[\mathrm{K}]$
ΔT	Temperature difference	$[\mathrm{K}]$
kA	Heat exchanger's characteristic value	$[\frac{\mathrm{J}}{\mathrm{K\,s}}]$
M	Mass	$[\mathrm{kg}]$
$\dot{Q}$	Heat flow rate	$[\frac{\mathrm{J}}{\mathrm{s}}]$
T	Temperature	$[\mathrm{K}]$

Miscellaneous

$\boldsymbol{J_f}$	The Jacobian, defined as $\boldsymbol{J_f} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$

Abbreviations

COG	Center of gravity
HE	Heat exchanger
LFT	Linear fractional transformation
LFR	Linear fractional representation
LMI	Linear matrix inequality
LPV	Linear parameter-varying
LQG	Linear quadratic gaussian
LQR	Linear quadratic regulator
MIMO	Multiple-input-multiple-output
MISO	Multiple-input-single-output
MOET	More Open Electrical Technologies
PID	Proportional Integral Differential
SISO	Single-input-single-output
TSK	Takagi-Sugeno-Kang

1. Introduction

In our modern society, the demand for more efficient engineering solutions continually rises due to economical and ecological reasons. In the course of the European research program "More Open Electrical Technologies" (MOET), Airbus Germany and the Institute of Thermofluiddynamics of the Hamburg University of Technology are developing cooling architectures, which shall abide by these increased efficiency standards. The novelty of the new cooling system consists in the utilisation of a *ram air channel* instead of drawing air in from the turbine engines. While the latter is an approved method, the new ram air architecture aims for significantly higher efficiency values, to the cost of an increased amount of electronics and control effort. Different architectures are evaluated with regard to weight, drag and power consumption, in order to find an optimised design. An important factor within the evaluation process is to determine, whether and how a given system can be controlled in an effective, efficient and safe way.

Proportional-integral-differential (PID) controllers, for example, are widely used in industrial applications. They provide basic and well developed methods for intuitive and computational tuning. often, their implementation is particularly easy and they are capable of achieving basic to even very rigorous design objectives, depending on the nature of the plant. Most design methods are confined to single-input-single-output (SISO) systems, though. For multivariable systems, however, even simple model-based optimal control approaches already yield very promising opportunities to optimise the transient behaviour as well as the energy consumption.

The purpose of this thesis is to investigate different possible controller designs and strategies using a simple cooling cycle as an example. The following introductory example will anticipate some benefits and design issues, such as suppression of measurement noise, inherent to some controller synthesis approaches presented in this thesis.

1.1. Introductory Example

Consider the following simple mass-spring combination as an example (figure 1.1.1). The cart moves frictionless on the ground and a force u can be applied to it in both directions. The system is governed by the differential equation

$$m\ddot{x} + kx = u.$$

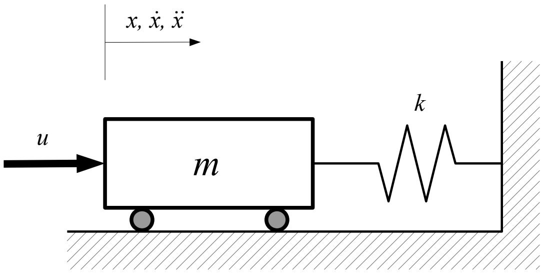


Figure 1.1.1.: Simple Mass-Spring Combination

It can be written in state space form:

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$
$$y = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix}.$$

From an initial deflection, it is the goal to bring the cart back to equilibrium position as quickly as possible with a minimum of mechanical work applied. The closed-loop performance is to be assessed for three different controllers: A PID controller heuristically tuned with the help of Ziegler-Nichols tuning rules, a state feedback controller and a Mamdani fuzzy controller. It is assumed, that the measurement of the mass block's position x and velocity $\dot{x}$ is prone to gaussian white noise w_y, which — for a start — enters the controller unfiltered. The general control loop is depicted in figure 1.1.2.

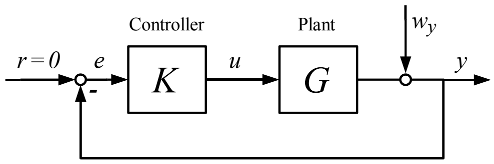


Figure 1.1.2.: General Control Loop for Mass-Spring Combination

Figure 1.1.3 shows plots of the respective closed-loop system responses as well as of the controller outputs. All controllers show good closed-loop behaviour in terms of the cart's position. Indeed, the controllers have been deliberately tuned, such that the respective performances are reasonably similar. Therefore model-based (LQR) as well as heuristic (Ziegler-Nichols PID, Fuzzy) controller synthesis methods are capable of controlling this simple system with approximately the same effect.

The controller output, however, reveals the relevant differences of the closed-loop behaviour: The PID controller amplifies the measurement noise, such that the output

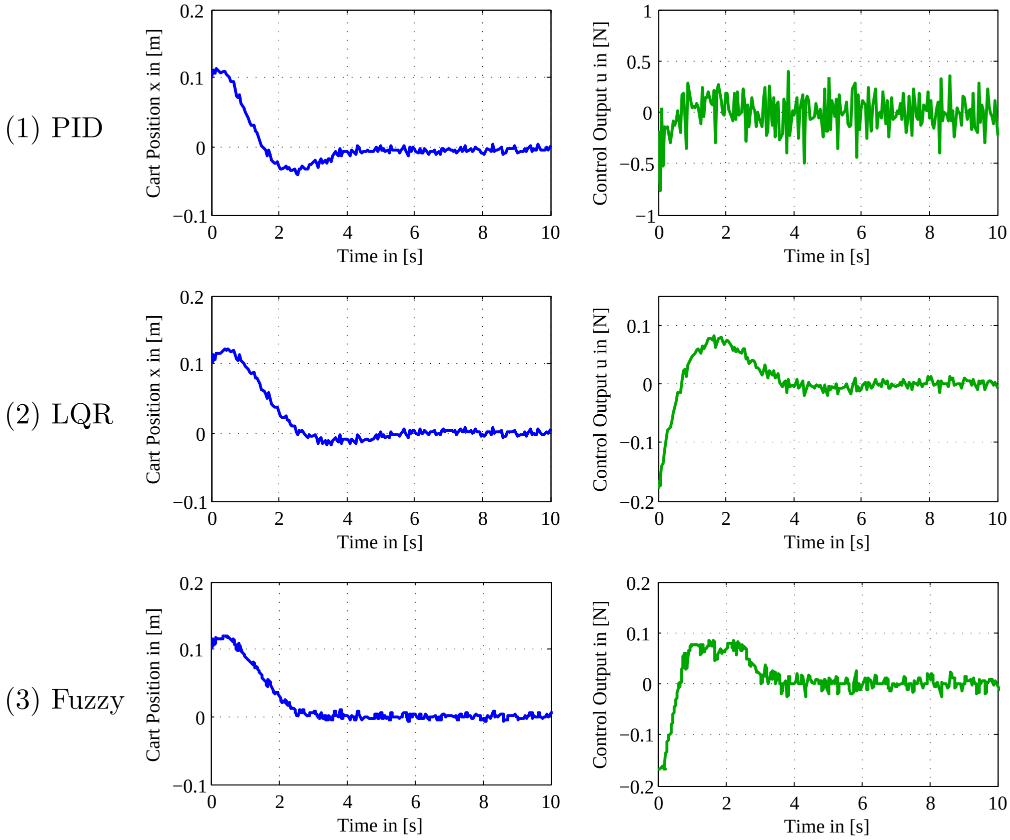


Figure 1.1.3.: Controlled Mass-Spring Combination Simulation Results (Unfiltered Measurement Output) for (1) Ziegler-Nichols Tuned PID Controller, (2) Linear Quadratic Regulator (LQR) Controller, (3) Mamdani Fuzzy Controller

shows heavy oscillations, which render the controller infeasible. This is due to the proportional and differential gain. The fuzzy controller only slightly amplifies the noise, whereas the LQR controller practically lets the noise pass unchanged. A downside to the LQR and fuzzy controller is their need for an additional controller input $\frac{d}{dt}e = \dot{e}$, though.

Filters may be utilised to alleviate the influence of measurement noise. The PID and fuzzy controller receive a low-pass filtered output signal of the plant, while a KALMAN filter is being employed to realise state estimate feedback control with integrated filtering of measurement noise. Figure 1.1.4 shows plots of the augmented control loops.

It can be observed, that all controller outputs become almost perfectly smooth. The controller performance slightly deteriorates, though.

Overall, the model-based LQG controller performs best in this case, because the mechanical work done is less than with the other controllers. The model-based approach clearly shows its advantages with respect to optimality in this example.

It will be a matter to investigate, to which extent the results found here hold true in case of the cooling cycle.

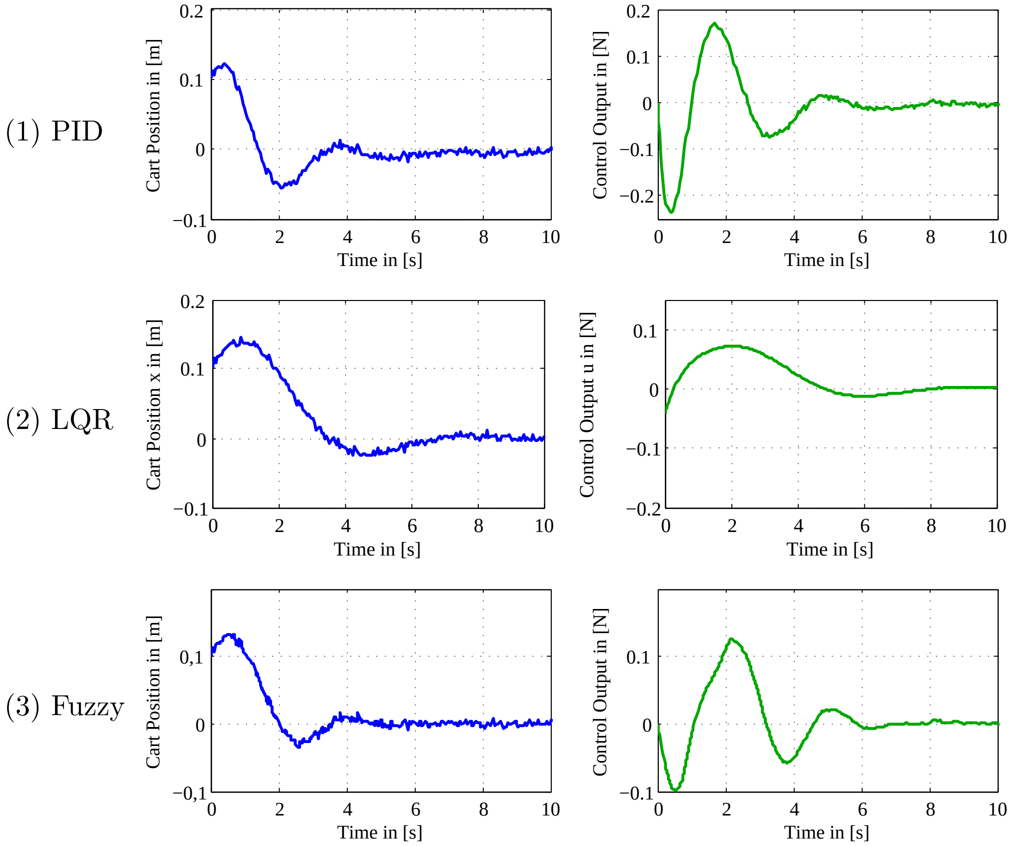


Figure 1.1.4.: Controlled Mass-Spring Combination Simulation Results (Filtered Measurement Output) for (1) ZIEGLER-NICHOLS Tuned PID Controller, (2) Linear Quadratic Gaussian (LQG) Controller, (3) MAMDANI Fuzzy Controller

1.2. Aims of the Thesis

This thesis aims at providing an analysis of the applicability and performance of a selected range of controllers. The evaluation will depend mainly on specific closed-loop design objectives, like given constraints on maximum and minimum plant outputs, energy optimality and robustness. These will be properly defined in the course of this thesis. However, the amount of rigorous mathematical analysis presented will be limited. This is to a large extent due to the non-linear characteristics of the plant under consideration. On the other hand, this thesis is more intended to focus on practical control issues such as a trade off between performance and rejection of measurement noise. The importance of considering such effects has already been hinted at in the introductory example. More specifically, the aims of this thesis are comprised of the following items:

- **Build-Up of Physical Understanding of the Plant**
  The cooling plant under consideration has been subjected to many changes during the work on this thesis. As a consequence, this thesis develops techniques of modelling the physical relationships in a reasonably accurate way. They are intended to apply to changed architecture just as well or with only minor modifications. It is the aim of this thesis to provide a thorough physical understanding of the dynamics inherent to the plant. This knowledge does not only enable for the construction of a non-linear simulation, but it also greatly facilitates the controller synthesis, be it in a model-based or model-free approach.

- **Construction of Non-Linear Simulation for Testing Purposes**
  Since for non-linear plants a rigorous mathematical closed-loop analysis is very difficult, testing a certain controller by non-linear simulation is common practice. As the *Flowmaster* software provides only insufficient tools for controller implementation, a non-linear simulation is constructed in *MATLAB/Simulink*.

- **Derivation of Linearised Plant Models for Controller Synthesis**
  In order to synthesise model-based controllers, that are optimal in the sense of a given cost functional, this thesis investigates and proposes a way to derive a linear plant model. The goal of model-based controller synthesis in this thesis is to justify the considerable amount of work involved to derive mathematical plant models. This thesis aims at evaluating, whether or not the benefits are significant with respect to energy consumption and closed-loop performance. Due to the temporary character of the architecture, the controllers are not needed to be optimally tuned. Their potential is made obvious and is subject to discussion, though.

- **Assessment and Comparison of Different Controllers**
  The controllers are assessed within the framework of unified test cases. This is done in order to provide a maximum amount of comparability. This thesis' intent consists of a thorough assessment of various controllers of practical relevance. While many more possibilities of choosing more specialised and sometimes more complex controller types exist, it is evaluated in which way more basic controllers are feasible for the control of the plant. A concluding discussion briefly treats controller

types that promise to amend remaining issues of the control loops considered more thoroughly in this thesis.

- **Description of Theoretical Background**
  This thesis aims at providing the necessary theoretical background to reproduce all analysis and derivations done. It is the author's goal to apprehensively explain relevant items, such that this thesis may be understood and utilised as the first step to a final controller design for a final cooling architecture.

1.3. Outline of the Thesis

After this introduction, the thesis begins with a chapter about the plant under investigation. After a short description of the plant's basic dynamics and its purpose, the physical equations, that govern the non-linear behaviour, are derived and explained in depth in sections 2.2 to 2.4. The system will be divided into three subsystems, that reflect the main functional aspects. A full non-linear simulation is possible with the information provided thus far, which is used throughout the remainder of the thesis. Since the practical applicability of any controller design has to be assured, a brief section also deals with possible difficulties involved with the measurement of certain plant parameters or signals (section 2.5.3). It will become obvious that full information state feedback is no possible choice for the controller design. The chapter concludes with a brief validation (section 2.6) by means of selected transients, which are compared to those realised by the industry standard software *Flowmaster*.

The third chapter represents the core of this thesis, since the actual controller designs are carried out and discussed there. Before this is done, the desired closed-loop properties and the test setups of the non-linear simulations are thoroughly defined (section 3.1). After that, the thesis proceeds to the description of a PID anti-windup controller design, which is heuristically tuned with the help of Ziegler-Nichols tuning rules. A discussion covers the main issues inherent to this approach. In order to take advantage of model-based controller synthesis approaches, a linearised plant model is derived in the following section 3.3. This includes to settle for a specific set of plant in– and outputs and the justification of a decentralised controller scheme, mainly due to time-varying time-delays inherent to the mass– and temperature transport. Before system equilibria are briefly discussed, a non-linear transformation of certain plant input variables is proposed to further remove non-linearities and enhance the accuracy of the linear model. With the linear models at hand, a LQG controller is designed within an $\mathcal{H}_2$ norm based controller synthesis framework in section 3.4. A quick and formal analysis of the linearised plant precedes a brief description of tuning the controller parameters. A discussion of the LQG controller reveals significant advantages as well as disadvantages due to the model-based approach. The following section 3.5 takes the LQG approach to the next level by introducing a simple, yet effective, gain scheduling algorithm, which greatly facilitates energy optimal control. The discussion of this approach mainly focusses on safety issues and how they might be taken care of. Robust control is considered to be a possible way, whose basic theory is briefly explained in the next section 3.6. Various ways of employing the concept of uncertainty into a robust LQG controller design are proposed and a short mention of application issues concludes the section. A final section summarises the evaluations of the different controllers and provides a clearly laid out comparison.

The fourth chapter covers fundamental fuzzy control theory, a different way of synthesising controllers based on heuristics. After defining the basic terms and nomenclature in section 4.1, some basic mechanisms of fuzzy systems are explained using the MAMDANI controller as an example. To motivate a deeper understanding of fuzzy control systems, the concept of general fuzzy systems and the TAKAGI-SUGENO-KANG (TSK) controller is briefly mentioned in section 4.1.9. The basic MAMDANI controller is then utilised to provide a short design sketch on the fuzzy control of the mass-spring combination used in the introductory example (section 4.2). The chapter continues with a general evaluation of the fuzzy control approach as compared to "classical control". Section 4.3 aims at providing an assembly of opinions commonly found in literature, as well as additional aspects based on the fuzzy control approach that relies on the principles explained in this thesis. Finally possible applications of fuzzy control to the cooling cycle are considered.

The fifth and final chapter summarises the preceding chapters and discusses possibilities of future work on the cooling plant as an outlook.

The appendices aim at furnishing the reader with the theoretical background necessary to understand the theory behind this thesis, including hydraulics, norms on the international standard atmosphere and control theory. Explicit equations of the physical plant modelling are assembled, as well as the linearised equations used for controller synthesis. Finally the contents of the accompanying CD-ROM and their usage are explained. They contain relevant files, which have been created during work on this thesis.

2. Physical Modelling of the Plant

This chapter will provide a detailed description of the system's governing equations, from which a non-linear simulation model has been derived. Starting with a general description of the complete system and its basic dynamics, the following sections will divide the plant into subsystems from the views of different, yet related, engineering disciplines. The goal of describing the individual subsystems in a mathematical way is to gain knowledge of the transient behaviour and to obtain a mathematical basis on which several controller synthesis approaches may depend.

Finally, the model will be validated by means of comparing selected transients that have been generated with the help of *MATLAB/Simulink* with those of the simulation software *Flowmaster*.

Note that the final architecture is still subject to change. Therefore all modelling approaches will be formulated in a most general way.

2.1. Description of the Plant

The complete architecture currently incorporates two cooling cycles: One of these is directly connected to the *ram air channel* via a *heat exchanger*. The other one has an additional *vapour cycle* connected upstream of it. Since the structures of the actual cooling loops are basically the same, this thesis will focus on the first one. The major difference, besides the direct connection to the ram air, consist in the different types of loads requiring different amounts of cooling. This difference, however, does not affect the modelling approach, as it would only mean a change of value in certain variables. The vapour process, of course, would require additional attention both in terms of modelling and controller synthesis investigations, but this has been postponed to future work for simplicity in the matter of this thesis.

Figure 2.1.1 shows the part of the system under investigation. The air flow in the ram air channel will either be provided by the natural inflow due to the aircraft velocity or it can be forcefully propelled into it by two fans F_1 and F_2, which would typically be used on ground level. In aircraft technology, almost everything is designed with the principle of redundancy in mind. Be it for reserve machine capacities or for simultaneous usage, at least one additional piece of machinery or electronics is needed to ensure a specific amount of *fail safe operation*.

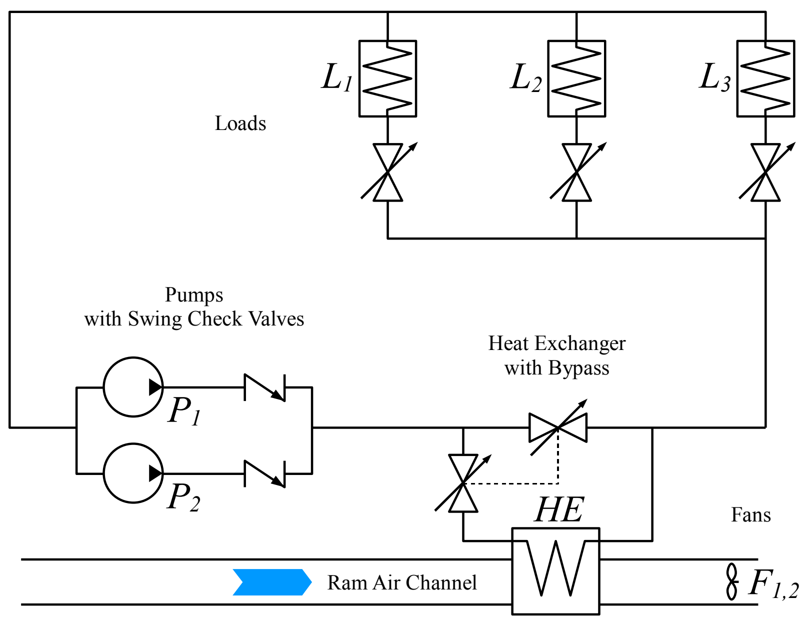


Figure 2.1.1.: Overview of the Plant

The ram air's temperature is affected by the surrounding atmosphere and the aircraft's velocity. During flight, the air heats up due to the transformation of kinetic energy at the entrance to the channel. It is still generally cooler than the *cooling liquid* in the pipes of the cycle, which is a mixture of *Propylene-Glycol* and water. Heat is transferred from cooling liquid to ram air inside the heat exchanger. In the cooling network two redundant *radial pumps* make sure that the liquid circulates. Swing check valves only allow the fluid to flow in one direction, not unlike a diode inside an electrical network.

The heat exchanger can be bypassed, such that the amount of cooling by the ram air flow can be controlled. In figure 2.1.1 this has been visualised by two separate — but complementarily coupled — valves, which means, that while one is $\Gamma\,\%$ open, the other one is open to $100 - \Gamma\,\%$. A lower temperature on the outlet of the heat exchanger bypass circuitry will lead to more effective cooling of the loads.

The loads represent the aircraft's units that need cooling. These can first and foremost consist of *power electronics*, which are vital for the aircraft's operability. The second cooling cycle mentioned earlier contains loads such as galleys or entertainment electronics, which are devices that are less important for the passengers' safety. The cooling fluid flow will split into fractions determined by the opening ratios of the control valves preceding the loads. The higher the mass flow rates entering the load branches, the better the cooling. Adversely a higher flow rate inside the pipes leads to increased friction, particularly in the valves, which produces heat. Additionally the pumps also generate heat from losses and transfer this to the fluid.

With these basic dynamics observed, it is obvious that controlling the plant demands to find the optimal trade off between applied pump power and valve positions to obtain an *energy optimal control* strategy.

The following sections will now depict the aforementioned dynamics in a mathematical manner. Therefore the system has been decomposed into three subsystems: A *hydraulic*, a *ram air channel* and a *thermodynamic* subsystem.

More specifically, the division into the *three subsystems* aims at the following capabilities of the simulation model:

- Ability to describe the transient response of the plant to changes in pump revolutions per second (see section 2.2.3).
- Ability to determine the individual mass flow rates flowing through the three different loads and the heat exchanger depending on valve opening ratios. (see section 2.2.4)
- Ability to account for the transient response in air mass flow due to changes in fan revolutions per second. (see section 2.3)
- Ability to simulate the temperature changes of the cooling fluid. (see section 2.4)
- Ability to approximate different flight conditions and to observe a complete flight envelope with changing environmental conditions. (see section 2.3.3 and B)

2.2. The Hydraulic Subsystem

Figure 2.2.1 again shows the plant under consideration, but with additional symbolic elements, that describe a first approximation of the system's *dynamic properties*:

The heat exchanger and loads are replaced by *hydraulic resistances* R_{L_i}, R_{HE}, whereas the new symbols R_p, R_{p_b} and $R_{p_{HE}}$ have been introduced to represent the pipes' hydraulic resistances. The control valves are modelled in the form of the variable resistances R_{v_i}, R_{v_b} and $R_{v_{HE}}$.

For each pipe element a *fluid inertia* L_p, L_{p_b} and $L_{p_{HE}}$, respectively, has also been introduced. Note that the subscript $_p$ denotes that the respective symbol belongs to a certain pipe element.

The pumps are now represented by pressure sources. These, however, will not be modelled as ideal, but as sources $\Delta p_{P_i}(\dot{m}_{P_i}, n_{P_i})$, that depend on the current mass flow rate and pump speed (refer to subsection 2.2.3).

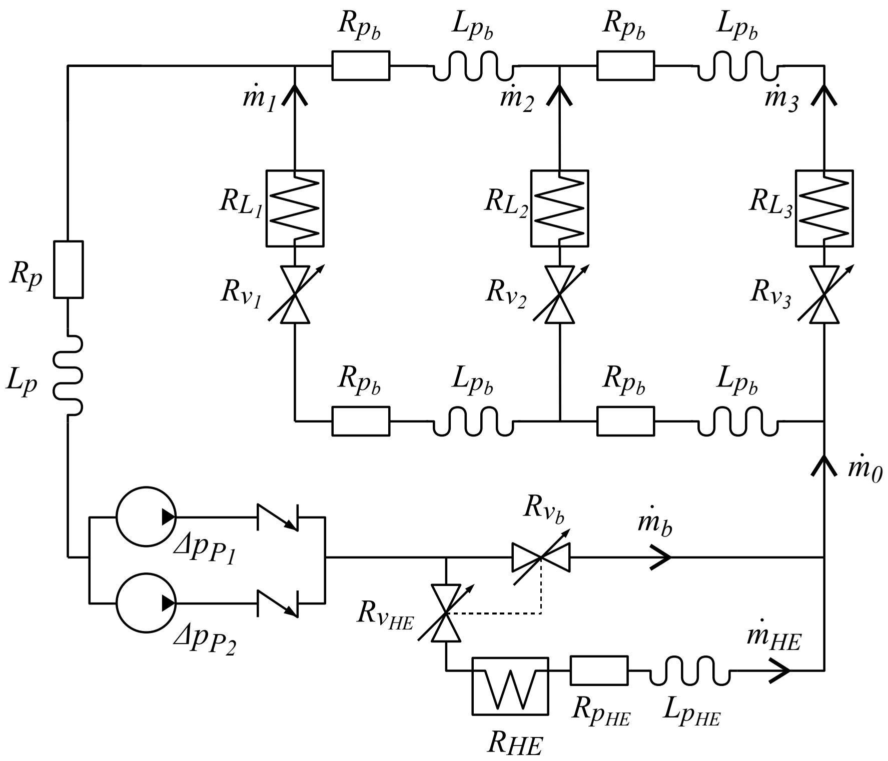


Figure 2.2.1.: Hydraulic Subsystem

2.2.1. Initial Assumptions

For the calculation of the hydraulic network, initial assumptions have to be made and verified, in order to determine, whether the system has turbulent or laminar flow behaviour and to compute the corresponding coefficients:

According to [20] the Reynolds number inside a circular pipe under estimated flow conditions can be calculated as follows:

$$\mathrm{Re} = \frac{\bar{w} \cdot d}{\nu} \tag{2.2.1}$$

with $\bar{w} = 0.8 \cdot w_{max}$ as the average flow velocity
d as the pipe's characteristic length (i.e. the diameter)
ν as the fluid's kinematic viscosity

In case of an assumed maximum velocity of $w_{P_{max}} \approx 3\,\mathrm{m/s}$ and a typical diameter of $d_p = 0.01905\,\mathrm{m}$ the Reynolds number ranges well below the critical border of $\mathrm{Re}_{crit} = 2300$ to turbulent flow:

$$\mathrm{Re} \approx \frac{0.8 w_{P_{max}} \cdot d_p}{\nu_P} = 653.$$

Although this would mean, that it is not necessary to compute the hydraulic network with turbulent flow, the geometry of heat exchangers, valves and pipe bends will inevitably lead to turbulences. Moreover, laminar flows may be calculated in the same way as turbulent ones, thus the system will be considered turbulent.

The resistance coefficient λ_p of all pipe elements can then be obtained by calculating the relative roughness d_p/μ_p and extracting the corresponding value from a diagramm (see A.1.1). This yields:

$$\lambda_p \approx 0.06$$
$$\text{since} \quad \frac{d_p}{\mu_p} = 762.$$

2.2.2. Simplifications

Until now, every pipe element has been described by a hydraulic resistance and a fluid inertia. Considering the inertia, it is only necessary to include one, if it significantly affects the *transient behaviour*, or more specific, if the resulting *time constant* is large enough to account for an extra delay in fluid acceleration.

With respect to hydraulic resistances, it has to be investigated, whether each pipe segment accounts for a significant pressure drop and/or, if the system's structure can be modelled in a simpler, more convenient way, without introducing too much of an error.

The following sections will describe in which way and under which conditions several of the hydraulic elements can be neglected.

Fluid Inertia

Fluid inertia is responsible for the transient behaviour of the mass flow rate inside the pipe network. To accurately model a pipe segment, it can be decomposed into a hydraulic resistance subject to pipe geometry and a fluid inertia of the respective amount of fluid contained inside the pipe segment. It may, however, not be necessary to consider the inertia inside each single pipe element. Instead, the way in which the inertia adds up and the individual magnitudes can be compared in order to determine negligible transient effects.

The current architecture incorporates short pipe branches between the loads of only $l_{p_b} = 1\,\mathrm{m}$ length and with a cross section diameter of $d_{p_b} = 2.5 \cdot 10^{-3}\mathrm{in} = 0.00635\,\mathrm{m}$.

The resulting hydraulic fluid inertia and hydraulic resistance is therefore (please refer to appendix A for some fundamental formulae in hydraulics):

$$L_{p_b} = \frac{l_{p_b}}{A_{p_b}} = 3.158 \cdot 10^4 \frac{1}{\mathrm{m}}$$
$$R_{p_b} = \zeta_{p_b} \cdot \frac{1}{2 \cdot \varrho_P \cdot A_{p_b}^2} = 4.6355 \cdot 10^6 \frac{\mathrm{m}}{\mathrm{kg}}$$
$$\text{with } \zeta_{p_b} = \lambda_p \cdot \frac{l_{p_b}}{d_{p_b}} = 9.8562$$
$$\text{and } A_{p_b} \text{ being the diameter of the circular cross section: } A_{p_b} = \pi \cdot \frac{d_{p_b}^2}{4}$$

It is assumed, that the response of these small pipe segments will be relatively fast compared to that of the longer pipes between loads and pumps. To test, whether this assumption holds, the step response of a first order differential equation for this pipe segment will be investigated:

$$\Delta p_{p_b} = R_{p_b} \dot{m}_{p_b}^2 + L_{p_b} \ddot{m}_{p_b} \tag{2.2.2}$$

Δp_{p_b} is an *ideal pressure source*, driving the mass flow rate $\dot{m}_{p_b}$ through the pipe.

When looking at the step response with input $\Delta p_{p_b} = 10^5 \frac{\mathrm{N}}{\mathrm{m}^2}$ applied, the similarity to a first order system's response becomes apparent (see figure 2.2.2).

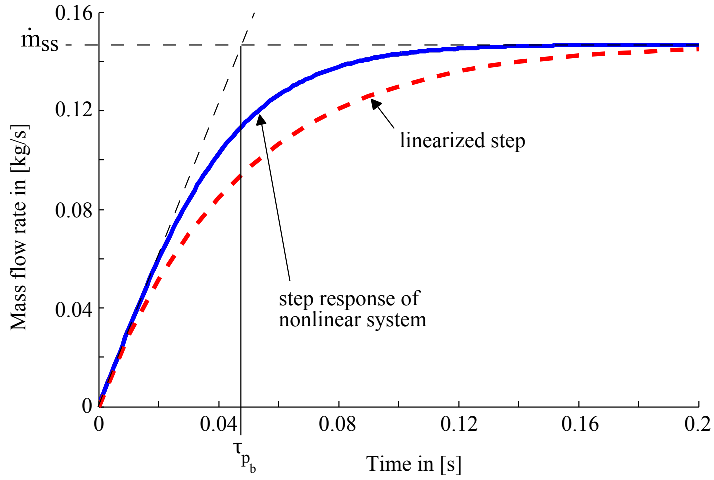


Figure 2.2.2.: Step Response to Non-Linear and Linearised Differential Equation of Pipe Segment

First order systems are well known with regard to their transient properties and how to quantify them, while non-linear systems generally are not. However, a linearisation around a certain operating point often allows for such a quantitative analysis even for non-linear systems. Approximating the term $R_{p_b}\dot{m}_{p_b}^2$ by a straight line from origin to steady state $\dot{m}_{ss}$ gives:

$$\Delta p_{p_b} = R_{p_b}\dot{m}_{ss} \cdot \dot{m}_{p_b} + L_{p_b}\ddot{m}_{p_b}$$

This actually yields the same transfer function, as if the time constant τ_{p_b} is taken from the step response of the non-linear system by graphical means:

$$\frac{\dot{M}_{p_b}(s)}{\Delta P_{p_b}(s)} = \frac{\frac{1}{\dot{m}_{ss}\cdot R_{p_b}}}{\frac{L_{p_b}}{\dot{m}_{ss}\cdot R_{p_b}}s+1} \approx \frac{K}{\tau_{p_b}s+1}$$

Figure 2.2.3 illustrates the linearisation.

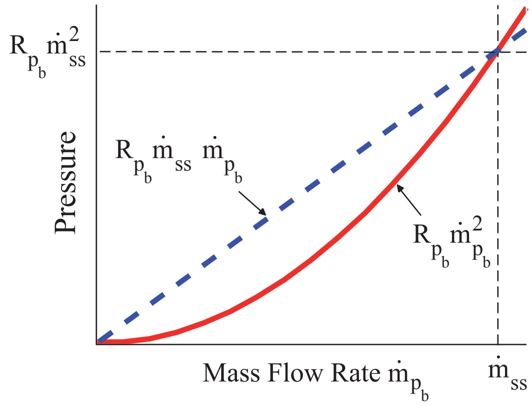


Figure 2.2.3.: Linearisation of Quadratic Turbulent Pressure Curve

In summary, the time constant τ_{p_b} of the linearised system is a function of the desired *steady state value* $\dot{m}_{ss}$ and $\tau_{p_b}^* = L_{p_b}/R_{p_b}$ can be considered the *normalised time constant* of the non-linear step response corresponding to a steady state mass flow rate of $\dot{m}_{ss} = 1\frac{\text{kg}}{\text{s}}$. Using this as a measure for the speed of response for each pipe segment, table 2.2.2 compares the individual normalised time constants.

Pipe Segment	**Normalised Time Constant in** [kg]
load branches	$\tau_{p_b}^* = L_{p_b}/R_{p_b} = 0.0068$
heat exchanger branch	$\tau_{pHE}^* = L_{pHE}/R_{pHE} = 0.1839$
main pipe	$\tau_p^* = L_p/R_p = 0.1839$

Table 2.2.1.: Normalised Time Constants of Pipe Segments ($\tau^* = L/R$)

Clearly, the load branches respond very fast. In addition this will only have an effect on the system's transient response, when the pump speed input changes. Due to the valve dynamics (see 2.2.4), which are slower than the fluid, the neglection of fluid inertia inside the load pipe branches will not be noticeable. With respect to the heat exchanger branch, depending on the valve position, the system's time constant could actually double. Therefore this pipe branch's inertia should be taken into account.

For simplicity the network will be modelled using variable inductance. For a valve opening of $v_{HE} = 1 \widehat{=} 100\,\%$, the resulting overall inductivity will be $L'_p = L_p + L_{p_{HE}}$. For a closed valve it will just be $L'_p = L_p$. Thus the fluid inertia will be modelled by linear interpolation as follows:

$$L'_p = L_p \cdot (1 + v_{HE}) \tag{2.2.3}$$

Hydraulic Resistances

Table 2.2.2 lists the hydraulic resistances of each pipe segment and those of the loads and heat exchanger.

Pipe Segment	**Hydraulic Resistance in** $[1/\,\mathrm{kg\,m}]$
load branches	$R_{p_b} = 4.6355 \cdot 10^6$
heat exchanger branch	$R_{p_{HE}} = 7.6305 \cdot 10^4$
main pipe	$R_p = 3.0522 \cdot 10^5$
load 1	$R_{L_1} = 4.4222 \cdot 10^6$
load 2	$R_{L_2} = 5.8962 \cdot 10^6$
load 3	$R_{L_3} = 5.8962 \cdot 10^6$
heat exchanger	$R_{HE} = 1.4741 \cdot 10^6$

Table 2.2.2.: Comparison of Hydraulic Resistances ($R = \zeta/(2\varrho_P A^2)$)

Apart from the hydraulic resistance of the heat exchanger branch, all of the values range from $0.3...6.0 \cdot 10^6$, which does not suggest any simplifications. However, the calculation of an *H-circuit* under turbulent conditions, i.e. with a pressure law $\Delta p = R \cdot \dot{m}^2$, leads to a non-linear system of equations. In order to obtain an approximation of the load H-circuit, it can be assumed that it can be turned into a *parallel circuit* (see figure 2.2.4).

The cumulative hydraulic resistance now calculates as follows:

$$R'_p = R_{Loads} + R_{HeatExchanger} + R_{MainPipe} \tag{2.2.4}$$

$$\text{with} \quad R_{Loads} = \left(\frac{1}{\frac{1}{\sqrt{R_{p_b}+R_{L_1}+R_{v_1}}} + \frac{1}{\sqrt{R_{p_b}+R_{L_2}+R_{v_2}}} + \frac{1}{\sqrt{R_{p_b}+R_{L_3}+R_{v_3}}}} \right)^2 \tag{2.2.5}$$

$$R_{HeatExchanger} = \left(\frac{1}{\frac{1}{\sqrt{R_{HE}+R_{v_{HE}}}} + \frac{1}{\sqrt{R_{v_b}}}} \right)^2 \tag{2.2.6}$$

For each single path, the fluid may take when only one valve is opened, the corresponding hydraulic resistance of the pipe is now half the resistance as in the H-circuitry. By

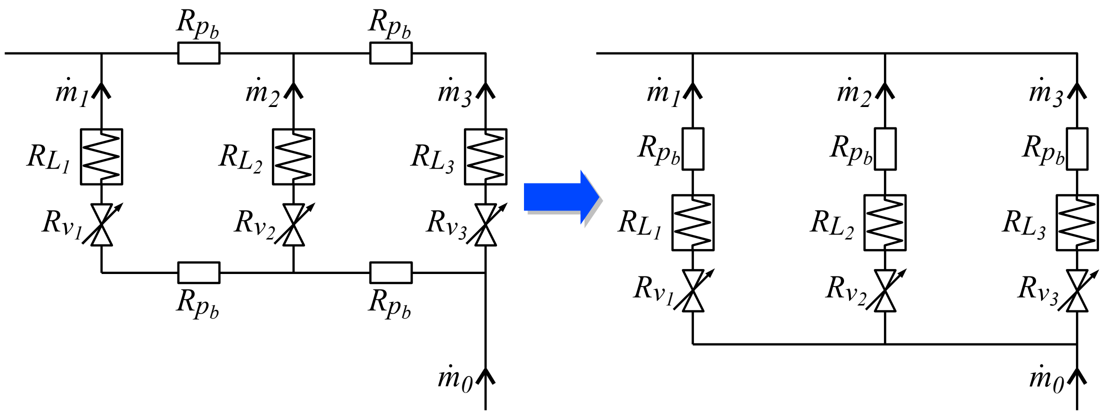


Figure 2.2.4.: Parallelisation of the H-Circuit

introducing a compensation factor $RC > 1$, the resulting discrepancies can be reduced, though.

Summary

The assumptions and deliberate simplifications to the system's structure lead to the network scheme as shown in figure 2.2.5.

In summary the performed simplifications and their implications are:

- Fluid inertia inside small pipe segments has been neglected. Instead the computation of the system's transient response depends on a reduced, system parameter dependent fluid inertia that accounts for the total inertia of the system.
- The system's cumulative hydraulic resistance has been reduced to a combination of individual resistances in series. The individual resistances can be computed as parallel circuits.
- The parallelisation of the H-circuit introduces an error on the system's hydraulic resistance. This error depends on the valve positions. In order to minimise this error, physical parameters could be deliberately multiplied with a small gain depending on the valve openings. For simplicity this gain has been chosen as a constant value.

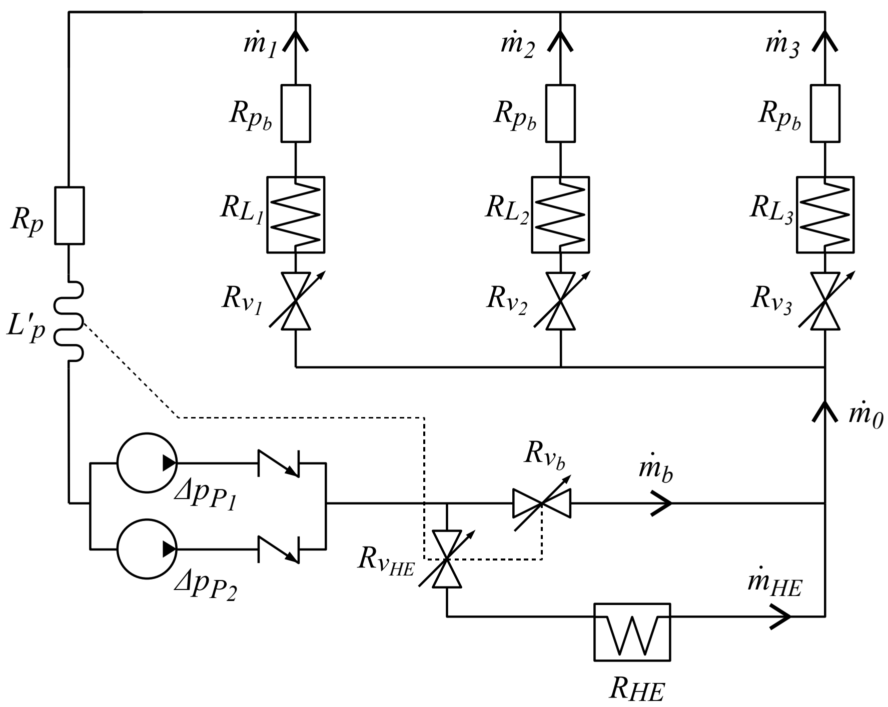


Figure 2.2.5.: Simplified Hydraulic Subsystem

2.2.3. Pump Dynamics

With the hydraulic network simplified, a reduced representation can be seen in figure 2.2.6.

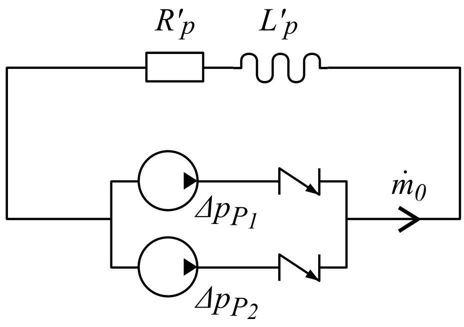


Figure 2.2.6.: Reduced Representation of the Hydraulic Subsystem

The network can first be considered with only one pump and after that, an additional pump can be added via *superposition*.

$$\dot{m}_0 = \dot{m}_{P_1} + \dot{m}_{P_2}$$

Let each pump drive a mass flow m_{P_i} through the network. The subscript $_i$, however, will be dropped for simplicity during the following passage.

The Pump's Characteristic Curve

A pump's capability to exert a certain amount of pressure is defined by a *characteristic curve*, that describes the relationship between pressure difference output Δp_P and the volume flow rate $\dot{V}_P$ at pump entrance. Generally, these characteristic curves often define the relationship not by absolute values, rather than by normalised values:

$$H^* = \frac{H}{H_R} \qquad \dot{V}^* = \frac{\dot{V}}{\dot{V}_R} \qquad n^* = \frac{n}{n_R}$$

with * denoting the normalised values

$_R$ denoting the rated values for which the pump is designed to operate.

This enables the *modelling of pump performance* for arbitrary revolution speeds.

In case of the given architecture a characteristic curve

$$H^* = H^*(\dot{V}^*)$$

with H^* as the *normalised pump head* is given in the form of measured data (see figure 2.2.7). These data points can be approximated by a third order polynomial.

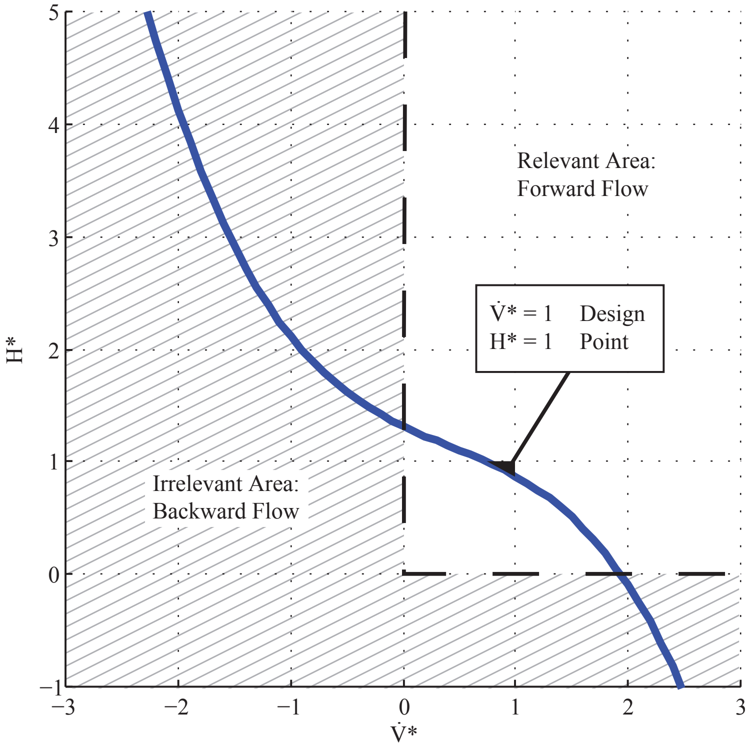


Figure 2.2.7.: Pump's Characteristic Curve Valid for the Rated Speed n^*

The pump is designed to operate at $H^* = \dot{V}^* = 1$, where in general a high efficiency is reached, albeit with some reserves. To obtain the current absolute head H from a given pump speed n and fluid inflow $\dot{V}$ the following equations hold [1, p.9]:

$$\frac{H_1}{H_2} = \frac{n_1^2}{n_2^2} \qquad \frac{\dot{V}_1}{\dot{V}_2} = \frac{n_1}{n_2} \tag{2.2.7}$$

Expanding the fractions by the rated values and also setting the value n_2 to the rated value n_R for which the characteristic curve 2.2.7 is valid gives:

$$H_1^* = H_2^* \cdot n_1^{*2} \qquad \dot{V}_1^* = \dot{V}_2^* \cdot n_1^* \tag{2.2.8}$$

Observing that $n_2 = n_R$ implies, that H_2^* and $\dot{V}_2^*$ are also values on the characteristic curve. The subscript $_2$ now turns to $_R$ and denotes values on the characteristic curve for the rated speed n_R, while the subscript $_1$ can be dropped an denotes head and volume flow for the actual pump speed n. Now the resulting head can be calculated from:

$$H^* = H_R^*(\dot{V}_R^*) \cdot n^{*2}$$
$$H^* = H_R^*\left(\frac{\dot{V}^*}{n^*}\right) \cdot n^{*2} \tag{2.2.9}$$

To calculate the output pressure difference Δp_P from the head a simple equation can be applied:

$$\Delta p_P = H^* \cdot H_R \cdot \varrho_P \cdot g = H \cdot \varrho_P \cdot g \tag{2.2.10}$$

with ϱ_P denoting the cooling fluid's density
g denoting the gravity constant

Figure 2.2.8 shows a block diagram of the transformations needed to use a single given characteristic curve for normalised values to obtain the actual output values, as explained above.

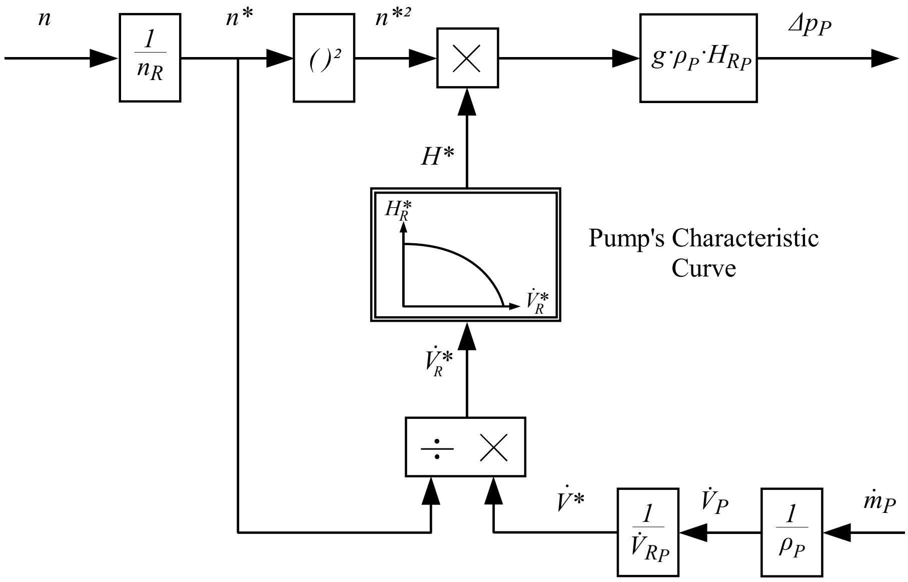


Figure 2.2.8.: Block Diagram of Pump Characteristic Curve Equations

The Governing Equation

With the relation between pump speed n, current mass flow rate $\dot m_P$ and output pressure Δp_P settled, the hydraulic system's response to this pressure difference has to be formulated mathematically. The chosen approach loosely follows [4, p. 285ff].

A *pressure balance* yields:

$$\begin{aligned} \Delta p_P(n, \dot m_P) &= \Delta p_{Resistance} + \Delta p_{Inertia} \\ \Delta p_P(n, \dot m_P) &= R'_p \cdot \dot m_P^2 + L'_p \cdot \ddot m_P. \end{aligned} \tag{2.2.11}$$

And since

$$\Delta p_P(n, \dot m_P) = H^*\left(\frac{\dot m_P^*}{n^*}\right) \cdot \frac{H_R \cdot \varrho_P \cdot g}{n_R^2} \cdot n^2$$

the *governing equation* is non-linear and actually includes two feedback loops. This becomes more apparent, when looking at the corresponding block diagram in figure 2.2.9.

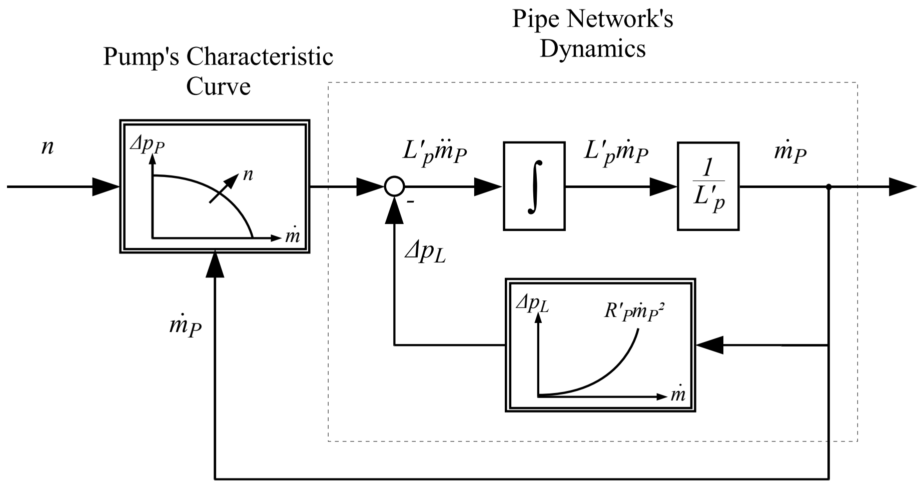


Figure 2.2.9.: Block Diagram of Hydraulic Network Pump Dynamics

Summary

After superposition the complete governing equations have the form:

$$\begin{aligned}
\Delta p_{P_1}(n_{P_1}, \dot{m}_{P_1}) = H^*_{P_1}\left(\frac{\dot{m}^*_{P_1}}{n^*_{P_1}}\right) \cdot \frac{H_{R_{P_1}} \cdot \varrho_P \cdot g}{n^2_{R_{P_1}}} \cdot {n_{P_1}}^2 &= R'_p \cdot \dot{m}^2_{P_1} + L'_p \cdot \ddot{m}_{P_1} \\
\Delta p_{P_2}(n_{P_2}, \dot{m}_{P_2}) = H^*_{P_2}\left(\frac{\dot{m}^*_{P_2}}{n^*_{P_2}}\right) \cdot \frac{H_{R_{P_2}} \cdot \varrho_P \cdot g}{n^2_{R_{P_2}}} \cdot {n_{P_2}}^2 &= R'_p \cdot \dot{m}^2_{P_2} + L'_p \cdot \ddot{m}_{P_2} \\
\dot{m}_0 = \dot{m}_{P_1} + \dot{m}_{P_1}
\end{aligned} \tag{2.2.12}$$

With subscripts $_1$ and $_2$ denoting the respective pump's parameter and output values.

The pumps' *operating points* are the intersections of the network's characteristic curve with the pumps' characteristic curves (see figure 2.2.10). The network's characteristics will change with varying opening ratios of the control valves, or more generally, with varying hydraulic resistance. This introduces a *time-varying parameter* to the system.

Note, that pump friction has been neglected throughout the modelling and the treatment of mechanical dynamics has been excluded from the model. It is assumed that there exist standard controllers for the control of a pump's rotational speed. Thus an input parameter for the pump's speed is sufficient for this thesis' analysis.

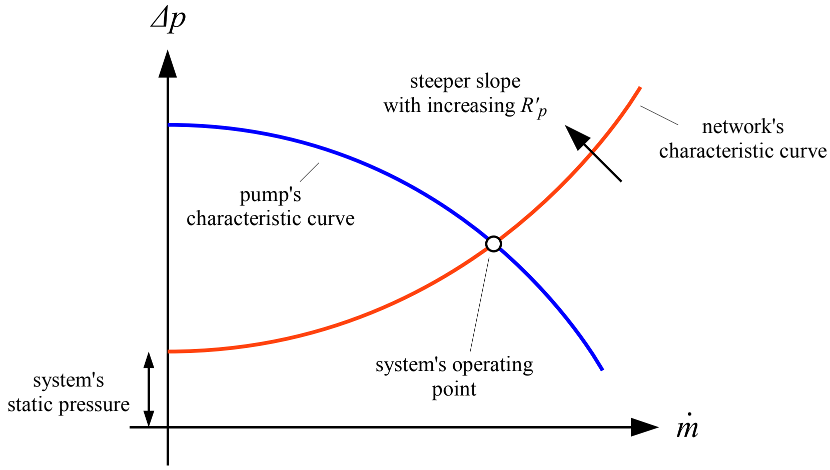


Figure 2.2.10.: Operating Point of a Hydraulic System

2.2.4. Control Valves

When considering the pipe network's *control valves* the important aspects that need attention are threefold:

- The fractioning of the total mass flow $\dot{m}_0$ into mass flows $\dot{m}_1$, $\dot{m}_2$, $\dot{m}_3$, $\dot{m}_{HE}$ and $\dot{m}_b$ can be determined from the opening ratios of the respective control valves.
- Resistance coefficients K_v must be obtained from opening ratios v.
- The valves' delayed response to reference opening positions has to be modelled accordingly. For this purpose a simple first order dynamic behaviour will be assumed.

Fractioning of the Mass Flow

In section 2.2.2 the hydraulic network has been transformed into a network with simple series and parallel connections only. This allows for the application of the *current divider rule* to the network:

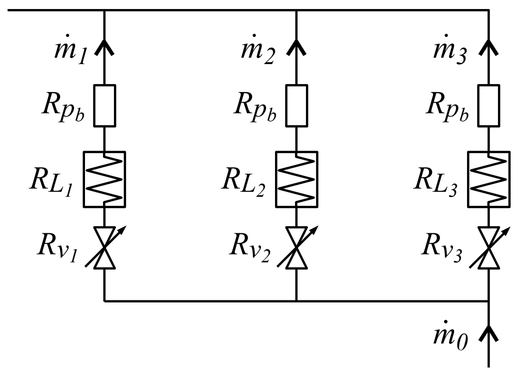


Figure 2.2.11.: Parallel Circuitry of Load Branches

Figure 2.2.11 shows the part of the hydraulic network scheme, where $\dot{m}_0$ splits into $\dot{m}_i$, $i = 1, 2, 3$. For this representation, the current divider rule for turbulent flow yields:

$$\dot{m}_1 = \dot{m}_0 \cdot \frac{\sqrt{R_2 \cdot R_3}}{\sqrt{R_1 \cdot R_2} + \sqrt{R_1 \cdot R_3} + \sqrt{R_2 \cdot R_3}}$$
$$\dot{m}_2 = \dot{m}_0 \cdot \frac{\sqrt{R_1 \cdot R_3}}{\sqrt{R_1 \cdot R_2} + \sqrt{R_1 \cdot R_3} + \sqrt{R_2 \cdot R_3}}$$
$$\dot{m}_3 = \dot{m}_0 \cdot \frac{\sqrt{R_1 \cdot R_2}}{\sqrt{R_1 \cdot R_2} + \sqrt{R_1 \cdot R_3} + \sqrt{R_2 \cdot R_3}}$$

$$\text{with } R_i = R_{L_i} + R_{v_i} + R_{p_b} \quad , i = 1, 2, 3$$

Since for small opening ratios, the resistance values of the valves R_{v_i} are very large compared to the load and pipe resistances R_{L_i} and R_{p_b}, the substituting resistance values R_i reduce to:

$$R_i = R_{v_i} \quad , R_{v_i} \gg R_{L_i}, R_{p_b}$$

Then again, for completely opened valves, the fractions are dominated by the remaining hydraulic resistances:

$$R_i = R_{L_i} + R_{p_b} \quad , \text{as } R_{v_i} \to 0$$

The neglection of the other hydraulic resistances is feasible, since the simulation results actually equal those of the fluid flow simulation software *Flowmaster* far better, if only the ratios of the valve resistances account for the *mass flow fractioning* (refer to section 2.6).

Thus the final equations for the model become:

$$\dot{m}_i = \beta_i \cdot \dot{m}_0 \quad , i = 1, 2, 3$$
$$\text{with } \beta_i = \frac{\sqrt{\frac{R_{v_1} \cdot R_{v_2} \cdot R_{v_3}}{R_{v_i}}}}{\sqrt{R_{v_1} \cdot R_{v_2}} + \sqrt{R_{v_1} \cdot R_{v_3}} + \sqrt{R_{v_2} \cdot R_{v_3}}} \tag{2.2.13}$$

Similarly, equations can be found for the fractioning of $\dot{m}_0$ into $\dot{m}_{HE}$ and $\dot{m}_b$. The corresponding parallel circuit is shown in figure 2.2.12.

$$\dot{m}_{HE} = \beta_{HE} \cdot \dot{m}_0$$
$$\dot{m}_b = \beta_b \cdot \dot{m}_0$$
$$\text{with } \beta_{HE} = \frac{\sqrt{R_{v_b}}}{\sqrt{R_{v_b}} + \sqrt{R_{v_{HE}}}} \quad \text{and } \beta_b = \frac{\sqrt{R_{v_{HE}}}}{\sqrt{R_{v_b}} + \sqrt{R_{v_{HE}}}} \tag{2.2.14}$$

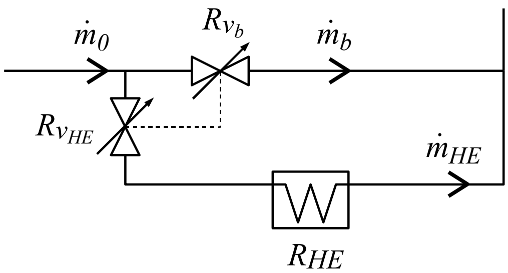


Figure 2.2.12.: Heat Exchanger Bypass Circuitry

Obtaining Resistance Coefficients K_v from Opening Ratios v

Modelling control valves as hydraulic resistances demands — apart from the valve's cross section A_v for opened state — for a *resistance coefficient*, that is a function of the opening ratio v:

$$R_v = K_v(v) \cdot \frac{1}{2\varrho_P A_v^2} \tag{2.2.15}$$

$K_v(v)$ is determined by a characteristic curve as shown in figure 2.2.13.

To obtain a function $K_v(v)$ the measured data is first put on a *logarithmic scale* and then approximated by a *third order polynomial* $\Pi_3(v)$. After that, $K_v(v)$ can be expressed like:

$$K_v(v) = e^{\Pi_3(v)} = e^{p_1 \cdot v^3 + p_2 \cdot v^2 + p_3 \cdot v + p_4} \tag{2.2.16}$$

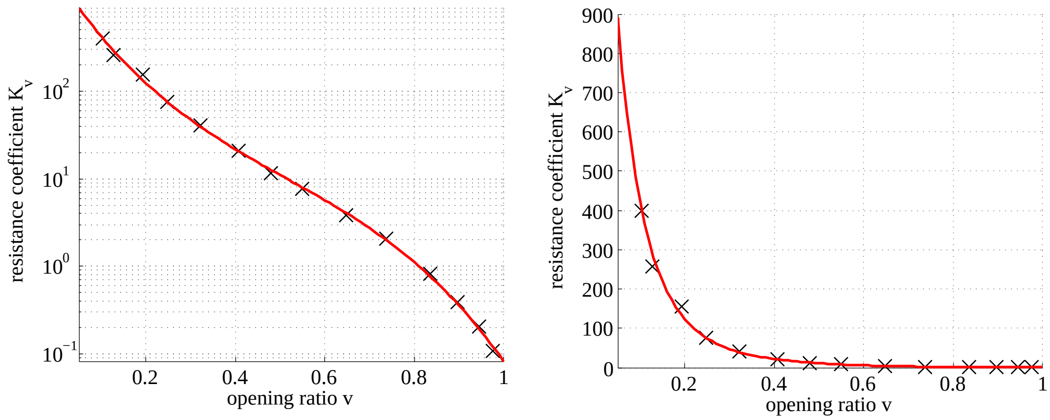


Figure 2.2.13.: Characteristic Curve $K_v(v)$ in Logarithmic and Linear Scale

Transient Behaviour

Under the assumption of a first order transient behaviour with time constant τ_v, the valves' dynamics formulated in a state space model look like:

$$\dot{\boldsymbol{x}}_{\boldsymbol{v}} = \boldsymbol{A}_{\boldsymbol{v}} \cdot \boldsymbol{x}_{\boldsymbol{v}} + \boldsymbol{B}_{\boldsymbol{v}} \cdot \boldsymbol{u}_{\boldsymbol{v}_{\boldsymbol{ref}}} \tag{2.2.17}$$

The subscript $_v$ denotes, that the system matrices and state vectors belong to the control valve state space model.

The time constant is chosen such that the valves assume the desired opening ratios in about 4 seconds. The coupled bypass valve v_b always assumes the complementary state of v_{HE}:

$$v_b = 1 - v_{HE} \tag{2.2.18}$$

2.3. The Ram Air Channel Subsystem

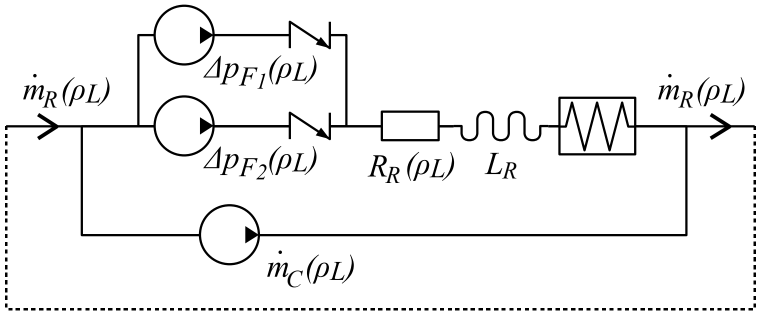


Figure 2.3.1.: Ram Air Channel Subsystem

The ram air channel subsystem as shown in figure 2.3.1 represents the ram air channel's dynamic behaviour with respect to the air mass flow rate. The air flow is determined by the aircraft's velocity and flight height and by the fans' speeds.

The complete mass flow rate $\dot{m}_R$ is a superposition of the mass flow rates that are accounted for by the three pressure sources.

$$\dot{m}_R = \begin{cases} \dot{m}_{F_1} + \dot{m}_{F_2} & \text{, on ground level} \\ \dot{m}_C & \text{, during flight.} \end{cases} \tag{2.3.1}$$

The individual mass flow rates $\dot{m}_{F_1}, \dot{m}_{F_2}$ are governed by equations that will be described in the following sections, whereas $\dot{m}_C$ will simply be simulated by setting a fixed value.

2.3.1. Initial Assumptions

The underlying assumptions to figure 2.3.1 are

- Due to the air's low viscosity, the fluid flow is assumed turbulent inside the ram air channel ($\mathrm{Re} \gg 10^3$ under typical estimated conditions: $w \approx 3\,\frac{\mathrm{m}}{\mathrm{s}}$).
- The fans are modelled as parallel pressure sources. The swing check valves have been introduced in order to provide some coherency with how the separate air mass flow rates will be simply superimposed, as has been done with the pumps of the hydraulic subsystem (refer to section 2.2.3).
- The ram air channel's flow properties are characterized by its fluid inertance L_R and hydraulic resistance $R_R(\varrho_R)$, which is a function of the air's density.
- In terms of magnitude, the hydraulic resistance of the heat exchanger $R_{HE}(\varrho_R)$ approximately ranges in the same size as the hydraulic resistance of the ram air channel. Thus $R_R(\varrho_R) \to 2 \cdot R_R(\varrho_R)$ and $R_{HE}(\varrho_R) \to 0$.

- Fluid capacitance will be neglected. This means, that the air's density is assumed to be constant while it remains inside the channel.
- The air mass flow $\dot{m}_C$ that is accounted for by the aircraft's velocity c and flight height h is modelled as an ideal flow source $\dot{m}_C$ parallel to the fan pressure sources and ram air channel's dynamic elements. This is done, because an actuator at the ram air channel intake manifold is controlled to keep the mass flow rate at a constant level during flight. However, this control loop is not modelled for the sake of simplicity.
- It is assumed, that the air mass inflow equals the outflow. Therefore the circuit can be closed.

2.3.2. Fan Dynamics

The dynamics of the fans work similar to the way the pumps have been modelled (refer to section 2.2.3). The major difference consists in the fact, that the ram air's density cannot be assumed constant but is a function of the inbound ram air pressure p_{R_0} and the ram air's entry temperature T_{R_0}. The pressure at the ram air channel's inlet p_{R_0} again is a function of flight speed and height, as is T_{R_0}.

For a given characteristic curve of the fans, the governing equations can be written as follows:

$$\begin{aligned}
\Delta p_{F_1}(n_{F_1}, \dot{m}_{F_1}) &= H^*_{F_1}\left(\frac{\dot{m}^*_{F_1}}{n^*_{F_1}}\right) \cdot \frac{H_{R_{F_1}} \cdot \varrho_R(h,c) \cdot g}{n^2_{R_{F_1}}} \cdot {n_{F_1}}^2 = 2R_R(\varrho_R) \cdot \dot{m}^2_{F_1} + L_R \cdot \ddot{m}_{F_1} \\
\Delta p_{F_2}(n_{F_2}, \dot{m}_{F_2}) &= H^*_{F_2}\left(\frac{\dot{m}^*_{F_2}}{n^*_{F_2}}\right) \cdot \frac{H_{R_{F_2}} \cdot \varrho_R(h,c) \cdot g}{n^2_{R_{F_2}}} \cdot {n_{F_2}}^2 = 2R_R(\varrho_R) \cdot \dot{m}^2_{F_2} + L_R \cdot \ddot{m}_{F_2}
\end{aligned} \tag{2.3.2}$$

with $n_{R_{F_i}}, H_{R_{F_i}}$ as the respective fan's rated speed and head,
p_{F_i}, H_{F_i} as the respective fan's output pressure and head,
$\dot{m}_{F_i}$ as the respective fan's entrance mass flow, and
* denoting the normalised values.

The ram air's density $\varrho_R(h, c)$ is a time-varying parameter, whose dependence on the altitude h and velocity c will be explained in the next section (2.3.3).

Note, that since the ram air channel is approximatively more square than circular, a hydraulic diameter d_h (refer to equation A.1.3 in appendix A.1) has to be calculated, in order to determine its hydraulic resistance coefficient ζ_R.

2.3.3. Air Inflow due to Aircraft Velocity

At the ram air intake the aircraft's velocity is used to create a certain pressure difference Δp_R, that drives a mass flow rate $\dot{m}_C$ through the ram air channel. Under the following assumptions thermodynamics' first fundamental theorem can be utilised to calculate the pressure difference (refer to [17, p. 61]).

- At the ram air intake manifold the air's velocity is reduced to zero.
- The change of state is completely adiabatic. The deceleration results in an increase of the air's enthalpy.
- A change of density is not neglected.
- Assume that the pressure at the ram air channel's outlet is the ambient pressure $p_A(h)$ at the respective flight height.

Under consideration of all aforementioned assumptions, the pressure at the ram air intake computes as follows:

$$p_R(h,c) = p_A(h) \cdot \left(1 + \frac{\kappa - 1}{2} \cdot Mach^2\right)^{\frac{\kappa}{\kappa-1}} \tag{2.3.3}$$

with $p_A(h)$ as the ambient pressure at height h,

$Mach = \dfrac{c}{a}$ as the aircraft's velocity in fractions of the sonic speed a,

$a = \sqrt{\kappa \cdot R_A \cdot T_A(h)}$ as the temperature dependent sonic speed,

$\kappa = 1.4$ as the isentropic expansion factor of air.

The same thermodynamic approach is valid to compute the temperature at the ram air intake for simulation purposes and yields:

$$T_{R_0}(h,c) = T_A(h) \cdot \left(1 + RF \cdot \frac{\kappa - 1}{2} \cdot Mach^2\right) \tag{2.3.4}$$

with $RF \approx 0.9$ as an empirical recovery factor

A recovery factor RF is introduced into this formula, for it to better approximate the measured temperature in practical applications. The ram air's density can be computed with the help of the *ideal gas law*:

$$\varrho_R(h,c) = \frac{1}{R_A} \cdot \frac{p_R}{T_{R_0}}(h,c) \tag{2.3.5}$$

The simulation of other important quantities like the ambient pressure $p_A(h)$, the ambient temperature $T_A(h)$ and the temperature dependent sonic speed $a(T_A)$ is based upon standardized formulas defined in [3]. A brief overview of the relevant aspects of ISO 2533 can be found in appendix B.

Returning to the dynamics of the ram air channel, the pressure difference that drives $\dot{m}_C$ through the channel is calculated from:

$$\Delta p_R = p_R - p_A(h) = p_A(h) \cdot \left(\left(1 + \frac{\kappa - 1}{2} \cdot Mach^2 \right)^{\frac{\kappa}{\kappa-1}} - 1 \right) \tag{2.3.6}$$

Thus, the governing equation is:

$$\Delta p_R = (2R_R(\varrho_R) \cdot R_{actuator}) \cdot \dot{m}_C^2 + L_R \cdot \ddot{m}_C \tag{2.3.7}$$

The actuator hydraulic resistance $R_{actuator}$ is always controlled during flight, such that $\dot{m}_C$ assumes a certain value $\dot{m}_C^0$, which — as has been already mentioned — will simply be simulated by applying a constant value to the mass flow rate. The controller aims for a minimised flow resistance of the aircraft and therefore has a higher priority than the controller considered in this thesis.

Accordingly, this differential equation will accordingly not be solved and all prior considerations may be deemed unnecessary — except, how to calculate for the ram air channel's height and velocity dependent temperature T_{R_0}. The respective equations have still been included into this thesis as to allow for a complete understanding of the basic dependencies of all environmental parameters and to facilitate future work, which includes the control of the fans.

2.4. The Thermodynamic Subsystem

Figure 2.4.1 shows the architecture's network from a thermodynamic point of view. The coloring scheme illustrates the relative differences in temperature. Relevant temperatures have been marked.

The important aspects of modelling the thermodynamic subsystem are threefold:

- Differential equations have to be derived, in order to simulate the *time dependent temperature curves* that are accounted for by *heat transfer*, both over and within the *system boundaries*.
- The transient behaviour of temperatures that result from a *mixture of fluid flows* needs a feasible differential formulation.
- The temperature propagation is prone to the transport delay of mass. While the mass flow rates change sufficiently fast, due to quasi instant exertion of pressure over the complete network, individual volume elements of cooling fluid take their time travelling through the pipes. As temperature is a local property, the transient behaviour suffers from dead-times determined by flow velocity and travelling distances, i.e. pipe length.

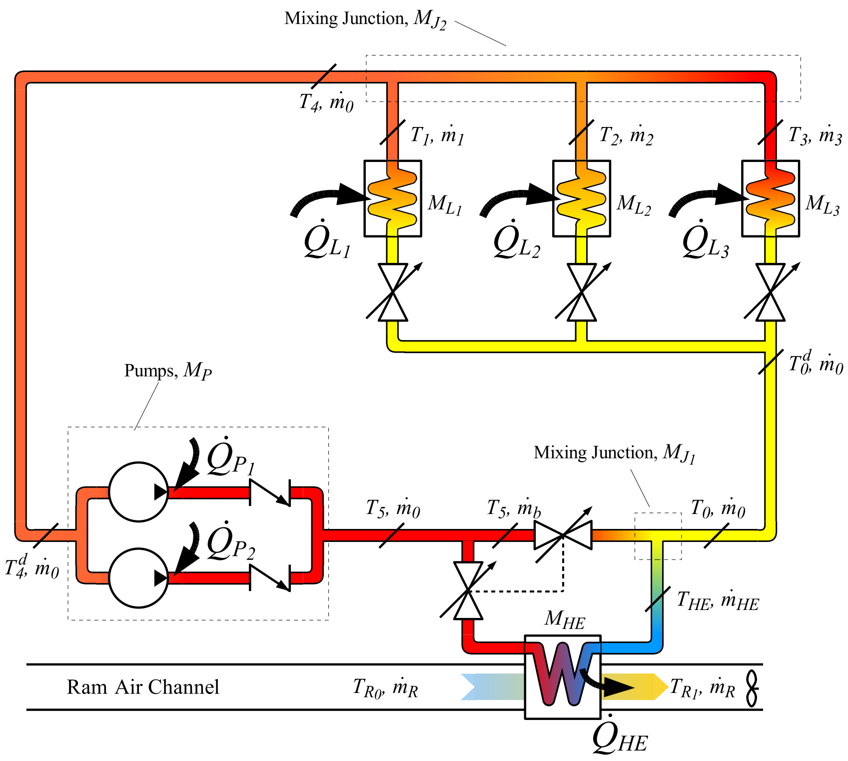


Figure 2.4.1.: Thermodynamic Subsystem

2.4.1. Initial Assumptions

Some simplifying assumptions have been made for the simulation of the thermodynamic subsystem.

- All pipes are considered adiabatic. Apart from the heat exchangers, loads and pumps, no heat is transferred from or to the cooling liquid inside the network cycle.
- The thermal capacitance of the pipes can be neglected.
- The physical and chemical properties of all fluids considered remain constant. More specifically, the heat capacitances under constant pressure c_p or volume c_v, as well as the densities ϱ do not vary.
- Heat conductance can be neglected compared to convective heat transfer, even in its absence.

2.4.2. Modelling of Thermodynamic Phenomena

Two important thermodynamic phenomena — heat transfer and fluid mixture — have to be modelled, each of which with some underlying assumptions. The heat transfer can be divided into the cases of a particular known time-varying *heat load* and heat transfer between two different fluid flows, namely between cooling cycle and ram air channel.

Heat Transfer from Known Heat Load

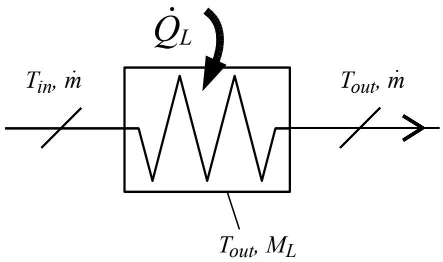


Figure 2.4.2.: Heat Transfer from Known Heat Load

Figure 2.4.2 shows a single heat load $\dot{Q}_L$ applied to a fluid flow $\dot{m}$. A differential equation can be derived under the following assumptions:

- The fluid mass inside the heat exchanger is $M_{fluid} = \text{const}$.
- The contribution of the heat exchanger material to the dynamics will therefore be limited to an effect on the time constant only. An artificial "fluid" mass M_L is introduced: $M_{HE}c_p + M_{fluid}c_v \approx M_L c_v$.
- The temperature inside the heat exchanger is a function of time only: $T_{HE} = T_{HE}(t) \neq T_{HE}(x,t)$. Therefore it remains constant over the exchanger's whole length.
- Accordingly, the temperature inside the heat exchanger is assumed to be the *outlet temperature*: $T_{HE}(t) = T_{out}(t)$.

Thus, the application of the *first fundamental theorem of thermodynamics* for *instationary flows* yields:

$$\underbrace{(M_{HE}c_p + M_{fluid}c_v)}_{M_L c_v} \cdot \dot{T}_{out} = \dot{m}c_p \cdot T_{in} - \dot{m}c_p \cdot T_{out} + \dot{Q}_L \qquad (2.4.1)$$

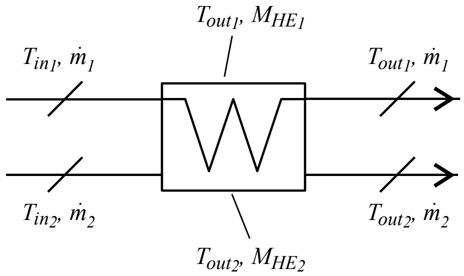


Figure 2.4.3.: Heat Transfer between two Fluid Flows

Convective Heat Transfer between two Fluid Flows

Figure 2.4.2 shows two separate fluid flows $\dot{m}_1, \dot{m}_2$, exchanging heat inside a heat exchanger. Under the assumption of the following properties of such a system, a convenient differential equation to simulate the approximate behaviour can be derived ([13] with added items):

- The fluid mass on each side of the heat exchanger is M_{HE_i} = const. Note, that the masses of both fluid flows are not the same even if the volumes are equal, since they may have different densities.
- The heat exchanger's thermal capacity will be neglected. Thus the material structure is not considered separately. Instead, the temperature of the aluminium walls will be assumed to be the temperature of the fluid flows on the respective sides. Therefore the wall thickness will be approximately 0.
- The heat exchanger can be regarded as two separate tubes, which can receive as well as emit convective flows of energy from/to the other.
- Fluid velocity and temperature will be assumed not to vary radially inside the pipes.
- The fluids' velocity averaged across the heat exchanger's total length is constant.
- There will be no significant heat transfer to the environment.

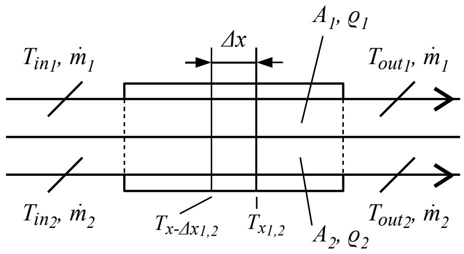


Figure 2.4.4.: Discretisation of Heat Exchanger

Figure 2.4.4 shows a discretisation scheme for a simplified heat exchanger structure. The application of the first fundamental theorem of thermodynamics for instationary flows yields:

$$(\Delta x \cdot A\varrho c_v) \cdot \dot{T} = \dot{m}c_p \cdot (T_{x-\Delta x} - T_x) - k \cdot A_{HE} \cdot \Delta T \tag{2.4.2}$$

with A as the cross section area and

T as the temperature of the heat exchanger tube under consideration,

ΔT as the local temperature difference between the tubes,

A_{HE} as the heat exchanger's energy permissive area,

k as the heat exchanger's heat transfer coefficient,

Δx as the heat exchanger's incremental distance.

Divided by Δx and considering the limit as $\Delta x \to 0$, equation 2.4.2 yields partial differential equations of the form:

$$(A_1\varrho c_{v_1}) \cdot \dot{T}_1 = -\dot{m}_1 c_{p_1} \cdot \frac{\partial T_1}{\partial x} - k \cdot C_{HE} \cdot (T_1 - T_2) \tag{2.4.3}$$

$$(A_2\varrho c_{v_2}) \cdot \dot{T}_2 = -\dot{m}_2 c_{p_2} \cdot \frac{\partial T_2}{\partial x} + k \cdot C_{HE} \cdot (T_1 - T_2) \tag{2.4.4}$$

with subscripts $_1$ and $_2$ denoting the respective fluid

and the symbol C_{HE} describing the heat exchanger's circumference.

Generally, the temperatures do not remain constant over the complete length of the heat exchanger. Therefore, a possible way to simulate the dynamic behaviour would be to discretize the heat exchanger into many small volume elements with thickness Δx to account for both time dependent and locational temperature gradients [13]. However, this would lead to a large number of states inside the state equation and since only an appropriate input-output behaviour is needed for the simulation, the temperature gradient with respect to x can be linearised. To do this, the heat exchanger is simply discretized by only one single element.

$$(M_{HE_1} \cdot c_{v_1}) \cdot \dot{T}_{out_1} = \dot{m}_1 c_{p_1} \cdot (T_{in_1} - T_{out_1}) - k \cdot A_{HE} \cdot (T_{out_1} - T_{out_2}) \tag{2.4.5}$$

$$(M_{HE_2} \cdot c_{v_2}) \cdot \dot{T}_{out_2} = \dot{m}_2 c_{p_2} \cdot (T_{in_2} - T_{out_2}) + k \cdot A_{HE} \cdot (T_{out_1} - T_{out_2}) \tag{2.4.6}$$

with $M_{HE_{1,2}}$ now denoting the respective fluid masses inside the heat exchangers.

As can be seen from equation 2.4.6 the heat exchanger is still assumed to be infinitely thin with respect to the transferred heat flow $\dot{Q}_{HE} = k \cdot A_{HE} \cdot (T_{out_1} - T_{out_2})$, but not with the heat exchanger characteristics (e.g. A_{HE},$M_{HE_{1,2}}$). Therefore the local temperatures $T_{1,2}$ become $T_{out_{1,2}}$ (Since the flow direction is constant and the model should define an input/output behaviour).

A different way to look at this stems from some standard formulas, that enable for the calculation of the transferred heat flow $\dot{Q}_{HE}$, which depends on the heat exchanger's characteristic *kA-value* and on the *average logarithmic temperature difference* $\Delta\vartheta_m$ (refer to [19, p. 208]). The overall heat transfer coefficient actually depends on certain variables that change with flow conditions. The relevant relations will be mentioned later.

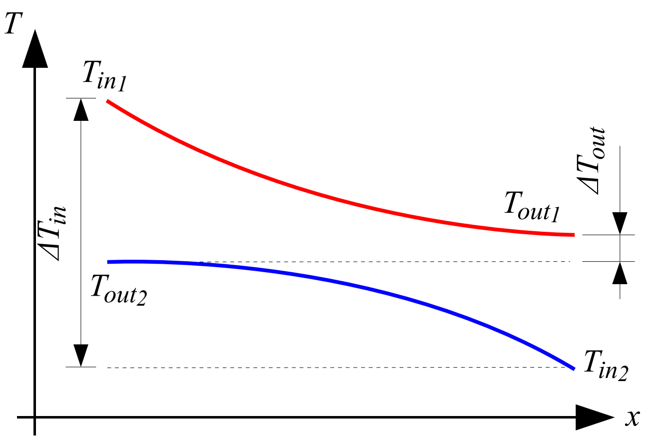


Figure 2.4.5.: Interior Temperature Curves Inside Heat Exchanger

Refer to figure 2.4.5 for temperature differences ΔT_{in}, ΔT_{out}.

$$\dot{Q}_{HE} = kA \cdot \Delta\vartheta_m \tag{2.4.7}$$

$$\Delta\vartheta_m = \frac{1}{A_{HE}} \cdot \int\limits_{A_{HE}} \Delta T \mathrm{d}A \tag{2.4.8}$$

$$\Delta\vartheta_m = \frac{\Delta T_{in} - \Delta T_{out}}{\ln(\Delta T_{in}/\Delta T_{out})} \qquad , \text{for } \Delta T_{in} - \Delta T_{out} \neq 0 \tag{2.4.9}$$

$$\Delta\vartheta_m = \frac{\Delta T_{in} + \Delta T_{out}}{2} \qquad , \text{for } \Delta T_{in} \approx \Delta T_{out} \tag{2.4.10}$$

From this, it can be seen, that during the previous simplification, $\Delta T_{in} = \Delta T_{out}$ and therefore $\dot{Q}_{HE} = kA \cdot \Delta T_{out}$ have been set, which leads to the following linearisation:

$$\Delta\vartheta_m = \Delta T_{out} \qquad , \text{for } \Delta T_{in} = \Delta T_{out} \tag{2.4.11}$$

Comparing all three variants 2.4.9, 2.4.10 and 2.4.11 to calculate the temperature difference that determines the magnitude of the transferred heat $\dot{Q}_{HE}$, reveals large discrepancies. Most importantly the approximated logarithmic temperature difference given in [19, p. 208] as quoted in equation 2.4.10 leads to a non-zero value as $\Delta T_{out} \to 0$ such that it is — apart from being exact only within a certain area around $\Delta T_{in} = \Delta T_{out}$ — therefore not applicable for a simulation that seeks for stable equilibria near $\Delta T_{out} = 0$. On the contrary, equation 2.4.11 provides an approximation, which would lead to stable equilibria, but drastically underestimates the transferred heat.

Since only the exact solution 2.4.9 of the integral 2.4.8 would lead to correct magnitudes of transferred heat $\dot{Q}_{HE}$, it should be considered the appropriate way to simulate. Tests showed, that this leads to numerical difficulties as $\Delta T_{out} \to \Delta T_{in}$ or $\Delta T_{out} \to 0$.

Figure 2.4.6 shows a plot of $\Delta\vartheta_m$ over ΔT_{out} for $\Delta T_{in} = 80\,\mathrm{K}$ for the different approximations thus far.

To cope with the numerical difficulties a linear approximation for $\Delta T_{out} < 10, \Delta T_{in} \approx 40...80\,\mathrm{K}$ has to be found, since the outlet temperature difference ΔT_{out} is expected to be small, while the inlet temperature difference ΔT_{in} is typically dependent on flight height h and aircraft velocity c. A Taylor series expansion around 0 for equation 2.4.9 is not possible due to the infinite slope as $\Delta T_{out} \to 0$. Because of this, the only possible solution is to act on the slope in a heuristic manner by introducing a compensation factor QC:

$$\Delta\vartheta_m = QC \cdot \Delta T_{out} \qquad , \text{for } \Delta T_{out} \approx 0 \tag{2.4.12}$$

This may seem like a very crude approximation, but as can be seen in validation plots in section 2.6, it yields rather good results.

Until now, the overall heat transfer coefficient kA has been assumed constant. Imagine one of the mass flows on either side of the heat exchanger comes to a halt, no heat is

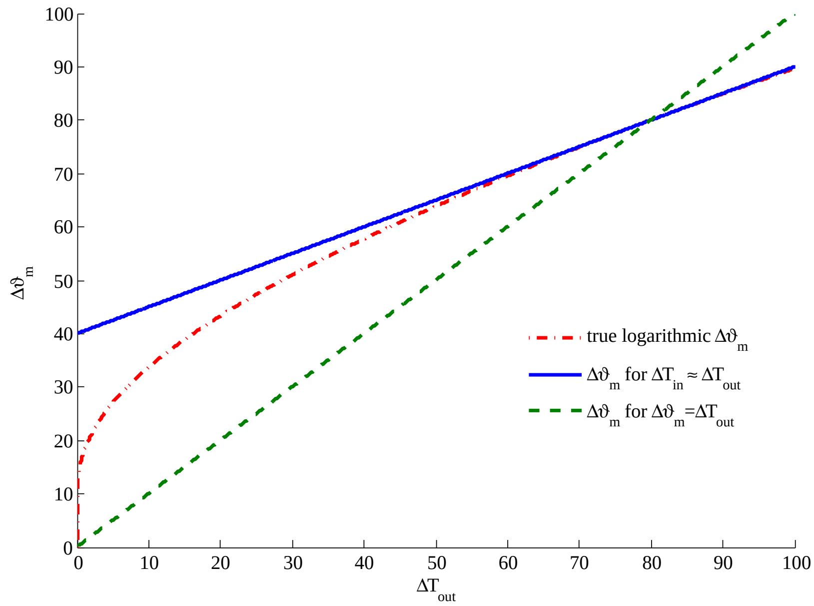


Figure 2.4.6.: Approximations of Average Logarithmic Temperature Difference $\Delta\vartheta_m$

then to be transferred by convection. With the current formulation this is not the case. Instead, the amount of transferred heat is largely independent of the mass flow rates, despite the fact, that they affect the fluids' heat transfer coefficient α. With a wall thickness near zero, kA is calculated from:

$$kA = \frac{1}{\frac{1}{\alpha_1 \cdot A_1} + \frac{1}{\alpha_1 \cdot A_1}} \tag{2.4.13}$$

with α_i, $\quad i = 1, 2$ as the respective fluid's heat transfer coefficient,

A_i, $\quad i = 1, 2$ as the heat transfer area on either side.

The heat transfer coefficient can be computed with the help of the characteristic NUSSELT number Nu:

$$\alpha = \frac{\lambda}{d_h} \cdot \mathrm{Nu} \tag{2.4.14}$$

with λ as the heat conductivity,

d_h as the hydraulic diameter of the pipe,

Nu as the hydraulic diameter of the pipe,

The NUSSELT number describes, how the heat transferred by convection is increased over the heat transferred by conductance. The DITTUS-BOELTER correlation for forced convection describes an approximation for the NUSSELT number, that yields a feasible possibility to simulate the dependence of the overall heat transfer coefficient on the respective mass flow rates: $kA(\dot{m}_1, \dot{m}_2)$ (slightly modified from [9, p.538]). The following criteria define the correlations applicability:

- The flows considered are turbulent with Re > 10000.
- No boiling, condensation or significant radiation takes place, but only forced convection.
- The flows' PRANDTL numbers range from 0.7...160
- The distance from the pipe entrance is at least 10 times the hydraulic diameter.
- The flow can be considered hydraulically smooth.

Strictly, the first assumption is violated under the conditions given in the cooling network, but as has been mentioned earlier, pipe bends and other pipe geometries, not to forget the heat exchangers themselves, sufficiently make for turbulent flow conditions.

The DITTUS-BOELTER correlation looks as follows:

$$\mathrm{Nu} = 0.023\mathrm{Re}^{0.8} \cdot \mathrm{Pr}^{0.3...0.4} \tag{2.4.15}$$

with Pr as the PRANDTL number,

and the exponent 0.3...0.4 depending on heating or cooling.

For known, approximatively constant PRANDTL numbers $\mathrm{Pr}_{1,2}$ and by reformulating the REYNOLDS number as a function of the mass flow rates: $\mathrm{Re}(\dot{m}_{1,2})$, the overall heat transfer coefficient can be expressed as:

$$kA(\dot{m}_1, \dot{m}_2) = \frac{1}{\frac{1}{H_1 \cdot \dot{m}_1^{0.8}} + \frac{1}{H_2 \cdot \dot{m}_2^{0.8}}} \tag{2.4.16}$$

$$\text{with } H_i = \frac{\lambda_i}{d_{h,i}} \left(0.023\mathrm{Pr}_i^{0.3...0.4} \cdot \left(\frac{d_{h,i}}{\varrho_i \cdot A_i \cdot \nu_i} \right) \right)^{0.8} \cdot A_i \quad i = 1, 2$$

while ν_i is the fluid's kinematic viscosity.

Fluid Mixture in Network Junctions

Figure 2.4.7 shows an arbitrary number of separate fluid flows $\dot{m}_i, i = 1, ..., n$ mixing at a *network junction* into the outlet flow $\dot{m}_0$. Under the following assumptions, governing equations for the transient behaviour can be derived:

- The fluid mass inside the junction is $M_J = \text{const.}$
- The temperature inside the junction is only a function of time: $T_J = T_J(t) \neq T_J(x, t)$. Therefore it remains constant over the junctions' whole geometry.
- Accordingly, the temperature inside the junction is assumed to be the outlet temperature: $T_J(t) = T_{out}(t)$

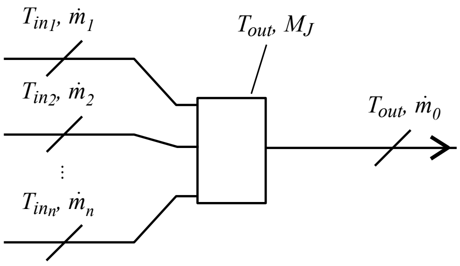


Figure 2.4.7.: Fluid Mixture in Network Junctions

- All fluid mass flows are presumed to be of the same type of liquid or gas.

Therefore the instationary balance over the junction gives:

$$M_J c_v \cdot \dot{T}_{out} = \sum_{i=1}^{n} \dot{m}_i c_p \cdot T_{in_i} - \dot{m}_0 c_p \cdot T_{out} \tag{2.4.17}$$

Remark

The masses M introduced to describe the amount of fluid inside the respective network's element determine the time constant of the temperatures' transient behaviour. Hence, these values can be deliberately adjusted to achieve the desired characteristics, despite its physical meaning. Moreover it may sometimes be difficult to find the respective value by means of physical considerations, even more so, if the system boundaries of a junction has been expanded to include a 3-way junction with some neglected pipe elements in between. Therefore a system's model has to undergo at least some tuning with respect to *parameter identification*. Values of $M \rightarrow 0$ would lead to instant mixture or heat transfer, which is theoretically correct when considering infinitesimal volume elements. However, initially choosing a time constant close to one leads to transients in the time range of the mass flow rates and is a feasible starting point.

2.4.3. The Governing Equations

The thermodynamic subsystem has to be decomposed into two main blocks, stemming from the fact, that they are connected by pipes, which induce a significant dead-time. All other dead-times which result from pipes of inferior length are neglected.

In the following, the superscripts ^{1} and ^{2} denote that a certain variable belongs to one of the respective main blocks. The superscript ^{d} identifies a temperature to be *delayed* by the respective dead-time t_d^1 or t_d^2.

The following set of equations govern the dynamics of block 1:

$$\begin{aligned}
\dot{T}_0^d &= -\frac{1}{\tau_d} \cdot T_0^d + \frac{1}{\tau_d} \cdot T_{0,2}^d \\
M_{L_1} c_{v_P} \cdot \dot{T}_1 &= \dot{m}_1 c_{p_P} \cdot T_0^d - \dot{m}_1 c_{p_P} \cdot T_1 + \dot{Q}_{L_1} \\
M_{L_2} c_{v_P} \cdot \dot{T}_2 &= \dot{m}_2 c_{p_P} \cdot T_0^d - \dot{m}_2 c_{p_P} \cdot T_2 + \dot{Q}_{L_2} \\
M_{L_3} c_{v_P} \cdot \dot{T}_3 &= \dot{m}_3 c_{p_P} \cdot T_0^d - \dot{m}_3 c_{p_P} \cdot T_3 + \dot{Q}_{L_3} \\
M_{J_2} c_{v_P} \cdot \dot{T}_4 &= \dot{m}_1 c_{p_P} \cdot T_1 + \dot{m}_2 c_{p_P} \cdot T_2 + \dot{m}_3 c_{p_P} \cdot T_3 - \dot{m}_0 c_{p_P} \cdot T_4
\end{aligned} \tag{2.4.18}$$

Block 2 is governed by:

$$\begin{aligned}
\dot{T}_4^d &= -\frac{1}{\tau_d} \cdot T_4^d + \frac{1}{\tau_d} \cdot T_{4,1}^d \\
M_P c_{v_P} \cdot \dot{T}_5 &= \dot{m}_0 c_{p_P} \cdot T_4^d - \dot{m}_0 c_{p_P} \cdot T_5 + \dot{Q}_{P_1} + \dot{Q}_{P_2} \\
M_{HE} c_{v_P} \cdot \dot{T}_{HE} &= \dot{m}_{HE} c_{p_P} \cdot T_5 - \dot{m}_{HE} c_{p_P} \cdot T_{HE} - kA(\dot{m}_{HE}, \dot{m}_R) \cdot QC \cdot (T_{HE} - T_{R_1}) \\
\dot{T}_{R_0} &= -\frac{1}{\tau_T} \cdot T_{R_0} + \frac{1}{\tau_T} \cdot T_{R_0,meas.} \\
M_{HE_R} c_{v_A} \cdot \dot{T}_{R_1} &= \dot{m}_R c_{p_A} \cdot T_{R_0} - \dot{m}_R c_{p_A} \cdot T_{R_1} + kA(\dot{m}_{HE}, \dot{m}_R) \cdot QC_R \cdot (T_{HE} - T_{R_1}) \\
M_{J_1} c_{v_P} \cdot \dot{T}_0 &= \dot{m}_b c_{p_P} \cdot T_5 + \dot{m}_{HE} c_{p_P} \cdot T_{HE} - \dot{m}_0 c_{p_P} \cdot T_0
\end{aligned} \tag{2.4.19}$$

It is convenient to introduce a prefilter to the ram air's temperature T_{R_0} and the delayed temperatures T_0^d and T_4^d into the state space formulation. This enables to treat them as state variables. T_{R_0} will assume the value of the measured input $T_{R_0,meas.}$ with time constant $\tau_T \to 0$, while the delayed temperatures have time constants $\tau_d \neq 0$, which can be exploited to simulate a diffusion process along the pipe.

Each of the two blocks can be written in the form of a *linear parameter-varying system* state space model, whose system matrix is varying with the parameter vector $\boldsymbol{\theta_m} = \begin{pmatrix} \dot{m}_0 & \dot{m}_1 & \dot{m}_2 & \dot{m}_{HE} & \dot{m}_R & kA(\dot{m}_0, \dot{m}_R) \end{pmatrix}^T$ containing the mass flow rates and the mass flow rate dependent overall heat transfer coefficient. Note, that it has been made use of the fact that $\dot{m}_3 = \dot{m}_0 - \dot{m}_1 - \dot{m}_2$ and $\dot{m}_b = \dot{m}_0 - \dot{m}_{HE}$ to reduce the number of parameters, the system matrices depend on. This gives:

$$\begin{aligned}
\boldsymbol{M_{T^1}} \cdot \dot{\boldsymbol{x}}_{\boldsymbol{T^1}} &= \tilde{\boldsymbol{A}}_{\boldsymbol{T^1}}(\boldsymbol{\theta_m}) \cdot \boldsymbol{x_{T^1}} + \tilde{\boldsymbol{D}}_{\boldsymbol{T^1}} \cdot \boldsymbol{v_{T^1}} \\
\dot{\boldsymbol{x}}_{\boldsymbol{T^1}} &= \underbrace{\boldsymbol{M_{T^1}}^{-1} \tilde{\boldsymbol{A}}_{\boldsymbol{T^1}}(\boldsymbol{\theta_m})}_{\boldsymbol{A_{T^1}}(\boldsymbol{\theta_m})} \cdot \boldsymbol{x_{T^1}} + \underbrace{\boldsymbol{M_{T^1}}^{-1} \tilde{\boldsymbol{D}}_{\boldsymbol{T^1}}}_{\boldsymbol{D_{T^1}}} \cdot \boldsymbol{v_{T^1}}.
\end{aligned} \tag{2.4.20}$$

And similarly:

$$\dot{\boldsymbol{x}}_{\boldsymbol{T^2}} = \underbrace{\boldsymbol{M_{T^2}}^{-1} \tilde{\boldsymbol{A}}_{\boldsymbol{T^2}}(\boldsymbol{\theta_m})}_{\boldsymbol{A_{T^2}}(\boldsymbol{\theta_m})} \cdot \boldsymbol{x_{T^2}} + \underbrace{\boldsymbol{M_{T^2}}^{-1} \tilde{\boldsymbol{D}}_{\boldsymbol{T^2}}}_{\boldsymbol{D_{T^2}}} \cdot \boldsymbol{v_{T^2}}. \tag{2.4.21}$$

The system matrices $\boldsymbol{A_{T^{1,2}}}$ can be disjoint into affine combinations of submatrices:

$$\begin{aligned}\boldsymbol{A_{T^1}}(\boldsymbol{\theta_m}) &= \boldsymbol{A_{T^1}^{const}} + \boldsymbol{A_{T^1}^{0}} \cdot \dot{m}_0 + \boldsymbol{A_{T^1}^{1}} \cdot \dot{m}_1 + \boldsymbol{A_{T^1}^{2}} \cdot \dot{m}_2 \\ \boldsymbol{A_{T^2}}(\boldsymbol{\theta_m}) &= \boldsymbol{A_{T^2}^{const}} + \boldsymbol{A_{T^2}^{0}} \cdot \dot{m}_0 + \boldsymbol{A_{T^2}^{HE}} \cdot \dot{m}_{HE} + \boldsymbol{A_{T^2}^{R}} \cdot \dot{m}_R + \boldsymbol{A_{T^2}^{kA}} \cdot kA(\dot{m}_0, \dot{m}_R)\end{aligned}$$

Figures 2.4.8 and 2.4.9 show block diagrams of the linear parameter-varying state space models of the respective blocks.

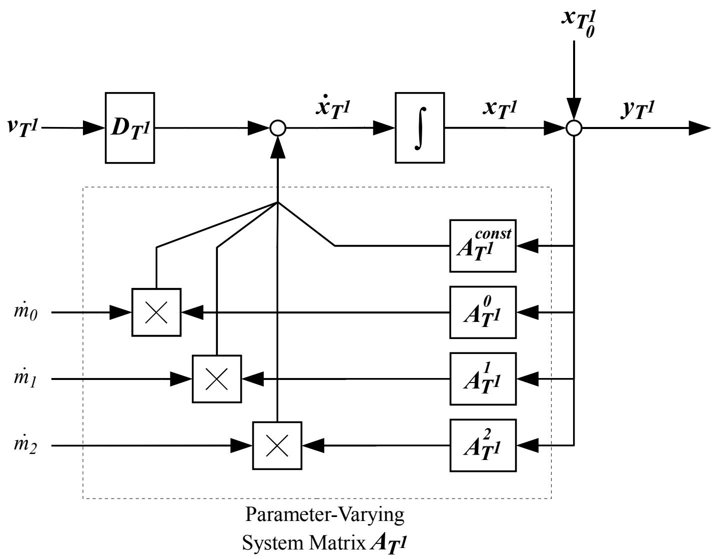


Figure 2.4.8.: Block Diagram of Thermodynamic Subsystem Block 1

Note, that there is no direct controllable input $\boldsymbol{u_T}$ to the system. It can only be stabilised indirectly by controlling for certain mass flow rates. The inputs $\boldsymbol{v_{T^{1,2}}}$ are vectors of certain variables determined by the hydraulic subsystem or given quantities, like the loads applied $\dot{Q}_{L_i}, i = 1,2,3$, the pumps' electric power $P_{el}^{P_j}, j = 1,2$ and the ram air channel's temperature $T_{R0,meas.}$.

$$\begin{aligned}\boldsymbol{v_{T^1}} &= \begin{pmatrix} T_{0,in}^d & \dot{Q}_{L_1} & \dot{Q}_{L_2} & \dot{Q}_{L_3} \end{pmatrix}^T \\ \boldsymbol{v_{T^2}} &= \begin{pmatrix} T_{4,in}^d & P_{el}^{P_1} & P_{el}^{P_2} & T_{R0,meas.} \end{pmatrix}^T\end{aligned}$$

After the temperatures T_0 and T_4 are taken from the respective block's output, time delays $t_d^{1,2}$ are applied to them before feedback. The delays depend on the fluid's flow velocity and therefore on the total mass flow rate $\dot{m}_0$.

$$t_d^{1,2} = l_p^{1,2} / \left(\frac{\dot{m}_0}{\varrho_P \cdot A_p^{1,2}} \right) \tag{2.4.22}$$

with $l_p^{1,2}$ and $A_p^{1,2}$ as the respective pipe's length and cross section area.

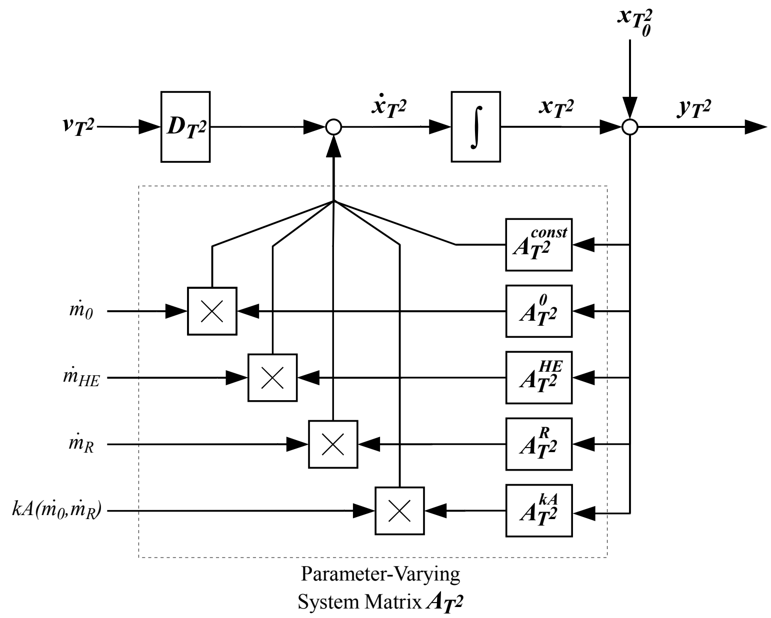


Figure 2.4.9.: Block Diagram of Thermodynamic Subsystem Block 2

To see how both blocks are connected to form one thermodynamic subsystem refer to figure 2.4.10.

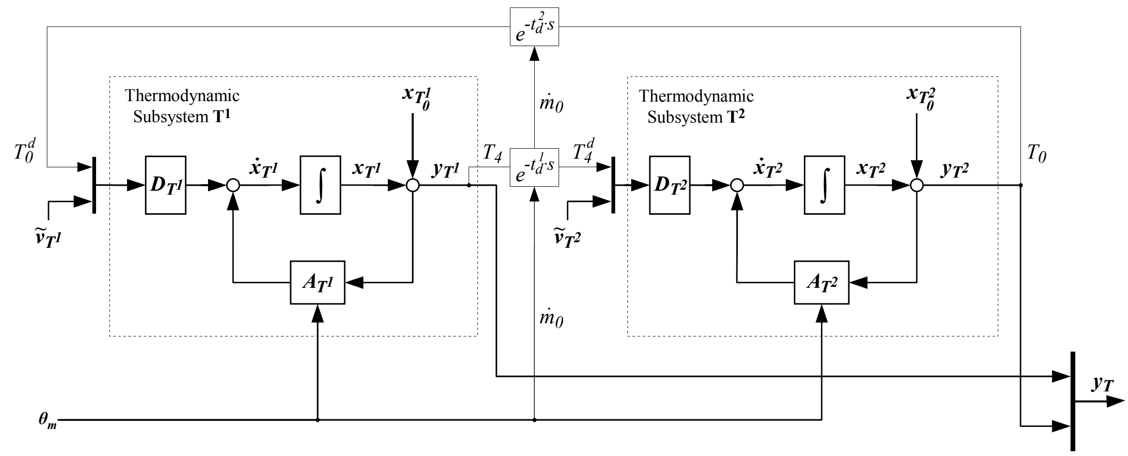


Figure 2.4.10.: Thermodynamic Subsystem Blocks with Time Delay

The explicit system matrices can be found in appendix D.1.3.

2.5. Plant Parameters, Inputs and Outputs

The system's parameters can be subdivided into two classes: Environmental parameters and system inherent parameters.

The system inherent parameters are variables, that can be computed from the system's state variables and also affect the dynamic behaviour of the system in terms of eigenvalues. For instance, the cooling fluid's mass flow $\dot{m}_0$ and its fractions $\dot{m}_i, i = 1, 2, 3$ determine the thermodynamic subsystem's eigenvalues. Some of these parameters become important when considering the control of the system. It will be a matter to investigate, which quantities have to become controlled variables. As an example, since there is actually no direct input to the thermodynamic subsystem, the only way to stabilise it, is to control for certain mass flow rates and mass flow rate fractions.

The environmental parameters are defined as all time-varying variables that have an influence to the cooling system from outside the system's boundary. They are needed to describe the current flight conditions that account for changes in the system's dynamic behaviour. Examples for environmental parameters are the aircraft's altitude h or velocity c. These parameters cannot be controlled by means of external inputs, but either change the system's dynamic properties or can be interpreted in terms of external disturbances to the system. The surrounding air's temperature $T_A(h)$, for instance, influences the amount of heat transferred to the cooling cycle, while the loads applied to the system take the form of time-varying disturbances. In steady state, these values affect the equilibrium, such that different mass flow rates are needed, to keep certain desired temperature levels.

The following section will provide an overview of quantities classified as environmental parameters, which until now have not yet been covered in terms of how they have been implemented into the simulation model. Furthermore, the plant's system inputs and outputs will be summarised.

2.5.1. Environmental Parameters

The current phase of the aircraft's flight plan yields the main set of environmental parameters that accounts for the most important changes in the cooling system's dynamic behaviour. During flight, the loads $\dot{Q}_{L_i}, i = 1, 2, 3$ applied to the cooling cycle will vary, as they may represent vital subsystems as well as minor peripherals. For example, there may be need for the power electronics to be online only during some particular period in flight, which will — in general — be well known beforehand. In certain failure scenarios, however, one component can fail, such that the load has to be redistributed. These cases cannot be foreseen. This is why — considering safety issues — a controller should also be able to handle unforeseen loads. Thus, from a control systems perspective the loads can be regarded as disturbances to the system, which the controller has to reject in appropriate ways.

The same goes for the aircraft's altitude h and flight velocity c. These parameters describe the airplane's flight envelope or determine whether it is still on the airfield. As seen in section 2.3.3 these quantities have a significant impact on the cooling abilities of the primary heat exchanger. As a consequence, the surrounding air's temperature $T_A(h)$ and density $\varrho_A(h)$ affect the cooling heat transfer. While these quantities are easy to measure on a real airplane, the simulation needs realistic assumptions. As has already been mentioned in section 2.3.3 [3] provides a valid model for atmospheric values, that is widely used in aerospace applications (please refer to appendix B for a brief overview of the relevant equations).

2.5.2. Plant Inputs and Outputs

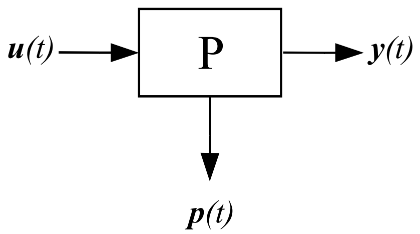


Figure 2.5.1.: Plant Inputs and Outputs

Since there exist three different subsystems, the sum of all controllable inputs over the subsystems give the plant inputs. Together these are:

$$\boldsymbol{u_P} = \left.\begin{pmatrix} n_{P_1} \\ n_{P_2} \end{pmatrix}\right\} \text{Pumps' 1 and 2 reference rotational speeds}$$

$$\boldsymbol{u_F} = \left.\begin{pmatrix} n_{F_1} \\ n_{F_2} \end{pmatrix}\right\} \text{Fans' 1 and 2 reference rotational speeds}$$

$$\boldsymbol{u_v} = \left.\begin{pmatrix} v_{1_{ref}} \\ v_{2_{ref}} \\ v_{3_{ref}} \\ v_{HE_{ref}} \end{pmatrix}\right\} \text{Valves' reference opening ratios}$$

Thus the input vector $\boldsymbol{u}$ is:

$$\boldsymbol{u} = \begin{pmatrix} n_{P_1} & n_{P_2} & n_{F_1} & n_{F_2} & v_{1_{ref}} & v_{2_{ref}} & v_{3_{ref}} & v_{HE_{ref}} \end{pmatrix}^T \tag{2.5.1}$$

The main output of the system is a concatenation of the output temperatures of the two thermodynamic subsystem blocks. Together they form the main output vector $\boldsymbol{y}$:

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y_{T^1}} \\ \boldsymbol{y_{T^2}} \end{pmatrix} \tag{2.5.2}$$

$$\boldsymbol{y_{T^1}} = \begin{pmatrix} T_0^d & T_1 & T_2 & T_3 & T_4 \end{pmatrix}^T \tag{2.5.3}$$

$$\boldsymbol{y_{T^2}} = \begin{pmatrix} T_4^d & T_5 & T_{HE} & T_{R_0} & T_{R_1} & T_0 \end{pmatrix}^T \tag{2.5.4}$$

Any further variables may be of use, for instance, for controller scheduling purposes and may be included into a separate, time-varying plant output parameter vector $\boldsymbol{p(t)}$ if needed.

This set of inputs and outputs provide a most general framework for further considerations concerning controller strategies. The different types of controllers discussed later in this thesis will only use part of the interface and it should be noted, that the definition of a convenient set of input and output variables varies with different types of controllers or different controller structures. Furthermore, so far no restrictions have been imposed on the availability of certain outputs or inputs. This will be covered in the following section.

2.5.3. Measurement of Plant Parameters

An adaptive controller relies on some set of plant parameters by which its respective gains are scheduled. This kind of *gain scheduling* of a controller may be required, in order to cover not only one but multiple operating conditions. However, the feasibility of gain scheduling is subject to certain possible constraints. Apart from hardware limitations, which will not be considered in this thesis, the measureability of the plant parameters has to be considered. To do this in a systematic way, all time-varying plant parameters have to be assembled in a list and their respective issues concerning measureability and reliability have to be assessed. This is done in the following overview.

Environmental Parameters

- **Ram Air Temperature T_{R_0}:** The temperature inside the ram air channel is — as are all temperature values — easy to measure. Appropriate devices neither are large in size, nor especially expensive. There exist various measurement methods with high reliability. Considering the small size, even if there were reliability problems of single instruments, redundant deployment would be possible to achieve a high degree of feasibility. The temperature of the cooling ram air is one of the major sources of disturbance to the system and would allow to schedule controllers for different operating points.

- **Loads Applied to the Cooling Cycle Q_{L_i}:** The loads applied to the cooling cycle are the other major sources of disturbance. They could be measured indirectly by the power consumption of the respective electronics. From there the amount of convective heat transfer may be inferred. Even better, their magnitudes typically depend on the flight schedule and are thus know in advance, which further lowers the demands on reliability since no measurement instrument is needed.
- **Aircraft Velocity c and Ram Air Mass Flow Rate $\dot{m}_R$:** The aircraft's velocity is a quantity that is constantly measured during flight. From it, a value for the ram air mass flow rate may be estimated. But since a high priority controller controls for a constant and assuredly known ram air mass flow rate during flight, the significance of both parameters for scheduling purposes is low.
- **Pressure Difference across Ram Air Channel Δp_R and Ram Air Density ϱ_R:** Again, these parameters are insignificant for controller scheduling.

System Inherent Parameters

- **Temperatures T_i:** As mentioned before, temperatures are easy to measure. In fact, without measuring at least an essential set of temperatures no controller could be synthesised.
- **Mass Flow Rates $\dot{m}_i$:** The measurement of mass flow rates based on pressure differences usually involves the introduction of a significant pressure drop in a pipe, leading to energy dissipation. Different constructions, like ultra sonic flowmeters, on the other hand may lead to reliability issues. Engineers at the institute of thermofluiddynamics affirm, that the measurement of mass flow rates is not common practice in aircraft engineering. Therefore it is considered infeasible.
- **Normalised Time Constant of the Hydraulic Subsystem L'_p/R'_p:** The normalised time constant of the hydraulic subsystem is a complex parameter expressed by a non-linear equation depending on the valve opening ratios v_i 2.2.4. Within uncertainty bounds this value might be calculated from the plant inputs.

Conclusion

Reliability is a common issue to be concerned with in aircraft engineering. With respect to this, a controller with as few inputs and scheduling parameters is desirable. However, the above list shows that there are some promising opportunities to exploit plant parameters for controller scheduling in order to raise performance.

2.6. Validation

This section aims at validating the results obtained from the non-linear simulation of the software tools *MATLAB/Simulink*. Since the plant examined in this thesis does not yet exist, the transients will only be verified against data retrieved from an already existing *simulation model* built with the software *Flowmaster*. But as *Flowmaster* is widely considered a standard industrial engineering tool, a positive comparison will be deemed sufficient to prove the model's validity.

For this purpose, selected simulation test setups will be presented here, as to discuss the most important aspects of the simulation. It should be noted, that building the non-linear simulation in *MATLAB/Simulink* has been an iterative process. Many more simulation data have been recorded and evaluated in the process of debugging and determining the relevant characteristic physical effects inherent to the plant. In this way some features have been included retroactively, after a certain crucial abnormality had been detected. This, however, does not become evident during the study of the previous chapters, since those aim at providing a brief but thorough description of the needed equations.

Please also notice, that in order to avoid singularities or numerical hazards, few adjustments have been made to the simulation, that have not been explicitly mentioned. Those may include saturation blocks, that prevent particular values from being of zero magnitude in denominators, or small dead zones applied to time-varying parameters, that compensate for numerical inaccuracies. An exhaustive description of these adjustments will be omitted here. It is suggested to closely examine the *Simulink* model, which should not be cumbersome, because the adjustments are few.

2.6.1. Comparison of Selected *Flowmaster* and *MATLAB/Simulink* Simulation Test Runs

Due to the lack of data and because of the fact, that the ram air channel subsystem has originally not been included in the *Flowmaster* model, the validation of this part of the system will be postponed to future work on the cooling plant. It will be justified later, that the analysis with regard to controller design can be done under the neglection of that part of the system (please refer to section 3.1.2).

The main focus of the validation therefore lies on the hydraulic and thermodynamic subsystems.

The Hydraulic Subsystem

Figures 2.6.1 and 2.6.2 show the respective system's response to the *test run specifications* listed in table 2.6.1 and 2.6.1.

Time t in [s]	v_1	v_2	v_3	v_b	v_{HE}
0	1	0	0	1	0
2	1	1	0	1	0
4	1	1	1	1	0
6	0	1	1	1	0
8	0	0	1	1	0
10	0	0	1	0	1
12	0	0	1	0.5	0.5
14	0.66	0.33	0	1	0
16	0.5	0.5	0	1	0
18	0.33	0.66	0	1	0

Table 2.6.1.: Valve Opening Ratios during Hydraulic Subsystem Test

Time t in [s]	n_{P_1} **in** $[\min^{-1}]$	n_{P_2} **in** $[\min^{-1}]$
0-20	1000	0

Table 2.6.2.: System Inputs and Parameters during Hydraulic Subsystem Test

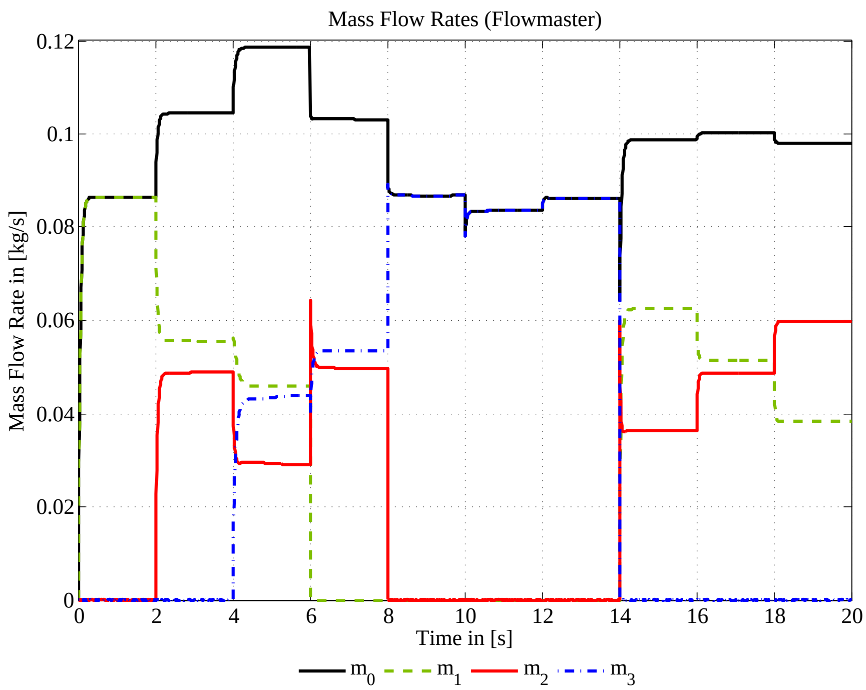


Figure 2.6.1.: *Flowmaster* Generated Test of the Hydraulic Subsystem

Figures 2.6.1 and 2.6.2 show good overall correspondence both in terms of absolute magnitudes and relative variations of the mass flow rates. The step sizes arising in the fractioned mass flow rates $\dot{m}_i, i = 1, 2, 3$ are somewhat higher in the *MATLAB*

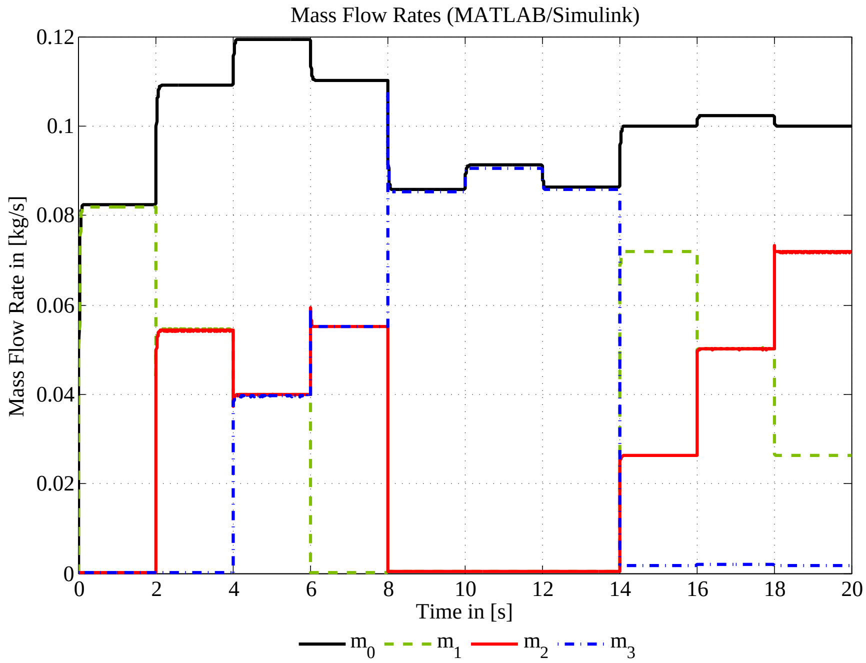


Figure 2.6.2.: *MATLAB/Simulink* Generated Test of the Hydraulic Subsystem

simulation, which is probably an effect of the calculation of hydraulic resistances from valve opening ratios. The software *Flowmaster* incorporates additional factors depending on the REYNOLDS number and may thus provide more accurate results. The tendency, however, is very well within acceptable bounds and does not influence the controller design process.

The only two spots, where the tendency is directly opposite is at time $t = 4\,\text{s}$ and at time $t = 10\,\text{s}$. At time $t = 4\,\text{s}$ a decrease of mass flow rate $\dot{m}_2$ occurs in the *Flowmaster* simulation, resulting in the mass flow fractions $\dot{m}_i, i = 1, 2, 3$ not being level, as is the case with the *MATLAB* simulation. This can be explained by the simplifications employed on the hydraulic H-shaped network (refer to section 2.2.2). Additional efforts to attempt a solution of the non-linear system of equations, that the H-shaped circuitry results in, could be made to alleviate this issue. This has been, however, decidedly left to possible future work on this topic.

At time $t = 10\,\text{s}$ an increase in total mass flow rate $\dot{m}_0$ can be detected with the *MATLAB* simulation. According to *Flowmaster* it should actually decrease. An explanation for this can be found by observing the hydraulic subsystem model at the heat exchanger bypass. When the valve v_b is opened the hydraulic resistance of the heat exchanger is bridged. Therefore the overall network's hydraulic resistance should decrease, as in the *MATLAB* simulation. There, the bypass pipe's hydraulic resistance has been neglected, but not in the *Flowmaster* model. The effect is minor, but this should be kept in mind, when a final architecture of the cooling cycle is considered. It may well be, that the bypass pipe has significant influence by then, because of constructional demands.

The Thermodynamic Subsystem

Two separate tests will be presented on the functionality of the thermodynamic subsystem simulation. Figures 2.6.3 and 2.6.4 show plots of the results.

Test run 1 and 2 are defined by flight altitude $h = 10,000\,\mathrm{m}$ at velocity $c = 300\,\mathrm{m/\,s}$. At time $t = 100\,\mathrm{s}$ or $t = 200\,\mathrm{s}$, respectively, the heat exchanger valve v_{HE} is fully opened. All other valves remain opened throughout the whole simulation time. The only difference between test run 1 and 2 is, that loads are applied to the cooling cycle during the second test run with magnitudes as listed in table 2.6.1.

Q_{L_1} **in** [kW]	Q_{L_2} **in** [kW]	Q_{L_3} **in** [kW]
1	1	2

Table 2.6.3.: Loads Applied during Thermodynamic Subsystem Test 2

The *MATLAB/Simulink* curves compare very well to those generated with *Flowmaster* with regard to temperature slopes and time delay. Minor deviations are subject to the quality of parameter estimation. The artificial liquid masses M_{L_i}, M_{J_i} introduced in the differential equations can be tuned, in order to achieve an even closer resemblance. On the other hand, *Flowmaster* incorporates time-varying heat capacities $c_{v_P}(\cdot)$ and $c_{p_P}(\cdot)$, which leads to abiding deviations, that are small, though.

The most significant difference can be noticed with the heat exchange. *Flowmaster* handles this in a very abrupt manner, which suggests, that the *Simulink* simulation may be even more simplified. With the latter, one can detect a stair-shaped curve until the temperatures finally reach their minimum. This can clearly be put to the responsibility of the way the heat exchange is realised and the way time delay is introduced. A momentarily stable intermediate state is the effect of the interplay. Under the assumption, that this will only introduce an additional disturbance to the system, the controller has to cope with, the *Simulink* simulation is deemed acceptable.

2.6.2. Summary

The comparison showed, that the *Simulink* simulation derived in the previous sections yields results close to industry standard simulation tools like *Flowmaster*. Minor glitches have been detected and explanations found.

The usefulness of *engineering software tools* with respect to rapid controller design varies strongly. While *MATLAB* can be considered industry– as well as scientific standard and provides a host of tools for control engineering, *Flowmaster* (of version as is available at the institute of thermofluiddynamics) merely incorporates a scripting language. This is

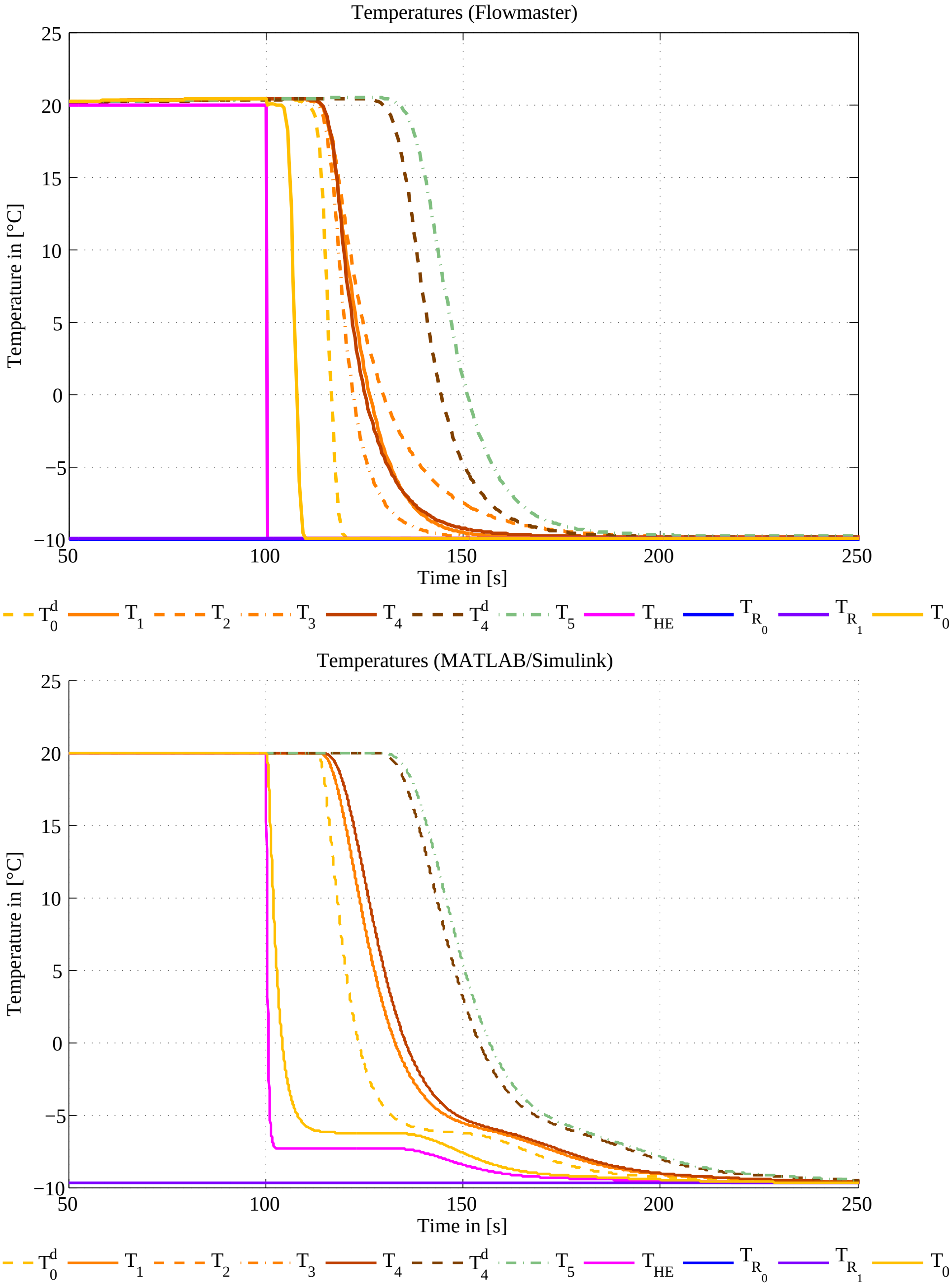


Figure 2.6.3.: *Flowmaster* (t.) and *MATLAB* (b.) Generated Plots of Test Run 1 of the Thermodynamic Subsystem

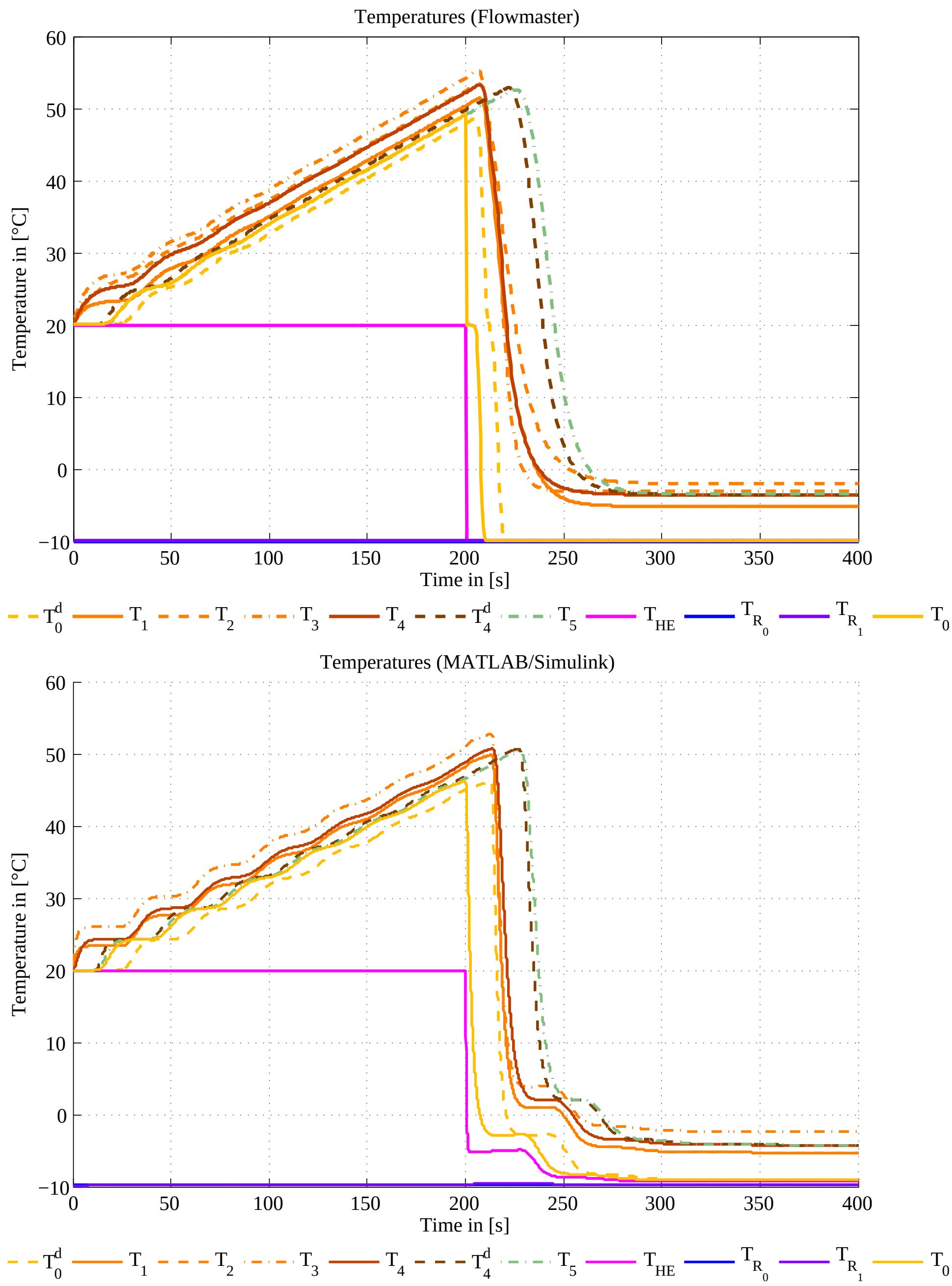


Figure 2.6.4.: *Flowmaster* (t.) and *MATLAB* (b.) Generated Plots of Test Run 2 of the Thermodynamic Subsystem

powerful as well and enables for the programming of several types of controllers. Prefabricated controller synthesis functions including tremendous mathematical computations, however, greatly facilitate controller design and are not present. Therefore, being able to simulate the non-linear plant, to compute and to implement various controllers in one single environment is a desirable achievement.

The validation of the *MATLAB/Simulink* simulation has now paved the way to effective controller synthesis, which will be covered next.

3. Controller Design

This chapter describes different approaches to controller synthesis for the cooling cycle modelled in the last chapter. After having formulated specific design objectives and test setups under which the closed-loop behaviour is evaluated, a simple PI-controller structure is considered and model-free tuning is applied. A next step involves the derivation of a linearised plant model. To enhance the linear range to which the model can be applied, a deterministic algorithm has been conceived, that transforms mass flow rate fractions to valve opening ratio signals. Based on the linearisation, linear quadratic gaussian (LQG) controllers are computed in an $\mathcal{H}_2$ controller synthesis framework, after having formulated a generalised plant. This framework allows for the inclusion of model uncertainties into the plant model, such as uncertain plant parameters. Some parameters of the plant's system matrix will be difficult to measure in practice. A $\mathcal{H}_\infty$ norm based LQG controller synthesis based on [22, p.150ff] is attempted for the modelling of a particular uncertainty. A few others are proposed additionally.

Emphasis will be given to considerations that lead to the different designs, tuning possibilities and comparisons of the achieved performances. Each section will end with a discussion on the results and issues observed.

Theoretical background is reproduced to the amount necessary to understand the synthesis methods in appendix C.

3.1. Design Objectives and Test Setups

This section aims at providing a solid basis for the comparison and evaluation of different controller strategies and structures. A discussion on the resulting closed-loop responses can only be based on a set of previously defined *design objectives* and test conditions under which the simulation is performed. This thesis does not pursue the goal to simulate every possible case of system failure and prove a controller's capability of handling it. Instead, a reasonable but strict test situation will be adopted to provide results on which the worst case behaviour of the controllers can be inferred.

3.1.1. Design Objectives

To facilitate a general nomenclature, it will be made use of a definition common to control theory literature [8, p.369]:

Definition 3.1.1 (Robust Stability and Robust Performance) *Let a set* Π *of plants be given such that the nominal plant* $\boldsymbol{G}$ *is an element of* Π*. Moreover, let a set of performance objectives be given and suppose that* $\boldsymbol{K}$ *is a controller which meets these objectives. Then the feedback system is said to have*

nominal stability (NS)*, if* $\boldsymbol{K}$ *internally stabilises the nominal plant* $\boldsymbol{G}$*;*

robust stability (RS)*, if* $\boldsymbol{K}$ *internally stabilises every plant belonging to* Π*;*

nominal performance (NP)*, if the performance objectives are satisfied for the nominal plant* $\boldsymbol{G}$*;*

robust performance (RP)*, if the performance objectives are satisfied for every plant belonging to* Π*;*

With respect to the cooling cycle plant, the desired closed-loop properties are comprised of the following items and are discussed with regard to their significance:

- **Robust Stability and Robust Performance:** It is desired that the controller provides closed-loop stability under a variety of environmental conditions and time-varying plant-parameters in the sense of definition 3.1.1. To do this analytically might, however, be only possible under some restrictive or conservative assumptions regarding the uncertainty of a mathematical model. The main destabilising factor is supposed to be found in the time-varying dead-times. To support the robustness analysis, the non-linear plant simulation and test setup has to provide enough information to predict the closed-loop behaviour under worst-case conditions.
- **Rejection of Ramp-Shaped Output Disturbance:** More specifically, the most crucial objective is appropriate rejection of output disturbances. This disturbance enters the respective differential equations in the form of a step (the loads Q_{L_i} applied to the cooling cycle). In terms of nomenclature common to control theory these disturbances can be regarded as ramp-shaped *output disturbances* with an adequate slope.
- **Maximum Allowed Overshoot:** As a consequence an important performance criterion to the cooling cycle behaviour in closed-loop is a *maximum allowed temperature* of $T_{max} \approx 70\,°\mathrm{C}$ absolute temperature. The minimum allowed temperature is $T_{min} \approx 30\,°\mathrm{C}$. In addition, temperature differences between the heat exchanger bypass outlet temperature T_0 and the load branch outlet temperatures $T_i, i = 1, 2, 3$ of more than $\Delta T \approx 30\,\mathrm{K}$ are also undesirable. This is a vital condition, in order not to expose the electronics to extreme temperatures and local temperature differences.
- **Reference Tracking:** The control problem will be posed as a reference tracking or regulator problem, respectively. That is, controlled temperatures are to reach a

minimal steady state tracking deviation from the reference value in a finite period of time. Please note, that actually the temperatures are only required to lie within certain bounds, specified by the amount of wear to the electronics. To facilitate the controller design, safe reference temperatures of $T_i = 50\,°\mathrm{C}, i = 1,2,3$ and $T_0 = 40\,°\mathrm{C}$ are desired. Small steady state errors are, however, not crucial.

- **Actuator Constraints and Energy Optimality:** The actuators only operate within certain bounds. The controller may take advantage of the full magnitudes of valve opening ratios, but opened valves imply lower hydraulic resistances. Minimum losses and pump speed are desired, since energy consumption is a major issue in aircraft engineering. A controller should therefore exert reasonable control effort.
- **Noise Rejection and Actuator Wear:** The controller shall not engage in excessive control action involving fast oscillations of high amplitudes. This is to prevent premature wear of mechanical parts within the actuator components. To retain practical relevance, this objective is vital to good controller design. To assess controllers with regard to this objective, output noise of amplitude $\Delta T = \pm 1.5\,\mathrm{K}$ has been added in the non-linear simulation.

Next, it is important to define under which test setups, the controllers are required to fulfill the design requirements.

3.1.2. Test Setups

The non-linear closed-loop simulation will rely on two main *test setups* entitled *"benchmark setup 1/2"*. For simplicity, this thesis will only cover flight conditions within the STANDARD ISA atmosphere. With respect to colder or warmer environments the expectations with regard to the tracking of low reference temperatures have to be alleviated accordingly.

Furthermore, it is hypothesised, that the so called *"hot ground case"* essentially consists of the same design issues, merely incorporating an additional controller, driving the fans to provide a certain air mass flow rate for cooling. The *"hot ground case"* deals with an airplane located on the airfield in a hot environment. For the sake of evaluating the applicability of different types of controllers to the non-linear cooling plant, the *"hot ground case"* does not yield additional crucial issues to cover.

In addition, the controller synthesis will be based on one pump input only. The other pump will act as a redundant device, because it is assumed, that in terms of energy consumption, there will be no significant difference.

Benchmark Setup 1

The *"benchmark setup 1"* describes an airplane flying at typical long range cruise height of $h = 10,000\,\mathrm{m}$ with a velocity $c = 300\,\mathrm{m/s}$. In stable equilibrium position, the loads Q_{L_i} are applied in the pattern depicted by figure 3.1.1.

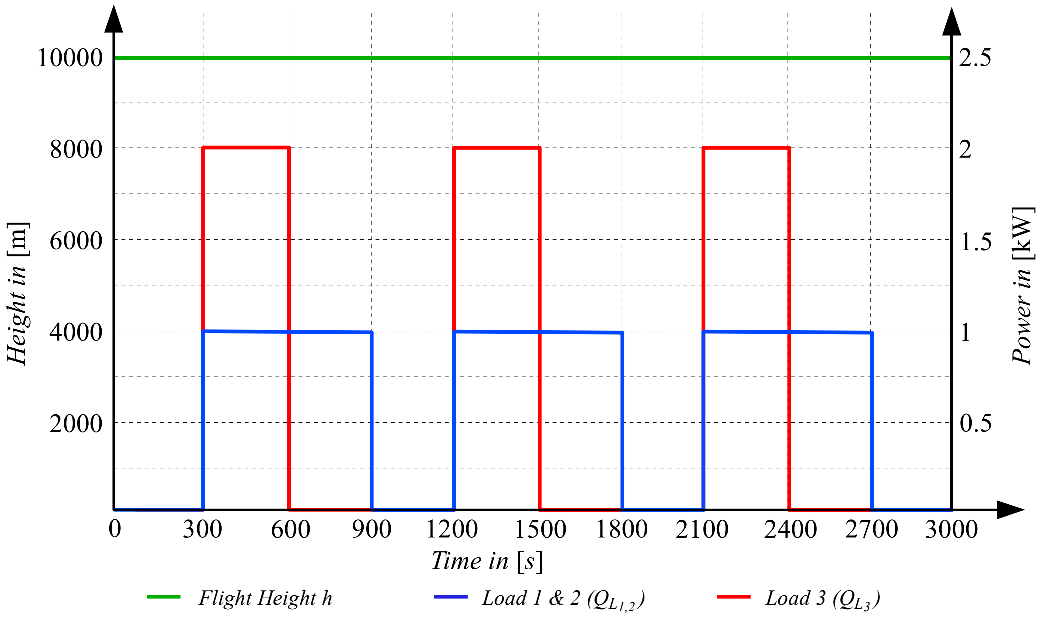


Figure 3.1.1.: Controller Benchmark Setup 1

Benchmark Setup 2

"Benchmark setup 2" describes an exaggerated starting phase of an airplane's flight envelope. The aircraft is supposed to fly at constant velocity of $c = 300\,\mathrm{m/s}$ while rapidly acquiring height in two separate phases:

- First, the airplane rises with a slope of 8000 m per 300 s, starting from altitude $h_0 = 0\,\mathrm{m}$.
- After having reached an altitude of $h_1 = 8000\,\mathrm{m}$, the climb is reduced to 25 %, which means 2000 m per 300 s.
- At a total simulation time of $t = 600\,\mathrm{s}$, the airplane acquires the cruise altitude of $h_2 = 10,000\,\mathrm{m}$ and holds it.

During the climb, the loads are applied at a constant level. Figure 3.1.2 illustrates the setup.

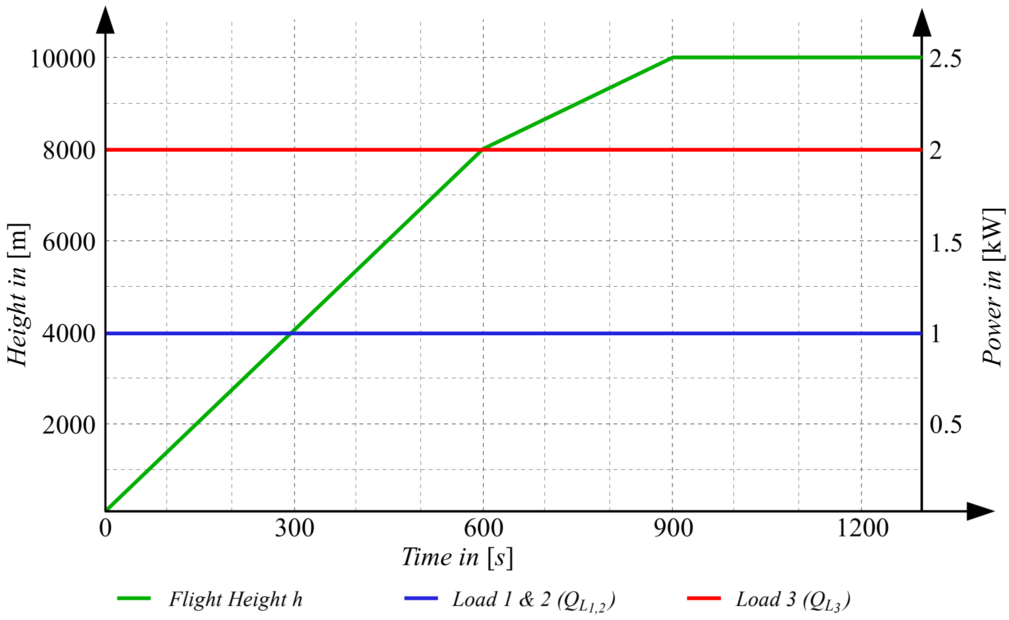


Figure 3.1.2.: Controller Benchmark Setup 2

Measurement Noise

Because of the turbulent flow characteristics inherent to the system and sensor inaccuracies, temperature measurement is assumed to be prone to measurement noise. This noise is simulated as zero-mean, white noise and is arbitrarily chosen to have amplitudes of $\approx 1.5\,\text{K}$.

3.2. Heuristically Tuned PI-Controller

PI– and PID-controllers are among the most widely used controller types in control theory applications. Their behaviour is well known and in addition to some various rigorous single-input-single-output (SISO) controller design techniques, like the root locus method, they can also be tuned intuitively or by heuristic methods. This implies, that no mathematical formulation of the plant model is needed to design a PID-controller, though it greatly facilitates the tuning process. On the contrary heuristic tuning of controllers always depends on the availability of some experimental setup. Typically this is a physical prototype of the plant or an already built system on which the experiments may be conducted. In the case of this thesis, the plant is a virtual non-linear simulation as derived in the previous chapter. Originally some similar tuning of a PI-controller has been done in the simulation software *Flowmaster*, which does not require the engineer to understand all the equations that the simulation is based upon.

All this makes PID-controllers applicable for some class of non-linear or time-varying plants. The desired performance specifications, however, should be confined to stability and reference tracking and should in general not include any demands regarding the dynamic behaviour.

Many practical control tasks need an anti-windup configuration, since a small overshoot is often a vital performance objective. This is also true in the case of the cooling cycle, where the overshoot of the temperatures needs to be kept as low as possible, in order not to expose the electronics to high absolute temperatures and temperature fluctuations.

The actual design carried out on the cooling cycle plant will be explained next.

3.2.1. Design of Decentralised PI-Controllers for the Cooling Cycle

Since the cooling cycle is a multiple-input-multiple-output (MIMO) plant, one could either try to design a single centralised PI-controller, or a set of independent single-input-single-output (SISO) PI-controllers, whose control error feeds are chosen in an intuitive fashion. Though there exist heuristic tuning rules for MIMO PI-controllers [7, p.203], intuitive tuning requires a maximum awareness of changes that occur to the dynamic behaviour if a single "tuning knob" is changed. Therefore the complex task of controlling a MIMO plant should be split up, in order to analytically determine weaknesses.

The actual design of a decentralised controller can be divided into three parts:

1. **PI-control of the pump speed** n_P with the aim to provide enough mass flow rate $\dot{m}_0$ available for the cooling process
2. **PI-control of the load branch control valves** v_i with the aim to divide the total mass flow rate $\dot{m}_0$ into fractions $\dot{m}_i$ required to stabilise the load outlet temperatures T_i, $i = 1, 2, 3$.

3. **PI-control of the heat exchanger bypass valve** v_{HE} with the aim to stabilise the heat exchanger bypass outlet temperature T_0.

Figure 3.2.1 depicts the controller structure designed. The following sections will describe the individual PI-controllers listed above.

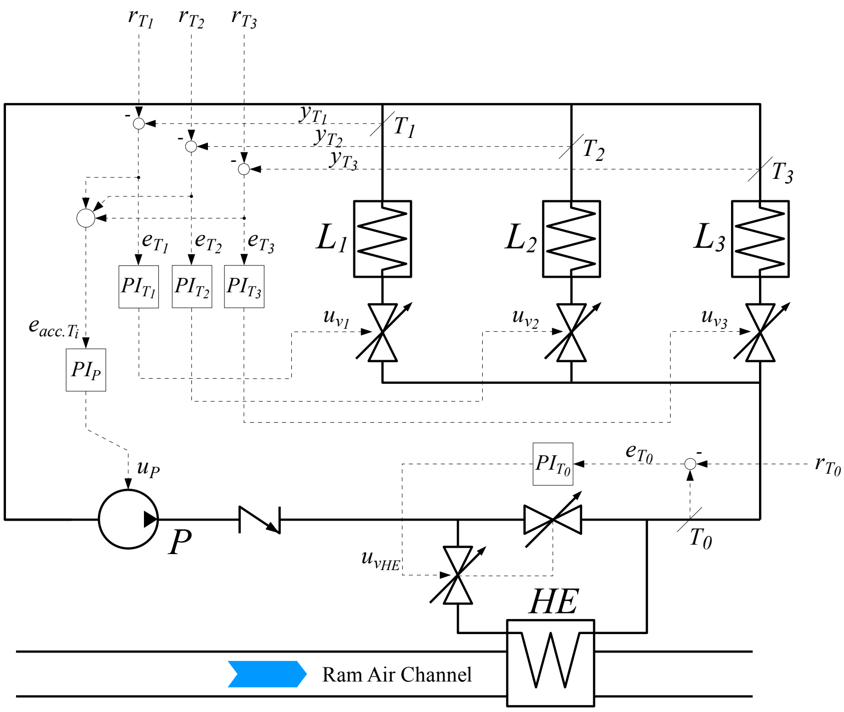


Figure 3.2.1.: PI-Controller Structure Scheme

Pump PI-Controller

The control error $e_{acc.T_i}$ entering the pump PI-controller is the accumulated control error of all load branches $e_{T_i}, i = 1, 2, 3$.

$$e_{acc.T_i} = \sum_{i=1}^{3} (y_{T_i} - r_{T_i}) \tag{3.2.1}$$

To avoid confusion, it should be noted, that the control error is the negative of the control error as defined in common literature. Generally it is defined as $e = r - y$, but since higher control outputs result in lower plant outputs (i.e. temperatures) in the case of the cooling cycle, this relation would be reflected by negative tuning parameters.

The resulting controller dynamics without considering the output saturation limit are as follows:

$$\dot{x}_{I_P} = e_{acc.T_i} \tag{3.2.2}$$
$$u_{n_P} = k_{p_P} e_{acc.T_i} + k_{I_P} x_{I_P} \tag{3.2.3}$$

The output saturation limit for the pump rotational speed u_P is defined by the operational range $[0...2500]\,\mathrm{min}^{-1}$. With the help of the Ziegler-Nichols tuning rules a set of controller parameters has been found, that has undergone some further manual adjustment. This includes finding an appropriate anti-windup gain k_{aw}, which is non-trivial and is best done by analysing the respective plots. Table 3.2.1 lists the obtained controller parameters.

Parameter	**Value**
k_{p_P}	0.3
T_{I_P}	6
k_{aw_P}	5

Table 3.2.1.: Final Pump PI-Controller Parameters by Manual Tuning

Load Branch Control Valves PI-Controller

The PI-controller of the load branch valves $v_i, i = 1, 2, 3$ does actually consist of three separate and independent controllers of the same structure. The control error e_{T_i} is simply calculated from

$$e_{T_i} = (y_{T_i} - r_{T_i}), \qquad \text{for } i = 1, 2, 3 \tag{3.2.4}$$

whereas the controller dynamics without saturation limits may be denoted by

$$\dot{x}_{I_{T_i}} = e_{T_i} \tag{3.2.5}$$
$$u_{v_i} = k_{p_{T_i}} e_{T_i} + k_{I_{T_i}} x_{I_{T_i}} \tag{3.2.6}$$

The controller output may not exceed the valve opening ratios [0...1]. Table 3.2.1 lists the obtained controller parameters.

Heat Exchanger Bypass Control Valves PI-Controller

The PI-controller of the heat exchanger bypass valves v_{HE} and v_b is a single controller with exactly the same structure as each of the three load branch PI-controllers.

$$e_{T_0} = (y_{T_0} - r_{T_0}) \tag{3.2.7}$$

Parameter	Value
$k_{p_{T_i}}$	0.04
$T_{I_{T_i}}$	4
$k_{aw_{T_i}}$	100

Table 3.2.2.: Final Load Branch Control Valve PI-Controller Parameters by Manual Tuning

Again, the controller dynamics without saturation limits may be written as

$$\dot{x}_{I_{T_0}} = e_{T_0} \tag{3.2.8}$$

$$u_{v_{HE}} = k_{p_{T_0}} e_{T_0} + k_{I_{T_0}} x_{I_{T_0}} \tag{3.2.9}$$

So far, this controller might have well been included into the set of the previously mentioned valve controllers. It has nevertheless been separated from the others for two reasons:

1. The controlled subjects are locally remote. It should be taken into account that the time-varying dead-time due to the transport of mass between the loads and the heat exchanger bypass significantly affects the plants non-linear dynamics. Since the influence of the temperature T_0 entering the loads is not taken into account during design, individual adjustments might be needed. For instance, it has been observed during the tuning phase, that reducing the integrator gain and therefore lowering the integrator's time constant, leads to fewer oscillations in the queued load branch outlet temperatures. A quicker control of T_0 results in less input disturbances with regard to the load branches.
2. The primary objective for controller design is to drive the control error e_{T_i} of the load branch temperatures towards zero. Controlling for a constant heat exchanger bypass outlet temperature T_0 is of lower priority as long as T_0 does not breach the margin critical to temperature shocks for the electronics.

Table 3.2.1 lists the obtained controller parameters.

Parameter	Value
$k_{p_{T_0}}$	0.04
$T_{I_{T_0}}$	3.5
$k_{aw_{T_0}}$	100

Table 3.2.3.: Final Heat Exchanger Bypass Control Valve PI-Controller Parameters by Manual Tuning

3.2.2. Discussion

With the help of heuristic tuning measures a set of PI-controllers has been designed, that meets the desired objectives of stability and sufficient reference tracking. The robustness with regard to the non-linear simulation and the benchmarking setups is given. Figures 3.2.2 and 3.2.3 show, that the temperatures never exceed 60 °C. Lower temperatures are acquired, which is acceptable within the range present. Relatively strong peaks can be observed, where it is desirable to have a smoother curve, but the manual tuning of the PI-controllers showed, that a smoother curve would also lead to higher peaks.

A major disadvantage of the PI-controllers is the fact, that their speed of response is directly coupled with the amplification of measurement noise. In the results shown by the plots, the controller output oscillations actually are too strong. It should be noted however, that — though PI-controllers are sensitive to perturbations — the intensity of the measurement noise has been chosen arbitrarily and may not reflect circumstances present in practical application. Application of simple low-pass filters with corner frequency $\omega_{lp} = 1\frac{1}{\mathrm{s}}$ to the measurement signals results in lower controller output oscillations, but further degrades the control performance. The maximum temperatures now breach the border of 60 °C and more violent oscillations occur until the temperatures settle for a stable equilibrium near the reference value. The heat exchanger bypass outlet temperature T_0 even crosses the lower margin of 30 °C for the fraction of a second. Figure 3.2.4 shows the simulation results when the measurement signals are filtered.

Another disadvantage to the chosen PI-controller structure is the fact, that every single PI-controller is effectively operating on a SISO system. Generally this leads to a considerable performance degradation, because of conflicting controller outputs. A certain equilibrium has to be reached, which is not guaranteed to be optimal in any sense. This can be observed in figure 3.2.2, where during the phase of no loads applied, the valve openings are changed without effect. The controller is not aware of the fact, that the valve control is in vain, for the total mass flow rate is zero. Yet still the control error is simply integrated, leading to almost unforeseen initial conditions at the brink of the next application of loads to the circuit. While this is beneficial in the case of the heat exchanger bypass, the control valves v_1, v_2 and v_3 are completely shut when the loads are applied again. This leads to a high peak in the temperature curves, since the valves take slightly more than four seconds to fully open again. Furthermore, the overall hydraulic resistance is very high, when the pump starts to rotate again. In practice, this would lead to considerable losses. These effects can be alleviated by more sophisticated controller structures, involving extra rules and scheduling parameters that modify the controllers' outputs. For the PI-controller structure, such adjustments will not be considered in this thesis, because, the main focus lies on the synthesis of a controller, that can operate safely on a minimum of sensor signals.

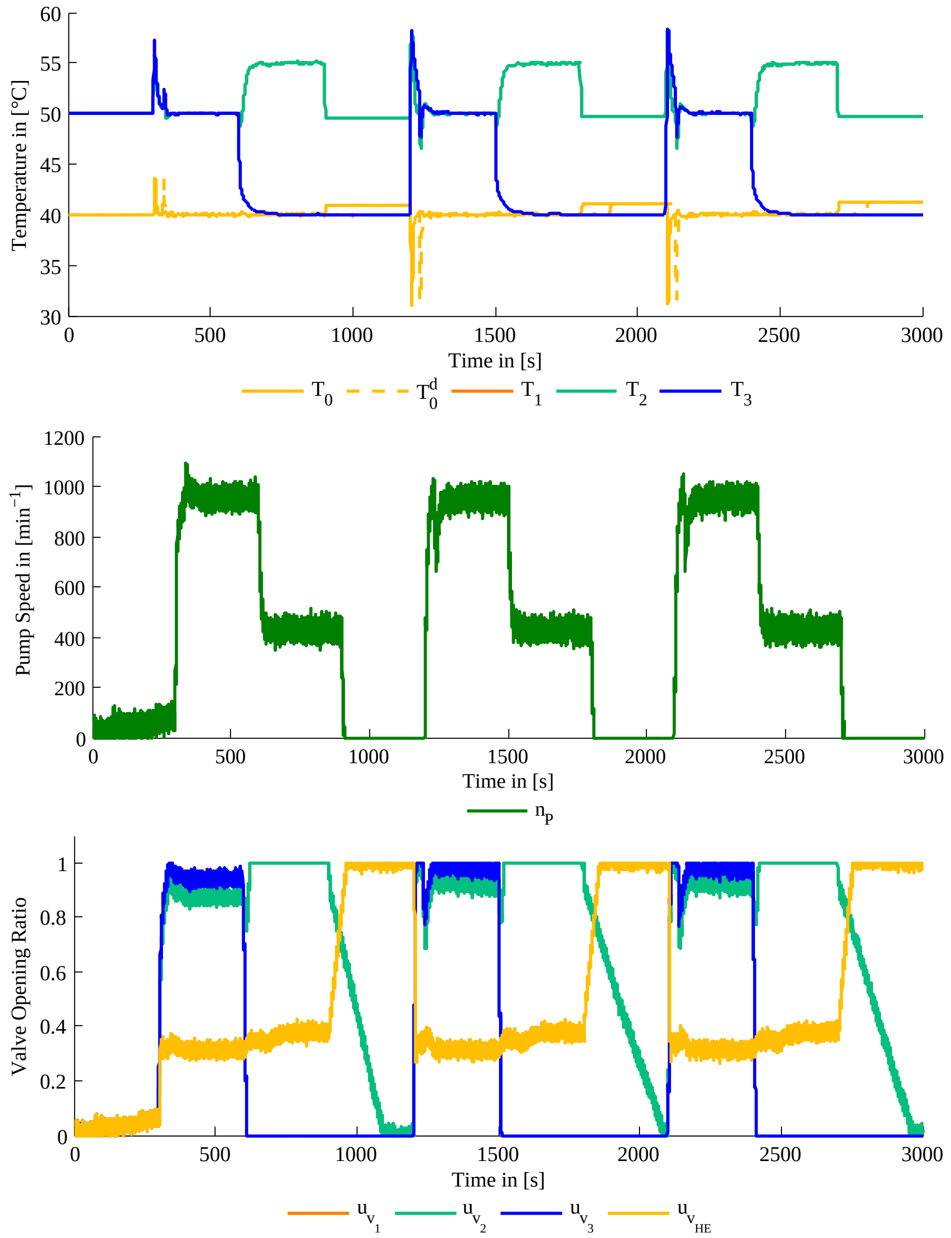


Figure 3.2.2.: PI-Controller Setup: Simulation Results — Benchmark Setup 1

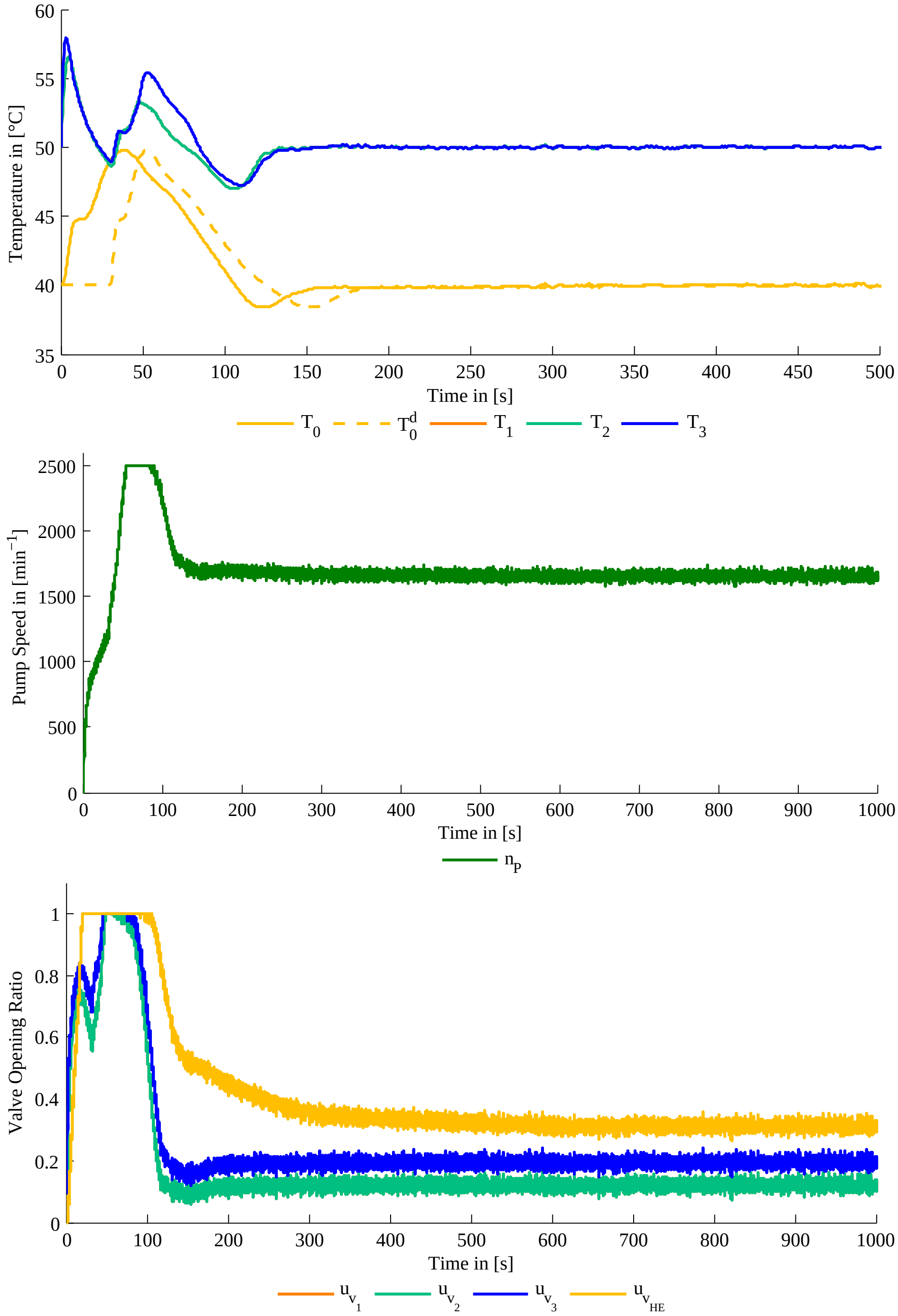


Figure 3.2.3.: PI-Controller Setup: Simulation Results — Benchmark Setup 2

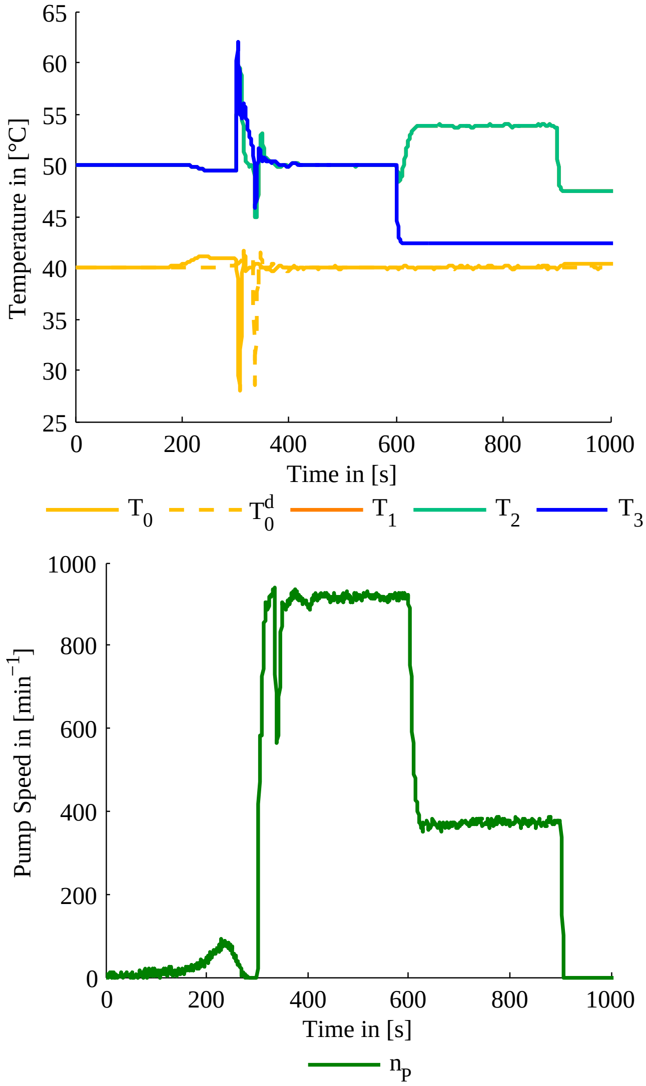


Figure 3.2.4.: PI-Controller Setup with Filtered Measurement Signals: Temperature Curves (t.) and Pump Speed Output (b.) — Benchmark Setup 1

Summary

Because there is no need to derive a mathematical model linearised around a desired equilibrium, the PI-controllers are an attractive choice when it comes to easy implementation. However, the simulation results showed, that the designer should be seriously concerned with the effect of measurement noise on the controller performance. The results, though inacceptable in the present configuration, should not be understood, that the PI-controller configuration is infeasible for the cooling cycle in general. With less measurement noise, i.e. high quality sensors, the controller output oscillations could be brought to an acceptable level.

With regard to energy consumption, the PI-controller structure seems to actually approach optimality already. The pump speed output quickly assumes fixed and rather low levels in the range of only about $[0...1000\,\mathrm{min}^{-1}]$. The controller output noise, however, leads to the mechanical components being subjected to heavy wear — the consequence of the noise amplification issue already mentioned.

The PI-controller structure involves only five integrators, thus having a controller order of 5, a quantity, that is frequently used to measure the computational complexity of a given controller. This order is particularly low and the implementation will not demand for extraordinary hardware, which is always an advantage of simple PID variant controller types.

The overall performance could be improved by a scheduling algorithm, which, for example, opens the load branch control valves during zero pump action. Yet, more complex strategies will come at the expense of more sensors and lower reliability in case of failure. Furthermore, to effectively design online tuning of controller parameters, model-based tuning, e.g. via root locus plots, is probably a preferable way.

The control of the valve openings in the absence of loads, can be considered to be some kind of malfunction. This could be improved by the design of a multivariable controller, which is investigated in the following sections.

3.3. Derivation of Linearised Plant Models

Most modern controller design algorithms and synthesis methods rely on mathematical models of the plant in state space form. $\mathcal{H}_\infty$ and $\mathcal{H}_2$ based controller design further takes advantage of the concept of a *generalised plant*, where fictitious inputs and outputs are added to the system. The weighted fictitious outputs are then subject to minimisation problems, for example, to guarantee a certain degree of optimality or robustness.

In the following sections, a linearised model of the plant is derived, on which to base controller synthesis on. First, a set of input and output variables is defined, followed by the definition of a general controller scheme for . This decentralised scheme is bound to limitations arising from the time-varying dead-times. The next section then verifies the feasibility of the selected variables: A *deterministic transformation* on the valve opening variables is proposed to remove the non-linear relations between mass flow rate fractions and valve openings. This enables the application of standard linearisation procedures to the remaining portion of the system's differential equations. In the end, system equilibria are discussed.

3.3.1. Plant Model Inputs and Outputs

In section 2.5.3 the feasibility of the measurement of various plant parameters has been discussed briefly. This showed, that reliable measurement of mass flow rates cannot be guaranteed. Therefore, controller outputs have to stick to pump speed n_P and valve openings v_{HE}, v_i, $i = 1, 2, 3$, while the control error will rely on the measurements of temperatures T_i, $i = 0, 1, 2, 3$ alone. The next section further proves, that it is possible to avoid the inclusion of the non-linear relations $v_{HE}(\beta_{HE})$ and $v_i(\beta_i)$, such that the controller may directly operate on mass flow rate fractions as plant inputs. In summary, the following set of linearised plant model inputs and outputs is deemed to be sufficient:

$$\boldsymbol{u} = \left.\begin{pmatrix} n_P \\ \beta_1 \\ \beta_2 \\ \beta_{HE} \end{pmatrix}\right\} \text{Inputs of linearised plant model}$$

$$\boldsymbol{y} = \left.\begin{pmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \end{pmatrix}\right\} \text{Outputs of linearised plant model}$$

It should be noted, that the number of plant inputs and outputs are the same. In general this implies, that the control task is well-posed in a sense, that there is neither lack of information, nor conflicting input behaviour. This, however, must not necessarily be the case: β_1 and β_2 cannot assume values larger than 0.5 at the same time, hence these inputs can be conflicting. This should be kept in mind, as controller design continues.

3.3.2. Decentralised Controller Scheme

Not much attention has been paid to the occurrence of *time-varying dead-times* so far. These surely affect the possible controller structures and can be treated in numerous ways. Some of them are explained in [14], which mainly focusses on widely industry-used first to second order plant models with dead-times. In [14, p.30] it is further stated, that the representation of *dead-time* in state space is difficult. This becomes apparent, when looking at the TAYLOR series of the LAPLACE-transform of a dead-time term:

$$e^{-t_d s} = \frac{1}{1 + \sum_{i=1}^{\infty} \frac{(st_d)^i}{i!}}$$

From this it can be inferred, that a dead-time process would actually include an infinite amount of additional states (infinite poles). This can be compensated for in discrete time domain, where only a multiplicity of sampling durations is added as additional states. This, however, does not address time-varying dead-times, which could be modelled with the help of an uncertain time delay.

Due to the difficulties arising from time delays, the plant will be controlled by a decentralised controller structure. This subdivides the control of desired temperatures $T_i, i = 0, 1, 2, 3$ into a set of two independent controllers. One controller tries to hold the heat exchanger bypass outlet temperature T_0 at a safe and constant level, while a second controller controls the pump flow and mass flow rate fractioning to make sure, the loads are not exposed to high temperatures. This is valid under the following assumptions:

- **No Delocation:** The temperature sensors can be placed sufficiently close to the actuators.
- **Fast Computation:** No time delay arises from controller computations or signal lag.
- **Robustness:** The respective controllers can be designed to be robust against varying disturbances to the inlet temperatures of the respective plants.

Under these assumptions block T^1 and T^2 of the plant (as defined in in the previous chapter under section 2.4.3) can be controlled individually.

Figure 3.3.1 depicts the general decentralised controller structure during non-linear simulation. A simplified illustration of the control of the two resulting linearised nominal plant models can be seen in figure 3.3.2. The figures also explain the nomenclature used to differentiate between both control loops.

3.3.3. A Deterministic Transformation on the Valve Opening Variables

Since state space models are linear models, it is a good idea to find a set of input and output variables in advance, which behave as linearly as possible. If a non-linear and *deterministic transformation* can be found to connect at least some subset of the "true"

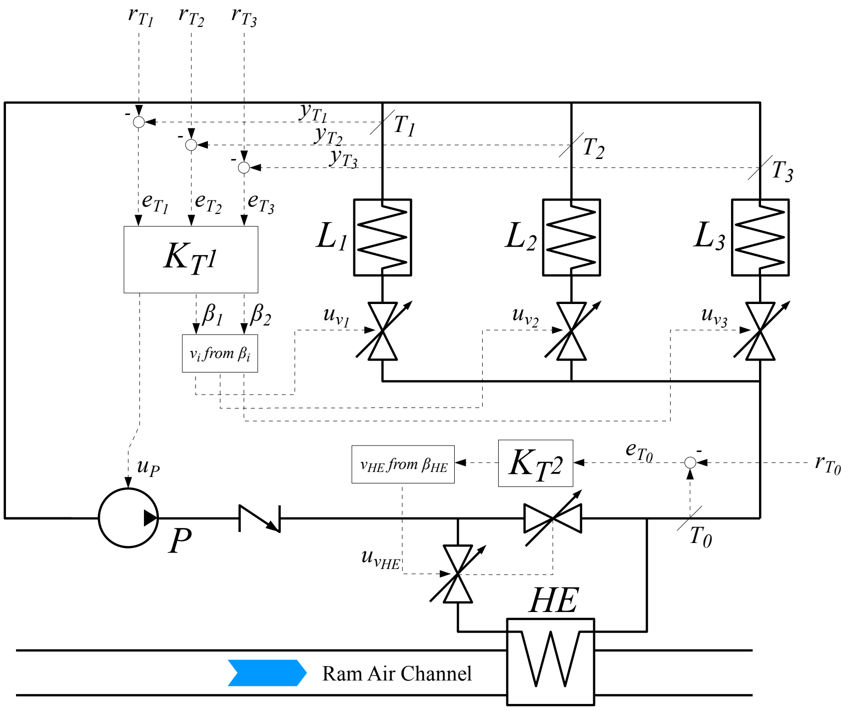


Figure 3.3.1.: General Decentralised Controller Scheme

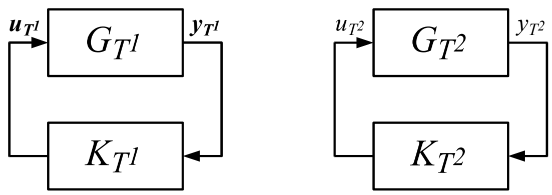


Figure 3.3.2.: Nominal Controller Scheme of Plant Blocks T^1 and T^2

inputs and outputs of the system to a set of "virtual" inputs and outputs, controller design can continue without methods inherent to non-linear control. By "virtual", inputs and outputs are meant to be no real signal, one might be able to measure along a wire or by some other means.

This section focusses on the proposal of such a transformation between the valve opening ratios v_{HE}, v_i and the mass flow rate fractions β_{HE}, β_i for $i = 1, 2, 3$. This transformation provided, a linearised state space model could rely on outputs $T_i, i = 0, 1, 2, 3$ and inputs $n_P, \beta_{HE}, \beta_i, i = 1, 2$ alone, which would mean having as few inputs as outputs. This is in general a desired situation, because either no redundancy with the inputs or lack of information due to too few outputs occurs.

For convenience, the relevant equations 2.2.13, 2.2.14, 2.2.15 and 2.2.16 will be reproduced here:

$$\dot{m}_i = \beta_i \cdot \dot{m}_0 \qquad , \beta_i = \frac{\sqrt{\frac{R_{v_1} \cdot R_{v_2} \cdot R_{v_3}}{R_{v_i}}}}{\sqrt{R_{v_1} \cdot R_{v_2}} + \sqrt{R_{v_1} \cdot R_{v_3}} + \sqrt{R_{v_2} \cdot R_{v_3}}} \quad , i = 1, 2, 3 \tag{3.3.1}$$

$$\begin{aligned} \dot{m}_{HE} &= \beta_{HE} \cdot \dot{m}_0 \\ \dot{m}_b &= \beta_b \cdot \dot{m}_0 \end{aligned} \quad , \beta_{HE} = \frac{\sqrt{R_{v_b}}}{\sqrt{R_{v_b}} + \sqrt{R_{v_{HE}}}} \text{ and } \beta_b = \frac{\sqrt{R_{v_{HE}}}}{\sqrt{R_{v_b}} + \sqrt{R_{v_{HE}}}} \tag{3.3.2}$$

$$R_v = K_v(v) \cdot \frac{1}{2 \varrho_P A_v^2} \tag{3.3.3}$$

$$K_v(v) = e^{\Pi_3(v)} = e^{p_1 \cdot v^3 + p_2 \cdot v^2 + p_3 \cdot v + p_4} \tag{3.3.4}$$

For simplicity assume now, that the valve cross section areas do not significantly differ, such that all relations can be expressed with only the resistance coefficients $K_v(v)$.

Mass Flow Rate Fraction β_{HE} to Valve Opening v_{HE}: $v_{HE}(\beta_{HE})$

By division and taking the square, one can obtain:

$$\frac{\beta_{HE}^2}{\beta_b^2} = \frac{K_{v_b}}{K_{v_{HE}}} = \frac{e^{\Pi_3(v_b)}}{e^{\Pi_3(v_{HE})}} = e^{p_1 \cdot (v_b^3 - v_{HE}^3) + p_2 \cdot (v_b^2 - v_{HE}^2) + p_3 \cdot (v_b - v_{HE})} \tag{3.3.5}$$

Since $\beta_b = 1 - \beta_{HE}$ and $v_{HE} = 1 - v_b$, 3.3.6 yields an exponential/cubic non-linear relation between β_{HE} and v_{HE}:

$$\frac{\beta_{HE}}{(1 - \beta_{HE})} = e^{1/2 \cdot \left(p_1 \cdot \left((1 - v_{HE})^3 - v_{HE}^3\right) + p_2 \cdot \left((1 - v_{HE})^2 - v_{HE}^2\right) + p_3 \cdot ((1 - v_{HE}) - v_{HE})\right)} \tag{3.3.6}$$

Although this cannot be easily solved for v_{HE}, the relation can be calculated for values ranging from $v_{HE} = 0, ..., 1$. The result and its inversion is plotted in figure 3.3.3. This information can be exploited in the form of a *look-up table*, such that the controller may mathematically act on the input β_{HE}, which is then transformed to an actual *valve opening ratio* signal v_{HE}. The opening signal has to be transformed into a "real" voltage signal in a real application, of course.

Mass Flow Rate Fractions β_i to Valve Openings v_i: Minimisation of Pressure Losses

With regard to the control valves $v_i, i = 1, 2, 3$, there is no inherent coupling, which means, that all valves can be opened to an arbitrary extent simultaneously. This implies,

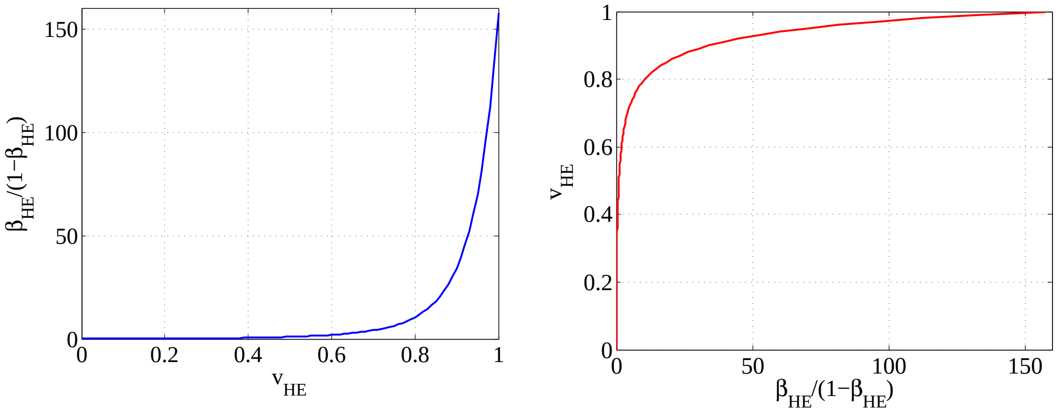


Figure 3.3.3.: Deterministic System Input Transformation $\beta_{HE}(v_{HE})$ And $v_{HE}(\beta_{HE})$

that it is possible to divide the total mass flow rate $\dot{m}_0$ into given mass flow rate fractions in an unlimited number of ways. Theoretically, it would be even possible to close every valve and still obtain an evenly divided flow. Naturally, this leads to an overall hydraulic resistance $R'_p \to \infty$ and therefore infinite losses. The following *algorithm* provides a way to find valve opening values, that achieve the desired fractioning of the total mass flow rate, while minimising the losses occurring from hydraulic resistances.

Suppose, that the maximum and minimal resistance coefficients $K_{v_{max}}$ and $K_{v_{min}}$ are known, for instance, from experimental data. The controller output may include β_1 and β_2 only, since the continuity equation holds A.1.5:

$$\beta_1 + \beta_2 + \beta_3 = 1 \quad \text{and thus} \quad \beta_3 = 1 - \beta_1 - \beta_2 \tag{3.3.7}$$

In that way, all desired mass flow rate fractions can be acquired and are made components of the vector $\boldsymbol{\beta} = \begin{pmatrix} \beta_1 & \beta_2 & \beta_3 \end{pmatrix}^T$. The basic idea is to fully open the valve corresponding to the largest mass flow fraction and set the opening ratios of the others accordingly. The algorithm is presented in pseudo-code:

1. Set every resistance coefficient K_{v_i} of valve $i = 1$ to 3, to minimum $K_{v_{min}}$
2. Sort entries of $\boldsymbol{\beta}$ in descending order and remember the permutation of components in vector $\boldsymbol{j}$. (Vector entries will be referenced by $\boldsymbol{vector}(index)$.)
3. For every iteration starting from $l = 1$ to 3 set the resistance coefficient of the valve belonging to the largest remaining mass flow fraction β_i to:

$$K_{v_{i=\boldsymbol{j}(l)}} = \left(\frac{\text{maximum fraction } \beta_{i_{max}}}{\boldsymbol{\beta}(i = \boldsymbol{j}(l))} \right)^2 \cdot K_{v_{min}} \tag{3.3.8}$$

4. If in each iteration $l = 1$ to 3 the respective resistance coefficient is larger than the maximum value $K_{v_{max}}$, set it to the maximum:

$$K_{v_{i=\boldsymbol{j}(l)}} > K_{v_{max}} \tag{3.3.9}$$

5. Calculate each valve opening ratio v_i from logarithmic fitting equation:

$$v_{i=j(l)} = p_1 \cdot \ln(K_{v_{i=j(l)}})^3 + p_2 \cdot \ln(K_{v_{i=j(l)}})^2 + p_3 \cdot \ln(K_{v_{i=j(l)}}) + p_4 \tag{3.3.10}$$

By increasing the number of iterations, an arbitrary set of parallel valves can be controlled this way.

3.3.4. Linearisation around Equilibrium Positions

With a properly defined controller structure, it is now possible to derive the linearised plant *state space model* from the non-linear equations. For convenience, the required equations (2.2.12, 2.4.18, 2.4.19) the linearisation depends on, will be recapitulated here and assigned to block T^1 and T^2:

Non-Linear Equations — Block T^1

$$\begin{aligned}
H_P^*\left(\frac{\dot{m}_0^*}{n_P^*}\right) \cdot \frac{H_{R_P} \cdot \varrho_P \cdot g}{n_{R_P}^2} \cdot {n_P}^2 - R_p' \cdot \dot{m}_0^2 - L_p' \cdot \ddot{m}_0 = 0 \\
M_{L_1} c_{v_P} \cdot \dot{T}_1 - \dot{m}_1 c_{p_P} \cdot T_0^d + \dot{m}_1 c_{p_P} \cdot T_1 - \dot{Q}_{L_1} = 0 \\
M_{L_2} c_{v_P} \cdot \dot{T}_2 - \dot{m}_2 c_{p_P} \cdot T_0^d + \dot{m}_2 c_{p_P} \cdot T_2 - \dot{Q}_{L_2} = 0 \\
M_{L_3} c_{v_P} \cdot \dot{T}_3 - \dot{m}_3 c_{p_P} \cdot T_0^d + \dot{m}_3 c_{p_P} \cdot T_3 - \dot{Q}_{L_3} = 0
\end{aligned}$$

Since the controller directly operates on mass flow rate fractions with $\dot{m}_i = \beta_i \cdot \dot{m}_0$, the dynamics of the control valves have to be taken into account by *prefiltering* these inputs:

$$\dot{\beta}_i + \frac{1}{\tau_v} \cdot \beta_i - \frac{1}{\tau_v} u_{\beta_i} = 0$$

Non-Linear Equations — Block T^2

$$\begin{aligned}
M_{HE} c_{v_P} \cdot \dot{T}_{HE} - \dot{m}_{HE} c_{p_P} \cdot T_5 + \dot{m}_{HE} c_{p_P} \cdot T_{HE} + kA(\dot{m}_{HE}, \dot{m}_R) \cdot QC \cdot (T_{HE} - T_{R_1}) = 0 \\
M_{HE_R} c_{v_A} \cdot \dot{T}_{R_1} - \dot{m}_R c_{p_A} \cdot T_{R_0} + \dot{m}_R c_{p_A} \cdot T_{R_1} - kA(\dot{m}_{HE}, \dot{m}_R) \cdot QC_R \cdot (T_{HE} - T_{R_1}) = 0 \\
M_{J_1} c_{v_P} \cdot \dot{T}_0 - \dot{m}_b c_{p_P} \cdot T_5 - \dot{m}_{HE} c_{p_P} \cdot T_{HE} + \dot{m}_0 c_{p_P} \cdot T_0 = 0
\end{aligned}$$

Again, the mass flow rate fraction with $\dot{m}_{HE} = \beta_{HE} \cdot \dot{m}_0$ needs to be prefiltered:

$$\dot{\beta}_{HE} + \frac{1}{\tau_v} \cdot \beta_{HE} - \frac{1}{\tau_v} u_{\beta_{HE}} = 0$$

Block T^2

[8, p.86] explains a systematic way to obtain the linearised equations. First, a set of state variables, inputs and disturbances are defined and noted as vectors $\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}$:

$$\boldsymbol{x_{T^1}} = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ \dot{m}_0 \\ n_P \\ \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} x_{1_{T^1}} \\ x_{2_{T^1}} \\ x_{3_{T^1}} \\ x_{4_{T^1}} \\ x_{5_{T^1}} \\ x_{6_{T^1}} \\ x_{7_{T^1}} \end{pmatrix} \quad \boldsymbol{u_{T^1}} = \begin{pmatrix} n_P \\ u_{\beta_1} \\ u_{\beta_2} \end{pmatrix} = \begin{pmatrix} u_{1_{T^1}} \\ u_{2_{T^1}} \\ u_{3_{T^1}} \end{pmatrix} \quad \boldsymbol{d_{T^1}} = \begin{pmatrix} Q_{L_1} \\ Q_{L_2} \\ Q_{L_3} \\ T_0^d \end{pmatrix} = \begin{pmatrix} d_{1_{T^1}} \\ d_{2_{T^1}} \\ d_{3_{T^1}} \\ d_{4_{T^1}} \end{pmatrix}$$

$$\boldsymbol{x_{T^2}} = \begin{pmatrix} T_0 \\ T_{HE} \\ T_{R_1} \\ \beta_{HE} \end{pmatrix} = \begin{pmatrix} x_{1_{T^2}} \\ x_{2_{T^2}} \\ x_{3_{T^2}} \\ x_{4_{T^2}} \end{pmatrix} \quad \boldsymbol{u_{T^2}} = \begin{pmatrix} u_{\beta_{HE}} \end{pmatrix} = \begin{pmatrix} u_{1_{T^2}} \end{pmatrix} \quad \boldsymbol{d_{T^2}} = \begin{pmatrix} T_{R_0} \\ \dot{m}_R \end{pmatrix} = \begin{pmatrix} d_{1_{T^2}} \\ d_{2_{T^2}} \end{pmatrix}$$

It should be noticed, that all non-linear equations are to be expressed with the help of constants or terms comprised of the state, input or disturbance variables. This becomes important, when later the Jacobian is computed. Important dynamics may be missed, if a certain partial derivative is being overlooked. In case of the cooling system equations, the mass flow rates should be expressed as products of the mass flow rate fraction and total mass flow: $\dot{m}_i = \beta_i \cdot \dot{m}_0$ and $\dot{m}_{HE} = \beta_{HE} \cdot \dot{m}_{HE}$.

For both blocks $T^m, m = 1, 2$ the *non-linear equations* can then be written in the form

$$\begin{aligned} 0 &= \boldsymbol{f_{T^m}} \left(\boldsymbol{\dot{x}_{T^m}}(t), \boldsymbol{x_{T^m}}(t), \boldsymbol{u_{T^m}}(t), \boldsymbol{d_{T^m}}(t) \right) \\ \boldsymbol{y_{T^m}}(t) &= \boldsymbol{g_{T^m}} \left(\boldsymbol{x_{T^m}}(t), \boldsymbol{u_{T^m}}(t), \boldsymbol{d_{T^m}}(t) \right) \end{aligned}$$

Let now $\boldsymbol{x^0_{T^m}}, \boldsymbol{u^0_{T^m}}, \boldsymbol{d^0_{T^m}}$ denote the equilibrium positions and suitable linear deviations from these may be written as

$$\boldsymbol{\Delta x_{T^m}} = \boldsymbol{x_{T^m}} - \boldsymbol{x^0_{T^m}}, \quad \boldsymbol{\Delta u_{T^m}} = \boldsymbol{u_{T^m}} - \boldsymbol{u^0_{T^m}}, \quad \boldsymbol{\Delta d_{T^m}} = \boldsymbol{d_{T^m}} - \boldsymbol{d^0_{T^m}},$$

Since $\boldsymbol{f_{T^m}}$ is *differentiable*, the linearised state matrices can be obtained by *partial differentiation* with regard to states, inputs and disturbances. $\boldsymbol{J_{f_{T^m}}}$ denotes the respective *Jacobian* of $\boldsymbol{f_{T^m}}$.

$$\begin{aligned} 0 &= \boldsymbol{f_{T^m}} \left(\boldsymbol{\Delta \dot{x}_{T^m}}, \boldsymbol{x^0_{T^m}} + \boldsymbol{\Delta x_{T^m}}, \boldsymbol{u^0_{T^m}} + \boldsymbol{\Delta u_{T^m}}, \boldsymbol{d^0_{T^m}} + \boldsymbol{\Delta d_{T^m}} \right) \\ &\approx \boldsymbol{f_{T^m}} \left(\boldsymbol{0}, \boldsymbol{x^0_{T^m}}, \boldsymbol{u^0_{T^m}}, \boldsymbol{d^0_{T^m}} \right) + \boldsymbol{J_{f_{T^m}}} \left(\boldsymbol{0}, \boldsymbol{x^0_{T^m}}, \boldsymbol{u^0_{T^m}}, \boldsymbol{d^0_{T^m}} \right) \cdot \begin{pmatrix} \boldsymbol{\Delta \dot{x}_{T^m}} \\ \boldsymbol{\Delta x_{T^m}} \\ \boldsymbol{\Delta u_{T^m}} \\ \boldsymbol{\Delta d_{T^m}} \end{pmatrix} \end{aligned}$$

The Jacobian can the be written in the form:

$$\boldsymbol{J_{f_{T^m}}} = \begin{pmatrix} -\boldsymbol{M} & \tilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}} & \tilde{\boldsymbol{D}} \end{pmatrix}$$

$$\text{with } \tilde{\boldsymbol{A}} = \begin{pmatrix} \frac{\partial f^1_{T^m}}{\partial x_{1_{T^m}}}\left(\boldsymbol{0}, \boldsymbol{x^0_{T^m}}, \boldsymbol{u^0_{T^m}}, \boldsymbol{d^0_{T^m}}\right) & \cdots & \frac{\partial f^1_{T^m}}{\partial x_{n_{T^m}}}\left(\boldsymbol{0}, \boldsymbol{x^0_{T^m}}, \boldsymbol{u^0_{T^m}}, \boldsymbol{d^0_{T^m}}\right) \\ \vdots & & \vdots \\ \frac{\partial f^r_{T^m}}{\partial x_{1_{T^m}}}\left(\boldsymbol{0}, \boldsymbol{x^0_{T^m}}, \boldsymbol{u^0_{T^m}}, \boldsymbol{d^0_{T^m}}\right) & \cdots & \frac{\partial f^r_{T^m}}{\partial x_{n_{T^m}}}\left(\boldsymbol{0}, \boldsymbol{x^0_{T^m}}, \boldsymbol{u^0_{T^m}}, \boldsymbol{d^0_{T^m}}\right) \end{pmatrix}$$

while r denotes the number of equations in $\boldsymbol{f_{T^m}}$ and n the number of states in $\dot{\boldsymbol{x}}_{\boldsymbol{T^m}}$.

All other matrices are calculated accordingly, by partial differentiation with respect to $\dot{\boldsymbol{x}}_{\boldsymbol{T^m}}$, $\boldsymbol{u_{T^m}}$ or $\boldsymbol{d_{T^m}}$.

The system output is given by:

$$\begin{aligned} \boldsymbol{\Delta y_{T^m}} &= \boldsymbol{g_{T^m}}\left(\boldsymbol{x^0_{T^m}} + \boldsymbol{\Delta x_{T^m}}, \boldsymbol{u^0_{T^m}} + \boldsymbol{\Delta u_{T^m}}, \boldsymbol{d^0_{T^m}} + \boldsymbol{\Delta d_{T^m}}\right) - \boldsymbol{g_{T^m}}\left(\boldsymbol{x^0_{T^m}}, \boldsymbol{u^0_{T^m}}, \boldsymbol{d^0_{T^m}}\right) \\ &\approx \boldsymbol{J_{g_{T^m}}}\left(\boldsymbol{x^0_{T^m}}, \boldsymbol{u^0_{T^m}}, \boldsymbol{d^0_{T^m}}\right) \cdot \begin{pmatrix} \boldsymbol{\Delta x_{T^m}} \\ \boldsymbol{\Delta u_{T^m}} \\ \boldsymbol{\Delta d_{T^m}} \end{pmatrix} \end{aligned}$$

In this way, linear state space models

$$\begin{aligned} \boldsymbol{\Delta \dot{x}_{T^m}} &= \boldsymbol{M}^{-1}\tilde{\boldsymbol{A}} \cdot \boldsymbol{\Delta x_{T^m}} + \boldsymbol{M}^{-1}\tilde{\boldsymbol{B}} \cdot \boldsymbol{\Delta u_{T^m}} + \boldsymbol{M}^{-1}\tilde{\boldsymbol{D}} \cdot \boldsymbol{\Delta d_{T^m}} \\ \boldsymbol{\Delta \dot{x}_{T^m}} &= \boldsymbol{A} \cdot \boldsymbol{\Delta x_{T^m}} + \boldsymbol{B} \cdot \boldsymbol{\Delta u_{T^m}} + \boldsymbol{D} \cdot \boldsymbol{\Delta d_{T^m}} \\ \boldsymbol{\Delta y_{T^m}} &= \boldsymbol{C} \cdot \boldsymbol{\Delta x_{T^m}} \end{aligned}$$

can be computed.

A common practice in control engineering is the scaling of the system matrices [10]. It is used to improve the conditioning of the optimisation and calculation by reducing the numerical range. Diagonal transformation matrices are applied to state variables, inputs and outputs.

In this thesis, however, the system matrices have been left unattended, while the inputs and outputs for the nominal system have been scaled, such that they approximately lie in the range of $(-1; 1)$. This facilitates the controller synthesis, since in general it reduces the amount of tuning knobs to be considered. More specifically, temperature variations have been scaled by a factor of $1/20$, such that $20\,\mathrm{K}$ change in temperature account for an output of the linearised system of 1. The mass flow rate fractions β_i, β_{HE} already operate in that range, but the pump speed input is scaled according to its saturation level.

3.3.5. System Equilibria

For the calculation of system equilibria all derivatives of variables with respect to time need to be set to zero. However, care must be taken, because in equilibrium the cooling system is still to be described by stationary flow and heat transfer equations. The respective system is then:

Stationary Equations — Block T^1

$$\begin{aligned} H_P^{*0}\left(\frac{\dot{m}_0^{*0}}{n_P^{*0}}\right) \cdot \frac{H_{R_P} \cdot \varrho_P \cdot g}{n_{R_P}^2} \cdot n_P^{2,0} - R_p^0 \cdot \dot{m}_0^{2,0} &= 0 \\ -\dot{m}_1^0 c_{pP} \cdot T_0^{d0} + \dot{m}_1^0 c_{pP} \cdot T_1^0 - \dot{Q}_{L_1}^0 &= 0 \\ -\dot{m}_2^0 c_{pP} \cdot T_0^{d0} + \dot{m}_2^0 c_{pP} \cdot T_2^0 - \dot{Q}_{L_2}^0 &= 0 \\ -\dot{m}_3^0 c_{pP} \cdot T_0^{d0} + \dot{m}_3^0 c_{pP} \cdot T_3^0 - \dot{Q}_{L_3}^0 &= 0 \end{aligned} \tag{3.3.11}$$

Stationary Equations — Block T^2

$$\begin{aligned} -\dot{m}_{HE}^0 c_{pP} \cdot T_5^0 + \dot{m}_{HE}^0 c_{pP} \cdot T_{HE}^0 + kA(\dot{m}_{HE}^0, \dot{m}_R^0) \cdot QC \cdot (T_{HE}^0 - T_{R_1}^0) &= 0 \\ -\dot{m}_R^0 c_{pA} \cdot T_{R_0}^0 + \dot{m}_R^0 c_{pA} \cdot T_{R_1}^0 - kA(\dot{m}_{HE}^0, \dot{m}_R^0) \cdot QC_R \cdot (T_{HE}^0 - T_{R_1}^0) &= 0 \\ -\dot{m}_b^0 c_{pP} \cdot T_5^0 - \dot{m}_{HE}^0 c_{pP} \cdot T_{HE}^0 + \dot{m}_0^0 c_{pP} \cdot T_0^0 &= 0 \end{aligned} \tag{3.3.12}$$

Still, every equation retains at least bilinear characteristics, such that it is necessary to assert some assumptions:

- In equilibrium, the temperatures $T_i, i = 0, 1, 2, 3$ to be controlled assume the desired reference values as given in section 3.1.
- The environmental conditions, to which the aircraft's cooling system is exposed, are known. This includes temperatures T_{R_0}, T_{R_1} and the ram air mass flow rate $\dot{m}_R$.

The non-linear *MATLAB/Simulink* simulation may also greatly help in finding equilibrium positions.

In the course of controller design the most important equilibrium — when all loads are applied — will be incorporated first, in order to evaluate the feasibility with regard to different operating points. As has been mentioned in section 2.5.1, a *fail-safe* controller should stabilise the system regardless of the load configuration. The main focus on the controller design for the next section will therefore be the synthesis of a single controller for this equilibrium.

3.4. LQG Controller Synthesis

According to MACKENROTH [8, p.13] for some plants PI-controllers may be employed as a rapid tool for the assessment of possible ideal performance, as well as performance restrictions. MACKENROTH also states, that PI-controllers are often very sensitive to certain perturbations, which lessens their practical relevance. In the previous section, this has been verified with respect to the cooling plant: While the PI-controller configuration may achieve extremely good performance under the assumption of ideal measurements, the performance degrades to merely acceptable performance in the presence of measurement noise. This section will examine the linear quadratic gaussian (LQG) controller's ability to improve the performance even in the presence of perturbed system outputs. Theoretical background on LQG control is given in appendix C.2. Please note, that in this thesis LQG control is presented in the form of an $\mathcal{H}_2$ norm optimisation problem with the help of the concept of a generalised plant. This enables for the employment of uncertain plant models, for example, if some plant parameters are uncertain, without change of paradigm. This will be briefly covered in section 3.6.

LQG controllers take advantage of a mathematical model to ensure optimality in the sense of a certain cost functional. The underlying controller structure is that of *observer based state feedback*, which is well established. The following section will describe the preliminary analysis of the linearised plant as a requirement of the controller design. After that, the actual design will be explained and evaluated. The controller is designed for the equilibrium, where all loads are applied to the system and the airplane is at cruise speed and altitude. It will be investigated to which extent the controller satisfies the design objectives even under conditions, that are far from the equilibrium point around which the plant model is linearised.

3.4.1. Analysis of the Linearised Plant Model

For LQG control to be applicable to the linearised plant models, definitions C.2.1 and C.2.2 quoted in the control theory addendum in section C.2 must hold true.

Stabilisability

More specifically, for a system to be stabilisable, the controllable states have to be identified first. This is done by construction of the controllability matrices for the plant models of both blocks:

$$\mathcal{C}(\boldsymbol{A_{T^m}}, \boldsymbol{B_{T^m}}) = \begin{pmatrix} \boldsymbol{B_{T^m}} & \boldsymbol{A_{T^m}} \boldsymbol{B_{T^m}} & \boldsymbol{A_{T^m}}^2 \boldsymbol{B_{T^m}} & \dots & \boldsymbol{A_{T^m}}^{n-1} \boldsymbol{B_{T^m}} \end{pmatrix} \tag{3.4.1}$$

with $m = 1, 2$ denoting the respective block
and n denoting the respective block's system order.

It turns out, that the controllability matrix of block 1, $\mathcal{C}(\boldsymbol{A_{T^1}}, \boldsymbol{B_{T^1}})$, has full row rank $n = 7$. Therefore the system is stabilisable by state feedback. The controllability matrix of block 2, $\mathcal{C}(\boldsymbol{A_{T^2}}, \boldsymbol{B_{T^2}})$, however, looses rank by one. In order for this system to be stabilisable nonetheless, it has to be verified, that the uncontrollable eigenvalue is stable, i.e. it is in the left half plane.

One way to do this, is to compute the *controllability staircase form*. Since controllability is invariant under similarity transformations [23], define a transformation matrix

$$\boldsymbol{T^c_{T^2}} = \begin{pmatrix} \boldsymbol{B_{T^2}} & \boldsymbol{A_{T^2}}\boldsymbol{B_{T^2}} & \boldsymbol{A_{T^2}}^2\boldsymbol{B_{T^2}} & \dots & \boldsymbol{A_{T^2}}^{r-1}\boldsymbol{B_{T^m}} & \boldsymbol{q}^{r+1} & \dots & \boldsymbol{q}^n \end{pmatrix} \tag{3.4.2}$$

with r denoting the row rank of $\mathcal{C}(\boldsymbol{A_{T^2}}, \boldsymbol{B_{T^2}})$

and $\boldsymbol{q}^{r+1}\dots\ \boldsymbol{q}^n$ vectors to fill up the matrix, such that it remains regular.

The result is the controllability staircase form, from which controllable (^{c}) and uncontrollable ($^{\bar{c}}$) modes of the system matrix can be determined.

$$\boldsymbol{T^c_{T^2}}^{-1}\boldsymbol{A_{T^2}}\boldsymbol{T^c_{T^2}} = \boldsymbol{\bar{A}_{T^2}} = \begin{pmatrix} \boldsymbol{\bar{A}^c_{T^2}} & \boldsymbol{\bar{A}^{12}_{T^2}} \\ \boldsymbol{0} & \boldsymbol{\bar{A}^{\bar{c}}_{T^2}} \end{pmatrix}. \tag{3.4.3}$$

From this, it can be seen, that the uncontrollable mode of $\boldsymbol{\bar{A}^{\bar{c}}_{T^2}}$ is stable, such that block 2 is also stabilisable.

Detectability

The dual concept of detectability can be verified in a very analogue way: First, the observability matrices have to be constructed:

$$\mathcal{O}(\boldsymbol{C_{T^m}}, \boldsymbol{A_{T^m}}) = \begin{pmatrix} \boldsymbol{C_{T^m}} \\ \boldsymbol{C_{T^m}}\boldsymbol{A_{T^m}} \\ \boldsymbol{C_{T^m}}\boldsymbol{A_{T^m}}^2 \\ \vdots \\ \boldsymbol{C_{T^m}}\boldsymbol{A_{T^m}}^{n-1} \end{pmatrix} \tag{3.4.4}$$

with $m = 1, 2$ denoting the respective block

and n denoting the respective block's system order.

As has been anticipated for both blocks, the observability matrices have exactly the same column rank as the respective block's number of outputs. Therefore, it has to be investigated, whether the unobservable states are stable in the sense, that the estimation error vanishes for $t \to \infty$. A similarity transformation

$$\boldsymbol{T^o_{T^m}}^{-1} = \begin{pmatrix} \boldsymbol{C_{T^m}}^T & \boldsymbol{A_{T^m}}^T\boldsymbol{C_{T^m}}^T & \dots & \boldsymbol{A_{T^m}}^{r-1\ T}\boldsymbol{C_{T^m}}^T & \boldsymbol{q}^{r+1\ T} & \dots & \boldsymbol{q}^{n\ T} \end{pmatrix}^T \tag{3.4.5}$$

with r denoting the column rank of $\mathcal{O}(\boldsymbol{C_{T^m}}, \boldsymbol{A_{T^m}})$

and $\boldsymbol{q}^{r+1\ T}\dots\ \boldsymbol{q}^{n\ T}$ vectors to fill up the matrix, such that it remains regular.

can be constructed, which leads to the *observability staircase form*

$$\boldsymbol{T^o_{T^m}}^{-1}\boldsymbol{A_{T^m}}\boldsymbol{T^o_{T^m}} = \boldsymbol{\bar{A}_{T^m}} = \begin{pmatrix} \boldsymbol{\bar{A}^o_{T^m}} & \boldsymbol{0} \\ \boldsymbol{\bar{A}^{21}_{T^m}} & \boldsymbol{\bar{A}^{\bar{o}}_{T^m}} \end{pmatrix}. \tag{3.4.6}$$

Both systems are detectable, because the unobservable modes are stable.

3.4.2. Decentralised LQG Controller Tuning

The tuning of the decentralised LQG controller setup has been done directly with the non-linear simulation, after some initial simulations of the nominal plants.

Initially, the LQG weighting matrices where chosen as:

$$\begin{aligned}
&\boldsymbol{Q_{T^m}} = \gamma^m \cdot \boldsymbol{C_{T^m}}^T\boldsymbol{C_{T^m}} \\
&\boldsymbol{R_{T^m}} = \rho^m \cdot \boldsymbol{I} \\
&\boldsymbol{Q^e_{T^m}} = \gamma^m_e \cdot \boldsymbol{B_{T^m}}\boldsymbol{B_{T^m}}^T \\
&\boldsymbol{R^e_{T^m}} = \rho^m_e \cdot \boldsymbol{I} \\
&\text{with } \gamma^m, \rho^m, \gamma^m_e, \rho^m_e = 1, \qquad\qquad m = 1, 2.
\end{aligned}$$

In the presence of measurement noise, this resulted in heavily oscillating control outputs, which could be alleviated mainly by increasing ρ_e, which leads to a lower observer bandwidth and more filtering of output noise. Faster control has been achieved by further increasing the weight on $\boldsymbol{Q_{T^m}}$, thus raising γ. The final tuning parameters are listed in table 3.4.2. Note, that these parameters do not pretend to be the optimal tuning of the LQG control for the cooling cycle. There is room for improvements left, but the results show, that it is possible to achieve satisfying control.

Tuning Parameter	Value	Tuning Parameter	Value
γ^1	$2 \cdot 10^2$	γ^2	$2 \cdot 10^2$
ρ^1	1	ρ^2	1
γ^1_e	1	γ^2_e	1
ρ^1_e	$6 \cdot 10^2$	ρ^2_e	$1 \cdot 10^3$

Table 3.4.1.: LQG Controller Tuning Parameters

3.4.3. Discussion

The LQG controller structure is capable of meeting the requested design objectives in a satisfactory manner. It even controls for stable equilibria far from the one specified in the linearised plant model. Thus, with regard to the non-linear simulation, the LQG controller can be said to be robustly stable.

Around operating points different from the one specified in the plant model used for controller synthesis, the reference tracking capabilities deteriorate. They are acceptable, in that they are always within sufficiently narrow bounds. Indeed, the maximum temperature never even exceeds 55 °C in the first test setup and it only almost touches 60 °C during the climb simulated in test setup 2. With respect to the given objectives of sufficient reference tracking, maximum overshoot and disturbance rejection, considering the non-linear simulation, the controller can therefore be said to also provide robust performance.

With regard to energy optimality, however, the LQG controller based on a single operating point reveals a major drawback, though. Since control effort is being penalised with respect to the operating point, the controller also tries to avoid the reduction of pump speed lower than the specified equilibrium. During the phase, when only loads 1 and 2 are applied, the pump speed is remaining at a level of $1000\,\mathrm{min}^{-1}$ and even goes a little higher, while the heat exchanger bypass control valve could easily be opened to a greater extent to provide the cooling. It becomes clear, that more rigorous performance criteria than those specified in section 3.1 are only met near the linearisation point. Lowering the penalty for the control effort mainly introduces more undesired noise to the controller output, while only slightly alleviating the energy consumption.

The influence of measurement noise to the control output can be abated quite well, making use of the equivalence of the KALMAN filter and the LUENBERGER observer structure. It is possible to achieve oscillation levels in the controller outputs, that can be deemed harmless to mechanical wear on the valves without the use of additional filters on the measurement signals.

The non-linear deterministic transformation from mass flow fractions to valve opening ratios works well enough, though the upper and lower saturation limits do not match the "real" saturation limits of 0 and 1. This is due to the parameters of maximum and minimum resistance coefficients $K_{v_{max}}$ and $K_{v_{min}}$, which can be more properly defined.

Summary

Multivariable control and optimisation with respect to a linear quadratic cost function works well in the vicinity of a specified operating point. Beyond that, efficiency deteriorates, when the controller operates farther from the linearisation point. The observer structure is needed, since not all states of the plant can be measured. At the same time, it can be exploited as a KALMAN filter. The tuning of the LQG controller is again a task based more or less on heuristics. This is especially true, if the spectral densities of the noise inputs are unknown, which are often used as weightings. The observer structure heavily influences the robustness properties of state feedback [22] and caution is needed, because an inappropriately tuned observer may render the closed-loop unstable.

Since the controller simulates the linearised plant in real-time, the controller order is the same as that of the plant used for the controller synthesis. Due to this, the controller

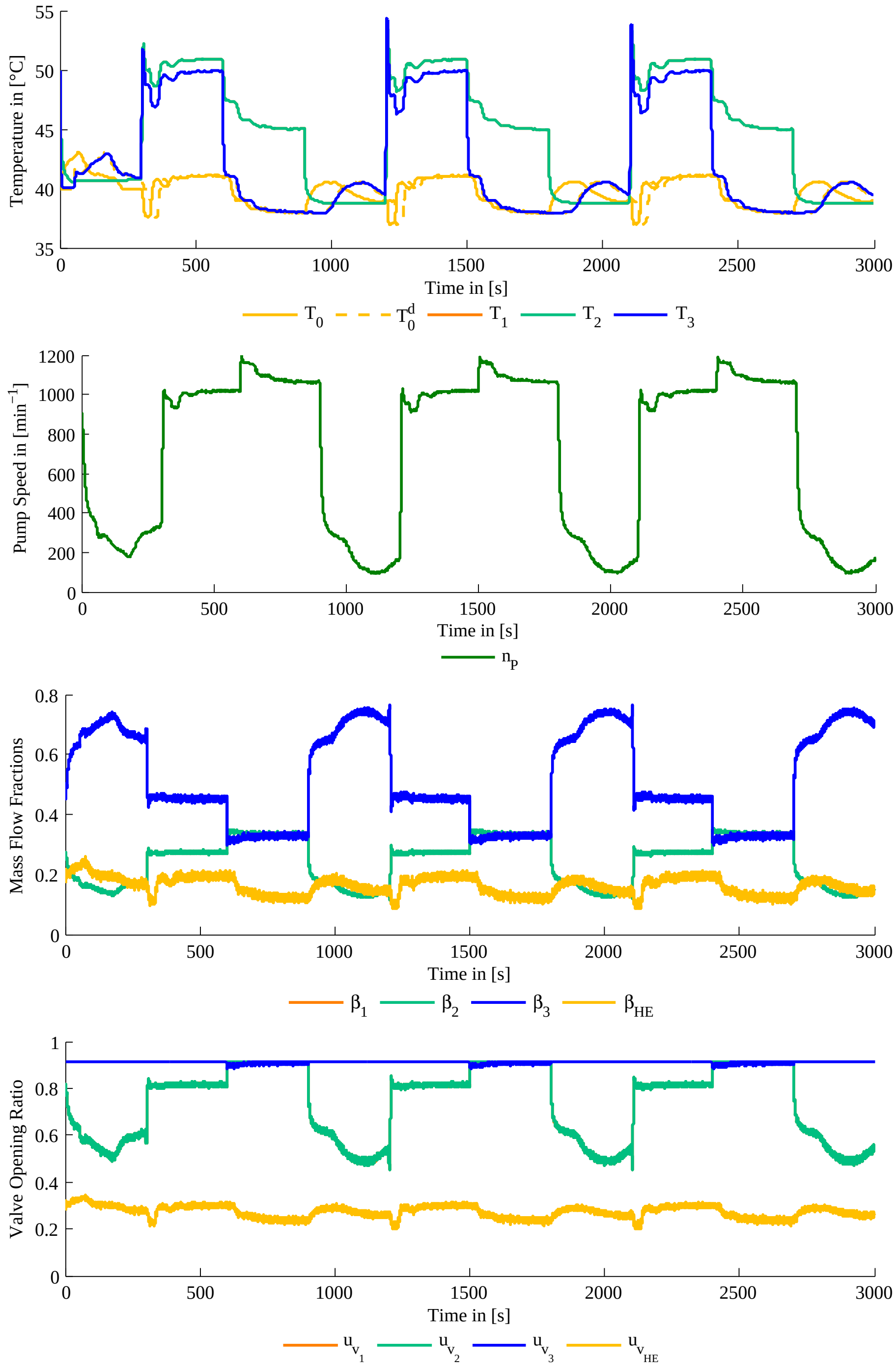


Figure 3.4.1.: LQG Controller Setup: Simulation Results — Benchmark Setup 1

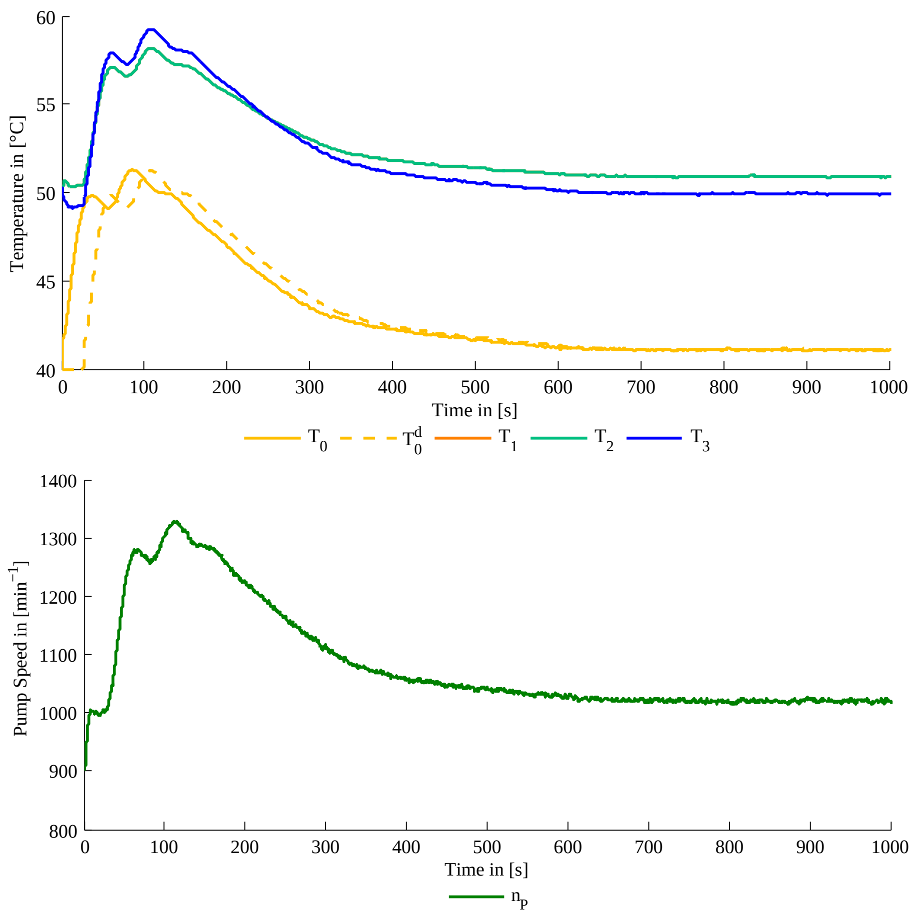


Figure 3.4.2.: LQG Controller Setup: Simulation Results — Benchmark Setup 2 (1)

order adds up to $7 + 4 = 11$. Modern hardware should have no problems to handle this amount of complexity.

The energy consumption still leaves room for optimisation, but the overall controller design is appropriate for practical use, because it provides good closed-loop performance and stability with respect to safety critical design objectives. Stability can be regarded robust, in terms of the non-linear simulation of benchmark setups, which reflect strongly exaggerated demands. Since the dependence on the linearisation point is responsible for the bad degree of energy optimality, a scheduling algorithm can be incorporated to dynamically select a respective LQG controller dedicated to a specific equilibrium to ensure a higher degree of optimality also in the vicinity of the other equilibria. In case of unforeseen loads, the scheduling algorithm can automatically return to the single LQG controller as designed in this section. The next section will briefly consider a basic design of a gain-scheduled LQG controller, in order to investigate the benefits.

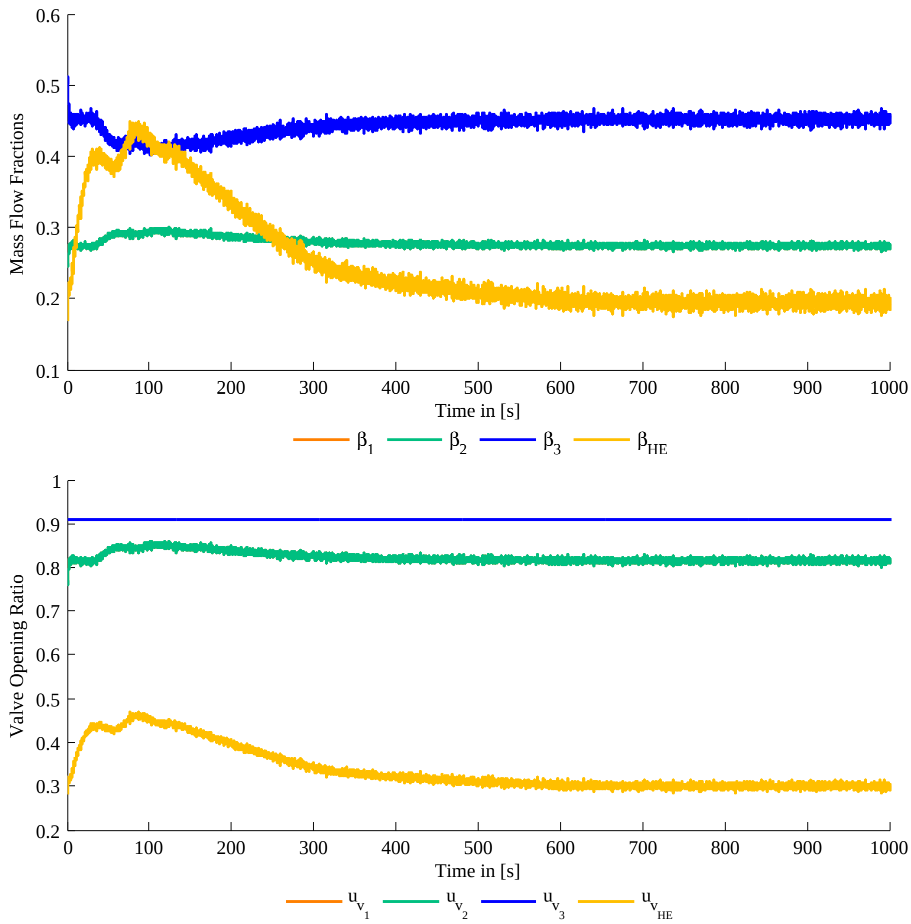


Figure 3.4.3.: LQG Controller Setup: Simulation Results — Benchmark Setup 2 (2)

3.5. Gain Scheduled LQG Controller Synthesis

The previous section has showed, that the LQG controller is basically applicable to the non-linear cooling cycle plant, since a linearised plant model has been found on which the controller synthesis can rely. It has also illustrated, that the LQG controller's capability of guaranteeing optimality with respect to a cost functional V is restricted to a certain interval around the linearisation point.

So far the only operating point considered had all loads applied to the system. A common idea in control engineering is the concept of *gain scheduling*: Multiple controllers are designed for multiple equilibria, while a set of time-varying scheduling parameters determines, which controller is active. Figure 3.5.1 illustrates a very general scheme for gain scheduled control. The plant P is providing a set of scheduling signals $\boldsymbol{p}$ for the controller, which adapts accordingly. For instance, a prepended logic simply chooses from a set of precalculated controllers.

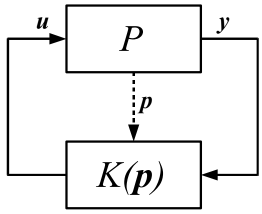


Figure 3.5.1.: General Gain Scheduling Control Loop

This idea has been adopted to the cooling cycle in a simple way, such that the improvement in performance can roughly be estimated.

3.5.1. Scheduling Parameters, Switching Algorithms and Design Issues

To illustrate the effectiveness, only *benchmark setup 1* will be considered in this section. This is due to the fact, that the scheduling parameters, which are best suited to switch between controllers, are the loads $Q_{L_i}, i = 1, 2, 3$, because they determine the required mass flow rates for the cooling. Care should be taken, when pondering other possible scheduling signals, though. [5] points out, that the scheduling algorithm and the individual controllers may work against each other, if the scheduling parameter is subject to the same dynamics as the plant dynamics. Therefore external scheduling signals should be taken into account preferably.

In theory — and under the assumption, that the loads are only either turned on or off — there exist nine combinations of loading conditions. For this thesis only the three relevant cases have been considered for benchmarking as provided in table 3.5.1.

Case	Q_{L_1} **in** [W]	Q_{L_2} **in** [W]	Q_{L_3} **in** [W]
1	1000	1000	2000
2	1000	1000	0
3	0	0	0

Table 3.5.1.: LQG Gain Scheduling Loading Conditions

It is assumed, that the information, whether the loads are active or not, is readily available.

The switching algorithm has been chosen to be as simple as possible: No interpolation between the loading cases is done. For fast mechanical systems, like, for instance, an inverted pendulum, so called *"hard switching"* is a serious issue, which may destabilise the whole system. In the case of the cooling cycle all equilibria are asymptotically stable, such that this is no issue. Figure 3.5.2 illustrates the selection of a particular controller K_i based on the scheduling parameter $\boldsymbol{p}$, which in this case could simply be a binary code, from which the index of the controller to be activated can be extracted.

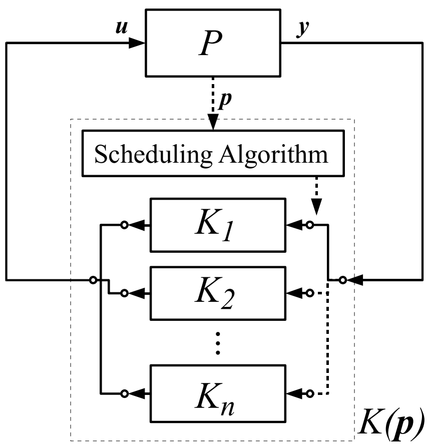


Figure 3.5.2.: Hard Switching Gain Scheduling Control Loop

However, a common problem with gain scheduling is the possible loss of controllability in certain linearisation points. This is also true with the system given, since for all loads turned off, zero mass flow is required. It has been chosen, to go for zero plant inputs in this case. of course, no regulation is possible this way and it has to be guaranteed, that the coolant's temperature is at a moderate level, such that it may remain there.

3.5.2. Discussion

The main purpose to introduce a scheduling algorithm was to improve the energy efficiency in all operating points, that are known a priori. It can be seen from the plots 3.5.3, that the full range of plant inputs is now used for more efficient control.

Additionally, the reference tracking has been vastly improved and the over– and undershoot peaks have been removed almost entirely.

Summary

The gain-scheduling algorithm is in fact a simple steering overhead. The pump and valve control inputs are set to known equilibrium values, while it is the controller's task to alter these values regulating for a quick transition from one equilibrium to another. While this significantly improves the system's performance under nominal conditions, i.e. under conditions, that match the a priori knowledge of applied load magnitudes, it is a matter to investigate, how the system will behave under unforeseen conditions.

The scheduling algorithm would either have to switch to a robust controller, that can safely operate under any condition, while making amends with regard to energy optimality, or the robustness of each controller K_i on the whole range of operating points would have to be guaranteed. With the current scheduling configuration, the latter is impossible to achieve, since a controller in the equilibrium of zero inputs looses controllability completely. The design of a robust "backup"-controller seems to be a feasible solution, though. Section 3.4 showed, that, in essence, this is very well possible. On the other hand, it might also be a promising strategy to always choose a controller, that is closest to the true operating point. This could be done, if an array of different controllers for various nominal and failure-case operating points could be synthesised, while reasonable hardware requirements and constraints can still be met. For operating points between two equilibria, for which controllers have been designed for, the outputs of both controllers could be smoothly blended. A fuzzy controller would be a good choice to operate on the steering level.

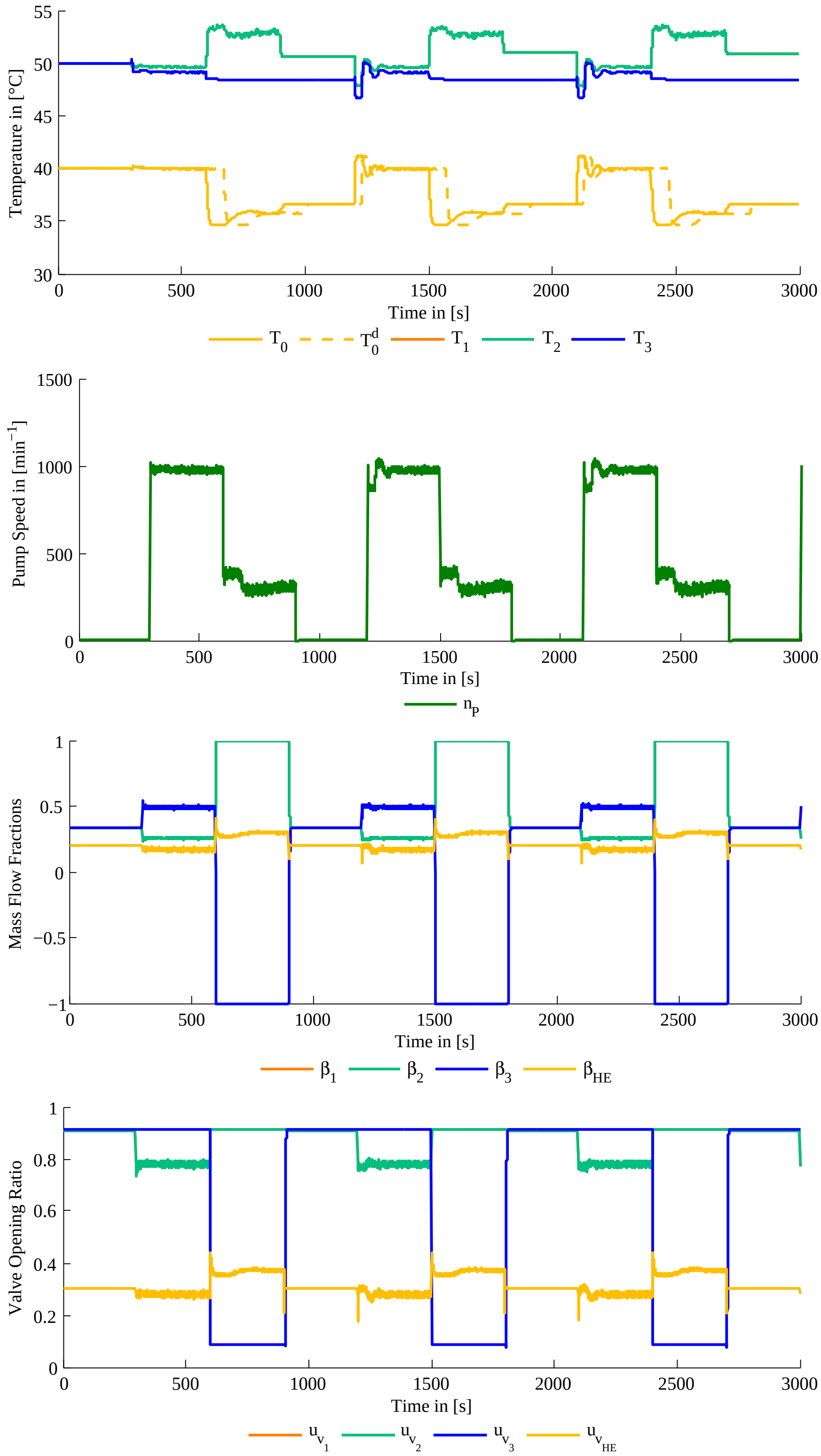


Figure 3.5.3.: LQG Gain Scheduling Controller Setup: Simulation Results

3.6. Robust $\mathcal{H}_\infty$ Controller Synthesis

So far, the robustness of the LQG controller has only been show with regard to the non-linear simulation. The non-linear *Simulink* simulation, however, is in principle governed by the same equations as the linearised plant, provided the plant is operating in the vicinity of the linearisation point. But for some deviations from that operating point or for some deviations in the plant dynamics the robustness can also be proven analytically. A tool, frequently used in modern control, is $\mathcal{H}_\infty$ norm based controller design with the help of the *small gain theorem*.

This section will provide a brief overview of a possible way to design a robust LQG controller presented in [22, p.150] and a short discussion to which extent this may be useful to apply to the cooling cycle.

3.6.1. The $\mathcal{H}_\infty$ Norm, Model Uncertainty and the Small Gain Theorem

The $\mathcal{H}_\infty$ norm is the maximum singular value, i.e. the maximum gain for multivariable systems, over all frequencies:

$$\|\boldsymbol{G}(s)\|_\infty = \sup_{w} \bar{\sigma}\left(\boldsymbol{G}(j\omega)\right). \tag{3.6.1}$$

In combination with the small gain theorem, this can be utilised to express a required constraint on the generalised plant with modelled uncertainty.

The Small Gain Theorem

If $L(s)$ is stable, the closed-loop (figure 3.6.1) is stable if $\|L(s)\| < 1$ for all ω.

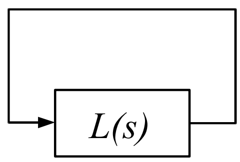


Figure 3.6.1.: The Small Gain Theorem

In essence, the *small gain theorem* states, that a closed-loop system is stable, if the Nyquist-plot remains inside the unit disc for all frequencies, thus avoiding encirclement of the critical point -1. Strictly, this is only true for SISO systems, while for MIMO systems $\|\boldsymbol{L}(s)\| < 1$ implies $\|\boldsymbol{L}(s)\|_\infty < 1$.

The small gain theorem poses a safe but restrictive condition, since in the case of SISO systems for stability, the Nyquist-plot may very well leave the unit disc as long as it does not encircle the -1.

Representing Parametric Model Uncertainty

A linear model of a plant may have the form:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}^i \boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \\ \text{where, } \boldsymbol{A}^i &= \boldsymbol{A}^0 + \boldsymbol{B}_{\boldsymbol{w}}^{\infty} \boldsymbol{\Delta} \boldsymbol{C}_{\boldsymbol{z}}^{\infty} \end{aligned} \tag{3.6.2}$$

The system matrix $\boldsymbol{A}^i$ contains parametric uncertainties and can be decomposed into a nominal matrix $\boldsymbol{A}^0$ and a weighted uncertain matrix $\boldsymbol{\Delta}$ with $\|\boldsymbol{\Delta}\| < 1$. The decomposition of the uncertain part of $\boldsymbol{A}^i \in \mathbb{R}^{n \times n}$ into $\boldsymbol{B}_{\boldsymbol{w}}^{\infty} \boldsymbol{\Delta} \boldsymbol{C}_{\boldsymbol{z}}^{\infty}$ can be done with the help of the *singular value decomposition*, where the uncertainty can be mapped to a multidimensional space of dimension $\leq n$. This procedure requires to create an array of different samples of system matrices, in order to cover all possible numerical combinations of the uncertain parameters. The amount of samples needed increases fast with the number of uncertain parameters and there is no strict rule how many samples to take. For k uncertain parameters and l samples for each, a total number of l^k samples is needed. With *MATLAB* on a normal computer this quickly leads to memory shortage.

Figure 3.6.2 illustrates the extension of the generalised plant, where the uncertainty can be understood as an uncertain feedback loop. The mathematical representation of the uncertainty with respect to the generalised plant is called an upper linear fractional transformation (upper LFT). A thorough theoretical background will be omitted here for brevity. In this way, the fictitious input $\boldsymbol{w}_\infty$ and output $\boldsymbol{z}_\infty$ have been added to

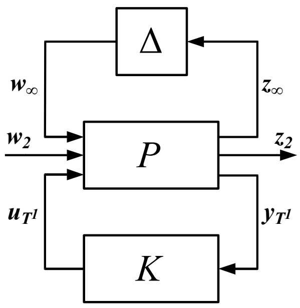
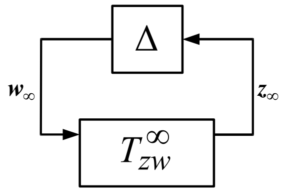


Figure 3.6.2.: Generalised Plant with Uncertainty

the system. Since the uncertain part of $\boldsymbol{A}^i$ has been decomposed, such that $\|\boldsymbol{\Delta}\| < 1$, the small gain theorem now yields a specific requirement for the closed-loop system to assure robust stability:

$$\|\boldsymbol{\Delta} \boldsymbol{T}_{\boldsymbol{zw}}^{\infty}(j\omega)\| = \|\boldsymbol{\Delta}\| \|\boldsymbol{T}_{\boldsymbol{zw}}^{\infty}(j\omega)\| < 1 \iff \|\boldsymbol{T}_{\boldsymbol{zw}}^{\infty}(j\omega)\|_\infty < 1 \tag{3.6.3}$$

Please note, that the closed-loop system shown in figure 3.6.3 (l.) includes a controller K, that stabilises the nominal plant.



$$\boldsymbol{P}(s) = \left[\begin{array}{c|c|c} \boldsymbol{A}^0 & \begin{bmatrix}\boldsymbol{B}^\infty_{\boldsymbol{w}} & \boldsymbol{B}^{\boldsymbol{2}}_{\boldsymbol{w}}\end{bmatrix} & \boldsymbol{B} \\ \hline \begin{bmatrix}\boldsymbol{C}^\infty_{\boldsymbol{z}} \\ \boldsymbol{C}^{\boldsymbol{2}}_{\boldsymbol{z}}\end{bmatrix} & \boldsymbol{0} & \begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{D}^{\boldsymbol{2}}_{\boldsymbol{zu}}\end{bmatrix} \\ -\boldsymbol{C} & \begin{bmatrix}\boldsymbol{0} & \boldsymbol{D}^{\boldsymbol{2}}_{\boldsymbol{vw}}\end{bmatrix} & \boldsymbol{0} \end{array}\right]$$

Figure 3.6.3.: Robust Stability Closed-Loop and Generalised Plant

3.6.2. Uncertain Parameters

The approach summarised in the previous section is applicable, if the system matrix contains uncertain parameters. If these uncertain parameters are included in the input and output gain matrices $\boldsymbol{B}, \boldsymbol{C}$, a pre– or postfilter is needed, respectively, in order to move the uncertainties to the system matrix. [18, p.105] provides a systematic approach to pre-/postfiltering in the appendix.

The uncertain parameters should be understood to reflect uncertain linearisation points or parameters, that can be accurately measured only within some interval. They have to be clearly distinguished from time-varying parameters, whose presence means, that a plant behaves in a non-linear way. In fact, almost every parameter of the cooling cycle in the state matrices of the linearised blocks T^1 and T^2 is a time-varying parameter: The temperatures T_i, mass flow rate fractions β_i, β_{HE}, overall hydraulic resistance R'_p and fluid inertance L'_p, etc., all change with varying plant inputs and changing environmental conditions. It is only near certain equilibria, that they can be assumed constant. Regarding these as uncertain parameters suggests, that the exact values of a stable equilibrium are only known within some uncertainty bound.

This notion can be useful for the cooling cycle plant in numerous ways, as will be discussed in the following:

Inaccurate Measurements of Hydraulic Resistance R'_p, Fluid Inertance L'_p, Pump Head H and Heat Exchanger Characteristics QC

In this thesis, the equilibrium values of the aforementioned parameters have all been accurately determined from the non-linear simulation and equations. In a practical application of a model-based controller these coefficients may be very difficult to determine exactly for the real plant. To provide an analytical proof, that a given controller is able to stabilise the real plant even under circumstances that differ from the precalculated plant model, an uncertainty interval could be imposed on these plant parameters.

Uncertain Ram Air Mass Flow Rate $\dot{m}_R$ and Temperature T_{R_0}

During all previous simulations, the ram air channel mass flow rate $\dot{m}_R$ has been assumed constant due to a high priority control loop, that keeps the aircraft's flow resistance low. Should this assumption be violated in practice, the cooling cycle control could be required to be robustly stable against mass flow rates differing from the reference value. The same applies to the ram air temperature T_{R_0}, which will surely vary, depending on flight height or atmosphere in general. However, to demand robustness against varying environmental conditions would mean to demand robustness against a complete shift of the equilibrium position — at least with respect to block T^2. To ensure a sufficient reference tracking of the heat exchanger bypass outlet temperature T_0, the equilibrium mass flow rate fraction β_{HE} has to be expressed in terms of the uncertainty of T_{R_0} and $\dot{m}_R$:

$$\begin{aligned}
\dot{m}_R^\delta &= \dot{m}_R^0(1+\delta_{\dot{m}_R}) \\
T_{R_0}^\delta &= T_{R_0}^0(1+\delta_{T_R}) \\
T_{R_1}^\delta &= T_{R_1}^0(1+\delta_{T_R}),
\end{aligned}$$

$$\text{where } \delta_{\dot{m}_R}, \delta_{T_R} \text{ reflect the relative uncertainty}$$
$$\text{and } T_{R_0} \text{ and } T_{R_1} \text{ are assumed to vary in the same way.}$$

The stationary equations 3.3.12 of block T^2 yield an expression for the uncertain equilibrium values of the mass flow rate fraction β_{HE}^δ and the heat exchanger temperature T_{HE}^δ depending on $\delta_{\dot{m}_R}$ and δ_{T_R}:

$$\begin{aligned}
T_{HE}^\delta &= \left(T_{R_1}^0 + \dot{m}_R^0(1+\delta_{\dot{m}_R})c_{pA}(T_{R_1}^0 - T_{R_0}^0)\right)(1+\delta_{T_R}) \\
\beta_{HE}^\delta &= \frac{T_0^0 - T_5^0}{T_{HE}^\delta - T_5^0}
\end{aligned}$$

As an underlying assumption to this derivation, all equilibrium Temperatures and mass flow rates of block T^1 remain unchanged as well as the heat exchanger characteristic value kA.

Uncertain Temperatures T_0 and T_5 due to Deteriorated Reference Tracking

The simulations have shown, that the reference tracking capabilities deteriorate, if the controllers operate on a stable equilibrium that is slightly detached from the design model. Since a decentralised controller structure has been chosen under the assumption, that both controllers provide perfect reference tracking, robustness against stable steady state errors can be demanded.

$$\begin{aligned}
T_0^\delta &= T_0^{d0}(1+\delta_{T_0}) \\
T_5^\delta &= T_5^0(1+\delta_{T_5})
\end{aligned}$$

The stationary equations 3.3.11 of block T^1 and the equations 3.3.12 of block T^2 yield expressions, that reveal the propagation of the uncertainty to the other coefficients:

$$\beta_{HE}^\delta = \frac{T_0^0(1+\delta_{T_0}) - T_5^0(1+\delta_{T_5})}{T_{HE}^0 - T_5^0(1+\delta_{T_5})}$$
$$\beta_i^\delta = \frac{\dot{Q}_{L_i}/c_{pP}}{\dot{m}_0^\delta}$$
$$\dot{m}_0^\delta = \frac{\dot{Q}_{L_1}}{T_1^0 - T_0^{d0}(1+\delta_{T_0})} + \frac{\dot{Q}_{L_2}}{T_2^0 - T_0^{d0}(1+\delta_{T_0})} + \frac{\dot{Q}_{L_3}}{T_3^0 - T_0^{d0}(1+\delta_{T_0})}$$

For simplicity, the propagation to pump head H^δ and pump speed n_p^δ will be omitted here. It should be also noted, that regarding T_0 and T_5 as uncertain instead of time-varying implies, that the transient changes of both temperatures may still introduce unstable or at least unforeseen behaviour. The robustness can only be guaranteed for stable values or — as could be argued — if the control is fast compared to changes in T_0 and T_5.

3.6.3. Application Issues

A combination of the proposed ways of introducing parametric uncertainties to the model is, of course, possible. Trying to analytically trace in which way the uncertainty propagates to other parameters helps to keep the number of independent uncertainties and therefore the memory usage low. Additionally, unrelated uncertain parameters may lead to unrealistic combinations of numerical values, which introduces conservatism to the control loop. Therefore it is always a good idea, to relate the uncertain parameters when possible.

With all this kept in mind, there still is no guarantee, that a controller exists, which robustly stabilises a given uncertain plant in the sense of the small gain theorem. Robustness is a desirable property, but usually the benefit of widening the applicability to more operating points appears along with a degradation of nominal performance. With respect to robustness against uncertain environmental conditions, for example, it was only possible to synthesise a robust controller, which in turn showed infeasible oscillations with respect to the control output $u_{v_{HE}}$. In these cases, alternative synthesis methods might provide more suitable results.

As an example, the Nyquist plot of the uncertain open loop transfer function

$$L_{T^2}(s) = G_{T^2} K_{T^2}(s)$$

suggests, but does not provide a proof, that the LQG controller is already stable against perturbations occurring in T_{R_0} and $\dot{m}_R$. Figure 3.6.4 shows Nyquist plots of randomly chosen samples of the uncertain transfer function $L_{T^2}(s)$.

$T_{R_0}^\delta$ has been defined with a relative uncertainty ranging from 40 to 120 %, while $\dot{m}_R^\delta$ may vary between 0.325 and 0.375 kg/ s. Clearly, the Nyquist plot leaves the unit disc

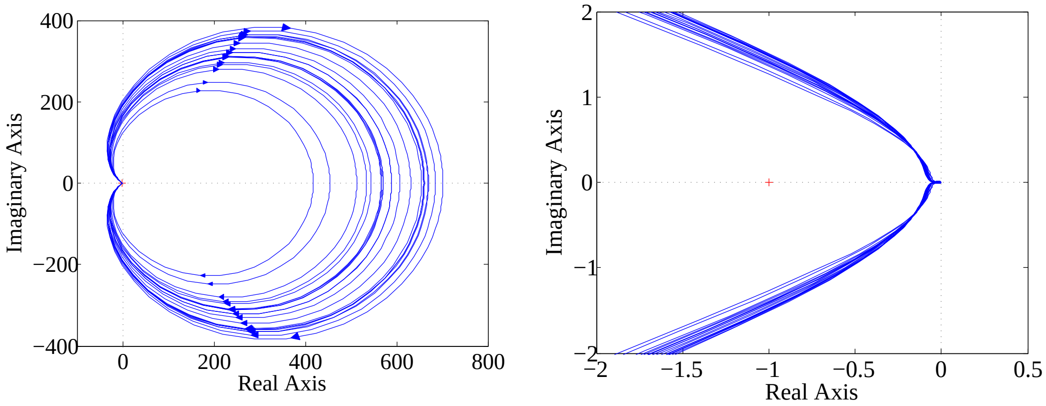


Figure 3.6.4.: Random Nyquist Plots of Uncertain Open Loop Transfer Function $L_{T^2}(s)$ with Close Up

by far, explaining why the $\mathcal{H}_\infty$ controller synthesis fails. However, it can be assumed, that the curve always closes to the right hand side, never coming even close to encircling the -1. An analytical proof can be easily done using the NYQUIST theorem, since block T^2 is a SISO system. It can also be calculated, that the gain margin varies between 20.5 and 37.1dB, while the phase margin varies between 56.9° and 63.8°. Which proves, that the LQG controller is robust against the specified changes in environmental conditions with respect to $\dot{m}_R$ and T_{R_0}.

[16] gives a brief overview over the capabilities and features of *MATLAB's Robust Control Toolbox* to define uncertain plant models.

3.7. Summary

The following table provides an overview over the different results obtained.

Aspect	**Heuristically Tuned PI Controller**	**LQG Controller**	**LQG Gain Scheduling Controller**
Tuning Procedure	• By heuristic measures: manual or automated tuning, controller structure tuning.	• Manual tuning of weightings. • Automated optimisation in terms of linear quadratic cost function.	• Each single controller tuned like normal LQG. • Linearisation point resolution and scheduling algorithm.
Controller Complexity	• Low.	• Same as linearised plant model.	• High: Multiple controllers. • Additional computations for scheduling algorithm.
Disturbance Rejection	• Acceptable: Moderate damping results in high peaks. • Controller operates independently from different equilibria. • Oscillations quickly fade.	• Good: Minor peaks and sufficient operation on different equilibria. • Slow oscillations.	• Dedicated controllers for different equilibria. • Scheduling algorithm's behaviour towards disturbances and failures still to be assessed.
Reference Tracking	• Steady state temperatures never exceed given limits. • Considerable undershoot.	• Steady state temperatures never exceed given limits. • Considerable undershoot.	• Steady state temperatures never exceed given limits. • Reduced undershoot.

Aspect	Heuristically Tuned PI Controller	LQG Controller	LQG Gain Scheduling Controller
Energy Optimality	• Operation close to minimum energy consumption.	• No optimality in equilibrium different from the linearisation operating point. • Control effort penalty is generally defined relative to the linearisation operating point: Linearisation has to be defined in optimal operating point.	• Varying linearisation offsets of control inputs for each controller ensure higher efficiency.
Noise Rejection	• Tradeoff between fast control (high proportional gain) and suppression of measurement noise. • Filtering negatively affects closed-loop performance.	• Built in KALMAN filtering. • Tradeoff between fast control and filtering: Low observer gain needed.	• Application of controllers only near design operating point leads to more effective regulation: filtering could be increased further.

Table 3.7.1.: Summary and Comparison of Simulation Results

From the table it can be observed, that LQG control has considerable advantages over PI control. LQG gain scheduling is necessary, though, to ensure an increase in efficiency with regard to power consumption, which is vital for a successful control strategy in the sense of the MOET research program. This, however, demands for a relatively large family of tuned LQG controllers to cover the full spectrum of operating points. From the non-linear simulation, it can be inferred, that both single controller types are able to ensure safe operation and are thus feasible for providing backup control in case of scheduling signal sensor failures. Bad noise rejection of the PI configuration is limiting, but can be alleviated with output filters.

Once a linearised plant model is available, the LQG controller synthesis for different operating points can even be automated. Scheduling with respect to the magnitudes of the loads applied is a preferable choice over other possibilities, because the operating point mainly depends on them. Since the environmental parameters change independently from the loads, it is desirable to design controllers, which are robustly stable with respect to these. Robust controller design utilising the small gain theorem might force results, that are too conservative though. An $\mathcal{H}_\infty$ norm based controller design will affect the loop and synthesis with regard to the aspects mentioned in table 3.7 in the following ways (refer to table 3.7):

Aspect	**Robust LQG Controller**
Tuning Procedure	• Demand for robustness introduces conservatism and more elaborate tuning might be required. • Additional effort for the analytical determination of uncertainty propagation.
Controller Complexity	• Uncertainty representation increases controller order. Certain controller synthesis approaches like μ-synthesis will likely lead to infeasible complexity, rendering controller order reduction a necessary step.
Disturbance Rejection	• Uncertainty representation may lead to improved disturbance rejection by increasing the applicable range of operating points.
Reference Tracking	• Reference tracking could, but does not need to, deteriorate due to increased conservatism.
Energy Optimality	• Demand for robustness may, but does not need to, deteriorate efficiency of control.
Noise Rejection	• Robust controller synthesis may and actually did lead to infeasible constraints on observer gain or other weightings, resulting in high noise amplification.

Table 3.7.2.: Possible Effects of Robust Controller Design

As can be inferred from table 3.7, robust controller design may yield analytical proofs of robustness against uncertain perturbations, which further increases safety of control. However, section 3.6.3 showed, that controller synthesis not dedicated to robustness may yield robust controllers. Nevertheless, it is necessary to find a reasonable uncertainty representation to be able to do the useful analysis steps.

4. Fuzzy Control

This chapter aims at a brief introduction to a different approach to heuristic controller design: The design of fuzzy controllers. First, some basics on fuzzy control are given and the MAMDANI controller is introduced. A short section deals with another special type of fuzzy controller, the TAKAGI-SUGENO-KANG controller. A brief section provides some basic methodology for fuzzy controller design using the mass-spring combination of the introduction as an example. After that, a general discussion on fuzzy versus classical control follows, in order to motivate the employment of fuzzy controllers under certain circumstances. The chapter concludes with a theoretical outlook towards the application of fuzzy control to the cooling cycle.

4.1. Basics on Fuzzy Control

Fuzzy controllers — or fuzzy systems in general — reflect the human way of thinking, insofar as it is the core idea to make decisions depending on vague knowledge, instead of explicit data. An example to illustrate the difference to the mathematical way of thinking is often given in the literature [12, p.1]: A human being driving a car will not think in values like *"reduce gas by 2 centimeters per second to reduce speed by 20 km/h"*. Instead, speed and the application of force to the gas pedal are identified by vague concepts like *"high velocity"* and *"moderate throttle"*. Loosely speaking, the latter concept of ambiguous or vague terms is the idea behind fuzzy control.

This section will provide some basic understanding of fuzzy controllers and terms involved. Some definitions are necessary, as to enable for the mathematical description of fuzzy systems. The notation and definitions follow [15] for the most part.

4.1.1. General Structure of a Fuzzy Controller

Generally, a fuzzy controller is comprised of four main items:

- **The Rule-Base:** The rule-base is the knowledge, how best to control a given plant. It can be regarded as the expertise of a control engineer stored in a set of If-Then instructions.
- **The Inference Mechanism:** The inference mechanism determines whether a certain rule is active and how the plant input should look like.

- **The Fuzzification:** The fuzzification is the process of transforming the controller input into a form, that applies to the rule-base. This may include the assignment to unsharp linguistic expressions, such as *"hot"* — hence the name.
- **The Defuzzification:** The defuzzification is the process of determining a crisp controller output value from the conclusions reached by the inference mechanism.

A common figurative term for a fuzzy controller is *"expert-in-the-loop"*, which already hints at advantages, as well as disadvantages the fuzzy logic approach yields. Figure 4.1.1 depicts the general structure of a fuzzy controller and the way it is incorporated into the control loop. To avoid confusion, it should be noted, that from now on controller inputs are denoted by $u_i \in \mathcal{U}_i$ and controller outputs as $y_i \in \mathcal{Y}_i$, such as to regard the fuzzy controller as a general fuzzy system with standard nomenclature for in– and outputs following [15].

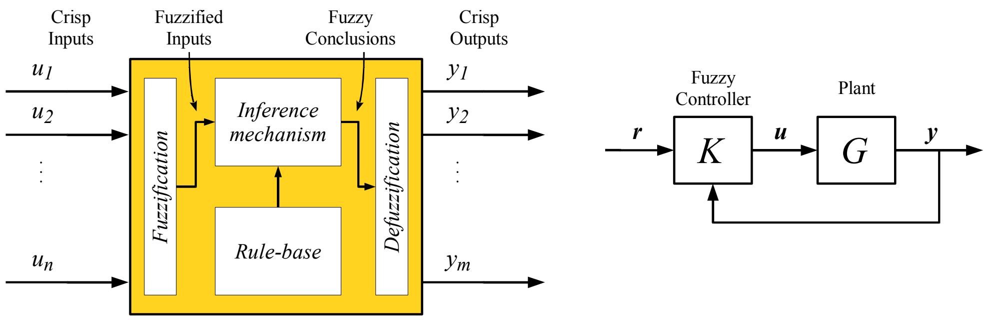


Figure 4.1.1.: Structure of a General Fuzzy Controller

The next section will define and explain the basic nomenclature involved in fuzzy systems.

4.1.2. Basic Linguistic Nomenclature of Fuzzy Systems

The inputs $u_i \in \mathcal{U}_i$ and outputs $y_i \in \mathcal{Y}_i$ of the fuzzy controller are the concrete or crisp values measured from the plant or assigned to its input, respectively. The sets $\mathcal{U}_i$ and $\mathcal{Y}_i$ are defined according to definition 4.1.1.

Definition 4.1.1 (Universe of Discourse) *A universe of discourse $\mathcal{U}_i$ or $\mathcal{Y}_i$ is the set of crisp values, that u_i or y_i may assume, respectively.*

An expert for a given plant would denote the inputs and outputs by their respective meaning, e.g. $\tilde{u}_1 =$*"velocity"*. This leads to the following definition 4.1.2:

Definition 4.1.2 (Linguistic Variable) *A linguistic variable $\tilde{u}_i$, $\tilde{y}_i$ is the trivial description of a certain input or output variable u_i or y_i.*

Definition 4.1.3 (Linguistic Value, Set of Linguistic Values) *Linguistic values are used to describe the characteristics a certain linguistic variable may take on.* $\tilde{A}_i^j$ *denotes the* j*th linguistic value of the linguistic variable* $\tilde{u}_i$*, while* $\tilde{B}_i^p$ *denotes a linguistic value for* $\tilde{y}_i$*, respectively.*

The linguistic values are arranged in sets

$$\tilde{A}_i = \{\tilde{A}_i^j : j = 1, 2, ..., N_i\}$$
$$\tilde{B}_i = \{\tilde{B}_i^p : p = 1, 2, ..., M_i\}$$

For example, consider a linguistic variable $\tilde{u}_1 =$ *"velocity"* with linguistic values $\tilde{A}_1^1 =$ *"negative"*, $\tilde{A}_1^2 =$ *"zero"*, $\tilde{A}_1^3 =$ *"positive"*.

To effectively make use of the intuitive way of describing the input and output data, the rule-base is comprised of mappings from input to output formulated in a linguistic manner:

Definition 4.1.4 (Linguistic Rule) *A linguistic rule is a mapping* $\tilde{u}_i \rightarrow \tilde{y}_i$*, defined in* modus ponens *form:*

If *premise* ***Then*** *consequent.*

A multiple-input-single-output (MISO) representation is

$$\textbf{\textit{If }} \tilde{u}_1 \textbf{\textit{ is }} \tilde{A}_1^j \textbf{\textit{ and }} \tilde{u}_2 \textbf{\textit{ is }} \tilde{A}_2^k ... \textbf{\textit{ and }} \tilde{u}_n \textbf{\textit{ is }} \tilde{A}_n^l \textbf{\textit{ Then }} \tilde{y}_q \textbf{\textit{ is }} \tilde{B}_q^p$$

According to [15, p.54], rules represented in multiple-input-multiple-output (MIMO) form can always be formulated by separate MISO rules, that are together linguistically equivalent. Therefore, definition 4.1.4 is regarded sufficient.

With the previous nomenclature settled, it is possible to formulate linguistic control laws, e.g.:

If *"velocity"* **is** *"too fast"* **Then** *"gas pedal"* **is** *"not pressed"*.

The next sections describe how the concrete input and output values u_i, y_q are processed according to a set of rules based on the linguistic nomenclature.

4.1.3. Membership Functions and Fuzzy Sets

The rules need a way of quantification, in order to assess the concrete meaning. This quantification is heavily based on heuristics and can be designed in the form of *"membership functions"*, that define to which extent a certain crisp value, e.g. of an input u_i, can be characterized in the sense of a certain linguistic value.

Definition 4.1.5 (Membership Function) *A function $\mu_{A_i^j}(u_i)$ associated with a linguistic value $\tilde{A}_i^j$ determines the degree of truth a value $u_i \in \mathcal{U}_i$, with linguistic description $\tilde{u}_i$, may be linguistically classified as $\tilde{A}_i^j$.*

Typical membership functions are [12, p.6ff] (refer to figure 4.1.2):

$$\textbf{Triangular:}\quad \Lambda_{a,b,c} : \mathbb{R} \to [0,1], \quad u \mapsto \begin{cases} \frac{u-a}{b-a} & \text{, if } a \leq u \leq b \\ \frac{c-u}{c-b} & \text{, if } b \leq u \leq c \\ 0 & \text{, else,} \end{cases} \tag{4.1.1}$$

where $a < b < c$ holds.

$$\textbf{Trapezoidal:}\quad \Pi_{a',b',c',d'} : \mathbb{R} \to [0,1], \quad u \mapsto \begin{cases} \frac{u-a'}{b'-a'} & \text{, if } a' \leq u \leq b' \\ 1 & \text{, if } b' \leq u \leq c' \\ \frac{d'-u}{d'-c'} & \text{, if } c' \leq u \leq d' \\ 0 & \text{, else,} \end{cases} \tag{4.1.2}$$

where $a' < b' < c' < d'$ holds.

$$\textbf{Bell-shaped:}\quad \Omega_{m,s} : \mathbb{R} \to [0,1], \quad u \mapsto e^{\frac{-(u-m)^2}{s^2}} \tag{4.1.3}$$

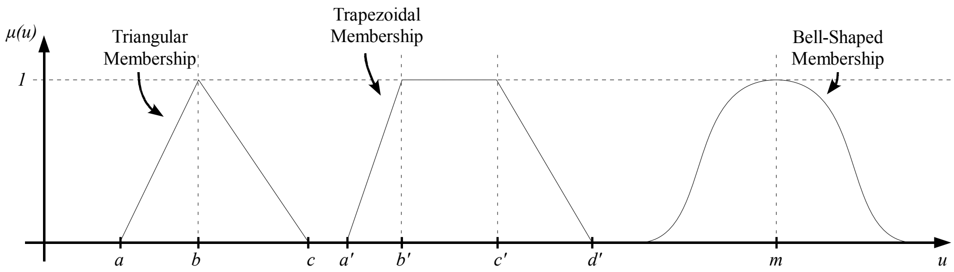


Figure 4.1.2.: Typical Membership Functions

The extension of classical set theory to the concept of vague membership is given in definition 4.1.6 [15, p.57, slightly altered] of a fuzzy set. In simple words, a fuzzy set is a classical crisp set comprised of tuples of crisp values and their respective degrees of membership with respect to a certain linguistic value.

Definition 4.1.6 (Fuzzy Set) *Given a linguistic variable $\tilde{u}_i$ with a linguistic value $\tilde{A}_i^j$ defined on the universe of discourse $\mathcal{U}_i$ and a membership function $\mu_{A_i^j}(u_i)$ that maps $\mathcal{U}_i$ to $[0,1]$, a fuzzy set A_i^j is defined as*

$$A_i^j = \Big\{ \Big((u_i, \mu_{A_i^j}(u_i)) : u_i \in \mathcal{U}_i \Big) \Big\} \tag{4.1.4}$$

4.1.4. Fuzzification

Prior to being able to use the inference mechanism, in order to determine conclusions, the direct inputs need to be fuzzified as explained in section 4.1.1 and shown in figure 4.1.1.

Practical applications often skip the fuzzification part of the fuzzy controller and directly use the explicit input values. Theoretically, this can be generalised, such that this approach is only a special case of fuzzification: The *singleton* fuzzification.

Definition 4.1.7 (Singleton Fuzzification) *The fuzzification is a mapping $\mathcal{F} : \mathcal{U}_i \to \mathcal{U}_i^*$ of an input u_i from the universe of discourse $\mathcal{U}_i$ to the set of all possible fuzzy sets $\mathcal{U}_i^*$, that can be defined on $\mathcal{U}_i$.*

The singleton fuzzification operator $\mathcal{F}^s$ is defined as:

$$\mathcal{F}^s(u_i) = \hat{A}_i^s, \tag{4.1.5}$$

where $\hat{A}_i^s$ is a fuzzy set whose membership function

$$\mu_{\hat{A}_i^s}(x) = \delta(x - u_i) = \begin{cases} 1 & , x = u_i \\ 0 & , otherwise \end{cases} \tag{4.1.6}$$

is dynamically changing over time as u_i changes.

Figure 4.1.3 illustrates the singleton fuzzification.

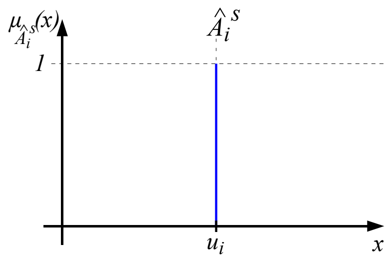


Figure 4.1.3.: Singleton Fuzzification

In [15, p.62] the authors reason, that other fuzzification methods are not frequently used in practical applications, because the computational complexity increases and their benefit has not been well justified. For example, a gaussian bell-shaped membership function could be used, to alleviate the influence of noise to the controller output. Still other methods, like applying a filter to the plant output, may yield computationally less expensive, but practically similar results. This has been done in the introductory example.

4.1.5. Fuzzy Operators — The Mamdani Controller

It can be inferred from the definitions given so far, that the **is** operator is just the determination of a degree of membership with respect to a certain linguistic value $\tilde{A}_i^j$, under the assumption, that singleton fuzzification is used (refer to figure 4.1.4).

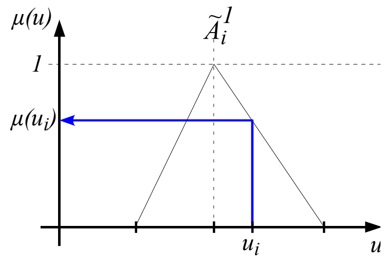


Figure 4.1.4.: Determination of Degree of Membership (**IS**)

A more general representation is needed for different fuzzification procedures, which will be omitted here. Refer to [15, p.63] for more information on that topic.

It is, however, still unclear, how to interpret the **If-Then** rules exactly, since possible fuzzy operators **and**, **or**, **not** and **Then** are still undefined. There are multiple ways to define these operations, but those presented here are limited to those used by the Mamdani *controller*, which is for reasons, that will become obvious, also frequently called Min-Max *controller*.

Definition 4.1.8 (Fuzzy Complement (NOT)) *The complement $\bar{A}_i^j$ of a fuzzy set A_i^j with membership function $\mu_{A_i^j}(u_i)$ is given by*

$$\mu_{\bar{A}_i^j}(u_i) = 1 - \mu_{A_i^j}(u_i) \tag{4.1.7}$$

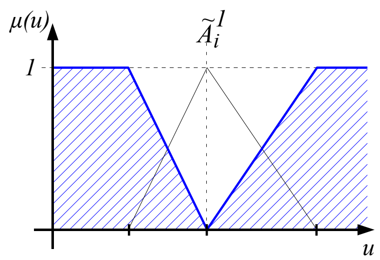


Figure 4.1.5.: Fuzzy Complement (**NOT**)

Definition 4.1.9 (Fuzzy Intersection (AND)) *The intersection of two fuzzy sets A_i^1 and A_i^2, denoted by $A_i^1 \cap A_i^2$, defined on the same universe of discourse $\mathcal{U}_i$ has the membership function given by*

$$\mu_{A_i^1 \cap A_i^2}(u_i) = \min\{\mu_{A_i^1}(u_i), \mu_{A_i^2}(u_i) : u_i \in \mathcal{U}_i\} \tag{4.1.8}$$

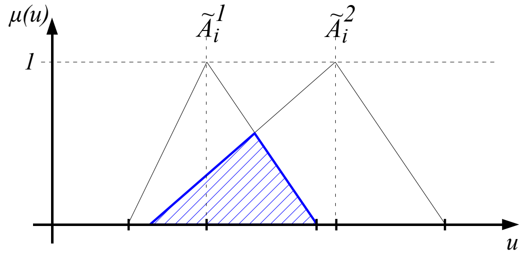


Figure 4.1.6.: Fuzzy Intersection (**AND**)

Definition 4.1.10 (Fuzzy Union (OR)) *The union of two fuzzy sets A_i^1 and A_i^2, denoted by $A_i^1 \cup A_i^2$, defined on the same universe of discourse $\mathcal{U}_i$ has the membership function given by*

$$\mu_{A_i^1 \cup A_i^2}(u_i) = \max\{\mu_{A_i^1}(u_i), \mu_{A_i^2}(u_i) : u_i \in \mathcal{U}_i\} \tag{4.1.9}$$

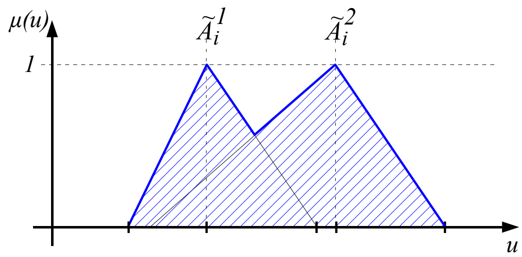


Figure 4.1.7.: Fuzzy Union (**OR**)

The intersection and union only act on a single universe of discourse. Typical rules, however, will often combine two premises defined on different universes. The Cartesian product is used to create a multidimensional fuzzy set from the combination of multiple fuzzy sets.

Definition 4.1.11 (Fuzzy Cartesian Product) *If $A_1^j, A_2^k, ..., A_n^1$ are fuzzy sets defined on the respective universes of discourse $\mathcal{U}_1, \mathcal{U}_2, ..., \mathcal{U}_n$, the Cartesian product $A_1^j \times A_2^k \times ... \times A_n^l$ is a fuzzy set with membership function*

$$\mu_{A_1^j \times A_2^k \times ... \times A_n^1}(u_1, u_2, ..., u_n) = \mu_{A_1^j}(u_1) * \mu_{A_2^k}(u_2) * ... * \mu_{A_n^l}(u_n). \tag{4.1.10}$$

The operator $*$ can be understood as a kind of **AND** intersection. The illustration in figure 4.1.8 helps visualising the membership function of the Cartesian product for two fuzzy sets.

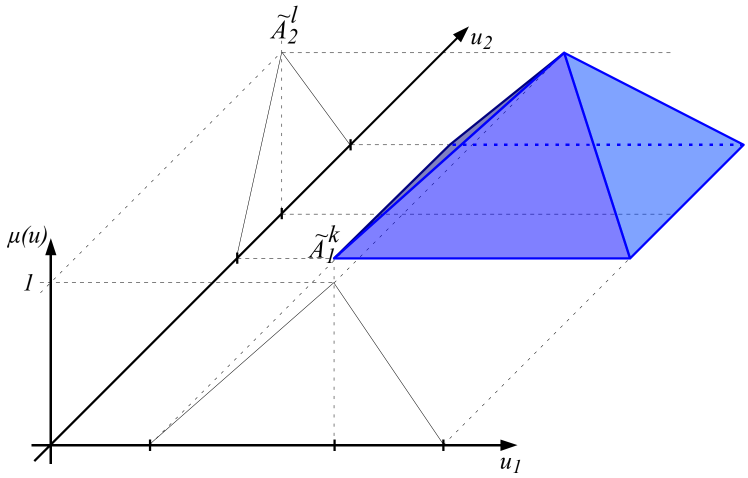


Figure 4.1.8.: Fuzzy Cartesian Product

As mentioned before, the definitions presented here only cover one single type of each operator. There are various other possibilities in fuzzy logic. For instance, a choice to define the fuzzy intersection also common to fuzzy control theory is the *algebraic product*, as well as the *algebraic sum* for the fuzzy union. Please refer to [15, p.58ff] for the respective definitions. An even more general overview, which is not limited to control theory applications, is provided in [12, p.19ff]

4.1.6. Fuzzy Inference Mechanism

The inference mechanism can be divided into two steps: The *matching* and the *inference step*.

Matching

With singleton fuzzification, the matching step is the task to determine, which rules are "on": For the i^{th} rule the degree of membership $\mu_i(u_1, u_2, ..., u_n)$, that is, the degree of certainty, that the premises hold true for the given inputs $u_1, u_2, ..., u_n$, is computed.

$$\mu_i(u_1, u_2, ..., u_n) = \mu_{A_1^j}(u_1) * \mu_{A_2^k}(u_2) * ... * \mu_{A_n^l}(u_n) \tag{4.1.11}$$

$\mu_i(u_1, u_2, ..., u_n)$ can be regarded as a multidimensional certainty surface [15, p.63].

Inference Step

Drawing the conclusion from the rule-base and the degree of certainty that the rules hold is called the *inference step*. For every rule, the membership function of the implied fuzzy set with regard to the output variable is determined:

$$\mu_{\hat{B}_q^i}(y_q) = \mu_i(u_1, u_2, ..., u_n) * \mu_{B_q^p}(y_q) \tag{4.1.12}$$

Note, that again the hat on the output fuzzy set $\hat{B}_q^i$ implies, that the shape of the membership function $\mu_{\hat{B}_q^i}$ changes over time.

The fuzzy set $\hat{B}_q^i$ can be interpreted in the following way: With respect to the i^{th} rule, the degree of certainty, that a crisp output $y_q \in \mathcal{Y}_q$ should be chosen as the controller output is specified by the membership function $\mu_{\hat{B}_q^i}(y_q)$.

The next step is to determine, which exact value the output should assume. This task is called *defuzzification*.

4.1.7. Defuzzification

As is most often the case in fuzzy logic control theory, there are multiple ways of achieving a goal. This is also true with the defuzzification, which could be both applied to the implied fuzzy sets, derived during the inference step for every single rule, or to an overall implied fuzzy set, after the individual implied fuzzy sets have been combined. This thesis will focus on the first procedure only.

Center of Gravity (COG)

In analogy to the center of mass, one way to determine a crisp output value y_q^{crisp} is as follows:

$$y_q^{\text{crisp}} = \frac{\sum_{i=1}^{R} b_q^i \int_{\mathcal{Y}_q} \mu_{\hat{B}_q^i}(y_q)\ dy_q}{\sum_{i=1}^{R} \int_{\mathcal{Y}_q} \mu_{\hat{B}_q^i}(y_q)\ dy_q}, \tag{4.1.13}$$

where R is the number of rules,

b_q^i is the center of area of the membership function $\mu_{\hat{B}_q^i}(y_q)$

and $\int\limits_{\mathcal{Y}_q} \mu_{\hat{B}_q^i}(y_q)\ dy_q$ is the area under each membership function

belonging to the fuzzy set $\hat{B}_q^i$.

Center Average

Instead of weighting the center of area with the area under the membership function, the center average method uses the supremum (or in simpler terms: the maximum) value of the membership function.

$$y_q^{\text{crisp}} = \frac{\sum_{i=1}^{R} b_q^i \sup_{y_q}\{\mu_{\hat{B}_q^i}(y_q)\}}{\sum_{i=1}^{R} \sup_{y_q}\{\mu_{\hat{B}_q^i}(y_q)\}}, \tag{4.1.14}$$

where R is again the number of rules,

b_q^i is the center of area of the membership function $\mu_{\hat{B}_q^i}(y_q)$

and $\sup\limits_{y_q}\{\mu_{\hat{B}_q^i}(y_q)\}$ can be interpreted as the maximum

possible degree of certainty over the membership function.

Figure 4.1.9 illustrates the difference between both methods in a rather extreme case. The red lines indicate, which component is responsible for the weighting of the center of area. The thickness of the arrows indicate their respective weights.

Remark

It should be noted, that the respective denominators for both methods should fulfill

$$\sum_{i=1}^{R} \int\limits_{\mathcal{Y}_q} \mu_{\hat{B}_q^i}(y_q)\ dy_q \neq 0 \qquad\qquad \sum_{i=1}^{R} \sup_{y_q}\{\mu_{\hat{B}_q^i}(y_q)\},$$

which requires, that no case exists, where all output membership functions become zero.

4.1.8. Summary

The covered aspects of fuzzy logic and fuzzy controller structures require some rigorous definitions and indexing when described mathematically. On the other hand, a visual explanation is most often very helpful and feasible. Most of the fuzzy controller mechanism can be summarised with just a few pictures and sentences. Actually, the intuitive ease of use is one of the main strengths of fuzzy control theory. Figure 4.1.10 illustrates the complete inference mechanism and defuzzification procedure on the basis of two rules:

1. Via singleton fuzzification, determine the degrees of truth of each part of the premise.
2. Combine the truth values found by fuzzy operators, such as **and**.
3. Draw conclusions for each rule by determining the degree of membership with regard to the consequent.
4. Defuzzify the conclusions of the single rules to one crisp output value.

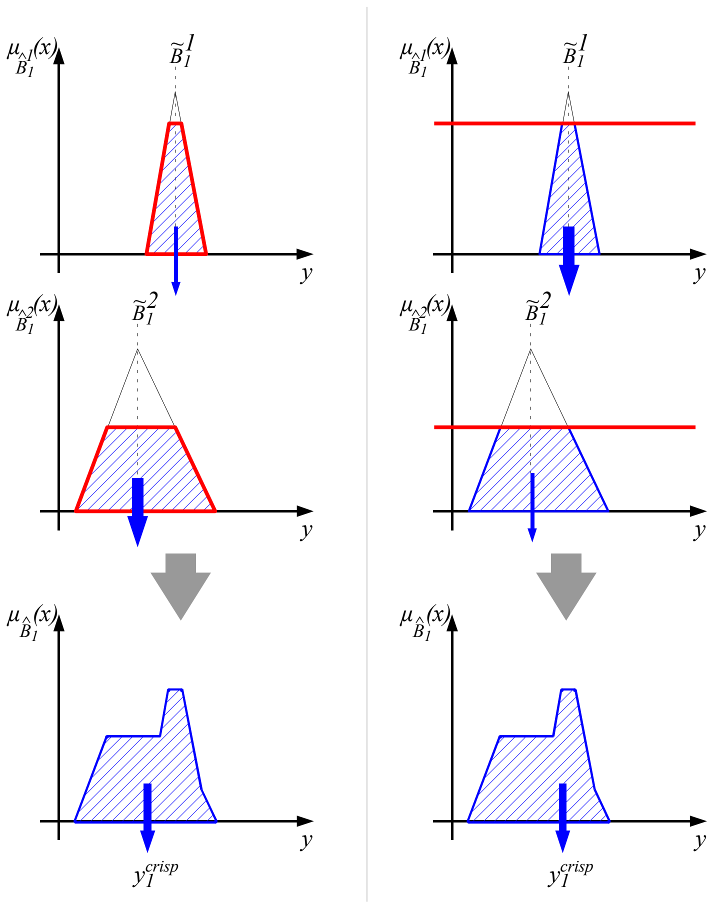


Figure 4.1.9.: Comparison of Center of Gravity (l.) and Center Average (r.) Defuzzification

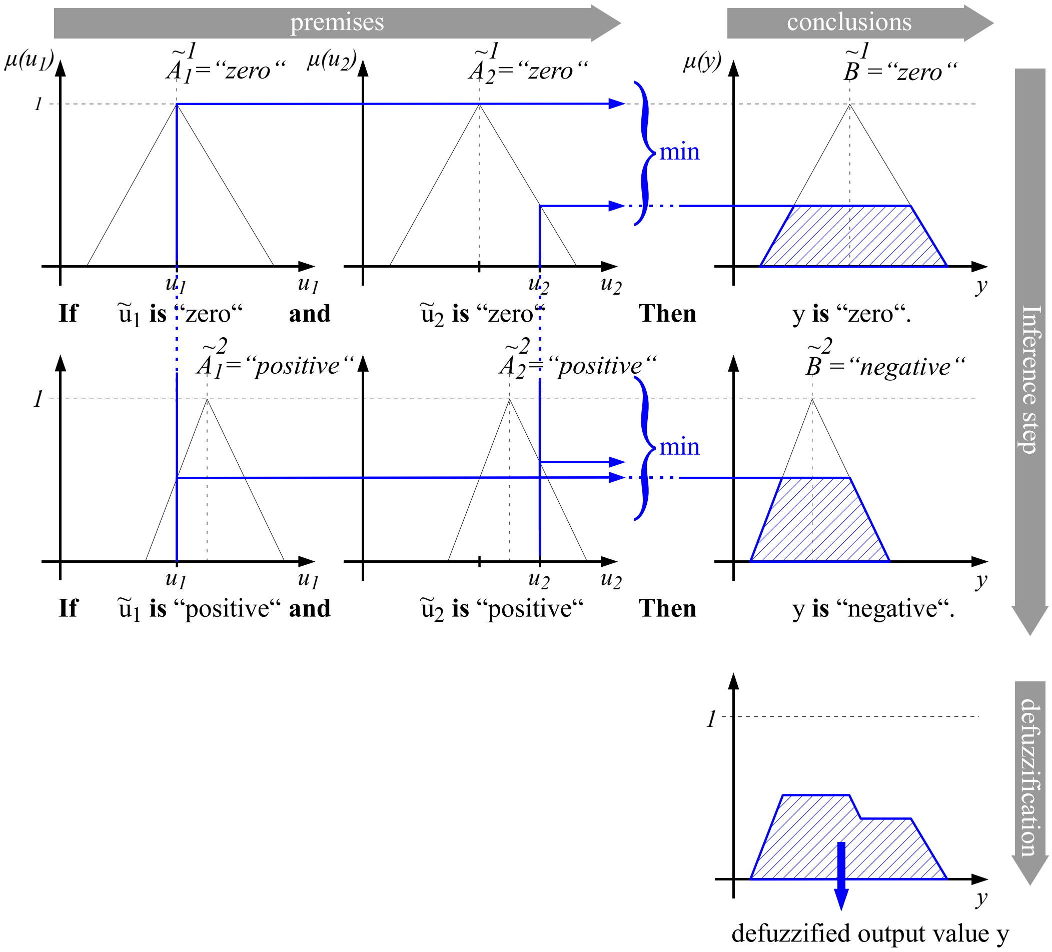


Figure 4.1.10.: Overview of MAMDANI Fuzzy Controller Operations (altered from [12, p.238])

4.1.9. General Fuzzy Systems and the Takagi-Sugeno-Kang Fuzzy Controller

In order to provide an outlook to the vastness and possible power of the general fuzzy system framework, a more general representation of a fuzzy system may be formulated. By partly dumping the linguistic nature of the rule-base and substituting the consequent linguistic term by a function $b_i = g_i(\cdot)$, a so-called *functional fuzzy system* is obtained.

$$\textbf{If } \tilde{u}_1 \textbf{ is } \tilde{A}_1^j \textbf{ and } \tilde{u}_2 \textbf{ is } \tilde{A}_2^k ... \textbf{ and } \tilde{u}_n \textbf{ is } \tilde{A}_n^l \textbf{ Then } b_i = g_i(\cdot) \tag{4.1.15}$$

Singleton fuzzification and the operator definitions are kept. The function $b_i = g_i(\cdot)$ may depend directly on the inputs $u_1, u_2, ..., u_n$ and may assume any possible form — linear, or non-linear. In case of the TAKAGI-SUGENO-KANG controller, the function is restricted to be linear with respect to the inputs. The mapping can also be a neural network, which might be trained automatically. The general description allows for the interpretation of a fuzzy system to be a *universal approximator* for non-linear functions [15, p.77]. Without limits on computational expense or tuning effort, a fuzzy system has thus the power to yield very powerful controllers for a given plant, provided the plant in– and outputs allow for the achievement of certain design objectives.

4.2. Fuzzy Controller Design

Fuzzy controller design heavily relies on heuristics. For simple control tasks this can be compared to PID controller design, for more complex plants, however, it requires a well-founded understanding of the plant dynamics. In the following sections, some basic fuzzy controller design approaches are presented using the mass-spring combination from the introduction as an example. The schematic mechanical diagram is reprinted here for convenience (refer to figure 4.2.1).

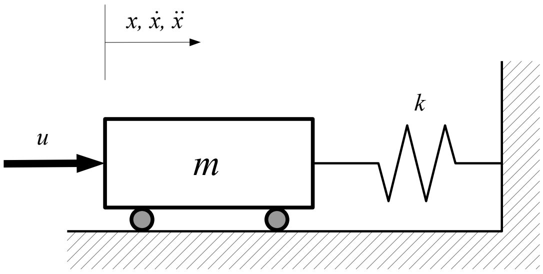


Figure 4.2.1.: Simple Mass-Spring Combination from Introductory Example

4.2.1. Adding Dynamics to the Plant Output

Controller design most often begins with a detailed analysis of the plant and the determination and manipulation of inputs and outputs. Conventional control theory ideas are helpful, since they still apply to fuzzy controllers. Assume, for instance, that only the position x of the cart is measured. Like a PID controller uses the derivative and integral of a given output signal as an additional "input" to the controller, such modifications are suitable for fuzzy control as well. They can be utilised to assure reference tracking or to achieve a quicker response. Section 4.1 on basics of fuzzy control views fuzzy systems as non-dynamical systems. Storage or accumulation of signals (integrative action) is not a trait inherent to the fuzzy system. Instead, the fuzzy system only describes simple input-output behaviour via certain transformations. Therefore it is often possible to store the resulting non-linear control law in a simple look-up table, as will be made obvious later.

In the introductory example, the fuzzy control system is supplied with a derivative dynamic element, such that also the velocity $\dot{x}$ may act as an input to the controller. Figure 4.2.2 depicts the resulting control loop. Gains g_0, g_1 and h have been introduced, which act as scaling factors on the u- or y-axis of the membership functions. The gains are a simplified way to tune the fuzzy control loop, once the definition of the fuzzy system is basically working for $g_0 = g_1 = h = 1$. This way of tuning is restrictive, but since there are virtually unlimited ways to tune either the rule-base or the membership functions, reducing the number of "tuning knobs" for final tweaking is a good idea, although this means to abandon some of the flexibility of fuzzy controller design.

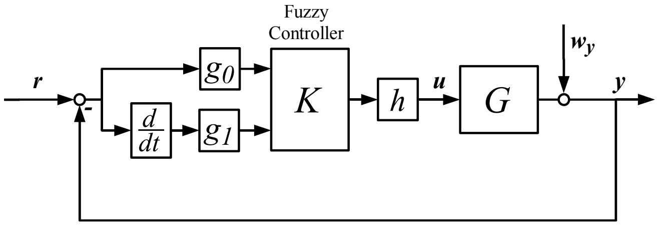


Figure 4.2.2.: General Control Loop for Fuzzy PD Controller

The control loop basically has the form of PD control.

4.2.2. Defining the Membership Functions and Rule-Base

There is no definite or unique way for defining both membership functions and rules. The number of rules or linguistic values, that are needed to achieve good design, cannot be determined beforehand. To keep the fuzzy system simple and understandable, try for as few as possible and as many as necessary. Table 4.2.2 at the end of this section lists a short overview of methods to obtain a set of rules, their advantages and disadvantages.

Because of the simplicity of the mass-spring combination, it is possible to derive a rule-base based on a set of membership functions by heuristics only. Since a PD fuzzy controller has been chosen, for each of the two inputs and single output a set of membership functions is needed. Figure 4.2.3 illustrates a possible definition.

The fact, that the velocity is not as densely divided into linguistic values as the position is a rather random choice. Bearing simplicity in mind, it has turned out, that it works. Thus no additional complexity is needed.

It is vital, that the full range of possible input values, as well as output values, is properly covered by membership functions. Otherwise, certain states are unknown to the fuzzy system. Mathematically, this can lead to singularities during COG defuzzification, for instance.

The rule-base can typically quickly enlarge to a vast list of **If-Then** statements. Whenever possible, it should be depicted in a comprehensible, e.g. graphical, way. In case of the mass-spring combination, displaying the rules in a convenient form is particularly easy. Due to the two-input-single-output form a table is well suited (refer to table 4.2.2).

The symmetry inherent to the representation is a natural characteristic of the plant to be controlled. Such phenomena may often occur and become visible only by some effort of displaying the rule-base. The advantage of doing so is a better understanding of the control law and makes the design procedure less prone to errors.

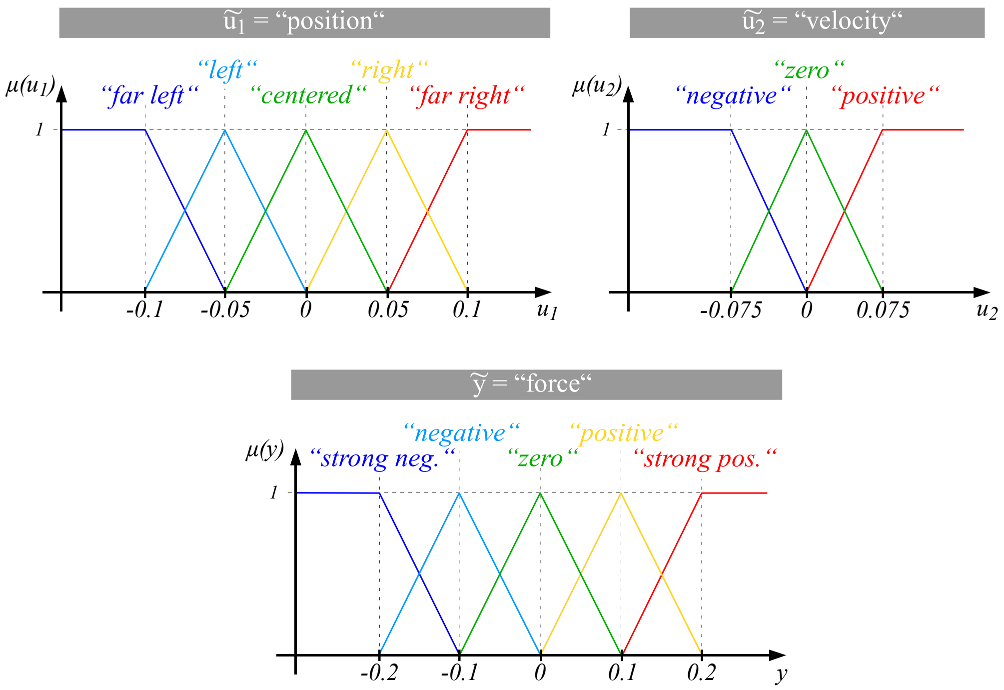


Figure 4.2.3.: Membership Function Definitions for PD Control of Mass-Spring Combination

		$\tilde{u}_2$		
		"negative"	**"zero"**	**"positive"**
$\tilde{u}_1$	**"far left"**	"strong positive"	"strong positive"	"positive"
	"left"	"strong positive"	"positive"	"zero"
	"centered"	"positive"	"zero"	"negative"
	"right"	"zero"	"negative"	"strong negative"
	"far right"	"negative"	"strong negative"	"strong negative"

Table 4.2.1.: Rule-Base in Table Notation

Method	Advantages	Disadvantages
Observation and recording of manual control	• Easy and intuitive approach, which may require only some learning. • May cover non-linearities very well	• May not cover full spectrum of operation or capabilities of actuators. • Restricted to relatively simple systems. • Restricted to relatively slow systems.
Analysis of physical plant model	• Rigorous descriptions of non-linearities. • Full spectrum of operation can be analysed	• Modelling effort may justify a different controller synthesis approach. • Difficulty in inferring rules from the mathematical description.
Analysis of fuzzy plant model	• Rules can be directly inferred from fuzzy plant model. • Potentially high performance.	• Fuzzy modelling effort may be very complicated.
Automated tuning or learning	• Optimality as specified by cost functions possible. • Applicable also if further heuristic tuning is very difficult.	• Cost function must cover all important aspects. • Knowledge of genetic algorithms or neural networks needed.

Table 4.2.2.: Advantages and Disadvantages of Fuzzy Controller Synthesis Methods

4.2.3. Implementation and Round-Up

As mentioned earlier, a fuzzy system like the MAMDANI controller does not contain dynamical elements by itself. They have to be introduced to the control loop by heuristic means, following classical control theory. Because of this and despite the fuzziness inherent to the system, the fuzzy controller has a clearly defined input-output behaviour. It can be imagined as a multidimensional space with several input base directions and output base directions. Every possible combination of inputs has to lead to an appropriate set of outputs. In case of the mass-spring combination, the control law without dynamic elements is a three dimensional surface as depicted by figure 4.2.4.

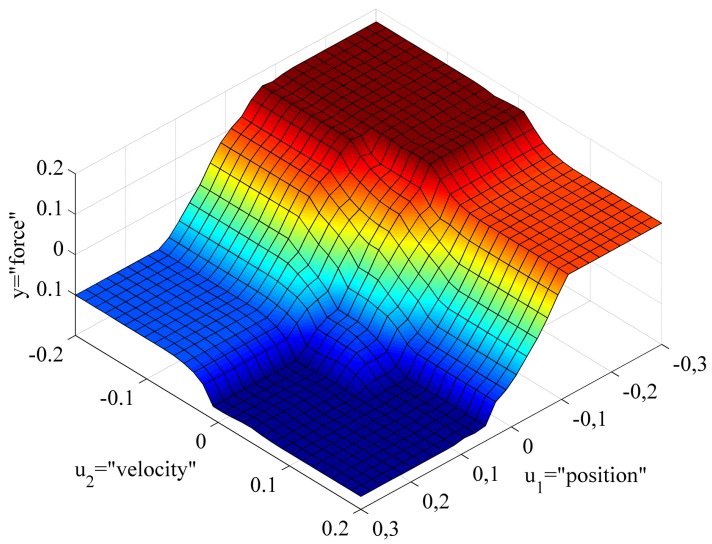


Figure 4.2.4.: Control Surface of Fuzzy Controller

In this simple example, it is therefore possible to reduce the fuzzy controller to a mere look-up table. The fuzzy controller would thus become particularly easy to implement and the only restrictions are then found in hardware memory space, which determines the accurateness of the control surface stored. In case of figure 4.2.4, 30×30 input values have been sampled. Implementing the controller is now as simple as storing a table of 900 output values and adding a derivative to the plant output. Compared to static full information feedback, whose the feedback gain matrix has the dimensions $F \in \mathbb{R}^{1\times 2}$, this is extremely much, though. Thus, the benefits of fuzzy control over classical control have to be clearly justified with regard to controller complexity. More specifically, if the control surface approaches a shape, which can be modelled by classical controllers more easily (e.g. a plane, which corresponds to proportional feedback), the advantage of a fuzzy controller is to be doubted.

4.3. Evaluation of the Fuzzy Control Approach

Fuzzy control is a very promising alternative for many different applications due to its intuitive nature. Commercial software tools, like the *Fuzzy Control Toolbox* for *MATLAB*, are readily available, that provide user-friendly interfaces to express the rule sets. The seemingly ease of use in fuzzy controller design raises questions for the drawbacks. This section aims at a brief discussion of fuzzy versus classical control, based on some main issues in controller design: plant modelling and synthesis methods, tuning, safety (robustness) and implementation.

4.3.1. Fuzzy Control versus Classical Control

To properly define, which controller synthesis approaches are regarded in this comparison, the terms "classical controller" and "fuzzy controller" should be understood in the following way:

- **Fuzzy Controllers:** By fuzzy controllers, the controller is understood to be similar to those, that can be developed with the theory given in section 4.1. Advanced controllers, like neuro-fuzzy controllers, which are optimised by neural networks, are mentioned explicitly, when considered.
- **Classical Controllers:** Classical control theory is regarded to consist of design approaches like, PID control, state feedback or state estimate feedback control, where linearised plant models are frequently utilised during controller synthesis.

Plant Modelling and Synthesis Methods

It is sometimes stated, that with fuzzy control design, there is no need for a plant model. While, theoretically speaking, the designer does not need a mathematical model to setup the rules and membership functions for a fuzzy controller in the same way as a plant model is needed to calculate, e.g., a state feedback controller, there are several other advantages to the process of deriving a suitable model for simulation. For one point, a simulation is always useful if the controller cannot be tested on the actual plant for safety or cost issues. At the same time, the mathematical analysis in the form of differential equations can be helpful to gain a basic understanding of the plant dynamics in the first place. In addition, a measure for optimality with respect to some states of the plant is often inherent to the mathematical representation

Depending on the expertise and personal preferences of the controller designer, the process of formulating a linguistic rule-base may be an easier task, though this must not necessarily be the case. It should be kept in mind, that a manual definition of a set of rules is prone to errors, whereas plant modelling has to be accompanied by validation, where assumptions and simplifications on the plant dynamics are evaluated and are less likely to be overlooked. On the other hand, plant models that are used for controller synthesis are often subject to restrictions (e.g. linearity), which prevent reasonable results.

Tuning

Fuzzy controller design is heavily based on heuristics, which make the tuning process more elaborate, but also more flexible. Many industrial applications of PID controllers involve a comparable amount of heuristics, which has proven to be quite successful.

For the ease of tuning, modern control theory frequently aims at minimising the amount of parameters, that need tuning. With fuzzy controller design, the number of tuning "knobs" is virtually unlimited, since membership functions, input and output gains, operator definitions or the controller structure may be adjusted in numerous ways.

A possible solution for the automated tuning of both PID and fuzzy controllers is optimisation via genetic algorithms.

Safety

When considering the fuzzy controller as an expert-in-the-loop, one of the primary notions, why controllers are being employed becomes apparent: Naturally, computers are quicker to react on changing measurement data, but in addition, they are very often more reliable than a human operator. A human expert can not necessarily foresee any possible disturbance to the system, which may be crucial in hazardous applications, where failure may lead to loss of human life. With a deeper understanding of fuzzy control theory, the description of a fuzzy controller as a non-linear controller, enables for the evaluation of the transfer behaviour without any fuzziness to it. But it should be noted, that the term robustness is meaningful only with respect to some quantified deviation from a specified nominal plant. A plant does not exist in fuzzy controller design and thus, the term robust can only be evaluated by simulation test runs. Conventional controller design incorporates some well developed techniques, which can guarantee robustness under certain deviations from the nominal model (e.g. parameter uncertainty). On the contrary these uncertainties have to be properly defined beforehand, too, though they can be defined, such as to analytically introduce robustness margins, taking into account all possible combinations of uncertainties.

This mainly leaves the systematic approach of conventional controller design as an advantage to provide higher safety standards.

Implementation

There are two different sides of implementation issues: First, the adoption of a controller to a real environment, while formerly only simulation results have been taken into account, and second, the computational complexity involved with a specific choice of controller, which has to be considered beforehand according to prevailing constraints. The first issue is mainly covered, when previously considering the tuning of controllers. A final tuning is more quickly done, with fewer parameters, whereas a vast set of rules is not that easily adopted. However, this all depends on the designers expertise, which may lead to only a very small amount of in-practice tuning being necessary.

The computational complexity of a controller is a factor, which might disqualify a certain approach right from the beginning design phase. It should be noted, that the computational complexity of fuzzy controllers as explained in previous sections generally outweighs that of simple PID controllers even for basic applications. But in real applications, all possible combinations of input and output data can be precalculated with a certain desired precision. The resulting *control surface* can then be implemented like a simple look-up table, defining the transfer behaviour of the fuzzy controller just as sharply as any transfer function does. The amount of storage space required could quickly rise with the use of adaptive fuzzy controllers, though. However, to compare the computational efforts for adaptive controllers — fuzzy or conventional — are beyond this work.

Summary

Table 4.3.1 summarises the comparison between fuzzy and classical control.

Some guidelines, when a fuzzy control approach is useful is provided in [12, p.255]. The authors state, that if a model in the form of differential or difference equations suitable for the application of classical control exists, the classical methods should be tried first. Fuzzy controllers are a promising alternative, if

- No model exists or the non-linearities prevent classical linear control approaches.
- The design objectives are given in a vague way.
- The construction of a fuzzy controller for a given plant can be done more quickly than the derivation of a linearised plant model and the appropriate synthesis of a classical controller.

Aspect	**Fuzzy Control**	**Classical Control**
Plant Modelling and Synthesis Methods	• Plant modelling mainly facilitates understanding and enables for testing by simulation. • Expert rule set needed. • Linguistic representation yields high intuitive potential.	• Plant modelling and linearisation most often necessary. • Restrictive assumptions needed. • Systematic approach and mathematical representation facilitates optimal control.
Tuning	• Various tuning possibilities provide flexibility. • Heuristic tuning procedure may become cumbersome and confusing. • Automated tuning possible	• Reduced amount of tuning "knobs". • Implementation still involves heuristic tuning. • Automated tuning possible.
Safety	• No accurate measurement of robustness. • Human expertise may be lacking.	• Rigorous mathematical robustness analysis possible. • Some uncertainties may remain unmodelled.
Implementation	• Relatively high complexity even for simple applications. • Computational effort can be traded for storage demands.	• Complexity depends on synthesis method. • Controller order reduction possible.

Table 4.3.1.: Comparison of Fuzzy vs. Classical Control

4.4. Possible Applications to the Cooling Cycle

This section briefly introduces and discusss some ideas of how fuzzy control might be useful for the control of the cooling cycle.

4.4.1. Full Fuzzy Control

As has been argued in section 4.2.3, fuzzy controllers pay off, if their control "surface" (mathematically, a hyperplane) is non-linear enough, such that it is very difficult to generate by means of classical control. Actuator constraints introduce such non-linearities, which are, in general, very hard to achieve by classical means, while still being able to retain analytical proof of stability, for instance. On the other hand, fuzzy control theory, to the degree covered in this thesis, provides no guarantee for stability at all. Further study of this topic is needed for this.

The fact, that desired reference values of the temperatures are given in a vague way is a strong argument for fuzzy control. A fuzzy control approach promises to enable a further reduction of the energy consumption. In cases, where the temperatures lie in a fuzzy boundary around the reference value, control action could be very soft, not only keeping the pump speed low, but also at a value, which corresponds to the highest degree of efficiency. However, a full fuzzy control approach might not even be necessary to improve this. Instead, the reference values could be scheduled via a fuzzy system as illustrated in figure 4.4.1. Still, following a fuzzy approach is only useful, if the scheduling is subject to considerable non-linearities. Otherwise a more simpler method should be preferred. Furthermore, care should be taken, such that the scheduling of reference values does not interfere with the actual control. It should be comparatively slow and limited to a dependence on external parameters.

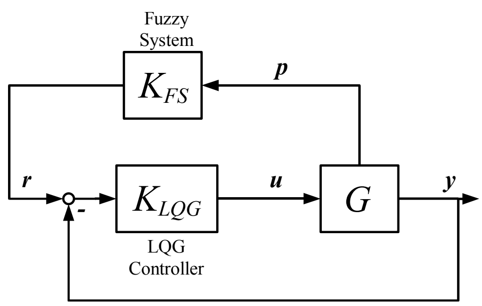


Figure 4.4.1.: Scheduling of Reference Values via a Fuzzy System

Since basically classic controllers work quite well with the plant, there is no imminent need to design a substitute fuzzy controller. For scheduling purposes, a fuzzy approach might be useful, though.

4.4.2. Fuzzy Gain Scheduling

For linearised gain scheduling, an interpolation between certain linearisation points can be done, in order to also cover operating points in-between. Nothing can be stated with regard to the stability or performance in these intermediate states of the system, though. [5] points out, that the blending between a set of local linear controllers can be very difficult, which also depends on the kind in which the scheduling parameter is influenced by the operating point.

In [15, p.] fuzzy gain scheduling is divided into three different approaches.

"Heuristic gain schedule construction" is described as either defining a way to interpolate between a set of local linear controllers or, if possible, to directly schedule the gains of a certain simple controller. A heuristically tuned controller, like the PI-controller structure presented in this thesis, would be particularly well suited.

In *"identification for gain schedule construction"* a fuzzy system is used as an approximator for non-linear functions. The goal of this approach is to identify and construct a non-linear scheduling law.

"Parallel distributed compensation" can be used for plant models, that can be written in polytopic LPV form

$$\dot{\boldsymbol{x}}(t) = \left(\sum_{i=1}^{R} \boldsymbol{A}_i \boldsymbol{\theta}_i(\boldsymbol{x}(t)) \right) \boldsymbol{x}(t) + \left(\sum_{i=1}^{R} \boldsymbol{B}_i \boldsymbol{\theta}_i(\boldsymbol{x}(t)) \right) \boldsymbol{u}(t). \tag{4.4.1}$$

Sometimes also considered simply as smooth blending, instead of fuzzy gain scheduling, a control law

$$\boldsymbol{u}(t) = \sum_{i=1}^{R} \boldsymbol{K}_i \boldsymbol{\theta}_i(\boldsymbol{x}(t)) \boldsymbol{x}(t) \tag{4.4.2}$$

can be realised.

5. Conclusion and Outlook

As a final chapter of this thesis, the following sections summarise the approaches and their respective results, as well as provide an outlook to possible future work.

5.1. Conclusion

This thesis consists of three main parts. In the first one, a thorough physical analysis of a simple cooling cycle has been done and a non-linear model has been constructed for the simulation in *MATLAB/Simulink*. This has facilitated the determination of the main issues, controller design is confronted with, and has already given hints for possible solutions. The non-linear simulation has helped to assess the controllers' performance under changing operating conditions. A quick and intuitive heuristic tuning in *MATLAB/Simulink* was possible with has improved work flow over *Flowmaster*.

The second part is dedicated to the actual controller design. Going from simple to more complex approaches, the goal to find effective, yet feasible ways of controller synthesis has been met. The physical modelling of the plant has enabled to systematically derive a linearised plant model. To further improve the applicable range of the linearised model, a non-linear transformation of plant inputs has been found, which simplified the plant-controller interface. With the help of the linearisation and model-based controller synthesis, it has been shown, that highly effective control is possible via LQG gain scheduling. Still, even heuristically tuned PI control readily provides a choice for backup control, that can be fallen back to in case of scheduling signal sensor failure. In case of considerable output noise, PI control can be rendered infeasible, though, and the negative effects of output noise filtering have been discussed. In contrast, LQG control provides integrated KALMAN filtering, which can be utilised to alleviate controller output noise as a trade off for control performance. The simulation results have shown closed-loop behaviour superior to PI control. A simple gain scheduling scheme has been proposed, which — apart from improving the power consumption — has led to a more accurate specification of desired robustness properties of the controller. $\mathcal{H}_\infty$ norm based robust control in conjunction with the small gain theorem has been briefly introduced as an opportunity to go beyond heuristic assessment of robustness via non-linear simulation. Furthermore, an exemplary robustness analysis has been done with respect to perturbations in environmental parameters for a normal LQG controller.

In part three, an additional theoretical chapter on fuzzy control has depicted a different heuristic approach as compared to classical controller synthesis. A discussion on the advantages and drawbacks has led to a theoretical view on the applicability of fuzzy control to the cooling cycle. As a result, fuzzy control can, for instance, be proposed as a systematised scheduling solution on the steering level.

Although this thesis provides applicable control solutions for cooling cycle architectures, that basically resemble the one considered, there are multiple ways of enhancing the presented ideas and, of course, even very different approaches. The following outlook aims at a brief view at various promising possibilities not covered in this thesis.

5.2. Outlook

This thesis has mainly covered conventional controller synthesis methods, whose practical and industrial relevance is often stated in common literature. This mainly stems from the fact, that in order to be economically effective, simple solutions are preferred over more complex ones, if a satisfying result can be achieved with less effort. While this approach is sound, increasing shortage of resources and therefore energy already shifts the prevailing attitude towards the development of more sophisticated methods prior to implementation.

Additional Measurements and Cascaded Non-Linear Control

This thesis focusses on a single set of measured output data and controlled input data. It has been discussed in section 2.5.3 that the measurement of mass flow rates is not a common practice in aircraft engineering. Despite this, possibly reliable and non-invasive means exist, e.g. in the form of ultra sonic flowmeters. An indirect method to measure the total mass flow rate $\dot{m}_0$ could also be invented: Single pressure sensors in front of and behind the pumps could enable for a minimal-invasive measurement of the total pressure difference Δp. With the measured electrical power P_{el} and an approximatively constant efficiency factor η_P the following formula can be used to measure $\dot{V}_0$:

$$\eta_P P_{el} = \Delta p \cdot \dot{V}_0$$

A controller based on a linear plant model would then supply a cascaded dedicated non-linear pump controller with the desired reference value for the total mass flow rate. It is a matter to investigate, to which extent this will improve the closed-loop behaviour, since the hydraulic system's transient behaviour is relatively fast compared to the valve action. On the other hand, the neglection of the pump's mechanical inertia might turn out to be inadmissible in real application. In this case, this approach yields a feasible solution.

$\mathcal{H}_\infty$ Mixed Sensitivity Control

$\mathcal{H}_\infty$ mixed sensitivity control is an optimal control approach, which might provide an alternative to the $\mathcal{H}_2$ norm based LQG controller synthesis. In mixed sensitivity design weighting transfer functions are applied to the fictitious outputs of the generalised plant to shape the respective transfer functions

$$\begin{aligned} \boldsymbol{S}(s) &= (\boldsymbol{I} + \boldsymbol{GK}(s))^{-1} & \quad \boldsymbol{r} \rightarrow \boldsymbol{e} \text{ (Sensitivity)} \\ \boldsymbol{T}(s) &= \boldsymbol{I} - \boldsymbol{S}(s) & \quad \boldsymbol{r} \rightarrow \boldsymbol{y} \text{ (Complementary sensitivity)} \\ \boldsymbol{KS}(s) &= \boldsymbol{K}(s)(\boldsymbol{I} + \boldsymbol{GK}(s))^{-1} & \quad \boldsymbol{r} \rightarrow \boldsymbol{u} \text{ (Control sensitivity)} \end{aligned}$$

Shaping the sensitivity allows to take influence on a number of closed-loop properties implied by the sensitivity transfer functions $\boldsymbol{S}(s), \boldsymbol{T}(s)$ and $\boldsymbol{KS}(s)$. For example, integral control can be enforced by requiring the sensitivity function $\boldsymbol{S}(s)$ to have a slope of +20dB/dec at low frequencies and the control action can be kept low by demanding a certain bandwidth and maximum gain from the control sensitivity $\boldsymbol{KS}(s)$. Implementing mixed sensitivity $\mathcal{H}_\infty$ control is already fully automated within *MATLAB* and only requires the designer to define appropriate weights. For this, some expertise is needed, though, but a good explanation and a design example can be found in [22, p.99]. An additional benefit of $\mathcal{H}_\infty$ optimal controllers is their relatively good robustness property.

μ-Synthesis Robust Control

As has been shown in section 3.6.3, even seemingly simple robustness properties can be unable to be achieved by the restrictive assumption imposed by the small gain theorem. In the words of [24], the approach covered in this thesis "covers up" the multiple "sources" of uncertainties with a "large, arbitrarily more conservative perturbation". The use of the structured singular value μ promises the treatment of "individual" uncertainties right "where they occur", thus reducing the conservatism in controller design. It is also possible to ensure robust performance, this way [2, p.71]. μ-synthesis, however, generally leads to controllers of infeasibly high order (take a 20^{th} order controller for a 4^{th} order plant as an example from [2, p.140]). Thus, controller order reduction is necessary prior to implementation. The robustness capabilities may deteriorate with reduced controller order, though.

Linear Parameter-Varying Control

The modelling of the thermodynamic subsystem in section 2.4 as a linear parameter-varying (LPV) system already hints at the possibility to design a LPV gain scheduling controller. Still the problem of measuring certain parameters remains, which is why a plant model for controller synthesis will never include all non-linear effects. The mathematical structure of the LPV model and its scheduling parameters are important for successful controller design. An LPV representation is not unique and different models yield different properties with regard to controller synthesis. [5] develops automated ways of generating LPV representations and provides means to assess LPV models. Not unlike uncertain parameter representations may yield combinations of parameters, the real system can never reach, an LPV model can suffer from overbounding. In [5] a systematic approach called "parameter set mapping" is proposed to improve this. An LPV gain scheduling approach requires more advanced theoretical knowledge than provided in this thesis, though. The following table is taken from [5] and motivates the employment and additional effort of LPV gain scheduling over linearised gain scheduling as presented in this thesis. As an example [5] deals with the successful application of

Aspect	**Linearised Gain Scheduling**	**LPV Gain Scheduling**
External Scheduling Signals	well suited (+)	well suited (+)
Internal Scheduling Signals	less suited (-)	well suited, quasi LPV (+)
Representation of The Plant	local linear models (+)	LPV model (-)
Controller Design	local linear controllers	LPV controller
Controller Interpolation	by hand, tedious (-)	automated (+)
Closed-Loop Stability	weak conditions (-)	guaranteed (+)
Performance Evaluation	by testing (-)	guaranteed (+)
Achievable Performance	high, if applicable (+)	potentially conservative (-)
Controller Structure	arbitrary (+)	full order (-)

Table 5.2.1.: Properties of Linearised and LPV Gain Scheduling (taken from [5])

advanced controllers to injection engines on existing and constrained hardware.

An approach, where robust as well as adaptive controller design is discussed with respect to traditional controllers (like PID, state feedback and output feedback controllers) is presented in [18]. The solution of *linear matrix inequalities* (LMIs) is necessary to obtain such types of controllers. While this is true with $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norm based approaches as well, the latter are already automated in common software packages and do not require the engineer to manually employ LMI solvers.

Fuzzy Control and Fuzzy Gain Scheduling

The ideas proposed in section 4.4 could be carried out and developed further. Neural networks and genetic algorithms yield powerful ways to automate tuning.

However, it might not be advantageous to incorporate a complete fuzzy systems overhead for scheduling purposes. Fuzzy control should be regarded as a heuristic and intuitive way of designing non-linear control laws. Linear interpolation between linear controllers does not necessarily require a full fuzzy approach.

A. Hydraulics Fundamentals

This chapter will provide basic formulas for the calculation of hydraulic networks. Most of the information has been taken from [20].

A.1. Hydraulic Resistances

A.1.1. Resistance Coefficient for Straight Pipes

Pressure loss occurs due to viscous friction of the fluid and is proportional to the square of the flow velocity w. Its reference value is therefore the dynamic pressure p_{dyn} at the entrance:

$$\Delta p = \zeta(w, ...) \underbrace{\frac{\varrho}{2} w^2}_{p_{dyn}} \tag{A.1.1}$$

ϱ denotes the fluid's density, which — for incompressible fluids — remains constant. The resistance coefficient ζ depends on pipe geometry, surface properties and flow conditions. Strictly speaking this proportionality only holds true in case of turbulent flow conditions, where $\zeta(w, ...) = \zeta$ is a true proportionality constant. In case of laminar flow conditions, the resistance coefficient is a function of the flow velocity w, which actually leads to a linear equation in w.

For straight, circular pipe elements the resistance coefficient ζ is a function of the pipe's diameter d and its length l, as well as of the pipe resistance coefficient λ:

$$\zeta = \lambda \cdot \frac{l}{d} \tag{A.1.2}$$

For non-circular pipes the characteristic diameter can be calculated from

$$d = 4\frac{A}{U} \tag{A.1.3}$$

$$\text{with } A \text{ being the cross section area,}$$
$$\text{and } U \text{ as the circumference.} \tag{A.1.4}$$

λ depends on the flow conditions (represented by the Reynolds number Re) and the relative roughness of the pipe (see figure A.1.1).

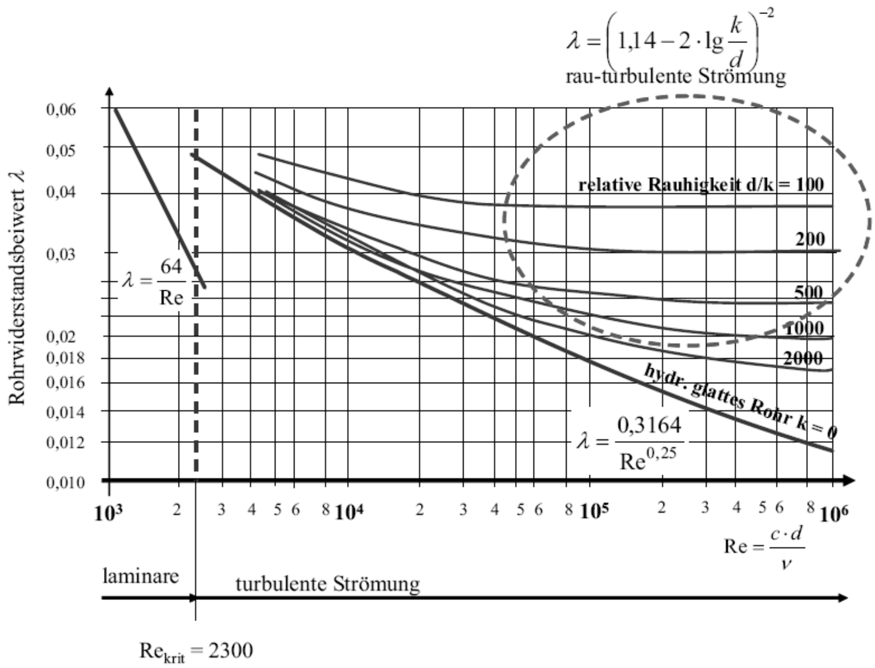


Figure A.1.1.: Pipe Resistance Coefficient (taken from [20, p.49])

In order to obtain a formula to equate pressure losses with mass flow rate $\dot{m}$ multiplied with a hydraulic resistance R, the continuity equation has to be introduced.

$$\dot{m} = \varrho \cdot A \cdot w \tag{A.1.5}$$

This yields:

$$\Delta p = \zeta \cdot \frac{\varrho}{2} \left(\frac{\dot{m}}{\varrho \cdot A} \right)^2 = \underbrace{\left(\frac{\zeta}{2 \varrho A^2} \right)}_{R} \cdot \dot{m}^2 = R \cdot \dot{m}^2 \tag{A.1.6}$$

This holds true for both the turbulent and laminar case. However, in case of laminar flow conditions the non-linearity can be eliminated

Laminar Case

For REYNOLDS numbers below $\mathrm{Re}_{crit} = 2300$ the pipe resistance coefficient becomes

$$\lambda_{lam} = \frac{64}{\mathrm{Re}} \tag{A.1.7}$$

for circular pipes.

Inserting

$$\mathrm{Re} = \frac{w \cdot d}{\nu} \tag{A.1.8}$$

with d as the pipe's characteristic length (i.e. the diameter)
ν as the fluid's kinematic viscosity

yields:

$$\lambda_{lam} = 64 \frac{\nu}{w \cdot d} \tag{A.1.9}$$

The pressure loss can then be calculated as follows:

$$\Delta p = \zeta \cdot \frac{\varrho}{2} \cdot w^2 = \left(\lambda_{lam} \cdot \frac{l}{d}\right) \cdot \frac{\varrho}{2} \cdot w^2 = 64 \frac{\nu}{w \cdot d} \cdot \frac{l}{d} \cdot \frac{\varrho}{2} \cdot w^2 = 64 \frac{\nu \cdot l \cdot \varrho}{2 \cdot d^2} \cdot w \tag{A.1.10}$$

Which leads to:

$$\Delta p = 64 \frac{\nu \cdot l \cdot \varrho}{2 \cdot d^2} \cdot \left(\frac{\dot{m}}{\varrho \cdot A}\right) = \underbrace{\left(\frac{128}{\pi} \frac{\nu \cdot l}{d^4}\right)}_{R_{lam}} \cdot \dot{m} = R_{lam} \cdot \dot{m} \tag{A.1.11}$$

Equivalent Pipe Length

For linearisation purposes, turbulent flow conditions can be assumed laminar. In order to obtain a corresponding laminar hydraulic resistance, an equivalent pipe length l_{eq} can be calculated from given resistance coefficients ζ_{turb}, e.g. for heat exchangers.

$$l_{eq} = \frac{\zeta_{turb}}{\lambda_{lam}} \cdot d \tag{A.1.12}$$

This equivalent length can then be introduced into the formula for R_{lam}.

A.1.2. Series and Parallel Circuits, Current Divider Rule

Kirchhoff's laws apply to hydraulic networks in the same way as they apply to electric networks. The only thing to bear in mind is the squared mass flow rate $\dot{m}$ (its equivalent being the electric current i) in case of turbulent flow conditions, which, for example, hinders the derivation of linear systems of equations to systematically calculate a complete network.

The following formulas provide the most important tools to calculate hydraulic circuitry in both laminar and turbulent case.

Laminar Case

In case of laminar hydraulic networks, multiple hydraulic resistances in series can just be added:

$$R_{lam_{series}} = \sum R_{lam_i} \tag{A.1.13}$$

Parallel connections are calculated by the sum of the reciprocals:

$$\frac{1}{R_{lam_{parallel}}} = \sum \frac{1}{R_{lam_i}} \tag{A.1.14}$$

The current divider rule applies as follows:

$$\frac{\dot{m}_1}{\dot{m}_0} = \frac{R_{lam_2}}{R_{lam_1} + R_{lam_2}} \tag{A.1.15}$$

Turbulent Case

In case of turbulent flow conditions inside hydraulic networks, series connections mathematically still remain a simple summation

$$R_{turb_{series}} = \sum R_{turb_i} \tag{A.1.16}$$

With parallel connections the non-linearity accounts for square roots in the sum of the reciprocals:

$$\frac{1}{\sqrt{R_{turb_{parallel}}}} = \sum \frac{1}{\sqrt{R_{turb_i}}} \tag{A.1.17}$$

The same goes for the current divider rule:

$$\frac{\dot{m}_1}{\dot{m}_0} = \frac{\sqrt{R_{turb_2}}}{\sqrt{R_{turb_1}} + \sqrt{R_{turb_2}}} \tag{A.1.18}$$

Important Remark

Please note, that this only holds true for incompressible fluid flows, as this has been assumed at the beginning of this chapter. For compressible flows the volume flow rate may alter even in simple series circuits.

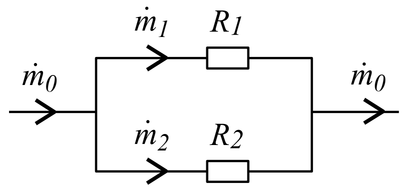


Figure A.1.2.: Current Divider Rule

A.2. Fluid Inertia

Fluids are accelerated by a pressure difference acting along a pipe. By applying Newton's lex II, the force balance yields:

$$\Delta p = \frac{F}{A} = \frac{1}{A} \cdot \frac{d}{dt}(m \cdot w) = \frac{1}{A} \cdot m \cdot \dot{w} \tag{A.2.1}$$

Again, introducing the continuity equation A.1.5 gives a formulation depending on the mass flow rate:

$$\Delta p = \frac{1}{A} \cdot m \cdot \left(\frac{\ddot{m}}{\varrho A}\right) = \frac{1}{A} \cdot (\varrho A l) \cdot \left(\frac{\ddot{m}}{\varrho A}\right) = \underbrace{\left(\frac{l}{A}\right)}_{L} \cdot \ddot{m} = L \cdot \ddot{m} \tag{A.2.2}$$

B. International Standard Atmosphere (ISA) — Basic Facts

The *international standard atmosphere* (or ISA in short) is a set of values and equations to describe the characteristics of the atmosphere over a wide range of altitudes. It has been standardized by the International Organisation For Standardisation (ISO) in 1975 and has since been extended by organisations like the International Civil Aviation Organisation (ICAO) or the U.S. government.

Since the typical cruising level of civil long-range aircrafts like the *Airbus A340* belongs to altitudes of about $11,000\,\mathrm{m}$ in the tropopause, only equations and norms within these boundaries will be covered.

B.1. Temperature Curve

The ISA assumes temperature gradients that are more or less linear. Figure B.1.1 depicts the temperature curves prone to different starting conditions, which represent classifications from extremely hot to extremely cold environments.

The Standard ISA atmosphere's temperature gradient is:

$$\alpha_A = -0.0065\frac{\mathrm{K}}{\mathrm{m}}$$

Therefore a linear dependency of the temperature from altitude with respect to reference values $T_{A_0} = 15\,°\mathrm{C}$ and $h_0 = 0\,\mathrm{m}$ at sea level has the form:

$$T_A(h) = T_{A_0} + \alpha_A \cdot h \tag{B.1.1}$$

B.2. Pressure and Density

Under the assumption of a linear temperature gradient α_A, a pressure balance of an infinitesimal volume element

$$p = \varrho \cdot g \cdot dh + (p + dp)$$

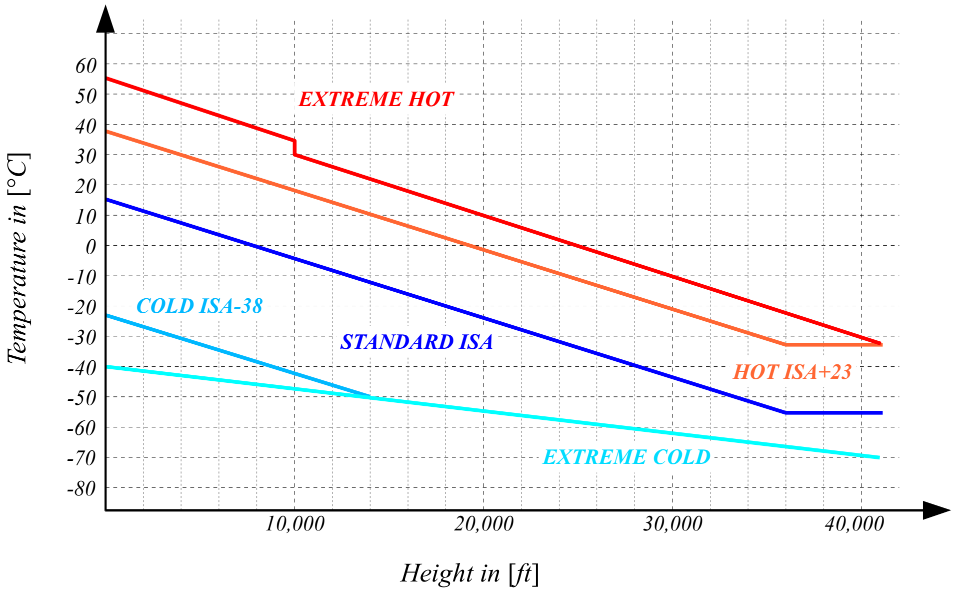


Figure B.1.1.: ISA Temperature Curves versus Height

and integration gives the barometric formula, which is the same as the ISA standard equation for height dependent pressure:

$$p_A(h) = p_{A_0} \cdot \left(1 + \frac{\alpha_A}{T_{A_0}} \cdot h\right)^{-\frac{g}{R_A \cdot \alpha_A}} \tag{B.2.1}$$

with R_A as the air's gas constant

g as the gravity constant

Since the interdependencies of temperature, pressure and density follow the ideal gas law $p = \varrho \cdot R \cdot T$, the density can be computed from:

$$\varrho_A(h) = \varrho_{A_0} \cdot \left(1 + \frac{\alpha_A}{T_{A_0}} \cdot h\right)^{-\frac{g}{R_A \cdot \alpha_A} - 1} \tag{B.2.2}$$

Table B.2 summarises all reference values of the *international standard atmosphere* in the range from $-5,000$ to $11,000$ m altitude.

h_0[m]	α_A[K/m]	T_{A_0}[K]	p_{A_0}[N/m^2]	ϱ_{A_0}[kg/m^3]
0	-0.0065	288.15	101,325	1.225

Table B.2.1.: Standardized ISA Reference Values for Standard ISA

C. Control Theory Addendum

C.1. Basics on Heuristic PID-Controller Design

The following section will recall some elementary basics on heuristic PID-controller design. The Ziegler-Nichols tuning rules will be reproduced first and the requirements to their applicability will be considered. After that, the important notion of integrative control with anti-windup configuration for plants with actuator constraints will be explained.

C.1.1. Ziegler-Nichols Tuning Rules

Heuristic methods include the well known Ziegler-Nichols tuning rules that apply to SISO plants. According to [6] the required properties of the system are either

- to be a stable, approximatedly first order system, or
- to be system that can at least temporarily be operated close to or on the stability margin.

With respect to temperature control, the first criterion is often fulfilled. The standard formulation of a physically realisable PID-controller in continuous time is as follows:

$$C(s) = k_p \left(1 + \frac{1}{T_I s} + \frac{T_D s}{1 + \gamma s}\right) \tag{C.1.1}$$

where $\gamma \ll 1$ is some small constant

Table C.1.1 reproduces the Ziegler-Nichols tuning rules.

C.1.2. Anti-Windup Configuration

For most practical control problems, actuator constraints have to be taken into account. A normal PID-controller might be tuned in a way, that the controller output u reaches saturation, while the control error e remains at a level, that enforces more control action. Figuratively speaking, the integrator is loading up nevertheless, even though the controller output cannot be raised anymore. When the control error comes down again, the plant will clearly overshoot, because the integrator has first to unload all of the control energy he has acquired during the saturation phase.

Prerequisite to the Plant	**Type**	**Controller-Parameters**
Approximation by first order system	P	$k_p = \frac{1 \cdot T}{k_s T_d}$
k_s: static gain	PI	$k_p = \frac{0.9 \cdot T}{k_s T_d}, T_I = 3.33 T_d$
T_d: dead-time, T: time constant	PID	$k_p = \frac{1.2 \cdot T}{k_s T_d}, T_I = 2T_d$, $T_D = 0.5 \cdot T_d$
Marginal stability possible	P	$k_p = 0.5 k_{crit}$
k_{crit}: critical gain	PI	$k_p = 0.45 k_{crit}, T_I = 0.85 T_{crit}$
T_{crit}: critical period	PID	$k_p = 0.6 k_{crit}, T_I = 0.5 T_{crit}, T_D = 0.12 T_{crit}$

Table C.1.1.: ZIEGLER-NICHOLS Tuning Rules ([6, p.434], slightly altered)

The anti-windup configuration (or non-linear-PID as it is sometimes called) as depicted in figure C.1.1 addresses this issue by subtractinging the amount of control action above the level of saturation from the integrators input. The overshoot will sometimes be drastically lower than in normal configuration.

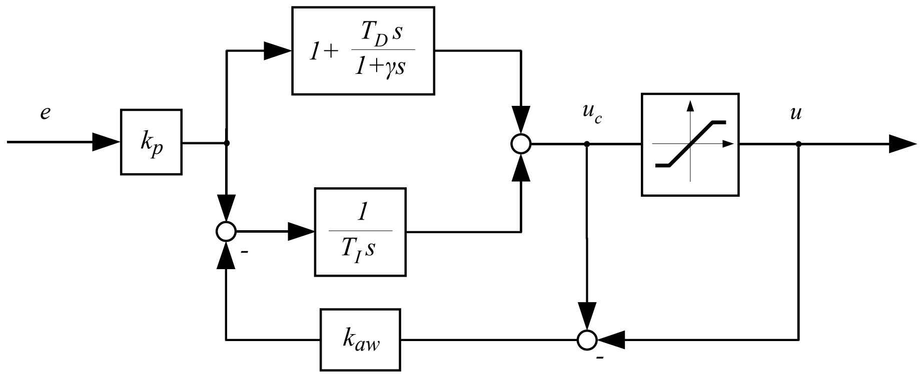


Figure C.1.1.: Anti-Windup Configuration ([21, p.62], slightly altered)

Tuning the gain k_{aw} is often done in an heuristic manner, too. A useful starting value can be obtain by looking at the amount the unsaturated control output is larger than the saturation limit.

C.2. LQG Controller Synthesis — A Special Case $\mathcal{H}_2$ Controller

Linear quadratic gaussian controllers have the ability to filter state and output noise, while minimising a cost functional to reduce control effort and settling time. They combine a Kalman filter, which is a special kind of the Luenberger observer, and optimal state feedback of the estimated plant states. Figure C.2.1 illustrates the general control loop.

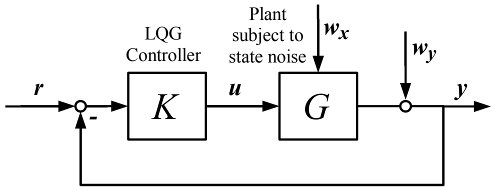


Figure C.2.1.: General LQG Control Loop (based on [22, p.74])

The concept of state feedback and state estimate feedback will be briefly explained. After that, the LQG control problem will be posed as a minimisation problem in a general $\mathcal{H}_2$ controller synthesis framework.

C.2.1. Full Information State Feedback

State feedback aims at altering the plant's system matrix' eigenvalues by applying a gain $\boldsymbol{F}$ to the state variables and then adding this to the plant's input. The control law is therefore:

$$\boldsymbol{u}(t) = \boldsymbol{F}\boldsymbol{x}(t). \tag{C.2.1}$$

It is straightforward to see, that the plant's eigenvalues can be altered this way:

$$\begin{aligned} \boldsymbol{u}(t) &= \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{u_v}(t) \\ \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) && \text{(C.2.2)} \\ \dot{\boldsymbol{x}}(t) &= (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{F})\,\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u_v}(t) && \text{(C.2.3)} \end{aligned}$$

The gain matrix $\boldsymbol{F}$ can be chosen for $\boldsymbol{A} + \boldsymbol{B}\boldsymbol{F}$ to match desired pole locations. An optimal solution $\boldsymbol{F}^*$ minimising the cost functional

$$V = \frac{1}{T}\int_0^T \left(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}\right) dt \tag{C.2.4}$$

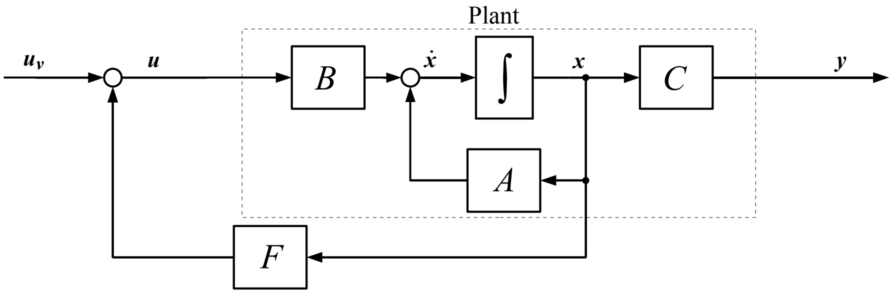


Figure C.2.2.: State Feedback (based on [21, p.7])

can be computed by solving a matrix RICCATI equation, which will not be covered here. Please refer to [22] for full coverage. $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive definite and positive semidefinite matrices and define the amount by which control error or control effort is penalised, respectively. $\boldsymbol{Q}$ is usually chosen as $\boldsymbol{C^T C}$, such that the plant output $\boldsymbol{y}$ is penalised, when differing from zero. It should be noted, that until now and in the coming sections, that without loss of generality, a regulator problem is considered, where it is the aim to drive all states and outputs towards zero.

Necessary assumptions for full state feedback are:

- All states can be measured.
- $(\boldsymbol{A}, \boldsymbol{B})$ is stabilisable, i.e. the following definition holds [23, p.30] (slightly altered):

> **Definition C.2.1 (Stabilisability)** *The system with state space realisation C.2.2 is said to be stabilisable if there exists a state feedback law $\boldsymbol{u}(t) = \boldsymbol{F}\boldsymbol{x}(t)$ such that the resulting system is stable.*

Definition C.2.1 implies, that all unstable poles of the system are *controllable*, which means, that they can be changed by state feedback. [23] provides thorough information about how this can be determined.

C.2.2. Luenberger Observer — State Estimate Feedback

State estimate feedback is needed, if not all states of the plant are available for state feedback, i.e. some of them cannot be directly measured. In this case, the control law becomes:

$$\boldsymbol{u}(t) = \boldsymbol{F}\hat{\boldsymbol{x}}(t) \tag{C.2.5}$$

with $\hat{\boldsymbol{x}}(t)$ denoting the estimated states.

The idea behind state estimation is to compute estimated plant outputs $\hat{\boldsymbol{y}}(t)$ and then feedback the output estimation error to the input of the plant model. $\hat{\boldsymbol{y}}(t) - \boldsymbol{y}(t)$.

The feedback gain $\boldsymbol{L}$ is calculated, such that the state estimation error $\tilde{\boldsymbol{x}}(t) = \hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)$ eventually assumes zero. Figure C.2.3 shows a block diagram of the LUENBERGER observer.

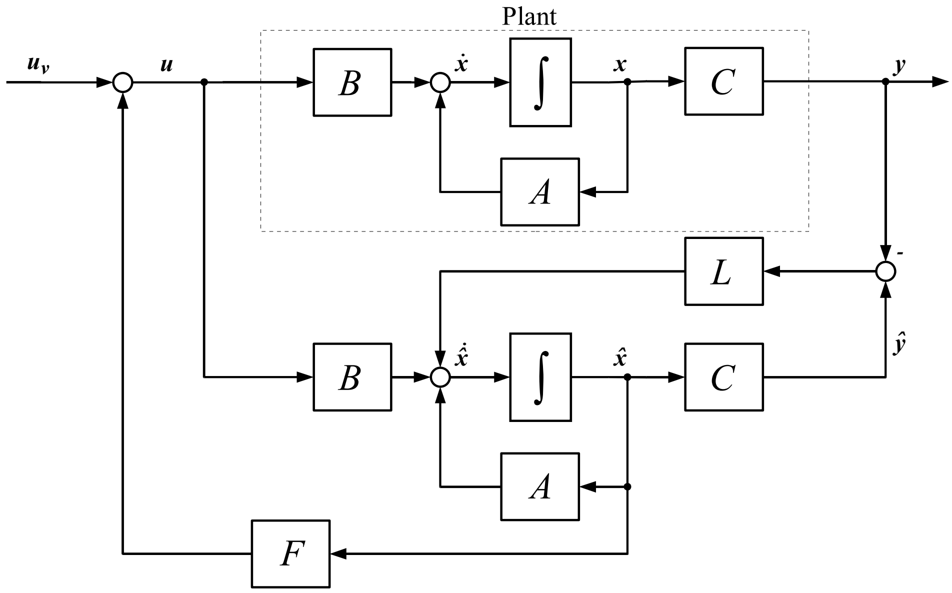


Figure C.2.3.: LUENBERGER Observer Structure with State Estimate Feedback (based on [7, p.336])

Linear quadratic gaussian controllers aim at minimising the cost functional

$$V_{LQG} = \lim_{T\to\infty} E\left[\frac{1}{T}\int_0^T \left(\boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{u}^T R \boldsymbol{u}\right) dt\right], \tag{C.2.6}$$

which is now the expected value of the same cost functional given for full information state feedback.

Necessary assumptions for observer-based state feedback are:

- Some states cannot be measured — otherwise full information state feedback can be used.
- $(\boldsymbol{C}, \boldsymbol{A})$ is detectable, i.e. the following definition holds [23, p.30] (slightly altered):

Definition C.2.2 (Detectability) *The system with state space realisation C.2.2 is said to be detectable if there exists a gain vector $\boldsymbol{L}$ such that all eigenvalues of $\boldsymbol{A} + \boldsymbol{LC}$ are in the left half plane.*

Definition C.2.2 implies, that all unstable poles of the system are *observable*, which means, that they can be changed by state feedback with regard to the dual pair $(\boldsymbol{A}^T, \boldsymbol{C}^T)$

being stabilisable. [23] also provides thorough information about the methods to verify this.

Separation Principle

As has been hinted at in the explanation of C.2.2, designing an observer or full state feedback are dual problems, which can be solved independently, if solved "by hand". Thus the gain matrix $\boldsymbol{L}^T$ can be obtained by considering state feedback with regard to system matrix $\boldsymbol{A}^T$ and input matrix $\boldsymbol{C}^T$.

C.2.3. LQG Control Posed as a $\mathcal{H}_2$ Control Problem

The $\mathcal{H}_2$ Norm

The definition of the $\mathcal{H}_2$ norm for multivariable systems in frequency and time domain is

$$\|\boldsymbol{G}(s)\|_2 = \sqrt{\frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \|\boldsymbol{G}(j\omega)\|_F^2 \; dw} = \sqrt{\int\limits_{0}^{\infty} \|\boldsymbol{g}(t)\|_F^2 \; dt} \tag{C.2.7}$$

$$\text{with } \|\boldsymbol{G}\|_F = \sqrt{trace\left(\boldsymbol{G}^H\boldsymbol{G}\right)} \text{ as the FROBENIUS norm} \tag{C.2.8}$$

$$\text{and } \boldsymbol{g}(t) = \boldsymbol{C}e^{\boldsymbol{A}t}\boldsymbol{B} \text{ as the impulse response to a} \tag{C.2.9}$$

corresponding state space realisation.

Since the eigenvalues of $\boldsymbol{G}^H\boldsymbol{G}$ are the squares of the singular values $\sigma(\boldsymbol{G})$ of $\boldsymbol{G}$ [22, p.84], the $\mathcal{H}_2$ norm is the integral over the sum of squared singular values:

$$\|\boldsymbol{G}(s)\|_2 = \sqrt{\frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \sum_{k=1}^{n} \sigma_k^2\left(\boldsymbol{G}(j\omega)\right) \; dw} \tag{C.2.10}$$

, where n is the system order.

For the adaption to the LQG problem, the $\mathcal{H}_2$ norm needs interpretation. With regard to a system input $\boldsymbol{w}(t)$, being a vector of white noise, which has a DIRAC shaped autocorrelation function

$$E\left[\boldsymbol{w}(t)\boldsymbol{w}^T(t+\tau)\right] = \delta(t)\boldsymbol{I}$$

the $\mathcal{H}_2$ norm may be understood as the *root mean square* of the output signal $\boldsymbol{z}(t)$:

$$\|\boldsymbol{G}(s)\|_2 = \|\boldsymbol{z}(t)\|_{rms} = \lim_{T\to\infty}\sqrt{\frac{1}{2T}\int\limits_{-T}^{T} \|\boldsymbol{z}(\tau)\|^2 \; d\tau},$$

thus the LQG problem can again be formulated as the minimisation of the expected value of a cost functional, as in equation C.2.6.

The Generalised Plant Model

A generalised plant in state space form follows the nomenclature

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_w\boldsymbol{w} + \boldsymbol{B}_u\boldsymbol{u} \\
\boldsymbol{z} &= \boldsymbol{C}_z\boldsymbol{x} + \boldsymbol{D}_{zw}\boldsymbol{w} + \boldsymbol{D}_{zu}\boldsymbol{u} \\
\boldsymbol{v} &= \boldsymbol{C}_v\boldsymbol{x} + \boldsymbol{D}_{vw}\boldsymbol{w} + \boldsymbol{D}_{vu}\boldsymbol{u}
\end{aligned}$$

where $\boldsymbol{w}$ comprises the fictitious inputs
and $\boldsymbol{z}$ contains the fictitious outputs

The following notation is also common and will be employed.

$$\boldsymbol{P}(s) = \left[\begin{array}{c|cc} \boldsymbol{A} & \boldsymbol{B}_w & \boldsymbol{B}_u \\ \hline \boldsymbol{C}_z & \boldsymbol{D}_{zw} & \boldsymbol{D}_{zu} \\ \boldsymbol{C}_v & \boldsymbol{D}_{vw} & \boldsymbol{D}_{vu} \end{array}\right]$$

Figure C.2.4 shows the depiction of generalised control loops common to control theory literature.

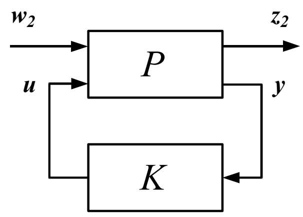


Figure C.2.4.: Common Illustration of Generalised Control Loop

Generalised Plant for LQG Controller Synthesis

To accurately reflect the LQG control problem in generalised plant notation, the fictitious inputs and outputs have to be interpreted accordingly. The input $\boldsymbol{w_2}$ represents noise disturbances and can be subdivided into

$$\boldsymbol{w_2} = \begin{pmatrix} \boldsymbol{w_2^1} \\ \boldsymbol{w_2^1} \end{pmatrix} \begin{array}{l} \text{state noise} \\ \text{output noise} \end{array}$$

while the output $\boldsymbol{z_2}$ can be equivalently described as

$$\boldsymbol{z_2} = \begin{pmatrix} \boldsymbol{z_2^1} \\ \boldsymbol{z_2^1} \end{pmatrix} \begin{array}{l} \text{weighted control error} \\ \text{weighted control effort} \end{array}$$

when considering a regulator problem (reference tracking with $\boldsymbol{r} = 0$). Figure C.2.5 illustrates the generalised plant as a block diagram. The resulting generalised plant

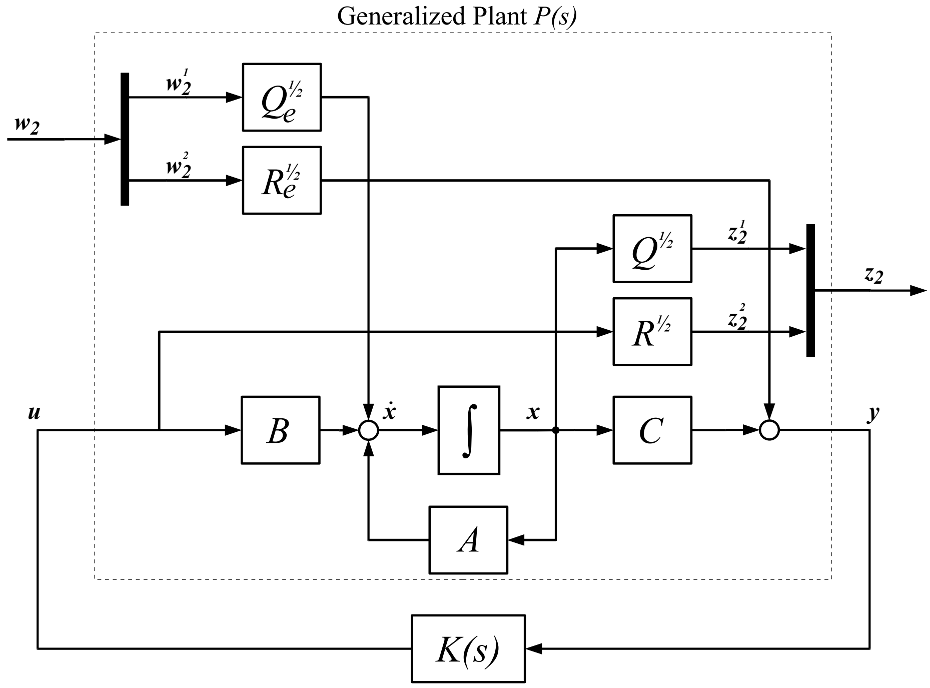


Figure C.2.5.: Generalised Plant Configuration of LQG Problem [22, p.76] (slight modifications)

representation is

$$\boldsymbol{P}(s) = \left[\begin{array}{c|cc} \boldsymbol{A} & \begin{bmatrix}\boldsymbol{Q_e}^{1/2} & \boldsymbol{0}\end{bmatrix} & \boldsymbol{B} \\ \hline \begin{bmatrix}\boldsymbol{Q}^{1/2} \\ \boldsymbol{0}\end{bmatrix} & \boldsymbol{0} & \begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{R}^{1/2}\end{bmatrix} \\ -\boldsymbol{C} & \begin{bmatrix}\boldsymbol{0} & \boldsymbol{R_e}^{1/2}\end{bmatrix} & \boldsymbol{0} \end{array}\right], \tag{C.2.11}$$

with $\boldsymbol{0}$ denoting zero matrices of appropriate dimensions.

Tuning

From C.2.5 the influence of the weighting matrices $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{Q_e}$ and $\boldsymbol{R_e}$ can be inferred. Usually $\boldsymbol{Q}$ and $\boldsymbol{Q_e}$ are chosen as

$$\begin{aligned} \boldsymbol{Q} &= \gamma \cdot \boldsymbol{C}^T\boldsymbol{C} \quad \text{as to reflect penalisation of the output } \boldsymbol{y} \\ \boldsymbol{Q_e} &= \gamma_e \cdot \boldsymbol{B}\boldsymbol{B}^T \quad \text{as to reflect input noise.} \end{aligned}$$

γ and γ_e can still be adjusted to give more weight to the respective input and output, but in general they first remain fixed at 1 and $\boldsymbol{R}$ and $\boldsymbol{R_e}$ are tuned. The following interdependencies should be kept in mind:

- Increasing $\boldsymbol{R}$ leads to smaller control amplitudes. The control will become more effective with regard to energy consumption, but the control error will vanish more slowly.
- Decreasing $\boldsymbol{R}$ will result in a faster regulation to the expense of larger control amplitudes.
- Increasing $\boldsymbol{R_e}$ leads to more filtering, since the output noise is regarded to be of greater intensity. The bandwidth of the filter is lower, which leads to a slowed down observation of the states. The resulting observer poles move to the right, which may eventually result in observer poles right of the plant poles. Then, accurate tracking of the states is no longer possible.
- Decreasing $\boldsymbol{R_e}$ leads to less filtering, accordingly. It is assumed, that the state noise is of greater intensity. More output noise is applied to the controller output, which may lead to excessive control action.

Computation

As mentioned before, the mathematics behind the computation of a certain controller $\boldsymbol{K}(s)$ that minimises the cost functional C.2.6 will be omitted here, since in depth knowledge is not necessary for the application of tools provided by *MATLAB*. The command `hinfsyn` is used for controller synthesis with regard to $\mathcal{H}_2$ based minimisation problems and the command `pck` can be used to manually create the generalised plant. An explanation of the interface can be found in [11].

D. Explicit Equations

D.1. Simulation Model

D.1.1. Hydraulic Subsystem

Pump Dynamics

$$
\begin{aligned}
\Delta p_{P_1}(n_{P_1}, \dot{m}_{P_1}) = H^*_{P_1}\left(\frac{\dot{m}^*_{P_1}}{n^*_{P_1}}\right) \cdot \frac{H_{R_{P_1}} \cdot \varrho_P \cdot g}{n^2_{R_{P_1}}} \cdot {n_{P_1}}^2 = R'_p \cdot \dot{m}^2_{P_1} + L'_p \cdot \ddot{m}_{P_1} \\
\Delta p_{P_2}(n_{P_2}, \dot{m}_{P_2}) = H^*_{P_2}\left(\frac{\dot{m}^*_{P_2}}{n^*_{P_2}}\right) \cdot \frac{H_{R_{P_2}} \cdot \varrho_P \cdot g}{n^2_{R_{P_2}}} \cdot {n_{P_2}}^2 = R'_p \cdot \dot{m}^2_{P_2} + L'_p \cdot \ddot{m}_{P_2} \\
\dot{m}_0 = \dot{m}_{P_1} + \dot{m}_{P_1}
\end{aligned}
\tag{D.1.1}
$$

$$
R'_p = R_{Loads} + R_{HeatExchanger} + R_{MainPipe} \tag{D.1.2}
$$

$$
\text{with} \quad R_{Loads} = \left(\frac{1}{\frac{1}{\sqrt{R_{p_b} + R_{L_1} + R_{v_1}}} + \frac{1}{\sqrt{R_{p_b} + R_{L_2} + R_{v_2}}} + \frac{1}{\sqrt{R_{p_b} + R_{L_3} + R_{v_3}}}} \right)^2 \tag{D.1.3}
$$

$$
R_{HeatExchanger} = \left(\frac{1}{\frac{1}{\sqrt{R_{HE} + R_{v_{HE}}}} + \frac{1}{\sqrt{R_{v_b}}}} \right)^2 \tag{D.1.4}
$$

$$
L'_p = L_p \cdot (1 + v_{HE}) \tag{D.1.5}
$$

Control Valves

$$
\begin{aligned}
\dot{m}_i = \beta_i \cdot \dot{m}_0 \quad , i = 1, 2, 3 \\
\text{with } \beta_i = \frac{\sqrt{\frac{R_{v_1} \cdot R_{v_2} \cdot R_{v_3}}{R_{v_i}}}}{\sqrt{R_{v_1} \cdot R_{v_2}} + \sqrt{R_{v_1} \cdot R_{v_3}} + \sqrt{R_{v_2} \cdot R_{v_3}}}
\end{aligned}
\tag{D.1.6}
$$

$$
\begin{aligned}
\dot{m}_{HE} = \beta_{HE} \cdot \dot{m}_0 \\
\dot{m}_b = \beta_b \cdot \dot{m}_0 \\
\text{with } \beta_{HE} = \frac{\sqrt{R_{v_b}}}{\sqrt{R_{v_b}} + \sqrt{R_{v_{HE}}}} \quad \text{and } \beta_b = \frac{\sqrt{R_{v_{HE}}}}{\sqrt{R_{v_b}} + \sqrt{R_{v_{HE}}}}
\end{aligned}
\tag{D.1.7}
$$

$$
\begin{aligned}
R_v &= K_v(v) \cdot \tfrac{1}{2\varrho_P A_v^2} \\
K_v(v) &= e^{\Pi_3(v)} = e^{p_1 \cdot v^3 + p_2 \cdot v^2 + p_3 \cdot v + p_4}
\end{aligned}
\tag{D.1.8}
$$

$$
\dot{\boldsymbol{x}}_{\boldsymbol{v}} = \boldsymbol{A}_{\boldsymbol{v}} \cdot \boldsymbol{x}_{\boldsymbol{v}} + \boldsymbol{B}_{\boldsymbol{v}} \cdot \boldsymbol{u}_{\boldsymbol{v_{ref}}}
\tag{D.1.9}
$$

$$
\begin{aligned}
\boldsymbol{x}_{\boldsymbol{v}} &= \begin{pmatrix} v_1 & v_2 & v_3 & v_{HE} \end{pmatrix}^T \\
\boldsymbol{A}_{\boldsymbol{v}} &= \begin{pmatrix} -\frac{1}{\tau_v} & & & \\ & -\frac{1}{\tau_v} & & \\ & & -\frac{1}{\tau_v} & \\ & & & -\frac{1}{\tau_v} \end{pmatrix} \qquad \boldsymbol{B}_{\boldsymbol{v}} = \begin{pmatrix} \frac{1}{\tau_v} & & & \\ & \frac{1}{\tau_v} & & \\ & & \frac{1}{\tau_v} & \\ & & & \frac{1}{\tau_v} \end{pmatrix} \\
\boldsymbol{u}_{\boldsymbol{v_{ref}}} &= \begin{pmatrix} v_{1_{ref}} & v_{2_{ref}} & v_{3_{ref}} & v_{HE_{ref}} \end{pmatrix} \\
v_b &= 1 - v_{HE}
\end{aligned}
$$

D.1.2. Ram Air Channel Subsystem

$$
\dot{m}_R = \begin{cases} \dot{m}_{F_1} + \dot{m}_{F_2} & \text{, on ground level} \\ \dot{m}_C & \text{, during flight.} \end{cases}
\tag{D.1.10}
$$

$$
\dot{m}_C = \text{const.}
\tag{D.1.11}
$$

$$
\begin{aligned}
\Delta p_{F_1}(n_{F_1}, \dot{m}_{F_1}) &= H^*_{F_1}\left(\frac{\dot{m}^*_{F_1}}{n^*_{F_1}}\right) \cdot \frac{H_{R_{F_1}} \cdot \varrho_A(h) \cdot g}{n^2_{R_{F_1}}} \cdot {n_{F_1}}^2 = R_R(\varrho_A(h)) \cdot \dot{m}^2_{F_1} + L_R \cdot \ddot{m}_{F_1} \\
\Delta p_{F_2}(n_{F_2}, \dot{m}_{F_2}) &= H^*_{F_2}\left(\frac{\dot{m}^*_{F_2}}{n^*_{F_2}}\right) \cdot \frac{H_{R_{F_2}} \cdot \varrho_A(h) \cdot g}{n^2_{R_{F_2}}} \cdot {n_{F_2}}^2 = R_R(\varrho_A(h)) \cdot \dot{m}^2_{F_2} + L_R \cdot \ddot{m}_{F_2}
\end{aligned}
\tag{D.1.12}
$$

D.1.3. Thermodynamic Subsystem

Thermodynamic Subsystem Block 1

$$
\begin{aligned}
\boldsymbol{M}_{\boldsymbol{T^1}} \cdot \dot{\boldsymbol{x}}_{\boldsymbol{T^1}} &= \tilde{\boldsymbol{A}}_{\boldsymbol{T^1}}(\boldsymbol{\theta_m}) \cdot \boldsymbol{x}_{\boldsymbol{T^1}} + \tilde{\boldsymbol{D}}_{\boldsymbol{T^1}} \cdot \boldsymbol{v}_{\boldsymbol{T^1}} \\
\dot{\boldsymbol{x}}_{\boldsymbol{T^1}} &= \underbrace{{\boldsymbol{M}_{\boldsymbol{T^1}}}^{-1} \tilde{\boldsymbol{A}}_{\boldsymbol{T^1}}(\boldsymbol{\theta_m})}_{\boldsymbol{A}_{\boldsymbol{T^1}}(\boldsymbol{\theta_m})} \cdot \boldsymbol{x}_{\boldsymbol{T^1}} + \underbrace{{\boldsymbol{M}_{\boldsymbol{T^1}}}^{-1} \tilde{\boldsymbol{D}}_{\boldsymbol{T^1}}}_{\boldsymbol{D}_{\boldsymbol{T^1}}} \cdot \boldsymbol{v}_{\boldsymbol{T^1}}
\end{aligned}
\tag{D.1.13}
$$

$$
\boldsymbol{A}_{\boldsymbol{T^1}}(\boldsymbol{\theta_m}) = \boldsymbol{A}^{\boldsymbol{const}}_{\boldsymbol{T^1}} + \boldsymbol{A}^{\boldsymbol{0}}_{\boldsymbol{T^1}} \cdot \dot{m}_0 + \boldsymbol{A}^{\boldsymbol{1}}_{\boldsymbol{T^1}} \cdot \dot{m}_1 + \boldsymbol{A}^{\boldsymbol{2}}_{\boldsymbol{T^1}} \cdot \dot{m}_2
\tag{D.1.14}
$$

$$\boldsymbol{x_{T^1}} = \boldsymbol{y_{T^1}} = \begin{pmatrix} T_0^d & T_1 & T_2 & T_3 & T_4 \end{pmatrix}^T$$

$$\boldsymbol{M_{T^1}} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & M_{L_1}c_{v_P} & \ddots & & \vdots \\ \vdots & \ddots & M_{L_2}c_{v_P} & \ddots & \vdots \\ \vdots & & \ddots & M_{L_3}c_{v_P} & 0 \\ 0 & \cdots & \cdots & 0 & M_{J_2}c_{v_P} \end{pmatrix}$$

$$\boldsymbol{\tilde{A}^{const}_{T^1}} = \begin{pmatrix} \frac{1}{\tau_d} & 0 & \cdots & \cdots & 0 \\ 0 & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & \cdots & \cdots & 0 \end{pmatrix} \qquad \boldsymbol{\tilde{A}^{0}_{T^1}} = \begin{pmatrix} 0 & \cdots & & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & & \cdots & 0 \\ c_{p_P} & 0 & 0 & -c_{p_P} & 0 \\ 0 & 0 & 0 & c_{p_P} & c_{p_P} \end{pmatrix}$$

$$\boldsymbol{\tilde{A}^{1}_{T^1}} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ c_{p_P} & -c_{p_P} & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ -c_{p_P} & 0 & 0 & c_{p_P} & 0 \\ 0 & c_{p_P} & 0 & -c_{p_P} & 0 \end{pmatrix} \qquad \boldsymbol{\tilde{A}^{2}_{T^1}} = \begin{pmatrix} 0 & \cdots & & \cdots & 0 \\ 0 & \cdots & & \cdots & 0 \\ c_{p_P} & 0 & -c_{p_P} & 0 & 0 \\ -c_{p_P} & 0 & 0 & c_{p_P} & 0 \\ 0 & 0 & c_{p_P} & -c_{p_P} & 0 \end{pmatrix}$$

$$\boldsymbol{\tilde{D}_{T^1}} = \begin{pmatrix} \frac{1}{\tau_d} & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & 0 \\ \vdots & & \ddots & 1 \\ 0 & \cdots & \cdots & 0 \end{pmatrix} \qquad \boldsymbol{v_{T^1}} = \begin{pmatrix} T^d_{0,2} \\ \dot{Q}_{L_1} \\ \dot{Q}_{L_2} \\ \dot{Q}_{L_3} \end{pmatrix}$$

Thermodynamic Subsystem Block 2

$$\begin{aligned} \boldsymbol{M_{T^2}} \cdot \boldsymbol{\dot{x}_{T^2}} &= \boldsymbol{\tilde{A}_{T^2}}(\boldsymbol{\theta_m}) \cdot \boldsymbol{x_{T^2}} + \boldsymbol{\tilde{D}_{T^2}} \cdot \boldsymbol{v_{T^2}} \\ \boldsymbol{\dot{x}_{T^2}} &= \underbrace{\boldsymbol{M_{T^2}}^{-1}\boldsymbol{\tilde{A}_{T^2}}(\boldsymbol{\theta_m})}_{\boldsymbol{A_{T^2}}(\boldsymbol{\theta_m})} \cdot \boldsymbol{x_{T^2}} + \underbrace{\boldsymbol{M_{T^2}}^{-1}\boldsymbol{\tilde{D}_{T^2}}}_{\boldsymbol{D_{T^2}}} \cdot \boldsymbol{v_{T^2}} \end{aligned} \tag{D.1.15}$$

$$\boldsymbol{A_{T^2}}(\boldsymbol{\theta_m}) = \boldsymbol{A^{const}_{T^2}} + \boldsymbol{A^{0}_{T^2}} \cdot \dot{m}_0 + \boldsymbol{A^{HE}_{T^2}} \cdot \dot{m}_{HE} + \boldsymbol{A^{R}_{T^2}} \cdot \dot{m}_R + \boldsymbol{A^{kA}_{T^2}} \cdot kA(\dot{m}_0, \dot{m}_R) \tag{D.1.16}$$

$$\boldsymbol{x_{T^2}} = \boldsymbol{y_{T^2}} = \begin{pmatrix} T_4^d & T_5 & T_{HE} & T_{R_0} & T_{R_1} & T_0 \end{pmatrix}^T$$

$$kA(\dot{m}_0, \dot{m}_R) = \frac{1}{\frac{1}{H_P \cdot \dot{m}_0^{0.8}} + \frac{1}{H_A \cdot \dot{m}_R^{0.8}}} \tag{D.1.17}$$

$$\text{with } H_P = \frac{\lambda_P}{d_p}\left(0.023\mathrm{Pr}_P^{0.3} \cdot \left(\frac{d_p}{\varrho_P \cdot A_p \cdot \nu_P}\right)\right)^{0.8} \cdot A_p$$

$$H_A = \frac{\lambda_A}{d_R}\left(0.023\mathrm{Pr}_A^{0.4} \cdot \left(\frac{d_R}{\varrho_A \cdot A_R \cdot \nu_A}\right)\right)^{0.8} \cdot A_R$$

$$\boldsymbol{M_{T^2}} = \begin{pmatrix} 1 & 0 & \cdots & & \cdots & 0 \\ 0 & M_P c_{v_P} & \ddots & & & \vdots \\ \vdots & \ddots & M_{HE} c_{v_P} & & & \\ \vdots & & & 1 & \ddots & \vdots \\ \vdots & & & \ddots & M_{HE_R} c_{v_A} & 0 \\ 0 & \cdots & & \cdots & 0 & M_{J_1} c_{v_P} \end{pmatrix}$$

$$\boldsymbol{\tilde{A}^{const}_{T^2}} = \begin{pmatrix} -\frac{1}{\tau_d} & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & \vdots & \vdots \\ \vdots & & \vdots & 0 & & \\ & & & \frac{1}{\tau_T} & & \\ \vdots & & \vdots & 0 & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 \end{pmatrix} \qquad \boldsymbol{\tilde{A}^{0}_{T^2}} = \begin{pmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 \\ c_{p_P} & -c_{p_P} & \vdots & & & \vdots \\ 0 & 0 & & & & \\ \vdots & \vdots & \vdots & & & \vdots \\ 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & c_{p_P} & 0 & 0 & 0 & -c_{p_P} \end{pmatrix}$$

$$\boldsymbol{\tilde{A}^{HE}_{T^2}} = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & 0 & 0 & \vdots & & \vdots \\ & c_{p_P} & -c_{p_P} & & & \\ & 0 & 0 & & & \\ \vdots & 0 & 0 & \vdots & & \vdots \\ 0 & -c_{p_P} & c_{p_P} & 0 & \cdots & 0 \end{pmatrix} \qquad \boldsymbol{\tilde{A}^{R}_{T^2}} = \begin{pmatrix} 0 & \cdots & & & \cdots & 0 \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ 0 & \cdots & & & \cdots & 0 \\ 0 & 0 & 0 & c_{p_A} & -c_{p_A} & 0 \\ 0 & \cdots & & & \cdots & 0 \end{pmatrix}$$

$$\boldsymbol{\tilde{A}^{kA}_{T^2}} = \begin{pmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & 0 & \cdots & \cdots & 0 \\ & & -QC & 0 & QC & \vdots \\ & & 0 & 0 & 0 & \\ \vdots & \vdots & QC_R & 0 & -QC_R & \vdots \\ 0 & 0 & \cdots & & \cdots & 0 \end{pmatrix}$$

$$\boldsymbol{\tilde{D}_{T^2}} = \begin{pmatrix} \frac{1}{\tau_d} & 0 & \cdots & 0 \\ 0 & 1-\eta_P & 1-\eta_P & 0 \\ 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & \frac{1}{\tau_T} \\ \vdots & & \vdots & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix} \qquad \boldsymbol{v_{T^2}} = \begin{pmatrix} T^d_{4,1} \\ P_{el}^{P_1} \\ P_{el}^{P_2} \\ T_{R_0,meas.} \end{pmatrix}$$

Dead-Times between Block 1 and 2

$$t_d^{1,2} = l_p^{1,2} / \left(\frac{\dot{m}_0}{\varrho_P \cdot A_p^{1,2}} \right) \tag{D.1.18}$$

with $l_p^{1,2}$ and $A_p^{1,2}$ as the respective pipe's length and cross section area.

D.1.4. Environmental Parameters

$$p_R(h,c) = p_A(h) \cdot \left(1 + \frac{\kappa - 1}{2} \cdot Mach^2 \right)^{\frac{\kappa}{\kappa-1}} \tag{D.1.19}$$

with $p_A(h)$ as the ambient pressure at height h,

$Mach = \dfrac{c}{a}$ as the aircraft's velocity in fractions of the sonic speed a,

$a = \sqrt{\kappa \cdot R_A \cdot T_A(h)}$ as the temperature dependent sonic speed,

$\kappa = 1.4$ as the isentropic expansion factor of air.

$$T_{R_0}(h,c) = T_A(h) \cdot \left(1 + RF \cdot \frac{\kappa - 1}{2} \cdot Mach^2 \right) \tag{D.1.20}$$

with $RF \approx 0.9$ as an empirical recovery factor

$$\varrho_R(h,c) = \frac{1}{R_A} \cdot \frac{p_R}{T_{R_0}}(h,c) \tag{D.1.21}$$

D.2. Linearised Plant Models

D.2.1. LQG Controller Synthesis

Block T^1

$$\begin{aligned} \boldsymbol{M_{T^1}} \cdot \boldsymbol{\Delta \dot{x}_{T^1}} &= \boldsymbol{\tilde{A}_{T^1}} \cdot \boldsymbol{\Delta x_{T^1}} + \boldsymbol{\tilde{B}_{T^1}} \cdot \boldsymbol{\Delta u_{T^1}} + \boldsymbol{\tilde{D}_{T^1}} \cdot \boldsymbol{\Delta d_{T^1}} \\ \boldsymbol{\Delta \dot{x}_{T^1}} &= \underbrace{\boldsymbol{M_{T^1}}^{-1} \boldsymbol{\tilde{A}_{T^1}}}_{\boldsymbol{A_{T^1}}} \cdot \boldsymbol{\Delta x_{T^1}} + \underbrace{\boldsymbol{M_{T^1}}^{-1} \boldsymbol{\tilde{B}_{T^1}}}_{\boldsymbol{B_{T^1}}} \cdot \boldsymbol{\Delta u_{T^1}} + \underbrace{\boldsymbol{M_{T^1}}^{-1} \boldsymbol{\tilde{D}_{T^1}}}_{\boldsymbol{D_{T^1}}} \cdot \boldsymbol{\Delta d_{T^1}} \\ \boldsymbol{\Delta y_{T^1}} &= \boldsymbol{C_{T^1}} \cdot \boldsymbol{\Delta x_{T^1}} \end{aligned} \tag{D.2.1}$$

$$\boldsymbol{x_{T^1}} = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ \dot{m}_0 \\ n_P \\ \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} x_{1_{T^1}} \\ x_{2_{T^1}} \\ x_{3_{T^1}} \\ x_{4_{T^1}} \\ x_{5_{T^1}} \\ x_{6_{T^1}} \\ x_{7_{T^1}} \end{pmatrix} \quad \boldsymbol{u_{T^1}} = \begin{pmatrix} n_P \\ u_{\beta_1} \\ u_{\beta_2} \end{pmatrix} = \begin{pmatrix} u_{1_{T^1}} \\ u_{2_{T^1}} \\ u_{3_{T^1}} \end{pmatrix} \quad \boldsymbol{d_{T^1}} = \begin{pmatrix} Q_{L_1} \\ Q_{L_2} \\ Q_{L_3} \\ T_0^d \end{pmatrix} = \begin{pmatrix} d_{1_{T^1}} \\ d_{2_{T^1}} \\ d_{3_{T^1}} \\ d_{4_{T^1}} \end{pmatrix}$$

$$
\boldsymbol{M_{T^1}} = \begin{pmatrix}
M_{L_1}c_{v_P} & 0 & \cdots & & & \cdots & 0 \\
0 & M_{L_2}c_{v_P} & \ddots & & & & \vdots \\
\vdots & \ddots & M_{L_3}c_{v_P} & & & & \\
 & & & 1 & & & \\
 & & & & 1 & \ddots & \vdots \\
\vdots & & & & \ddots & 1 & 0 \\
0 & \cdots & & & \cdots & 0 & 1
\end{pmatrix}
$$

$$
\boldsymbol{\tilde{A}_{T^1}} = \begin{pmatrix}
-\beta_1^0\dot{m}_0^0 c_{p_P} & 0 & 0 & (T_0^0-T_1^0)\beta_1^0 c_{p_P} & 0 & (T_0^0-T_1^0)\dot{m}_0^0 c_{p_P} & 0 \\
0 & -\beta_2^0\dot{m}_0^0 c_{p_P} & 0 & (T_0^0-T_2^0)\beta_2^0 c_{p_P} & 0 & 0 & (T_0^0-T_2^0)\dot{m}_0^0 c_{p_P} \\
\vdots & \ddots & -\beta_3^0\dot{m}_0^0 c_{p_P} & (T_0^0-T_3^0)\beta_3^0 c_{p_P} & 0 & (T_3^0-T_0^0)\dot{m}_0^0 c_{p_P} & (T_3^0-T_0^0)\dot{m}_0^0 c_{p_P} \\
 & & & -2\frac{R_p^{,0}}{L_p^{,0}}\dot{m}_0^0 & 2H_P^{*0}\frac{H_{R_P}\varrho_P g}{n_{R_P}^2}n_P^0 & 0 & 0 \\
 & & & & -\frac{1}{\tau_{n_P}} & 0 & 0 \\
\vdots & & & & \ddots & -\frac{1}{\tau_v} & 0 \\
0 & \cdots & & & \cdots & 0 & -\frac{1}{\tau_v}
\end{pmatrix}
$$

$$
\boldsymbol{\tilde{B}_{T^1}} = \begin{pmatrix}
0 & \cdots & 0 \\
\vdots & & \vdots \\
\vdots & & \vdots \\
0 & \cdots & 0 \\
\frac{n_P^{scale}}{\tau_{n_P}} & 0 & 0 \\
0 & \frac{1}{\tau_v} & 0 \\
0 & \cdots & \frac{1}{\tau_v}
\end{pmatrix}
\qquad
\boldsymbol{\tilde{D}_{T^1}} = \begin{pmatrix}
1 & 0 & 0 & \beta_1^0\dot{m}_0^0 c_{p_P} \\
0 & 1 & 0 & \beta_2^0\dot{m}_0^0 c_{p_P} \\
\vdots & \ddots & 1 & \beta_3^0\dot{m}_0^0 c_{p_P} \\
 & & \ddots & 0 \\
 & & & 0 \\
\vdots & & & \vdots \\
0 & \cdots & \cdots & 0
\end{pmatrix}
$$

$$
\boldsymbol{C_{T^1}} = \begin{pmatrix}
T^{scale} & 0 & \cdots & \cdots & & \cdots & 0 \\
0 & T^{scale} & \ddots & & & & \vdots \\
0 & 0 & T^{scale} & 0 & \cdots & \cdots & 0
\end{pmatrix}
$$

Block T^2

$$
\begin{aligned}
\boldsymbol{M_{T^2}} \cdot \boldsymbol{\Delta\dot{x}_{T^2}} &= \boldsymbol{\tilde{A}_{T^2}} \cdot \boldsymbol{\Delta x_{T^2}} + \boldsymbol{\tilde{B}_{T^2}} \cdot \boldsymbol{\Delta u_{T^2}} + \boldsymbol{\tilde{D}_{T^2}} \cdot \boldsymbol{\Delta d_{T^2}} \\
\boldsymbol{\Delta\dot{x}_{T^2}} &= \underbrace{\boldsymbol{M_{T^2}}^{-1}\boldsymbol{\tilde{A}_{T^2}}}_{\boldsymbol{A_{T^2}}} \cdot \boldsymbol{\Delta x_{T^2}} + \underbrace{\boldsymbol{M_{T^2}}^{-1}\boldsymbol{\tilde{B}_{T^2}}}_{\boldsymbol{B_{T^2}}} \cdot \boldsymbol{\Delta u_{T^2}} + \underbrace{\boldsymbol{M_{T^2}}^{-1}\boldsymbol{\tilde{D}_{T^2}}}_{\boldsymbol{D_{T^2}}} \cdot \boldsymbol{\Delta d_{T^2}} \\
\boldsymbol{\Delta y_{T^2}} &= \boldsymbol{C_{T^2}} \cdot \boldsymbol{\Delta x_{T^2}}
\end{aligned}
\tag{D.2.2}
$$

$$
\boldsymbol{x_{T^2}} = \begin{pmatrix} T_0 \\ T_{HE} \\ T_{R_1} \\ \beta_{HE} \end{pmatrix} = \begin{pmatrix} x_{1_{T^2}} \\ x_{2_{T^2}} \\ x_{3_{T^2}} \\ x_{4_{T^2}} \end{pmatrix} \qquad \boldsymbol{u_{T^2}} = \begin{pmatrix} u_{\beta_{HE}} \end{pmatrix} = \begin{pmatrix} u_{1_{T^2}} \end{pmatrix} \qquad \boldsymbol{d_{T^2}} = \begin{pmatrix} T_{R_0} \\ \dot{m}_R \end{pmatrix} = \begin{pmatrix} d_{1_{T^2}} \\ d_{2_{T^2}} \end{pmatrix}
$$

$$\boldsymbol{M_{T^2}} = \begin{pmatrix} M_{J_1} c_{v_P} & 0 & \cdots & 0 \\ 0 & M_{HE} c_{v_P} & \ddots & \vdots \\ 0 & \ddots & M_{HE_R} c_{v_A} & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

$$\tilde{\boldsymbol{A}}_{\boldsymbol{T^2}} = \begin{pmatrix} -\dot{m}_0^0 c_{p_P} & \beta_{HE}^0 \dot{m}_0^0 c_{p_P} & 0 & -(T_5^0 - T_{HE}^0) c_{p_P} \\ 0 & -\beta_{HE}^0 \dot{m}_0^0 c_{p_P} - QC & QC & (T_5^0 - T_{HE}^0) c_{p_P} \\ 0 & QC_R & -\dot{m}_R^0 c_{p_A} - QC_R & 0 \\ 0 & 0 & & -\frac{1}{\tau_v} \end{pmatrix}$$

$$\tilde{\boldsymbol{B}}_{\boldsymbol{T^2}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\tau_v} \end{pmatrix} \qquad \tilde{\boldsymbol{D}}_{\boldsymbol{T^2}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \dot{m}_R^0 c_{p_A} & T_R^0 c_{p_A} \\ 0 & 0 \end{pmatrix}$$

$$\boldsymbol{C_{T^2}} = \begin{pmatrix} T^{scale} & 0 & 0 & 0 \end{pmatrix}$$

E. Contents of the Accompanying Disc

The CD ROM, which accompanies this thesis, contains the relevant *MATLAB* files created for this work. The following will provide a quick overview of the files and their use. In order to easily run the *Simulink* simulations, it is recommended, that the disc is first copied to a hard drive, for instance to a directory `C:\CoolingSystemControl`. In *MATLAB* the complete directory with all subfolders should be added to the path list by using the `File` → `Set Path...`.

- **.\Introductory Example** This folder contains files used for the design and assessment of different controllers of the mass-spring combination in the introductory example.
  - **\ControllerDesigns.m** Model data and synthesis of LQR, LQG and PID controllers.
  - **\MISOFuzzyInferenceSystem.fis** Fuzzy inference system of the fuzzy controller generated with the *MATLAB* command `fuzzy`.
  - **\SimpleCartAndSpring.mdl** Simulink model simulation framework containing all controllers.
- **.\Non-Linear Simulation** This folder contains files used for the modelling, design and assessment of the different controllers applied to the cooling cycle.
  - **.\Data** Folder, which contains modelling relevant data.
    - **\ball_valve_loss_coefficient_Kv_vs_valve_openingratio.m** Approximation of control valve characteristic curve. Results are hard-coded in `ModelConstants.m`.
    - **\lookup_v_from_beta_ratioHE_b.mat** Lookup table data for determining β_{HE} from v_{HE}. Will be run by `ModelConstants.m`.
    - **\model_constants.m** Definition of simulation model parameters and constants. First file to run.
    - **\model_data.m** System matrices of non-linear model. Run after `ModelConstants.m`.
    - **\normalised_head_vs_normalised_volume_flowrate.m** Approximation of pump characteristic curve. Results are hard-coded in `ModelConstants.m`.
    - **\normalised_cump_curve_ HvsQ.mat** Pump characteristic curve data as exported from *Flowmaster*.

- **v_HE_from_beta_ratioHE_b.m** Generation of lookup table data in `Lookup_v_from_beta_ratioHE_b.mat`.

.**H2 Hinf Controller Framework**\ Folder, which contains decentralised $\mathcal{H}_2$ and $\mathcal{H}_\infty$ controller synthesis data and simulation framework.

- **blockT1_H2Controller_LQG.m** $\mathcal{H}_2$ LQG controller synthesis for block T^1.
- **blockT1_RobustH2HinfController.m** Robust $\mathcal{H}_2\mathcal{H}_\infty$ LQG controller synthesis for block T^1.
- **blockT2_GainAndPhaseMargins.m** Determination of gain and phase margins of LQG controller for block T^2 with respect to uncertain ram air temperature and mass flow rate parameters.
- **blockT2_H2Controller_LQG.m** $\mathcal{H}_2$ LQG controller synthesis for block T^2.
- **blockT2_RobustH2HinfController.m** Robust $\mathcal{H}_2/\mathcal{H}_\infty$ LQG controller synthesis for block T^1.
- **sim_GeneralH2HinfControllerFramework.mdl** *Simulink* controlled loop framework model.
- **uncertainPlantModels.m** Definition of uncertain linearised plant models for controller synthesis. Needs to be run first.

.**H2 LQG Gain Scheduling**\ Folder, which contains decentralised gain scheduled LQG controller synthesis data and simulation framework.

- **linearisedPlantModelsGainScheduling.m** Definition of linearised plant models for controller synthesis. Needs to be run first.
- **sim_GeneralH2HinfGainSchedulingControllerFramework.mdl** *Simulink* gain scheduled controlled loop model.

.**PID Control**\ Folder containing PID controller tuning data and simulation framework.

- **PIDControl.m** Definition of tuning parameters and evaluation plot script.
- **PIDControl.mdl** *Simulink* PID controlled loop model.

controllerBenchmarking.m Script for plotting simulation results of controlled loop.

environmentalParameters.mdl Plant subsystem used for simulating environmental parameters.

evaluation.m Script for plotting various simulation results.

hydraulicSubsystem.mdl Plant subsystem used for simulating the hydraulic subsystem.

plant.mdl Open loop *Simulink* model of the cooling cycle plant. References all subsystem `.mdl` files.

ramAirSubsystem.mdl Plant subsystem used for simulating the ram air channel subsystem.

thermodynamicSubsystem.mdl Plant subsystem used for simulating the thermodynamic subsystem.

List of Figures

Bibliography

[1] Flowmaster Ltd., Flowmaster Limited. The Maltings, Pury Hill Business Park, Alderton Road, Towcester, Northants, NN12 7TB, UK. *Flowmaster Reference - SS and FS Rotodynamic Pumps.*

[2] D.-W. Gu, P. Hr. Petkov, and M. M. Konstantinov. *Robust Control Design with MATLAB.* Springer-Verlag London Limited, 2005.

[3] International Organization For Standardization, http://www.iso.org. *International Standard Atmosphere as defined in ISO 2533:1975.*

[4] Rolf Isermann. *Mechatronische Systeme - Grundlagen.* Springer Berlin Heidelberg New York, 2. vollständig neu bearbeitete auflage edition, 2008.

[5] Andreas Kwiatkowski. *LPV Modelling and Application of LPV Controllers to SI Engines.* PhD thesis, Technische Universität Hamburg-Harburg, 2007.

[6] Jan Lunze. *Regelungstechnik 1 - Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen.* Springer Berlin Heidelberg New York, 6., neu bearbeitete auflage edition, 2007.

[7] Jan Lunze. *Regelungstechnik 2 - Mehrgrößensysteme, Digitale Regelung.* Springer Berlin Heidelberg New York, 5., neu bearbeitete auflage edition, 2008.

[8] Uwe Mackenroth. *Robust Control Systems - Theory And Case Studies.* Springer Berlin Heidelberg New York, 2003.

[9] Mahmoud Massoud. *Engineering Thermofluids, Thermodynamics, Fluid Mechanics, and Heat Transfer.* Springer Berlin Heidelberg New York, 2005.

[10] The Mathworks, http://www.mathworks.com. *MATLAB Control Systems Toolbox 8 - User's Guide.*

[11] The Mathworks, http://www.mathworks.com. *MATLAB Robust Control Toolbox 3 - User's Guide.*

[12] Kai Michels, Frank Klawonn, Rudolf Kruse, and Andreas Nürnberger. *Fuzzy Control - Fundamentals, Stability and Design of Fuzzy Controllers.* Springer-Verlag Berlin Heidelberg, 2006.

[13] Nanyang Technological University, School of Electrical and Electronic Engineering. *Modelling and Dynamic Feedback Linearisation of a Heat Exchanger Model*, Nanyang Avenue, Singapore 2263, 1994. IEEE.

[14] J.E. Normey-Rico and E.F. Camacho. *Control Of Dead-Time Processes*. Springer-Verlag London Limited, 2007.

[15] Kevin M. Passino and Stephen Yurkovich. *Fuzzy Control*. Addison Wesley Longman, Inc., 1998.

[16] Proc. American Control Conf., Boston, MA. *Next generation of tools for robust control*, 2004.

[17] Gerhard Schmitz. *Technische Thermodynamik II*. Institut für Thermofluiddynamik, skript nur für den vorlesungsgebrauch edition, 2006.

[18] Steffen Sommer. *Regelung linearer parameterveränderlicher (LPV-) Systeme mit Hilfe klassischer Regelungsstrukturen und Anwendung auf nichtlineare Regelstrecken*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2003.

[19] Peter von Böckh. *Wärmeübertragung - Grundlagen und Praxis*. Springer Berlin Heidelberg New York, zweite, bearbeitete auflage edition, 2006.

[20] Holger Watter. *Hydraulik und Pneumatik*. Friedr. Vieweg und Sohn Verlag, 1. auflage edition, 2007.

[21] Herbert Werner. *Control Systems 1 - Lecture Notes*. Institut für Regelungstechnik, ver. november 2007 edition, 2007.

[22] Herbert Werner. *Optimal And Robust Control*. Institut für Regelungstechnik, ver. september 2007 edition, 2007.

[23] Herbert Werner. *Control Systems Theory And Design - Lecture Notes*. Institut für Regelungstechnik, ver. juli 2008 edition, 2008.

[24] Kemin Zhou and John C. Doyle. *Essentials Of Robust Control*. Prentice Hall, 1998.