Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria
PhD program in Information Technology

Modelling and Control of Aircraft Environmental Control Systems

Doctoral Dissertation of:
Alexander Pollok

Coordinator:
Prof. Andrea Bonarini

Supervisor:
Prof. Francesco Casella

Tutor:
Prof. Luca Bascetta

2017 – Cycle XXX

Acknowledgments

> *The inexpressible sense of betrayal inside her was welling and welling, taking over everything, her rising voice, her glaring eyes, her nose that she was certain was starting to run, the burning in her throat. She shoved herself up from the table and took a step back; the better to look down on her betrayer, and her voice was very nearly screeching as she yelled,*
> *"That is not okay! You can't do science with two people at once!"*
>
> Eliezer Yudkowsky, Harry Potter and the Methods of Rationality

This thesis has benefitted from a unique relationship between the German Aerospace Center (DLR) and the Politecnico di Milano. A fortunate situation enabled me to work fulltime at the DLR in Munich, while using a part of the resulting achievements for my PhD work in Milan. Two men took risks during this endeavour: Francesco Casella under whose supervision the work of this thesis was carried out and Dirk Zimmer who organised my work in such a way that writing a dissertation in parallel was not only possible, but straightforward. Both Francesco and Dirk proved to be excellent guides, in technical topics as well as general support. I am more than happy to have had them on my side and that it has worked out so well.
No scientific activity happens in a vacuum[1], and this contribution is no exception. A large number of close colleagues influenced this work in form of discussions, help, and support. This work would never have been completed without Andreas Klöckner, Daniel Bender and Manuel Pusch. Thank you for technical and moral support.
A large part of this thesis is motivated by various insights from Airbus Germany. I am especially grateful to Tim Giese and Bettina Öhler for the discussions, the

[1] Not actually true, if you work at the german aerospace agency, you should know better.

support and the data that made this work possible.
I thank my friends Marinette, Lisa, and Joachim for putting up with me, emotional support, awesome trips to Korea and Japan and the 'memorable' visit to India.
I thank Alexandra to Xenia for keeping me sane during this time.
I thank my father, for being an all-around good guy.
I thank Trey Parker and Matt Stone for inspiration. I thank Justin Roiland and Dan Harmon for emotional restoration.
I thank Alberto Leva from Politecnico di Milano and Maria Cruz Varona from the Technical University of Munich for helpful comments and discussions, which greatly improved parts of this work.
I thank all anonymous reviewers of the publications on which this thesis is based. Their comments improved the quality of this thesis indirectly, but significantly.
I thank all participants of the psychological experiments as described in Chapter 6 for their time, effort, and comments. I also thank the leaders of the Modelica User Groups Sachsen, North America, Japan, Hamburg, and Baden-Württemberg for their help with recruiting.
I thank my students Ansgar Lechtenberg, Andreas Schröffer, and Ines Kerling for their dedication, their diligent work and their help with the creation of this thesis[2].
I thank Marinette Iwanicki, Dirk Zimmer, Joachim Kreutzer and Daniel Bender for proofreading this thesis. I had a guilty conscience after seeing the amount of time you put into this task. Thanks also goes to Christoph Rennebeck for help with the translation of the italian abstract.
My apologies go to the legal departments of both DLR and Politecnico for causing so much work with the initial setup of the contract.
Ramen.

[2] As usual, the ratio of *actual work done* to *length of acknowledgements* discriminates against the students. But the world is not fair, and i want to keep with conventions.

Abstract

English Version

Unseen by the passengers, aircraft environmental control systems are complex thermodynamic systems, requiring a large quantity of power. The tasks of designing, modelling, optimising and controlling all these systems leave many degrees of freedom to the respective expert, and typically require many design loops to arrive at satisfactory results. This thesis contributes to multiple aspects of this process.

In aircraft environmental control systems (ECSs), limit cycle oscillations (LCOs) can occur. Those are problematic since the life expectancy of the ECS is affected. Using an equation-based, object-oriented modelling language (EOOML), a complete, detailed and dynamic simulation model of an ECS is developed. This model includes the engine bleed air system (EBAS), the air conditioning pack, the cabin and ducting dynamics as well as the recirculation system. Using simulations, it is shown that LCOs occurring in ECSs cannot be explained by *Helmholtz* resonance effects.

To further investigate the cause of the LCOs, electropneumatic valves - as used in EBAS - are modelled in more detail, using the Lund-Grenoble (Lu-Gre) friction model. Using this model, LCOs in aircraft ECS are predicted for the first time. Several control strategies are devised, implemented and evaluated against this model. A strategy based on a combination of feed-forward control, feed-back control and online tuning of the integral action outperforms all other candidates. A 46% reduction of the developed objective function is achieved when compared to the state of the art.

Current architectures for aircraft cabin climatisation only allow for a small number of temperature zones. Differences in heat load, generated for instance by nonconforming seating class layouts, cannot be compensated by the control

system. A new architecture is presented that allows for an infinite number of temperature control zones - at the cost of a more involved control system. Suitable control strategies, as well as failure management strategies are demonstrated.

For optimisation studies in the context of simulation models, a class of controllers is found, based on boundary layer sliding mode control (BLSMC). These controllers do not require any tuning effort and show good performance for many systems during simulations. The high sensitivity to noise is unproblematic, as the system is purely virtual at this stage. These features make them suitable for modelling experts at development stages where the architecture design is not yet finalised.

On the most basic level, usability aspects of EOOMLs are explored. It is found that the use of inheritance can severely retard the understanding of simulation models. Results also suggest graphical representations to be superior to block diagrams with equation-based and algorithm-based representations taking the middle spot.

Versione Italiano

Non visibili dai passeggeri, i sistemi di controllo ambientale di aeromobili sono complessi sistemi termodinamici, che richiedono una grande quantità di potenza. Il compito di progettazione, modellazione, ottimizzazione e controllo di tutti questi sistemi lascia molta libertà all'esperto e richiede in genere molti cicli di progettazione per ottenere risultati soddisfacenti. Questa tesi di dottorato contribuisce a molti livelli di questo processo.

Nei sistemi di controllo ambientale di aeromobili (ECS) possono verificarsi oscillazioni di ciclo di limite (LCO). Queste sono problematiche poiché l'aspettativa di vita della ECS viene ridotta. Utilizzando un linguaggio di modellazione orientato all'oggetto, basato sulla equazione (EOOML) viene sviluppato per la prima volta un modello di simulazione completo, dettagliato e dinamico di un'ECS. Questo modello include il "engine bleed air system (EBAS)", il condizionatore d'aria, le dinamiche di cabina e condotta, nonché il sistema di ricircolo. Usando le simulazioni, è stato dimostrato che le oscillazioni che si verificano in ECS non possono essere spiegate con gli effetti di risonanza di Helmholtz.

Per approfondire la causa delle oscillazione, le valvole elettropneumatiche - come usate in EBAS - sono modellate in modo più dettagliato, utilizzando il modello "Lund-Grenoble friction". Utilizzando questo modello, le oscillazioni sistemi di controllo ambientale di aeromobili sono previsti per la prima volta. Diverse strategie di controllo sono state concepite, implementate e valutate con questo modello. Una strategia basata su una combinazione di controllo feed-forward, controllo feed-back e tuning online dell'azione integrata supera tutte le altre strategie. Rispetto allo stato dell'arte viene raggiunta una riduzione del 46% della funzione obiettiva sviluppata.

Le architetture correnti per la climatizzazione delle cabine di aeromobili consentono solo un piccolo numero di zone di temperatura. Le differenze nel carico di calore, generate per esempio dai layout non conformi alle classi di posti a sedere, non possono essere compensati dal sistema di controllo. Viene qua presentata una nuova architettura che consente un numero infinito di zone di controllo della temperatura - a costo di un sistema di controllo più coinvolto. Sono state dimostrate strategie di controllo appropriate, nonché strategie di "failure management".

Per studi di ottimizzazione nel contesto di modelli di simulazione, viene individuata una classe di controller, basata sul "boundary layer sliding mode control" (BLSMC). Questi controllori non richiedono alcuno sforzo di sintonia e mostrano buone prestazioni per molti sistemi durante le simulazioni. L'alta sensibilità al rumore non è problematica, poiché il sistema è puramente virtuale in questa fase. Queste caratteristiche li rendono adatti per esperti di modellazione durante le fasi di sviluppo in cui il design dell'architettura non è ancora conclusa.

Sul piano di base, sono esplorati aspetti di usabilità di EOOML. Si scopre che l'uso dell' "ereditarietà" può rallentare seriamente la comprensione dei modelli di simulazione. I risultati suggeriscono inoltre che le rappresentazioni grafiche siano superiori ai diagrammi a blocchi mentre le rappresentazioni a base di equazioni e quelle a base di algoritmi condividono i posti due e tre.

Deutsche Version

Obwohl für Passagiere unsichtbar, sind Flugzeuglebenserhaltungssysteme komplexe thermodynamische Systeme, die eine große Menge an Leistung benötigen. Diese zu gestalten, zu modellieren, zu optimieren und zu regeln läuft für den jeweiligen Experten auf eine größere Menge an Freiheitsgraden hinaus. Typischerweise sind auch mehrere Designiterationen nötig, um ein akzeptables Ergebnis zu erreichen. Diese Arbeit unterstützt diesen Prozess auf mehreren Ebenen.

In Flugzeuglebenserhaltungssystemen (ECS) können Grenzzyklen (LCOs) auftreten. Diese sind problematisch, da die Lebensdauer des ECS negativ beeinflusst werden kann. Mithilfe einer gleichungsbasierten und objektorientierten Modellierungssprache (EOOML) wurde erstmals ein vollständiges, detailliertes und dynamisches Modell eines ECS entwickelt. Das Modell umfasst sowohl das Zapfluftsystem (EBAS) als auch die Kühlturbine, die Kabine samt Luftzuführung als auch das Rezirkulationssystem. Durch Simulationen wurde gezeigt, dass LCOs in ECS nicht durch Helmholtzresonanz erklärbar sind.

Um die Ursache dieser LCOs weiter zu untersuchen, wurden elektropneumatische Ventile - wie sie in EBAS eingesetzt werden - detaillierter modelliert. Dabei wurde das Lund-Grenoble (Lu-Gre) Reibungsmodell eingesetzt. Mit diesem Modell wurden LCOs in ECS erstmals vorhergesagt. Mehrere Regelungsstrategien wurden entworfen, implementiert und gegen das Modell evaluiert. Eine gemischte Steuerungs- und Regelungsstrategie zusammen mit einem Onlinetuning des Integralanteils zeigte die besten Ergebnisse. Hierbei wurde eine 46% Reduzierung der entwickelten Zielfunktion gegenüber dem Stand der Technik erreicht.

Aktuell verwendete Architekturen zur Klimatisierung von Flugzeugkabinen lassen nur eine sehr begrenzte Anzahl an Temperaturzonen zu. Abweichungen der Hitzelast über die Länge der Flugzeugkabine, wie sie zum Beispiel durch unglücklich dimensionierte Beförderungsklassen erzeugt werden, können so durch das Regelungssystem nicht kompensiert werden. Eine neuartige Architektur wurde vorgestellt, die eine unendliche Zahl an Temperaturzonen ermöglicht. Im Gegenzug steigt die notwendige Komplexität des Regelungssystems. Passende Regelungsstrategien sowie Strategien zur Fehlerbehandlung werden gezeigt.

Für Optimierungsstudien, basierend auf Simulationsmodellen, wurde eine Klasse an Reglern gefunden, die auf einer Randschicht-Sliding-Mode-Regelung (BLSMC) basiert. Diese Regler müssen nicht eingestellt werden, und zeigen gute Leistungen für viele Systeme während einer Simulation. Die hohe Sensitivität bezüglich Messrauschen ist unproblematisch, da das System zu diesem Zeitpunkt rein virtuell existiert. Damit sind diese Regler attraktiv für Modellierungsexperten zu Projektphasen, an denen die Gesamtarchitektur noch nicht festgelegt ist.

Auf der elementaren Ebene wird die Benutzerfreundlichkeit von EOOMLs analysiert. Es wird gezeigt, dass die Verwendung von Vererbungsoperatoren die Verständlichkeit von Simulationsmodellen deutlich erschwert. Die Ergebnisse legen außerdem nahe, dass eine grafische Repräsentationsform von Simulationsmodellen gegenüber Blockdiagrammen überlegen ist. Gleichungsbasierte und algorithmenbasierte Darstellungen nehmen dabei die Mittelplätze ein.

Acronyms

BLSMC boundary layer sliding mode control. IV, 73, 75, 77, 79, 81, 82

CAD computer aided design. 67

CFD computational fluid dynamics. 51, 67

DAE differential algebraic equation. 5, 76

DDCSD data-driven control system design. 73, 74

DLR German Aerospace Center. I, II, 20, 88, 91

EBAS engine bleed air system. III, 1, 2, 7, 9, 11, 12, 17, 18, 26, 33, 39, 107

ECS environmental control system. III, 1, 2, 6, 7, 6, 8, 9, 11, 12, 13, 14, 18, 19, 52, 53, 54, 107

EOOM equation-based, object-oriented modelling. 2, 5, 6, 72, 74

EOOML equation-based, object-oriented modelling language. III, IV, 2, 5, 6, 9, 12, 24, 60, 69, 71, 72, 107

FCV flow control valve. 17, 33, 34, 36, 38, 39

HEPA high efficiency particulate air. 8

HPV high pressure valve. 33, 34

IQCE integrated quadratic control error. 79

LAN local area network. 20

LCO limit cycle oscillation. III, 2, 8, 9, 11, 12, 24, 31, 32, 107

LPV linear parameter varying. 24

LQE linear quadratic estimator. 59, 72

LQG linear quadratic gaussian. 34, 39, 41, 59, 61, 62, 63, 72

LQR linear quadratic regulator. 59, 62, 72

LTI linear time-invariant. 72

Lu-Gre Lund-Grenoble. III, 29

M mean. 91, 92

MEA more electric aircraft. 1

MIMO multiple input multiple output. 2, 72, 73, 81

MOPS Multi-Objective Parameter Synthesis. 61

MPC model predictive control. 74

MSL Modelica Standard Library. 49, 91, 92, 94

OFV outflow valve. 8, 17, 18

PI proportional-integral. 33, 34, 36, 39, 40, 57, 70, 82

PID proportional-integral-derivative. 23, 25, 57, 58, 59, 61, 62, 63, 70, 71, 79

PRV pressure reduction valve. 17, 33, 34, 36, 38

RMS robotic motion simulator. 20, 21

SD standard deviation. 91, 92

SiL software in the loop. 67

SISO single input single output. 71, 75, 81

SMC sliding mode control. 73, 74, 75, 77, 79, 80, 81

SRPV self-regulating pneumatic valve. 9, 12, 23, 24, 25, 31

UDP user datagram protocol. 20

UML unified modelling language. 86

VRFT virtual reference feedback tracking. 73

XML extended mark-up language. 98

Contents

Acknowledgements **I**

Abstract **III**
English . . . III
Italian . . . V
German . . . VII

Acronyms **VIII**

this() **XI**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Thesis overview . . . 2
1.2.1 Structure . . . 3
1.2.2 Associated publications . . . 3
1.3 Equation-based object-oriented modelling . . . 5
1.3.1 Objectives and contributions . . . 6
1.4 Aircraft environmental control systems . . . 6
1.4.1 Objectives and contributions . . . 9

2 Helmholtz resonance in aircraft environmental control systems **11**
2.1 Problem definition . . . 11
2.2 Introduction to Helmholtz resonance . . . 12
2.3 Simulation of aircraft ECS . . . 14
2.3.1 Thermal domain . . . 15
2.3.2 Pneumatic domain . . . 17

2.3.3 Control logic . . . 17
2.3.4 Mechanical domain . . . 18
2.4 Results and discussion . . . 18
2.5 Outlook and future work . . . 20
2.6 Acknowledgments . . . 21

3 Countermeasures for bleed system oscillations **23**
3.1 Problem definition . . . 23
3.2 Valve modelling . . . 24
3.2.1 Functioning principle . . . 24
3.2.2 Detailed valve model . . . 26
3.2.3 Actuator model . . . 27
3.2.4 Statistics . . . 31
3.3 Cause of limit cycle oscillations . . . 32
3.4 Development of a bleed system controller . . . 33
3.4.1 Baseline controller . . . 34
3.4.2 Stiction compensation . . . 37
3.4.3 Evaluation criterion . . . 38
3.4.4 Optimisation and validation . . . 39
3.5 Results and discussion . . . 39
3.6 Outlook and future work . . . 41
3.7 Acknowledgments . . . 43

4 A concept for aircraft temperature regulation **45**
4.1 Problem definition . . . 45
4.2 Concept . . . 46
4.3 Modelling . . . 48
4.3.1 Ducting . . . 50
4.3.2 Cabin . . . 51
4.4 Pack energy consumption . . . 53
4.4.1 System description . . . 53
4.4.2 Energy analysis . . . 55
4.5 Control . . . 56
4.5.1 Requirements . . . 56
4.5.2 Control strategies . . . 57
4.5.3 Results . . . 61
4.6 Failure management . . . 64
4.6.1 Pack failure . . . 65
4.6.2 Valve failure . . . 66
4.6.3 Recirculation failure . . . 66
4.7 Discussion . . . 66
4.8 Outlook and future work . . . 67
4.9 Acknowledgements . . . 68

5 One-size-fits-all-control for architecture optimisation **69**
5.1 Problem definition . . . 69
5.2 Suitability of major control approaches for architecture simulation 71
5.2.1 PID . . . 71
5.2.2 Model inversion . . . 71
5.2.3 LQR/LQG . . . 72
5.2.4 H_∞ . . . 72
5.2.5 Sliding mode control . . . 73
5.2.6 Data driven control system design . . . 73
5.2.7 Model predictive control . . . 74
5.2.8 Others . . . 74
5.3 Universal controllers for equation-based simulation environments 75
5.3.1 Concept . . . 75
5.3.2 Boundary layer sliding mode control . . . 75
5.3.3 Implementation . . . 76
5.4 Examples . . . 77
5.4.1 Low-order-systems . . . 77
5.4.2 Thermal network . . . 79
5.4.3 Cabin temperature control . . . 80
5.5 Discussion . . . 81

6 Usability aspects of equation-based modelling **85**
6.1 Problem definition . . . 86
6.2 Literature summary . . . 87
6.3 Explorative expert surveys . . . 88
6.3.1 Method . . . 88
6.3.2 Selected results . . . 89
6.4 The cost of inheritance . . . 90
6.4.1 Method . . . 91
6.4.2 Results . . . 92
6.4.3 Discussion . . . 94
6.5 Viscosity of representations . . . 95
6.5.1 Method . . . 95
6.5.2 Results . . . 101
6.5.3 Discussion . . . 104
6.6 Acknowledgments . . . 106

7 Conclusions **107**

A Appendix **109**
A.1 Modelica code of simple controller . . . 109

Bibliography **112**

Publications **119**

CHAPTER 1

Introduction

1.1 Motivation

As of 2017, the civil air transport sector continues to grow. According to the Airbus global market forecast, the global passenger aircraft fleet is expected to grow from 18.020 aircraft in 2015 to 37.710 in 2035 [1]. This increase of 109% corresponds to 3.75% annually. Boeing predicts an increase from 22.510 aircraft[1] in 2015 to 45.240 in 2035, corresponding to an increase of 101%, or 3.55% annually [2]. As always in engineering, a larger number of produced systems increases the motivation for further optimisation even of mature architectures.

Civil aircraft manufacturers continue to stick to the tried and tested "tube with wings"-concept. Modern passenger aircraft look largely similar to their decade-old counterparts. Nevertheless, fuel efficiency[2] has roughly doubled since 1960, albeit at a diminishing rate [3]. This is mostly due to improvements in the engines, replacing turbojets with high-bypass-ratio turbo-fans.

The contribution of aircraft energy systems to these efficiency improvements has been relatively smaller. The topology of environmental control systems (ECSs) has been stable since at least the 1960s, with a few exceptions like the Boeing 787. Interest in alternative architectures such as the more electric aircraft (MEA) [4] remains mostly on an academic level so far.

[1] This figure is larger than the Airbus figure because it also includes freighter aircraft.
[2] Defined by the number of passengers times flight distance per unit fuel burn.

Despite being an old and tested design, conventional ECSs still suffer from problems. Stability issues in the engine bleed air system (EBAS) lead to longer development times and additional flight tests. Nonuniform temperature regulation inside the aircraft cabin leads to degraded passenger comfort.

Equation-based, object-oriented modelling (EOOM) is heavily used to design system architectures, size components, and develop control systems in the context of aircraft ECSs [5]. Languages like *Modelica* enable fast and accurate modelling of complex systems, resulting in faster and cheaper development of aircraft energy systems. These languages already reached a degree of technological maturity and have grown to cover a wide area of application cases. At the time of writing, new approaches based on Modelica are prepared that even combine systems modelling and conventional programming paradigms [6]. However, even a powerful modelling language falls short of its possibilities, if the users cannot utilise them. While the aspect of usability has always been a concern in the advancement of equation-based, object-oriented modelling languages (EOOMLs), there have not been any systematic investigations or published results.

1.2 Thesis overview

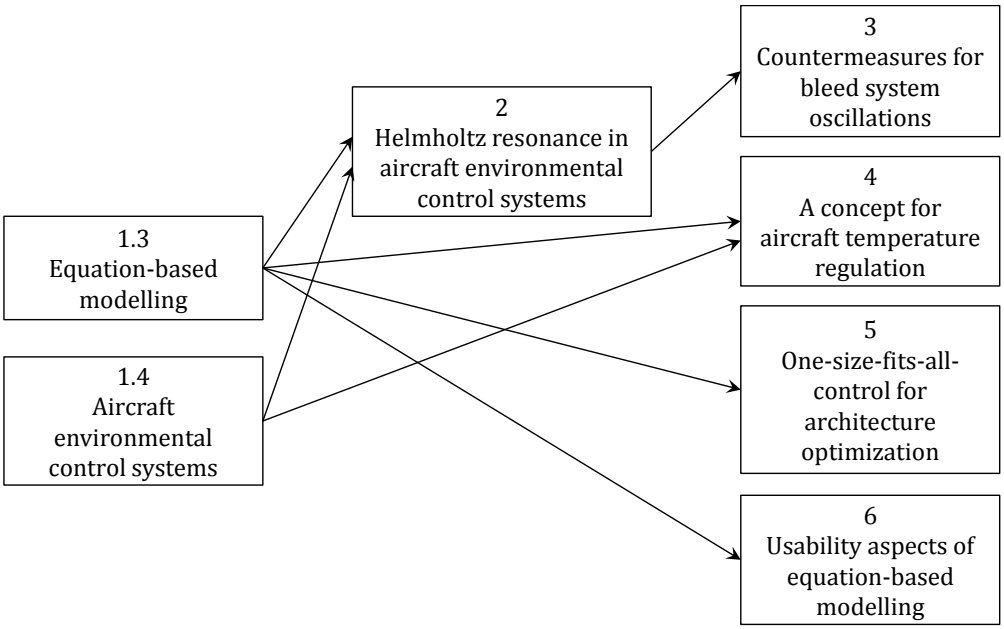
The contributions of this thesis apply to modelling, control and advancement of aircraft energy systems, with a focus on environmental control systems. These topics are distributed over several steps of a typical development process and are dependent on each other: a control problem gets easier, if the system is designed reasonably and a good model is available. Modelling gets easier if the underlying modelling environment supports the user and a large number of high quality libraries are available.

The current chapter introduces the general topic, describes the state of the art and identifies research gaps for each of the relevant subtopics. In chapter 2, a dynamic model of a complete ECS is developed. The relationship between limit cycle oscillations (LCOs) and Helmholtz resonance is investigated. Chapter 3 shows how EBASs can be modelled in detail. LCOs based on valve stiction are reproduced. Suitable control strategies and stiction compensation approaches are compared. The topic of cabin temperature regulation is explored in chapter 4. Drawbacks of commonplace temperature regulation systems are discussed and an alternative architecture is presented. A flexible low-order model is created, the connection to ECS energy efficiency is made, and several control approaches are compared. Finally, the handling of failure modes is discussed. Chapter 5 presents a class of controllers that show good performance for multiple input multiple output (MIMO)-systems without any kind of tuning. They only work inside a simulation environment, which makes them suitable for optimization studies at early design stages. In chapter 6, usability aspects

of EOOMLs are investigated. Several psychological experiments are conducted to measure the influence of inheritance and different model representations on user comprehension. The thesis is concluded in chapter 7.

1.2.1 Structure

PhD theses contain large amounts of text, and this work is no exception. Additionally, some of the readers are in positions where time is in short supply. To help the reader skip sections, and illustrate the inner structure of this thesis, see the following dependence graph:



Additionally, the thesis is optimised for reading of individual chapters. This means that important technical systems are explained and acronyms are written out repeatedly at the beginning of each chapter. If you are intending to read the thesis from beginning to end, some repetition is therefore unavoidable; the author apologises in advance.

1.2.2 Associated publications

The majority of the content of this thesis has already been published at scientific conferences and journals. Other publications resulting from the PhD work were deliberately left out of this work to ensure a minimum of cohesiveness [7, 8, 9, 10, 11]. For a quick overview about how the remaining publications roughly relate to the sections of this thesis, see the following table:

		Modelica 2015 [12]	EOOLT 2016 [13]	MCMDS [14]	AST 2017 [15]	IFAC 2017 [16]	Modelica 2017 [17]	CCTA 2017 [18]	SAE Aerotech 2017 [19]	EOOLT 2017 [20]
		High-fidelity Modelling of Self-regulating Pneumatic Valves	The use of Ockham's Razor in object-oriented modeling	Modelling and Simulation of Self-regulating Pneumatic Valves	Simulation of Helmholtz Resonance Effects in Aircraft ECS	Comparison of Control Strategies for Aircraft Bleed-Air Systems	Rapid development of an aircraft cabin temperature regulation concept	Control Strategies for an Advanced Aircraft-Cabin Temperature System	Advanced temperature control in aircraft cabins – a digital prototype	Representations of equation-based models are not created equal
ECS dynamics	2.1				■					
	2.2				■					
	2.3				■					
	2.4				■					
Bleed system oscillations	3.1	■		■		■				
	3.2	■		■						
	3.3			■						
	3.4					■				
	3.5					■				
Aircraft temperature regulation	4.1						■	■	■	
	4.2						■	■	■	
	4.3						■			
	4.4						■			
	4.5							■		
	4.6								■	
	4.7								■	
One-size-fits-all control	5.1									
	5.2									
	5.3									
	5.4									
	5.5									
Usability aspects of eq.-based modelling	6.1									
	6.2									
	6.3									
	6.4		■							
	6.5									■

1.3 Equation-based object-oriented modelling

For modelling and simulation of large scale dynamical systems, just writing down the corresponding system of differential equations is not a realistic option anymore. Alternative methodologies to do so are needed to get adequate results within a reasonable timeframe. A popular way is to build up block diagram representations of system components using a graphical user interface, with an optional second step of combining several of those in a hierarchical fashion. However, this approach requires the modeller to reformulate the system as directional block diagrams; a task that can be quite tricky. Furthermore, the different block diagrams have to be compatible in the sense that inputs and outputs of the subsystem are balanced.

Alternatively, modellers can use the approach of EOOM. Here, some of the previously mentioned tasks are handled by the modelling language and simulation tool. The components of the system are described in the form of differential algebraic equations (DAEs). The modeller does not have to think about the link between the physical equations and the computation strategy used for simulation. The components are then connected in an acausal way. This means that the direction of computation is left undefined. Only before the simulation is started, the simulation tool gathers together the equations from all components and reduces the overall system to a DAE of index one [21] before starting the numerical integration.

Five concepts are typical for EOOMLs:

Equation-basis: the dependencies between all occurring variables are described by mathematical equations, in contrast to programming languages, where assignments are used. These equations can be static as well as differential.

Object-orientation: a component (described by equations, other components or any combination thereof) can be instantiated several times. Parameters change the behaviour of different instantiations. Inheritance can be used to modify base components.

Acausality: since equations are used instead of assignments, the direction of computation is not given by the user. As long as all variables are defined correctly, the compiler independently tries to find a suitable representation. This can be used to automatically compute optimal control inputs by prescribing the demanded system trajectory.

Physical connectors: components share some variables to the outside using connectors. These connectors are possibly but not mandatorily physically motivated, they often represent a pair of variables that correspond to effort and flow, which multiply to the unit of power (as in voltage and current, or torque and rotational speed).

Graphical modelling: components can be represented and instantiated in a graphical user interface. The components can also be interfaced graphically via their connectors. Depending on the definition of the connectors, connections correspond to physical equations that are generated in the background. For example, when connecting two or more electrical connectors, the interface voltage is set equal, and the sum of all currents into the interface node is set to zero.

1.3.1 Objectives and contributions

Modern EOOMLs and tools are relatively mature and flexible. There is usually more than one way to skin a cat: the same model can be created by equations, as an algorithm, or graphically. If a library is created, the total number of lines of code can optionally be decreased by heavy use of inheritance. The best way to do it is not always clear. Contrary to most programming languages, there are not many published investigations into usability aspects of EOOMLs.

The objective of this thesis with respect to EOOM languages and tools is to conduct investigations into the usability and psychology. This thesis contributes to the state of the art in the following areas:

- The literature regarding the usability and psychology of programming languages is reviewed. Some of these investigations may possibly be relevant for EOOMLs as well. The corresponding results are collected and presented.
- Expert interviews are conducted that explore how EOOMLs and tools help or constrain modelling experts during tasks.
- An experiment is conducted to estimate the effect of inheritance on the comprehensibility of models.
- Another experiment is conducted to estimate the effect of different model representations on identifiability, comprehensibility and perceived trustworthiness.

These contributions are described in chapter 6.

1.4 Aircraft environmental control systems

Aircraft cabins have to be supplied with fresh air. Also, pressure, temperature, and humidity have to be regulated. These are the tasks of the ECS. A conventional ECS is pneumatically powered using bleed air from the compressor stages of the aircraft engine. The hot and pressurised air is expanded, cooled down, and distributed into the cabin. The geometrical shape of a typical ECS

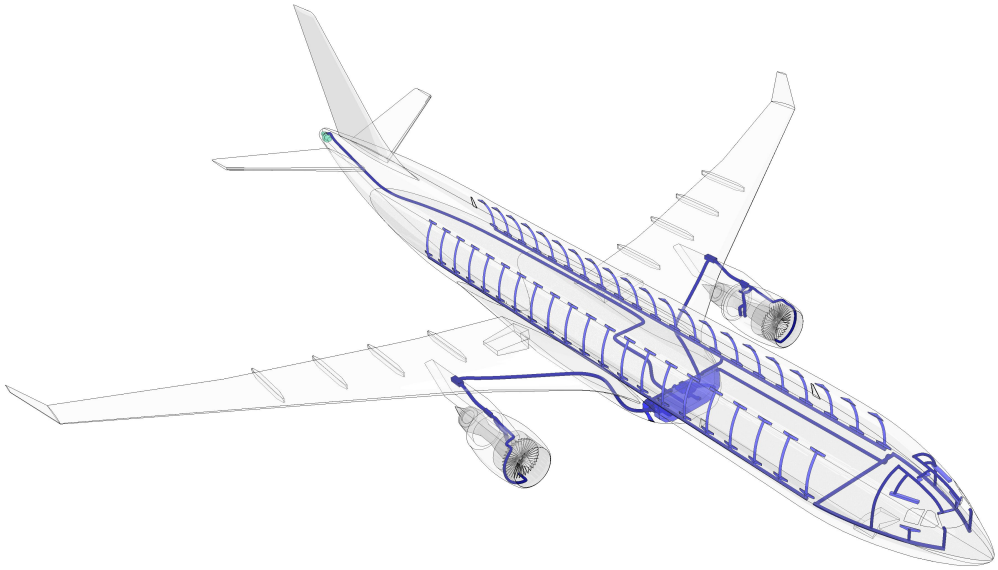


Figure 1.1: *Illustration of an aircraft ECS*

is illustrated in Figure 1.1, whereas a simplified topology can be seen in Figure 1.2.

The energy offtake of ECSs is quite large and can easily be estimated: aviation codes (FAR25) demand a fresh air mass flow rate of around 4.16g per passenger and second[22][3]. The allocation of the minimum amount of fresh air for 200 passengers requires around 565kW continuously[4]. Typical numbers are even higher, because most airframers provide passengers with more air than required by code.

Conventional ECSs are composed as follows: hot and pressurised air is diverted from one of two compressor stages of the aircraft engines, depending on the engine state. Inside the engine nacelle, the air is expanded and precooled against the fan air flow to around 200°C. A set of valves called the EBAS regulates the pressure and massflow.

The air then enters the air conditioning pack, where it is cooled against outside air. Passively powered turbomachinery is used to increase the temperature difference in the pack heat exchangers, making the heat transfer more efficient. The mass flow of cooling air is regulated by a controllable ram air inlet, which utilises the aircraft velocity to scoop environment air into the ram air channel, where it is ducted through the heat exchangers. The ram air inlet is used in

[3]The actual number is defined in freedom-units as 0.55lbs per passenger and minute.

[4]The energy offtake of an ECS it completely defined by the power required to provide air at the compressor stages. Bleed air can be assumed to have a temperature of 600°C and a pressure of 10bar on average. Outside air has around -50°C and a pressure of 0.25bar. The difference in enthalpy is around 680kJ/kg. 0.624kg/s of air are required for 200 passengers. Total power offtake can be computed as $m_{flow} \cdot \Delta h = 565.5\text{kW}$.

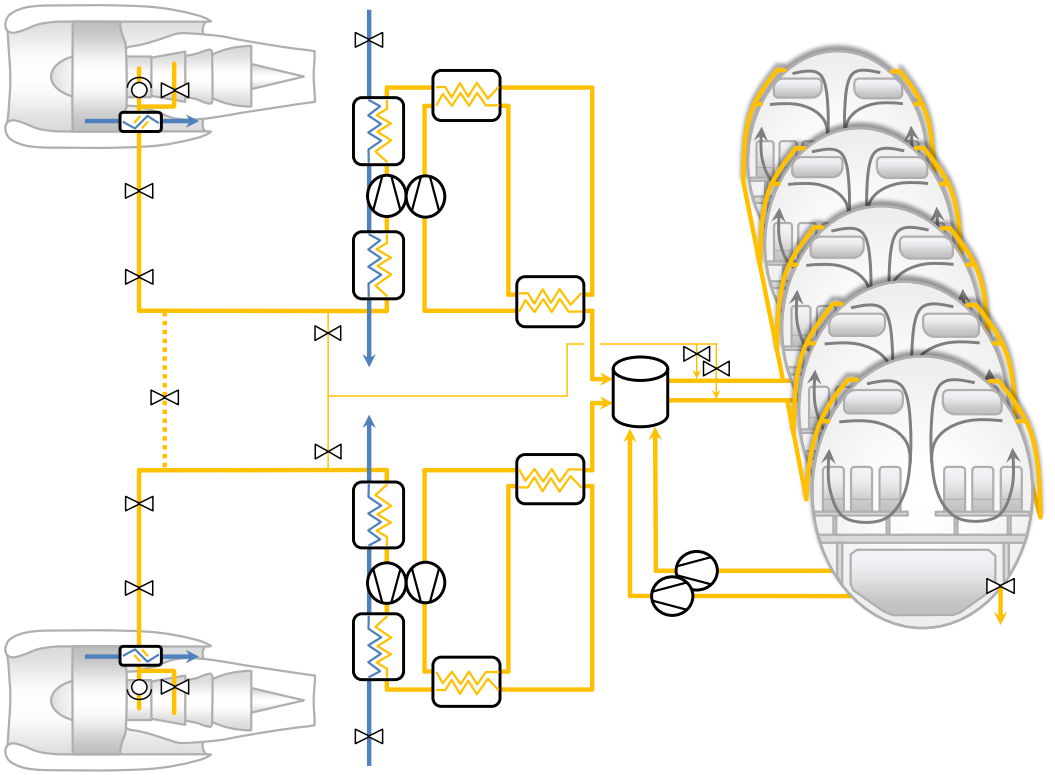


Figure 1.2: *Topology of an aircraft ECS*

this way to regulate the pack discharge temperature.

After the bleed air is cooled to an appropriate temperature, it is ducted into the mixing chamber. Here it is mixed with recirculation air and divided into several main ducts. For each duct, a small amount of hot trim air (air that bypasses the air conditioning pack) is added to regulate the temperatures for each duct individually. These ducts transport the air to their respective cabin temperature zones, where the air is distributed to several riser ducts each. The air is then blown into the upper part of the aircraft cabin, inducing a pair of curls that mix the air inside the cabin effectively.

Excess air leaves the cabin through small openings at the sides of the cabin floors into the underfloor volume. From there, a fixed mass flow is filtered using high efficiency particulate air (HEPA) filters and recirculated to the mixing chamber using recirculation fans. The remaining air is vented to the environment by the outflow valve (OFV), which is also tasked with regulating cabin pressure.

1.4.1 Objectives and contributions

Even though the structure of ECSs has been similar for a few decades, there is still some room for improvement in a few areas. During the development of new aircraft models, different ECS subsystems are developed independently, and interaction effects are noticed only during the maiden flight. Often, LCOs are a concern. To guarantee performance requirements, major controller tuning efforts can be necessary, requiring additional flight tests and creating significant costs. Additionally, current cabin temperature regulation systems are limited to a low number of temperature zones by their topology. Alternative topologies are conceivable, but would necessitate complex simulation models and control systems. The objective of this thesis with respect to aircraft ECSs is to advance the state of affairs regarding modelling technologies. One of the goals is the reduction of the number of flight tests during aircraft development. Also, an alternative cabin temperature regulation concept is developed. This thesis contributes to the state of the art in the following areas:

- Helmholtz resonance is a candidate for the cause of LCOs in aircraft ECSs that have been observed in the past. A meaningful investigation of this hypothesis requires detailed modelling of a complete aircraft ECS. An appropriate model is developed in the EOOML Modelica. A strategy to investigate the occurence of Helmholtz resonance is presented and executed. This contribution is described in chapter 2.
- In the EBAS, the massflow and pressure of bleed air is regulated using self-regulating pneumatic valves (SRPVs). These valves use a piston for pneumatic self-actuation, while a superimposed electronic controller is used to augment the performance. Stiction behaviour can occur in the piston with significant consequences on the system behaviour. A detailed multiphysical model of SRPVs is developed in the Modelica language. Based on this result, a complete EBAS is represented. With this model, LCOs in aircraft ECSs are successfully predicted. Several control strategies and stiction compensation techniques are implemented and compared. Some of them are shown to significantly improve system level performance. This contribution is described in chapter 3.
- Current architectures for cabin temperature regulation divide the aircraft cabin into a low number of temperature control zones. This number is bounded due to weight and space-constraints. For compensation of localised heat sources in the cabin, a larger number of temperature zones and therefore an alternative architecture is needed. An appropriate architecture is proposed. A suitable model is created and several control system candidates are presented and evaluated. Failure case handling is discussed. This contribution is described in chapter 4.

CHAPTER 2

Helmholtz resonance in aircraft environmental control systems

The Pella shuddered as the drive passed through some resonance frequency. Normally when that happened they weren't under high burn. It was strange how something that was hardly more than a chime at a third of a g could sound like the coming apocalypse at two and a third. He tapped out a message for Josie down in engineering: keep us in one piece.
A few seconds later, Josie wrote back something obscene, and Marco chuckled through the pressure on his throat.

Daniel Abraham and Ty Franck, The Expanse

2.1 Problem definition

In environmental control systems (ECSs) of passenger aircraft, limit cycle oscillations (LCOs) are always a potential concern. These are bounded oscillations caused by nonlinear system dynamics. Current modelling and simulation approaches are not perfectly up to the task of predicting related problems. This can result in additional flight tests later down the line with corresponding costs.

A common approach is to split the air generation pipeline into engine bleed air system (EBAS), air conditioning pack, mixing unit/ducting system, and cabin. The control systems of those parts are then developed and simulated

independently. Sometimes this approach is justified by the argument of a frequency split - the dynamic time constants of neighbouring systems are supposed to be sufficiently different to not interfere. It is further suggested that the dynamics downstream only have a minor influence on the dynamics upstream. In any case, end results vary.

In literature, many modelling approaches for aircraft ECS are described. Most of them are concerned with thermodynamic optimisation of the process ([23], [5], [24]). Here, transient effects are not relevant, therefore control system stability cannot be predicted with those models. In [25], air conditioning packs for ECSs were modelled on different detail levels, corresponding to different design stages of the aircraft. In [12] and [14], self-regulating pneumatic valves (SRPVs) were modelled in the equation-based, object-oriented modelling language (EOOML) *Modelica*. These valves exhibit strong stick-slip behaviour, and are assumed to be a fundamental cause for LCOs in ECS.

While some of those works can help predicting control system instabilities, they consistently do not model the complete path of the bleed air from engine to cabin. For reasons explained in section 2.2, consideration of the complete system is crucial for accurate results.

In this chapter it is investigated if the entirety of aircraft ECSs and cabin can be considered as a *Helmholtz resonator*, resulting in a pronounced resonance frequency on aircraft level. It is shown how the topology of traditional ECSs compares to Helmholtz resonators. Aircraft level transient simulation is presented as a necessary measure for reliable prediction of LCOs. A suitable model and associated simulation results are presented to substantiate this claim.

2.2 Introduction to Helmholtz resonance

Helmholtz resonators were first described by Hermann von Helmholtz in 1863 [26]. They are composed by air-filled cavities connected to a pipe. When exposed to pressure variation, Helmholtz resonators exhibit a second-order behaviour with low damping, i.e. a pronounced resonance frequency. This effect is used heavily in mid-tier audio equipment, where so called *bass-reflex systems* are used to increase power efficiency.

The operating principle is similar to mechanical spring-damper systems or electric oscillating-circuits: two modes of energy storage are connected and perfectly out of phase. An overpressure inside the volume drives the air out of the pipe, the inertia of this massflow causes the pressure compensation to overshoot, resulting in an underpressure, and so on. This is illustrated in Figure 2.1.

Aircraft ECSs share their topology with Helmholtz resonators: the closed volume is represented by the cabin, while the pipe is represented by the EBAS and air conditioning pack, which are connected in series. While the aircraft cabin

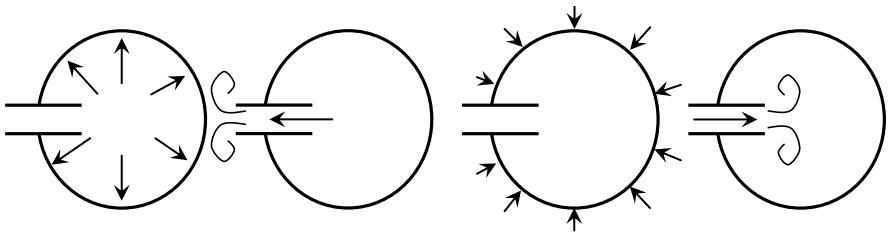


Figure 2.1: *Illustration of a Helmholtz resonator cycle*

is large compared to other examples of Helmholtz resonators like loudspeakers, the principle of operation is not limited by size. The only difference is the resulting resonance frequency, which is dependent on air volume, pipe length and pipe diameter.

Simulation of this phenomenon is straightforward. In Figure 2.2, a simple Modelica model and corresponding simulation results are presented. A cabin of 100m^3 is connected to a component representing two parallel pipes with a length of 10m and a diameter of 0.15m. The pipes are subjected to a pressure chirp with an amplitude of 1000Pa. The model is built from components of the *Modelica Fluid library* [27], with activated dynamic impulse balance. It can be seen that the resulting pipe flow exhibits a strong resonance at around 0.33Hz. The maximum amplitude of the pipe flow is 1.77kg/s.

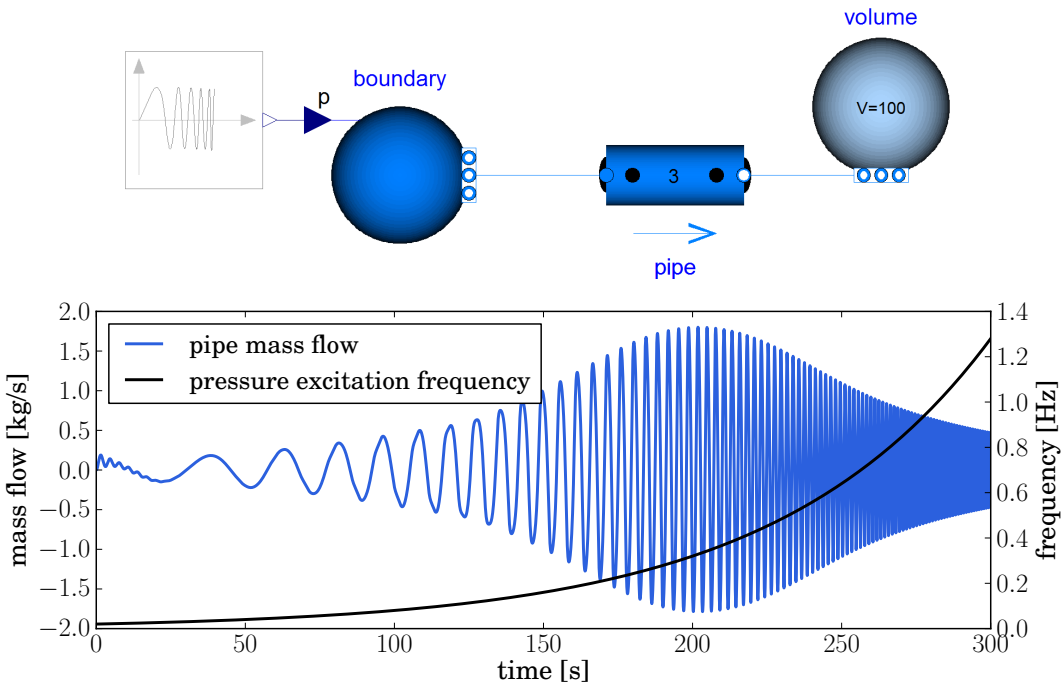


Figure 2.2: *Simulation of a Helmholtz resonator subjected to a chirp signal*

Of course, ECSs have some peculiarities: contrary to typical Helmholtz resonators, the pipe flow in ECSs does not average to zero and contain flow restrictions. To take this into account, the simulation was repeated with an added offset pipe flow of 1kg/s and heavily increased friction. The result can be seen in Figure 2.3. The maximum amplitude of the pipe flow decreases to 0.5kg/s and the resonance frequency stays at 0.33Hz. Still, the overall resonance effect is significant.

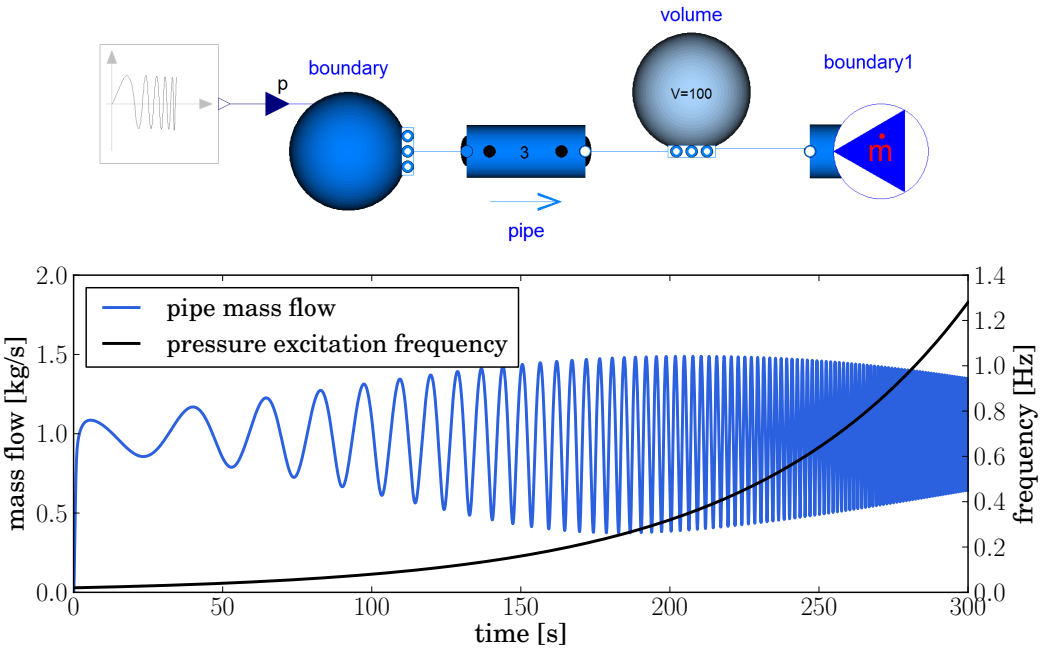


Figure 2.3: *Simulation of a Helmholtz resonator with extraction subjected to a chirp signal*

This analysis is evidence that Helmholtz resonance plays an important role in the system dynamics of aircraft ECSs. To the best knowledge of the author, this effect has not been considered on earlier aircraft projects. A single patent could be found where Helmholtz resonance was given consideration for bleed air scoop inlet design [28], but no similar analysis was done on aircraft level.

2.3 Simulation of aircraft ECS

Of course, for a final control system design, a much more detailed analysis is necessary. Aside from pipe friction, several other effects have to be considered: the air conditioning pack contains turbomachinery, which has its own inertia. Some air from the cabin is recirculated, filtered and ducted to the mixing chamber, where it is mixed with fresh air from the air conditioning pack. Several valves in the bleed air system actively control pressure and mass flow, interfering with each other as well as with the Helmholtz resonance effect. To

adequately simulate an aircraft ECS, detailed modelling of the complete system is therefore necessary. An example for this is presented in the following.

Modelica was used as a modelling platform. The overall model contains 9 top-level components for the bleed system, 12 for the air conditioning pack, and 9 for ducting and cabin. It can be seen in Figure 2.4. Generally, no discretization is used, except in pipe models, where a one-dimensional *staggered grid* scheme is used to model transport phenomena realistically. For other components, zero-dimensional modelling is used. The fluid state, expressed for instance as a tupel of pressure, enthalpy and moisture, is only computed at important points along the flow path. For example, a compressor model typically features two fluid states, representing the fluid before and after the compressor.

For a more detailed look, the thermal, pneumatic, control logic and mechanical domains are separately regarded in the following. Therefore, all modelled effects are described in this section.

2.3.1 Thermal domain

Air bled from the engine compressor stages is far too hot to be directly ducted into the cabin. It can reach several hundred degrees Celsius. It is therefore precooled inside the engine fan (not included in this model), then ducted to the air conditioning pack. There it is cooled in the primary heat exchanger against cold ram air from the environment, and compressed again. The compression increases the temperature, increasing the efficiency of the main heat exchanger, where it is again cooled against ram air. Downstream, a combination of two heat exchangers, a water extractor, and a turbine expand and cool the air and decrease humidity. From there the air is ducted into the mixing chamber.

The air enters the cabin through inlets at the top of the cabin, generating two longitudinal vortexes. These help with the mixing of the cabin air. Air then leaves the cabin to the underfloor volume through outlets near the bottom of the cabin. A portion of the cabin air is vented outside, the remainder is filtered and recirculated by fans into the mixing chamber. Here, the recirculation air is mixed with fresh air from the air conditioning pack and ducted into the different cabin temperature zones. The air in the mixing chamber has the correct temperature for one of the temperature zones, but is too cold for the others. For these zones, a small amount of uncooled bleed air is added to control the temperature.

Contrary to intuition, aircraft cabins have to be cooled rather than heated most of the flight time. Although outside temperatures can reach -70°C, the cabin wall is well isolated, while the heat generation of passengers, cabin lights and inflight entertainment systems can be quite substantial. On the other side of the spectrum, another critical situation for aircraft air conditioning is the ground case at hot locations. To keep the cabin at the desired temperature, the cabin heat balance has to be brought to zero by controlling the inlet air

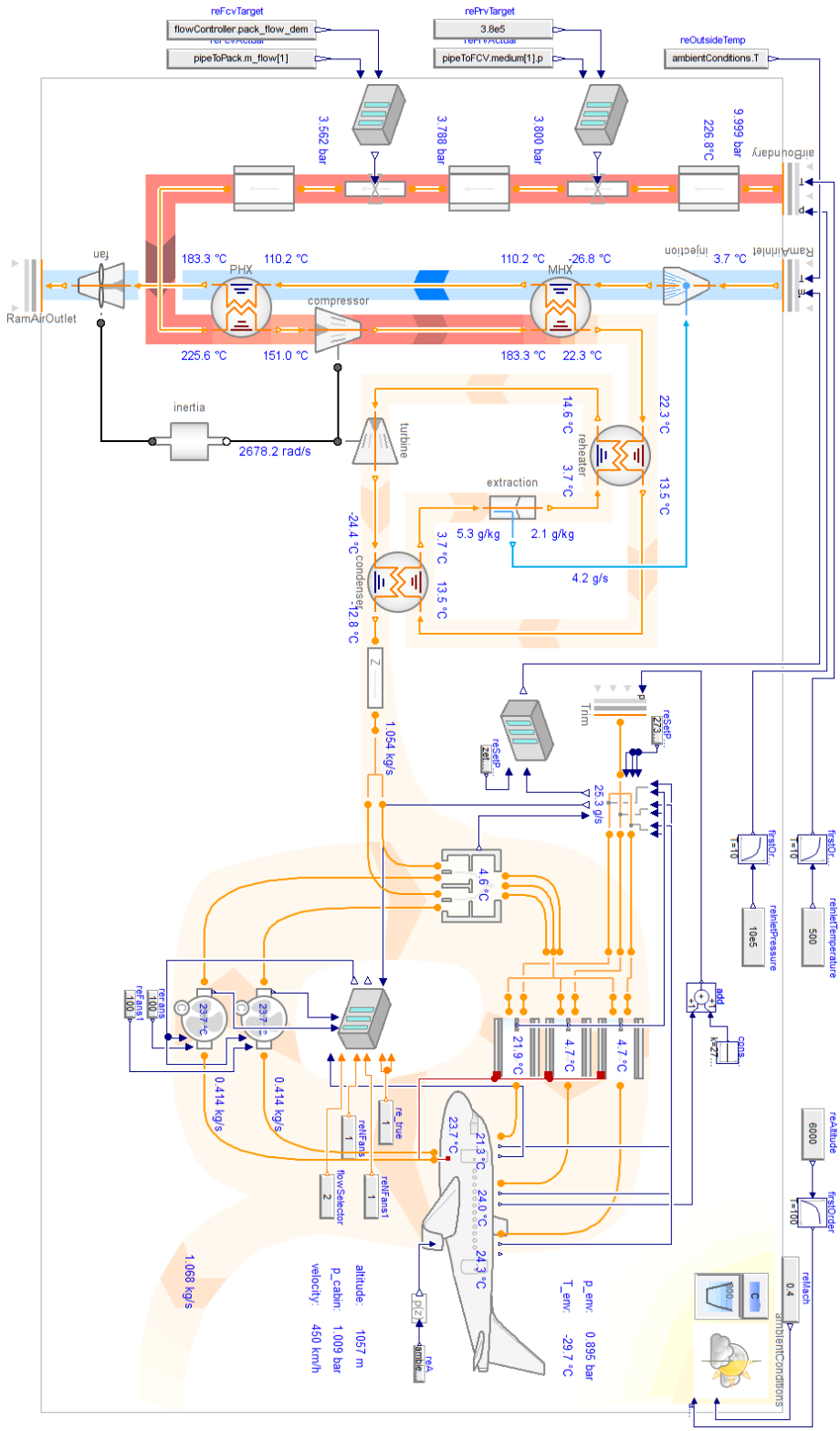


Figure 2.4: *Modelica diagram of aircraft energy system model*

temperature.

2.3.2 Pneumatic domain

Hot dense air is expanded through the pressure reduction valve (PRV) and the flow control valve (FCV) in the EBAS. The distance between those valves can be quite substantial, resulting in a significant air mass in the connecting pipe. By extension, the corresponding inertia should be significant as well. This is relevant especially in this work, where resonance behaviour in aircraft energy systems is analysed.

Inside the air conditioning pack, air pressure is still much higher than cabin pressure. This energy is used to drive a turbine, which in turn powers a compressor. Excess power is used to power a fan that ensures airflow through the ram air channel when ram pressure is inadequate, for example during ground operations.

To distribute air evenly over the length of the aircraft cabin, a system of distribution- and riser-ducts is employed. In this work, the ducting system for each zone was modelled as a single pipe including a thermal inertia of the piping material. In a preliminary study, the discretisation of these pipes was fitted in such a way that the dynamic response closely matched the average response of the complete ducting pipe network.

While the air volume flow into the cabin is constant, a varying volume flow is vented into the environment through the outflow valve (OFV). During altitude changes, the target pressure of the cabin is adjusted to limit hoop stress on the fuselage. The OFV acts as an actuator to achieve this.

2.3.3 Control logic

The FCV regulates the mass flow of fresh air entering the air conditioning pack. Usually the target mass flow is set in such a way that the air volume flow entering the cabin is constant.

The PRV regulates pressure downstream to a fixed value, to keep the connecting pipe within its design limits. Small deviations are not necessarily problematic, but can degrade performance of the FCV.

Temperature regulation inside the cabin is done by a cascading control scheme. For each temperature zone as well as the flight deck, a controller regulates the corresponding ducting temperature, to keep the cabin volume temperature at the target value which is provided by the pilot. The lowest of these ducting temperatures is in turn regulated by changing the pack temperature demand. All other ducting temperatures are then increased to their setpoint by adding hot trim air.

Pack outlet temperature is regulated by opening and closing the inlet of the ram air channel. Opening the channel results in a higher cooling mass flow

(see the blue mass flow in Figure 2.4), decreasing the outlet temperature on the warm side of the heat exchangers. In this work, the ram air channel is not modelled in detail. Instead, ram air massflow is directly given by the controller output.

Cabin pressure is regulated by the OFV, controlling the air mass flow from the underfloor volume to the environment. Target pressure is generated as a function based on altitude.

The speed of the recirculation fans is regulated to keep the recirculation air volume flow rate constant.

2.3.4 Mechanical domain

The model features rotational inertias for the recirculation fans. Compressor, turbine and ram air channel fan include a common rotational inertia, as those components reside on a common shaft.

2.4 Results and discussion

Similar to the preliminary analysis as presented in section 2.2, the complete system model was subjected to a bleed pressure chirp, i.e. a harmonic pressure boundary condition of increasing frequency (0.001 to 2Hz) but constant amplitude of 0.1bar. For the analysis, controllers for both valves in the bleed system were replaced by constant-blocks.

To estimate the significance of a Helmholtz resonance effect in aircraft ECSs, the simulation was repeated. This time the dynamic impulse balance in all simulated pipes was turned off. That means that air was assumed to have no inertia. Without inertia, Helmholtz resonance is not possible, so the difference between both simulations gives some indication about the relative significance of the resonance effect. The resulting mass flows can be seen in Figure 2.5.

It is apparent that there is no significant difference between both simulations. Contrary to the hypothesis of this chapter, Helmholtz resonance does not seem to play a role in the context of aircraft ECSs.

Several reasons for this discrepancy are conceivable: first, the structure of an ECS does not exactly mirror classical Helmholtz resonators. The cabin is pneumatically divided from the underfloor volume, while recirculation air creates a pneumatic loop. Second, the behaviour of the turbomachinery in the air conditioning pack can introduce additional damping. This is especially true for large mass flows. However, the simulations were repeated for strongly reduced mass flows, with no change in behaviour. Third, the ratio between system inlet pressure and cabin pressure was much larger than in the preliminary analyses, resulting in a greater ratio of dissipated energy. Fourth, there is a strong coupling between the system and its controllers. The controllers of the EBAS, which were assumed to interact most strongly to the pressure oscillations

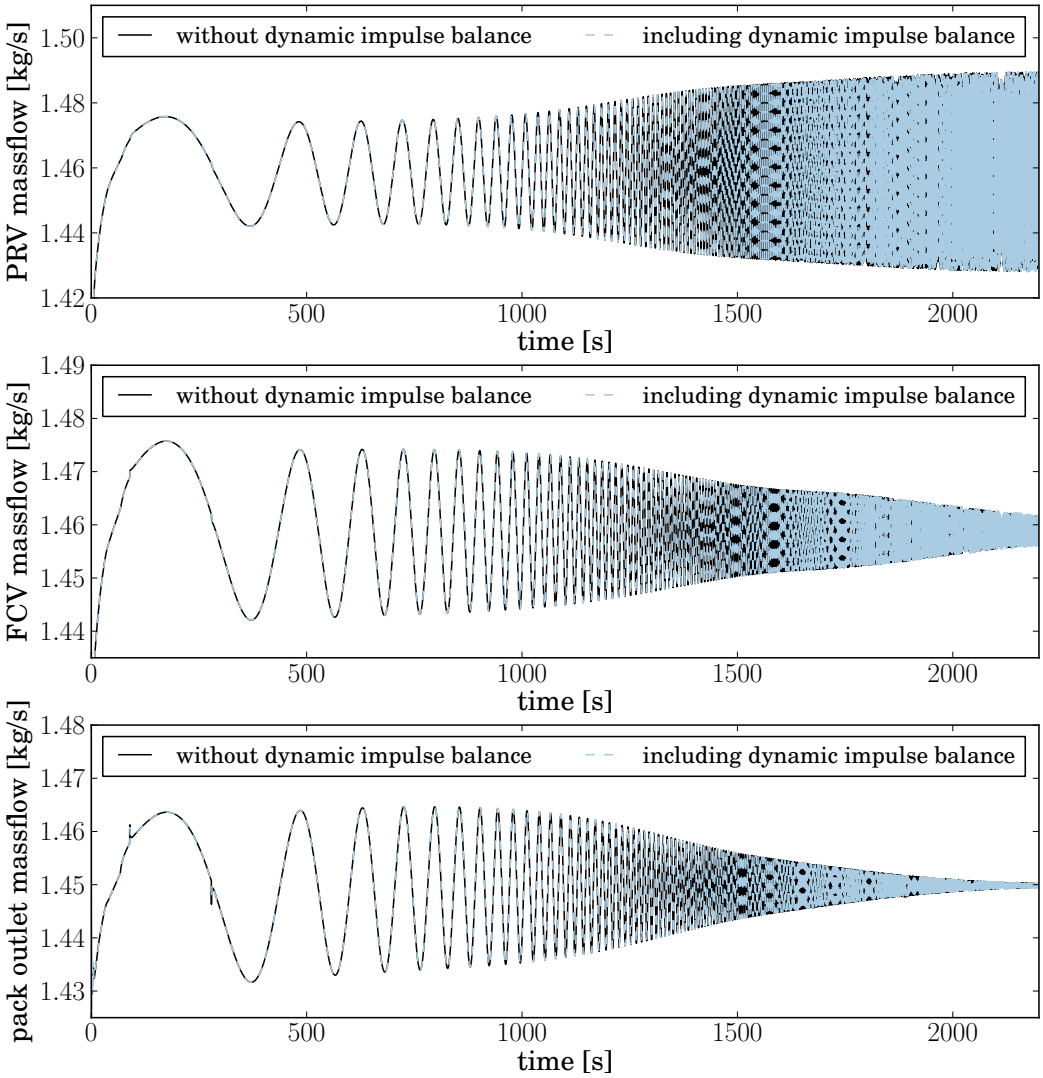


Figure 2.5: *Model response to bleed air pressure chirp*

were turned off in this analysis, but the remaining controllers may still play a role in the results.

While Helmholtz resonance does not seem to be of relevance in the context of aircraft ECSs, other interaction effects between different subsystems might still occur. Aircraft level physical simulation is the most reliable way of predicting this interactions during early design stages.

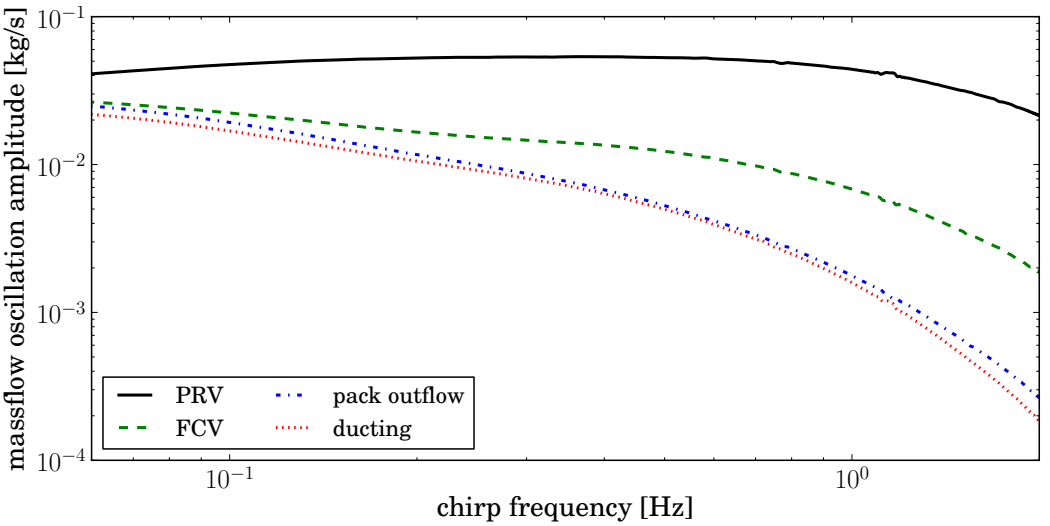


Figure 2.6: *Bode diagram of model response to bleed air pressure*

2.5 Outlook and future work

The work presented in this chapter uses coupled modelling of different subsystems to investigate the dynamic behaviour of aircraft ECSs. However, reality does not stop at the boundary of the ECS. The bleed offtake of the ECS is directly coupled to decrease engine power, with a corresponding impact on the flight dynamics. This effect is even more pronounced in modern aircraft engines with high bypass ratios. Additionally, the flight state is coupled to the ECS via engine pressure, environment temperature and environment pressure. To adequately represent these interactions, a simulation has to include an aerodynamic model.

At the German Aerospace Center (DLR) Institute of System Dynamics and Control in Oberpfaffenhofen, Germany, the robotic motion simulator (RMS) was developed and built [29] based on an industrial robot. One of the applications for this platform is the training of pilots. For this application, an aeroelastic aircraft model was developed in the scope of the *Digital-X* project [30]. Pilots can enter the RMS capsule and put on virtual reality glasses. They see a virtual cockpit and steer the virtual aircraft using a non-virtual sidestick, while the RMS non-virtually accelerates capsule and pilot according to the accelerations of the virtual aircraft.

As part of this ongoing effort, the model presented in this chapter was modified to be capable of realtime simulation and coupled to the flight dynamics of the aircraft. Engine pressure, engine temperature, true air speed, environment pressure and environment temperature are distributed in the local area network (LAN) using the user datagram protocol (UDP). These values are first-order

filtered and set as boundary conditions for the ECS simulation. Additionally, a model of the electric system [31] is integrated in the same way. A picture of the setup can be seen in Figure 2.7.

Figure 2.7: *Coupled thermal, pneumatic, electric and aeroelastic simulations on the DLR robotic motion simulator platform*

At the moment, only monodirectional information exchange is used. Bleed air usage is not fed back into the aircraft engine model. In future upgrades, the setup is intended to be completely coupled. With the addition of all major aircraft energy systems, an unprecedented level of realism for flight simulators can then be achieved.

2.6 Acknowledgments

Many people were involved in the RMS project as described in section 2.5 and shown in Figure 2.7. The team around Tobias Bellmann developed and built the RMS platform. Andreas Seefried organised the project and implemented the links between hardware and the different simulation models. Gertjan Looye, Daniel Milz and Martin Leitner prepared the aeroelastic flight dynamics simulation and adapted it for the RMS. Christian Schallert developed the model of the electrical system.

CHAPTER 3

Countermeasures for bleed system oscillations

"Thank you," Solomon said. "I'm an engine engineer for Masstech."
"Engine engineer," she said. "Seems like it ought to be redundant."
"I always thought thrust specialist sounded dirty," he said.

Daniel Abraham and Ty Franck, The Expanse

3.1 Problem definition

In process control-related applications often relatively simple valve models are used. They are based on flow coefficients and relate mass flow to pressure drop by the use of a quadratic relationship. This helps keeping the system model at a low-order, benefitting understanding as well as control design. Most of the time, these simple models are accurate enough. There are however applications where simple models are inadequate. This can be the case if high accuracy is needed, when choking occurs, or when internal valve phenomena are relevant. Neglecting these cases can lead to unwanted behaviour in the controlled system: internal valve dynamics often contain nonlinearities like stiction, backlash and deadband, which in turn can lead to oscillations [32].

Indeed, according to [33], about 30% of controlled loops in the process industry are oscillating. In [34], 26.000 proportional-integral-derivative (PID) controllers in the process industry are surveyed: 16% are classified as excellent, 16% as acceptable, 22% as fair, 10% as poor and 36% run in open-loop.

In aircraft, self-regulating pneumatic valves (SRPVs) are used to control the pressure and flow rate of the engine bleed air. An illustration of the working principle can be found in Figure 3.1, more detailed descriptions can be found in section 3.2.

SRPVs operate under harsh conditions inside the engine nacelle. Since several SRPVs are operated in-line, their dynamic behaviour has to be tuned to avoid the occurrence of limit cycles. This can be done in situ, but the associated costs are substantial. Being able to predict the system behaviour better during the design phase would reduce those costs considerably, but for a sufficient level of prediction-accuracy a high-fidelity model is needed.

Related research on the modelling side has been done by several authors. In [35] a simple model of an electro-hydraulic valve in *Modelica* and *HyLib* is presented. In [36], a pneumatic drive system is modelled in Modelica, combining pneumatic, mechanical and electronic domains. A free-piston-engine modelled in Modelica is described in [37], containing detailed submodels of several physical domains. In [38] a Modelica model of a pneumatic muscle is presented, combining fluid with mechanical modelling. [39] presents a detailed Modelica valve model for refrigeration cycles.

Some authors also treated the control of such systems. The authors of [40] and [41] developed a control strategy for both air conditioning pack and bleed-air valves. However, the underlying model does not include the non-linear dynamics of the valve actuators. Valve dynamics are modelled as first-order lag instead. [42] developed an adaptive controller for bleed-air pressure control, modelling the valves as a combination of first-order lag and hysteresis. Only a single valve is controlled; interaction effects are not a concern. In the patent by [43], a linear parameter varying (LPV)-like technique is used to control the surge bleed valve of an auxiliary power unit. The technique is not expanded to the complete bleed-air system or to the main engines.

In this chapter, it is shown how high-fidelity multi-physical models of SRPVs can be developed in the equation-based, object-oriented modelling language (EOOML) Modelica and how limit cycle oscillations (LCOs) occurring in real-life industrial applications can be reproduced. Afterwards, suitable controllers and stiction compensation techniques are presented and compared.

3.2 Valve modelling

3.2.1 Functioning principle

The main functioning principle of an SRPV is based on automatic pressure balancing. A small pipe connects the main pipe downstream of the valve with the lower end of the valve actuator chamber. The hydraulic resistance of this pipe dominates the time constant of the valve dynamics. Its diameter is therefore an important design parameter. Inside the chamber, a piston divides

the chamber into two volumes.

The piston is connected to the butterfly valve disk by a mechanical link. In this way, if the downstream pressure increases, the pressure in the lower part of the chamber increases as well and moves the piston upwards. This closes the valve disk, leading to a lower valve mass flow. Depending on the flow configuration in the remainder of the circuit, this usually decreases the downstream pressure, closing the feedback loop.

Additionally, a second control loop is present. By the use of pressure-reducers, vents, and/or small electric regulating valves, air can flow from the upstream part of the pipe to the upper chamber, or from the upper chamber to the ambient. The implementation of the second loop can differ greatly. In this work two implementations are modelled:

1. Pneumatic actuator: a pneumatic system using two pressure-reducers keeps the pressure in the upper chamber inside a predefined interval.

2. Electro-pneumatic actuator: a PID-controller directly imposes the air mass flow from or into the upper chamber.

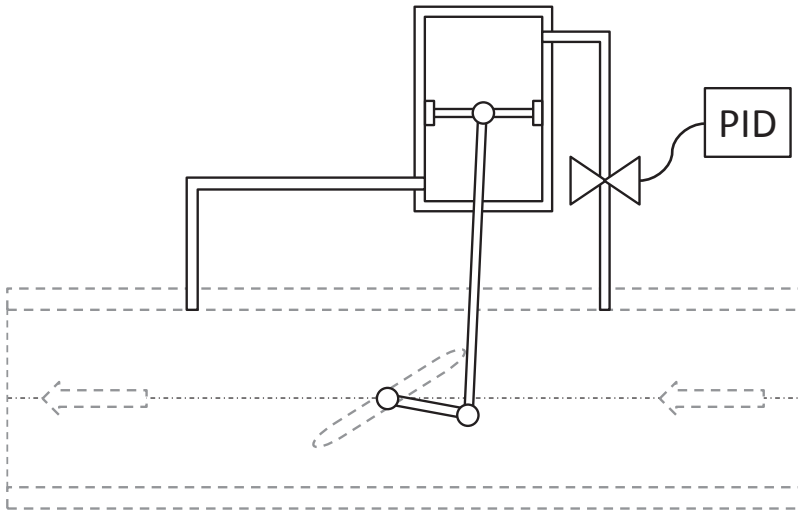


Figure 3.1: *Illustration of a self-regulating pneumatic valve*

For the sake of clarity, the top-level model of the SRPV has been split into two parts: one valve part and one actuator part. The partitioning is illustrated in Figure 3.1, where the valve part is depicted in dashed-grey lines.

3.2.2 Detailed valve model

The valve model calculates the mass flow through the valve depending on the up- and downstream fluid properties and the valve angle. The symbol and the connectors of the valve model are depicted in Figure 3.2. One (rectangular) mechanical multi-body connector is used to connect the valve disc with the valve actuator. Two (round) fluid-connectors are used to connect the valve to pipe models upstream and downstream.

Figure 3.2: *Modelica symbol layer of the valve model*

In an aircraft engine bleed air systems (EBASs), flow velocity is quite high. Physical effects of high-speed compressible flow cannot be neglected [5], so the capabilities of the *Modelica Fluid library* [27] are not sufficient. Thus, as fluid interface, higher-order stencil-based connectors for gas-dynamics as presented in [44] are used. These connectors include far more information than the connectors from the Modelica Fluid library: for a variable number of fluid cells, pressure, temperature and fluid velocity are included. This is illustrated in Figure 3.3. With this information, higher order discretisation schemes of computational fluid-dynamics can be used.

The numerical scheme to compute the desired variables was chosen based on several numerical experiments. A *Godunov* type scheme, *Roe monotone flux* [45], is chosen for its combination of robustness, accuracy and performance.

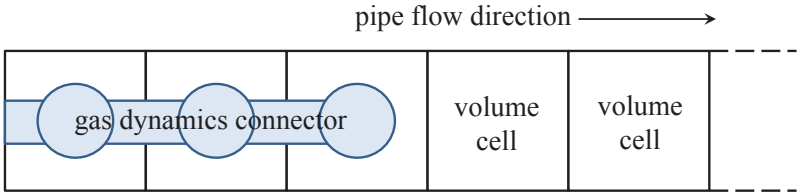


Figure 3.3: *Illustration of the gas-dynamics connector principle, including information about multiple volume cells*

Choked flow effects can occur and have to be taken into account. Therefore the standard calculation using flow coefficients is discarded. Instead, a flow function approach is used, based on an enthalpy-balance and adiabatic state change. The corresponding equations are shown in Equation 3.1, where r

denotes the pressure ratio and us, ds denote upstream and downstream values[1]. If the flow direction is fixed, the equations can be simplified further. The result of this function is multiplied by a factor that is dependent on the valve angle. Note that the flow calculation is symmetric in regard to flow direction. This is not completely accurate, but since the valve is used as a pressure-reducer, back-flow cases are only hypothetically relevant.

$$\begin{aligned} r_{crit} &= \left(\frac{2}{\kappa+1}\right)^{\frac{\kappa}{\kappa-1}} \\ p_{max} &= max(p_{us}, p_{ds}) \\ p_{min} &= min(p_{us}, p_{ds}) \\ r &= p_{min}/p_{max} \\ \psi &= \begin{cases} \left(\frac{2}{\kappa+1}\right)^{\frac{1}{\kappa-1}} \left(\frac{\kappa}{\kappa+1}\right)^{\frac{1}{2}} & \text{if } r < r_{crit} \\ \sqrt{\frac{\kappa}{\kappa-1} r^{\frac{1}{\kappa}} (r^{\frac{1}{\kappa}} - r)} & \text{if } 1 > r > r_{crit} \end{cases} \\ m_{flow} &= \psi \cdot area \cdot \sqrt{\rho_{us} \cdot 2 \cdot p_{max}} \end{aligned} \tag{3.1}$$

Fluids moving through a butterfly valve at high velocities induce a fluid-dynamic torque on the valve disk. This generates an interesting coupling between the fluid and mechanic domains of a valve model. For the calculation of the torque, two approaches are often used: one based on the pressure difference, one based on the fluid velocity. In [46], the different approaches are compared. A classical approach based on pressure difference is used, as the pressure difference is more clearly defined than the fluid velocity in the context of lumped parameter models. Here, the torque T is calculated as:

$$T(\alpha) = K(\alpha) \cdot \Delta P \cdot D^3 \tag{3.2}$$

where K is the torque coefficient, ΔP is the pressure difference, α is the valve angle and D is the valve diameter. A spline-based approach is used to describe the dependency between torque coefficient and valve angle. This data can be generated either from CFD-calculations, measurements or vendor data. A Modelica multibody connector provides the valve angle and feeds back the induced fluid-dynamic torque.

3.2.3 Actuator model

Two actuator models as described in section 3.2.1 are needed for two different implementations of the second control loop. Accordingly, one partial model

[1]The equations can be simplified since the upstream pressure is usually higher than the downstream pressure. But in the context of equation based modelling languages, presented version is more practical, since hypothetical cases have to be considered.

together with two extending models were created. The Modelica diagram of the purely pneumatic version can be seen in Figure 3.4.

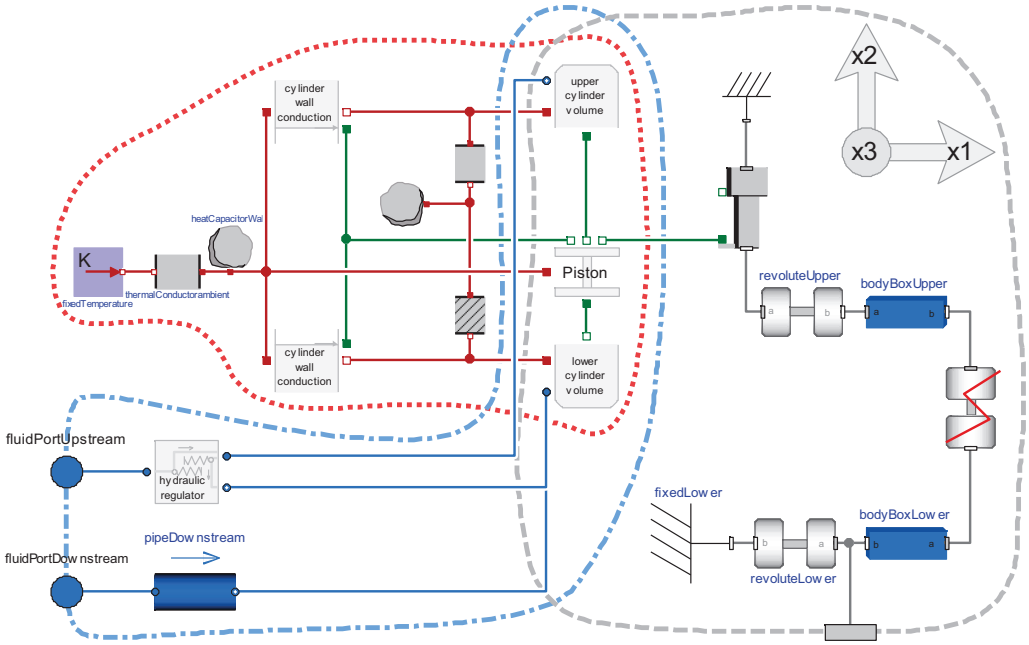


Figure 3.4: *Modelica component layer of the pneumatic valve actuator model*

Three physical domains are significant for the modelling of the valve actuator: the fluid-dynamics inside the chambers, the multi-body mechanics of the mechanism, and the thermal behaviour of the parts. They are connected through the piston and chamber components, where all domains have considerable influence. The domains are indicated in Figure 3.4 by coloured lines: dashed-grey indicates the mechanical domain, dotted-red indicates the thermal domain, and dash-dotted-blue indicates the fluid domain.

Mechanical domain

The core of the mechanical domain is the piston model, where a one dimensional force balance over the piston is calculated, see Equation 3.3. The occurring forces are commented in the following:

$$\begin{aligned} &F_{pressure,upper} + F_{pressure,lower} + F_{constraint} \\ &+ F_{friction} + F_{d'Alembert} + F_{joint} = 0 \end{aligned} \tag{3.3}$$

Pressure forces: the piston model and both chamber models are connected by translational mechanical connectors. In this way, the position and the forces generated by fluid pressure are exchanged.

Constraining forces: based on the construction, the movement allowance of the piston is limited. To represent this, stiff quadratic spring forces are implemented. These come into effect as soon as the end of the stroke is reached.

Friction force: the friction forces between piston and cylinder are mainly responsible for unwanted stiction-effects. Detailed modelling of friction phenomena is therefore necessary. Furthermore, a simple model based on two static and dynamic friction coefficients is numerically unfavourable when the piston position is used as a state. In this work, the Lund-Grenoble (Lu-Gre) friction model is used [47]. It is a detailed model of friction with an internal state that represents the deflection of the bristles (micro-bumps in the material surface). The implementation in Modelica was done according to [48], but instead of rotational coordinates, translational coordinates were used. An example of the trajectory of the friction force over piston velocity can be seen in Figure 3.5.

d'Alembert force: the d'Alembert force, or inertial force, of the piston is calculated by multiplying the second time-derivative of the position by its mass. Of course, this makes the system quite stiff from a numerical point of view, but then, there are solvers of production-quality available to handle stiff systems.

Joint force: the joint force is the linking force between the translational piston dynamics and the planar dynamics of the mechanism. The prismatic joint model of the multibody library provides the interface.

For the dynamics of the mechanism, the *Modelica Multibody library* as presented in [49] is used. With this library, the mechanism can be represented exactly; also an extension to alternative designs can be done with little effort. A nonlinear system of equations remains after causalisation of the complete model, which is downgrading the simulation speed, but inevitable.

Fluid domain

For the air in the valve actuator, high-speed fluid effects can be neglected. Consequently, the Modelica fluid library as presented in [27] is used wherever possible.

Both valve chambers correlate to variable volume models. The governing equations of a variable volume model are a generalisation of the standard volume model equations, and take the form of Equation 3.4, with the density ρ, the

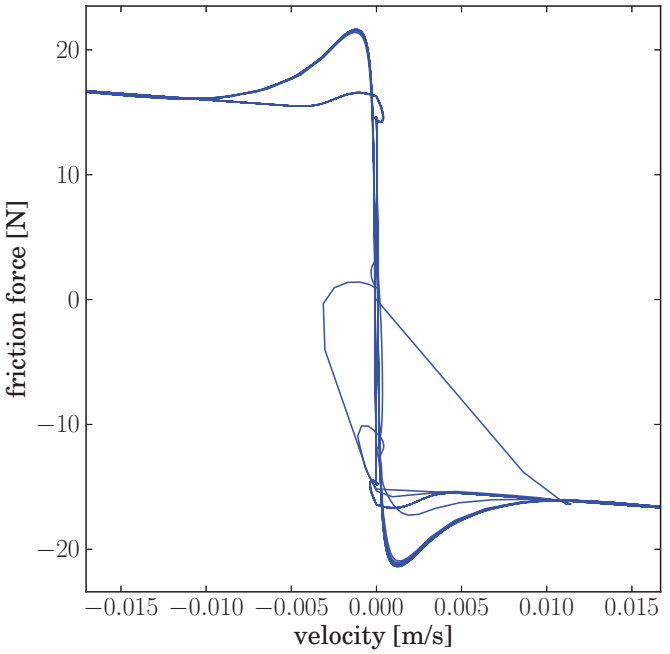


Figure 3.5: *A trajectory (friction force w.r.t. velocity) of the Lund-Grenoble friction model*

volume V, and $\phi \in (1, u, \mathbf{x})$ representing mass, energy and substance balance respectively.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\phi \cdot \rho \cdot V\right) = \sum flow + \sum source \tag{3.4}$$

In the case of energy-balance, mechanical work on the cylindrical chamber volume now creates an interesting interaction between the fluid and mechanical domain. The implementation in Modelica can be seen in Listing 3.1. Note: the actualStream-operator indicates the properties of the fluid crossing the port boundary. The noEvent-operator indicates that no state-events shall be generated when the fluid mass flow changes direction.

Listing 3.1: *Extract of Modelica code for lower variable volume model*

```
//translational mechanics interface
medium.p = - flange.f/area;
pos = flange.s;
volume = volume_0 + area*pos;

//mass balance
mass = volume*medium.d;
der(volume*medium.d) = sum(fluidPort.m_flow);
```

```
//energy balance (dU = dQ + dW)
der(volume*medium.d*medium.u) = sum(fluidPort[i].m_flow *
    noEvent(actualStream(fluidPort[i].h_outflow))
    for i in 1:ninf)
- medium.p*der(volume) + heatPort.Q_flow;

//substance balance
der(volume*medium.d*medium.Xi) = sum(fluidPort[i].m_flow *
    noEvent(actualStream(fluidPort[i].Xi_outflow))
    for i in 1:ninf);
```

Thermal domain

The thermal effects in SRPV systems are largely dominated by the advection in the air. This is obviously already included in the fluid modelling. Nonetheless, conduction through the solid components still has to be modelled if high-fidelity results are necessary.

On the thermal side, the model is structured as follows. The environment is modelled as boundary condition of constant temperature. The actuator cylinder wall and piston are both modelled as thermal masses. A further discretisation is discarded based on the high internal conductivity of the used materials. The energy dissipated by friction is added to the piston wall. Between the fluid volumes and the piston mass, as well as between the cylinder wall and the environment, constant thermal conductances are assumed. Between the fluid volumes and the cylinder wall, the thermal conductance is dependent on the wetted area, which is in turn dependent on the piston position.

As a consequence, a heat-conduction component was composed that connects heat conductivity with the piston position. The remainder was modelled using the *Modelica thermal heat transfer library*, the details of which are described by [50]. In Figure 3.6, the structure of the thermal model is illustrated.

3.2.4 Statistics

Before simulation, the previously acausal model is causalised by the Modelica simulation tool and index-reduction takes place wherever necessary (for more details see [51]). The resulting models of valve and actuator features zero and ten states respectively, and 21 and 161 time-varying variables. After tearing, a nonlinear system of size three remains for the actuator model, caused by the multibody mechanics.

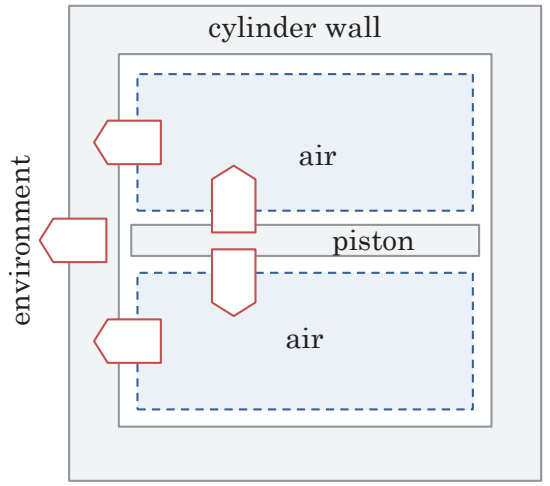


Figure 3.6: *Thermal structure of the valve actuator*

3.3 Cause of limit cycle oscillations

In this section, it is shown how valve friction can cause LCOs and how the effect can be predicted by the presented model. A system is shown, where two valves are used to reduce the pressure in a pipe. The Modelica diagram of the composition can be seen in Figure 3.7. The pipe models are based on the gas dynamics library as presented by [44]. Each pipe component represents a pipe of 20m length and a diameter of 0.1m, totalling at a length of 80m and a volume of around 0.63m^3.

As boundary conditions, the input pressure (left side) is set to three bars, while the output boundary is modelled as a quadratic resistance, normalised to a fluid velocity of $10\frac{\text{m}}{\text{s}}$ at a pressure of 1bar. The valve actuators are run in pneumatic-mode and set to regulate the downstream pressure to 2 and 1bars respectively.

When the composite model is simulated, LCOs occur. These are displayed in Figure 3.8. For both valves, the piston gets stuck at the outmost deflection, until the restoring forces are high enough to overcome the friction forces.

To demonstrate the influence of friction and pneumatic forces on the oscillations, the friction force and the volume of the cylinders were multiplied with a scaling parameter. A two-dimensional sweep of the quasi-steady-state amplitudes and periods over both scaling-parameters is shown in Figure 3.9.

It can be seen that the oscillations are strongly dependent on both factors. Furthermore, the interaction between both factors can be visibly divided into three sections: for friction forces or cylinder volumes below a certain threshold,

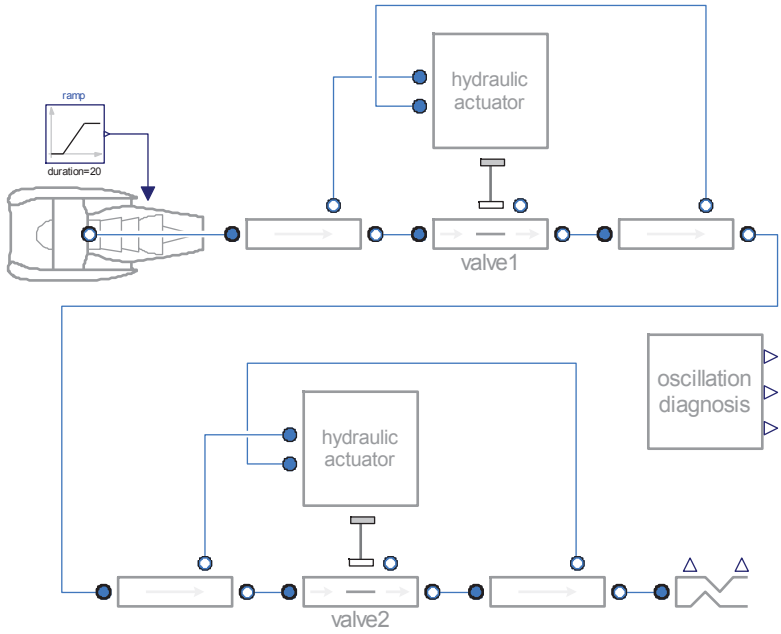


Figure 3.7: *Modelica diagram of oscillation test case*

no oscillation takes place, no matter how large the other scaling factor is. As long as both factors are relatively small, some interactions can be seen.

3.4 Development of a bleed system controller

Most EBASs contain several valves connected in series. In the scope of this work, an architecture using three valves is considered: high pressure valve (HPV), pressure reduction valve (PRV) and flow control valve (FCV). The HPV is purely pneumatically actuated, the PRV additionally features an electronic control loop to regulate downstream pressure, the FCV regulates mass flow downstream while also employing an electronic control loop. The control problem therefore includes two electronic controller outputs, for PRV and FCV. For each control approach, seven variables were assumed to be measured: pressure values upstream and downstream of all valves, mass flow downstream of the FCV, and valve angles of PRV and FCV. This is illustrated in Figure 3.10.

In the aviation industry, each layer of complexity has to be traded against the cost of the necessary certification procedures. Also, added weight and package dimensions have to be considered. Therefore, controllers tend to be as simple as possible for the given performance requirements. Since decentralised proportional-integral (PI)-controllers are the state of the art for bleed-air systems, the approaches proposed in this work are kept as simple as possible.

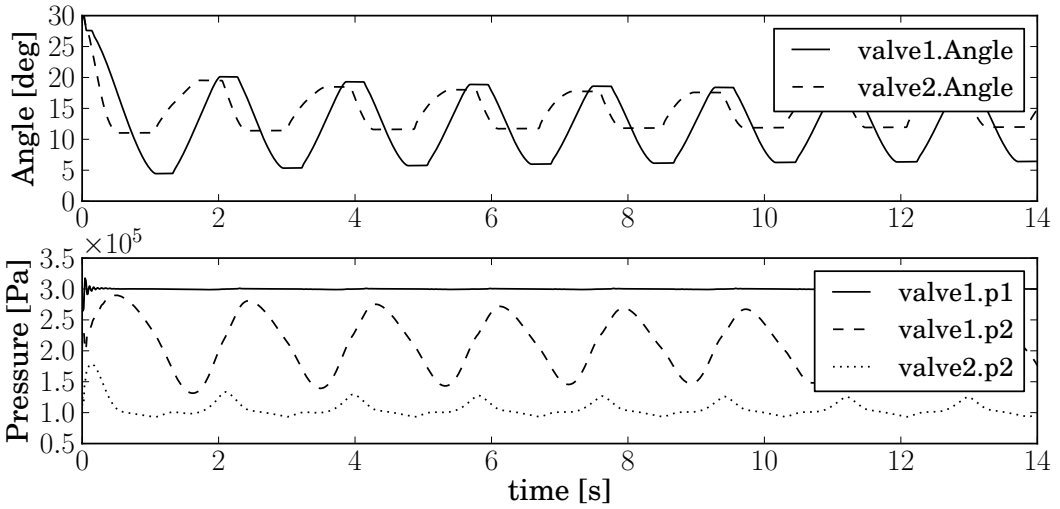


Figure 3.8: *Results of oscillation test case*

Suitable control strategies for bleed-air systems are defined by the baseline control approach as well as an optional superimposed stiction compensation technique. In this work, three control approaches as well as two stiction compensation techniques were implemented, optimised and tested. Baseline controllers can be used without stiction-compensation, resulting in nine combinations. This is illustrated in Table 3.1. Stiction compensation is especially important since oscillations in aircraft bleed air systems seem to originate from stiction effects, and not from aggressive controllers. This can be seen by the shape of the oscillations, according to the work of [52].

compensator	baseline controller →		
none	PI (state of the art)	LQG	MAP
Knocker	PI + K	LQG + K	MAP + K
I-Tuning	PI + I	LQG + I	MAP + I

Table 3.1: *Combinations of baseline controller and stiction compensator*

3.4.1 Baseline controller

PI:

The state of the art is quite simple: both PRV and FCV are controlled by a PI-controller. Manufacturers typically tune the PRV to be much slower than the HPV, and the FCV to be even slower. This is done to reduce interaction effects between the subsystems by frequency decoupling.

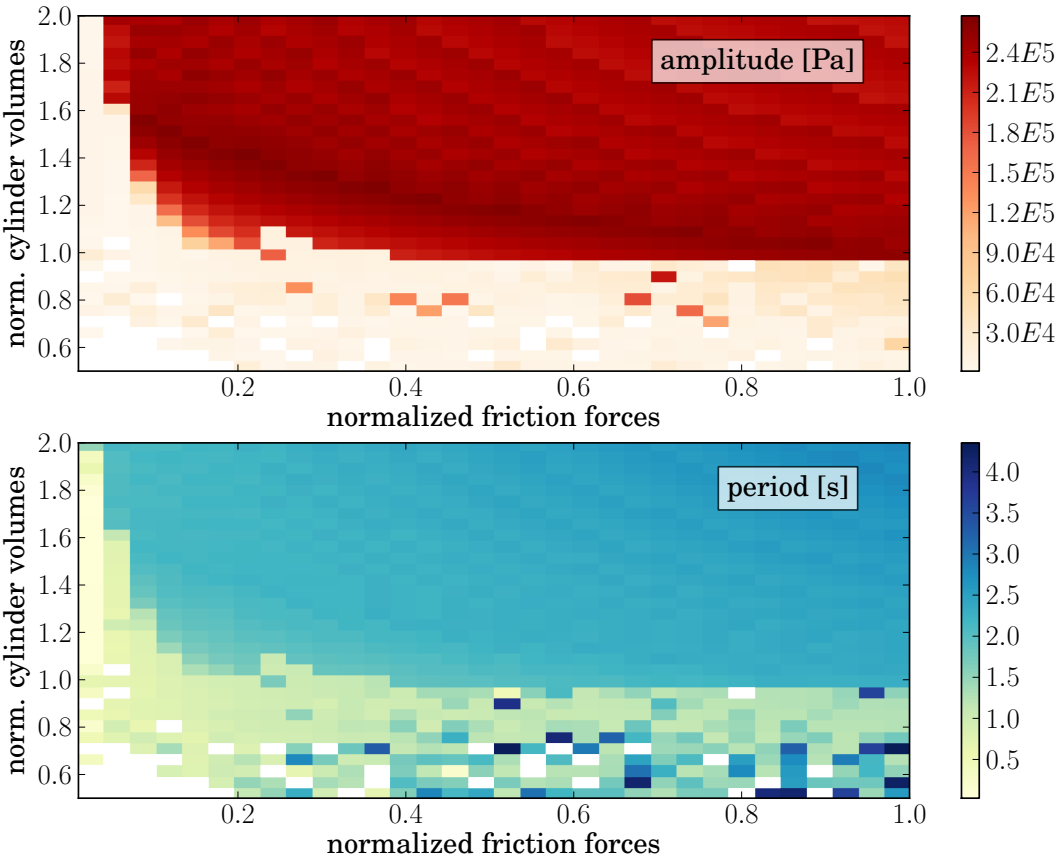


Figure 3.9: *Amplitude and Period of oscillations for scaled friction forces and cylinder volumes*

LQG:

One alternative approach is to use centralised control for the electronically controlled valves. For the development of linear quadratic gaussian (LQG)-regulators a model linearisation was needed, which included two obstacles.

First, linearisation should be done in steady state, but the slope of the piston/cylinder-friction curve at zero velocity is not representative for the slope at typical velocities. Also, friction effects prevented any steady state from appearing. Therefore, the modelled friction was reduced to the viscous (linear) friction term. For this modified model, no oscillations occurred.

The second problem is related to the small control pipe that connects the main bleed mass flow to the lower cylinder volume. During steady state, the mass flow in the control pipe is equal to zero. The pneumatic resistance in the control pipe is roughly proportional to the velocity, at zero mass flow the resistance is vanishing. Any linearisation at zero mass flow does not represent

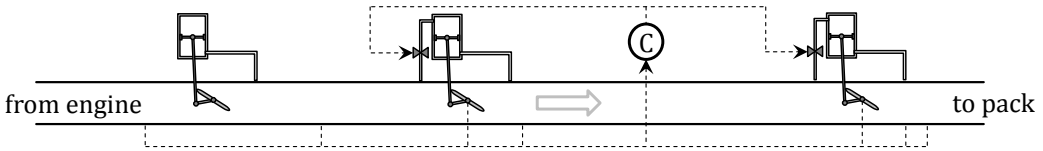


Figure 3.10: *Bleed-air system structure*

the actual system adequately. For linearisation, the pneumatic calculation in the control valve was replaced with a linear dependency based on average absolute mass flows, during normal operation.

After linearisation, the model was augmented with an integrator as described in [53, p. 348] to enable integral action and zero steady-state error. The model was then balanced and model order was truncated to twenty states. This was based on a preliminary analysis, which promised good results between 15 and 30 states.

MAP:

Another approach is to decrease interaction effects using some aspects of feed-forward control for the PRV. The steady-state PRV angle can be computed from exogenous variables only: in a small offline study, the resulting steady state valve angle (friction turned off) was computed for varying engine pressures. In the online system, the target angle is simply interpolated from those results. A simple three-point controller is used to set the valve angle to the interpolated target. The approach is called *MAP* since the inlet pressure is mapped to the target valve angle.

The hysteresis in this controller has to be chosen big enough to avoid limit cycles based on valve friction. This was tuned in some offline experiments. The amplitude of the controller output was set to the electronic actuator limit.

One should note that for this concept, a small hardware change is necessary. Without modifications, even with feed-forward electronic control, pneumatic self-actuation would still result in valve interactions. In this concept, the pneumatic self-actuation is switched off. A small on-off valve is integrated in the resistance between the downstream flow and the lower cylinder chamber. This valve is closed in normal operating mode, as a result, the actuator is influenced by electronic control only. The failsafe position of the valve is open, activating the self-actuation in pneumatic backup mode.

Aside from small aerodynamic effects from the moving air in the main pipe on the valve disc, there is no remaining interaction between the PRV and the other valves. The FCV is controlled with a PI-controller. With this concept, zero steady state error for the first controlled variable (pressure downstream of the PRV) cannot be guaranteed. This is usually not a substantial problem, as long as the pressure keeps below the pipe stress limits.

3.4.2 Stiction compensation

The two classic approaches to combat valve stiction are dithering and impulsive control. However, both approaches are not suitable for pneumatically actuated valves, where the actuator input is essentially first-order filtered [54].

In this work, two stiction compensation approaches are compared that are in principle suitable for valves with filtered inputs. All baseline controller approaches were also implemented as is without any superimposed compensation, resulting in nine combinations in total.

Knocker:

This technique was first proposed by [55]. The controller output is superimposed with pulses that have the same direction as the first derivative of the controller output. This is illustrated in Figure 3.11.

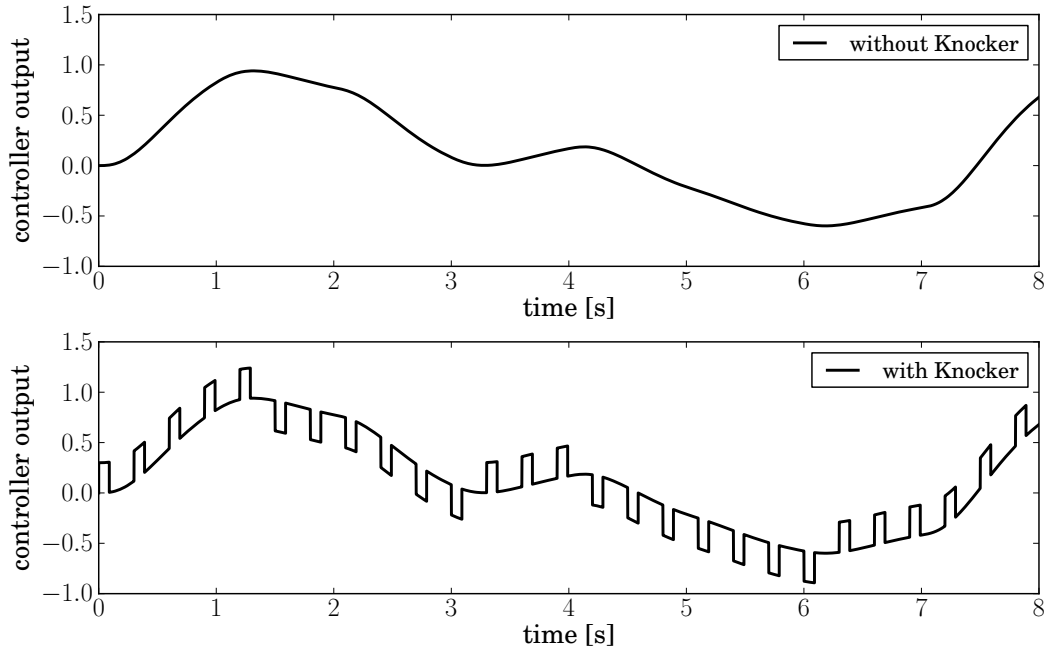


Figure 3.11: *Knocker approach for stiction compensation*

Pulse height and pulse frequency are tuning parameters that have to be adapted to the plant. The basic idea is that the energy content of a pulse is enough to overcome static friction.

I-tuning:

This technique was first proposed by [56] (as cited by [54] and others). A variable gain is inserted prior to the integral element of the controller. The gain

is calculated as

$$\alpha = [1 - exp(-ae^2 - b\dot{e}^2)]^2. \tag{3.5}$$

with the control error e and tuning parameters a, b. Essentially, integral action is continuously decreased as long as both control error and its first derivative approach zero. This is illustrated in Figure 3.12.

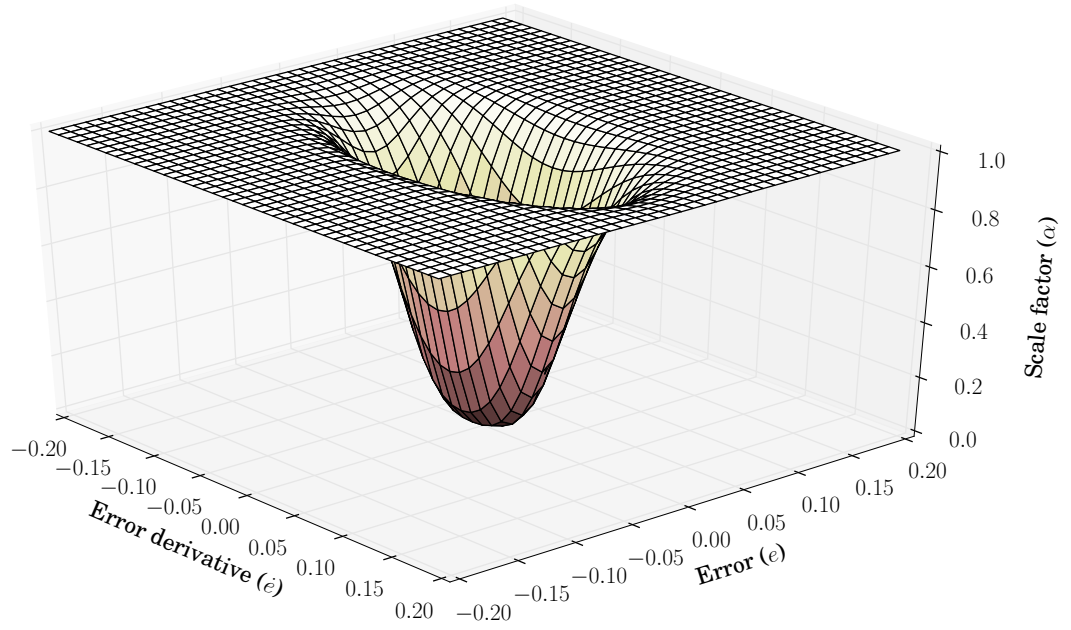


Figure 3.12: *I-Tuning approach for stiction compensation*

3.4.3 Evaluation criterion

Before the strategies to control bleed-air systems were evaluated, a suitable evaluation criterion was defined. This enables quantifiable ratings of control strategies.

The used criterion penalises valve movement and control error. It is divided into two parts. The first part integrates the absolute control error for both controlled variables: pressure downstream the PRV p, and mass flow downstream the FCV $\dot{m}$. Both are normalised against their target values:

$$J_1 = \frac{1}{T} \int_{t_1}^{t_2} \left| \frac{p_{error}}{p_{target}} \right| + \left| \frac{\dot{m}_{error}}{\dot{m}_{target}} \right| \mathrm{d}t. \tag{3.6}$$

The second part integrates the absolute velocities of the valve angles ϕ_i. This is a measurement for the total valve movement during simulation time:

$$J_2 = \frac{1}{T}\int_{t_1}^{t_2} |\dot{\phi}_1| + |\dot{\phi}_2| + |\dot{\phi}_3| \, \mathrm{d}t. \tag{3.7}$$

Both parts are combined using the weighted 2-norm. The weighting parameters w_i were chosen in such a way that J_1 and J_2 would be equal for the state of the art system before optimisation:

$$J = \sqrt{w_1 \cdot J_1^2 + w_2 \cdot J_2^2}. \tag{3.8}$$

3.4.4 Optimisation and validation

For each of the nine combinations of baseline controllers and stiction compensators as illustrated in Table 3.1, a Modelica model of the corresponding control structure was realised as well as a tuning script using *MATLAB*.

Each combination was optimised using the pattern search algorithm [57]. Any function evaluation involved a model simulation. The chaotic nature of the physical system hindered optimisation progress. To get meaningful results, EBAS models have to be simulated for a large number of oscillation cycles, and a broad range of boundary conditions. For reasons of available computational power, a compromise of 100s of simulated time was used, resulting in around 600s of computation walltime[2].

During simulation, the engine pressure was ramped up from 5 to 20bars. To keep the comparison fair, each of the nine combinations was granted 250 function evaluations. Using parallelisation, total walltime was reduced to around two days. For comparison reasons, the state of the art architecture was optimised as well since differences in behaviour between the model and the actual airplane could not be ruled out.

The nine optimised architectures were subjected to an independent validation run. This time, simulated engine pressure was changed during the simulation, as well as EBAS outlet-resistance. During validation, all measurements are superimposed by white noise, using the Modelica Noise library by [58].

3.5 Results and discussion

In Figure 3.13, plots of the FCV mass flow are shown together with the FCV control variable.

The combination of PI-control and no stiction compensation can be seen as the baseline architecture. Rectangle-shaped oscillations are persistent with varying amplitude. The LQG-approach results in oscillations of similar shape, but there are oscillation-free periods. The control variable is much lower on average. For the MAP approach, the shape of the oscillations changes

[2]Walltime describes the time needed for a computer to simulate a model. In other words: If your computer is twice as fast, simulated time stays the same, but walltime gets cut in half.

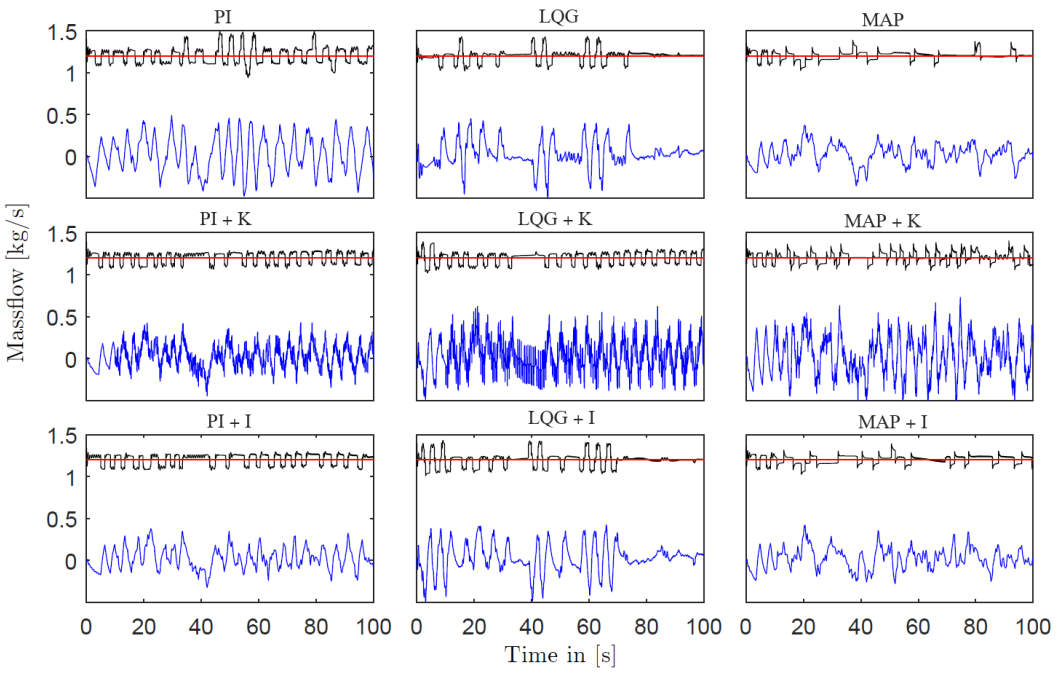


Figure 3.13: *Validation results for different control architectures: FCV mass flow (black), constant target mass flow (red), and scaled FCV control variable (blue)*

considerably. Instead of the usual rectangles, the predicted mass flow features steps followed by decay-like behaviour. The amplitude of the oscillations is varying, but in general smaller than the amplitude of the PI- and LQG-approaches.

Adding a Knocker compensator results in some kind of regression to the mean. The behaviour of the PI-controller gets improved by decreased maximum oscillation amplitudes. The behaviour of the other approaches worsens by a higher occurrence of oscillations. Adding an I-Tuning compensator clearly improves the behaviour of the PI-controller. For the other approaches, the differences are small.

To quantify those results, the evaluation criterion as defined in section 3.4.3 was evaluated for each combination of baseline controller approach and stiction compensation technique. The values after optimisation and validation (shown after the arrow) are shown in Table 3.2. All values are given in percent and are normalised to the value of the criterion with the state-of-the-art controller during validation.

Generally, performance drops during validation as expected. The drop is greatest for Knocker-augmented strategies, implying a low robustness for Knocker-based approaches.

The best, or lowest value after validation corresponds to a combination of

	after optimisation			after validation		
	PI	LQG	MAP	PI	LQG	MAP
none	69	54	34	100	64	56
Knocker	32	43	29	81	75	59
I-Tuning	32	37	37	73	61	54

Table 3.2: *Evaluation criteria after optimisation and validation (less is better, normalised to state of the art)*

MAP and I-Tuning. The criterion is 46% smaller than the state of the art upon validation. MAP standalone without any stiction compensation nearly reaches the same value. A simplicity-based argument can be made that the small improvement given by the additional stiction compensator might not be worth the additional complexity. Also, the I-Tuning compensator adds an additional source of nonlinearity to the system, making it even harder to use frequency-based analysis techniques.

Centralised control by LQG shows promising results as well with an evaluation criterion 36% smaller than the state of the art. If zero steady state error for both controlled variables is needed, or if the proposed hardware change for the MAP approach is prohibitive, then centralised control is a viable option. Again, adding an I-Tuning stiction compensator improves the behaviour, but not much. The considerations mentioned in the last paragraph still apply.

Performance of the Knocker compensator seems relatively poor during validation, but is much better during optimisation: the best value of the evaluation criterion after optimisation corresponds to a combination of MAP and Knocker compensator. This suggests poor robustness properties of the Knocker concept in the scope of this application.

3.6 Outlook and future work

In this chapter, suggestions for better bleed system controllers have been explored. Before implementation on an industrial scale, i.e. for new aircraft projects, technology readiness has to be increased. Also, it is expected that the performance of bleed air systems can be further improved using global optimisation of both system hardware and controller. This is motivated by the following quote taken from [59].

> *In the application of automatic controllers, it is important to realize that controller and process form a unit; credit or discredit for results obtained are attributable to one as much as the other. A poor controller is often able to perform acceptably on a process, which is easily controlled. The finest controller made, when applied to a miserably designed process, may not deliver the desired performance.*

> *True, on badly designed processes, advanced controllers are able to eke out better results than older models, but on these processes there is a definite end-point which can be approached by the instrumentation and it falls short of perfection.*
>
> Ziegler & Nichols, 1943

Here, a workflow is proposed for the control system design to accompany a typical aircraft development process. It is illustrated in Figure 3.14.

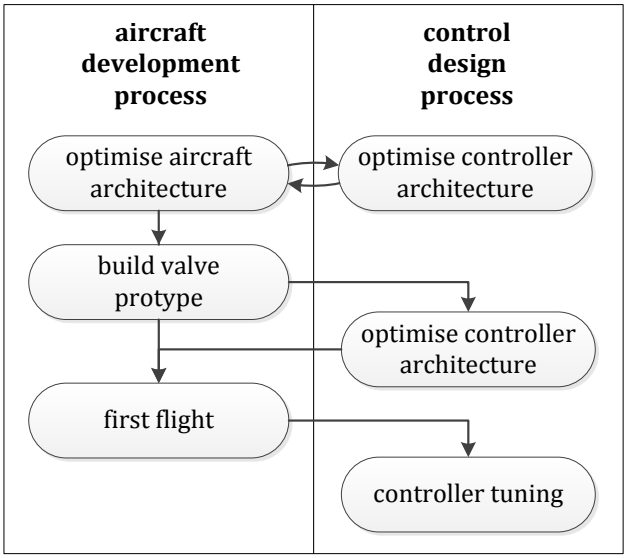


Figure 3.14: *Proposed workflow for the development of bleed system controllers*

For the best results, control design has to be integrated into architecture optimisation, for example using a two-level approach as described in [5]. In this approach, during the evaluation of architecture candidates (first level), suitable control systems are generated (second level). But even if this is not possible, there is a number of physical degrees of freedom that can be tuned during control design without major repercussions on nominal architecture performance or system integration. For instance, the pneumatic resistances of the self-regulation pipes as shown in Figure 3.1 can be tuned for a better interaction between pneumatic self-actuation and electronic control.

As soon as the system architecture is frozen, prototypes for the actual valve are manufactured. These should be used to conduct experiments, providing the friction model with more accurate data. The model can then be validated against valve assemblies, possibly augmenting the model to fit the measured

behaviour of the actual valves and resulting in a grey box model. At this time, the control system design procedure can be repeated. Depending on the difference between old and new model, the optimal controller structure might deviate from the original plan.

After the first maiden flight, careful tuning of the controller parameters might be sensible, if the expected return on investment in gained performance is deemed worth the effort.

3.7 Acknowledgments

Andreas Schröffer implemented the MATLAB script for the optimisation of the different controller variants in the context of his Master thesis at the Technical University of Munich. He also helped with the analysis of the results.

CHAPTER 4

A concept for aircraft temperature regulation

4.1 Problem definition

Aircraft cabin innovations are a major opportunity for airlines to stay competitive [60]. The typical aircraft cabin has a life cycle of 5-10 years, compared to an aircraft life cycle of around 30 years [61]. For this reason, seating configurations can vary considerably over an aircraft life cycle. These configurations correspond to different heat loads per unit cabin length: imagine a sparsely populated first class, followed by a business class, full of business people producing hot air. These differences in heat load cannot be compensated by the control system, if the cabin temperature control zones do not correspond to the borders of the seating classes. This is illustrated in Figure 4.1.

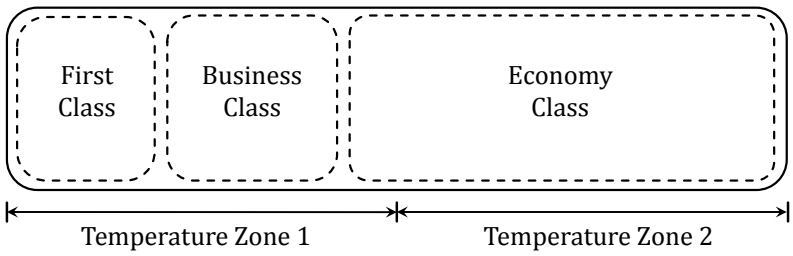


Figure 4.1: *Illustration of seating classes and temperature control zones in an aircraft cabin*

An obvious solution is to increase the number of control zones to a point where the regulation is fine-grained enough to cover all possible seating class layouts. A larger number of control zones would also synergise with concepts that offer individual temperature zones for each passenger, as presented in [62, 63, 64]. This is however not possible using a conventional ducting architecture as illustrated in Figure 4.2.

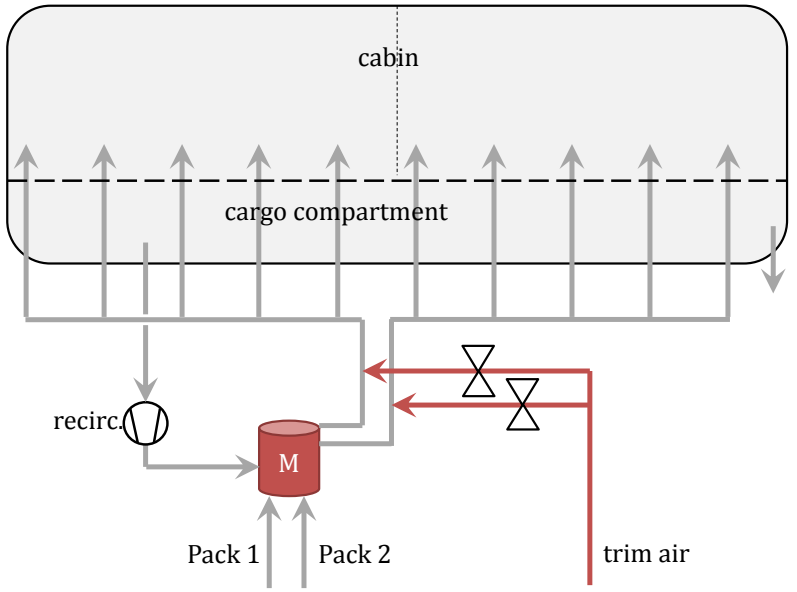


Figure 4.2: *Conventional cabin temperature regulation architecture (two temperature zones)*

Conventional architectures are structured as follows: air from the mixing chamber is divided into main ducts leading to the different temperature zones. Hot trim air is added individually to regulate the temperature for each temperature zone. Riser ducts split up from the main ducts to distribute the air into the respective temperature zone. It is difficult to scale up the number of zones with this architecture since the total amount of ducting increases significantly, as is illustrated in Figure 4.3.

As more and more zones are added, the number of parallel main ducts goes up, resulting in increased weight and installation space - features that are at a premium in the aerospace industry.

4.2 Concept

In the proposed concept, two main ducts each span the complete cabin length. They are fed by two distinct mixing chambers, which are in turn fed by different air conditioning packs. The *warm* mixing chamber ducts warm air to the corresponding warm main duct, the *cold* side behaves accordingly. At the

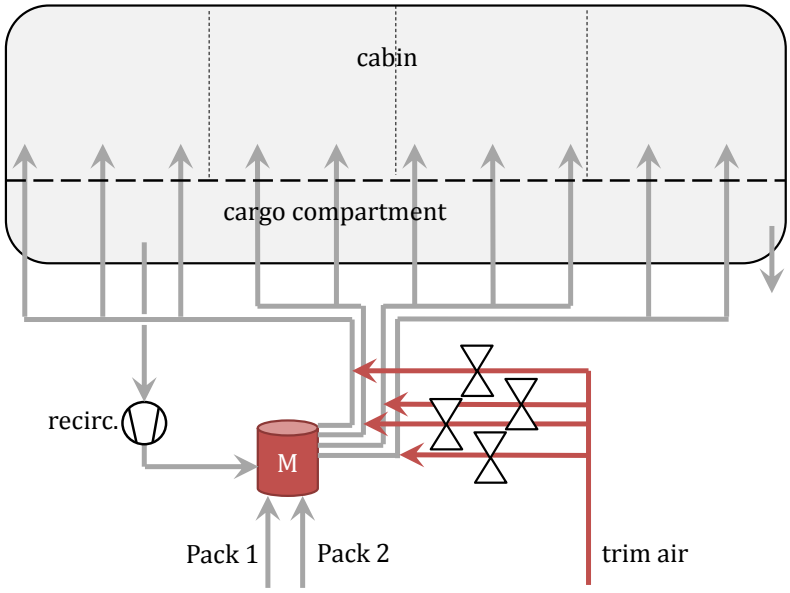


Figure 4.3: *Conventional cabin temperature regulation architecture (four temperature zones)*

different cabin temperature zones, air blown into the cabin is locally mixed from both main ducts. Recirculated air from the cargo compartment is split up and ducted into both mixing chambers. This is illustrated in Figure 4.4.

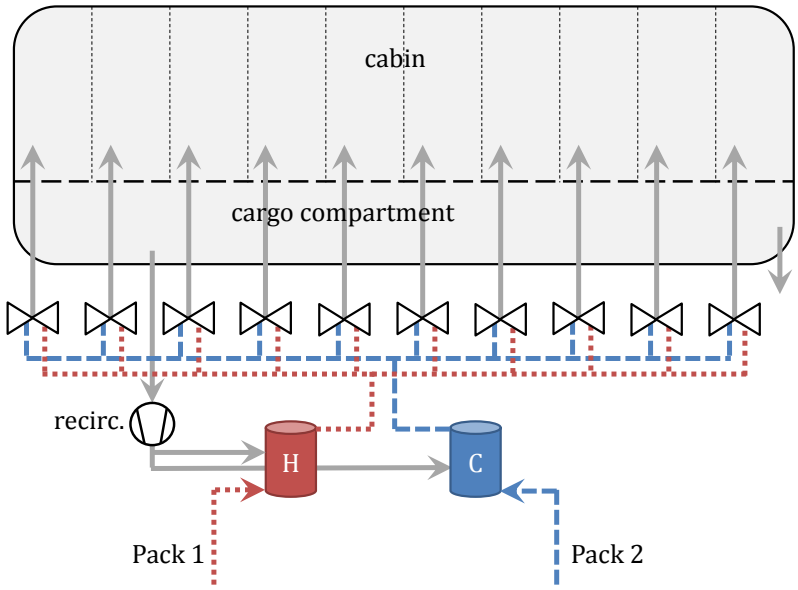


Figure 4.4: *Proposed cabin temperature regulation architecture (ten temperature zones)*

With this architecture, a number of valves are needed for each temperature

zone. Several variants are conceivable:

1. For convection-based/displacement-ventilation, where air is blown into the cabin through distributed inlets at the cabin floor, the architecture can be realised with one mixing valve per temperature zone, located along the longitudinal axis (one valve per zone).
2. Like 1, but augmented with a second control valve connected in series to regulate the mass flow into the control zones (two valves per zone).
3. Like 2, but instead of a mixing valve and a mass flow control valve, two mass flow control valves connected in parallel can be used, one for warm air and one for cold air (two valves per zone).
4. For mixing/mixed ventilation, where air is blown into the cabin at high velocity through inlets above the hat-racks, mixing valves both port and starboard would be necessary (two valves per zone).
5. Like 4, but augmented with a second control valve per side connected in parallel to regulate the mass flow into the control zones (four valves per zone).
6. Like 5, but instead of a mixing valve and a mass flow control valve per side, two mass flow control valves connected in series can be used, one for warm air and one for cold air (four valves per zone).

The proposed architecture offers the following advantages: no access to trim air is needed at the location of the mixing chambers and beyond, therefore no heat-resistant ducts from the engines to the mixing chamber are necessary. Abandoning trim air also enables a redesign of the bleed-system, allowing for more aggressive optimisation and savings regarding performance and weight. A large number of temperature zones can be realised without a significant increase in ducting weight or installation space. Each zone requires additional valves, but since the total mass-flow stays constant, the mass-flow per valve decreases, allowing for smaller and lighter valves. The blown air is mixed together near to the actual air outlets. This is in contrast to the conventional architecture, where the temperature regulation takes place directly next to the mixing chamber: changes in temperature are therefore delayed by the thermal inertia of the ducting. With the proposed architecture, the amount of ducting between temperature regulation and air outlets is smaller, resulting in decreased thermal inertia and faster temperature response.

4.3 Modelling

A good simulation model is the basis for all subsequent development steps. The following requirements apply in the context of this work (from most to least

important):

Development time: the time needed to plan, develop, and test the model shall be short.

Unity: if possible, all requirements shall be met in a unified model. Having several versions of a model often results in a significant increase in development time as well as project complexity.

Linearity: small perturbations around the design point shall result in linear model behaviour. Model inputs that are connected to saturating actuators shall be scaled symmetrically around zero.

Accuracy: the simulation model shall include all major physical effects. Deviations from reality should be small enough to be irrelevant for the subsequent development steps.

Robustness: the model should predict accurate transient responses for boundary conditions that are far from the design conditions.

Simulation speed: simulation of the model shall be fast. Numerical stiffness shall be avoided if possible.

Size: total size of the model shall be small. This includes several metrics like the number of variables, number of states and lines of code. A small model decreases development time and increases comprehensibility.

These requirements partially contradict each other. Some balancing can be done using multi-objective optimisation like shown in [7], but ultimately, some arbitrariness remains. The decision was to keep the model as simple as possible, with the exception of two physical effects that posed challenges for the control design: the first effect is the thermal interaction of air between the different cabin zones. The second effect is the pneumatic interaction of air in the asymmetric ducting system. Also, a high-fidelity medium model for moist air was used, since both the resulting nonlinearity and the additional modelling effort are small, but moist air effects can be important for cabin climatization. The complete model structure is shown in Figure 4.5 and presented in the following.

Six subsystems were created for air conditioning packs, mixing chambers, ducting system, cabin, heat sources and recirculation system. These subsystems were in turn composed from simple components like flow resistances or volume elements. The model structure was limited to those three layers of system, subsystem and components. Inheritance was avoided, based on the results as described in [13]. The Modelica Standard Library (MSL) [65] was used as much as possible to save additional modelling time.

All subsystems included an integer parameter n, denoting the number of discretised volume elements in the cabin. In this way, the scalability of the

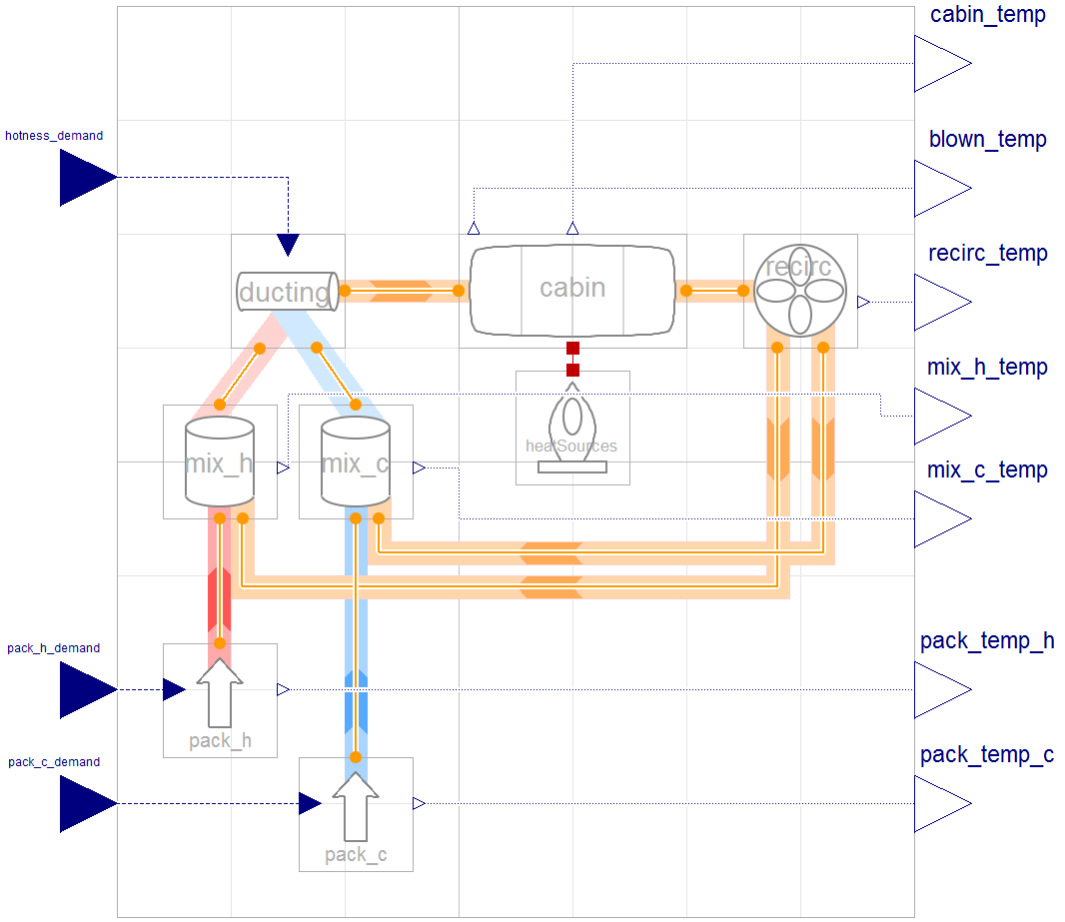


Figure 4.5: *Top level view of temperature regulation model*

concept can be tested later without additional modelling effort. Of those subsystems, ducting and cabin are especially interesting from a modelling perspective. These are presented in the following.

4.3.1 Ducting

As illustrated in Figure 4.4, air is ducted from the mixing chambers into the cabin via a network of ducts. This network is asymmetrical and interactive with regard to the cabin temperature zones. If a large amount of cold air is needed for the centre temperature zones, the effective hydraulic resistance from the cold mixing chamber to the outer temperature zones increases. For controller synthesis and concept validation, this effect has to be modelled.

This was done using vectorised flow elements together with customised connect-statements. The implementation is shown in Figure 4.6. Not shown

are the parameters for the individual air resistance components. These are also dependent on the discretisation parameter since for example the length per pipe is not constant.

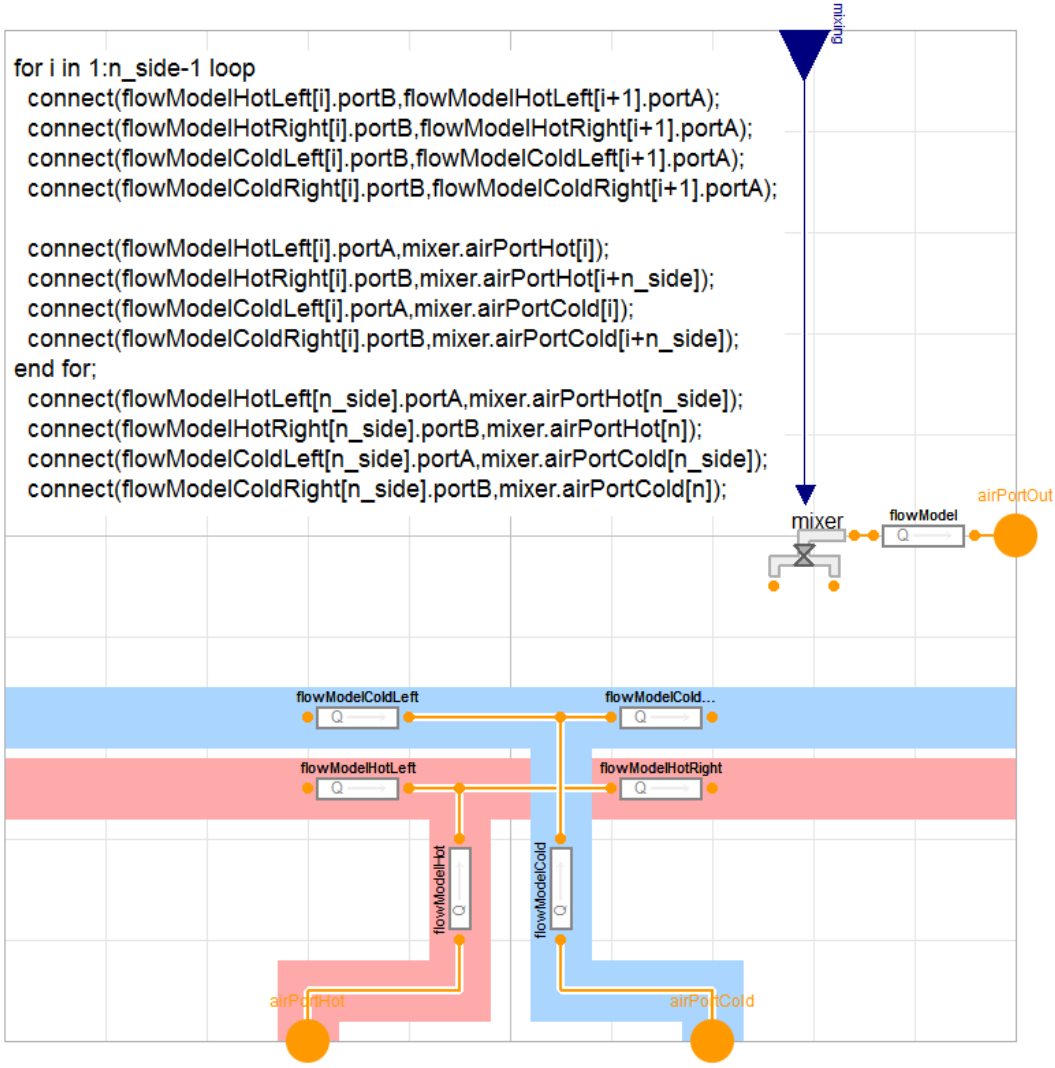


Figure 4.6: *Ducting subsystem model*

4.3.2 Cabin

The flow configuration inside the aircraft cabin is complex and can only accurately be determined by experiments or computational fluid dynamics (CFD)-calculations. However, for the evaluation of the presented concept, a low-order approximation suffices. The cabin is divided lengthwise into an adjustable number of volume elements, which remains constant during each simulation. These

elements are directly connected to enforce pressure equalisation. Fluid volume elements can directly be connected in Modelica, at the cost of nonlinear systems of equations stemming from the application of index reduction algorithms to the resulting index-2 system. This cost is however preferable to the alternative, where the very small flow resistances between cabin volumes leads to a very stiff simulation model.

If no equalisation mass flows occur, there is still some amount of thermal equalisation caused by diffusion. This is modelled using thermal resistance elements, coupled between the volume elements. They were parameterised according to empirical experience.

Again, the subsystem was realised using a combination of vectorised elements and customised connect statements. The implementation is shown in Figure 4.7.

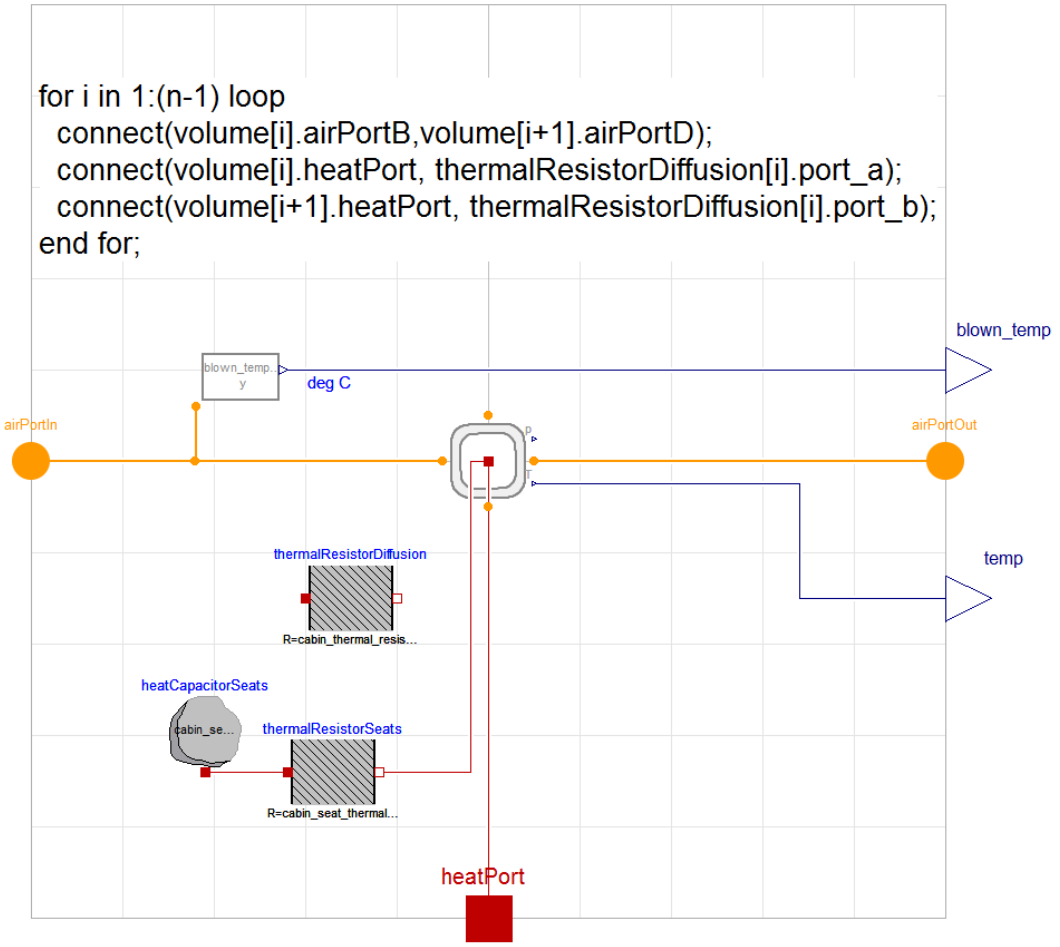


Figure 4.7: *Cabin subsystem model*

4.4 Pack energy consumption

The concept presented in this work requires a cold and a hot air reservoir (see Figure 4.4). Each of these reservoirs is supplied by a separate air conditioning pack. Conventional architectures duct the cooled air from the packs to a common mixer unit (see Figure 4.2). The packs therefore condition the fresh bleed air to the same pressure and temperature. The proposed concept now claims an asymmetrical air conditioning in terms of temperature. The pack would then run in different conditions compared to conventional in-service packs. About 2-3% of the whole energy consumption of a conventional civil aircraft applies to the environmental control system (ECS) [66]. Thus an energy analysis of the deviating pack operation is necessary.

4.4.1 System description

The key part of the ECS is the *air generation unit* (also called *pack*). The pack conditions the air flow in terms of temperature, pressure and humidity. Usually there are two packs installed in an aircraft. Conventional systems use engine bleed air as the power source. The bleed air is drawn off from the compressor stages upstream the combustion chamber. Provided at high temperature (around 220°C) and high pressure (around 2.5bar), the air must be conditioned before it is distributed into the cabin. First the air flow is led to the pack where it is cooled and dehumidified. It passes several heat exchangers, a compressor, a turbine and valves before the flow is led to the mixing unit. The ram air enters the pack from the ambient and passes a water injector, two heat exchangers and a fan. All of them are installed in the ram air channel. The ram air is used as a heat sink.

Figure 4.8 illustrates the Modelica diagram layer schematic of the air generation unit that is used for the energy analysis in this work. It includes a conventional three-wheel bootstrap-cycle, driven by bleed-air. Three different flows are considered: the bleed air arises from the compressor stage at the engine, passing at first the pneumatic distribution device before it enters the ozone converter. Inside the primary heat exchanger the hot air is cooled against the ram air flow. Before entering the compressor stage, a part of the air flow is separated and bypassed through the temperature control valve. Downstream the compressor stage the heated and compressed air is cooled a second time inside the main heat exchanger against the cold ram air flow. Here the most intense heat exchange takes place due to large temperature differences. The air flow now enters the hot side of the reheater and is cooled again before its temperature is further decreased inside the condenser in order to dehumidify the air flow and prevent downstream conditions from reaching the saturation point. This configuration of the three wheel bootstrap cycle uses the concept of high pressure water separation. In case of condensation, the free water is separated

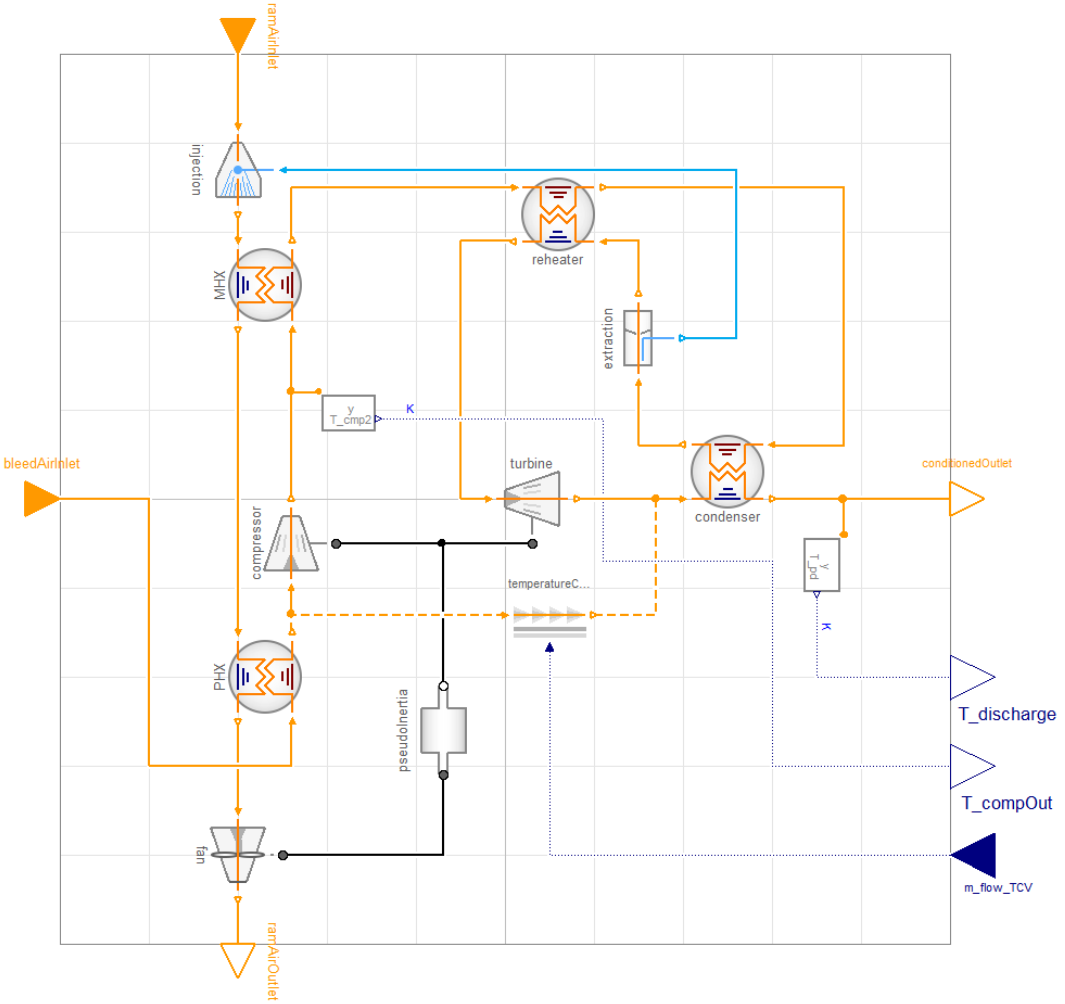


Figure 4.8: *Diagram layer of a pack system model with three wheel bootstrap cycle*

in the water extractor and carried to the injector located at the beginning of the ram air channel. The dehumidified air flow now passes the reheater a second time, this time at the cold side where it is reheated against its upstream air flow. Inside the turbine the air is expanded to a sufficient pressure level. Concurrently the temperature decreases significantly below ambient conditions. At this point the air reaches its coldest condition. Meeting the separated air from the temperature control valve, the flow gains a higher temperature und finally is heated inside the condenser again before it leaves the pack to the mixing unit.

The second flow occurring is the ram air flow. It functions as a heat sink and enters the aircraft through inlets mounted on the aircraft fuselage. The amount of air flow can be controlled by flaps installed at the inlet and outlet of the

ram air channel. Water from the water extractor is now injected into the ram air flow where it evaporates and subsequently the temperature of the ram air flow is decreased. The cool air passes successively the main heat exchanger and primary heat exchanger before it leaves the ram air channel to the ambient. In ground operation the ram air flow is driven by the ram air fan that is mounted on the same shaft as the compressor and turbine. The components shown in Figure 4.8 are taken from an ECS library [67].

4.4.2 Energy analysis

The proposed cabin temperature regulation concept is based on asymmetrical pack discharge temperatures for each pack. Therefore the energy analysis was performed for a wider range of discharge temperatures. Two identical packs were assumed so that simulations were carried out using the pack Modelica model shown in Figure 4.8 for a range of discharge temperatures varying from -30°C to 20°C. The mass flow and the discharge pressure were kept constant. Two control laws are implemented in the model. One that keeps the discharge temperature at the defined value by regulating the bypass mass flow through the temperature control valve and another law that limits the compressor outlet temperature by regulating the ram air mass flow.

The three wheel bootstrap cycle is a self-containing air generation unit, i.e. it does not need any additional power source to run the turbo components. This is realised by the turbine that is driven by pneumatic power of the bleed air. It is assumed that the bleed air is constantly provided by the engine, independent of the operation of the pack. However, the ram air flow changes due to different operating points and causes different amounts of aerodynamic drag. It is therefore directly linked to the pack discharge temperature. For the energy analysis, the variation of occurring drag caused by the ram air is calculated for each operating point. The drag of both packs is summed up and displayed with the average temperature of both packs and their anomaly in temperature. *Temp average* denotes the average discharge temperature of both packs, *Temp anomaly* denotes the deviation of both packs from the common average. For example, if one pack discharges air at 10°C while the other discharges air at 20°C, Temp average is 15°C, and Temp anomaly is ± 5°C.

Figure 4.9 shows the result of the simulations for the different discharge temperatures. The model was simulated for a cruise flight phase at 11.900m altitude. The horizontal axis shows the average temperature that can be achieved of the two packs. All possible combinations for a temperature range from -20°C to 30°C were considered. For each combination, the temperature anomaly to the average temperature was determined and presented by the vertical axe. The graphic shows a triangular shape what is related to the fact that for instance an average temperature of -10°C could be achieved by a maximum range of 0°C and -20°C what leads to an anomaly of 10K. The colouring represents the total

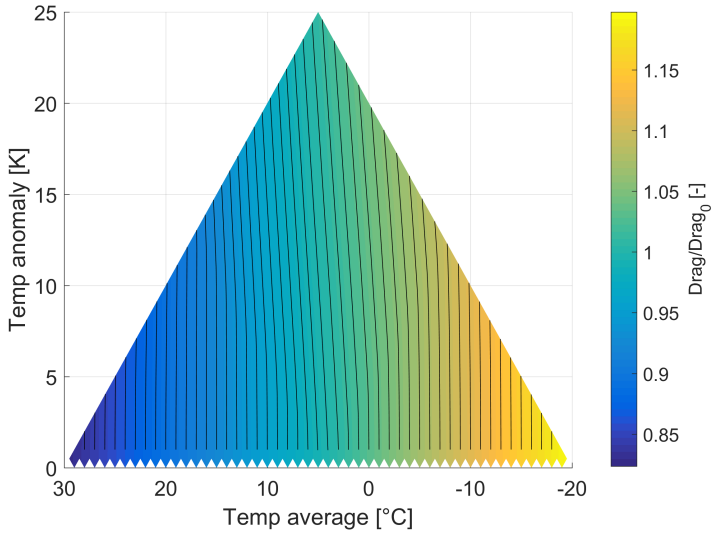


Figure 4.9: *Drag caused by Ram air vs. average temperature and temperature anomaly*

drag of both packs and the black lines represent lines of constant drag. Due to reasons of confidentiality, the values are normalised to an average drag value.

The results show that the drag for the border regions of high and low average temperature remains constant with increasing anomaly. However, for the medium temperature range, the lines of equal drag tip to the left with increasing anomaly. That means, the combined drag of both packs is slightly higher for packs operating with large temperature differences.

4.5 Control

4.5.1 Requirements

The following challenges for the control system were identified.

Competition

Air mass flow through both main pipes is constant. The valves can only redistribute the total mass flow between the temperature zones. If a zone receives an above average amount of cold air, the others receive less. If all valves open up, the net result is zero.

It is therefore important that the valves do not work against each other. If the average temperature in the main pipes is too high, the average cabin temperature will get too high as well, irrespective of the behaviour of the valves. Decentralised control strategies would result in a situation, where all valves move in parallel. Since the distribution of air is only dependent on the relative

valve positions, the net result is zero.

Thermal diffusion between the different temperature zones tends to bring the temperatures of neighbouring temperature zones together. This is helpful for the rejection of disturbances that act on one of the zones. It is counterproductive when the temperature set points of neighbouring zones diverge. In any case, centralised schemes should have an easier time exploiting this effect.

Pack energy

A related problem is the cabin heat balance. Since the mass flow rates in both main pipes are fixed and equal, the average temperature of the air in both pipes has to be controlled by regulating the pack outlet temperature demands. The resulting dynamics can interact with the control of the valves, as both packs and valves have a direct influence on the temperature of the cabin zones.

Also, to save fuel the temperature difference between both pipes should be minimised, as shown in section 4.4. However, a minimum temperature difference is necessary for the temperature regulation concept to work. This minimum temperature difference is dependent on the current state of heat-loads and temperatures inside the cabin. A suitable controller should therefore keep the temperature difference as small as possible and as large as necessary.

Variable gain

Movement of one of the valves has a direct first-order effect on the temperature in the respective zone. This effect is proportional to the temperature difference of the air in both main pipes. Therefore the valves have variable gain, introducing a nonlinearity in the system

4.5.2 Control strategies

In the following, two basic approaches are presented. Additionally, a nonlinear augmented variant is described for each basic approach.

PID

Controllers of aircraft energy systems are usually kept simple, often resorting to proportional-integral-derivative (PID)-controllers as first introduced 95 years ago by [68], or the even simpler proportional-integral (PI)-variants. Staying true to this tradition, the first control approach is completely based on PID-controllers, see Figure 4.10.

The approach features a cascading structure. For n cabin zones, $n + 2$ PID-controllers are used.

One PID-controller per cabin zone is dedicated to control the respective valve. This group of controllers is called *valve-controllers*. The working range of the valves is mapped to a range of -1 (only cold air) to 1 (only warm air).

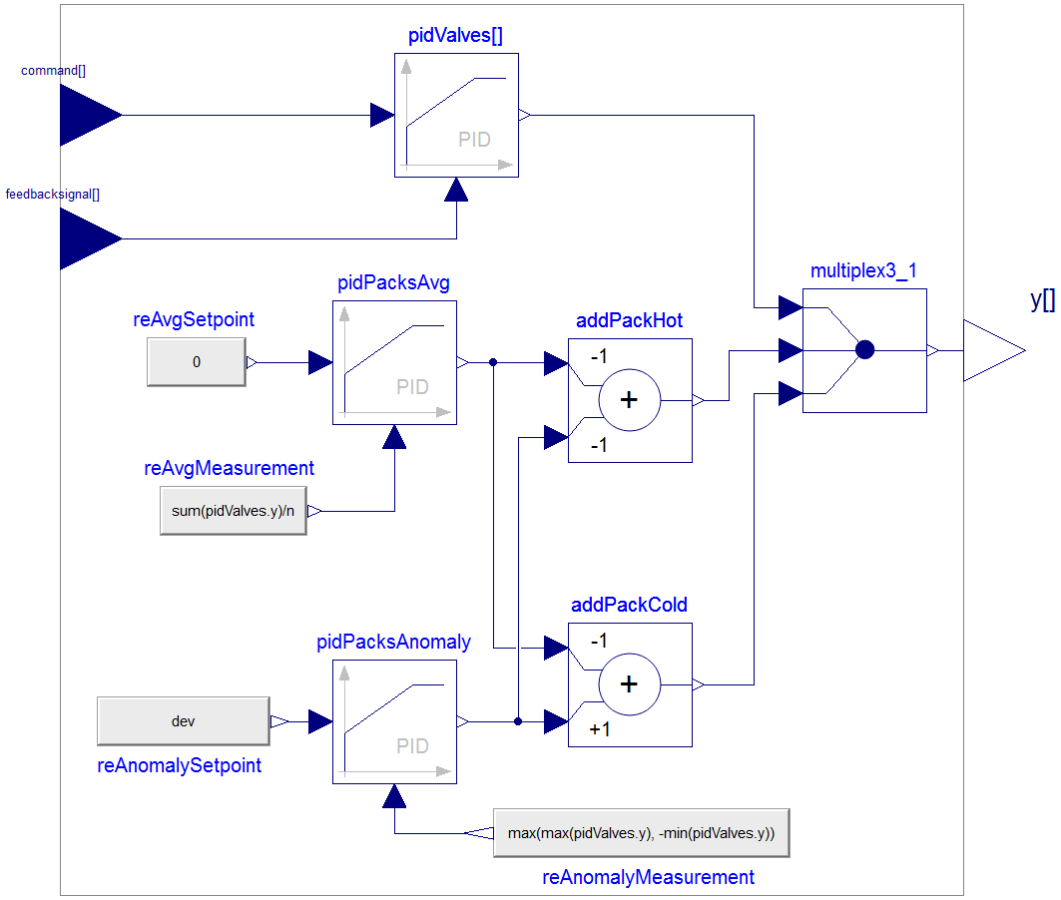


Figure 4.10: *PID control approach*

Accordingly, the controllers are limited to this range, employing an anti-windup track.

Two additional PID-controllers are used to control the pack demand temperatures. The first one of them controls the average pack temperature demand and is therefore called *pack-average-controller*. It tries to bring the average output of the valve-controllers to zero. If the average output of the valve-controllers is above zero, it means that the average temperature of the complete cabin is too low. The average pack temperature then has to be increased.

The last controller regulates the difference between both pack temperature demands and is therefore called *pack-anomaly-controller*. It is tasked with keeping the pack temperature anomaly as small as possible and as large as necessary. This is realised by the following strategy: for all valve-controllers, the outputs are continuously monitored, and the output with the largest absolute value is computed. If this output is larger than a fixed set-point (for instance 0.8), the temperature anomaly is increased. This gives more control authority/gain

to the valves, so the output of the Valve-controllers should drop.

The outputs of the pack-average-controller and the pack-anomaly-controller are then combined to find the temperature demands for both air conditioning packs.

PID with control-input-normalisation

The PID-approach as described in subsection 4.5.2 is a linear controller for a nonlinear system. However, it is easily possible to compensate for one important nonlinearity: the influence of the control valve positions on the cabin temperature is dependent on the temperature difference of the main pipes. The output of the valve-controllers can therefore be multiplied with the inverse of the temperature difference of the mixing chambers, thereby compensating for this effect. A correspondingly modified version of the limited PID-controller of the *Modelica Standard Library* is shown in Figure 4.11.

It is important to normalise the PID-controller-output at the correct location. The normalisation should take place after the integral element because it affects the actuation - not the measurement. Controller output still has to conform to actuator limits after normalisation, so anti-windup measures should be located behind the normalisation.

LQG

As already discussed, there are strong interaction effects between the valves. To take advantage of this, an obvious way is to use a centralised control scheme. In this work, a linear quadratic gaussian (LQG)-controller was used. LQG-controllers are based around a linear quadratic estimator (LQE) to estimate the complete state vector of the system based on noisy and incomplete measurements and a linear quadratic regulator (LQR) to control the system based on these estimated states. Often, LQEs are called *Kalman filters* instead.

Controller synthesis included artificial integrators to assure zero steady-state error, using the formulation as described in [53, p. 348]. To keep the number of tuners feasible, most LQG weighting matrices were defined as scalar matrices. The input cost matrix R was defined as a block diagonal matrix of two scalar matrices. In this way, weights for valve actuators and pack actuators can be tuned separately. The state cost matrix Q was defined as a combination of projected system outputs, system states, and artificial integrator states:

$$
\begin{aligned}
Q &= \begin{pmatrix} w_{states} I + w_{out} C^T I C & 0 \\ 0 & w_{int} I \end{pmatrix} \\
R &= \begin{pmatrix} w_{valves} I & 0 \\ 0 & w_{packs} I \end{pmatrix}
\end{aligned}
\tag{4.1}
$$

The resulting controller takes the form of a single state-space system.

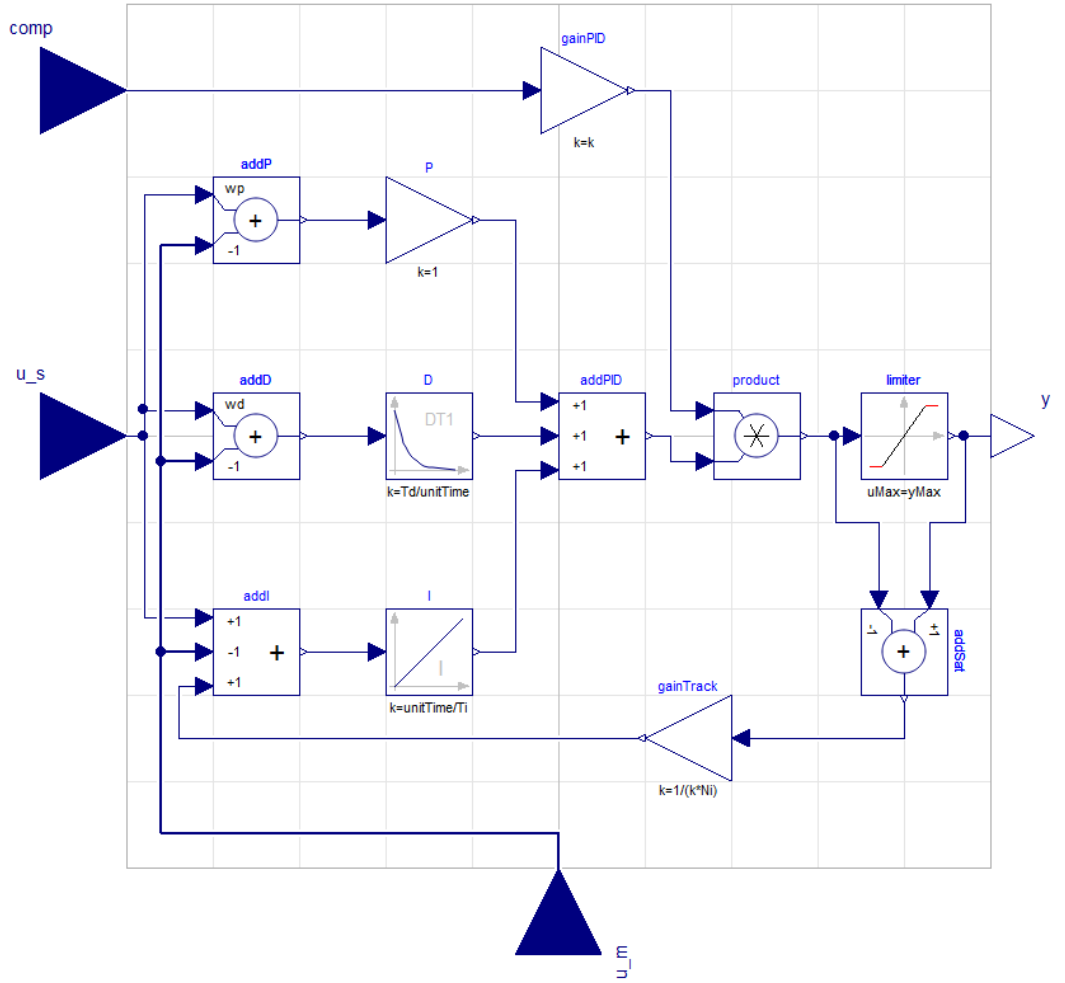


Figure 4.11: *PID with control-input-normalisation control approach*

Augmented LQG

For this approach, several aspects of the other approaches are combined. The first part of the controller is an LQG controller as defined in subsection 4.5.2. The outputs of the LQG are divided into two groups: valve positions and pack demand temperatures. The following steps are similar to the normalized PID-approach. The valve positions are multiplied with a normalisation factor which is dependent on the mixing chamber temperatures. The pack demand temperatures from the LQG controller are averaged, replacing the pack-average-controller. A PID-controller is added to control the pack anomaly. This is illustrated in Figure 4.12.

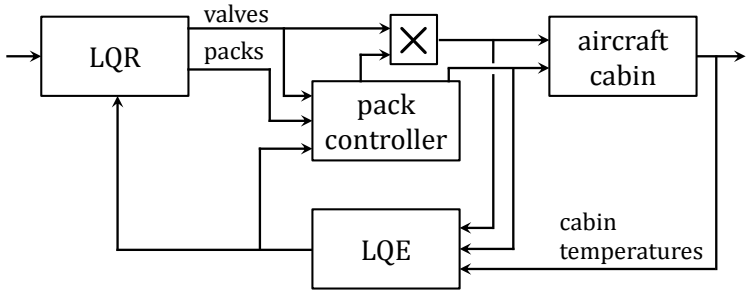


Figure 4.12: *Augmented LQG control approach*

4.5.3 Results

A simulation model of the aircraft-cabin temperature system was developed in the equation-based, object-oriented modelling language (EOOML) Modelica, for details see [17].

As an optimisation criterion, a weighted sum of the integrated quadratic control errors, integrated absolute valve movement and pack energy usage was used. This criterion was computed for the case of a standard flight profile including some disturbances in heat-loads.

The PID-based control approaches were implemented in Modelica. The respective tuners were optimised using the *Modelica Optimisation library* [69].

For the synthesis of the LQG-based control approaches, the model was linearised using the Modelica *linear systems library* as presented in [70]. Also, the nonlinear model was exported, so the optimisation criterion could be computed in *Simulink*. Optimisation was done using Multi-Objective Parameter Synthesis (MOPS) [71].

Both optimisation tools were configured to use the *Nelder-Mead Algorithm* [72]. This algorithm is favourable for several reasons: It does not require the computation of derivatives, which would be costly for the high-dimensional parameter space of this work. It is also suited and popular for nonlinear optimization problems [73], without the determinism problems of other algorithms, such as simulated annealing or evolutionary strategies. Also, the algorithm was available on both tools, increasing the comparability of the end results. However, small implementation differences of the algorithms can not be ruled out.

Performance

Selected simulation results for the four optimised control approaches can be seen in Figure 4.13. The corresponding performance metrics can be seen in Table 4.1. The performance numbers are based on the cost function as defined in Subsection 4.5.3. They were normalized against the respective best control

strategy for each case.

The PID with control-input-normalisation controller approach offers the best nominal performance followed by the basic PID controller, the Augmented LQG controller and the basic LQG controller.

Robustness

Preliminary simulations show that the main reason for potential instabilities of the controlled plant is the limited slope of the air conditioning packs. Slope limiters are hard nonlinearities. Conventional stability margin analysis is therefore not applicable for the nonlinear system in a meaningful way. Instead, three cases were selected to demonstrate controller performance under parameter changes.

Low time constants: in this case, all physical time constants such as air volumes, heat capacities and the time constants of the air conditioning pack dynamics are divided by two. This can reflect modelling errors, where the system behaves faster than assumed, as well as situations, where fewer seats or other pieces of furniture are installed by the airline.

High time constants: similar to the previous case but all time constants are multiplied by two instead.

Faulty valves: the mass flow of the valves in zones 5 and 7 is set to zero. This reflects a failure situation where the corresponding valves are clogged or failing.

Summary

The values of the optimisation criterion for all control approaches and scenarios are shown in Table 4.1. Selected simulation outputs for the nominal case are shown in Figure 4.13. While the valve positions are limited between 0 and 1, target valve positions can leave the allowed band since no anti-windup strategy was implemented for the LQG-based controllers. Selected simulation outputs for the case faulty valve case are shown in Figure 4.14.

All of the investigated approaches easily find a stabilising controller. It is apparent that augmentation of the controllers leads to a significant positive effect on controller performance. For each combination of basic control approach and simulation case, the augmented variants offer better performance.

In the nominal case, the PID-based approaches offer superior performance compared to the LQG-based approaches. This is unexpected since they have less degrees of freedom. The discrepancy may be caused by the anti-windup-compensation included in the PID-controllers, as well as incomplete tuning of the LQG-based controllers.

	Competition	Energy	Variable Gain	Anti-Windup	Simplicity	Number of tuners	Controller time states	Performance (less is better)	robust performance (less is better)
PID	-	+	-	+	+	10	24	1.079	1.039 1.035 1.261
N-PID	-	+	+	+	o	11	24	1.000	1.000 1.000 1.255
LQG	+	-	-	-	o	7	38	1.663	1.549 1.229 1.190
A-LQG	+	+	+	-	-	12	40	1.143	1.036 1.011 1.000

Table 4.1: *Summary of control approaches, their features and simulated performance*

Control approach	PID	N-PID	LQG	A-LQG
Maximum valve velocity	18.3%/s	6.9%/s	0.8%/s	1.7%/s

Table 4.2: *Maximum occurring valve velocities for nominal case*

For the cases with lower or higher physical time constants the distance in performance between both base approaches is significantly smaller. The normalised versions of each approach only differ by 3.6 and 1.1 percent respectively. This relative improvement of the LQG-based approaches may be explained by the large robustness margins of linear quadratic regulators, with LQR having at least 60 degrees phase margin and at least 50 percent gain reduction tolerance [74]. While these guarantees do not hold in the more general LQG case (see the very short abstract of [75]), the influence of the estimator on the robustness seems to be negligible in this case.

In the case of valve failures, LQG-approaches show significantly better behaviour. This is also unexpected, since the LQG-concept is based on the assumption that all actuators are working. It is generally believed that decentralized controllers are more robust to (unmanaged) sensor and actuator faults. Perhaps this system represents a lucky case, where centralised control is able to exploit interaction effects between neighbouring temperature zones in a meaningful way, without being designed to.

In this work, the response characteristic of air conditioning packs was realistically modelled, while the control valves were assumed to be infinitely fast. Of course, this does not hold in reality. The different control approaches exhibit varying requirements on the control valve velocities. The maximum valve velocities occurring for the nominal case are shown in Table 4.2.

It is apparent, that the maximum demanded velocity is about an order of magnitude slower for the LQG-based approaches. This allows the use of slower and cheaper valves as well as lighter valve actuators.

To summarise, the augmented LQG approach offers a little worse performance

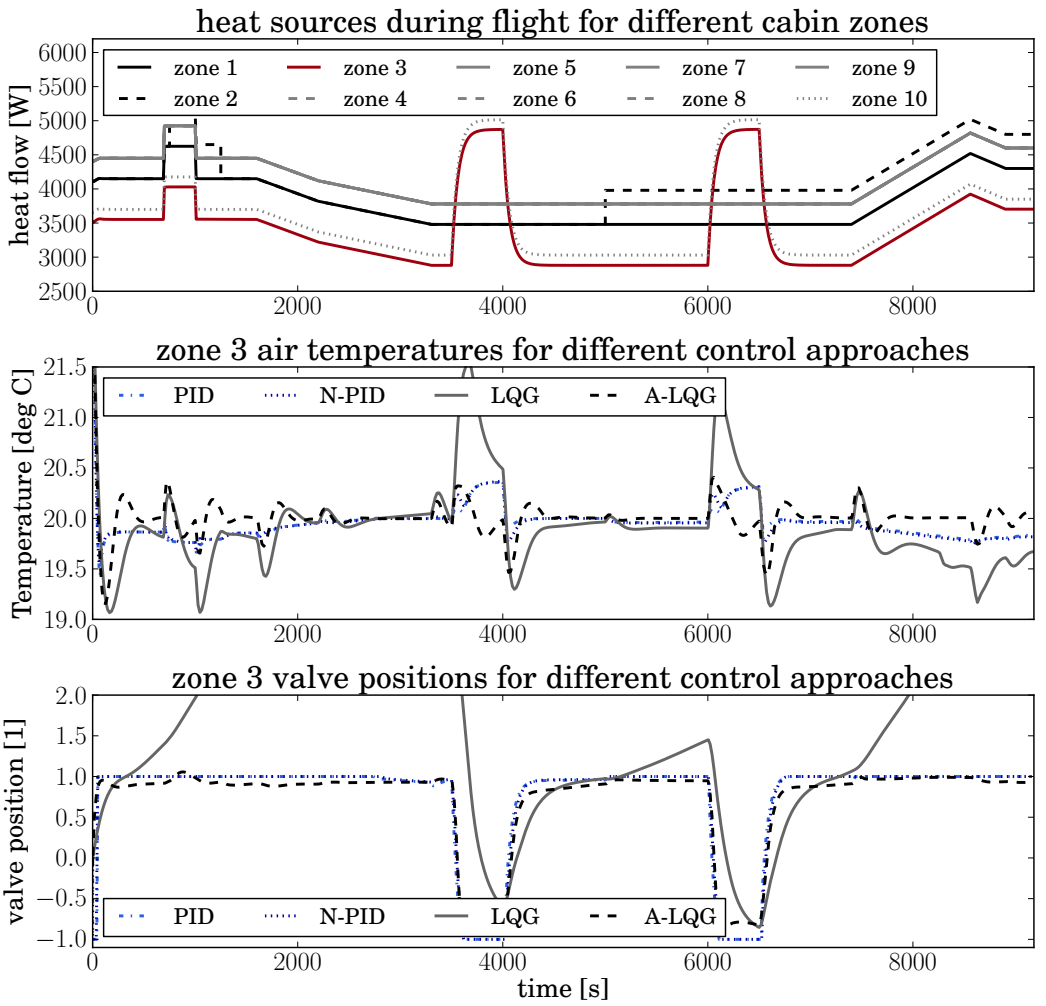


Figure 4.13: *Simulation results for optimised control approaches - standard case*

compared with the normalised PID approach in the nominal case, but equal or better performance during off-design cases. Also, its employment allows for the use of cheaper and lighter valves.

4.6 Failure management

The system performance as indicated in the last section holds for nominal operation. During the life-cycle of an aircraft fleet that is equipped with the proposed architecture, components will inevitably fail. It is therefore important to consider those cases and devise strategies to cope with them safely and with minimal loss of passenger comfort. In the scope of this work, three relevant failure scenarios are foreseen: failure of one or more air conditioning packs,

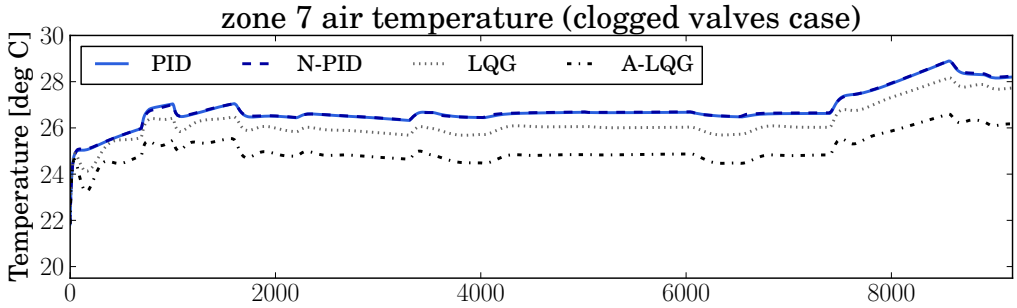


Figure 4.14: *Simulation results for robustness test with clogged valves in zones 5 and 7*

failure of one or more valves, and failure of recirculation. Strategies for dealing with these scenarios are described in the following.

4.6.1 Pack failure

In the event of a pack failure, the available total mass flow of fresh air is reduced. The remaining packs usually try to compensate by increasing their mass flow rate, but only up to a certain point. Since recirculated air is still available, the reduction of mass flow of air blown into the cabin is less severe than the reduction of mass flow of fresh air. If the cold or warm mixing chambers are each supplied by two or more packs, the situation is probably not critical. The remaining packs can then try to keep the mixing chamber temperature at the target temperature. Let us instead focus on the worst case, where only two packs are available in total. This case is also typical for single-aisle aircraft. If one of these two packs fails, the following strategy is proposed: using an emergency bypass, both mixing chambers are supplied by the remaining pack. The target discharge temperature of the pack is set to the original cold pack, irrespective of the situation before the failure took place. An electric heater inside the warm mixing chamber is used to generate the necessary temperature difference between the mixing chambers. This setup requires three additions to the system: first, an emergency bypass system for both packs; second, an electric heating system in the warm mixing chamber; third, a modified control algorithm in the event of a pack failure. Optionally, the electric heating system can also be used during normal operation, if the required temperature of the warm mixing chamber is too low, and the corresponding pack does not react fast enough. The power requirements for the electrical heater will be substantial, but similar heaters are already employed for some concepts, where an extra temperature zone is created in that way. Also, in an emergency case, shedding the power from the galleys is conceivable.

4.6.2 Valve failure

In the event of a valve failure, a meaningful response is dependent on the exact architecture. Earlier in this paper, three base variants were proposed for convection-based and mixed ventilation each:

1. one mixing valve per temperature zone
2. one mixing valve plus one mass flow control valve connected in series per temperature zone
3. two mass flow control valves for warm air and cold air connected in parallel

For the first variant, if the valve is unresponsive and stuck in an undesirable position, no meaningful compensation is possible. In any case, the mixing valves should be constructed to have their middle position as a fail-safe, so that passengers are subjected to less extreme temperatures and not die horribly. If the valve is clogged, the centralised control system will still partially compensate for the temperature deviation using the valves of neighbouring zones. For the second variant, if the mixing valve is unresponsive and stuck in an undesirable position, the mass flow control valve should close. The zone is then supplied via the neighbouring zones. If the mass flow control valve is unresponsive, the mass flow control valves of the remaining zones should match the position, so the overall mass flow is distributed evenly throughout the cabin. If one of the valves is clogged, the control system will partially compensate. For the third variant, if one of the valves is unresponsive, the other should try to control the mixed temperature. Mass flow control is then impossible. If one of the valves is clogged, the other should close. In any case, detailed fault detection is necessary.

4.6.3 Recirculation failure

If one or more of the recirculation fans fail, the total air mass flow will be lower. This does not require any physical response in the system. However, the control system should adapt to the lower mass flow, which corresponds to a lower control authority. The response could be as simple as an additional normalisation step, similar to the one proposed earlier.

4.7 Discussion

The proposed temperature regulation system allows for a much larger number of cabin temperature control zones than what is usually available today. A larger number of zones corresponds to smaller zone areas, enabling a more fine-grained temperature regulation inside the cabin. This decreases the risk of passenger discomfort. Also, local temperature deviations can be compensated faster since the thermal mass between the temperature control valve and the air inlet is much lower. Several recent developments and technological trends

emphasise the benefits of this system. Passenger density has been increasing steadily since the beginning of commercial aviation, to the point of the modern low-cost carriers that offer minimal seating distances of around 0.7m between rows. The result is a higher potential for thermal imbalances, if not all seats are occupied. On the other side of the spectrum, high end carriers experiment with individual seating configurations like mini-suites to distinguish themselves from the competition. These configurations also carry a high potential for thermal imbalances, illustrating the motivation for this chapter.

The proposed concept is bought at the expense of a more complicated and centralised control system, as well a potentially large number of necessary control valves. Since the total air mass flow is constant, a large number of valves corresponds to smaller valve units, which should help keeping the weight and installation space moderate.

4.8 Outlook and future work

In this chapter, an innovative architecture for aircraft cabin temperature regulation was presented. The topics of control and failure case management were discussed. Before implementation on an industrial scale, i.e. for new aircraft projects, technology readiness has to be increased.

Contrary to conventional architectures, the proposed concept requires multiple air conditioning packs to operate at different discharge temperatures. The operational regimes of air conditioning packs differ from each other. It may be efficient to design the packs accordingly. Potentially, the Pareto-optimal point with regard to system weight and system drag will be shifted to asymmetrical pack design. Additional analyses are necessary for this open question.

The simulation model in this chapter is based on first principles and is inherently control-oriented, having a low order and allowing for fast simulations. In particular, the cabin air dynamics are represented by a number of volume elements, similar to the number of cabin temperature zones. Perfect mixing, as well as linear temperature equalisation is assumed. Reality is not so simple. To guarantee system and control performance under more realistic conditions without having to build a functioning physical prototype, a CFD-based approach can be used, similar to software in the loop (SiL): after a cabin is implemented using computer aided design (CAD), the model is exported to a CFD model. The CFD-model is then coupled to the developed control system. Several typical scenarios are executed, and system performance is evaluated. As an intermediate step, quasi-threedimensional simulation as described in [76] is feasible.

Convection-based ventilation is considered by airframers as an alternative to mixed ventilation. With this ventilation concept, temperature differences between the right and left side of the cabin are larger since less horizontal heat

transport takes place. The proposed architecture could be expanded to supply both sides of the cabin with air of different temperatures.

Until now, only thin tube aircraft were considered. While representing the vast majority of today's civil aircraft, other shapes are used or in development. Double-deck aircraft such as the *Airbus A380* require a slightly different temperature regulation system. However, the proposed architecture is easily adaptable to these aircraft, while offering an even greater benefit: where conventional architectures would require twice the amount of ducting, the proposed concept gets along with shared main pipes at the height of the intermediate ceiling, supplying both decks. This would of course require additional analyses regarding the optimal controller structure.

4.9 Acknowledgements

Daniel Bender conducted and evaluated the investigation about pack energy uage for the presented temperature regulation concept, as shown in section 4.4. Ines Kerling implemented the *MATLAB*-script to export a controller in the shape of a statespace object to Modelica code.

CHAPTER 5

One-size-fits-all-control for architecture optimisation

Professor Quirrell sighed. "It is enchanted not to eat students, just spit them back out through the door. Now, boy, how would you recommend that we deal with this dangerous creature?"
"Uh," Harry stuttered, trying to think over the continued roaring of the chamber's guardian. "Uh. If it's like the Cerberus from the Muggle legend of Orpheus and Eurydice, then we have to sing it to sleep so we can pass -"
"Avada Kedavra."
The three-headed beast fell over.

Eliezer Yudkowsky, Harry Potter and the Methods of Rationality

5.1 Problem definition

In industrial projects, typical phases are modelling, architecture optimisation, and control systems design. For larger projects, the overlap between the responsible groups of people can be small to nonexistent. Furthermore not every modelling expert is also a control expert.

For preliminary studies on an architecture level, equation-based, object-oriented modelling language (EOOML) like *Modelica* are well suited since individual components can be connected and rearranged quickly and flexibly.

However, additional efforts can arise when the system architecture also includes controlled components, which are also to be sized. An example for this might be pumps that control the massflow through a pipeline. The actual pump characteristics are parameterised and determined during the architecture optimisation. During the optimisation, each function evaluation corresponds to a simulation of the architecture. For these simulations, the pump needs a controller model that controls the pump in such a way that the target massflow is reached. This holds, even if the function evaluation is only dependent on the steady state behaviour of the system.

A typical workflow for the development of such controllers looks like this: a predefined proportional-integral-derivative (PID)-controller-model from a standard library is included. The controller output is connected to the valve input. Two additional elements are created and connected to the controller model to retrieve or define controller target and actual value. The PID controller is set to proportional-mode with a k_p of 1, and the system is simulated. The dynamic behaviour of the controlled valve is checked, and the controller gain is adjusted for correct order of magnitude and correct sign, if necessary. This can take a few iterations. Subsequently, the controller is set to proportional-integral (PI)-mode, and a small k_i value is defined to assure zero steady state error. Again, this can take a few iterations until the correct order of magnitude is found. If the dynamical behaviour stays insufficient, or if the modelling expert is sufficiently motivated, a few experiments with added derivative action (PID-mode) might follow. Alternatively, optimisation tools might be used, shifting the bulk of the effort into the creation of the optimisation setup.

This workflow takes some time until the results are acceptable. Also, the resulting controller might not be robust against model changes, making additional effort during the optimisation phase necessary. If an architecture contains several controlled components, the necessary effort grows accordingly. This effort is ultimately wasted since the controllers will be redeveloped by actual control experts anyway, as soon as the architecture is finalised.

In this chapter, controllers are developed that are able to effectively control a wide range of simulated components without any tuning effort. To achieve this, the most important control approaches are reviewed and categorised according to their suitability. The best candidate is identified, then modified to suit the needs of modelling experts. Several examples are presented to demonstrate the performance of the resulting control approach.

5.2 Suitability of major control approaches for architecture simulation

5.2.1 PID

Principle: here, the controller output is computed as a sum of three components: one component is proportional to the control error, the second component is proportional to the derivative of the control error, and the third component is proportional to the time integral of the control error.

Advantages: PID-controllers are the most widely used controllers in the scope of object-oriented modelling. They are easy to understand, offer reasonable performance for most single input single output (SISO)-systems, as well as zero steady state error and simple addition of anti-windup measures.

Disadvantages: PID-controllers require the user to tune three parameters, either by hand or using optimisation. As soon as a good tuning is found, it might not be suitable for different parameterisations on architecture level or in case of strongly off-design operating points.

Variants: the concept of PID-controllers has been generalised into Fractional PIDs, as shown in [77]. Here, fractional differential operators are used instead of the usual integer operations (integral and derivative action). Fractional differential operators require infinite memory during simulation, but can be approximated in EOOMLs (see [8]). However, the number of tuners increases from three to five, therefore the usability for one-size-fits-all-control is even more limited.

5.2.2 Model inversion

Principle: EOOMLs are not causal. That means that there is no inherent direction of computation (as opposed to, for instance, a block diagram). Nothing prevents the modeller from defining the nominal output of a system, leaving the computation of the system input to the solver. This can be used for control: the controlled variable is equated to the target value, and the virtual controller output is computed during simulation.

Advantages: for this approach, no additional effort is necessary on the modeller's side. It is inherent in modern EOOML. Also, the controlled variable is perfectly tracked.

Disadvantages: bounds on the controller output cannot be implemented. Also, model inversion usually fails if the model has a high relative degree, or if there is no unique solution. This affects all but the simplest models. It also fails if the inverse model is unstable, which happens if the system to be inverted has unstable zero-dynamics.

Variants: many modern control approaches are based on model inversion. Examples are nonlinear dynamic inversion as used in [78] or incremental non-linear dynamic inversion as used in [79]. However, these approaches lose much of the simplicity of direct inversion, and are therefore not suited for a modelling expert.

5.2.3 LQR/LQG

Principle: linear quadratic regulators (LQRs) represent a static state-feedback-law that optimises a quadratic cost function (H_2-norm) for linear time-invariant (LTI)-systems. Since they use state-feedback, they require a complete state vector to compute the controller output. Typically, LQRs are combined with a linear quadratic estimator (LQE), also known as Kalman Filter, to get an estimate of the state vector based on measured variables. The combination is known as linear quadratic gaussian (LQG) [80].

Advantages: for simulation studies, the exact state vector is known to the solver, therefore LQR based control is possible without any need for an estimator. LQR offers guaranteed robustness properties, in contrast to LQG [75]. Control of multiple input multiple output (MIMO)-systems is possible.

Disadvantages: the computation of the LQR matrix requires a solution of the *Riccati-equation*, making the use of external tools like *MATLAB* necessary. LQR is based on the assumption that the system is LTI, which is often not the case for complex real world applications. There is no treatment of actuator limitations, as well as no guaranteed zero steady state error. Retrieving the actual state vector in the correct shape might not be that straightforward for the modeller since environments for EOOML usually encapsulate this information from the user. For this reason alone, the use of LQG in equation-based, object-oriented modelling (EOOM) environments might make sense.

Variants: while the computation of LQR and LQG controllers is straightforward, there are possibilities to extend the method. In [53] a variant of an LQG is described, where the system model is extended by artificial integral elements. Since the output of these integrators is controlled as well, zero steady state error of the native states can be guaranteed.

5.2.4 $\mathbf{H_\infty}$

Principle: based on the nonexistent robustness-guarantees of LQG regulators [75], H_∞-regulators were developed [81], [82]. The principle is similar, but a H_∞-performance-norm is optimised, instead of a H_2-norm. This roughly

corresponds to an optimisation of the worst case, instead of an average case.

Advantages: this approach offers good robustness properties against system uncertainties. Many different kinds of performance goals can be implemented during the controller synthesis. Control of MIMO-systems is possible.

Disadvantages: the synthesis of H_∞-controllers requires in-depth knowledge about the method. External tools like MATLAB are necessary to solve the linear matrix inequalities that describe the optimal controller. Also, the resulting controller can be quite large, with a number of states equal to the number of states of the system plus the number of states of the filters used for the performance metrics.

5.2.5 Sliding mode control

Principle: in sliding mode control (SMC), a desired subspace (sliding surface) of the system state space is defined in a way that exhibits desirable dynamics. Nonlinear control laws are used to drive the system state onto the sliding surface in finite time. Typically, this takes the form of a bang-bang control-law, where the controller tries to drive the system state into the direction of the sliding surface with maximum authority.

Advantages: SMC is robust against matched uncertainties. As soon as the sliding surface is reached, plant deviations do not lead to deviations in plant dynamics. No tuning based on experiments or complex calculations is necessary, only the desired dynamics have to be defined.

Disadvantages: as soon as the sliding surface is reached, chattering occurs. The controller output then changes significantly with each time step. Simulations using implicit solvers and clean event-detection get stuck.

Variants: several approaches have been described to alleviate the chattering effect: in Filtered SMC as described in [83], the controller output is first-order filtered. In boundary layer sliding mode control (BLSMC) as described in [84], the hard non-linearity at the sliding surface is replaced by a smooth transition inside a small boundary layer. A number of second-order SMC are described, see for example [85]. Here, both the sliding variable (roughly translatable as the distance of the state to the sliding surface) as well as its derivative are driven towards zero in finite time.

5.2.6 Data driven control system design

Principle: for most control design methods, models of the systems are created, and controls are developed on the basis of these models. Data-driven control system design (DDCSD) methods skip the modelling step, and controllers

are synthesised directly based on data obtained from experiments. One example is virtual reference feedback tracking (VRFT) as described by [86] and [87]. Here, the plant output for a given plant input is filtered by the inverted desired dynamics, giving the virtual controller input. Now a controller is fitted that responds to the virtual controller output with the given plant input.

Advantages: DDCSD methods can deliver good performance, even for nonlinear systems. Iterating variants of this approach allow to implicitly choose the perfect linearisation point for non-linear systems.

Disadvantages: the biggest advantage of DDCSD, the renouncement of any modelling effort, is unhelpful in the context of simulation studies, where models are used anyway. Additionally, external tools are necessary for controller synthesis. Often, a stabilising controller is needed as a basis for data generation, increasing the total controller design effort.

5.2.7 Model predictive control

Principle: model predictive control (MPC) as described in [88] uses real-time optimisation to find optimal controller outputs based on a model. On each discrete time step, current and future controller outputs are found so that future system trajectories optimise a performance goal.

Advantages: MPC is able to anticipate changes in system behaviour, which cannot be anticipated by reacting control approaches. The approach copes well with model nonlinearities and actuator saturations. Very good system performance is achievable.

Disadvantages: if MPC is used in architecture optimisation, total elapsed walltime increases dramatically. Finding optimal controller outputs at each time step implies that the actual model is simulated multiple times for each evaluation. Also, an additional optimisation tool is necessary, either integrated into the modelling and simulation environment, or external. At the moment, no tool is known to the author that automates this process, adding up additional effort on the modeller's side.

5.2.8 Others

The approaches mentioned here do not constitute a complete list. Naturally, it is difficult to review each and every control strategy that has been published. Examples of approaches that have been left out are adaptive control (as described in [89]) or intelligent control techniques like fuzzy control or neural networks [90]. However, to the best knowledge of the author, most of these approaches are either concerned with only a subset of controllable systems, are overly complicated for the task at hand, or have other relevant disadvantages.

5.3 Universal controllers for equation-based simulation environments

5.3.1 Concept

With direct inversion and SMC being the only exceptions, all of the other considered approaches require either manual tuning or external design software. Direct inversion cannot be used for all but the simplest of systems. Second-order SMC is overly complex, first-order SMC produces chattering effects, making them unusable for the implicit solvers typically used in EOOM environments.

However, modified variants of first-order SMC are available to alleviate this problem. Two of them, filtered SMC and BLSMC were implemented in Modelica and used to regulate the states of a deflected double-integrator back to zero, with a first-order decay as sliding surface. For comparison, a conventional first-order SMC was also implemented[1]. The controller output was limited to ± 5. The system behaviour for all three controlled systems was nearly identical, but the corresponding controller outputs showed stark differences. While the standard controller exhibited the expected amount of chattering, it was significantly reduced for the filtered SMC. The BLSMC showed no chattering. These results can be seen in Figure 5.1. Using implicit solvers, BLSMC simulated roughly 6000 times faster than filtered SMC, probably as a result of generating no state events (compared to 92.942 state events for filtered SMC). That makes BLSMC a promising candidate for the control of simulated architectures. A commonly cited drawback for BLSMC is their sensitivity to noise [92]. However, this is not a problem in the context of simulation studies using implicit solvers, where noise is minimal to nonexistent.

5.3.2 Boundary layer sliding mode control

In SMC, the principle is to force the system state to stay on a defined subspace with beneficial properties. A sliding variable s is defined based on the system output and its derivatives. To enforce first-order decay of a SISO-system, the sliding variable can for instance be defined as

$$s = w_1 \cdot (y - t_{target}) + w_2 \cdot \frac{d}{dt}(y - t_{target}) \tag{5.1}$$

with w_1 and w_2 as weighting parameters, determining the time constant of the first-order behaviour. In the simplest case, the control law takes the form of

$$u = u_{max} \cdot sgn(s) \tag{5.2}$$

[1]Standard first-order SMC cannot be simulated using Dassl [91], the standard solver for many Modelica environments. For this reason, an explicit solver (RK4) was used for this comparison.

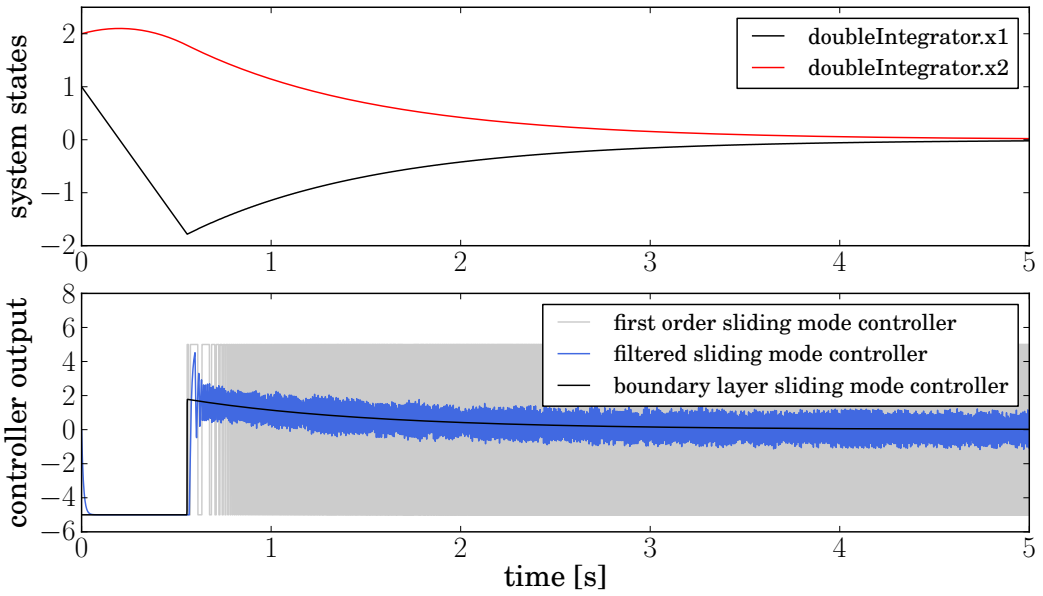


Figure 5.1: *Comparison of different first-order SMC variants*

The idea of BLSMC is to replace the hard discontinuity at $s = 0$ with a linear approximation:

$$u = \begin{cases} -u_{max} & \text{if } s < -l \\ u_{max} \frac{s}{l} & \text{if } -l < s < l \\ u_{max} & \text{if } l < s \end{cases} \tag{5.3}$$

with the layer width l. During simulation, this generates state events every time the layer thresholds are crossed, slowing down the simulation. For this reason, a formulation based on the hyperbolic tangent was used instead:

$$u = u_{max} \cdot tanh\left(\frac{s}{l}\right) \tag{5.4}$$

This has the advantage of being solver-friendly since no state events are generated during simulation. Also, the tangens hyperbolicus is C^{∞}-continuous, which can be exploited by Dassl and other differential algebraic equation (DAE)-solvers that use polynomial expansion.

5.3.3 Implementation

For implementation in Modelica, usability aspects have to be considered as well as technical aspects. Therefore, the controller model is designed to ensure applicability for a wide range of problems, while having a simple interface to the end user.

- For input of controlled variable and set point, the user can choose between conditionally defined connectors and input fields, where arbitrary expressions can be included.

- Minimum and maximum controller output can be set as fixed parameters, or can be defined adjustable using conditionally defined connectors.

- Three types of behaviour can be selected, corresponding to different structures of the sliding surface: zero-order dynamics corresponds to perfect matching between set point and actual value whenever possible. For first-order dynamics, a target time constant for control error decay can be entered. For second-order dynamics, both time constant and a damping value are used. In every case, the unused values are greyed out as to not confuse users.

- The relative width of the boundary layer is dependent on the order of magnitude of the controlled variable. If a pressure variable is controlled, typical values are in the order of 10^5. This makes the boundary layer much smaller in comparison. Therefore the user can also enter a unitsize variable to compensate. This variable can also be used to change the sign, if the controller output goes in the wrong direction, thereby making it the only tuning decision which is necessary, albeit a binary one.

- For users that know what they are doing, more options are available. These are grouped in an *Advanced*-tab. Here, the input can be flagged as smooth, so that exact derivatives are used for the calculation of the sliding variable instead of approximate derivatives. Also, the definition of the sliding surface can be overwritten. Finally, the boundary layer formulation can be replaced with a standard SMC, which might make sense if an explicit solver is used.

The user interface of the controller model is shown in Figure 5.2. The actual Modelica code listing can be seen in the Appendix in section A.1.

5.4 Examples

5.4.1 Low-order-systems

As a first test, the controller is used to control three different low-order-systems. The first system is a static unity-gain, the next system is a first-order system with a time constant of three seconds, the last system is a second-order system with an angular frequency of 0.5rad/s and zero damping.

All controllers use the standard values: first-order target behaviour with a time constant of one second. Output limits are set to ± 5. All systems were

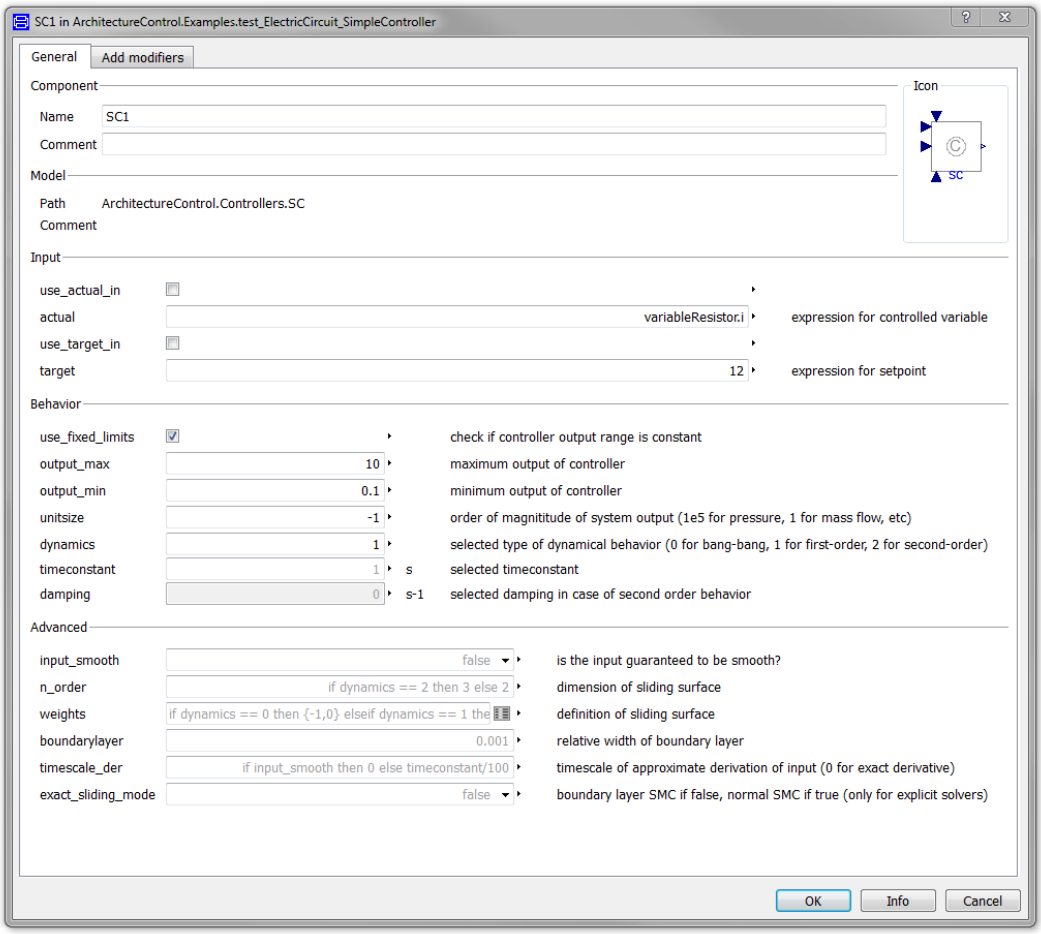


Figure 5.2: *User Interface of SimpleController parameters*

initialised at state vectors of zero, the target output was defined as a step at 1s. The results are shown in Figure 5.3.

The first-order system is forced on the reference behaviour almost exactly. A small overshoot in the system input/controller output is visible, caused by the time constant of the approximate derivative used for the controller formulation.

The behaviour of the second-order system is significantly limited by the controller limits. At 1.62s, this limitation is no longer relevant, and the system reverts to first-order behaviour. Again, a small overshoot occurs at the controller output, caused by the approximate derivative.

Interestingly, the static gain does not show any dynamic behaviour, instead the system output perfectly matches the target. This can be explained by the feedthrough behaviour of the controller, which instantly compensates the set point change. Since no control error is being generated, no control error has to

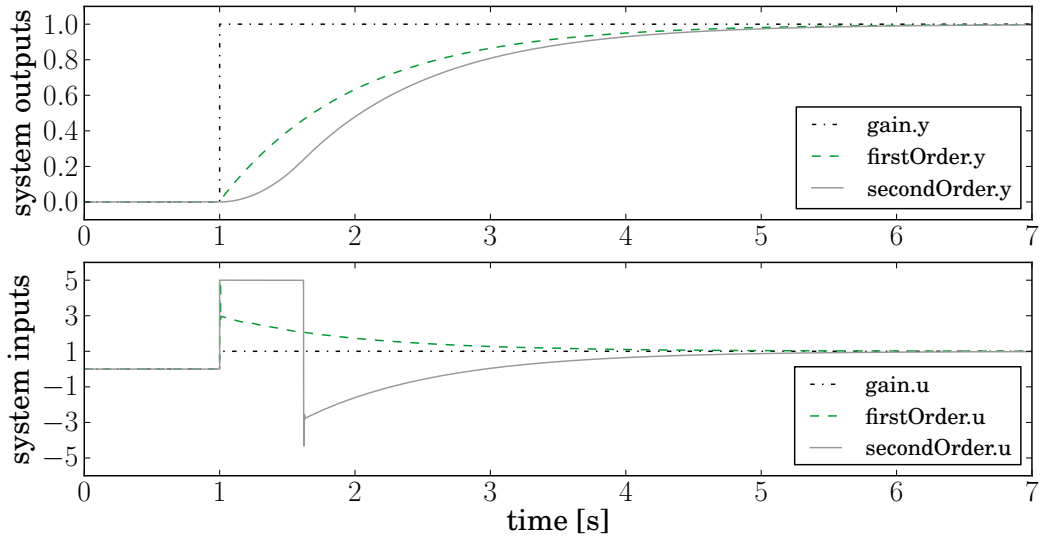


Figure 5.3: *Dynamic behaviour of several low order systems controlled by BLSMC*

fade away with a first-order behaviour. In contrast to this, a control error was being generated for the first- and second-order systems, when the controllers could not track the set point perfectly due to controller output limitations.

5.4.2 Thermal network

To estimate controller performance on systems with a high relative degree, a model of a thermal network was created. The network consists of nine thermal capacities, connected in a two-dimensional grid. At one corner, a fixed temperature boundary condition of -10°C is applied, at the opposite corner, a temperature is measured. This measured temperature is also the control variable, with a set point of 0°C. At one of the remaining corners, a disturbance heat flow is applied, taking the shape of a sawtooth function. The last corner is reserved for the actuator: a variable heat flow, with a range of 0 to 10W. The lower limit of 0W corresponds to a pure heating element, without any cooling capability. It also introduces a strong nonlinearity to the system. All thermal masses were initialised at 20°C. The system can be seen in Figure 5.4.

The system was instantiated two times, using different controllers. One time, the proposed BLSMC was used, with a selected second-order behaviour, a time constant of 1s and a damping of 1.

For comparison, the system was instantiated with a PID-controller, including anti-windup functionality. The three parameters of the PID were optimised using the *Modelica Optimisation library* [69]. As an optimisation goal, the integrated quadratic control error (IQCE) for a simulated time of 100s was used. The optimisation achieved an IQCE of 3016.3.

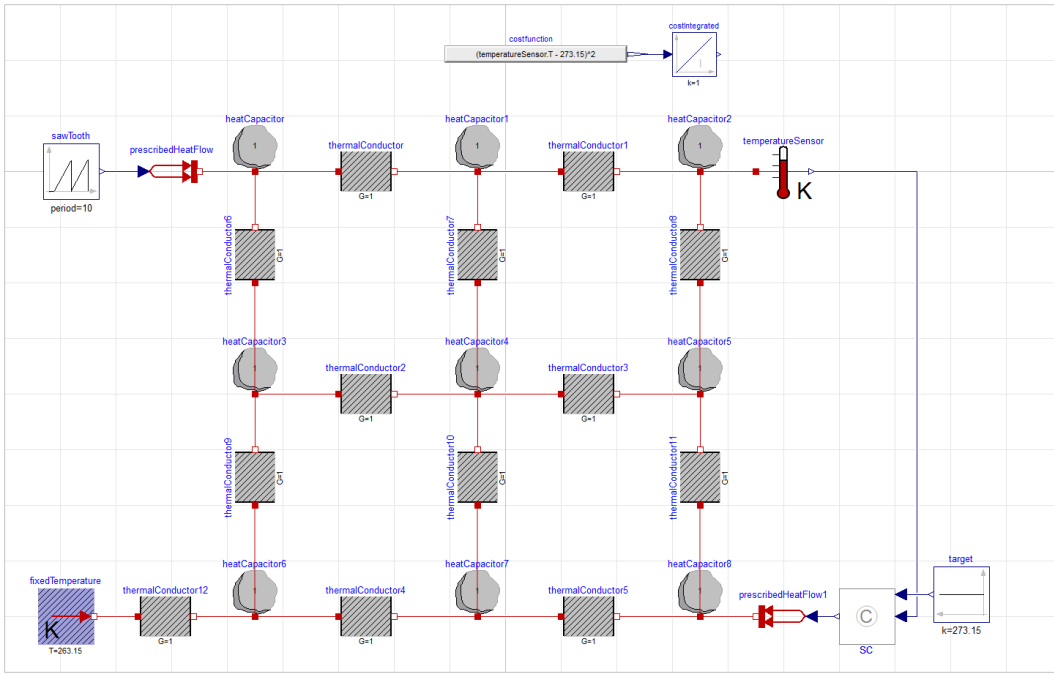


Figure 5.4: *Thermal network test model*

The SMC achieved an IQCE of 3014.8 without any optimisation. The corresponding temperatures and controller outputs can be seen in Figure 5.5.

5.4.3 Cabin temperature control

In chapter 4, a temperature regulation concept for aircraft cabins was presented and a suitable simulation model was created. This model contains some difficulties regarding the control system design: strong coupling between different control inputs is present. The air conditioning pack models includes slope limiters, a hard nonlinearity. A large number of continuous time states between the pack temperature demand input and the cabin temperature sensors results in a system of a large relative degree: a natural enemy of SMC.

This makes the model a perfect data point on how the proposed controller handles difficult systems. The cabin temperature regulation model was instantiated, and ten SMC were used to regulate the cabin temperature zones. Two additional controllers were used to control pack temperature average and deviation. The corresponding simulation results can be seen in Figure 5.6.

It can be seen that the control error is small for the majority of the time; the primary control goal is fulfilled. However, the system is not asymptotically and internally stable. The valve positions and the pack temperature demands show heavy oscillations of a high frequency. These oscillations are caused by the slope limiters of the pack models; removing the slope limiters also lead to

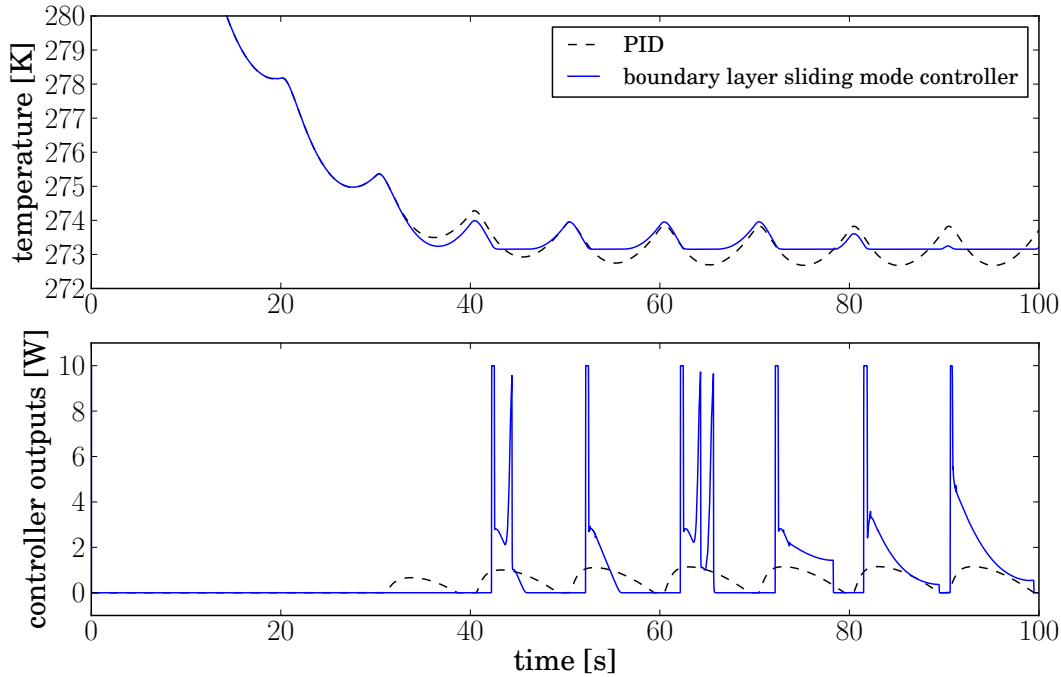


Figure 5.5: *Thermal network test results*

the disappearance of the oscillations.

In a second experiment, only the 10 valve controllers were implemented as SMC. Both pack controllers were adopted from the *N-PID* approach as described in chapter 4. The matching simulation results can be seen in Figure 5.7.

This time, no oscillations occur, while the control error is still small. Performance is similar to the performance of the controllers as developed in chapter 4, without any additional tuning effort.

5.5 Discussion

In the previous sections it is shown that BLSMC is a very easy to use tool, suitable for a wide range of simulation tasks. Performance wise, this class of controllers can compete with other control approaches. At the same time, configuration efforts for the modelling expert are minimal.

There are however situations, where the proposed controller is not adequate. For instance, this applies to MIMO-systems with strong coupling. If the water level in n water tanks has to be controlled using two valves, and there is no clear one-to-one assignment between the tanks and valves, the proposed controller will probably fail. On the other hand, as long as the influence of each valve on a single tank is strong, and the influence of the same valve on all other tanks is

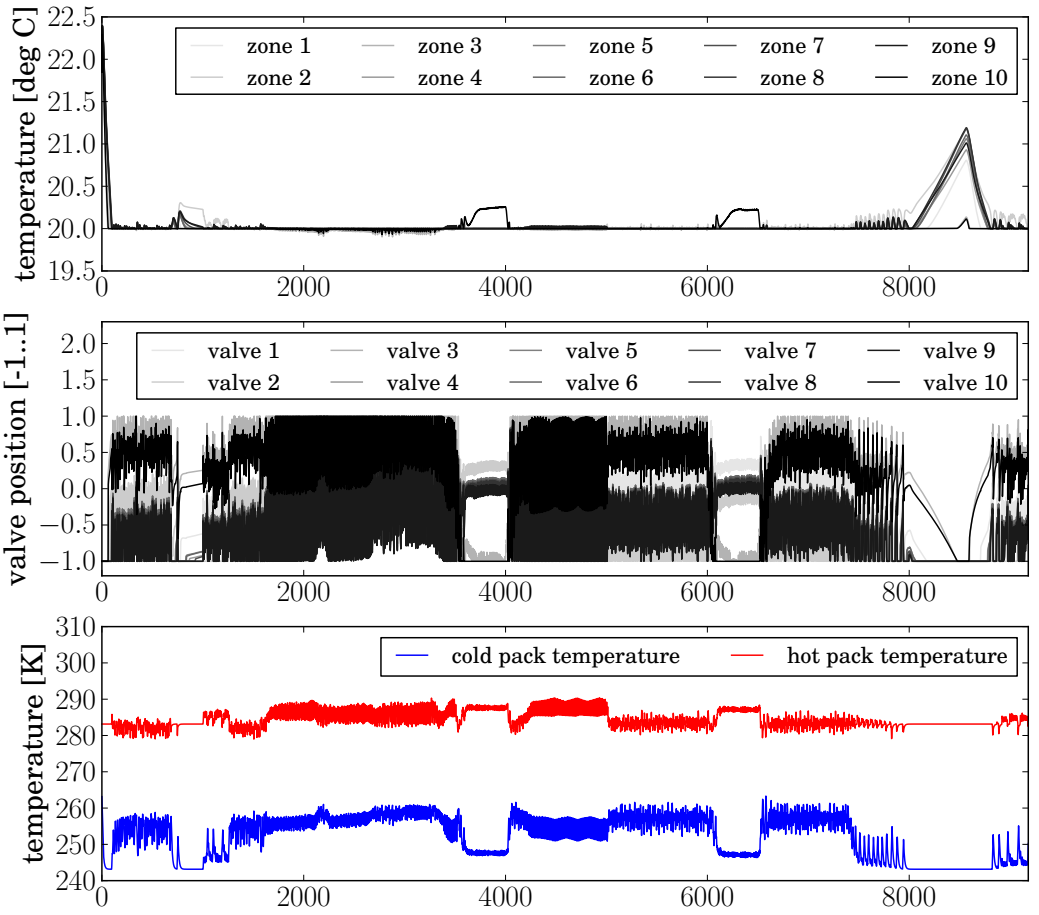


Figure 5.6: *Simulation results for the cabin temperature regulation system*

weak, a number of BLSMC SISO-controllers can be used.

Another limitation is given by systems that include a hard dead time. Numerical experiments (not shown here) showed that the introduction of delay-elements, however small, completely prevented any form of convergence.

Controllers based around sliding mode concepts do not have a well-defined gain. Small deviations in behaviour of the controlled system can however result in large changes in controller output. In this sense, they behave similar to linear controllers with large gain. As such, they can be classified as aggressive controllers. If simulation models are not robust, there may be situations where a BLSMC prevents the numerical solver from finding simulation results. For the same system, a cautiously tuned PI-controller will probably give better results.

Nevertheless, for a large class of simulation models, the proposed control concept will deliver good results, taking up only very little of the modelling experts time. As long as modelling experts keep the mentioned limitations in

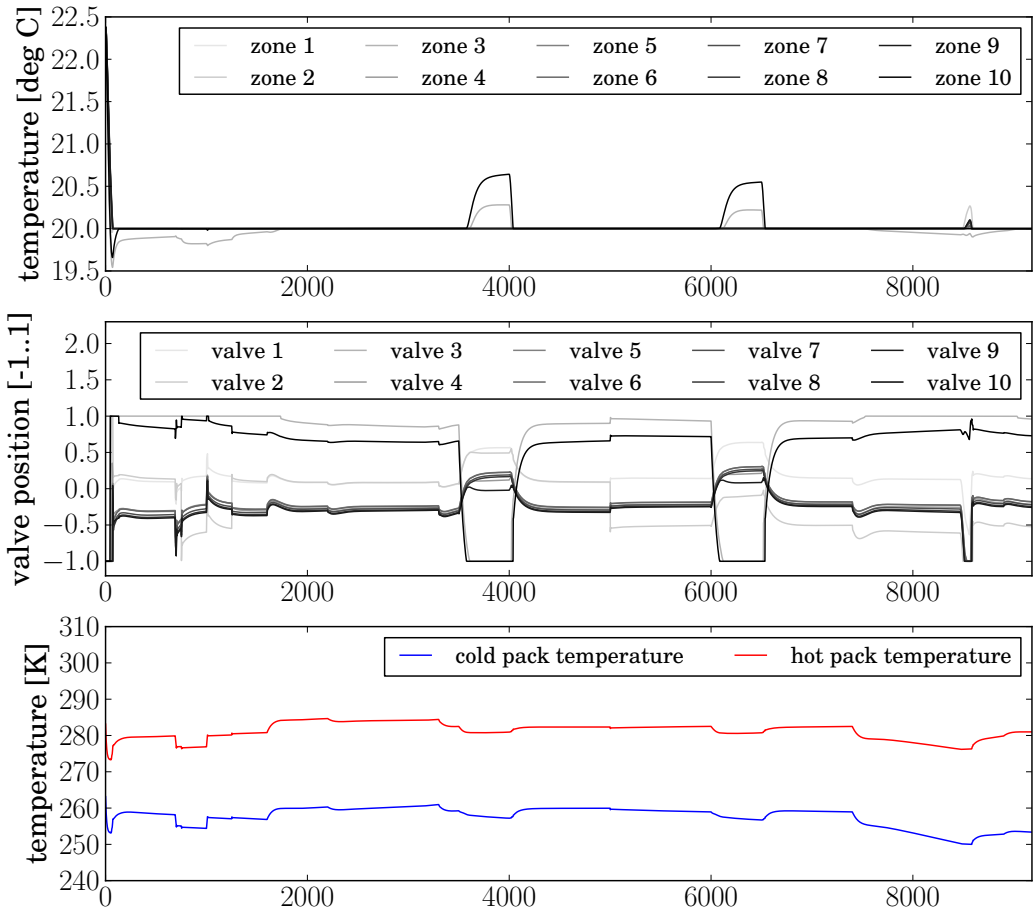


Figure 5.7: *Simulation results using SMC for the valve controls only*

mind, the proposed controller can greatly improve productivity.

CHAPTER 6

Usability aspects of equation-based modelling

"Magic. Where does it come from?"
"I do not know," said the man.
"And neither does anyone else?"
"Oh, the situation is far worse than that, Mr. Potter. There is hardly a scholar of the esoteric who has not unraveled the nature of magic, and every one of them believes something different."
"Where do new spells come from? I keep reading about someone who invented a spell to do something-or-other but there's no mention of how."
A shrug of robed shoulders. "Where do new books come from, Mr. Potter? Those who read many books sometimes become able to write them in turn. How? No one knows."
"There are books on how to write -"
"Reading them will not make you a famous playwright. After all such advice is accounted for, what remains is mystery. The invention of new spells is a similar mystery of purer form."
The man's head tilted.
"Such endeavors are dangerous. The saying is that one should either not have children, or else wait until after they are grown. There is a reason why so many innovators seem to hail from Gryffindor, rather than Ravenclaw as might be expected."

Eliezer Yudkowsky, Harry Potter and the Methods of Rationality

6.1 Problem definition

The every-day business of a modelling and simulation expert contains a significant amount of thinking of how to model a physical system in an adequate way. To make related tasks easier, modern modelling and simulation languages as well as tools have evolved with performance, flexibility, and usability in mind. Tool developers and language maintainers usually concern themselves with each of those aspects. The situation appears a bit different in scientific literature, where the first two aspects dominate the discussion, while the topic of usability is often neglected.

There are several relevant reasons for this mismatch. First, usability in general is hard to measure in any meaningful way. While benchmarks for simulation performance are readily available, no such thing exists for the understandability of models. Second, the usability of tools and the comprehensibility of models are features that are often taken for granted if done correctly. Only blatant flaws are noticed consciously, and even then, often forgotten.

In industrial practice, the performance, quality and integrity of simulation models can be limited by the comprehension of the modelling expert, if models get complex enough. This illustrates the significance of the corresponding research gap.

Of course, related research has been done in connected fields. In [93], general guidelines on programming language design are addressed. [94] and others tried to predict the time it takes to perform an atomic programming task, like inserting a word in a text editor. The authors of [95] conducted a review about usability aspects of software and coding styles. The field of software ergonomics deals with the effect of software design on the performance of the user [96]. This relates mostly to usability topics, implementation issues are addressed less often. [97] presents a framework of cognitive dimensions that play a role in understanding, learning and using a programming environment. In [98], this framework is used to analyse some aspects of the C++ programming language. The works of [99] and [100] mention several categories to evaluate programming languages. The following papers are diving into the domain of physical modelling: [101] describes how cognitively efficient visual notations can be designed, this relates directly to graphical layers of modelling languages. [102] presents a method to evaluate information modelling methods like unified modelling language (UML). Pedagogical aspects of modelling languages are addressed in [103], where an interactive teaching environment for the *Modelica* language is presented. These citations barely skim over the vast amount of work done in this research area. However, to the best knowledge of the author, no research regarding language design and no quantitative usability studies have been published in the context of equation-based modelling languages.

In this chapter, interesting psychological aspects in equation-based modelling

are identified, and a first attempt at quantification is made. Also, a foundation is laid for further research regarding the development of equation-based languages and corresponding tools.

6.2 Literature summary

In this section, results of previous research undertakings are presented. Most of the mentioned papers examined programming languages. The survey is especially focused on results that should (by the intuition of the author) also be relevant for equation-based modelling languages. Despite the quantity of papers growing a lot since around the year 2000[1], it was found that the percentage of interesting results is larger in older papers. Therefore, they make up the majority of mentioned sources.

When evaluating programming languages, analysing simple tasks like writing a single line of code is not adequate. Broad-brush analysis is more suitable, because it avoids "death by detail" [97].

For mechanical problems, different representations (diagram vs. symbolic) can carry the same information but differ greatly regarding cognitive effort [104].

A notation cannot highlight every aspect of information at once, but a good notation makes obscured (non-highlighted) information more visible [105]. This is especially relevant when looking at the ongoing discussion regarding custom annotations in Modelica [106].

Programs are represented mentally at a higher level than just code [97]. The representations are different for programs written in Basic, LabView and spreadsheets [107]. Mentally absorbing information is much easier (and more performant) if there is a match between the external representation and the mental representation [108].

In the work of Sporer and Soloway [109], two folk wisdoms about beginner mistakes are tested. The first one - "just a few types of bugs can account for a majority of the mistakes in students programs" - seems to hold true. The second one - "most bugs can be attributed to student misconceptions about language constructs" - does not seem to correspond to reality. Instead, bugs seem to arise as a result of plan composition problems, or, "difficulties in putting the pieces of a program together". The second result was later confirmed in [110].

In [111] it was postulated that the biggest difference between novice and expert programmers is the amount of strategies (for planning or debugging) that are available to the expert. From this it can be derived that an ideal modelling language should make at least some planning- and debugging strategies available

[1] A search on Google Scholar for the term `(software OR programming) AND (usability OR comprehension OR ergonomics OR psychology)` found 39.500 results for the years 1975 to 1985, 141.000 results for the years 1985 to 1995, 874.000 results for the years 1995 to 2005, 781.000 results for the years 2005 to 2015.

to the user. However, in [112] it is mentioned that experts talk more in terms of abstractions while novices tend to concern themselves with single words of code.

Low-level primitives are a potential cognitive barrier for programming novices ([113] as cited in [114]). Users are happy, if the language primitives map directly onto their problems, instead of having to build high-level constructs from low-level primitives themselves [115].

Newer research seems to suggest that at least some developers try to avoid comprehension of programs whenever possible, for example by copy-pasting code snippets that are known to work [116]. In the same work it was also observed that developers tried to put themselves in the role of the end-user whenever possible.

Programming ability does not seem to be correlated with age, sex or educational attainment. The effect of general intelligence is unclear, with some studies showing a positive correlation, while others do not. Some known predictors are consistency during the solving of ambiguous problems, possession of a mechanical mental model of computers[2] and expressed self-confidence before the first programming course [117][118][119].

6.3 Explorative expert surveys

As seen in section 6.2, a large amount of research has been conducted on cognitive aspects of programming languages. The same cannot be said for equation-based modelling languages, where nothing similar has been published to the best knowledge of the author. For an initial stake-out of this new field, a series of structured surveys was conducted.

6.3.1 Method

Five voluntary participants were recruited from the colleagues of the author at the German Aerospace Center (DLR) Institute of System Dynamics and Control and connected working groups. Among participants three were male and two female. The participants were aged between 22 and 50 years. They had previous experience with Modelica between 2 and 180 months. As an introduction, the following statement was given:

"We are doing an explorative expert survey, where experts for object-oriented modelling of differing experiences are taking part. We want to know how people are approaching modelling tasks, and how modelling languages or tools are helping or constraining them. We invited you to this survey since you are using the equation-based modelling language Modelica productively. There are 15

[2]Some participants tried to be gentle to the computer, touching the computer mouse softly. They hoped that the computer would return the favor by doing what they wanted. This would be the opposite of a mechanical mental model.

questions to this topic. Later we filter out interesting aspects from your answers, possibly translate, and summarise them. For this reason there are no right or wrong answers. Feel completely free to skip as many questions as you like."

Participants were then asked if they wanted to conduct the survey in writing by filling out a word-document or verbally, including a recording of the interview. All participants chose the written version. The questions given to the participants were as follows:

1. How did you obtain your Modelica knowledge?
2. Which kinds of tasks do you use object-oriented modelling languages like Modelica for?
3. Which aspects of the Modelica language took a long time to understand? Which aspects of the language do you still not understand?
4. What are the first steps you take while approaching a modelling task?
5. How do you split up your time for different parts of your modelling tasks? [concept, development, validation, documentation, productive usage and maintenance]
6. What do you like/dislike about the Modelica tools that you know?
7. Do you use a certain coding style?
8. Do you know some style templates?
9. What do you like/dislike if you have to use code developed by other people?
10. When developing big models, how do you test intermediate results?
11. When developing big models, how often do you test intermediate results?
12. How much do you trust models that you have developed yourself?
13. How much do you trust models that are developed by others?
14. What is needed for you to gain trust?
15. Is there anything else regarding this field that you want to share?

6.3.2 Selected results

1. Experienced participants mostly got their knowledge from colleagues, and from learning by doing. Novices mostly relied on books.
2. Participants are using Modelica for simulation of physical systems, development and tuning of control systems, derivation of linearised models (using the *LinearSystems-library* [70]) and simulation of failure cases.
3. The relationship between text- and graphical modelling, connecting vector- and matrix-based blocks, the replaceable/redeclare functionality, event-handling and embedding of C-code were seen as obstacles during the learning phase. None of these points except replaceable/redeclare was mentioned by more than one participant.
4. While novices start new modelling tasks by *divide-and-conquer* strategies, experts often start with a simple model which is later augmented.
5. Expert modellers refused to give numbers, based on the case dependency of the question. Novices gave wildly varying numbers: 1-20% for the concept

phase, 19-45% for development phase, 15-70% for validation, 5-20% for documentation, and 0-5% for both productive use and maintenance.

6. [there were no usable answers to this question]
7. Participants did not know any formalised coding style. One participant remarked the usage of a recurrent structure: import, public parameters, protected parameters, public models, protected models, equations, connects, documentation. The same participant noted that the use of graphical modelling makes structuring of the code layer difficult.
8. One participant mentioned the *DLR implementation guidelines for Modelica libraries*, the remaining participants did not know of any style guides.
9. Almost all participants positively mentioned documentation and comments to help with the understanding of models developed by other people. They also liked structured libraries, a consistent coding style, and examples. Two participants strongly disliked a mixture of graphical and textual modelling in a single component.
10. The usual strategy reported by novices and experts alike is to develop *Wrapper*models where the actual model is subjected to typical boundary conditions, or, if available, test data. This wrapper model also serves as a usage example for others.
11. Most modellers test their models every time a submodel, or a new physical functionality, is implemented. Only one participant had another strategy, where the complete system is developed before the testing-phase starts.
12. Modelica developers had medium to high levels of trust regarding their own models. There does not seem to be a correlation between trust in the own models and modelling experience.
13. Trust regarding models from other developers seems to be mixed. Model source (colleagues vs. anonymous website) and code quality were mentioned as influencing factors.
14. The following features were mentioned as trust-gaining factors: code clarity, quality of documentation, availability of test cases, availability of validation data, discussions between associates.
15. [there were no usable answers to this question]

6.4 The cost of inheritance

There is a reason why inheritance is less popular in equation-based languages than in conventional programming languages: typically in equation-based models, only the leaves of the inheritance tree are directly usable and concrete components. The inner elements of the tree mostly are all abstract and not ready for use. This forms a contrast to programming languages where often also the inner elements of the tree represent concrete and fully functional classes. Inheritance would probably be less popular among programming languages if

there would be only abstract father-classes. On the other hand, inheritance introduces inherent intricacy to a model [120].

The situation could be improved for equation-based languages if not only variables, or parameters can be added but also changed (or even removed). In this way, also code from concrete, usable components can be reused.

Nevertheless, an experiment was set up to quantify the extra cognitive load derived from inheritance in Modelica models.

6.4.1 Method

Design

Subjects were asked to open and study the code of models from the Modelica Standard Library (MSL) until they would feel comfortable using them in bigger models. The models were selected such that they did not contain relevant graphical annotations (as in: no composite models that where connected on the graphical layer) and varied in complexity. Subjects were told to record the time they needed to understand the model in the sense explained above. In particular, they were asked to also study inherited models.

The Modelica models selected for this study varied in complexity according to three factors: (1) number of symbols, (2) number of equations, and (3) total number of *extends* clauses used in the model. The number of symbols was counted using Word after deleting the main *annotation* blocks containing the model documentation and excluding whitespace. The subjects were told not to examine *extends* clauses leading to mere *Modelica.Icons* models.

The expectation was that each of these factors independently would lead to increased time to understand the models. Additionally, it could be expected that experts would understand complex models quicker than participants with less Modelica experience.

Participants

Twelve voluntary participants were recruited from colleagues at the DLR Institute of System Dynamics and Control. One of the participants only reported estimated timing measures and was therefore excluded from the study. Among the remaining 11 participants 10 were male and 1 female. The participants were aged between 22 and 35 years. They had previous experience with Modelica between 2 and 132 months (mean (M) = 49.36, standard deviation (SD) = 44.55).

Material

Ten models were used for the experiments as summarised in Table 6.1. The table contains the path in the MSL, the number of symbols *sym*, the number of

equations *eq*, and the number of inheritance levels *lvl*. The models were listed in random orders for each participant.

	Complexity			Time to understand	
Component Name	sym	eq	lvl	M[s]	SD[s]
Blocks.Discrete.FirstOrderHold	1514	7	2	269	62
Blocks.Continuous.SecondOrder	1285	2	1	94	21
ComplexBlocks.ComplexMath.Add	584	2	1	55	10
ComplexBlocks.Sources.ComplexStep	738	2	2	83	9
Electrical.Analog.Basic.Capacitor	634	6	1	58	16
Electrical.Analog.Ideal.IdealDiode	1452	10	2	389	89
Mechanics.Rotational.Interfaces.Flange_a	171	0	0	26	5
Mechanics.Translational.Components.Mass	1071	7	1	91	21
Thermal.HeatTransfer.HeatCapacitor	489	4	0	59	15
Thermal.HeatTransfer.Sources.PrescribedHeatFlow	566	2	0	59	9

Table 6.1: *Models used in the experiments*

Procedure

The experiments were conducted in the office of each participant. We asked the participants to open a window of the *Dymola* tool and load the MSL. We then presented them with the list of models and asked them to examine one model after the other until they would understand them. The participants were asked to take the time it took to understand each model and were then left alone. After the experiment, the results were collected and the intention of the study was explained. In the following discussions, some subjects reported problems during the experiment, such as being interrupted or not having understood a model at all. The timing of these models were subsequently excluded from the analysis. This resulted in 102 considered cases out of the 110 complete measurements.

6.4.2 Results

The time to understand a model varied between 10s and 567s (M = 96.30s, SD = 94.64s) for the valid measurements, indicating a wide range of considered model complexity. Averaged over all measurements for each participant, the mean time to understand a model varied between 65.3s and 213.5s (M = 121.39s, SD = 56.29s). This variation will be explained mostly by effects of the participants' experience in section 6.4.2.

Correlation of the independent variables

Since all three independent variables measure the complexity of the model in a way, they are naturally heavily correlated ($r > 0.53$) and they all increase the

time it takes to understand a model. In order to better condition the regression problem, the number of equations was not used as an indicator of the overall model complexity. Preliminary analyses had shown the influence of the number of equations on the time to understand a model to be significant but small, when correcting for the other two factors.

Controlling for experience

In order to check the assumption that experience influences the time to understand, a regression analysis is conducted according to

$$time = a + b_{exp} \cdot exp + b_{sym} \cdot sym + b_{\times} \cdot exp \cdot sym, \tag{6.1}$$

fitting the time to understand on the participants' experience ($b_{exp} = 0.191$ s/month, $t(98) = 0.6$, $p = 0.58$), the number of symbols in the model ($b_{sym} = 0.200$s/1, $t(98) = 8.1$, $p < 0.001$) and, especially, the interaction of both variables ($b_{\times} = -0.001$s/month, $t(98) = -2.9$, $p = 0.005$).

Note that values of $p < 0.05$ indicate a statistically significant contribution of the respective predictor variable with effect size b. In order to obtain this significance level, $t(n)$ tests in $n = 98$ degrees of freedom are conducted. The tests compare each effect size b against its associated standard error. Assuming an effect size of zero and normally distributed measurements (null hypothesis), the probability to observe the actual effect size b (or stronger) is given by p. That means, small values in p indicate that the null hypothesis has to be abandoned and that the observed effect is indeed statistically significant.

The regression results show that each additional symbol contributes with $b_{sym} = 0.2$s to the time to understand a model. Additionally, this contribution decreases with growing experience ($b_{\times} < 0$). The main effect of the experience is not statistically significant in this analysis. In the following, in order to account for the interaction between experience and number of symbols, the number of symbols is regarded a random effect pertaining to repeated measures on the subjects (see section 6.4.2).

Figure 6.1 qualitatively illustrates the effect of experience on the time to understand. The mean time to understand a model decreases with growing experience. The same pattern is observed for the influence of the number of symbols on the time to understand. This indicates that the differences in mean time to understand a model are indeed caused by the complexity of the model, and not by a main effect of experience. Recall that the mean number of symbols in this study is 850.4 (see Table 6.1). This number corresponds roughly to the ratio of mean time to understand and the symbol effects.

Effect of hierarchy levels

The effect of hierarchy levels within a model on the time to understand is estimated using mixed-effects regression with random effects of the number of

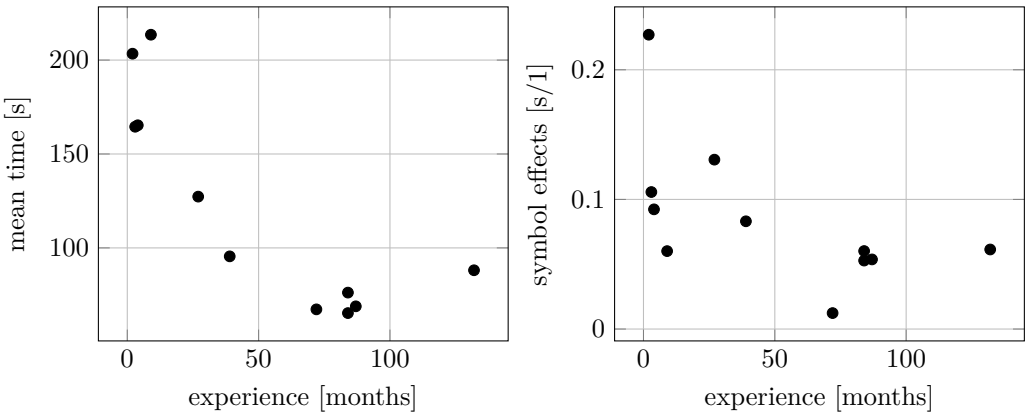


Figure 6.1: *Random effects as a function of subjects' experience*

symbols according to

$$time = a + b_{lvl} \cdot lvl + b_{sym}(exp) \cdot sym.$$

In this model, a first regression is conducted for each participant, estimating the random effects $b_{sym}(exp)$ which are different for each participant. These are the values shown in Figure 6.1. In the second step, a regression on the residual data set is conducted in order to estimate the fixed effect b_{lvl} which is assumed fixed for all participants. The fixed effect of hierarchy levels is found significant in this analysis ($b_{lvl} = 26.65\text{s}/1, t(90) = 2.8, p = 0.006$). A fixed offset is again not found to be statistically significant ($a = -1.25\text{s}, t(90) = -0.1, p = 0.9113$).

6.4.3 Discussion

We could provide some amount of experimental evidence for one particular piece of the given modelling advice: the significant effect of hierarchy levels on the time to understand a Modelica model supports the hypothesis that flat hierarchy levels are easier to understand than deeply nested model hierarchies.

However, this study can merely be regarded as a pledge for more evidence based support of modelling strategies. There are many things, which are not addressed in the experiments. First, many variables typically controlled in psychological experiments are not regarded, such as gender or age. (Admittedly, at least the first variable is difficult to cover in engineering environments.) Second, the experiment was not conducted in a controlled environment. For example, participants reported very different procedures for opening extended classes, suggesting that the additional time per *extends* clause is partly due to finding the extended model in the MSL. Some development tools like Dymola offer a (albeit hidden) functionality to flatten a model and represent the variables and equations in a single file (*flat Modelica*), but none of the participants knew about this functionality during the experiment. Third, Modelica models

have many more properties, such as documentation and graphical annotations, which need to be considered for sustainable modelling advice. Finally, time to understand may be argued not to be the best indicator for good modelling style as compared to, e.g., finding errors or actually using the models.

6.5 Viscosity of representations

In section 6.2, several findings from literature are cited concerning representations of information or programs. To summarise those findings, mental representations of computer programs are different for various programming language, and programming performance is higher if there is a match between the external representation and the mental representation ([97], [104], [107], [108]). We postulate that a similar dependency exists in the field of physical modelling.

In equation-based languages, several different representations are common. This is also true within the scope of individual equation-based languages. For example, in Modelica it is possible to model a system by writing down the necessary physical equations, or alternatively by joining together the necessary subcomponents graphically.

Based on this, an experiment was set up to quantify several performance metrics for different representations.

6.5.1 Method

Design

Participants were shown four different Modelica models, or representations, of physical systems. They were asked to *identify* those systems, *predict* the transient response, and *rate* their confidence for those models. The models were created in such a way, that they represented each physical system in four different ways. The time needed by the participants for the identification and prediction tasks was measured.

For each of the four physical systems considered, four different Modelica models were created, to a total of 16 different models. Participants were unknowingly split up into random groups of equal size. Each group was presented with one representation of each physical system. To eliminate first order training and carryover effects, a balanced latin square design[3] as described

[3]If participants have to do multiple tasks, one after another, several effects are at work. One task might be more difficult than another, resulting in participants taking longer to complete this task. On the other hand, if the tasks are similar, participants will have some experience by the time they get to the last task, resulting in a shorter time to complete the task. Also, some of the tasks might be a good preparation for others, and vice versa. To isolate this effects, a balanced latin square design can be used. Here, the participants are divided into several groups. The groups are assigned to tasks in a specific order, designed to eliminate the experience (or training) and preparation (or carryover) effects under the assumptions that these effects are first order.

Mechanical Engineering or similar	43
Electrical Engineering or similar	11
Mathematics	9
Engineering (other)	8
Physics	3
Other	2
Science (other)	2
Computer science	0
(not answered)	1

Europe	51
North-America	14
Asia	12
(not answered)	1

Table 6.2: *Origin and background of participants*

in [121] was used to assign representations to physical systems for each group of participants.

Effects of physical systems are not considered in this work, therefore the order of physical systems was the same for each group. The effect is thus indistinguishable from training effects and it is accounted for by the balanced latin square design.

The expectation was that the percentage of correctly executed tasks, as well as timings and reported confidence would vary between the different representations.

Participants

Since knowledge in Modelica was necessary for participation in the experiment, participants were recruited from the Modelica User Groups Sachsen, North America, Japan, Hamburg and Baden-Württemberg, as well as colleagues of the author at the Institute of System Dynamics and Control. Participants were not compensated. Over the course of one month, 98 participants took part in the experiment. From those 20 participants were excluded based on having not completed the tasks (5), unrealistic timings and reported technical problems (5), reports of being interrupted (9), or reporting a Modelica experience of zero (1).

Among the 78 remaining participants, 8 reported being female, 69 male, and 1 other. They were aged between 23 and 63 years (mean $M = 36.7$, standard deviation $SD = 8.7$). They had previous experience with Modelica between 0.1 and 17 years (mean $M = 5.2$, standard deviation $SD = 4.4$). Participants reported their professional background based on the provided categories shown in Table 6.2. Notably, most participants were engineers and there were no computer scientists. Table 6.2 shows the place of living of the participants ordered by continents. Most of them lived in Europe or North America.

systems		representations	
SD	a parallel spring-damper with connected mass	MSL	graphical models built from MSL components
OC	an electric resonant circuit without damping	EQ	models based on physical equations
T	two bodies in thermal contact	BLOCK	block diagrams built from the MSL.Block-package
BB	a mass bouncing on a floor	ALG	models implemented as algorithms

Table 6.3: *Physical systems and representations used in the experiment*

Material

For four different physical systems, four models were created each in Modelica, resulting in 16 models in total. The types of systems and representations are listed in Table 6.3.

Models were developed in such a way that they were behaving equally for each system, while keeping the models as simple as possible. All models for the Spring-Damper system are shown in Figures 6.2, 6.3, 6.4 and 6.5.

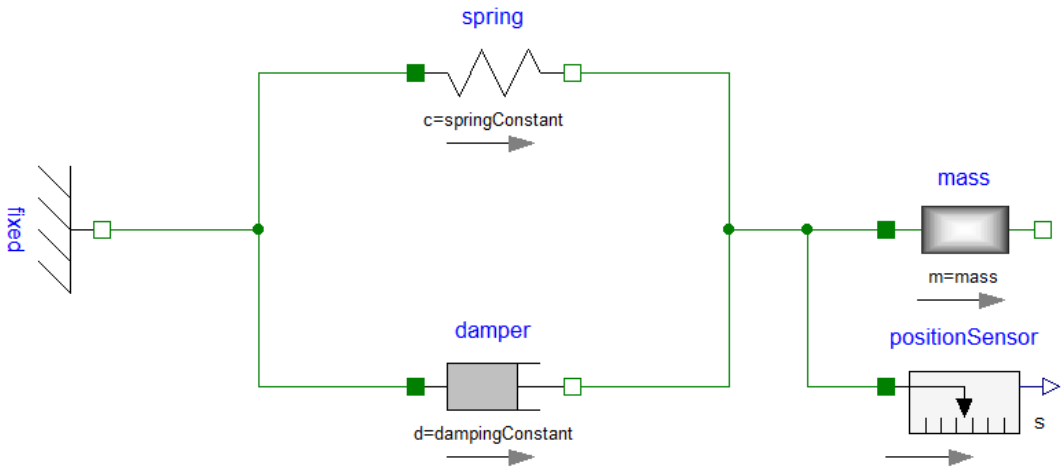


Figure 6.2: *MSL - graphical model of a Spring-Damper*

For each of the systems, eight names were created. Of those eight names, four were physically motivated, four were motivated from a system dynamics point of view. From each group of four, one of them correctly described the system. For example, the Spring-damper system was assigned the following names (correct names written in bold):

(1) **parallel spring-damper with connected mass** (2) force amplifying system with damping (3) electric resonant circuit without damping (4) op-

```
model Test1
  parameter Modelica.SIunits.Length          position0;
  parameter Modelica.SIunits.DampingConstant dampingConstant;
  parameter Modelica.SIunits.SpringConstant  springConstant;
  parameter Modelica.SIunits.Mass            mass;

  Modelica.SIunits.Length       position;
  Modelica.SIunits.Velocity     velocity;
  Modelica.SIunits.Acceleration acceleration;
equation
  //force balance
  0 = -    acceleration      * mass
      -      velocity        * dampingConstant
      - (position-position0) * springConstant;
  der(velocity) = acceleration;
  der(position) = velocity;
end Test1;
```

Figure 6.3: *EQ - equation-based model of a Spring-Damper*

```
model Test2
  parameter Modelica.SIunits.Length          position0;
  parameter Modelica.SIunits.DampingConstant dampingConstant;
  parameter Modelica.SIunits.SpringConstant  springConstant;
  parameter Modelica.SIunits.Mass            mass;

  Modelica.SIunits.Length   position;
  Modelica.SIunits.Velocity velocity;
  Modelica.SIunits.Force    force;
algorithm
  force          := -      velocity       *dampingConstant
                    - (position-position0)*springConstant;
  der(velocity) := force/mass;
  der(position) := velocity;
end Test2;
```

Figure 6.4: *ALG - algorithm-based model of a Spring-Damper*

erational amplifier (5) **second-order system with damping** (6) first-order system with connected integrator (7) proportional, integral, and derivative systems connected in parallel (8) non-linear second-order system without damping.

Furthermore, for each of the systems, nine images were created that illustrated possible trajectories. Of those trajectories, only one had an adequate connection to the system.

Figure 6.6 shows the possible answers for the spring-damper system (correct answer framed).

Procedure

The experiment was conducted online using *SoSci* survey [122]. Tasks were implemented as multiple-choice tests. Timings were measured using asynchronous

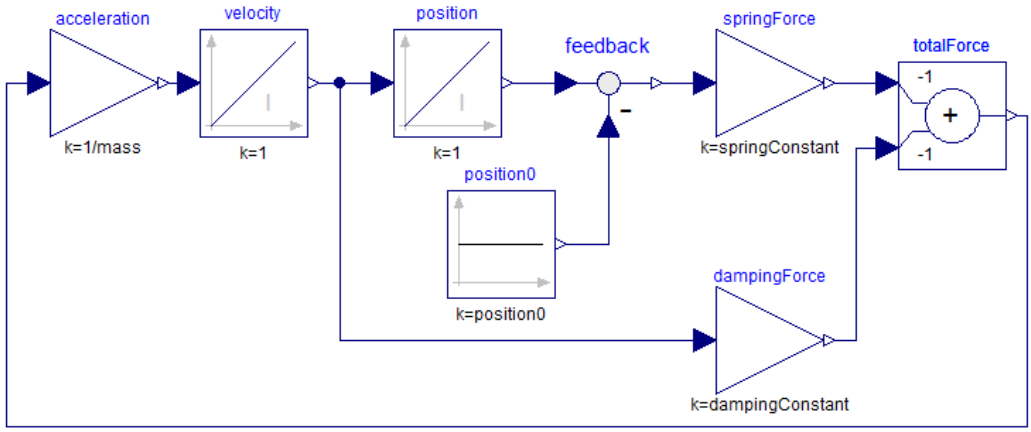


Figure 6.5: *BLOCK - block-diagram model of a Spring-Damper*

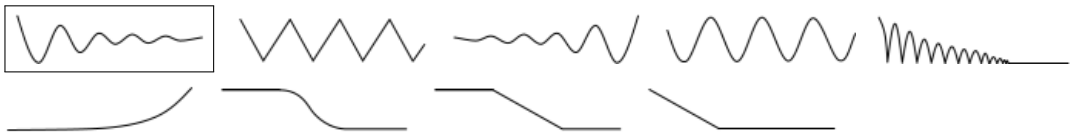


Figure 6.6: *Answer options for the transient responses of the damped spring-damper system (correct answer framed)*

java-script and extended mark-up language (XML) with an accuracy of a few milliseconds, independently of the internet connectivity.

At the start of the experiment, participants were shown a starting page, and were unknowingly assigned to one of four groups. Assignment was based on the number of completed experiments for all groups.

Before each task, participants were given a short introduction to the task. For the first two tasks, this introduction also included a request to *"hurry up"*.

During the first task of the experiment, participants were shown four different combinations of systems and experiments and had to identify the systems in questions. The available answers for each combination were scrambled randomly. The assignment of the 16 combinations of systems and representations was done using a balanced latin square approach in the representations and constructed as shown in [123]. The assignment can be seen in Table 6.4.

During the second task, participants were shown the same four combinations. This time, they were asked to predict the transient response. Again, the order of answers were scrambled.

During the third task, participants were again shown the same four combinations in the same order. They were asked how much they agree with the following statement: *"I would feel confident using this model."* Participants could

	subtask 1	subtask 2	subtask 3	subtask 4
group 1	OC-BLOCK	T-EQ	SD-MSL	BB-ALG
group 2	OC-EQ	T-ALG	SD-BLOCK	BB-MSL
group 3	OC-ALG	T-MSL	SD-EQ	BB-BLOCK
group 4	OC-MSL	T-BLOCK	SD-ALG	BB-EQ

Table 6.4: *Assignment of system/representation-combinations to groups*

choose from a five-point *Likert* scale[4], ranging from *"-2 (Strongly disagree)"* to *"+2 (Strongly agree)"*. This time, the order of answers was not scrambled.

For each task, the first subtask does not give valid timing results since the browser has to load the content first, and the participant has to understand the style of question. Therefore, for each task, an additional training subtask was added at the beginning. For example, at the beginning of the identification task, participants were shown a picture of bananas, together with the following question: *"Let us start with a small warmup question to get into the mood: What system is represented here (click one of the 2 correct options)?"*. Possible answers were: *"bananas"*, *"apples"*, *"yellow fruits"*, *"green fruits"*.

Performance in any kind of task can be effected by so called *stereotype threat*[5]. To lower the effects of stereotype threat on the results of this study, the actual tasks were executed by the participants at the beginning of the experiment. Only after conducting all three tasks (identification, prediction of transient response, rating confidence), personal information about the participants was gathered.

Participants were asked if they were interrupted during the experiment. They were also asked to estimate their Modelica experience in years, their professional background (dropdown-menu), age, gender and place of living (dropdown-menu with continents). Finally, they could make a guess regarding the goal of the experiment and could give miscellaneous remarks. Interested participants could also leave their E-Mail address to be notified of the experiments results, those addresses were stored independently, for the results to remain anonymous.

[4] A Likert scale is a psychometric scale to quantify responses. If participants are asked: "How much do you love your dog?", some answers might be: "a lot" or "with all my heart" or "meh...". Mapping these answers to a love-metric will be highly arbitrary at best. Here, Likert scales should be used instead. The better question will be: "How much do you agree with the following statement: I love my dog.", while participants can choose from three, five or seven (fun fact: the human brain cannot meaningfully differentiate between more than seven grades of a single concept) predefined answers, ranging from "agree" to "disagree". Odd numbers are used, because a neutral response has to be included. These answers are then translated to numbers. Often, equal distances between two neighboring answers are assumed. These numbers constitute the new attribute, which is far easier to analyse.

[5] if participants are under the impression that a sociocultural group they belong to is negatively stereotyped for that particular task, performance gets distorted to fit that stereotype [124, 125, 126]. For instance, women underperformed men in a math test, when the test got described as producing gender differences; the difference could be eliminated when the test got described as not producing gender-difference [127].

	identification type		identification errors		prediction errors		rating	
	Math.	Physical	Correct	Wrong	Correct	Wrong	Mean	SE
MSL	10	68	59	13	61	13	0.87	0.16
EQ	25	53	54	9	51	14	0.43	0.16
ALG	19	59	64	10	61	11	0.27	0.15
BLOCK	24	54	65	11	57	17	−0.31	0.15

Table 6.5: *Summary of answers grouped by type of representation*

Internal review

Models were individually reviewed and modified by Alexander Pollok, Andreas Klöckner and two additional colleagues at the Institute of System Dynamics and Control, to keep code and model quality equally high for each of the 16 models.

Both system names and trajectories were checked for ambiguity by the same people. In some cases the problem definition was augmented to ensure that each problem had a unique solution, for instance by adding information about the sign of a damping constant.

The complete experiment was run as a pretest by eight colleagues of the author to ensure functionality and comprehensibility. The data from those experiments was discarded.

Participants involved in pretests and internal reviews were compensated generously with cookies.

6.5.2 Results

The basic answer characteristics for the three main tasks are shown in Table 6.5. There are no significant differences in error rates of identification ($\chi^2(3) = 0.69, p = 0.875$) nor prediction ($\chi^2(3) = 1.74, p = 0.627$)[6]. The type of answer (mathematical or physical) in the identification task has significant differences over the four representations ($\chi^2(3) = 9.64, p = 0.022$). The p-values of the pairwise differences are assessed in Table 6.6. In summary, the *MSL* representation leads to significantly more physical answers as compared to representation as equation (*EQ*). Table 6.5 also shows the mean rating estimates and standard errors for each representation. Models are rated best, if they are built with MSL, and worst, if they are built as block diagram. Details on these values are described later in this section.

The influence of the type of answer in the identification task on the other measures is shown in Table 6.7. Holm correction[128] is used[7]. There are no

[6] $\chi^2(4-1)$ is a metric to test the statistical independence between two attributes. Here, the independence of the error rate and the representation is tested. The smaller the number, the more probable it is that both attributes are independent.

[7] If multiple hypothesis are tested against a given significance level, the risk of having at least one false

	MSL	EQ	ALG
EQ	0.043	—	—
ALG	0.399	1	—
BLOCK	0.058	1	1

Table 6.6: *Identification answer types post-hoc p-values with Holm correction[128]*

	identification		prediction		rating	
	Correct	Wrong	Correct	Wrong	Mean	SE
Mathematical	77.3 %	22.7 %	78.1 %	21.9 %	0.37	0.11
Physical	87.6 %	12.4 %	81.6 %	18.4 %	0.26	0.15

Table 6.7: *Summary of answers grouped by type of representation*

significant effects of the answer type on identification correctness ($\chi^2(1) = 3.80, p = 0.051$) nor prediction correctness ($\chi^2(1) = 0.24, p = 0.627$). Estimated rating means are included for illustration. There is no significant effect on these either.

The reaction times and ratings are subjected to a mixed-effects regression with random intercepts

$$time = a + b(participant) + b(representation) + b(id\_type) + b(id\_correct) \quad (6.2)$$

where a represents a fixed mean intercept, $b(participant)$ represents an additional intercept for each participant, and $b(representation)$, $b(id\_type)$ as well as $b(id\_correct)$ represent fixed effects of representation, type of answer in the identification task as well as correctness of the identification task. Table 6.8 shows the respective significance tests. There are significant effects for all fixed intercepts, for the influence of representation on identification reaction time ($F(3, 202) = 12.8, p < 0.001$)[8] and on rating ($F(3, 202) = 15.2, p < 0.001$).

There is no significant difference for representation forms in the prediction task ($F(3, 188) = 1.1, p = 0.343$). This could be due to the same model being presented twice, such that the reaction time is not influenced anymore by the representation of the model, but rather by the classification result from the prior task. This is supported by the finding that the type of answer from the naming task has indeed a significant influence on the timing for the prediction task ($F(1, 188) = 30.3, p < 0.001$). A physical answer type leads to a 8.8 s quicker response than a mathematical answer type ($t(188) = 5.11, p < 0.001$). In addition, a correct answer in the identification task carries over to the

positive is automatically increased. For a layman example, see `http://xkcd.com/882`. Holm correction is used to counteract this effect by adjusting the p-values.

[8] $F(4-1, 203-1)$ is a measure for the variance of the 203 results between the 4 groups, compared to the variance of the results inside of the groups. If this number is high, the variance between the groups is comparatively high. This implies, that the groups are a good predictor for the data.

		identification			prediction			rating		
	df	res	F	p	res	F	p	res	F	p
(Intercept)	1	202	599.4	0.000	188	462.3	0.000	202	44.3	<0.001
repr.	3	202	12.8	<0.001	188	1.1	0.343	202	15.2	<0.001
id_type	1	202	3.4	0.068	188	30.3	<0.001	202	1.2	0.284
id_correct	1	202	1.4	0.238	188	5.5	0.020	202	8.3	0.004

Table 6.8: *Test statistics of linear regression models for the identification, prediction, and rating tests*

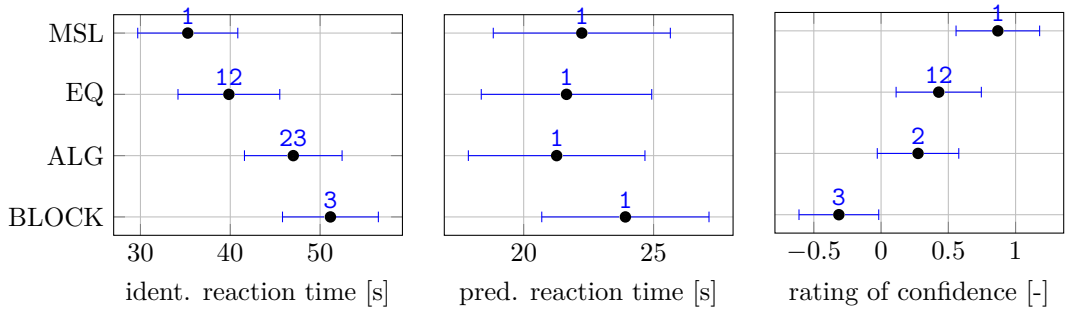


Figure 6.7: *Predicted reaction times and rating with 95%-confidence intervals and letter displays for non-significant differences.*

reaction time in the prediction task ($F(1, 188) = 5.5, p = 0.020$), indicating a more confident reaction in the second task for correct answers. This is confirmed by a significant effect from identification correctness on the actual rating ($F(1, 202) = 8.3, p = 0.004$).

Estimated means of the fitted models per representation type are shown in Table 6.7 along with their 95% confidence interval and letter displays for non-significant differences[9]. They are computed using the fitted values and averaging over participants, the different types of identification answer, and the correctness of the identification answer. Letter displays indicate groups, in which no significant differences between the representation forms are found. For example, there is no significant difference in the identification reaction time between MSL representation and EQ representation. But there is a significant difference between MSL representation and *ALG* representation. Table 6.9 shows the values of these differences along with the respective test statistics.

The graphical MSL representation performs best in both the identification and in the rating task. The *BLOCK* diagram representation performs worst.

[9]Letter displays indicate groups, in which no significant differences between the representation forms are found. For example, there is no significant difference in the identification reaction time between MSL representation and EQ representation. For this reason, they both contain the letter 1. There is a significant difference between MSL representation and ALG representation. For this reason, they do not have a letter in common.

reaction times for naming					
contrast	estimate	SE	*df*	*t*	*p*
MSL - EQ	-4.58	3.10	202	-1.47	0.455
MSL - ALG	-11.75	2.90	202	-4.05	<0.001
MSL - BLOCK	-15.88	2.91	202	-5.46	<0.001
EQ - ALG	-7.17	3.01	202	-2.38	0.084
EQ - BLOCK	-11.30	2.99	202	-3.79	0.001
ALG - BLOCK	-4.14	2.84	202	-1.46	0.466
rating of preference					
contrast	estimate	SE	*df*	*t*	*p*
MSL - EQ	0.44	0.19	202	2.30	0.101
MSL - ALG	0.59	0.18	202	3.30	0.006
MSL - BLOCK	1.18	0.18	202	6.58	<0.001
EQ - ALG	0.15	0.19	202	0.83	0.842
EQ - BLOCK	0.74	0.18	202	4.03	<0.001
ALG - BLOCK	0.59	0.19	202	3.34	0.005

Table 6.9: *Predicted differences between forms of representation for reaction times and rating*

There are no significant differences between the EQ equation and the ALG algorithm representations. This can likely be explained by the two representations resembling each other strongly.

6.5.3 Discussion

Modern modelling languages give experts a large amount of freedom regarding the choice on how to represent a physical system. It was found that this choice has a significant impact on the performance of modelling experts for varying tasks.

Based on the conducted experiments, graphical representations using the MSL should be preferred in most cases. Tasks were performed faster, and participants showed a higher amount of confidence. The opposite can be said for the representation by block diagrams. Achievable time savings can be rather important. A difference of up to 15.88 s was found between MSL and BLOCK representations. This corresponds to roughly one third of the absolute reaction time. Maximal rating difference was 1.18 between the same two representations, which also corresponds to one third of the total scale used for rating the model.

Textual models, represented by equations or by algorithms, can be grouped between graphical MSL representations and block diagrams. No significant difference was found between these textual models. While equation-based representations might be more common in the Modelica community, the final choice will usually depend on the use case. For example, algorithms might be used

more often when controllers are modelled, or when non-physical computations are represented.

Interestingly, the amount of errors was similar for all representations, while the necessary time was significantly different. This suggests that modelling experts completely compensate for tasks of higher difficulty by just taking more time. This is consistent with general finding on the speed-accuracy tradeoff and also manifests in the amount of confidence expressed explicitly.

While no direct influence of type of representation was found on the reaction time for the prediction task, there were significant influences mediated by the mental representation of the model. This can be counted as evidence towards more subconscious confidence in a physical than in a mathematical mental model. Another mediator variable is the correctness of the identification answer, which has significant influences not only on subconscious measure of prediction time, but also on the consciously expressed confidence in the model. Together, these results also suggest to prefer physical MSL models over equation-based or block-based models, in order to increase the confidence of users in the model and to increase the ease of working with these models.

External Validity

Results of these experiments are restricted to engineers and other users of equation-based modelling languages. This is due to the recruiting of the participants from the Modelica user groups. However, from an industry standpoint, this is also the group of people where such results are useful.

While the individual timings of the experiment will surely be dependent on the background of the participants, the study design should compensate any such effects for the derived results. Even if a mechanical engineer will be faster at recognizing a mechanical spring-damper system compared to a computer scientist, this will be true for any of the different representations, which are compared relatively to each other.

Results hold mainly for the western culture. Although a few asian participants took part in the experiment, there is no reasonable evaluation as to what differences exist between these cultural groups. However, no such differences would be expected anyway.

For reasons of practicality, the used models were quite small. Due to the lack of similar studies, not much is known about the effects of model size and complexity. It is completely plausible that the best representation is dependent on the size and complexity of the system. Further studies are necessary to make generalizations in that direction.

Many other typical threats to external validity should not be relevant for this study. While participants might show increased performance due to the awareness of being observed (Hawthorne effect), this increase will be in effect for all of the tasks. Since only relative statements are derived from the resulting

data, the Hawthorne effect should not decrease the external validity of this study. Similar arguments can be made for pre- and post-test effects, situational specifics, or Rosenthal effects (higher expectations lead to increased performance). Second order effects can not be ruled out, but their influence is expected to be small.

6.6 Acknowledgments

The content of this chapter was developed in collaboration by Alexander Pollok and Andreas Klöckner. Andreas Klöckner performed the statistical analyses for the experiments of sections 6.4 and 6.5. Alexander Pollok performed the literature review, designed and conducted the expert interviews, designed the experiment of section 6.4 and organised the implementation for all experiments. Design for the experiment of section 6.5 was done in collaboration. We thank the Modelica User Groups Saxony, North America, Japan, Hamburg and Baden-Württemberg for their help with the recruiting of participants. We thank all participants of the experiments for their support and time.

CHAPTER 7

Conclusions

The development, modelling and control of aircraft energy systems was developed further on the following aspects.

A dynamic simulation model including the engine bleed air system (EBAS), the air conditioning pack, the ducting system, the cabin dynamics and the recirculation system was created in the equation-based, object-oriented modelling language (EOOML) *Modelica*. It was found that *Helmholtz* resonance effects do not contribute to the occurrence of limit cycle oscillation (LCO).

EBAS were modelled in detail, predicting LCO in aircraft environmental control system (ECS). A control strategy was developed which is able to reduce those oscillations significantly.

A new architecture for cabin temperature control was developed, enabling an unlimited number of temperature control zones. This is in contrast to conventional architectures, which were limited to a small number of zones.

For simulation studies at early stages of a development cycle, a class of virtual controllers was identified that shows good tracking performance without any tuning effort.

Several experiments were performed to make dependable statements about the usability of EOOML.

APPENDIX A

Appendix

A.1 Modelica code of simple controller

Listing A.1: *Modelica Code of SimpleController*

```
model SC
  Modelica.Blocks.Interfaces.RealOutput y annotation (...);
  parameter Boolean use_actual_in = true
    annotation(Dialog(group="Input"),choices(checkBox=true));
  input Real actual = 0 "expression for controlled variable"
    annotation(Dialog(enable = not use_actual_in,
    group="Input"));
  parameter Boolean use_target_in = true
    annotation(Dialog(group="Input"),choices(checkBox=true));
  input Real target = 0 "expression for setpoint"
    annotation(Dialog(enable = not use_target_in,
    group="Input"));

  parameter Boolean use_fixed_limits = true
    "check if controller output range is constant"
    annotation(Dialog(group="Behavior"),choices(checkBox=true));
  input Real output_max = 1 "maximum output of controller"
    annotation(Dialog(group="Behavior",
    enable = use_fixed_limits));
  input Real output_min = -output_max
    "minimum output of controller" annotation(Dialog(
```

```
    group="Behavior", enable = use_fixed_limits));
  parameter Real unitsize = 1
    "order of magnititude of system output
    (1e5 for pressure, 1 for mass flow, etc)"
    annotation(Dialog(group="Behavior"));
  parameter Integer dynamics = 0
    "selected type of dynamical behavior (0 for bang-bang,
    1 for first-order, 2 for second-order"
    annotation(Dialog(group="Behavior"));
  parameter Modelica.SIunits.Time timeconstant = 1
    "selected timeconstant" annotation(Dialog(group="Behavior",
    enable = (dynamics > 0)));
  parameter Modelica.SIunits.Damping damping = 0
    "selected damping in case of second order behavior"
    annotation(Dialog(group="Behavior",
    enable = (dynamics == 2)));

  parameter Boolean input_smooth = false
    "is the input guaranteed to be smooth?"
    annotation(Dialog(group="Advanced"));
  parameter Integer n_order = if dynamics == 2 then 3 else 2
    "dimension of sliding surface"
    annotation(Dialog(group="Advanced"));
  parameter Real[:] weights = if dynamics == 0 then {-1, 0}
    elseif dynamics == 1 then {-1, -timeconstant} else
    {-1,-2*damping*timeconstant/(2*Modelica.Constants.pi),-
    (timeconstant/(2*Modelica.Constants.pi))^2}
    "definition of sliding surface"
    annotation(Dialog(group="Advanced"));
  parameter Real boundarylayer = 0.001
    "relative width of boundary layer"
    annotation(Dialog(group="Advanced"));
  parameter Real timescale_der = if input_smooth then 0 else
    timeconstant/100
    "timescale of approximate derivation of input
    (0 for exact derivative)"
    annotation(Dialog(group="Advanced"));
  parameter Boolean exact_sliding_mode = false
    "boundary layer SMC if false, normal SMC if true
    (only for explicit solvers)"
    annotation(Dialog(group="Advanced"));

  Real authority =
    (output_max_in_internal-output_min_in_internal)/2
    "range of controller output";
  Real mean =
    (output_max_in_internal+output_min_in_internal)/2
    "mean of range";
  Real x[n_order];
```

```
  model TD1
    input Real u;
    output Real y;
    Real x;
    parameter Real timescale_der;
  equation
    if timescale_der == 0 then
      der(u) = y;
      x = 0;
    else
      der(x)*timescale_der = u - x;
      der(x) = y;
    end if;
  end TD1;

protected
TD1 td1[n_order-1](each timescale_der = timescale_der);
Modelica.Blocks.Interfaces.RealInput actual_in_internal
    "Needed to connect to conditional connector";
Modelica.Blocks.Interfaces.RealInput target_in_internal
    "Needed to connect to conditional connector";
Modelica.Blocks.Interfaces.RealInput output_max_in_internal
    "Needed to connect to conditional connector";
Modelica.Blocks.Interfaces.RealInput output_min_in_internal
    "Needed to connect to conditional connector";

public
Modelica.Blocks.Interfaces.RealInput actual_in
  if use_actual_in  annotation (...);
Modelica.Blocks.Interfaces.RealInput target_in
  if use_target_in  annotation (...);
Modelica.Blocks.Interfaces.RealInput output_max_in
  if not use_fixed_limits  annotation (...);
Modelica.Blocks.Interfaces.RealInput output_min_in
  if not use_fixed_limits  annotation (...);

equation
  connect(actual_in, actual_in_internal);
  connect(target_in, target_in_internal);
  connect(output_min_in, output_min_in_internal);
  connect(output_max_in, output_max_in_internal);

  if not use_actual_in then
    actual_in_internal = actual;
  end if;
  if not use_target_in then
    target_in_internal = target;
  end if;
```

```
  if use_fixed_limits then
    output_min_in_internal = output_min;
    output_max_in_internal = output_max;
  end if;

  x[1] = actual_in_internal - target_in_internal;

  for i in 2:n_order loop
   x[i] = td1[i-1].y;
   x[i-1] = td1[i-1].u;
  end for;

  if exact_sliding_mode then
    y = if x*weights > 0 then output_max else output_min;
  else
    y = mean + authority*Modelica.Math.tanh(
         x*weights/(boundarylayer*unitsize));
  end if;

  annotation (...);
end SC;
```

Bibliography

[1] Airbus SAS. Airbus global market forecast 2016-2035. `http://www.airbus.com/company/market/global-market-forecast-2016-2035/`, 2016. accessed March 2017.

[2] The Boeing Company. Current market outlook 2016-2035. `http://www.boeing.com/resources/boeingdotcom/commercial/about-our-market/assets/downloads/cmo_print_2016_final_updated.pdf`, 2016. accessed March 2017.

[3] Daniel Rutherford and Mazyar Zeinali. Efficiency trends for new commercial jet aircraft 1960 to 2008. 2009.

[4] JA Rosero, JA Ortega, E Aldabas, and LARL Romeral. Moving towards a more electric aircraft. *IEEE Aerospace and Electronic Systems Magazine*, 22(3):3–9, 2007.

[5] Michael Sielemann. *Device-Oriented Modeling and Simulation in Aircraft Energy Systems Design*. PhD thesis, Hamburg University of Technology, 2012.

[6] Hilding Elmqvist, Toivo Henningsson, and Martin Otter. Systems modeling and programming in a unified environment based on julia. In *International Symposium on Leveraging Applications of Formal Methods*, pages 198–217. Springer, 2016.

[7] Alexander Pollok and Daniel Bender. Using multi-objective optimization to balance system-level model complexity. In *Proceedings of the 6th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, pages 69–78. ACM, 2014.

[8] Alexander Pollok, Dirk Zimmer, and Francesco Casella. Fractional-order modelling in Modelica. In *Proceedings of the 11th International Modelica Conference*, 2015.

[9] Federico Mothes, Andreas Klöckner, Jane Jean Kiam, Martin Köhler, Alexander Pollok, Alexander Knoll, and Axel Schulte. Autonomes missionsmanagement für unbemannte solarbetriebene flugzeuge mit extrem langer flugdauer. In *Deutscher Luft- und Raumfahrtkongress*, 2016.

[10] Hilding Elmqvist, Hans Olsson, Axel Goteman, Vilhelm Roxling, Dirk Zimmer, and Alexander Pollok. Automatic GPU code generation of Modelica functions. In *Proceedings of the 11th International Modelica Conference*, 2015.

[11] Daniel Schlabe, Dirk Zimmer, and Alexander Pollok. Exploitation strategies of cabin and galley thermal dynamics. In *SAE Technical Paper*. SAE International, 2017.

[12] Alexander Pollok and Francesco Casella. High-fidelity modelling of self-regulating pneumatic valves. In *Proceedings of the 11th International Modelica Conference*, 2015.

[13] Alexander Pollok and Andreas Klöckner. The use of ockham's razor in object-oriented modeling. In *Proceedings of the 7th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, pages 31–38. ACM, 2016.

[14] Alexander Pollok and Francesco Casella. Modelling and simulation of self-regulating pneumatic valves. *Mathematical and Computer Modelling of Dynamical Systems*, 2017.

[15] Alexander Pollok and Andreas Schröffer. Simulation of helmholtz resonance effects in aircraft ecs. In *6th International Workshop on Aircraft System Technologies*, 2017.

[16] Alexander Pollok and Francesco Casella. Comparison of control strategies for aircraft bleed-air systems. In *Proceedings of the 20th IFAC World Congress*, 2017.

[17] Alexander Pollok, Daniel Bender, Ines Kerling, and Dirk Zimmer. Rapid development of an aircraft cabin temperature regulation concept. In *Proceedings of the 12th International Modelica Conference*, 2017.

[18] Alexander Pollok. Control strategies for an advanced aircraft-cabin temperature system. In *1st IEEE Conference on Control Technology and Applications*, 2017.

[19] Alexander Pollok. Advanced temperature control in aircraft cabins - a digital prototype. In *SAE Technical Paper*. SAE International, 2017.

[20] Alexander Pollok, A. Klöckner, and D. Zimmer. Representations of equation-based models are not created equal. In *Proceedings of the 8th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*. ACM, 2017 (accepted).

[21] Dirk Zimmer. Equation-based modeling with modelica–principles and future challenges. *SNE*, page 67, 2016.

[22] Federal Aviation Administration. Federal aviation regulations (far), §25.831 'ventilation', last amended 1997.

[23] Jose VC Vargas and Adrian Bejan. Integrative thermodynamic optimization of the environmental control system of an aircraft. *International journal of heat and mass transfer*, 44(20):3907–3917, 2001.

[24] Juan Carlos Ordonez and Adrian Bejan. Minimum power requirement for environmental control of aircraft. *Energy*, 28(12):1183–1202, 2003.

[25] Philip Jordan and Gerhard Schmitz. A modelica library for scalable modelling of aircraft environmental control systems. In *Proceedings of the 10th International Modelica Conference*, 2014.

[26] Hermann von Helmholtz. On the sensations of tone. *Trans. AJ ELLIS, New York: Longmans, Green*, 1863.

[27] Francesco Casella, Martin Otter, Katrin Proelss, Christoph Richter, and Hubertus Tummescheit. The Modelica Fluid and Media library for modeling of incompressible and compressible thermo-fluid pipe networks. In *Proceedings of the 5th International Modelica Conference*, pages 631–640, 2006.

[28] Robert Hoover, Morris Anderson, Robert Romano, Jeff Mendoza, Dan Judd, and Don Weir. Flush inlet scoop design for aircraft bleed air system, September 27 2011. US Patent 8,024,935.

[29] Tobias Bellmann, Johann Heindl, Matthias Hellerer, Richard Kuchar, Karan Sharma, and Gerd Hirzinger. The dlr robot motion simulator part i: Design and setup. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4694–4701. IEEE, 2011.

[30] Norbert Kroll and Cord-Christian Rossow. Digital-x: Dlr's way towards the virtual aircraft. 2012.

[31] Christian Schallert. Ein integriertes entwurfswerkzeug für elektrische bordsysteme. In *Deutscher Luft-und Raumfahrtkongress*, 2010.

[32] MAA Shoukat Choudhury, Sirish L Shah, Nina F Thornhill, and David S Shook. Automatic detection and quantification of stiction in control valves. *Control Engineering Practice*, 14(12):1395–1412, 2006.

[33] WL Bialowski. Dreams vs. reality: a view from both sides of the gap. *Pulp and Paper Canada*, 94:19–27, 1993.

[34] Lane Desborough and Randy Miller. Increasing customer value of industrial control performance monitoring-honeywell's experience. In *AIChE symposium series*, pages 169–189. American Institute of Chemical Engineers, 2002.

[35] Peter Beater. Modeling and digital simulation of hydraulic systems in design and engineering education using modelica and hylib. In *Proceedings of the Modelica Workshop*, pages 23–24, 2000.

[36] Peter Beater and Christoph Clauß. Multidomain systems: Pneumatic, electronic and mechanical subsystems of a pneumatic drive modelled with modelica. In *Proceedings of the 3rd International Modelica Conference*, 2003.

[37] Sven-Erik Pohl and Markus Gräf. Dynamic simulation of a free-piston linear alternator in Modelica. In *Proceedings of the 4th International Modelica Conference*, 2005.

[38] Aron Pujana-Arrese, Javier Arenas, Iban Retolaza, Ana Martinez-Esnaola, and Joseba Landaluze. Modelling in Modelica of a pneumatic muscle: application to model an experimental set-up. In *21st European conference on modelling and simulation (ECMS)*, pages 4–6, 2007.

[39] Jürgen Köhler and Gerhard Schmitz. Untersuchungen zu thermostatischen expansionsventilen. 2009.

[40] Lan Shang and Guangjun Liu. Optimal control of a bleed air temperature regulation system. In *International Conference on Mechatronics and Automation*, pages 2610–2615. IEEE, 2007.

[41] Peter Hodal and Guangjun Liu. Bleed air temperature regulation system: modeling, control, and simulation. In *Proceedings of the 2005 IEEE Conference on Control Applications*, pages 1003–1008. IEEE, 2005.

[42] John R Cooper, Chengyu Cao, and Jiong Tang. Control of a nonlinear pressure-regulating engine bleed valve in aircraft air management systems. In *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2013.

[43] R.F. Stokes, J.D. Timm, S.R. LaCroix, and M.R. Adams. Compressor bleed air control apparatus and method, April 26 1983. US Patent 4,380,893.

[44] Michael Sielemann. High-speed compressible flow and gas dynamics. In *Proceedings of the 9th International Modelica Conference*, 2012.

[45] Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

[46] C Solliec and F Danbon. Aerodynamic torque acting on a butterfly valve. comparison and choice of a torque coefficient. *Journal of fluids engineering*, 121(4):914–917, 1999.

[47] C Canudas De Wit, Hans Olsson, Karl Johan Astrom, and Pablo Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, 1995.

[48] Martin Aberger and Martin Otter. Modeling friction in modelica with the lund-grenoble friction model. In *Proceedings of the 2nd International Modelica Conference*, 2002.

[49] Martin Otter, Hilding Elmqvist, and Sven Erik Mattsson. The new Modelica Multibody library. In *Proceedings of the 3rd International Modelica Conference*. Citeseer, 2003.

[50] Michael Tiller. *Introduction to physical modeling with Modelica*. Springer Science & Business Media, 2001.

[51] François E Cellier and Ernesto Kofman. *Continuous system simulation*. Springer Science & Business Media, 2006.

[52] Ashish Singhal and Timothy I Salsbury. A simple method for detecting valve stiction in oscillating control loops. *Journal of Process Control*, 15(4):371–382, 2005.

[53] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*, volume 2. Wiley New York, 2007.

[54] Ranganathan Srinivasan and Raghunathan Rengaswamy. Approaches for efficient stiction compensation in process control valves. *Computers & Chemical Engineering*, 32(1):218–229, 2008.

[55] Tore Hägglund. A friction compensator for pneumatic control valves. *Journal of process control*, 12(8):897–904, 2002.

[56] John Gerry and Michel Ruel. How to measure and combat valve stiction online. In *ISA International Fall Conference*, 2001.

[57] Robert Hooke and Terry A Jeeves. "direct search"solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961.

[58] Andreas Klöckner, Franciscus LJ van der Linden, and Dirk Zimmer. Noise generation for continuous system simulation. In *Proceedings of the 10th International Modelica Conference*, number 96, pages 837–846. Linköping University Electronic Press, 2014.

[59] JG Ziegler and NB Nichols. Process lags in automatic control circuits. *Trans. ASME*, 65(5):433–443, 1943.

[60] Ashley Hall, T Mayer, Ingo Wuggetzer, and PRN Childs. Future aircraft cabins and design thinking: optimisation vs. win-win scenarios. *Propulsion and Power Research*, 2(2):85–95, 2013.

[61] R Schliwa. Future trends and design thinking in cabin design. In *International Workshop on Aircraft System Technologies*, 2017.

[62] NP Gao and JL Niu. Personalized ventilation for commercial aircraft cabins. *Journal of aircraft*, 45(2):508–512, 2008.

[63] P Jacobs and WF De Gids. Individual and collective climate control in aircraft cabins. *International journal of vehicle design*, 42(1-2):57–66, 2006.

[64] Tengfei Tim Zhang, Penghui Li, and Shugang Wang. A personal air distribution system with air terminals embedded in chair armrests on commercial airplanes. *Building and Environment*, 47:89–99, 2012.

[65] Modelica-Association. The Modelica Standard Library. *Online, URL: http://www.modelica.org/libraries/Modelica*, 2008.

[66] Daniel Bender. Exergy-based analysis of aircraft environmental control systems - integration into model- based design and potential for aircraft system evaluation. In *29th International Conference on Efficiency, Cost, Optimisation, Simulation and Environmental Impact of Energy Systems*, 2016.

[67] Michael Sielemann, T Giese, B Öhler, and Martin Otter. A flexible toolkit for the design of environmental control system architectures. In *Proceedings of the First CEAS European Air and Space Conference*, 2007.

[68] Nicolas Minorsky. Directional stability of automatically steered bodies. *Naval Engineers Journal*, 32(2), 1922.

[69] Andreas Pfeiffer. Optimization library for interactive multi-criteria optimization tasks. In *Proceedings of the 9th International Modelcia Conference*, number 76, pages 669–680. Linköping University Electronic Press; Linköpings universitet, 2012.

[70] Marcus Baur, Martin Otter, and Bernhard Thiele. Modelica libraries for linear control systems. In *Proceedings of 7th International Modelica Conference*, pages 20–22, 2009.

[71] Hans-Dieter Joos. Multi-objective parameter synthesis (mops). In *Robust Flight Control*, pages 199–217. Springer, 1997.

[72] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[73] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147, 1998.

[74] Michael Safonov and Michael Athans. Gain and phase margin for multiloop lqg regulators. *IEEE Transactions on Automatic Control*, 22(2):173–179, 1977.

[75] John Doyle. Guaranteed margins for lqg regulators. *IEEE Transactions on Automatic Control*, 23(4):756–757, 1978.

[76] Marco Bonvini and Alberto Leva. Object-oriented sub-zonal modelling for efficient energy-related building simulation. *Mathematical and Computer Modelling of Dynamical Systems*, 17(6):543–559, 2011.

[77] Blas M Vinagre, Concepción A Monje, Antonio J Calderón, and José I Suárez. Fractional pid controllers for industry application. a brief introduction. *Journal of Vibration and Control*, 13(9-10):1419–1429, 2007.

[78] Michael Thümmel, Gertjan Looye, Matthias Kurze, Martin Otter, and Johann Bals. Nonlinear inverse models for control. In *Proceedings of the 4th International Modelica Conference*, pages 267–279, 2005.

[79] Paul Acquatella, Wouter Falkena, Erik-Jan van Kampen, and Q Ping Chu. Robust nonlinear spacecraft attitude control using incremental nonlinear dynamic inversion. In *AIAA Guidance, Navigation, and Control Conference*, page 4623, 2012.

[80] Rudolf Emil Kalman et al. Contributions to the theory of optimal control. *Bol. Soc. Mat. Mexicana*, 5(2):102–119, 1960.

[81] Keith Glover and John C Doyle. State-space formulae for all stabilizing controllers that satisfy an h-infinity-norm bound and relations to relations to risk sensitivity. *Systems and Control Letters*, 11(3):167–172, 1988.

[82] John C Doyle, Keith Glover, Pramod P Khargonekar, and Bruce A Francis. State-space solutions to standard h-2 and h-infinity control problems. *IEEE Transactions on Automatic control*, 34(8):831–847, 1989.

[83] Christopher Edwards and Sarah Spurgeon. *Sliding mode control: theory and applications*. Crc Press, 1998.

[84] Vadim Utkin, Jürgen Guldner, and Jingxin Shi. *Sliding mode control in electro-mechanical systems*, volume 34. CRC press, 2009.

[85] G Bartolini, A Ferrara, and E Usai. Chattering avoidance by second-order sliding mode control. *IEEE Transactions on Automatic control*, 43(2):241–246, 1998.

[86] Marco C Campi, Andrea Lecchini, and Sergio M Savaresi. Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, 38(8):1337–1346, 2002.

[87] Marco C Campi and Sergio M Savaresi. Direct nonlinear control design: The virtual reference feedback tuning (vrft) approach. *IEEE Transactions on Automatic Control*, 51(1):14–27, 2006.

[88] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[89] Karl J Åström and Björn Wittenmark. *Adaptive control*. Courier Corporation, 2013.

[90] Ali Zilouchian and Mohammad Jamshidi. *Intelligent control systems using soft computing methodologies*. CRC Press, Inc., 2000.

[91] Linda R Petzold et al. A description of dassl: A differential/algebraic system solver. *Scientific computing*, 1:65–68, 1982.

[92] K David Young, Vadim I Utkin, and Umit Ozguner. A control engineer's guide to sliding mode control. In *Proceedings of the IEEE International Workshop on Variable Structure Systems*, pages 1–14. IEEE, 1996.

[93] CA Hoare. Hints on programming language design. Technical report, DTIC Document, 1973.

[94] Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1983.

[95] Françoise Détienne and Frank Bott. *Software Design–Cognitive Aspect*. Springer Science & Business Media, 2002.

[96] Jon A Turner and Robert A Karasek Jr. Software ergonomics: effects of computer application design parameters on operator task performance and health. *Ergonomics*, 27(6):663–690, 1984.

[97] Thomas R. G. Green and Marian Petre. Usability analysis of visual programming environments: a cognitive dimensions framework. *Journal of Visual Languages and Computing*, 7(2):131–174, 1996.

[98] Steven Clarke. Evaluating a new programming language. In *13th Workshop of the Psychology of Programming Interest Group*, pages 275–289, 2001.

[99] James Howatt. A project-based approach to programming language evaluation. *ACM SIGPLAN Notices*, 30(7):37–40, 1995.

[100] Jarallah Al Ghamdi and Joseph Urban. Comparing and assessing programming languages: basis for a qualitative methodology. In *Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing: states of the art and practice*, pages 222–229. ACM, 1993.

[101] Daniel L Moody. The physics of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, 2009.

[102] Keng Siau and Matti Rossi. Evaluation of information modeling methods-a review. In *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, volume 5, pages 314–322. IEEE, 1998.

[103] Eva-Lena Lengquist Sandelin, Susanna Monemar, Peter Fritzson, and Peter Bunus. An interactive tutoring environment for modelica. In *Proceedings of the 3rd International Modelica Conference*, 2003.

[104] Jill H Larkin and Herbert A Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive science*, 11(1):65–100, 1987.

[105] TRG Green, ME Sime, and MJ Fitter. The art of notation. *Computing skills and the user interface*, pages 221–251, 1981.

[106] Dirk Zimmer, Martin Otter, Hilding Elmqvist, and Gerd Kurzbach. Custom annotations: Handling meta-information in Modelica. In *Proceedings of 10th International Modelica Conference*, volume 10, 2014.

[107] Thomas RG Green and R Navarro. Programming plans, imagery, and visual programming. In *Human Computer Interaction*, pages 139–144. Springer, 1995.

[108] Iris Vessey and Dennis Galletta. Cognitive fit: An empirical study of information acquisition. *Information systems research*, 2(1):63–84, 1991.

[109] James C Spohrer and Elliot Soloway. Novice mistakes: Are the folk wisdoms correct? *Communications of the ACM*, 29(7):624–632, 1986.

[110] Essi Lahtinen, Kirsti Ala-Mutka, and Hannu-Matti Järvinen. A study of the difficulties of novice programmers. In *ACM SIGCSE Bulletin*, volume 37, pages 14–18. ACM, 2005.

[111] David J Gilmore. Expert programming knowledge: a strategic approach. *Psychology of programming*, pages 223–234, 1990.

[112] Thomas D LaToza, David Garlan, James D Herbsleb, and Brad A Myers. Program comprehension as fact finding. In *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, pages 361–370. ACM, 2007.

[113] Clayton Lewis and Gary Olson. Can principles of cognition lower the barriers to programming? In *Empirical studies of programmers: second workshop*, pages 248–263. Ablex Publishing Corp., 1987.

[114] John Pane and Brad Myers. Usability issues in the design of novice programming systems. *Carnegie Mellon University, School of Computer Science Technical Report*, 1996.

[115] Bonnie A Nardi. *A small matter of programming: perspectives on end user computing*. MIT press, 1993.

[116] Tobias Roehm, Rebecca Tiarks, Rainer Koschke, and Walid Maalej. How do professional developers comprehend software? In *Proceedings of the 34th International Conference on Software Engineering*, pages 255–265. IEEE Press, 2012.

[117] Saeed Dehnadi and Richard Bornat. The camel has two humps (working title). *Middlesex University, UK*, pages 1–21, 2006.

[118] Peter Chalk, Tom Boyle, Poppy Pickard, Claire Bradley, Ray Jones, and Ken Fisher. Improving pass rates in introductory programming. In *4th Annual Conference of the LTSN Centre for the Information and Computer Sciences*, 2003.

[119] Sally Fincher, Anthony Robins, Bob Baker, Ilona Box, Quintin Cutts, Michael de Raadt, Patricia Haden, John Hamer, Margaret Hamilton, Raymond Lister, et al. Predictors of success in a first programming course. In *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*, pages 189–196. Australian Computer Society, Inc., 2006.

[120] Michael Tiller. Parsing and semantic analysis of modelica code for non-simulation applications. In *Proceedings of the 3rd International Modelica Conference*, 2003.

[121] EJ Williams. Experimental designs balanced for the estimation of residual effects of treatments. *Australian Journal of Chemistry*, 2(2):149–168, 1949.

[122] Dominik J Leiner. Sosci survey (version 2.5.00-i)[computer software]. 2014.

[123] James V Bradley. Complete counterbalancing of immediate sequential effects in a latin square design. *Journal of the American Statistical Association*, 53(282):525–528, 1958.

[124] Claude M Steele. A threat in the air: How stereotypes shape intellectual identity and performance. *American psychologist*, 52(6):613, 1997.

[125] Ap Dijksterhuis and Ad Van Knippenberg. The relation between perception and behavior, or how to win a game of trivial pursuit. *Journal of personality and social psychology*, 74(4):865, 1998.

[126] Margaret Shih, Todd L Pittinsky, and Nalini Ambady. Stereotype susceptibility: Identity salience and shifts in quantitative performance. *Psychological science*, 10(1):80–83, 1999.

[127] Steven J Spencer, Claude M Steele, and Diane M Quinn. Stereotype threat and women's math performance. *Journal of experimental social psychology*, 35(1):4–28, 1999.

[128] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

Publications

Journal papers

Alexander Pollok and Francesco Casella (2017). Modelling and simulation of self-regulating pneumatic valves. *Mathematical and Computer Modelling of Dynamical Systems*.
DOI: 10.1080/13873954.2017.1298623

Conference papers

Alexander Pollok, Andreas Klöckner and Dirk Zimmer (2017). Representations of equation-based models are not created equal. In: *Proceedings of the 8th International Workshop on Equation-Based Languages and Tools* (full paper review).
DOI: 10.1145/3158191.3158200

Alexander Pollok (2017). Advanced temperature control in aircraft cabins - a digital prototype. In: *SAE Technical Paper*. SAE International (abstract review + full paper review).
URL: papers.sae.org/2017-01-2161

Daniel Schlabe, Dirk Zimmer and Alexander Pollok (2017). Exploitation strategies of cabin and galley thermal dynamics. In: *SAE Technical Paper*. SAE International (abstract review + full paper review).
URL: papers.sae.org/2017-01-2037

Alexander Pollok (2017). Control Strategies for an Advanced Aircraft-Cabin Temperature-System. In: *1st IEEE Conference on Control Technology and Applications* (full paper review).
URL: s.dlr.de/l3y1

Alexander Pollok and Francesco Casella (2017). Comparison of control strategies for aircraft bleed-air systems. In: *Proceedings of the 20th IFAC World Congress* (full paper review).
URL: elib.dlr.de/113729

Alexander Pollok, Daniel Bender, Ines Kerling and Dirk Zimmer (2017). Rapid development of an aircraft cabin temperature regulation concept. In: *Proceedings of the 12th International Modelica Conference* (full paper review).
URL: s.dlr.de/5084

Alexander Pollok and Andreas Schröffer (2017). Simulation of Helmholtz resonance effects in aircraft ECS. In: *6th International Workshop on Aircraft System Technologies* (extended abstract review).
URL: s.dlr.de/32gk

Alexander Pollok and Andreas Klöckner (2016). The use of Ockham's Razor in object-oriented modelling. In: *Proceedings of the 7th International Workshop on Equation-Based Languages and Tools* (full paper review).
DOI: 10.1145/2904081.2904086

Federico Mothes, Andreas Klöckner, Jane Jean Kiam, Martin Köhler, Alexander Pollok, Alexander Knoll and Axel Schulte (2016). Autonomes Missionsmanagement für unbemannte solarbetriebene Flugzeuge mit extrem langer Flugdauer (autonomous mission management for unmanned solar aircraft with extremely long flight duration). In: *Deutscher Luft- und Raumfahrtkongress 2016* (extended abstract review).
URL: elib.dlr.de/110623

Alexander Pollok and Francesco Casella (2015). High-fidelity modelling of self-regulating pneumatic valves. In: *Proceedings of the 11th International Modelica Conference* (full paper review).
URL: s.dlr.de/x74l

Alexander Pollok, Dirk Zimmer and Francesco Casella (2015). Fractional-order modelling in Modelica. In: *Proceedings of the 11th International Modelica Conference* (full paper review).
URL: s.dlr.de/g88r

Hilding Elmqvist, Hans Olsson, Axel Goteman, Vilhelm Roxling, Dirk Zimmer and Alexander Pollok (2015). Automatic GPU code generation of Modelica functions. In: *Proceedings of the 11th International Modelica Conference* (full paper review).
URL: s.dlr.de/qqm8

Alexander Pollok and Daniel Bender (2014). Using multi-objective optimization to balance system-level model complexity. In: *Proceedings of the 6th International Workshop on Equation-Based Languages and Tools* (full paper review).
DOI: 10.1145/2666202.2666213