Rapid development of an aircraft cabin temperature regulation concept

Alexander Pollok[1,2] Daniel Bender[1] Ines Kerling[1] Dirk Zimmer[1]

[1]Institute of System Dynamics and Control, DLR German Aerospace Center, Wessling, Germany,
{alexander.pollok,daniel.bender,ines.kerling,dirk.zimmer}@dlr.de
[2]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy

Abstract

The air in aircraft cabins is controlled for pressure, temperature and humidity. The number of temperature zones is generally kept low, for reasons of necessary ducting space. We devise a new ducting concept, which enables a large number of temperature zones. Controllability of the system is however predicted to be a potential obstacle. For a quick resolution of this question, a Modelica model is created. Model creation is focused on a short development time as well as usefulness for controller synthesis. A workflow is presented that enables a quick iteration time between controller synthesis in Matlab and controller testing in a Modelica environment. Finally, the impact of this new concept on the energy consumption of the air generation unit is discussed.

Keywords: Modelica, energy, exergy, control, modelling

1 Introduction

In modern passenger aircraft, temperature control is realized using a small number of temperature zones. For instance, the Airbus A320 features two fixed-size temperature control zones for the cabin, plus one additional zone for the flight deck. A typical cabin temperature regulation system is illustrated in Figure 1. Fresh air is delivered by the air conditioning packs and ducted into the mixing chamber (M). There it is mixed with filtered and recirculated air from the cabin underfloor volume. It is split up into two mass flows. For each mass flow, very hot (around 200 °Celcius) trim air is added to increase the temperature to the desired value. The air is then ducted into the cabin volume. Displaced air is ducted into the underfloor, where some of it is recirculated, the remaining air is vented overboard.

Airlines like to customize their aircrafts with variations in travel classes, seat configurations, and availability of onboard entertainment systems. Generally, the demarcations of travel classes do not conform to the borders of cabin temperature zones. Imagine an expensive and therefore sparsely populated first class, followed by the business class, densely packed with business people producing hot air. This can lead to discrepancies with regard to the heat load per cabin length, which cannot be compensated by the control system, if they belong to the same temperature zone.

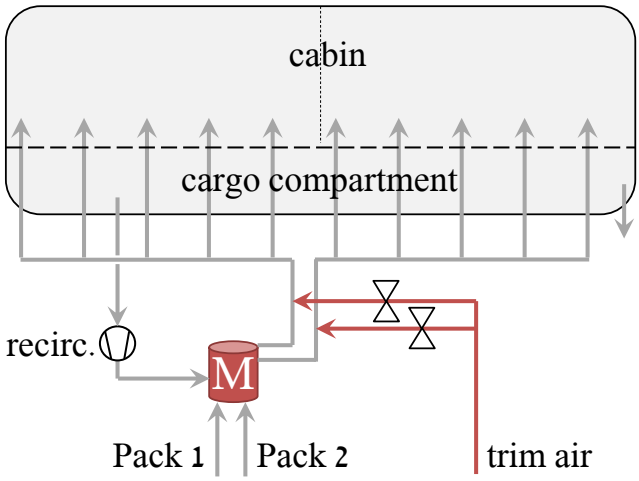

Figure 1. Conventional cabin temperature regulation system

Some ideas have been proposed to remedy this problem. In (Jacobs and De Gids, 2006) a concept is presented, where each passenger receives his own air outlet and individual temperature zone. However, the resulting size of the necessary ducting system within the crowded installation space of a modern aircraft is not considered. Similar concepts are mentioned in (Gao and Niu, 2008) or (Zhang et al., 2012), but the focus of the work is not on the control or feasibility side, but on air contamination reduction.

We propose an alternative cabin temperature regulation concept, which is illustrated in Figure 2. This concept is based around two main ducts, each one spanning the complete cabin length. One of them ducts air at a relatively cold temperature, the other one at a relatively hot temperature. At each cabin temperature zone, the air from both pipes is locally mixed using a small actuator, then ducted into the cabin.

This concept realises a variable number of temperature control zones. For a large number of zones, the amount of necessary ducting space and weight is lower than that of a conventional architecture, as a simple spreadsheet calculation for a typical single aisle aircraft shows, see Figure 3. Main reason for this is that only 2 pipes of cabin length L have to be fitted, instead of N pipes of average length $L/2$. System weight and volume even goes down

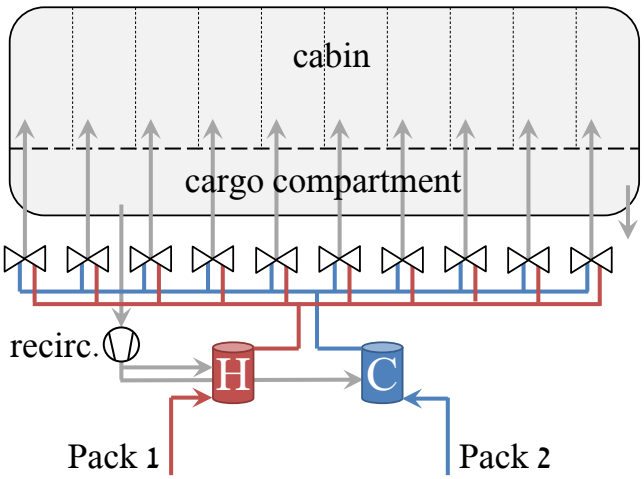

Figure 2. Proposed cabin temperature regulation concept

for a larger number of temperature zones, as less distribution ductwork between control valves and riser ducts is needed.

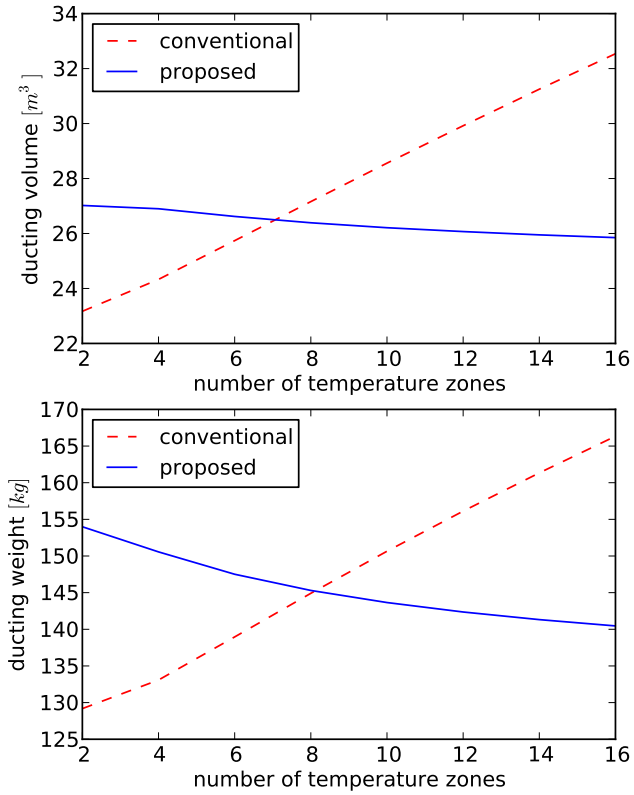

Figure 3. comparison of total ducting weight and ducting volume for conventional and proposed concept

This is bought at the expense of a potentially much more involved control system. Pneumatic and thermal interactions between temperature zones may be strong enough to prohibit the use of decentralized control. Also, the energy offtake of such a concept compared to a conventional architecture is unclear.

Given the high cost of testing facilities, a model-based approach is needed for an early design evaluation. The corresponding modeling environment must thus be able to cover all relevant aspects of the proposed concept. Beside the physical processes belonging to the pneumatic and thermal domain, this also contains the control design and of the temperature regulation system. Modelica is a well-established choice for the physical modelling (Sielemann, 2012; Schlabe and Zimmer, 2012) but provides also sufficient to represent and evaluate the control models (Baur et al., 2009; Bonvini and Leva, 2012). The control design can then be achived in interaction with Matlab.

The goal of this paper is to show how Modelica can be used for accelerated feasibility studies using the example of a new cabin temperature regulation concept. It is structured as follows: Section 2 presents the design requirements for a suitable model as well as the taken approach. Section 3 shows the controllability of the temperature regulation concept, based on the developed model. Section 4 treats energy considerations of the proposed architecture. Interesting points that came up during the development of the project are discussed in Section 5. Section 6 concludes the paper.

2 Modelling

A good simulation model is the basis for all subsequent development steps. The following requirements hold 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 behavior. Model inputs that are connected to saturating actuators shall be scaled symetrically 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 optimization like shown in (Pollok and Bender, 2014), but ultimately, some arbitrariness remains. We decided 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 asymetric ducting system. The complete model structure is shown in Figure 4 and presented in the following.

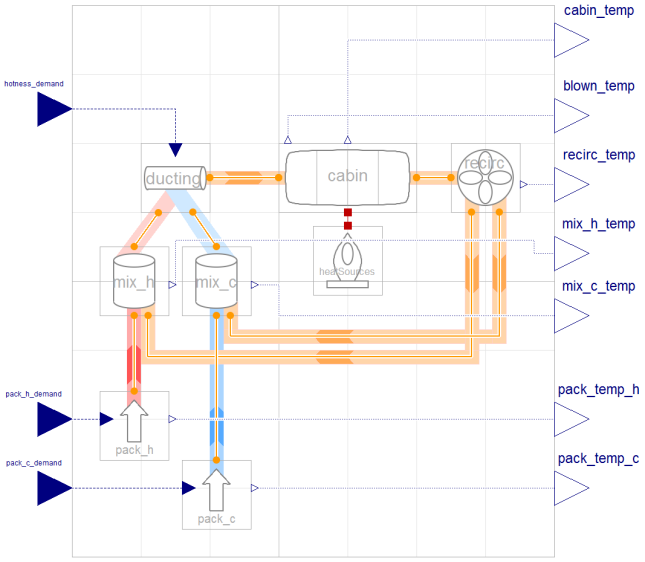

Figure 4. Top level view of temperature regulation model

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 (Pollok and Klöckner, 2016). The Modelica Standard Library (Modelica-Association, 2008) was used as much as possible to save additional modelling time.

All subsystems included an integer parameter n, denoting the number of discretized volume elements in the cabin. In this way, the scalability of the concept can be tested later without additional modelling effort.

Of those subsystems, ducting and cabin are especially interesting from a modelling perspective:

2.1 Ducting

As illustrated in Figure 2, air is ducted from the mixing chambers into the cabin via a network of ducts. This network is asymetrical and interactive with regard to the cabin temperature zones. If a large amount of cold air is needed for the center 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 vectorized flow elements together with customized connect-statements in a short amount of code and development time. The implementation is shown in Figure 5. Not shown are the parameters for the individual air resistance components. These are also dependent on the discretization parameter, since for example the length per pipe is not constant.

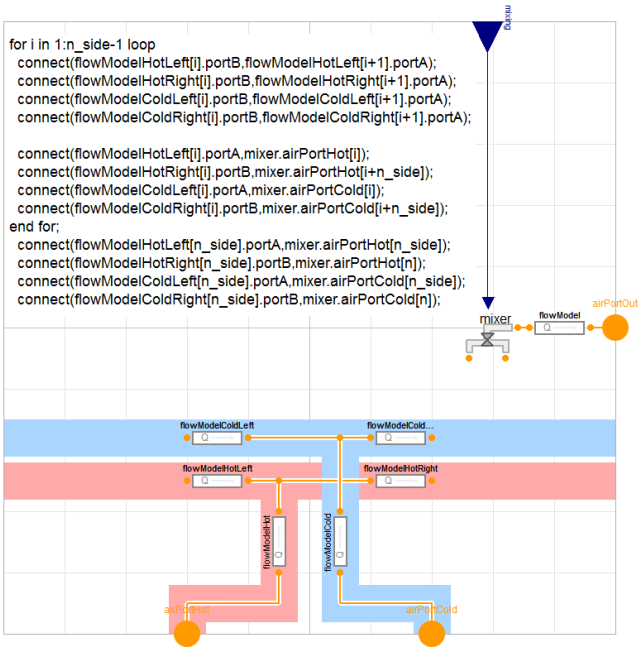

Figure 5. ducting subsystem model

2.2 Cabin

The flow configuration inside the aircraft cabin is complex and can only accurately be determined by experiments or CFD-calculations. However, for the evaluation of the presented concept, a low-order approximation suffices. The cabin is divided lengthwise into n volume elements. These elements are directly connected to enforce pressure equalization. Fluid volume elements can directly be connected in Modelica, at the cost of nonlinear systems of equations. 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 equalization mass flows occur, there is still some amount of thermal equalization caused by diffusion. This is modelled using thermal resistance elements, coupled between the volume elements. They were parameterized according to empirical experience.

Again, the subsystem was realized using a combination of vectorized elements and customized connect statements. The implementation is shown in Figure 6.

3 Control

A sufficient way to demonstrate controllability of a system is to find a stabilizing controller. For a demonstration of

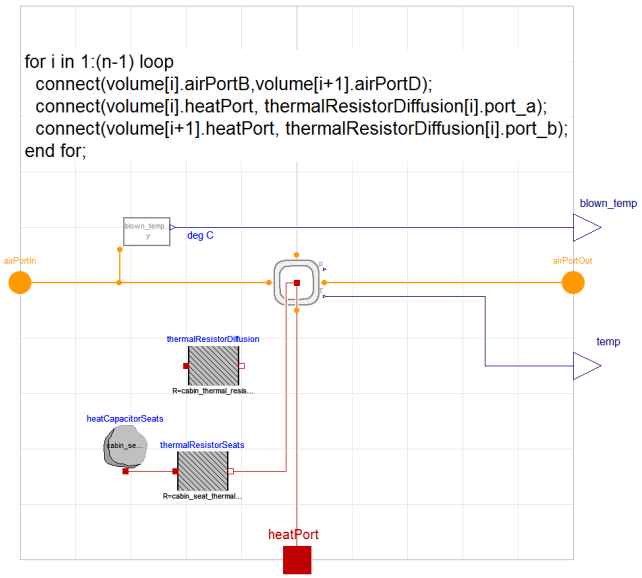

Figure 6. cabin subsystem model

robust performance, the response of the controlled system with regard to noise and nonlinearities can be shown.

We used linear quadratic Gaussian (LQG) control to find an optimal controller for the problem. The method is simple and often satisfactory in the development of multivariate, or MIMO-controllers for linear time-invariant (LTI)-systems. LQG controllers consist of a linear quadratic estimator (LQE, also known as Kalman filter) to estimate non-measured states, and a linear quadratic regulator (LQR), essentially an optimal state regulator. The regulator uses the estimated states to compute a control signal, the estimator uses the measured states as well as the control signal to estimate the states. This is illustrated in Figure 7. Both components can be designed independently, this will not compromise stability of the controlled system, but it can affect stability margins (for reference, see the very interesting abstract of Doyle (1972)), so robustness properties have to be verified after controller synthesis.

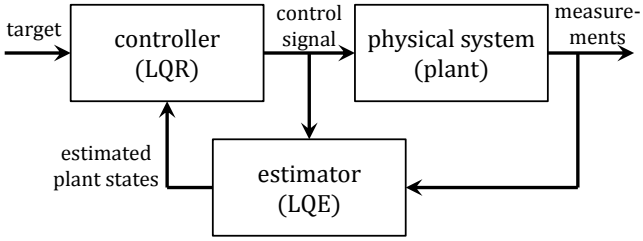

Figure 7. Structure of an LQG regulator

For LQG-synthesis, a linearized model is necessary. We used the linear systems library as presented in (Baur et al., 2009) as implemented in Dymola 2016 to linearize the model around the steady state. The model was instantiated with a pack temperature spread of 20Kelvins and 10 cabin temperature zones. Before linearization, the model was simulated for 1000s to come close to the steady state solution. Keep in mind that all model inputs are set to zero at linearization. Therefore, the valid range for all model inputs has to include zero and some buffer in both directions. This can be a problem for instance when a model input is connected to a valve opening with a valid range of 0 to 1. In this work, we scaled all such inputs to a valid range of -1 to +1. All other in- and outputs were scaled according to typical orders of magnitude. The linearized system was exported as a .mat-file using the writeMatrix-command.

Computation of the LQG controller was done in Matlab, based on the script as described in (Skogestad and Postlethwaite, 2007, p. 348). This formulation adds integrators to the plant outputs, ensuring zero steady state error for the controlled system. The model was reduced from 32 to 23 states, based on the results of the Hankel singular value decomposition as presented in Figure 8. Also, the 1-dof[1] variant was used, since setpoint changes are not a major concern in climate control systems.

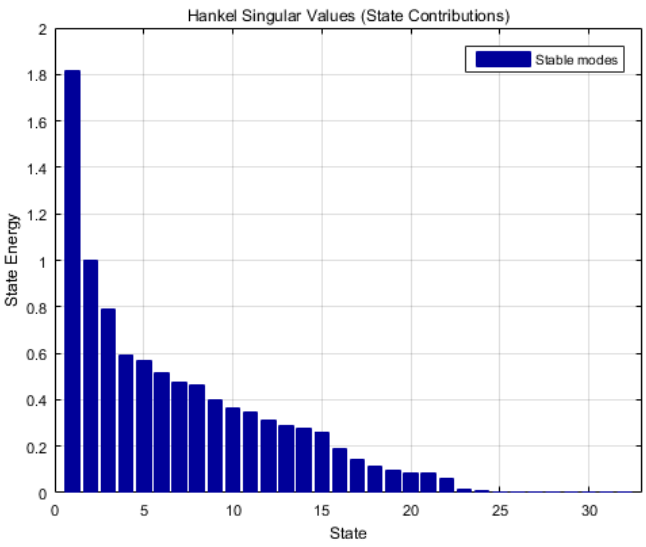

Figure 8. Hankel singular value decomposition of temperature control system

A Matlab-script was developed to automatically generate Modelica-code of the controller. This script is shown in Listings 1 and 2. In this way, a candidate controller can be tested in a Modelica environment in a few seconds.

Listing 2. Matlab code for the generation of Modelica controller code

```
function [ ] = fun_changeMatrixFormat( M )
% Changes the Matrix to a format like
% it is needed for an Matlab- Input
% f.e. M =  1 0
```

[1] In a 1-dof (one degree of freedom controller, setpoint changes and disturbances are handled equally. In a 2-dof controller, setpoint changes can be handled far more aggressively, as the setpoint is not part of the feedback-loop and therefore has no impact on system stability.

Listing 1. Matlab code for the generation of Modelica controller code

```
function [] = fun_dymola( sys )
%input: statespace object sys, representing combined estimator and controller
sys_x = size(sys.a,1);      % = m_A, n_A, m_B, n_C
sys_y = size(sys.d,1);      % = m_C, m_D
sys_u = size(sys.d,2);      % = n_B, n_D

% declarations
fprintf('\n\nmodel Controller "1-DOF LQG controller"\n\n');  %Name ""
fprintf('// import \n');
fprintf('import Modelica.Blocks; \n\n');
fprintf('// parameters \n'); %A,B,C,D...
fprintf('parameter Real A_controller[%.0f, %.0f] = ', sys_x, sys_x);
fun_changeMatrixFormat( sys.a );
fprintf('; \n');
fprintf('parameter Real B_controller[%.0f, %.0f] = ', sys_x, sys_u);
fun_changeMatrixFormat( sys.b );
fprintf('; \n');
fprintf('parameter Real C_controller[%.0f, %.0f] = ', sys_y, sys_x);
fun_changeMatrixFormat( sys.c );
fprintf('; \n');
fprintf('parameter Real D_controller[%.0f, %.0f] = ', sys_y, sys_u);
fun_changeMatrixFormat( sys.d );
fprintf('; \n');

% variables
fprintf('// variables \n');
% blocks
fprintf('Blocks.Continuous.StateSpace statespace_controller');
fprintf('(A = A_controller, B = B_controller, C = C_controller, D = D_controller) \n');
fprintf('annotation (Placement(transformation(extent={{36,-2},\n{56,18}})));\n');
fprintf('Blocks.Math.Feedback sum_controller [%0.i] \n', sys_u);
fprintf('annotation (Placement(transformation(extent={{-36,-2},\n{-16,18}})));\n')
% inputs/outputs
fprintf('Blocks.Interfaces.RealInput command[%0.f] "command signal" \n',sys_u);
fprintf('annotation (Placement(transformation(extent={{-120,40},{-80,80}})));\n');
fprintf('Blocks.Interfaces.RealInput feedbacksignal[%0.f] "sensor/feedback signal" \n',sys_u);
fprintf('annotation (Placement(transformation(extent={{-120,-78},{-80,-38}})));\n');
fprintf('Blocks.Interfaces.RealOutput outputsignal[%0.f] "driver/output signal" \n', sys_y);
fprintf('annotation (Placement(transformation(extent={{100,-10},{120,10}})));\n\n');

% equations
fprintf('// equations \n');
fprintf('equation \n');

% connecting the blocks and in/outputs
fprintf('for i in 1:%.0i loop\n', sys_y);
fprintf('connect(statespace_controller.y[i], outputsignal[i]) \n');
fprintf('annotation (Line(\npoints={{57,8},{110,8}},\ncolor={0,0,127}));\n');
fprintf('end for; \n\n');
fprintf('for i in 1:%.0i loop\n', sys_u);
fprintf('connect(statespace_controller.u[i], sum_controller[i].y) \n');
fprintf('annotation (Line(\npoints={{34,8},{-17,8}},\ncolor={0,0,127}));\n');
fprintf('connect(command[i], sum_controller[i].u1) \n');
fprintf('annotation (Line(\npoints={{-100,52},{-64,52},{-64,8},{-34,8}},\ncolor={0,0,127}));\n'
    );
fprintf('connect(feedbacksignal[i], sum_controller[i].u2) \n');
fprintf('annotation (Line(\npoints={{-100,-58},{-26,-58},{-26,0}},\ncolor={0,0,127}));\n');
fprintf('end for; \n\n');

% creating symbol for the block
fprintf('annotation (...)');

fprintf('end Controller; \n'); % Name;
```

```
%           0 1  to  [1, 0; 0, 1]
% for as much information as possible
format long;
mnbigness = size(M);
countm = 1;
countn = 1;
fprintf('[');
while (countm <= mnbigness(1))
    while(countn <= mnbigness(2))
        if (countn < mnbigness(2))
             fprintf('%f, ', M(countm, countn
                 ));
        else
            if (countm < mnbigness(1))
                fprintf('%f; ', M(countm,
                    countn));
            else
                fprintf('%f', M(countm,
                    countn));
            end
        end
     countn = countn+1;
    end
    countn = 1;
    countm = countm+1;
end
fprintf(']');
end
```

On the first try, equal disturbance and measurement noise was assumed, and a unity matrix was used for the input weight matrix R. The state weight matrix Q was calculated so that the projected plant output as well as the artificial integrator vector were also weighted with a unity matrix, using the Matlab code shown in Listing 3.

Listing 3. Matlab code for projection of the plant outputs to the state vector

```
R=weight_input*eye(n_u);
Q=blkdiag(weight_output.*transpose(C)*
eye(n_y)*C,weight_integrator*eye(n_y));
```

Since the actual system contains hard nonlinearities such as actuator saturation, compliance to those limits has to be tested using simulations of the controlled system. These simulations showed that the resulting controller outputs were exceeding the actuator limits. The variable weight-output was increased to 10.000, resulting in improved behavior[2]. Note that no actual optimization with regard to some performance criterion took place. The response of the controlled system to target temperature steps (from 20 to 21 °Celcius) on each temperature zone is shown in Figure 9. Overshoot[3] is generally low, but temperatures are still somewhat affected by temperature steps on neighboring zones. Rise time is at 68 to 102 seconds (M[4]: 85s, SD: 11.3s).

Robustness with regard to sensor noise was validated using the Noise library as presented by Klöckner et al. (2014). The qualitative behavior of the system remains unchanged. An illustration of the overall workflow can be seen in Figure 10.

[2]Using LQR/LQG, it is quite typical that small changes in controller behavior necessitate large changes in the weighting matrices.

[3]Overshoot describes the peak of the response to a step input, rise time describes the time it takes the output to increase from 0.1 to 0.9.

[4]M denotes the mean value, SD denotes the standard deviation.

4 Efficiency

The concept presented within this work requires a cold (C) and a hot (H) air reservoir (see Figure 2). Each of these reservoirs is supplied by a separate air conditioning pack. Conventional architectures duct the cooled air from the air packs to a common mixer unit (see Figure 1. 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 air pack would then run in different conditions compared to conventional in-service air packs. About 2-3% of the whole energy consumption of a conventional civil aircraft applies to the ECS (Bender, 2016). Thus an energy analysis of the deviating pack operation is necessary.

4.1 System Description

The key part of the ECS is the air generation unit (also called the 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 lead to the air pack where it is cooled down and dehumidified. It passes several heat exchangers, a compressor, a turbine and valves before the flow has reached the right condition to be lead 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 11 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 (PHX) the hot air is cooled down against the ram air flow. Before entering the compressor stage (CMP), a part of the air flow is separated and bypassed through the temperature control valve (TCV). Downstream the compressor stage the heated and compressed air is cooled down a second time inside the main heat exchanger (MHX) 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 down 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.

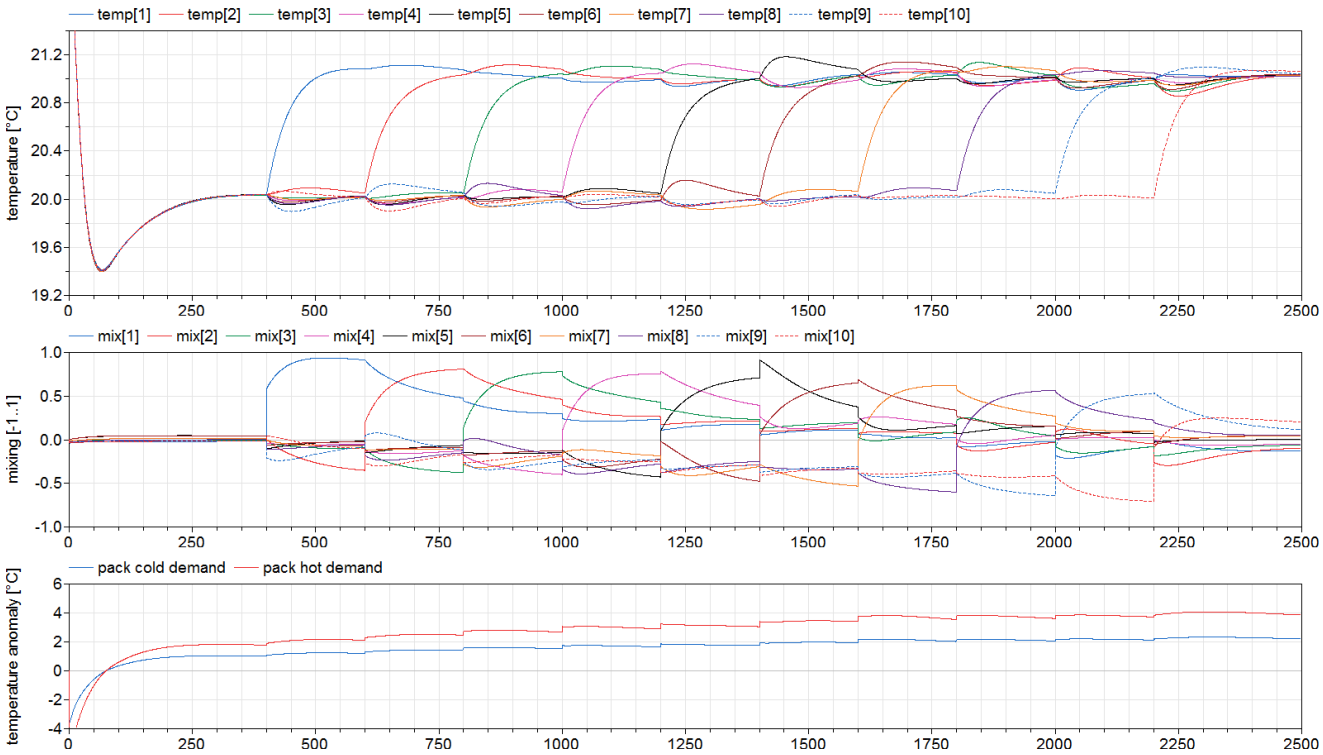

Figure 9. Response of controlled system to target temperature steps

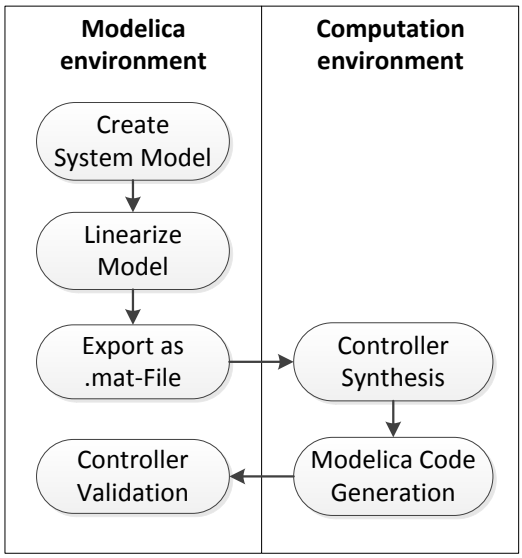

Figure 10. Workflow for controller synthesis

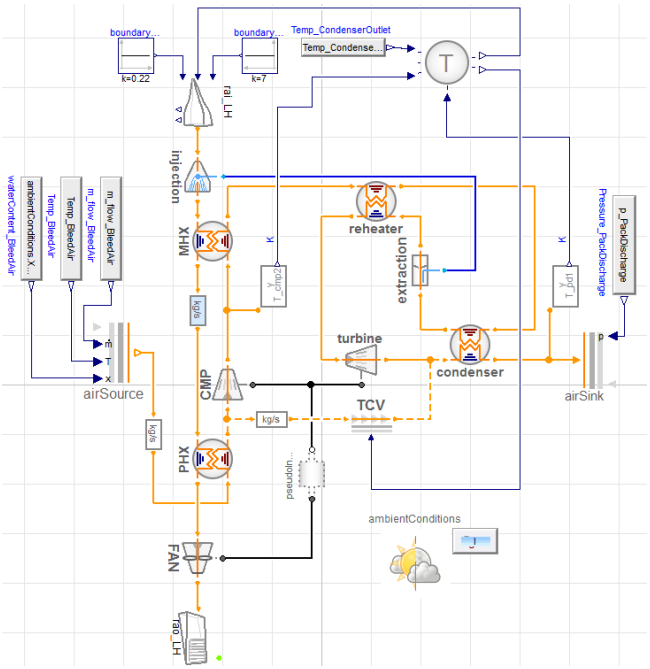

Figure 11. Diagram layer of air pack system model with three wheel bootstrap cycle

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 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 up inside the condenser again before it leaves the air 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 outside the aircraft's 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 11 are taken from an ECS library (Sielemann et al., 2007).

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 air packs were assumed so that simulations were carried out using the air pack Modelica model shown in Figure 11 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 TCV 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 realized 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 air 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 air 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 12 shows the result of the simulations for the different discharge temperatures. The model was simulated for a cruise flight phase at 39.000 feet 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 e.g. 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 10 K. The coloring represents the total drag of both packs and the black lines represent lines of constant drag. Due to confidential reasons, the values are normalized to an average drag value.

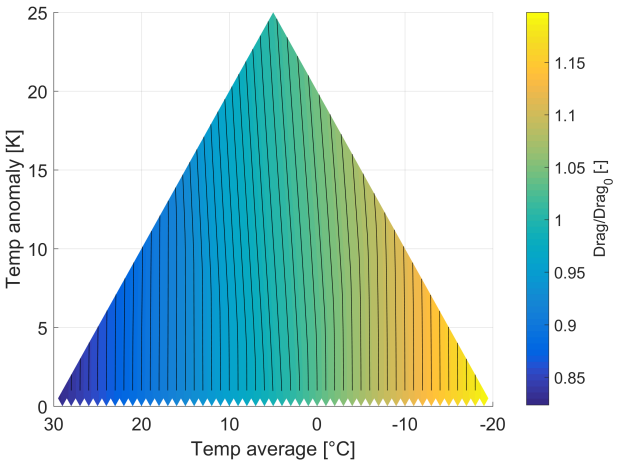

Figure 12. Drag caused by Ram air vs. average temperature and temperature anomaly

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.

5 Discussion

The resulting concept with its workflow depicted in Figure 10 proved to be effective for the analysis and optimization of a nonlinear MIMO control problem with a complex plant model. Yet, the concept of this paper for the temperature regulation of aircraft cabin represents only one item of a more general problem set. In this case, the control design was an integral part needed to evaluate the overall system design in an early phase. A rapid LQG controller provided a sufficient solution. Interfaces to and from Matlab eased the control design whereas Modelica served as main modeling and evaluation environment.

The long-term goal of this work however goes beyond this use case. Since many sub-systems are already highly optimized, further system optimization requires a higher level of sub-system integration and also more centralized control approach. This represents a higher level of integration and it often implies a low availability of corresponding test examples or rigs. Principal questions of controllability, performance of controllers and of the system as a whole need to be evaluated at an early design phase.

To this end, it is necessary to bring together the different software platforms that engineers use for control design (such as Matlab) and the modeling environments for system dynamics (such as Modelica). This is not the first paper addressing this problem. Typical attempts use the S-function standard to import the plant model to Matlab. Alternative approaches use the FMI-Standard to export the controller from Matlab to a Modelica environment. The presented work shows that code generation is also a feasible technique to achieve the desired result. It has the advantage that the final result is pure Modelica and does not require any further tools or interfaces.

Having the final result in pure Modelica is more suited when many variations of the plant model shall be created in order to test for various kinds of robustness. These changes may go beyond normal parameter changes since a variety of failure scenarios have to be modelled and simulated. In this application, typical faults are the malfunction of one pack and the malfunction of one or more valves in the ducting. Also the controller might be tested against Modelica models of higher fidelity. For all this work, having the controller in Modelica with all the internal controller signals openly available is the most convenient approach.

The aforementioned robustness tests still have to be performed to a large extent. These will hopefully not only further validate the temperature regulation concept but also the foreseen work-flow.

6 Conclusion

Modelica can be used to quickly generate models for validation studies of new concepts. However, the design of controllers based on this models makes additional tools necessary. Integrated modelling and computation environments such as Modia (Elmqvist et al., 2016) could be a remedy.

Acknowledgements

We thank Trey and Matt for inspiration.

References

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

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

Marco Bonvini and Alberto Leva. A modelica library for industrial control systems. In *Proceedings of the 9th International MODELICA Conference; September 3-5; 2012; Munich; Germany*, number 076, pages 477–484. Linköping University Electronic Press, 2012.

John Doyle. Guaranteed margins for lqg regulators. 1972.

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.

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

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.

Andreas Klöckner, Franciscus LJ van der Linden, and Dirk Zimmer. Noise generation for continuous system simulation. In *Proceedings of the 10th International Modelica Conference-Lund, Sweden-Mar 10-12, 2014*, number 96, pages 837–846. Linköping University Electronic Press, 2014.

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

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.

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.

Daniel Schlabe and Dirk Zimmer. Model-based energy management functions for aircraft electrical systems. Technical report, SAE Technical Paper, 2012.

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

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.

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

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.