Towards Hard Real-Time Simulation of Complex Fluid Networks

Dirk Zimmer[1]

[1] DLR German Aerospace Center, Insitute of System Dynamics and Control,
Münchener Strasse 20, 82234 Oberpfaffenhofen, Germany,
dirk.zimmer@dlr.de

Abstract

Complex fluid systems created with Modelica have often been difficult to simulate under hard real-time constraints since they typically involve non-linear equation systems that are difficult to solve especially within a predictable finite time frame. This paper explores the usage of a new approach that avoids non-linear equation systems and its suitability for real-time simulation. A model of an aircraft environmental control system is taken as a use case for this study.

Keywords: Thermal fluids, Thermal process systems, real-time simulation, environmental control systems

1 Introduction

1.1 Real-time simulation

Real-time simulation is a very useful tool, especially since it is a key enabler for virtual testing: control units can be tested against a virtual plant model or single components can be tested with software-in-the-loop in order to emulate a yet unavailable environment.

Although aviation, as our main application domain invests heavily in virtual testing efforts, real-time simulation has been difficult to achieve for the complex fluid systems that are contained within an aircraft, especially for the environmental control system (ECS). Figure 1 shows a Modelica example system that (simplistically) describes the flow of fresh air and recirculation air on board of a conventional aircraft.

Although this is not a realistic model, it is a suitable example for our purposes since it contains a variety of components and describes an active thermodynamic process as well as natural air distribution.

For public demonstration purposes, this model already runs in real-time (once the initialization phase has been passed) using DASSL as ODE solver. This is a soft real-time simulation since we cannot give any definitive statement on its computing speed or responsiveness other than a purely empirical: “it seems to be fast enough.”

Hard real-time requires definitive statements. The system needs to respond with a given frequency.

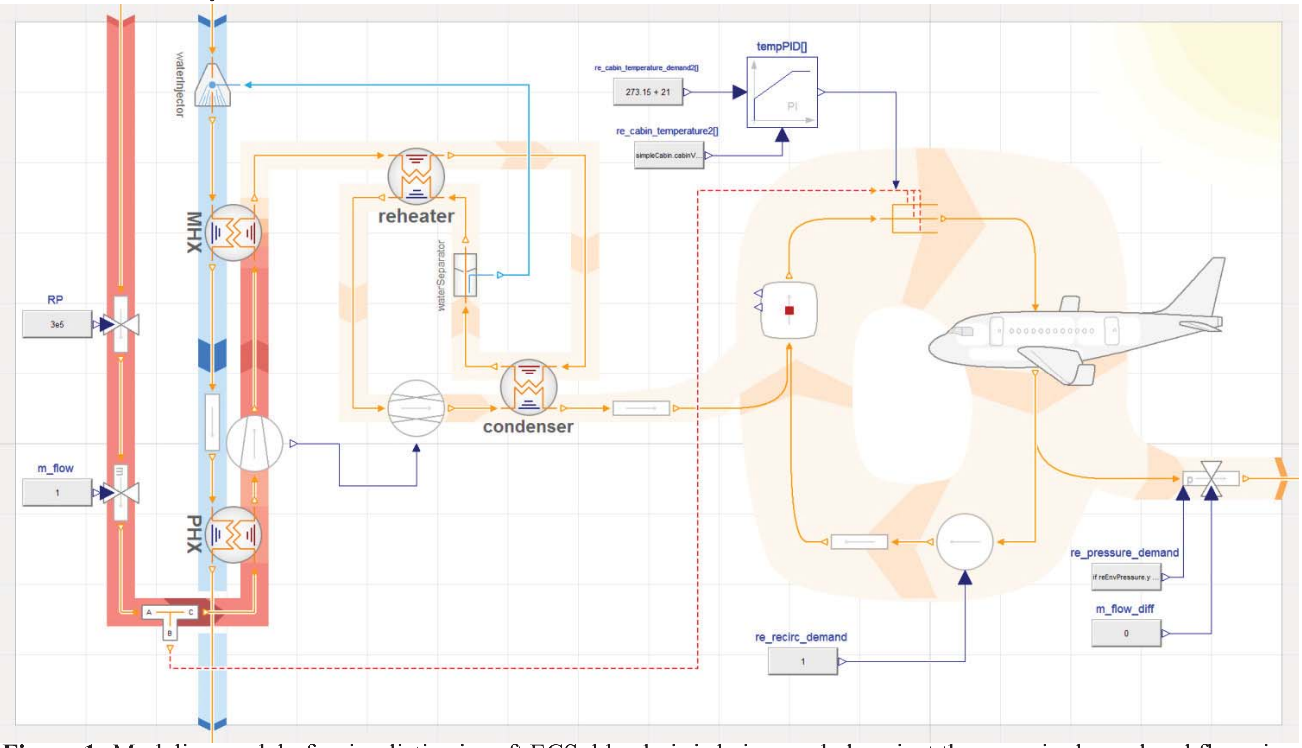

Figure 1: Modelica model of a simplistic aircraft ECS: bleed-air is being cooled against the ram-air channel and flows in a unit where it mixes with recirculated air. This mixture is then being warmed for 3 zones by further bleed-air and let into the cabin. From the 3 upper floor zones, the air flows to the underflow zone where a valve controls the cabin pressure. This model is derived from the work of Alexander Pollok (Pollok, 2017)

Within a period corresponding to this frequency, the system has to be completely updated and the error must be within certain bounds. These requirements heavily affect the suitability of the simulation code and its IT-infrastructure. Figure 2 shows a typical cascade of a real-time simulation.

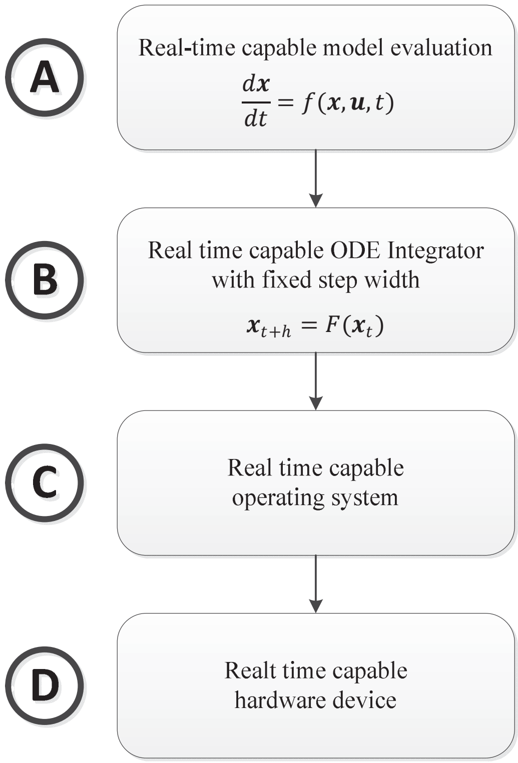

Figure 2: real-time cascade

In this paper, we focus on steps (A) and (B) since they are the most challenging for fluid systems. Once we have solved these issues, it is reasonable to assume that standard solutions for (C) and (D) will help us to go all the way.

1.2 Challenges specific to fluid systems

Why are (A) and (B) challenging for fluid systems? Of course, fluid systems have been simulated in real-time before (Schulze, 2011, Wetter, 2014) but rather in soft real-time. The systems either had to be simple or the simulation speed could only be determined by extensive testing.

Hard real-time is difficult since the model evaluation (A) often contain iterative numerical algorithms. This is because many systems are typically formulated using fluid streams (Casella 2006) and often form large systems involving non-linear algebraic equations (Zimmer 2013) that are then solved using iterative non-linear solvers. Once these solvers are required to evaluate the model function $dx/dt = f(x, u, t)$, hard real-time capability is lost. It is in general practically impossible to make definitive statements on how many iterations are needed for convergence and whether convergence can be guaranteed in the first place. To reach the area of convergence, homothopy methods are used (Casella 2011) primarily at initialization. During simulation the step-size control of the ODE-solver shall guarantee convergence. Again for hard real-time, variable step-size control is not feasible.

Since for (B) explicit ODE solvers (such as explicit Runge-Kutta methods) with fixed step size are preferred, the stiffness of the system becomes an issue. Flows between small volumes may cause such stiffness and small volumes are often introduced by modelers in order to fight the problem of (A): truly a viscous circle.

1.3 A new approach

Fortunately, we made recent progress regarding the DAE-based modeling of fluid streams (Zimmer, 2018). This progress was triggered by the needs to achieve a high level of robustness. Robustness in this context means that the simulation shall not return any errors due to solvability of non-linear equation systems and that the modeler shall not have to care overly about initialization of the system.

To this end, a computational scheme was developed that ensures that once the components robustly compute, also the complete system will robustly compute, meaning no error due to unsuccessful application of iterative numerical solvers.

We can use the same scheme to avoid non-linear equation systems altogether. If every component has a form that enables an explicit computation of the thermodynamic state in the downstream direction, then the thermodynamic states of the whole system can be computed explicitly in the downstream direction. Evidently this will address task (A) in our real-time cascade and Section 2 will briefly discuss its concept. Yet, this is only a very short repetition and the reader is invited to study (Zimmer, 2018).[1]

As part of this concept, the mass-flows $\dot{\boldsymbol{m}}$ become a state vector of the system that is (for natural parameters) typically associated with eigenvalues that are strongly negative or strongly imaginary. This prevents the application of explicit methods with sufficiently large step-width.

However, there are methods how these eigenvalues can be manipulated in such a way that they can be placed within the stability region of a stable method without interfering with the dynamics of the relevant thermodynamic processes (e.g. the heat capacity of a volume, etc.). This will address task (B) and is discussed in Section 3

Finally, we apply the lessons of Section 2 and Section 3 to our demonstration example. This shows the principal feasibility of the proposed methodology and lets us speculate about its limitation and potentials.

[1] Unfortunately the notation w.r.t p and $\hat{p}$ differs in the former paper. A proper journal paper is being written.

DOI
10.3384/ecp19157579

2 Avoiding non-linear equation systems

Non-linear equation systems are not only a problem for real-time systems, they are a problem for robustness in general. However, the pressure gradient of a flow along a straight pipe (see Figure 4) contains a component that is independent of the thermodynamic state and only linearly dependent on the time derivative of the mass flow rate. This gradient Δr corresponds to the inertial force f for accelerating an arbitrary line flow of length Δs:

$$\Delta r \cdot A = \Delta f = \frac{d\dot{m}}{dt}\,\Delta s$$

We can decompose the pressure p into the terms r (denoted here as inertial pressure) and $\hat{p}$ (denoted here as steady mass-flow pressure):

$$p = \hat{p} + r$$

This decomposition is useful since for gases, r is often very small. For liquids r can become sizeable but many liquids are in their properties (such as density or viscosity) relatively insensitive to r.

Hence, it makes sense to apply a different spatial resolution for $\hat{p}$ and r. We propose that the spatial discretization of $\hat{p}$ may be chosen as fine as the modeler wants but r is only discretized between boundaries and volume elements. The blue lines in Figure 3 show this discretization for our example.

The paper (Zimmer, 2018) demonstrates that if you apply this discretization scheme, $\hat{p}$ can be computed downstream and r results from the occurring differences in $\hat{p}$ by a linear equation system. If we also choose to use $\hat{p}$ and not p for the calculation of the thermodynamic properties, all non-linear thermodynamic computations in the total system can be brought into an explicit form.

Precondition is that the component models enable an explicit calculation of their outlet state from their inlet state and do not relate flows directly by algebraic equations. This precondition is not always easy to meet but it is practically possible (albeit with some effort).

In this way, we can achieve a fully explicit form by accepting an error that originates from neglecting the influence of r on the thermodynamic state. The error is often very small and is increasing in downstream direction until a volume or boundary is met. For steady mass-flow rates when $d\dot{m}/dt = 0$, the error is zero; see also (Otter, 2019) for a discussion of the error in a segmented pipe.

Another price for this explicit form is that all mass-flows now become state variables of the system. However, in many technical systems many components share the same mass flow, so this price is often acceptable. The implementation of this methodology in Modelica requires a new fluid library with its own connectors. Again, the paper (Zimmer, 2018) illustrates such a library called HEXHEX and the demonstration example of this paper has been built up solely with components from this library. Note that the modeler needs not concern itself with the discretization scheme. It becomes an inherent feature of the HEXHEX library.

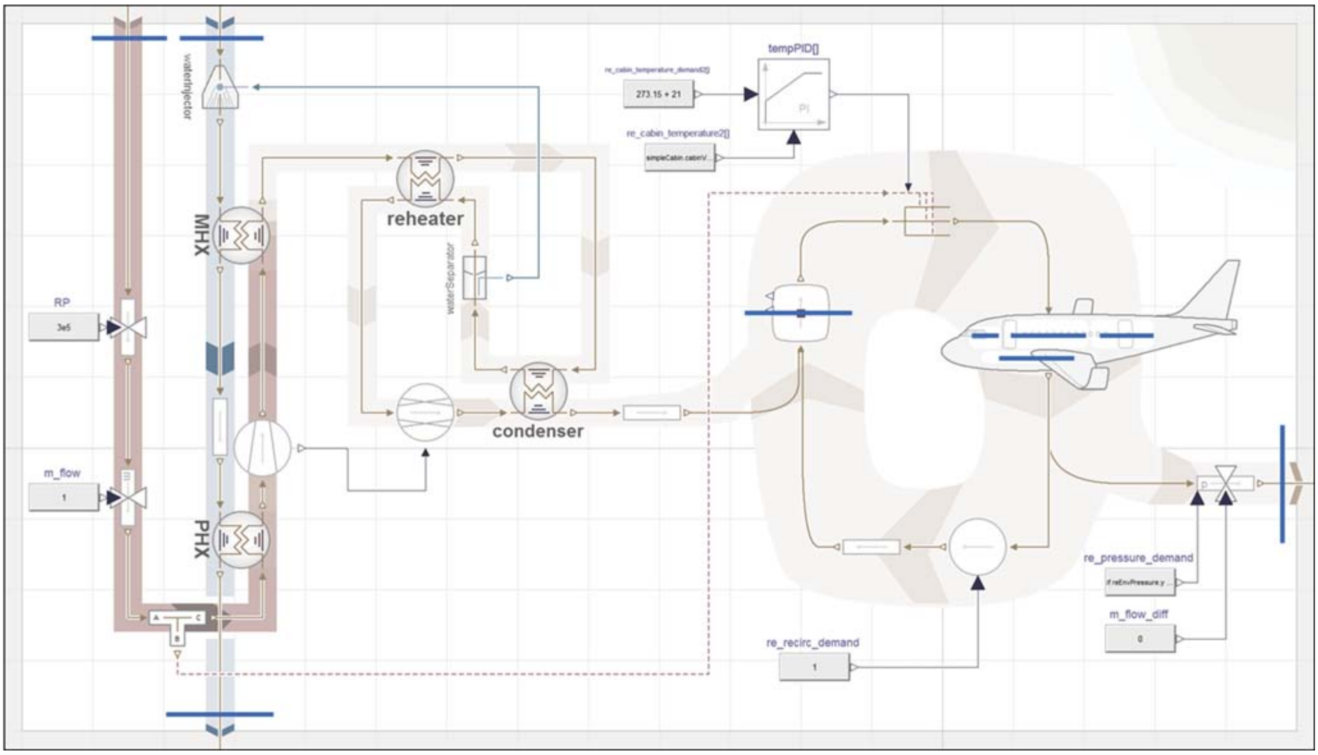

Figure 3: The blue bars indicate the discretization chosen for $\boldsymbol{r}$. In between these bars the influence of $\boldsymbol{r}$ on $\boldsymbol{p}$ is neglected.

3 Coping with Stiffness

Although the mass-flow dynamics help to avoid non-linear equation systems, they also create a problem on another level: typically they generate very fast transient towards equilibrium and hence corresponds to strongly negative eigenvalues. The system thus becomes stiff and explicit solvers can only be applied with very small step-size; often too small in order to meet the performance requirements.

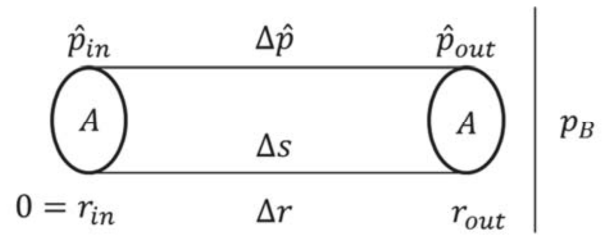

Figure 4: A simple pipe representing a generic 1-D system. The law for the pressure drop $\Delta\hat{p}$ can be arbitrary non-linear or time-dependent.

For better understanding we may look at a 1-dimensional, linearized system. Again: for a straight pipe as in Figure 4:

$$\frac{d\dot{m}}{dt} = \frac{A}{\Delta s}\Delta r$$

Δr now accounts for the pressure difference at the outlet boundary:

$$\Delta r = \hat{p}_{out}(\dot{m}) - p_B$$

As can be seen the outlet pressure is itself a function of the mass-flow. We can linearize this function for $\dot{m} = \dot{m}_0$ and retrieve:

$$\frac{d\dot{m}}{dt} = \frac{A}{\Delta s}\left(\hat{p}_{out}(\dot{m}_0) + (\dot{m} - \dot{m}_0)\frac{\partial\hat{p}_{out}(\dot{m}_0)}{\partial\dot{m}} - p_B\right)$$

Evidently, there are two sources of stiffness: the “inertia” of the mass flow, defined by its geometry $A/\Delta s$ and the sensitivity w.r.t. the steady mass-flow pressure $\hat{p}$: $\partial\hat{p}(\dot{m}_0)/\partial\dot{m}$ the latter shall also be negative in order to form a stable system (a condition that is typically fulfilled but can be violated for instance in the unstable area of a compressor).

Which of these two factors is dominant depends on the concrete (sub-)system at hand. However, distinguishing between two categories is helpful: there are

- uncontrolled mass-flows and
- controlled mass-flows.

For both categories, we will suggest means to manipulate the natural eigenvalues of the system into a range suitable for explicit solvers. Figure 5 shows the stability region of Runge-Kutta 3 and illustrates the idea of manipulation the set eigenvalues from $\boldsymbol{\lambda}$ to $\boldsymbol{\lambda}'$

The stability region is in grey and the area of interest for the actual dynamics is in orange. The eigendynamics of interest are correspondingly marked by green eigenvalues. The blue eigenvalues represent dynamics needed to compute the system but not of actual interest. These may prevent the application of an explicit solver and hence shall be manipulated to be within the stability region. The challenge is to only move specific eigenvalues without changing the complete dynamics.

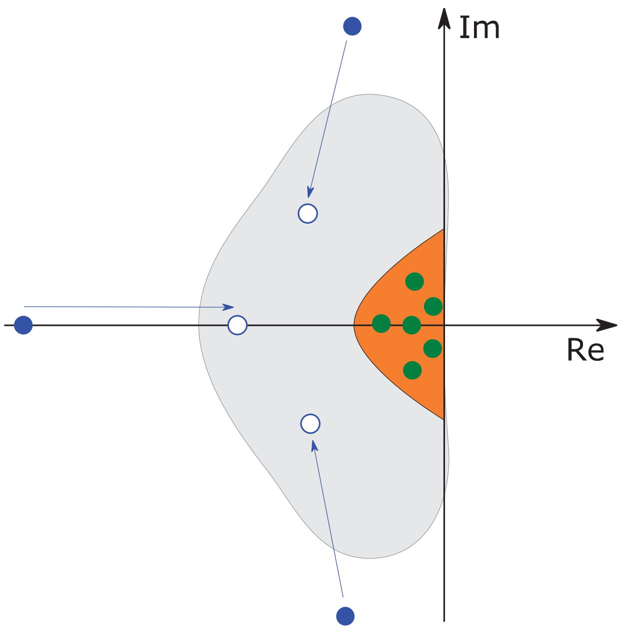

Figure 5: Manipulation of eigenvalues. Goal is to move the blue eigenvalues into the stability region of an explicit solver without influencing the green eigenvalues.

Without doubt such a manipulation of eigenvalues is an ugly thing to do but there is a price for achieving hard real-time capability. Loss of precision at fast transients is often an affordable price and hence can justify the manipulation of eigenvalues that are strongly negative or have a strong imaginary part.

Because it is an ugly practice, it is also often a little reported practice (although maybe quite commonly applied). But non-reporting does not make a practice irrelevant. It is better to report also on those things that sometimes just need to be done.

DOI 10.3384/ecp19157579

3.1 Coping with controlled mass-flows

Controlled mass-flows are typically part of actively driven systems. In our example in Figure 1, the bleed air mass-flows are controlled by valves and the recirculation mass-flow is controlled by a fan.

A typical valve model is an excellent example about the problems that are involved with controlled mass-flows. A classic implementation will regulate the valve opening (modeled here by linear resistance k) such that the measured mass-flow reaches the set-point:

$$\frac{d\dot{m}}{dt}\frac{\Delta s}{A} = \Delta r$$

$$\Delta\hat{p} = k \cdot \dot{m}$$

$$k \leftarrow \text{control input}$$

The inertial pressure then follows from the system composition. For a simple system we can assume that:

$$\Delta r = p_0 - \Delta\hat{p}$$

The eigenvalue of the system is then $-kA/\Delta s$ (at least if k is not directly dependent on $\dot{m}$ by the control input). Since k can become very large for a nearly closed valve, it is the dominating factor of the eigenvalue. Furthermore, this cannot be compensated by artificial increase of Δs, since k can also be very small if the valve is open.

To overcome this problem we propose to stipulate the mass-flow and in consequence make the valve a potential source of inertial pressure. The steady mass flow pressure drop $\Delta\hat{p}$ is then adopted so that Δr matches the fluid acceleration.

$$\frac{d\dot{m}}{dt}\,T = \dot{m}_{set} - \dot{m}$$

$$d\frac{\Delta\hat{p}}{dt}T = \frac{d\dot{m}}{dt}A/\Delta s + \Delta r$$

In this way, the eigenvalue corresponding to the mass flowrate can be arbitrarily stipulated to be $1/T$ with T being a parameter for this time constant.

Since this valve model is a source of inertial pressure, it can suppress all excitations of higher frequency. In consequence it violates the first law of thermodynamics for such high frequencies. The resulting error weakens for low frequencies and becomes zero for steady-state. When using such a model for a mass-flow control valve, one must thus ensure that excitations stay below a certain frequency. This, however, is almost a given for most real-time applications.

The method that has been shown here can also be adapted to other components used to control mass-flows such as compressors or fans. In a real implementation, one also has to deal with further constraints, for instance that valves must have a positive pressure drop $\Delta\hat{p}$.

3.2 Uncontrolled mass flows

Uncontrolled mass-flows are natural mass-flows that distribute the medium between volumes. Examples can be found again in Figure 1. The flow from the mixing unit to the cabin is assumed (in this model) as uncontrolled mass-flow. The flow between the different cabin zones and the flow from the cabin area to the cargo area also represent uncontrolled mass flows.

Again, it is a good idea to study a simple example: the flow between a volume through an orifice to a boundary element of fixed pressure p_0 . Using the bulk modulus K we can state the following equations:

$$\frac{d\hat{p}}{dt} = \frac{K}{V}\frac{\dot{m}}{\rho}$$

$$\frac{d\dot{m}}{dt} = \frac{A}{\Delta s}(\hat{p} - p_0 - k\dot{m})$$

where V is the volume and k the linear flow resistance of the orifice. The density is here only approximated by a constant ρ. The eigenvalues for this system are:

$$\lambda_{1,2} = \frac{k \pm \sqrt{k^2 - 4\frac{KA}{V\rho\Delta s}}}{2}$$

Here the difficulty for a real-time simulation is that the eigenvalues may be too strong in their imaginary part hence resulting in a high frequency behavior. This is for instance the case when small volumes are (almost) frictionless connected as small zonal elements in a cabin model. When the frequencies of such a model become too high for real-time simulation, we can only replace the actual dynamic behavior with a behavior representing a potential steady-state solution of the original model. To this end, we have to apply artificial damping.

The problem with just increasing the value of k to k' is that we may affect the steady-state solution. Fortunately, two neat little tricks help us to avoid this:

1. The artificial damping k' shall only be applied to the mass-flow attributed to the volume expansion and not applied to the mass-flow through the volume element.
2. The pressure drop resulting from the artificial damping k' is not attributed to the steady mass-flow pressure $\hat{p}$ but to r, the inertial pressure.

In this way, the steady-state solution remains unchanged.

For clarification, let us examine a volume model with two ports as illustrated by Figure 6. The mass of the volume follows from the mass balance

$$\frac{dM}{dt} = \dot{m}_1 + \dot{m}_2$$

The equation for mass-fraction follows from conservation of mass and the equation for the specific enthalpy follows from the conservation of energy. Mass, mass fraction and enthalpy then also stipulate the pressure $\hat{p}$. For $p = \hat{p} + r$ a term r can be added. To apply the proposed artificial damping we propose:

$$r = k' \frac{dM}{dt}$$

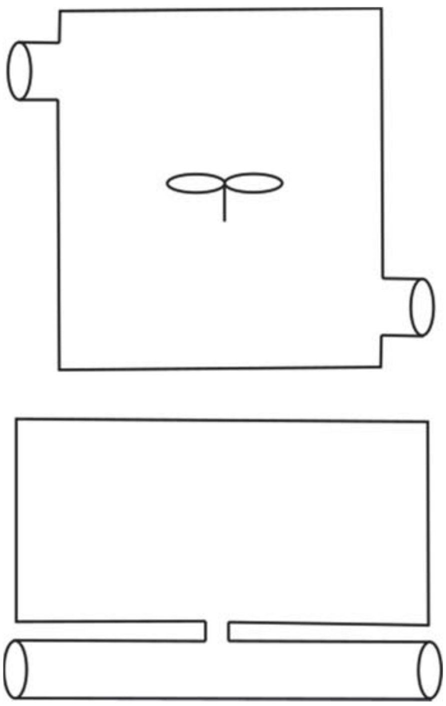

Figure 6: Schizophrenic view on a volume element with two ports. Top figure: in terms of enthalpy and mass fraction, the volume is idealized as an ideal mixing unit. Bottom figure: in terms of pressure and mass-flow rate, the volume is regarded as a frictionless pipe with an orifice to a larger expansion volume. The pressure drop of the orifice is only attributed to the inertial pressure $\boldsymbol{r}$ in order to dampen the dynamics of the mass flow.

There remains one last issue, the value for k' that is needed to achieve critical damping has the lower limit of $k' > 2\sqrt{KA/V\rho\Delta s}$. For a high frequent system, we thus simply replace strong imaginary eigenvalues by strong negative eigenvalues. This might be of no help for the real-time simulation. Hence, before implementing k', we may want to artificially lower the frequency. Manipulation Δs to a larger $\Delta s'$ will achieve exactly this without affecting the steady-state solution or the thermal properties of the system.

3.3 Additional remarks

As stated previously, the manipulation of eigenvalues is an ugly business and one is forced to sacrifice physical correctness to some degree.

Nevertheless, the methods shown above enable such a manipulation while upholding two important principles:

1. The solution for steady mass flows, meaning $d\dot{m}/dt = 0$ does not change.

2. Changes to the natural parameters do not involve any parameters for thermal capacities or the like.

For instance, another variant to fight stiffness would be to artificially increase volume sizes. But this would also influence the thermal capacity and the time constant for mixing processes. So this is not a good idea.

Altogether, the methods have their appeal but should be used with care.

4 Demonstration

Let us now simulate the aircraft ECS model of Figure 1 under hard real-time constraints. Figure 3 already illustrates that the concept of HEXHEX naturally avoids any non-linear equation systems as long as no component contains a non-linear equation system for the classic downstream computation.

Figure 7 highlights the components that have been manipulated from the original model in order to reduce the stiffness. The red elements have been manipulated for mass-flow control and the green volumes elements have been manipulated with an artificial damping on the mass-flow dynamics.

In total the system describes 52 continuous time states:

DOI 10.3384/ecp19157579

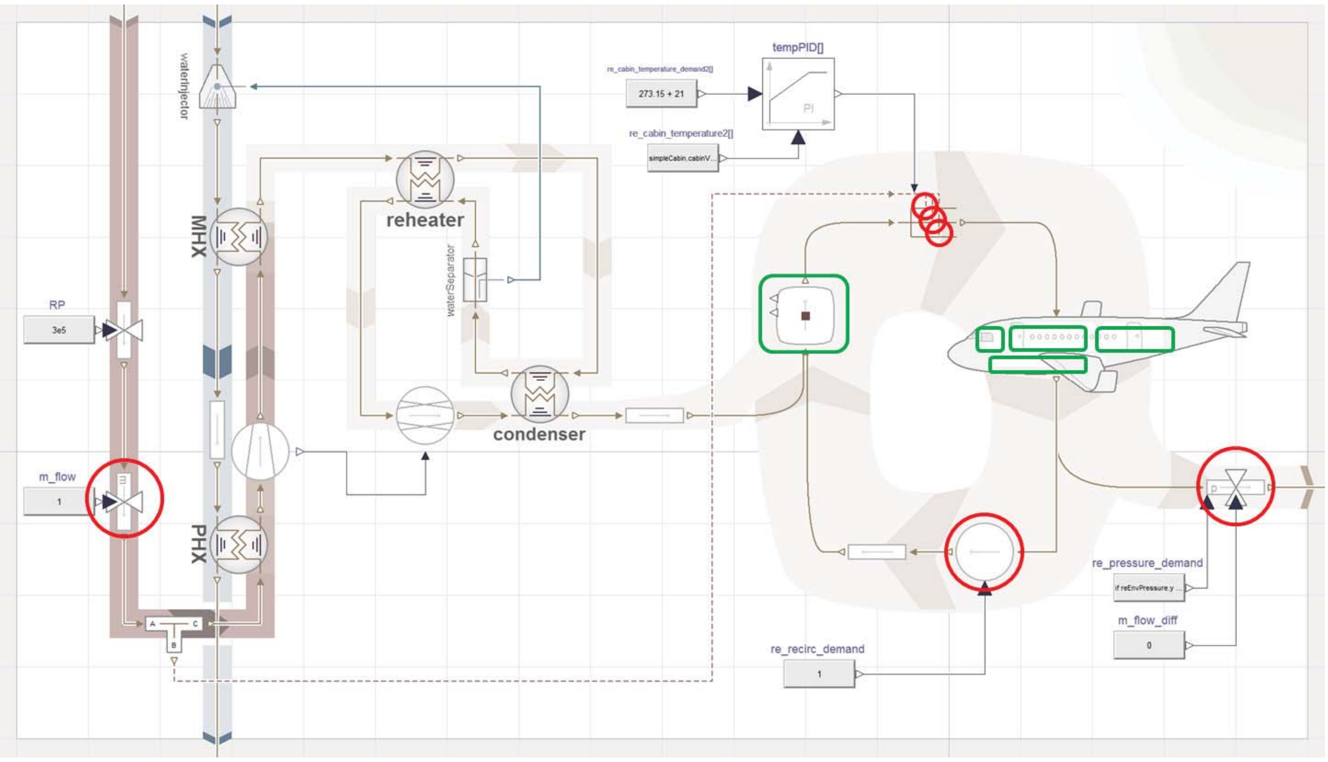

Figure 7: The red circles indicate components that have been manipulated for controlled mass flows. The green rectangles depict the volume elements where artificial damping has been applied.

- 6 states describe the aircraft altitude, the controllers for ram-air opening and zonal temperature.
- 15 states attribute to the pressure, enthalpy and water content of the 5 volume elements.
- 11 states describe the thermal dynamics of the heat exchangers and the cabin lining.
- 8 states describe the natural mass flows in the ram air channel, water separation and between the volume zones in the aircraft cabin as well as in the mixing unit.
- 12 states describe the 6 controlled mass flows and the corresponding pressure differences for the control component.

Since all components have been developed so that they enable an explicit computation of their outlet state from their inlet states, no non-linear equation system occurs in the whole translated model code.

The equations for the mass-flow dynamics result in 8 linear equation systems of size {5, 20, 15, 2, 8, 8, 8, 8, 8} that Dymola can reduce to the size to {0, 2, 3, 2, 3, 2, 2, 2} by the method of tearing.

Explicit Runge-Kutta 3 with a stepsize of 40ms (25Hz) is chosen as ODE solver. Figure 8 shows the eigenvalues of the system at t=10s. Evidently all eigenvalues are well within the boundaries of RK3 with the chosen step width. Even larger step widths could be chosen. Furthermore, no special effort is needed for initialization in this particular example.

In order to synchronize with real time, the corresponding block from the Modelica Device Drivers Library (Thiele 2017) has been used.

In terms of performance the simulation easily runs in real-time. Idle time (with deactivated outputs) is roughly 90% on an Intel Xeon 2,66GHz which indicates that the simulation on its own could run 10 times faster.

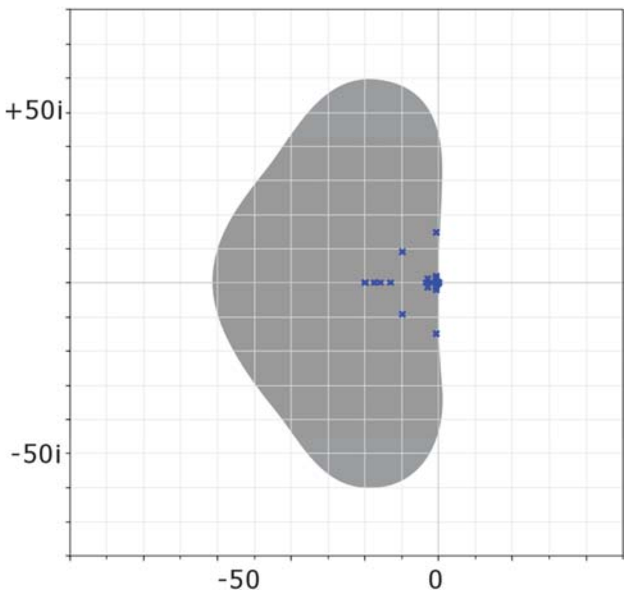

Figure 8: The eigenvalues of the system as blue crosses against the stability region of RK3 with h=0.04s

5 Outlook and Final Remarks

Although this new concept is a very promising path for hard real-time capability of thermofluid models, there are still a few missing links.

5.1 Missing parts

First of all, we shall not forget that we have only discussed the steps (A) and (B) of Figure 2. A more realistic test needs to be performed on real-time hardware (D) with a real-time operating system (C). However, it is a reasonable expectation that standard solutions will do the job.

Yet, there remain issues with the application of explicit solvers (B). So far, we have only shown for a few operating points that the eigenvalues are within the stability range of the chosen integration method. This is a too weak statement. Either this analysis must be performed for an extensive set of operating conditions or more favorably an algebraic analysis can be performed.

To this end, the simplistic 1-dimensional computation of eigenvalues as in section 3 must be replaced by a full multi-dimensional analysis for the system at hand. The model function:

$$d\boldsymbol{x}/dt = f(\boldsymbol{x}, \boldsymbol{u}, t)$$

must be bounded in its partial derivatives with respect to the mass-flow $\partial f / \partial \dot{m}$ for the complete state-space domain. Furthermore, strict statements are needed for all other eigenvalues.

Although such an analysis is by no means simple, it should be principally feasible under certain preconditions.

In terms of precision, the behavior at faster transient needs to be properly analyzed. So far, only a qualitative judgement of the simulation result was performed.

5.2 Future potential

The presented use case is still a relative simple example and hence it is yet unclear for how large or how complex system the proposed methodology can be successfully applied. For the future, more testing is required but for now, here are a few principal considerations.

First of all, the provided example already performs 10 times faster than real-time on a conventional desktop PC of 2012.

The model evaluation is certainly slowed down by the evaluation of the media models. Papers such as (Quoilin, 2014) suggest that there is substantial room for improvement.

The application of explicit Runge-Kutta makes parallelization an attractive choice. Especially those parts of the system that are separated by a volume element are attractive to compute in parallel.

Last but not least, also multi-rate integration (Ranade, 2014) becomes an attractive option since some thermal time constant can be simulated at slower rate.

Altogether, we estimate that the proposed methodology can be applied quite easily to systems 10-100 times larger (or more complex) than what has been presented here. A demonstration of this is, however, still pending.

5.3 Concluding Remarks

The inertial pressure (or its counterpart in terms of pressure-gradient force) is a very useful term to describe thermofluid systems without having to numerically solve non-linear equation systems. Only linear equation systems need to be solved during runtime. This can be done in predictable time.

The inertial pressure is also a very useful term to manipulate the eigenvalues of the system. In this way, the resulting model becomes suitable for the application of explicit ODE solvers such as Runge-Kutta.

In this way, the proposed methodology offers a promising way to model and simulate complex fluid systems under hard real-time constraints within a Modelica framework.

Acknowledgments

The author wants to thank Alexander Pollok for setting up the original demonstration example and Daniel Bender for his contributions to the HEXHEX library.

References

Casella, Francesco, et al., (2006) The Modelica Fluid and Media library for modeling of incompressible and compressible thermo-fluid pipe networks, *Proc. 5th International Modelica Conference*, Vienna, Austria, Vol.2, pp.559-568.

Casella, Francesco and Sielemann, Michael und Savoldelli, Luca (2011) Steady-state initialization of object-oriented thermo-fluid models for homotopy methods. In: *Proceedings of 8th International Modelica Conference. 8th International Modelica Conference*, 20.-22. Mar 2011, Dresden.

Otter, Martin et al., (2019) Thermodynamic Property and Fluid Modeling with Modern Programming Language Constructs. In: *Proceedings of 8th International Modelica Conference. 13th International Modelica Conference*, Mar 04-06, 2019, Regensburg, Germany.

Pollok, Alexander und Schröffer, Andreas (2017) Simulation of Helmholtz Resonance Effects in Aircraft ECS. In: *6th International Workshop on Aircraft System Technologies. 6th International Workshop on Aircraft System Technologies*, 21.-22. Feb. 2017, Hamburg, Deutschland.

Quoilin, Sylvain et. al. (2014) ThermoCycle: A Modelica library for the simulation of thermodynamic systems In: *Proceedings of the 10th International Modelica Conference* 2014, Lund, Sweden.

Ranade, Akschay, Francesco Casella (2012), Multi-rate integration algorithms: a path towards efficient simulation of object-oriented models of very large systems. *Proceedings of the 6th International Workshop on Equation-Based Object-Oriented Languages and Tools (EOOLT)* , Berlin, Germany

Schulze, Christian (2011) Real-Time Simulation of Vapour Compression Cycles. In: *Proceedings of 8th International Modelica Conference*., 20.-22. Mar 2011, Dresden.

Thiele, Bernhard, et. al. (2017) Towards a Standard-Conform, Platform-Generic and Feature-Rich Modelica Device Drivers Library. In: *Proceedings of 8th International Modelica Conference*. 12th International Modelica Conference, Prague, Czech Republic

Wetter, Michael, et al. (2014), The Modelica Buildings Library. In: *Journal of Building Performance Simulation*, 7(4), 253–270, 2014

Zimmer, Dirk, (2013), Using Artificial States in Modeling Dynamic Systems: Turning Malpractice into Good Practice. In: *Proceedings of the 5th International Workshop on Equation-Based Object-Oriented Languages and Tools (EOOLT)* , Nottingham, United Kingdom

Zimmer, Dirk and D. Bender, A. Pollok (2018) Robust Modeling of Directed Thermofluid Flows in Complex Networks. In: *Proceedings of the 2nd Japanese Modelica Conference*, Tokyo, Japan

DOI
10.3384/ecp19157579