Solving Flow Balancing Problem for Hybrid-Electric Aircraft Cooling Systems

Clément Coïc Michael Sielemann Nirmala Daniel Andersson
Modelon, `{name.surname}@modelon.com`

Abstract

A flow balancing problem consists of sizing restrictions on flow branches of a fluid system to match desired flow rates on each branch. The problem is rarely trivial as parallel branches routinely contain many components with nonlinear pressure loss characteristics each. This paper introduces the Physics-based Solving capabilities implemented in Modelon Liquid Cooling library. This new capability enables conveniently solving such flow balancing problems with steady-state requirements. The benefits of this solution are discussed using an aircraft thermal management system as example.

Keywords: Flow Balancing, Cooling System, Liquid Cooling, Thermal Management System, Aircraft, Sizing, Modelon Impact

1 Introduction

Recognizing the need for a sustainable future, the aerospace industry is heavily researching innovative technologies that could reduce its impact on the environment. While disruptive technologies such as hydrogen-powered aircraft are being investigated, more incremental ones such as hybrid-electric designs – combining a conventional propulsion system with an electric one – are expected to enter service before, providing substantial fuel efficiency and emissions improvement.

However, adding the hybrid electric propulsion system has impact: it adds direct and indirect weight. The weight of electric propulsion components – batteries, inverters, electric motors, etc. – directly reduces the aircraft payload. In addition, these components need to be cooled, indirectly increasing the weight of the thermal management system. Therefore, achieving an optimized design, when it comes to hybrid-electric aircraft, involves a careful trade-off between the electric propulsion system and the thermal management system designs.

One challenge to solve when designing both coupled systems is to size flow restrictions on the cooling branches. These restrictions, on each branch, define the nominal flow rate of cooling fluid and, therefore, the cooling capacity of a branch. These cooling requirements are directly derived from heat dissipated by each component – here, the electric propulsion components. If we decide to neglect storage effects of heat capacities in the design of the thermal management system, then the amount of heat to be extracted is thus constant over (a sufficiently long period of) time (in other words a steady-state value). The most demanding of these (quasi) steady-state conditions are identified by engineers as sizing conditions, and are, for instance, function of the components selected for the system to be cooled.

Optimizing the hybrid-electric aircraft design is thus an iterative process which aims at maximizing the aircraft range for a given minimum payload. Each iteration involves assessing the electric propulsion component sizes and associated heat loads in order to solve the flow balancing problem – which is key for the thermal management system sizing. This paper focuses on a robust solution to the steady-state flow balancing problem.

With the goal to give more technical background on the problem, section 2 introduces the flow balancing problem for the specific case of a thermal management architecture of a hybrid-electric aircraft. Section 3 discusses the tools we used to solve the flow balancing problem and introduces Physics-based Solving, a combined symbolic and numerical computation technique for the Modelica language. Section 4 shows how the flow balancing problem is solved. Section 5 concludes this publication and draws conclusions.

2 A Thermal Management System for Hybrid-Electric Aircraft

2.1 Architecture Selection

The scope of this paper is not to propose a new hybrid-electric aircraft architecture with an innovative thermal management system but rather to solve a recurrent problem in the currently suggested architectures. To support this argument, a typical architecture , presented by Gkoutzamanis (2022), is used in this paper.

The selected hybrid-electric aircraft architecture is targeting regional commuter aircraft. It consists of two conventional turbo-propellers, from which mechanical power is also extracted to feed electric generators that, in turn, power an electric motor connected to an aft Boundary Layer Ingesting (BLI) fan. For electric power management reasons, the alternating current (AC) electricity generated is converted to direct current (DC), potentially stored into a battery, and inverted back to AC, prior to consumption by the electric motor. **Figure 1** illustrates this architecture.

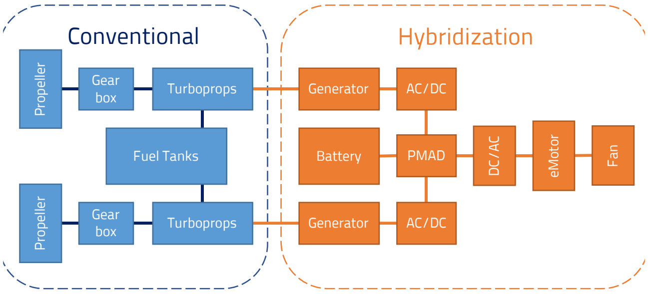


Figure 1. Selected hybrid-electric aircraft architecture.

A highlight of this architecture is that hybridization is added to the conventional propulsion. A typical engineering problem is to estimate differences with respect to the conventional propulsion architecture ("baseline"). How much weight does the hybridization add? One must consider the electrical power system, the thermal management system, and snowball effects on the aircraft structure. Beyond this, engineers face many more questions such as for installation constraints, safety aspects, availability, maintenance, etc. We focus on the thermal management system cooling the electric propulsion system – depicted in **Figure 2**.

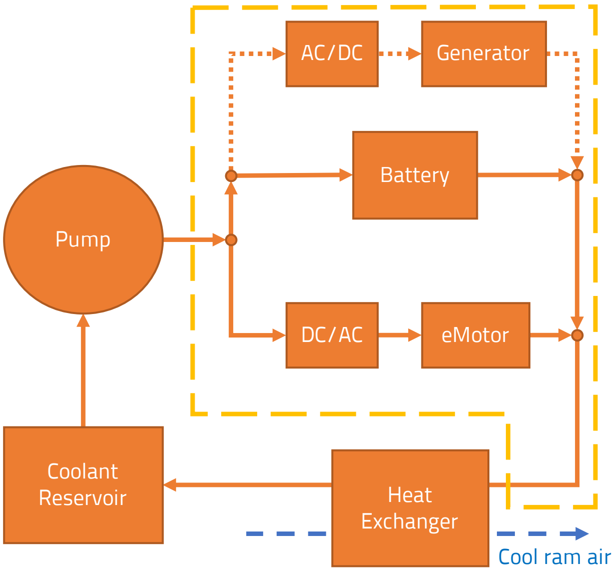


Figure 2. Added Thermal Management System.

The thermal management system on such hybrid electric aircraft must cool both the conventional components and the thrust-generating electric power system. Based on the location of the components on the aircraft, two main routes are identified:

1. Top two branches near the electric power generation and storage – near the turboprops, and thus wings. The dash line represents a branch that can potentially be by-passed, as the electrical components might not always be active, e.g., at take-off.
2. One bottom branch for the electronics and electric motor (eMotor) powering the aft BLI fan – on the rear part of the aircraft.

After capturing the heat loads, both routes merge before dissipating this heat via an air-cooled heat exchanger.

2.2 Flow Balancing Problem Statement

From the thermal management point of view, each component is treated as a heat load to dissipate. While each component can have several different characteristics, only the amount of heat it generates initially matters. In a later step of the design, the pressure loss characteristics and thermal resistance (between coolant bulk flow and heat load) of cooling elements, e.g., cold plates, are refined for each component to be cooled. Nevertheless, the overall need for coolant flow shall not be changed and is function of the heat loads.

Table 1 – also extracted from Gkoutzamanis (2022) – summarizes the heat loads to evacuate for each component and the temperature limits these components should not reach. These temperatures are typically imposed by material constraints. For instance, Budinger (2020) states that the "main design criterion for the motor is the maximum winding temperature" . Every technology might have different criteria driving these temperature limits but the limit shall not be reached to ensure component integrity. While heat loads are computed in such a way that the temperature is not reached, it is still relevant to monitor the component temperature in the simulation outputs and ensure that these requirements are met.

Table 1. Component heat loads.

Component	*Temperature limit* [°C]	*Heat loading* [kW]
eMotor	100	20
Generator	100	20
Inverter	65	10
Converter	65	10
Battery	40	10

To define the flow balancing problem in the thermal management context, we introduce an engineering design rule relating the heat load to the mass flow rate. For its derivation, we start with the First Law of Thermodynamics for a constant volume V, density ρ, internal energy u, enthalpy flow rate at each interface $\dot{H}_a$ and $\dot{H}_b$, and heat flow rate $\dot{Q}$:

$$V\frac{d(\rho u)}{dt} + \dot{H}_a - \dot{H}_b = \dot{Q}$$

If we assume steady state, then the mass flow rates at each interface are identical, $\dot{m}_a = \dot{m}_b = \dot{m}$, and the time derivative vanishes.

$$\dot{m}(h_a - h_b) = \dot{m}\Delta \mathrm{h} = \dot{Q}$$

DOI
10.3384/ecp19345

The engineering design rule for a typical coolant and cooling system technology, is to consider a design factor of, for instance, $\dot{Q}/\dot{m} = \Delta h = 30$ kW/(kg/s) – a mass flow rate of 1 kg/s can evacuate a heat load of 30 kW. This design factor combined with the heat loads of each components gives a first estimate of the mass flow rate required on each branch of the thermal management system to cool the electric propulsion system. It is important to understand that this is first step in an engineering workflow and assumes that, independently of the specific mass flow rate, all heat is transferred *as required* from the device being cooled to the coolant. The detailed design of the cooling surface, e.g., cold plate, can be conceived at a later stage. Here, it is sufficient to assume that the cooling is feasible.

Based on the thermodynamic properties of the coolant and the component inlet temperature, the design factor can be related to temperature change. With the simplifying approximation of constant specific heat capacity, we can deduce that the resulting temperature change can be roughly in the range of 8 K to 17 K for typical coolants propylene glycol water solution, polyalphaolefin, and turbine cooling oil MIL 23699.

The flow balancing problem consists of computing dimensions of calibration restrictions on each branch in order to match the desired mass flow rates, while ensuring that the physics laws are satisfied – e.g. here, the pressure drop of the three parallel branches are identical.

Obviously, each branch, in addition to these flow balancing restrictions, also includes a multitude of routing components – mainly bends and pipes – and aggregates their nonlinear characteristics. For the purposes of this paper, the pump and heat exchanger are substituted by ideal pressure boundaries so that only the branches and routings are considered for the problem solving – part of the system framed with a yellow dashed line. Several pipes and bends are added and parametrized to the system.

It is worth noticing that the flow balancing problem is a steady-state problem which is part of an overall design optimization loop. While Modelon Impact – the platform used in this paper to solve the flow balancing problem – is well suited to solve optimization problems (Coïc 2022), other platforms such as OpenMDAO (Zhao, 2019) (Hecken, 2020) or FAST-OAD (Delbecq, 2021) also proved to support this need. As Modelon demonstrated the optimization capability, with all these platforms, this paper focuses solely on the flow balancing problem solving.

3 Implementation of Physics-based Solving in Modelon Liquid Cooling

Modelon Liquid Cooling library (LCL) is used within Modelon Impact to model the thermal management system. As solving a steady-state problem is not the initial strength of the Modelica Language, Physics-based Solving (PbS) is added to LCL to support this workflow.

3.1 Modelon Impact

Modelon Impact is a next generation system modeling and simulation platform, leveraging the benefits of web and open standard technologies. With openness at its core, Modelon Impact supports standards such as Modelica, FMI, Python and REST (Modelon, 2022-a). The user-friendly browser interface provides modeling experts the tools they need to create, simulate, and experiment. Steady-state or dynamic solutions can be executed from the same model, reducing effort to get an answer (Coïc, 2020-b). Finally, the Modelon Impact API enables user-specific workflows through Python-based custom functions, and deployment of models to non-experts via targeted web applications or Jupyter Notebooks (Coïc, 2020-a).

3.2 Modelon Liquid Cooling Library

The library is used for modeling and simulation of liquid cooling systems in virtual prototyping, component dimensioning and control design.

The library includes more than 80 internal flow components such as pipes, bends and junctions with predictive geometry-based flow resistance correlations. It also includes generic components that can be calibrated from measurement data. More than twenty fluid models are provided in the library, with temperature-dependent properties to support cooling system modeling for water, customizable glycol-water and alcohol-water mixtures, every relevant water-salt mixture (e.g. potassium carbonate), calcium chloride, sodium chloride, potassium acetate, etc. The library also contains thermodynamic property models of aerospace-specific fluids such as turbine oil, polyalphaolefin, hydraulic oil and jet fuels.

Pre-configured templates guide users in creating simplified, high-performance heat exchanger stack models with 3D visualizations for parameter verification and presentation of resulting temperatures.

The Liquid Cooling Library (LCL) can be used effectively in conjunction with geometry-based models from the Heat Exchanger Library.

3.3 Steady-State and Physics-based Solving

When the answer expected from a model is the equilibrium point of the modeled system, steady-state simulation maximizes productivity; you obtain the result directly and faster, by orders of magnitude, and it simplifies post-processing of the results. You can extrapolate the gain on a design exploration, where you run hundreds or thousands of points (Modelon, 2022-b).

Modelon Impact includes steady-state solvers, as well as our Physics-based Solving (PbS) technology, that enables adding engineering insights derived from fundamental physical principles in models so that the steady-state simulation solves faster and in a robust way. Reconfiguring the numerical problem (without recompilation) also allows answering several questions with a single model.

PbS consists of instructions embedded in component models guiding the compiler and solver on iteration variable and residual selection for the steady-state simulation. This language construct enables changing the iteration variables and residuals based on Boolean parameters, without the need for recompilation. The information is stored in an object-oriented fashion, such that modelers can assemble systems graphically, and the desired solving can be deduced from the model topology (model instances and connections). This has been introduced in some detail before in (Coïc, 2020-b).

3.4 PbS Implementation in LCL

Typical Liquid Cooling models have a well-established flow direction. The fluid is directed from the pump toward the parts to cool-down and later cooled down through a heat exchanger before getting back to the reservoir.

As the flow direction is known, the authors implemented a serial approach of PbS in LCL. At the beginning of each branch of the cooling system, the mass flow rate is known or guessed. At the end of each branch, the type of component defines the residual equation of physics – either the user-defined value of a boundary condition, or the identical pressures for a junction.

In addition, the authors simplified the equations where possible, based on the steady-state and unidirectional flow assumptions. Notably, the Modelica language allows switching between such assumptions and thus the same library supports both dynamic and steady-state simulation – only a top-level parameter is changed to switch between the two simulation modes.

4 Solving the Flow Balancing Problem

In this section, a component model is first presented, to give more insights on the orifice sizing. Then, the thermal management system modeling and flow balancing solving are discussed.

4.1 Sizing Orifices

The specific case of the orifice sizing is presented here. PbS is added to the *orifice plate with circular opening* (see **Figure 3**). The component model can be configured into two different modes: orifice sizing and flow simulation. The former case imposes a user-defined flow rate to compute the orifice size $D_{opening}$, while the later enforces the orifice size to compute the pressure or flow unknown.

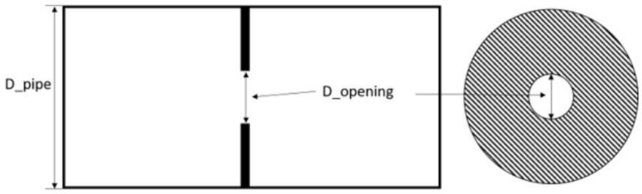
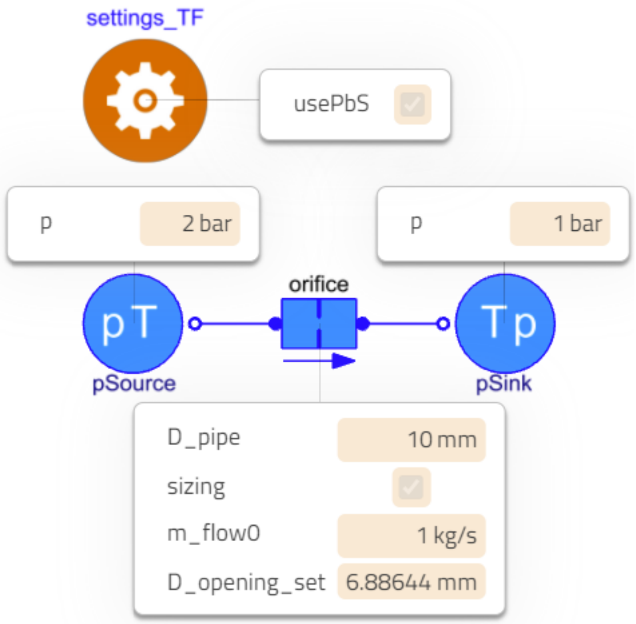


Figure 3. Orifice plate with circular opening.

A couple of statements are relevant to highlight:

- The orifice diameter $D_{opening}$ is of parameter variability. Solving the equations for it only makes sense at initialization or in a steady-state problem. To differentiate the user input parameter from the iteration variable for the sizing problem, $D_{opening_{set}}$ is used instead in PbS mode.
- At system level, it was mentioned that typically the mass flow is iterated on to match boundary conditions. For the case of the orifice sizing, the diameter being an unknown, the mass flow at the sizing point $m_{flow,0}$ shall be specified. Hence, the pressures are computed from the known mass flow rate and the pressure correlation in the branch.

A unitary test of the orifice component could thus be similar to **Figure 4**, where pressure boundaries are set, and the sizing point mass flow rate is prescribed. The pipe diameter D_{pipe} is a necessary parameter to compute the section change and would typically come from a pipe or previous component in a system level model. The greyed-orange background on the values indicates that it displays the results and thus cannot be changed. The orifice diameter $D_{opening_{set}}$ is the result of the simulation



usePbS ☑

p 2 bar

p 1 bar

D_pipe 10 mm

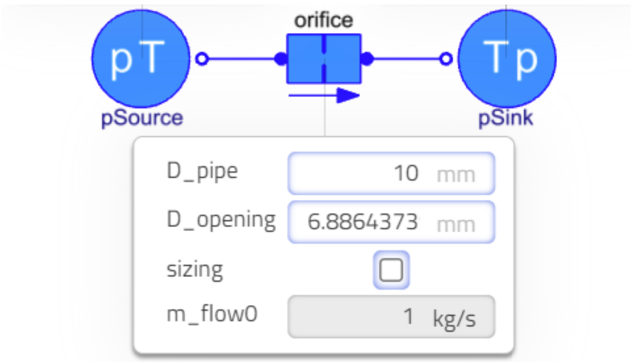
sizing ☑

m_flow0 1 kg/s

D_opening_set 6.88644 mm

Figure 4. Orifice sizing – unitary test 1.

The solver converged to an orifice diameter of about 6.89 mm for a pipe diameter of 10 mm to satisfy the 1 Bar pressure difference and 1 kg/s mass flow rate. Conveniently, in editing mode, it is possible to turn off the *sizing* mode and specify the desired $D_{opening}$ and simulate a standard flow simulation – without orifice sizing, hence $\mathrm{m}_{\mathrm{flow,0}}$ is not used and disabled.



D_pipe 10 mm
D_opening 6.8864373 mm
sizing ☐
m_flow0 1 kg/s

Figure 5. Orifice flow simulation – unitary test 2.

The orifice sizing is key in solving the flow balancing. It is now possible to introduce the thermal management system model and solving it is analogous to this section, only at a larger scale and for larger non-linear systems.

4.2 Modeling the System

The thermal management system is modeled in Modelon Impact using the Liquid Cooling Library. The model topology is based on **Figure 2** and shown in **Figure 6**.

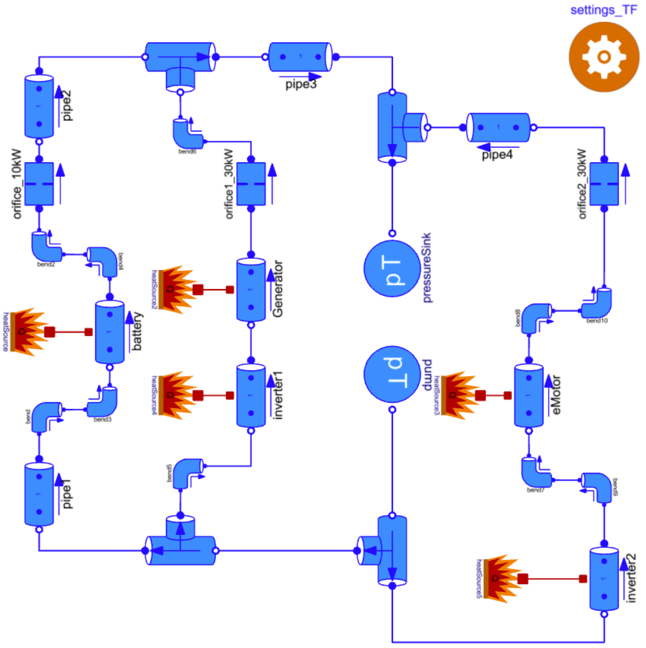


Figure 6. Thermal Management System model.

The following modeling decisions were stated previously but are repeated below for convenience:

- The pump and heat exchanger are substituted by ideal pressure boundaries so that only the branches and routings are considered for the problem solving – part of the system framed with a yellow dashed line.
- Several pipes and bends are added and parametrized to the system.

The components to be cooled are represented by heat sources. Their heat flow rates are set to the values specified in **Table 1** and their cooling interfaces are here modeled by pipes. Typically, a later refinement would involve modeling the cooling interface with higher fidelity, e.g., using several cold plates in parallel.

An orifice is added per branch to be sized as core aim of the flow balancing problem. These are named *orifice* followed by a number and postfix describing the total heat load that the branch needs to cool.

4.3 Solving the System

Once the system is modeled, the orifices are set in sizing mode and the mass flow rates at the sizing points are set to evacuate the heat loads with the considered design factor of 30 kW/(kg/s) – so $m_{flow,0}$ for *orifice1_30kW* is set to 1 kg/s.

The system is simulated with PbS in steady-state. The flow balancing solving happens without further user action. The compiler selects the iteration variables and residual equations – specified in the library at component level – and robustly solves the problem. **Figure 7** shows the results and associated thermal coloring.

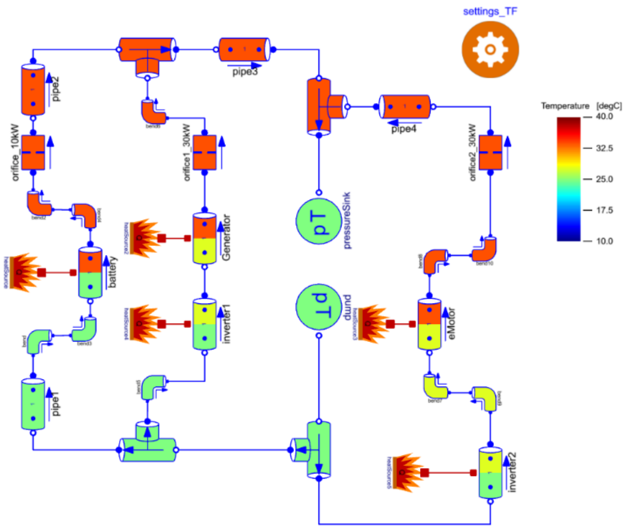


Figure 7. Thermal management system flow balancing solving

The sample value of the design factor results in $\Delta h_{heatload}$ =30 J/kg and, assuming 25 °C pump outlet temperature and propylene glycol water mixture 60%, the component outlet temperatures given in **Table 2**.

Table 2. Component temperatures.

Component	*Temperature inlet* [°C]	*Temperature outlet* [°C]
eMotor	28.0	33.9
Generator	28.0	33.9
Inverters	25.0	28.0
Converter	25.0	28.0
Battery	25	33.9

The computed orifices diameters can easily be applied to the system as Modelon Impact offers the option to start a simulation from the results of a previous one. Should you prefer entering a different value from a catalog, this is obviously also an option. A key result is to find the combined pressure loss characteristics of the branches for off-nominal pump head as shown in **Figure 8**. This then allows investigating further the design for additional requirement validations.

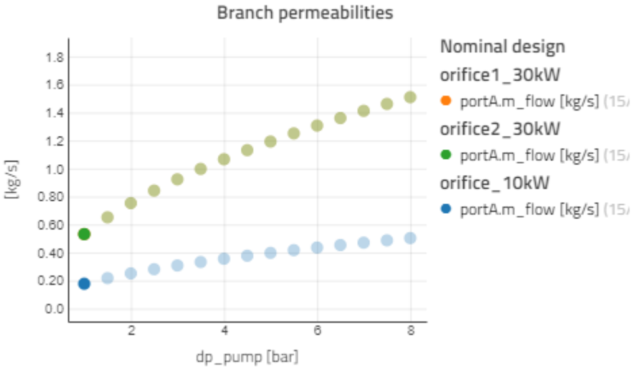


Figure 8. Off-design mass flow vs. pressure loss characteristics.

4.4 Further Requirement Verifications

First, the sizing point may not be defined as a deterministic single point but via ranges of expected heat loads, or it is defined by a deterministic single point but the engineer is interested in studying the impact of increasing or reducing the margin for thermal component performance (i.e., extract the same heat load at lower mass flow rate and increased temperature difference between coolant outlet and inlet). It is thus possible to perform multiple executions of the sizing simulation, covering the appropriate domain, and extracting the resulting ranges of calibrations. Modelon Impact provides dedicated functionalities for design of experiments – from a simple *choices* and *range* operators that enable defining a set of values, to more involved functionalities such as *Latin Hypercube Sampling*. Designing the orifices might thus involve several simulations and the selection of the most constraining design. This is simplified with steady-state simulation as the results are single values that can be easily compared and post-processed.

In the following, we assume that the orifice 2 design factor is swept between 37.5 kW/(kg/s) and 25 kW/(kg/s), i.e., $m_{flow,0}$ for *orifice2_30kW* is set to 0.8 kg/s to 1.2 kg/s. All other parameters are held constant. Orifice dimensions and resulting temperatures can be computed, and the latter are shown in **Figure 9**. The nominal design is highlighted.

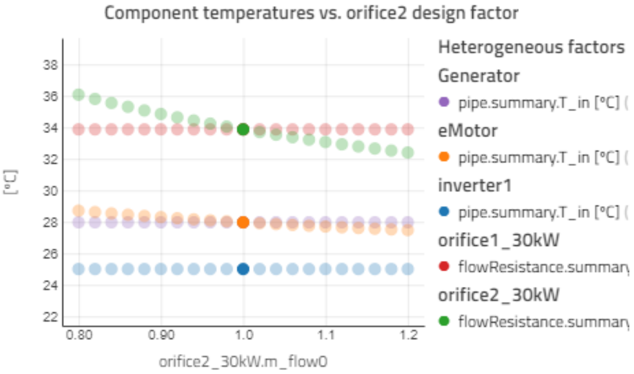


Figure 9. Temperature sensitivities over branch design factor.

Obviously, the thermodynamic model must always be satisfied. Based on the pressure loss characteristics of the branch components, a maximum mass flow rate of each branch must not be exceeded. If this point was reached, it would not be possible anymore to reduce the restriction in the orifice for calibration as the orifice diameter had already reached the pipe diameter. In the given problem, this occurs for single digit design factors and correspondingly high mass flow rate through *orifice2*; see **Figure 10**. In another network with more restriction, this can occur more easily.

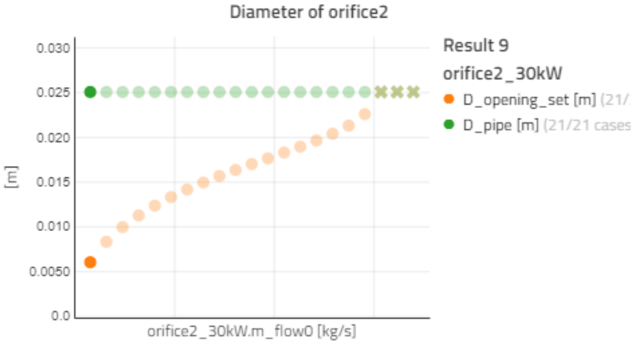


Figure 10. Diameter sensitivities over branch design factor.

Second, while there are heat load requirements on each component, there are also temperature limit constraints to satisfy. It is a good practice to simulate several points in the operational domain to ensure that the constraints are met – and that the requirements are correctly defined. For these simulations, the design of the orifices shall be set by a parameter and the focus would be on flow simulations. As illustrated for a single component in **Figure 5**, this does not require a new model to be developed but simply switching the *sizing* parameters to *false* and setting the diameter values.

Finally, Modelon Impact comes with a Python client (Modelon, 2022-c) that can conveniently define experiments, simulate and return results. System simulation and analysis to verify requirements can thus easily be automated, in a fully integrated manner.

5 Conclusion and Perspectives

This publication shows how the design of hybrid electric aircraft can be simplified with an efficient workflow for solving the flow balancing problem of the thermal management system. Modelon Impact enables automated steady-state solving and requirement verifications. Its openness makes it easy to be integrated in a system-level design loop that also involves the electrical system sizing.

The steady-state and flow balancing capabilities have been developed and tested on larger customer models including more than 300 individual components. The example discussed here has an order of magnitude fewer components and serves the purpose of illustrating the key workflow and modeling principles.

The flow balancing workflow can also be used more widely outside the thermal management system context. Whenever the distribution of fluid is of interest, restrictions can be calibrated to yield the desired split. This methodology is equally applicable to aircraft air distribution ducting from mixer to riser ducts and cabin air outlets, building air supply networks and so on.

While this proved the tool capability, the system model can be further refined to include the closed loop of the cooling system and design-specific cooling devices such as cold plates. A full workflow involving the Jet Propulsion and Thermal Management System – both running in steady-state, as embedded capabilities – would be a next step of this work.

References

Budinger Marc, Aurélien Reysset, Aitor Ochotorena and Scott Delbecq (2020). “Scaling laws and similarity models for the preliminary design of multirotor drones”. In *Aerospace Science and Technology*, 98. 1-15. ISSN 1270-9638.

Coïc Clément, Johan Andreasson, Anand Pitchaikani, Johan Åkesson and Hemanth Sattenapalli (2020). “Collaborative Development and Simulation of an Aircraft Hydraulic Actuator Model”. In *Asian Modelica Conference 2020*, Tokyo, Japan.

Coïc Clément, Moritz Hübel and Matthis Thorade (2020). “Enhanced Steady-State in Modelon Jet Propulsion Library, an Enabler for Industrial Design Workflows”. In *American Modelica Conference 2020*, Boulder, Colorado, USA.

Coïc Clément, Marc Budinger and Scott Delbecq (2022). “Multirotor drone sizing and trajectory optimization within Modelon Impact”. In *American Modelica Conference 2022*, Dallas, Texas, USA.

Delbecq Scott, Marc Budinger, Clément Coïc and Nathalie Bartoli (2021). “Trajectory and design optimization of multirotor drones with system simulation”. In *American Institute of Aeronautics and Astronautics, Inc, SciTech*, DOI: 10.2514/6.2021-0211.

Gkoutzamanis Vasilis G., Spyros E. Tsentis, Orestis S. Valsamis Mylonas, Anestis I. Kalfas, Konstantinos G. Kyprianidis, Panagiotis Tsirikoglou and Michael Sielemann (2020). “Thermal Management System Considerations for a Hybrid-Electric Commuter Aircraft”. In *Journal of Thermophysics and Heat Transfer*, by the American Institute of Aeronautics and Astronautics, Inc, 2020.

Hecken Tobias, Xin Zhao, Michael Iwanizki, Max J. Arzberger, Daniel Silberhorn, Martin Plohr, Konstantinos Kyprianidis, Smruti Sahoo, Giorgio Valente, Sharmila Sumsurooah, Michael Sielemann, Clément Coïc, Andreas Bardenhagen, Annika Scheunemann and Claire Jacobs (2020). “Conceptual Design Studies of “Boosted Turbofan” Configuration for short range”. In *AIAA SciTech Forum*, by the American Institute of Aeronautics and Astronautics, Inc, 2020.

Modelon website – Impact, Lowering barriers and bridging gaps, accessed on July 2022, https://www.modelon.com/modelon-impact-introduction/

Modelon website – Steady State and Dynamic Simulation: What is the difference?, accessed on July 2022, https://modelon.com/steady-state-and-dynamic-simulation-what-is-the-difference/

Modelon Help Center website – Modelon Impact Client, accessed on August 2022, https://help.modelon.com/latest/tutorials/jupyter_mi_client_new/?h=client

Zhao Xin, Smruti Sahoo, Konstantinos Kyprianidis, Jonathan Rantzer and Michael Sielemann (2019). “Off-design performance analysis of hybridised aircraft gas turbine”. In *The Aeronautical Journal*, by Royal Aeronautical Society, Cambridge University Press, 2019.