Energy Conversion and Management: X 29 (2026) 101410

Contents lists available at ScienceDirect

Energy Conversion and Management: X

journal homepage: www.sciencedirect.com/journal/energy-conversion-and-management-x

ProCoSim — A simulation tool for networks of energy facilities

Marlene Herbrik *, Martin Henke, Jan Zanger

German Aerospace Center, Institute of Combustion Technology, Pfaffenwaldring 38-40, Stuttgart, 70569, Germany

ARTICLE INFO

Keywords:
Simulation
Framework
Energy
Network
Modular
0-D
Decentralized
Hybrid-electric

ABSTRACT

The energy transition is making energy supply systems more complex. Well-known sources such as gas turbines are being combined with renewables, batteries, heat pumps, and further innovative technologies. Flexibility and resilience demand rise, as energy sectors are being coupled and new energy carriers like hydrogen are integrated. The engineering challenge is often to design all interactions to be efficient, reliable, and resilient. For that, simulative system designs are particularly valuable, as experiments are often not viable or would require strong simplifications.

To this end, a large variety of simulation frameworks are available. While there are many advantages to using existing frameworks, building a new framework allows for maximal customization and flexibility. The DLR Institute of Combustion Technology decided to develop its own in-house framework, named ProCoSim, for a variety of reasons. These include the modular integration of ODE and algebraic solvers, explicit and flexible handling of the interactions between component calculations and solvers, and the integration of an acausal component modeling approach with bidirectional signal flow to allow maximal flexibility in the integration of black-box component models. The aim of the framework is to support the design of energy systems consisting of producers and consumers, to analyze their interactions, to optimize operation strategies, and to provide boundary conditions for experiments, e.g. load trajectories for a gas turbine test emulating demands of a virtual energy system.

In this paper, the design and implementation process of the new framework are described in detail. The flexibility of its application is shown by two distinct demonstration cases: a thermodynamic system from the field of decentralized energy supply and a mechanical–electrical system in which a hybrid-electric aircraft propulsion system is modeled.

1. Introduction

1.1. Motivation

The energy transition requires efficient solutions for the energy supply. Proven concepts are combined with new technologies shown in the 7th Energy Research Program issued by the German Government [1]. With these complex supply systems, it is more difficult to test the designs experimentally. It would be impractical, expensive, and risky to fully test a new aircraft propulsion system consisting of a gas turbine and a battery immediately. Often test systems can only be built at a simplified level, as for the combustion of the new energy source hydrogen in gas turbines. In addition to the overall efficiency, the subsystems must prove themselves under real conditions.

We developed ProCoSim to best describe the level of the overall system. The simulator makes it possible to find the most efficient operating mode for a system and to optimize the design of both the system and its components. This is done by repeated simulations with parameter variation. Preliminary simulations before experiments reduce risk and therefore costs. For simplified test rigs, the simulator describes the environment that the component under test will experience in the actual application and provides the boundary conditions for the experiment.

1.2. Relation to other simulation frameworks

The field of energy management and energy system simulation is very active, with a large variety of open- and closed-source programs with vastly different capabilities, scopes, usability, and flexibility goals, as well as foci at different granularity levels. While some modeling frameworks are used for a very wide variety of domains, like Simulink [2] and Modelica [3], Industry and academia often develop in-house solutions that are tailored to their specific needs, like aviation system engineering [4] and can span several granularity levels of their modeling tasks [5]. Many specialized commercial software frameworks exist, that focus on certain systems engineering tasks like Building

* Corresponding author.
E-mail address: Marlene.Herbrik@dlr.de (M. Herbrik).

https://doi.org/10.1016/j.ecmx.2025.101410
Received 27 August 2025; Received in revised form 14 November 2025; Accepted 18 November 2025
Available online 19 November 2025

Nomenclature

Abbreviations

SOC	state of charge
AC	alternating current
ALG	algebraic
CHP	combined heat and power
DC	direct current
DLR	German Aerospace Center
INT	internal calculations
LUT	look-up table
MGT	micro gas turbine
MGT-CHP	micro gas turbine-based combined heat and power plant
MGTS3	MGT Steady State Simulator
ODE	ordinary differential equation
TPDE	Technology Platform Decentralized Energies

Symbols

$\dot{m}$	mass flow
η	efficiency
Φ	electrical potential
C	battery capacity
c_p	specific heat capacity
E	stored battery energy
f	humidity
H	enthalpy
h	height
I	electrical current
k	heat loss rate
N	rotation speed
P	power
p	pressure
Q	heat
T	temperature
t	time
U	voltage

Superscripts

env	environmental
in	in-flow
int	internal
max	maximum
min	minimum
out	out-flow

Subscripts

bat	battery
el	electrical
gen	generator
gt	gas turbine
inv	inverter
load	Mechanical Load
mech	mechanical
orig	original
pack	battery pack
scal	scaled
therm	thermal

energy and thermal systems [6] or Thermo-hydraulic and Mechatronic systems [7]. They simplify usability with detailed application libraries and predefined models. While some offer a variety of fixed pre-defined application cases that merely require parameterization [8], many are flexible and extensible to span the bridge from acquiring fast results for standard applications to scientific research on novel concepts [9]. Some limit deliberately model complexity to solve distinct problems like large scale linear mixed integer optimization of vast energy networks, like e.g. the open-source ecosystem of omoef.solph [10] and MTRESS [11]. Additionally, frameworks from different categories are often combined to achieve complex analysis and optimization tasks [12].

While this paper is not intended to give a review on these tools or a conclusive comparison between frameworks, some relations to Simulink and Modelica are pointed out in the following, as these frameworks are widely adopted and using them as context allows for a rough positioning of ProCoSim in the vast simulation framework landscape.

Simulink is a closed-source, commercial and very flexible framework with many applications including energy system analysis. It is extensively used also in the authors' department for thermodynamic process simulation [13] and formerly for energy system simulation. Without additional packages (e.g. Simscape), it is limited to causal modeling. This comes with benefits as certain components can be modeled intuitively with usual procedural programming techniques, including dynamic, event-driven structures of complex models. But the drawbacks are that inputs and outputs of a component and hence the direction of signal flow is fixed. Thus, modeling components like storages, hydraulic separators or electrical elements can be quite challenging.

Modelica, on the other hand, is a modeling language which is supported by several commercial and open-source frameworks (e.g. OpenModelica) and is tailored for acausal modeling. Modelica uses a rather complex toolchain to gather the system equations, apply a set of optimization techniques before further compiling and solving the resulting DAEs, see e.g. [14]. While this very integrated behavior clearly has performance benefits, the inner functionality, such as the interfaces between components and the solvers, become more complex and harder to modify. Additionally, dynamic changes in topology or component equations are harder to implement, due to the optimizations and handling of the equations.

The presented approach tries to combine the causal, dynamic modeling of Simulink with other elements like e.g. bidirectional signal flows and a simple modular interconnection to solvers, which is described in 2. The approach is focused on the analysis of the interactions between heterogeneous, nonlinear system components that can be integrated in a black-box manner to simplify integration and co-simulation of newly added components.

1.3. Concept

ProCoSim is designed to describe complex energy networks of various types at the top level with high performance, see Fig. 1. The system is represented as a network of components, e.g. power plants, loads, storages or connections. We study the time evolution of the component quantities and, depending on the application, their equilibrium states. Beyond that, there is no spatial resolution. We model the components in a flexible way using deposited equations or look-up table data, see Section 2.2. The data comes from experiments or detailed calculations from third-party simulators, in-house or from our partners.

ProCoSim might be applied to a heating circuit as shown in Section 3.1 with the transferred energy being thermal, a hybrid-electric aircraft propulsion system shown in Section 3.2 with mechanical and electrical energy, or any other energy network system.

ProCoSim provides a global overview and complements detailed experimental and numerical studies on specific components. The focus at the DLR Institute of Combustion Technology, for example, is on gas turbines, especially micro gas turbines. We examine them and their components in detail, both experimentally and numerically, and develop new circuits or burners, for example [15]. ProCoSim maps the environment and the higher level for these studies as shown in Fig. 1.

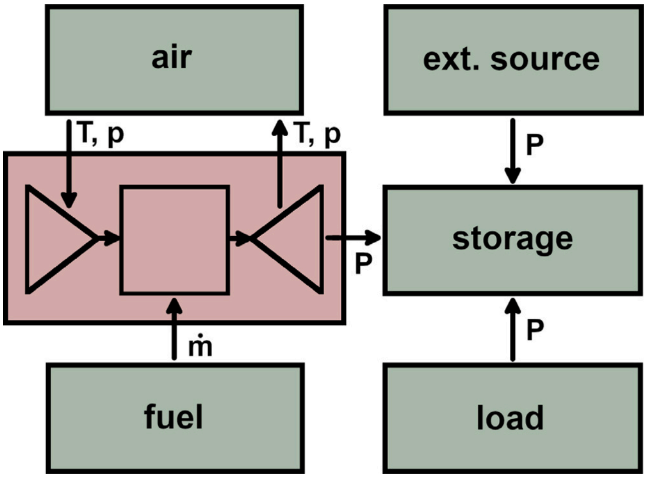


Fig. 1. The research focus gas turbine described in detail (red) and the higher level with the environment, load, storage and other components modeled by ProCoSim (green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

1.4. Contribution

This paper first shows how we proved the concept and qualitative functionality of ProCoSim in a thermodynamic system. After that, we demonstrate the flexibility of the simulator applying it to a hybrid-electric system where we prove its quantitative predictions.

We show that ProCoSim can help to design system components, optimize networks and operation modes as well as accompany experiments.

2. ProCoSim functionality

2.1. Overview

ProCoSim performs a simulation by calculating the system quantities of all components at each desired time step. In general, a system is time-dependent and subject to algebraic constraints.

First, we *model* the components and their interfaces in the *network* in Section 2.2. The time evolution of the system follows from *ordinary differential equations* (ODE) which ProCoSim solves approximately. For each time step, it determines the *algebraic equilibrium* (ALG) and for each algebraic iteration step, it runs a chain of *component-internal calculations* (INT), explained in Section 2.3, 2.4 and 2.5. The detailed simulation procedure can be found in Section 2.6.

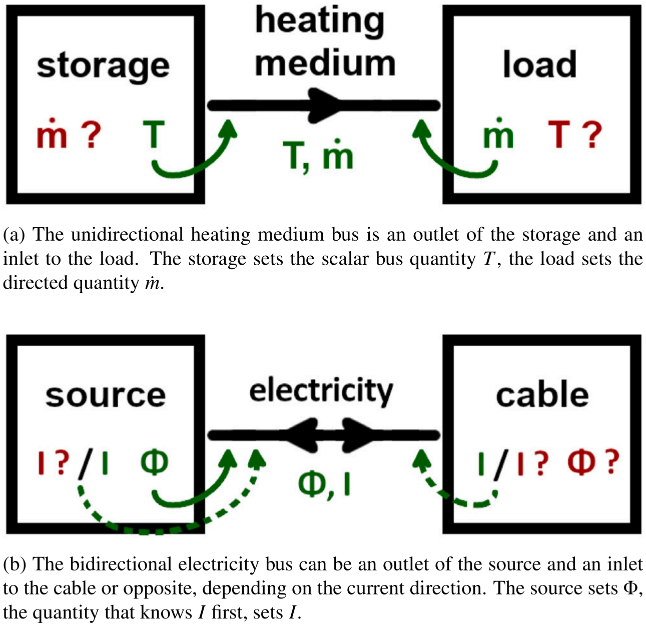
The user defines the system via an *input* file. Finally, ProCoSim outputs the result in an *output* file and output plots, which we discuss in Sections 2.6 and 2.7.

2.2. Network modeling

ProCoSim models the physical system as a network of components and buses, for an example see Fig. 7.

Each system unit is represented by a component in the model: e.g. a micro gas turbine-based combined heat and power plant (MGT-CHP), a heat storage tank or a heating water pipe. There are also mathematical and programming auxiliary components: The boundary condition component, for example, maps the interface of the system to its environment. The modeling determines the component's purpose and how accurately it is described. During the simulation, each component has the task of calculating certain system variables.

The interfaces between the components are called buses. They have two tasks. Firstly, they transfer the system variables between the components. A bus may represent an actual physical flow like a heating water flow. Or they model the flow of information from a mathematical auxiliary component, for example a shutdown signal. The bus quantities can be scalar values or directed values. Buses with directed quantities can be unidirectional, i.e. with a fixed expected flow direction. Opposite flows are then defined by a negative sign. Flows that switch direction often are described by bidirectional buses. They allow flows in both directions, marked by a flag variable. Secondly, buses carry the information whether their variables are set or not. That is, whether the values of the quantities are known at the current time. This is important because the system variables are constantly recalculated during a simulation. At any point in time, there are generally some quantities that are set and some that are not.



(a) The unidirectional heating medium bus is an outlet of the storage and an inlet to the load. The storage sets the scalar bus quantity T, the load sets the directed quantity $\dot{m}$.

(b) The bidirectional electricity bus can be an outlet of the source and an inlet to the cable or opposite, depending on the current direction. The source sets Φ, the quantity that knows I first, sets I.

Fig. 2. A bus between two components with set (green) and unset (red) quantities. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The interplay between components and buses is sketched in Fig. 2.

We have created an extensible library of components and buses. New components and buses can often be derived from existing ones.

To map the system, the user selects the desired components and buses from the library and specifies how they are interconnected. In general, each component has connected buses. We refer to a connected bus as a calculation inlet of the component if it provides the component with set values as a basis for calculation. The values in a calculation outlet bus, on the other hand, are set by the component itself. Each bus is the outlet of one component and at the same time the inlet to another. The interconnection of the components and buses is fixed for the duration of a simulation. Whether a bus acts as an inlet or outlet to a component can change during the course of the simulation. The flow of the physical variable can be identical or opposite to the calculation sequence, i.e. to the flow of the set variables, also sketched in Fig. 2.

2.3. Component-internal calculations

Each component must set the values of its outlets based on its inlet values, illustrated in yellow in Fig. 3. The calculation rule for this is stored in the component according to its modeling. Some components have to solve simple algebraic relations, an example follows in Fig. 8. Others have to fetch stored detailed data. In the latter case, this data comes from experiments or previous simulations. It is

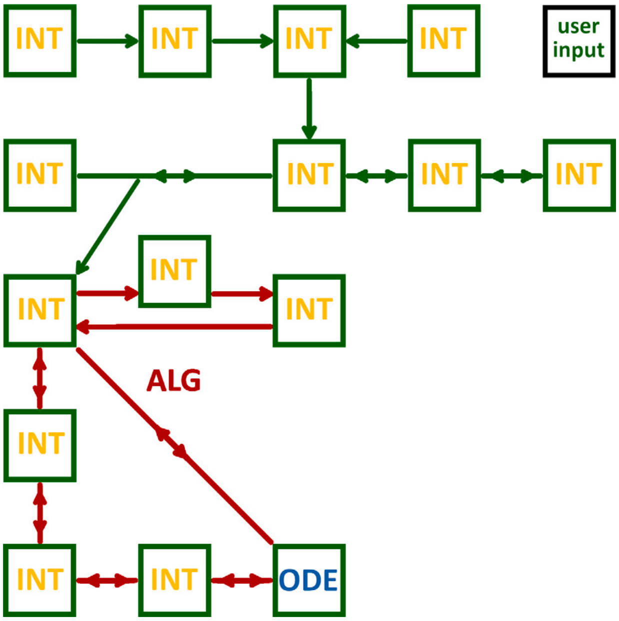


Fig. 3. The internal calculations of each component (yellow); the algebraic constraints between multiple components (red); the time evolution of certain components described by differential equations (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

deposited as a look-up table (LUT) for ProCoSim to read. This procedure saves computing time compared to actual co-simulations and is sufficiently accurate. The data in the look-up table are given for discrete sampling points and are interpolated by ProCoSim using multilinear interpolation.

A Calc-Internals Manager coordinates the internal calculations of all components since they are interdependent in the calculation process via their inlets and outlets. A specific component can only perform its calculation fully or partly if it knows the necessary inlet values for the current state. If the system is designed to be solvable, all components successively complete their internal calculations, starting from the specified boundary conditions. This is illustrated in yellow in Fig. 4.

The Calc-Internals Manager controls this calculation calling the components one by one, while they successively perform their calculations as far as possible. The Calc Internals Manager heuristically improves the call sequence to better select ready components: First, it selects a reasonable start component carrying a respective flag. For example, boundary conditions do not require any external input to start their calculation. Then, the Calc Internals Manager tries random call sequences and remembers the shortest one to use for the remaining calculation cycles.

Once all components have completed their calculations and the system values have been set accordingly, the system is fully defined for the current set of boundary conditions.

2.4. Algebraic constraints

Some systems of interconnected components are subject to a constraint, i.e. an algebraic relationship. It leads to a state of equilibrium for a particular quantity. An example will be shown in Fig. 14. The constraint occurs in addition to the internal calculations of each individual component from Section 2.3, and it involves multiple components as shown in red in Fig. 3. Therefore, a single calculation sequence, as described above, cannot determine this state of equilibrium.

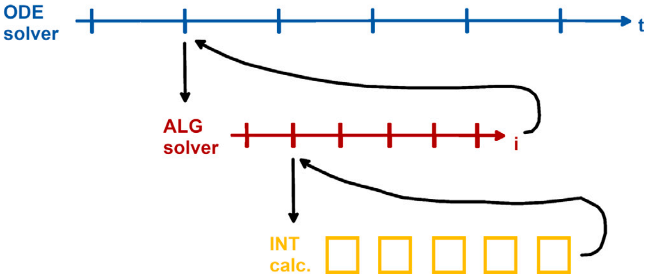


Fig. 4. The simulation procedure: The manager objects coordinate the necessary computation loops. For each ODE time step (blue) an algebraic calculation loop (red), for each ALG iteration step an internal calculations cycle (yellow). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Instead, ProCoSim iteratively determines the equilibrium value of the relevant system quantity using the network of components: Based on a starting value for the system quantity, ProCoSim performs the chain of internal calculations for all components. Once the relevant quantity can be determined from the set values, it will be compared: If the calculated value is equal to the guessed starting value, the equilibrium state has been found. In general, the two values are initially different. A so-called residual component in the system determines the deviation of the two values, the residual. ProCoSim passes the residual to an algebraic solver. The solver returns a corrected initial value for the quantity to ProCoSim. This new value is used to calculate the system in a new iteration step. These steps are repeated until the residual meets a defined convergence criterion. The algebraic calculation loop is illustrated in red in Fig. 4.

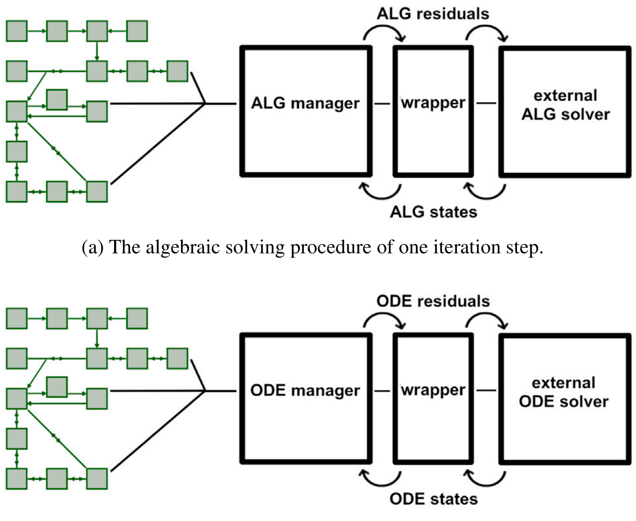
The coordination of the algebraic calculation cycles is done by the ALG manager. At each iteration step, it ensures that a start value is available and calls the Calc Internals manager to trigger the internal calculation chain. It then passes the residual to the solver, see Fig. 5(a). When the residual converges, the ALG manager terminates the algebraic calculation cycle.

We integrated the external algebraic solver Ceres in version 2.1.0 from Agarwal et al. [16] and use the default solver for constrained nonlinear optimization problems. The convergence criterion was set to a minimal relative residuum change of 1e-10 and a minimum relative parameter change of 1e-12. ProCoSim addresses the solver via a self-written wrapper. This gives us the flexibility to exchange the solver, if necessary, without affecting ProCoSim.

2.5. Ordinary differential equations

The energy network under consideration usually evolves over time. This is the case if the internal calculations of at least one component are time-dependent: i.e. an ordinary differential equation, an algebraic equation with time dependence, or a look-up table with time-dependent values. Fig. 8 shows the thermal storage as an example of an ODE active component. In the last two cases, the time evolution is determined before the simulation is started. In the case of differential equations, the evolution for the next time interval results from the current state. This means that the differential equation must be solved numerically during the simulation, compare Fig. 3.

ProCoSim solves the system of differential equations approximately for discrete time steps and with the help of the component network: For each ODE-active quantity, the user defines an initial value at the beginning of the simulation. For this first time step, ProCoSim performs the chain of algebraic and internal calculations for all components. As soon as possible, the ODE-active component calculates the temporal gradient of the quantity using the stored differential equation. ProCoSim passes this gradient to an ODE solver. The solver returns the value of the



(a) The algebraic solving procedure of one iteration step.

(b) The ODE-solving procedure for one time step.

Fig. 5.

system quantity in the next time step to ProCoSim. This new value is used to calculate the system for the next time step. These steps are repeated until the system is calculated for all required time steps. The ODE calculation is illustrated in blue in Fig. 4

The coordination of the ODE calculation cycles is done by the ODE manager. At each time step, it ensures that an initial value is available and calls the ALG manager or the Calc-Internals Manager to trigger the calculation chain. It then passes the calculated gradients to the ODE solver, see Fig. 5(b). When the last simulation time step is reached, the ODE manager ends the ODE calculation cycle.

We have integrated the external ODE-solver Odeint from the Boost library Dawes et al. [17] (Version 1.81.0). ProCoSim addresses the solver via a self-written wrapper. This gives us the flexibility to exchange the solver, if necessary, without affecting ProCoSim.

For both application cases, the second order, explicit, multistep, Adams Bashforth method was used. The explicit method showed no signs of numeric instability for the given time step sizes. Switching to implicit methods is not supported. Given the black-box design chosen for easier component modularity (see 1.2) an exact Jacobian of the system is not given. While it might be created via numerical approximation, such an approach is not implemented.

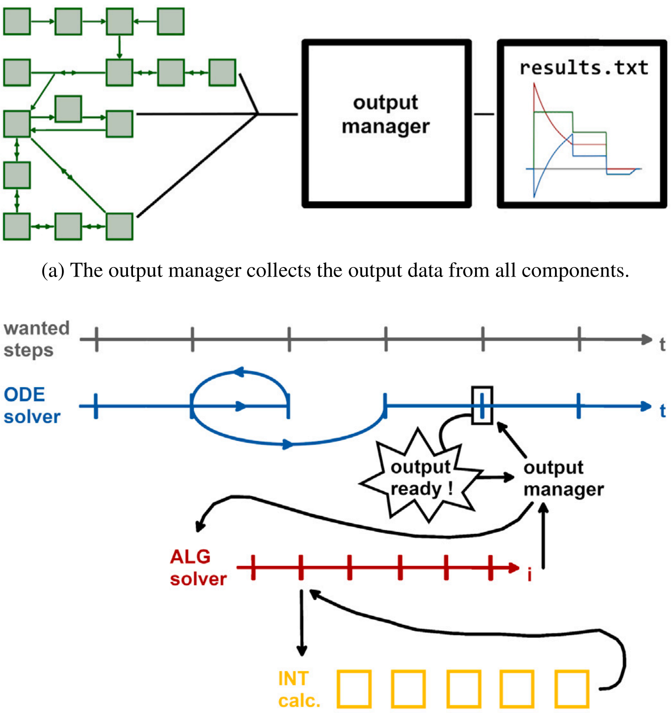
2.6. Simulation

To perform a simulation on a time-dependent system with algebraic constraints, ProCoSim runs the entire procedure of ODE, ALG and component-internal calculations shown in Fig. 4. Besides this, ProCoSim can handle the special cases: the desired time evolution without algebraic constraints or the desired algebraic equilibrium without time evolution. In this case, ProCoSim skips the ALG or ODE solution loop accordingly.

The simulation process is controlled by the Simulation Manager. It checks which type of system is present and triggers the ODE, ALG and Calc-Internals Manager accordingly.

2.7. Simulation input and output

The user defines the system they want to simulate modularly from the existing library components and buses in an input file. The user is responsible for a reasonable interconnection and that the system is solvable, i.e. neither over- nor under-determined. For time-dependent systems, the user specifies the simulation period and the desired time resolution. For the components they specify characteristic values, constants, paths to look-up tables, initial conditions and, if necessary, starting values for algebraic solution loops.



(a) The output manager collects the output data from all components.

(b) The output manager is triggered when the system is fully computed for a required time step.

Fig. 6. The output manager.

ProCoSim extracts the information from the input file, builds the model system accordingly and defines the required manager objects.

After the simulation the user receives the result in an output file. It contains the calculated values of all components and buses for all required time steps.

The output of the results is coordinated by the Output Manager, Collector and Writer. During the simulation, they work in parallel with the calculation managers Calc-Internals, ALG and ODE manager and collect the output-ready values. Compare Fig. 6. A value is output-ready if the current calculated state corresponds to a real state in the simulation process. If this is the case, the Output Collector collects the values of all components and buses. The Output Writer writes them collectively to the output file at the end of the simulation.

This procedure is necessary because the values of the system quantities are continuously recalculated and overwritten during the simulation. This improves performance, but the values often have no physical meaning, for example, until an algebraic iteration has converged.

The output format is HDF5 from The H.D.F. Group [18]. This can be adapted if necessary, since the HDF5 library is addressed via a wrapper.

The self-written tool ProCoVis plots the quantities of the output file using Matlab [19]. The generated plots are tailored to the model system: e.g. a comparison plot of the same quantity in several components over time or a plot collection that shows a set of related quantities. MATLAB was used primarily due to the authors' prior experience with it and its availability at the institute.

2.8. Software development

ProCoSim is programmed in C++. The components, buses, computation managers, wrappers etc. of ProCoSim are implemented as objects for modularity and code flexibility. For closely related components with different modeling, we use abstract component classes, e.g. for the Thermal Storage from Section 3.1.4. Concrete subclasses with the desired modeling are derived from the abstract class: For example, a thermal storage with a layer model for the temperatures or a simple, fully mixed water tank.

ProCoSim uses the C++ standard library [20] and some valuable extensions from the Boost library [17].

ProCoSim does not use or require performance optimization methods as it only performs time efficient 0D calculations. The computation time for a typical simulation is in the range of seconds to minutes as we show in Section 3.

The chosen third party solvers determine the accuracy, robustness and solver stability. In our case, Ceres from Agarwal et al. [16] and Odeint from the Boost library [17].

3. Applications

The following application cases demonstrate that the simulator can model distinct systems. They are examples for the intended application spectrum of ProCoSim. First, we qualitatively verify the functionality using an example from the *energy supply sector* with an overview of the most important system quantities. Then we prove the modularity and flexibility of the tool by applying it to *modern aviation with hybrid-electric concepts*. There, we show examples of quantitative results and thus the variety of investigations that are possible with ProCoSim.

The following application chapters are structured as follows: First, we describe the physical system we want to model and its operating modes. The problem section shows typical questions we plan to investigate with ProCoSim for these systems. The modeling section explains how we designed the component models for the ProCoSim library that are needed for the applications: First, we set up a general component model. It may be applied to any system or interconnection containing that component type. Then, we apply it to model the specific component used, with parameters and detail data. Our focus is on creating a modular component library, without restricting ourselves to certain application cases. We then define the test scenario, i.e. the time evolution for the application system, simulate it and show the results.

3.1. Application 1: Decentralized energy supply

3.1.1. System

The first use case is a heat supply and demand network. Combined heat and power (CHP) plants supply heating water to buildings and other loads in the heating circuit. A thermal storage tank acts as a buffer in case of demand fluctuations.

This system was built by the DLR Institute of Combustion Technology at the DLR site in Stuttgart [21]. It is shown in Fig. 7. Here, a laboratory with micro gas turbine (MGT) test rigs was connected to the site supply. The laboratory feeds heat and electricity into the site. It is called Technology Platform Decentralized Energies (TPDE). We focus on the heat supply as an interesting use case for ProCoSim. The sources of the system are a micro gas turbine-based combined heat and power plant (MGT-CHP) and an external cogeneration plant. The MGT-CHP is a cluster of one or more CHP units, representing the number of active test rigs in the laboratory. The required heat is provided jointly by the CHP units. For our simple test case, we assume only one MGT-CHP, specified in Section 3.1.4. The loads are 5 buildings with heating systems, an absorption chiller, and a recooling plant. A control system monitors the fill level of the thermal storage and ramps the heat-driven MGT-CHP units up or down accordingly. The integrated network generally consists of branched heating water pipes and control signals.

Note: We might apply ProCoSim to the power supply just as well. However, this is not a complex problem where we could test ProCoSim extensively. The power demand of the site is always higher than the power provided by the laboratory. Therefore, the latter can be fed into the site completely at any time. The connection to the external power grid is still given and supplies what is missing. Applying ProCoSim to the heat supply offers more interesting issues.

3.1.2. Operating modes

The TPDE system allows two modes of operation.

In the *machine test mode*, experiments are carried out on the MGT test rigs. The waste heat can be efficiently used to supply the site instead of being dissipated as in comparable laboratories. Depending on the current demand and the season, the heat can be used directly to supply the buildings or indirectly via the heat storage tank.

In the *interconnected operation mode*, the site can serve as a test platform for MGT-CHPs, consisting of multiple units. For this purpose, we set the consumption curve of a virtual thermal load. The MGT-CHP units on the test rigs work together as one joint CHP plant to meet the demand. We realize the virtual load via the connected recooling plant and the external cogeneration plant. These components allow us to dissipate or supply the desired amount of heat at any time. In this way, we can overlay the actual demand of the site with the virtual desired consumption curve.

In both cases, the heat supply to the actual site is permanently guaranteed through the external cogeneration plant.

3.1.3. Problem

In a heat supply network such as the TPDE system, the following questions are expected depending on the operating status.

For *machine test operation*, we are interested in: How can we efficiently use as much waste heat as possible from the test rigs on site? By distributing the heat to the suitable loads; by synchronizing the experiment schedule with consumption peaks; by designing the heat storage tank appropriately; or by adapting the connections?

In the *interconnected operation mode*, we want to investigate: How should the MGT-CHP units be controlled to most efficiently meet load demand and minimize start-up and shut-off processes that reduce the service life of the gas turbines? What heat storage volume and insulation is needed to smooth out fluctuations in heat supply and consumption?

3.1.4. Modeling

We model the TPDE system (Fig. 7 top) as a network of components and buses (Fig. 7 bottom), as described in Section 2.2.

The following paragraphs explain how we model each component and bus in detail.

External cogeneration plant. We design the External CHP as a simple black box because it is not the focus of our research. The CHP heats the heating water to the desired temperature T, which it receives via a signal bus:

$$T^{\mathrm{out}} = T^{\mathrm{wish}}. \tag{1}$$

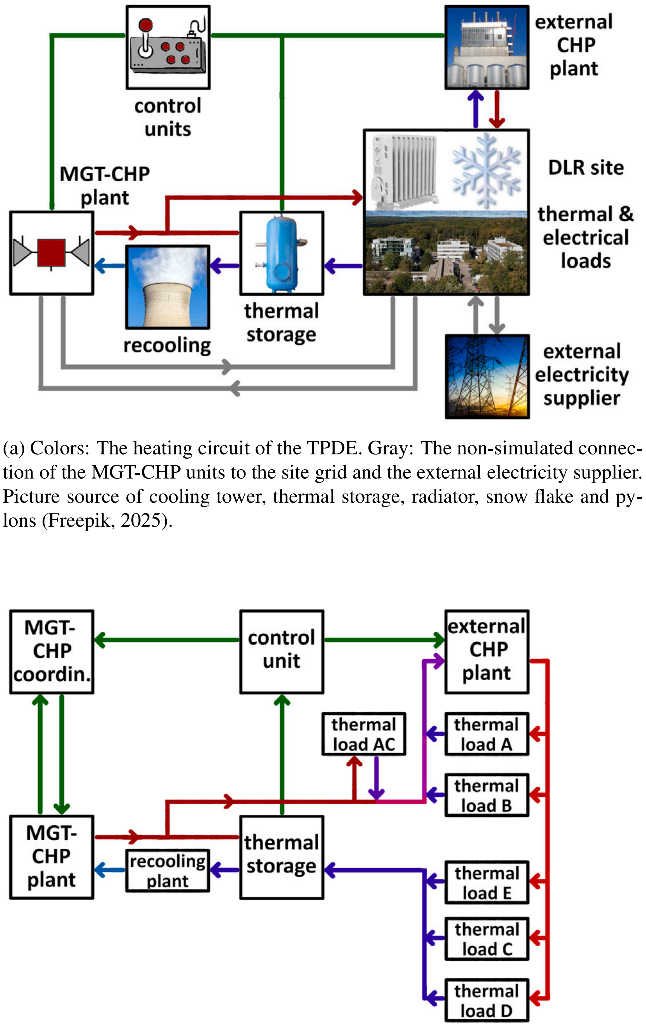
It sets the mass flow $\dot{m}$ of the heating water according to mass conservation, either in the inflow (i.e. the return to the CHP) or in the outflow (i.e. the supply):

$$\dot{m}^{\mathrm{out}} = \dot{m}^{\mathrm{in}} \qquad \text{or} \tag{2a}$$

$$\dot{m}^{\mathrm{in}} = \dot{m}^{\mathrm{out}}. \tag{2b}$$

MGT-CHP plant. The component MGT-CHP describes a micro gas turbine test rig. It acts as a power plant and heats the heating water. It is the central component for our research interest. Therefore, we describe it with stored detail data. The data stems from previous simulations and experiments of the gas turbine used. We store the data in look-up tables: One for the stationary operation of the CHP unit and one for dynamic load changes.

The MGT-CHP unit receives a heating water inflow with T^{in}, a power demand P_{therm} and the environmental conditions air temperature T^{env}, pressure p^{env} and humidity f^{env} via a signal bus. The MGT-CHP unit checks what needs to be done to meet the demand. If the current operating state is suitable, we describe the component as



(a) Colors: The heating circuit of the TPDE. Gray: The non-simulated connection of the MGT-CHP units to the site grid and the external electricity supplier. Picture source of cooling tower, thermal storage, radiator, snow flake and pylons (Freepik, 2025).

(b) The TPDE system modeled as a network of components and buses with one MGT-CHP unit. The loads A to E correspond to buildings, the load AC to the absorption chiller. Not shown for simplicity are heating water pipes at all heating water bifurcations and boundary conditions at the MGT-CHP unit and at all loads. The connection of the components is defined by the local conditions.

Fig. 7. The Technology Platform Decentralized Energies (TPDE). Signal buses are shown in green, hot heating medium buses in red and colder heating medium buses in blue [22]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

stationary. Load and rotation speed are constant. The MGT-CHP reads from the stationary LUT

$$T^{\text{out}} = T^{\text{out}}\left(T^{\text{in}}, P_{\text{therm}}, T^{\text{env}}\right). \tag{3}$$

If the current state does not meet the requirement, the MGT-CHP unit must start a maneuver, i.e. adjust load and rotation speed. The transient LUT provides

$$P_{\text{therm}} = P_{\text{therm}}(N, t) \tag{4}$$

and the stationary LUT still provides the dependency on T^{env}, T^{in} and T^{out}.

Furthermore, the model provides optional output parameters if requested. For example, it calculates the fuel mass flow that is required to achieve the current output power.

The steady-state look-up table data comes from validated simulations with MGTS3, our micro gas turbine simulator [23,24], and the transient data comes from interpolated experimental data [25]. We combine both tables based on our experience. In this example, we use data from the Ansaldo Turbec T100 micro gas turbine with a maximum thermal power output of approximately 200 kW [26]. We use multilinear interpolation to read out LUT data in between grid points, using no extrapolation.

MGT-CHP coordinator. The MGT-CHP Coordinator is a control component. It coordinates the individual MGT-CHP units on the test rigs in the laboratory, which together act as one micro gas turbine-based CHP unit. The MGT-CHP Coordinator receives the total power demand and distributes it to the individual MGT power plants i:

$$P^i_{\text{therm}} = P^i_{\text{therm}}\left(P^{\text{total}}_{\text{therm}}, X^j, P^{i,\text{min}}_{\text{therm}}, P^{i,\text{max}}_{\text{therm}}\right). \tag{5}$$

The distribution is based on the current operating states X^j of all MGT-CHPs and a stored control strategy. The coordinator communicates with all other components via signal buses.

The control strategy can be exchanged. For example, one could minimize the number of start/shut-off events of the MGT-CHP units to increase their lifetime, or try to utilize all power plants equally. In this way, we can study how different control strategies affect efficiency and demand coverage.

Our first test case shown below includes only one MGT-CHP unit. In this special case, the coordinator passes the entire power demand to the unit without splitting it.

Thermal load. The Thermal Load describes the heating system of a building. It extracts heat from the heating water. According to the temperature of the heating water in the inflow bus and its own demand profile $P_{\text{therm}}(t)$, it takes a certain amount of heating water. It then calculates the temperature at which it releases the heating water to the return:

$$\dot{m} = \dot{m}^{\text{in}} = \dot{m}^{\text{out}} = \dot{m}\left(P_{\text{therm}}(t), T^{\text{in}}\right) \tag{6a}$$

$$T^{\text{out}} = T^{\text{out}}\left(T^{\text{in}}, \dot{m}\right) \tag{6b}$$

We also describe the absorption chiller as a Thermal Load. It allows us to efficiently use the thermal energy of the test rigs, even during periods of low heat demand at the site. For that, it extracts heat from the heating circuit and converts it into cold. From the modeling point of view, it acts the same as the other thermal loads.

Recooling plant. We describe the Recooling Plant as a simple black box load, analogously to our approach in the paragraph *External Cogeneration Plant* above. It cools the heating water to a set temperature, which it receives via a signal bus. And it sets the unset mass flow according to mass conservation:

$$T^{\text{out}} = T^{\text{wish}}. \tag{7}$$

$$\dot{m}^{\text{out}} = \dot{m}^{\text{in}} \qquad \text{or} \tag{8a}$$

$$\dot{m}^{\text{in}} = \dot{m}^{\text{out}}. \tag{8b}$$

Converted power. All power plants and loads have an ODE for the total enthalpy transferred: $\dot{H} = c_p \dot{m}^{\text{in}} \left(T^{\text{out}} - T^{\text{in}}\right)$, where c_p is the specific heat capacity of the heating medium. The ODE adds up the power supplied or demanded during the simulation. In this way we calculate the total heat energy that the power plant or load has converted up to the current time.

Thermal storage. The component Thermal Storage describes a water tank. We use a layer model for its temperature distribution as described in [27]: At the top of the tank there is the hot inflow and the outflow that supplies the loads, at the bottom there is the cold return flow. The Thermal Storage receives heating water of a certain temperature in the inflow, compare Fig. 8. From this, it updates its internal temperature distribution in each layer l:

$$\dot{T}_l = \frac{1}{c_p m_l}\left[\dot{Q}^{\text{in}}_l + \dot{Q}^{\text{out}}_l + \dot{Q}^{\text{int}}_l\right], \tag{9a}$$

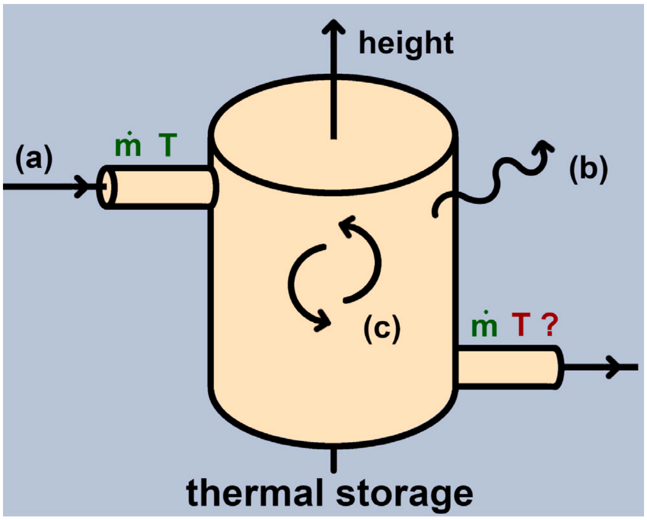


Fig. 8. The internal calculations (yellow) of the component Thermal Storage: It receives an inflow of given mass flow and temperature and a mass flow demand at the outflow. It calculates the outflow temperature from the ODE (10) (blue), considering: (a) inflows, (b) thermal losses and (c) its internal temperature distribution. Compare Figs. 3 and 4 for color code. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where m_l is the mass of the heating medium in layer l, c_p is the specific heat capacity of the heating medium and Q is the heat. It then sets the outflow temperature at height h and calculates the mass flows according to the conservation law:

$$T^{\text{out}}\big|_{h_l} = T_l. \tag{9b}$$

$$\sum_{i^{\text{in}}=1}^{I^{\text{in}}} \dot{m}_i^{\text{in}} = \sum_{j^{\text{out}}=1}^{J^{\text{out}}} \dot{m}_j^{\text{out}}. \tag{9c}$$

The temperature distribution (9a) is described by an ODE. It contains an inflow contribution $\dot{Q}_l^{\text{in}}$, the heat losses to the environment $\dot{Q}_l^{\text{out}}$ and an internal convective part $\dot{Q}_l^{\text{int}}$:

$$\dot{Q}_l^{\text{in}} = c_p \sum_{n=1}^{N} \dot{m}_{n\to l}^{\text{in}} \left(T_{n\to l}^{\text{in}} - T_l\right) \tag{10a}$$

$$\dot{Q}_l^{\text{out}} = k_{l\to\text{env}} \left(T^{\text{env}} - T_l\right). \tag{10b}$$

$$\dot{Q}_l^{\text{int}} = \begin{cases} c_p\, \dot{m}_l^{\text{int}} \left(T_{l+1} - T_l\right) & \text{for } \dot{m}_l^{\text{int}} > 0, \\ c_p\, \dot{m}_l^{\text{int}} \left(T_l - T_{l-1}\right) & \text{for } \dot{m}_l^{\text{int}} <= 0, \end{cases} \tag{10c}$$

where $k_{l\to\text{env}}$ is the heat loss rate from layer l to the environment. We choose $k_{l\to\text{env}} = 1\,\frac{\text{J}}{\text{Ks}}$.

Control unit. The control unit is an abstract component that monitors the interaction between the supply and consumption of thermal energy. It checks the temperatures in the thermal storage and sets the requirements for the MGT-CHP and the external CHP accordingly. If the storage tank gets too cold, the power plants have to supply more thermal power. The control unit holds a specific replaceable control strategy for that. The unit communicates with the storage tank and the power plants via signal buses.

Heating water pipe. This component transmits heating water. It can be linear or branched and allow for uni- or bi-directional heating water flow. The Heating Water Pipe sets $\dot{m}$ and T in those connected buses j where they are unknown: It calculates the mass flow based on continuity according to Eq. (9c) and the temperature based on thermodynamics according to

$$T_j^{\text{out}} = \frac{\sum_{i^{\text{in}}=1}^{I^{\text{in}}} \dot{m}_i^{\text{in}} T_i^{\text{in}}}{\sum_{i^{\text{in}}=1}^{I^{\text{in}}} \dot{m}_i^{\text{in}}}. \tag{11}$$

Boundary condition. The component Boundary Condition has a unique feature: it has no inlets, but sets its outlets according to fixed stored values. For one or more quantities, it has stored value pairs of discrete time steps and the corresponding quantity values. The user can specify these time-dependent data via an LUT or define them manually in the input file.

In this system, several boundary conditions describe these quantities: the demand $P_{\text{therm}}(t)$ of the thermal loads and the environmental conditions T^{env}, p^{env} and f^{env} of the MGT-CHP and the Thermal Storage.

Buses. The bus Heating Medium describes a heating medium transition, e.g. between a heating water pipe and the storage tank. The bus carries the two scalar variables mass flow and temperature of the heating medium. The user can store a database of heating media by defining their specific heat capacity and density. We use heating water.

The signal bus transmits values between components that do not necessarily exist in the real system. Examples are control signals or values that a boundary condition communicates to its associated component.

3.1.5. Scenario

We use the TPDE system as an example to demonstrate the functionality of the simulator qualitatively. To do this, we simulated the system over a typical constructed working day from 7 a.m. to 7 p.m. with the following initial and boundary conditions.

The consumption profiles for the buildings and the AKM follow the working day: In the morning, the demand increases when employees enter the office and turn on the heating. In the afternoon, the sun heats up the offices and consumption decreases. See Fig. 9(a). The absorption chiller runs for a while around noon.

We chose the environmental conditions T^{env}, p^{env} and f^{env} to match this: The environmental temperature rises from about 5 °C in the morning to 15 °C around noon and drops again in the afternoon. We assumed a foggy morning with high humidity, which then drops as the temperature increases from noon. In the evening, humidity rises sharply as it starts raining again. We assumed the environmental air pressure to be constant throughout the sample day.

The storage tank is full at the start of the simulation because we claim that more power was supplied than consumed the previous evening. The initial temperature distribution we choose can be seen in Fig. 9(b). In the afternoon we provoke a malfunction in the system to test the functionality of the control unit: We assume that the heating system of one of the loads fails at 3 p.m. and its return temperature rises steadily. To mimic this scenario, we feed hot return water into the thermal storage.

3.1.6. Results

The simulation delivers the following results, gathered in Fig. 9.

Fig. 9(a) shows the power requirements of the loads. By definition, they are negative because power leaves the ProCoSim system here. We did not use real data, but made schematic assumptions. Therefore, the curves are not continuously differentiable. As the absolute demand increases in the morning, the temperature in all storage layers decreases, shown by Fig. 9(b). Eventually, the temperature falls below the critical storage temperature. The control unit requests the MGT-CHP unit to start, as can be seen in Fig. 9(c). Fig. 9(b) also shows the temperature stratification in the storage tank. In the layer model, we assume that a temperature stratification forms in the water tank while it is active: Cooler water with a higher density collects at the bottom where the cool return flow ports are and warmer water with a lower density collects at the top where the MGT-CHP heats the water. The storage tank intermixes completely around 10 a.m. because a lot of hot heating water is drawn from it quickly. The hot water flow from the MGT-CHP cannot compensate for that. As the demand decreases in the afternoon, the cold return mass flows from the loads decrease. With this and the malfunction, the storage tank temperatures increase again from 3 p.m. The temperature stratification in the storage tank reverses. The hot return water prevents a new equilibrium from forming. Eventually, the storage temperature reaches the critical upper limit and the MGT-CHP unit shuts down.

The result shows: The simulation reflects reality as expected. The chosen calculation concept of the interacting components leads to a solution: the calculated time evolution matches our expectations. We have modeled the components with sufficient accuracy using the equations and look-up table data. For example, the control unit performs as expected during both the natural MGT-CHP start-up in the morning and the forced emergency shutdown in the afternoon.

3.2. Application 2: Hybrid-electric aircraft propulsion

The second application is from modern aviation and shows how flexible ProCoSim can be used. We started from the setup explained in the previous Section 3.1.4. Then we extended the simulator and the libraries: Here, we make intensive use of the algebraic equilibrium calculation from Section 2.4, we use new components and bus types, and we systematically plot the results. The aircraft propulsion use case is very different from the decentralized energy supply shown above. For ProCoSim, however, both applications behave quite analogously as we discuss in the first paragraph of Section 3.2.4. They show the same categories of components and buses and have the same typical questions.

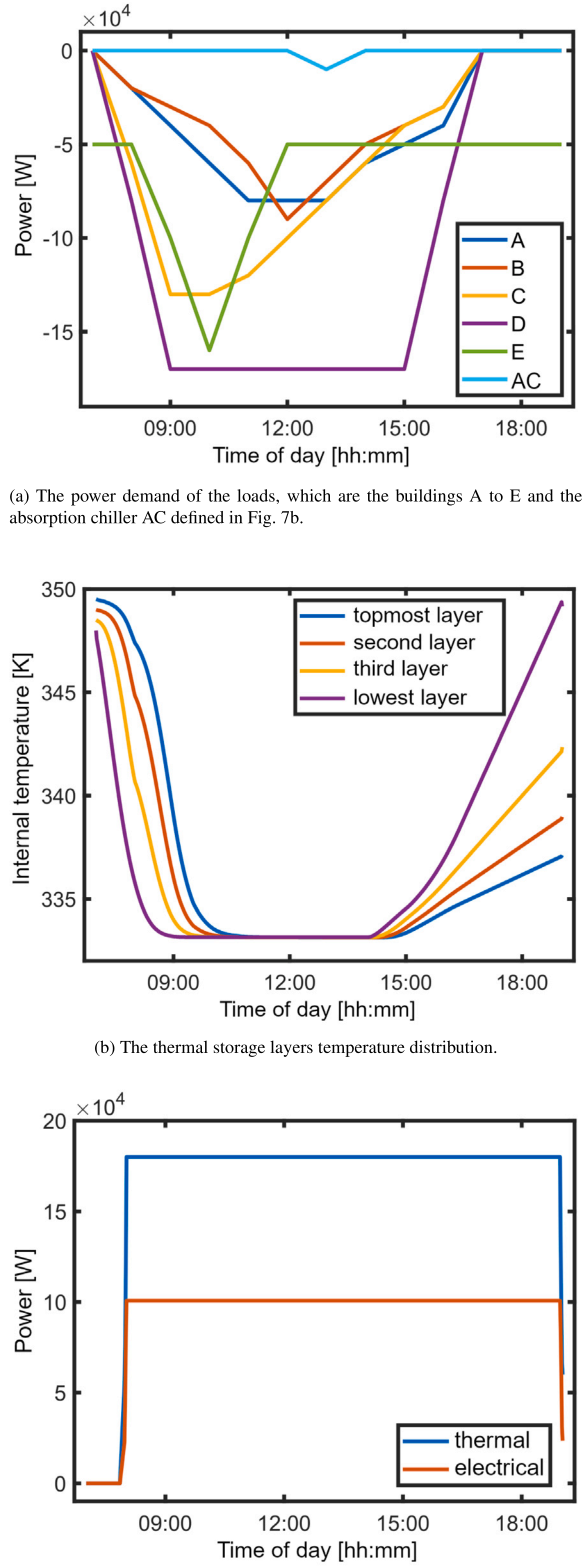
3.2.1. System

We use ProCoSim to simulate a hybrid-electric aircraft propulsion system, shown in Fig. 10. A gas turbine and a battery provide the propulsion power. The battery supplies or absorbs power as required during the flight mission. An electric motor/generator converts the electrical power into mechanical power and vice versa.

The DLR Institute of Combustion Technology built this system as a ground test bed called ‘HeBo’, described in Kislat et al. [28]. The propulsion propeller is emulated by a load machine and the battery by a DC source/sink. ProCoSim enables us to depict physically existent components as well as emulated components that are not yet part of the experimental setup. The construction took place as part of the EXACT project described in [29,30]. The aim of the project is to achieve eco-efficient flying with drastically reduced impact on the climate while being economically feasible. To this end, the project is investigating hybrid-electric propulsion concepts, aircraft configurations and the interaction with the airport infrastructure.

3.2.2. Operating mode

With the project aim from the previous section, multiple hybrid-electric configurations are conceivable. An overview over the aircraft configurations analyzed in the project EXACT is given in [31,32]. According to these results a concept combining a battery-electric propulsion system with a gas turbine range extender is highly interesting regarding a reduction in mission block energy of a 70 passenger regional aircraft. In these studies the block energy could be reduced by 30%–45% compared to the baseline turboprop aircraft. Furthermore, this range extender configuration is highly suitable for experimental investigations on DLR’s ground test bed ‘HeBo’. Therefore, as a test case of ProCoSim’s capabilities, we investigate this configuration as well as a Boosted Turboprop configuration with the component connection shown in Fig. 10. To give an idea of ProCoSim’s modularity, we also take a brief look at one alternative configuration (turboelectric) that requires another interconnection.



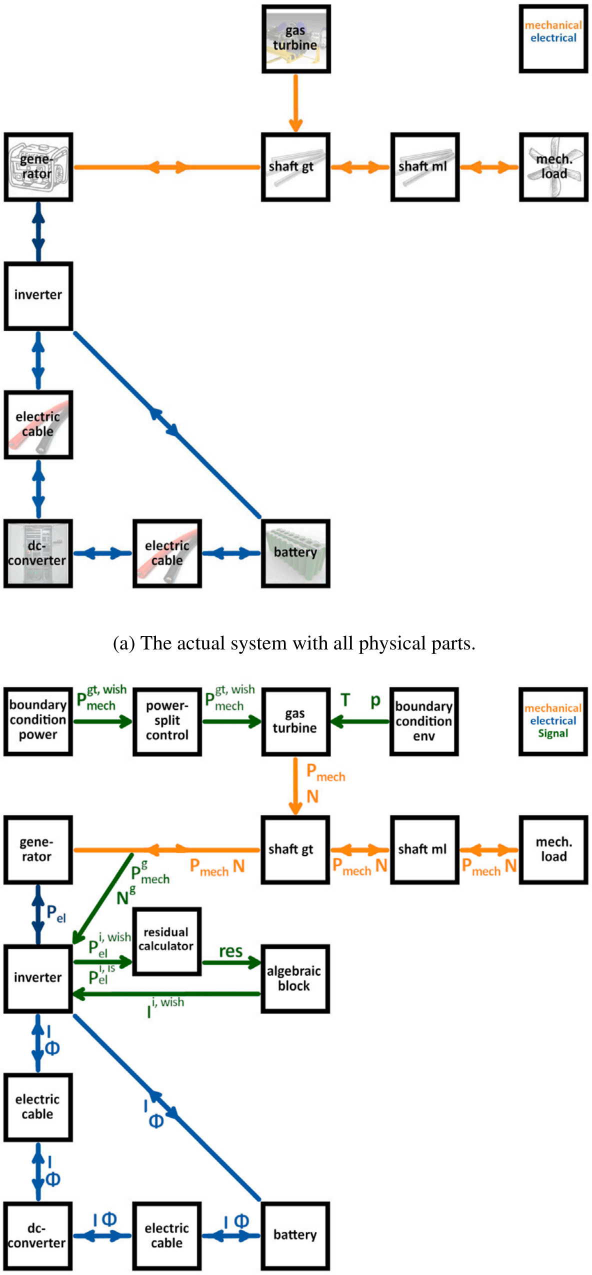
(a) The power demand of the loads, which are the buildings A to E and the absorption chiller AC defined in Fig. 7b.

(b) The thermal storage layers temperature distribution.

(c) The power supply from the MGT-CHP.

Fig. 9. Simulation results of the TPDE system over time and for one working day.

The first operating mode we study is called a Boosted Turboprop. Here, the gas turbine provides most of the propulsive power, see [33]. The idea is to use a gas turbine that is designed for the cruise, the main



(a) The actual system with all physical parts.

(b) The modeled system including all mathematical components.

Fig. 10. The hybrid-electric aircraft propulsion system in the configuration Boosted Turboprop or Range Extender. Mechanical buses are shown in orange, electrical buses in blue and signal buses in green. Some signal bus quantities correspond to the component given in the superscripts: *gt* for gas turbine, *g* for generator, *i* for inverter. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

phase of the flight, not for the climb that requires the most power. Thus, the gas turbine works most efficient during the cruise. This is beneficial especially for long distances where the cruise is by far the longest flight phase. During the climb, the battery supports the gas turbine. In the following cruise phase, less power is required and the smaller gas turbine is sufficient. The battery is being recharged during the descent.

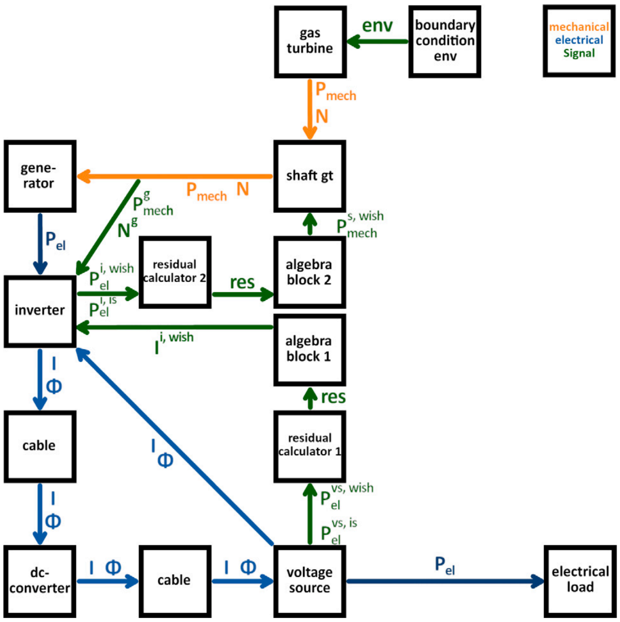


Fig. 11. The hybrid-electric aircraft propulsion system in the configuration Turboelectrical.

The second operating mode for the same configuration is using the gas turbine as a Range Extender with a predominantly battery-powered propulsion. This concept has been described by Balack et al. [34]. The idea is to use a smaller and lighter battery compared to a full electric drive to save weight. A flight mission requires a certain amount of buffer energy for potential reserve loops. It ensures a safe mission, even if, for example, it is not possible to land at the planned destination airport. Providing this buffer energy electrically would drastically increase the battery weight. The additional weight would have to be carried for the entire mission. In comparison to an all-battery configuration, it is more advantageous to add a small gas turbine and carry a bit of kerosene. It can be used up during the mission, even at an early stage. This is beneficial for short distances where the potential reserve loop accounts for a large portion of the entire mission. Other options to provide the reserve energy are possible e.g. fuel cells running on hydrogen. Since the focus of this paper is not on the comparison and evaluation of different hybrid-electric configurations, these other options will not be discussed here.

We demonstrate the use of ProCoSim with the Boosted Turboprop and Range Extender configuration from Fig. 10. However, both ProCoSim and the test bed allow other hybrid-electric connections of the components as well. For example, the Turboelectrical configuration shown in Fig. 11: It has an electrical load instead of the mechanical load. The electrical load comprises, for example, the on-board electronics, but also the flight propulsion via electric motors. We set up a modeling plan for this Turboelectrical configuration as well. With only a few additions such as the electrical load and another algebraic calculation loop, ProCoSim could also model the Turboelectrical system.

3.2.3. Problem

In this application, we focus on mapping the ground test bed with ProCoSim and on the coupled gas turbine. We want to know: Which interconnection and operating mode are efficient? Which are beneficial for the gas turbine service life? How are the test rig component limits complied with during the mission?

ProCoSim provides reference points for these questions based on the power balances. If you want to optimize the entire aircraft in the flight configuration, more parameters are needed. Our partners in the EXACT project have simulated and investigated these issues in optimization studies [35,36]. They also took into account aspects that are inaccessible to ProCoSim: Aircraft geometry, weight, cost, and supporting processes such as de-icing. Examples and overviews are given in [32,33].

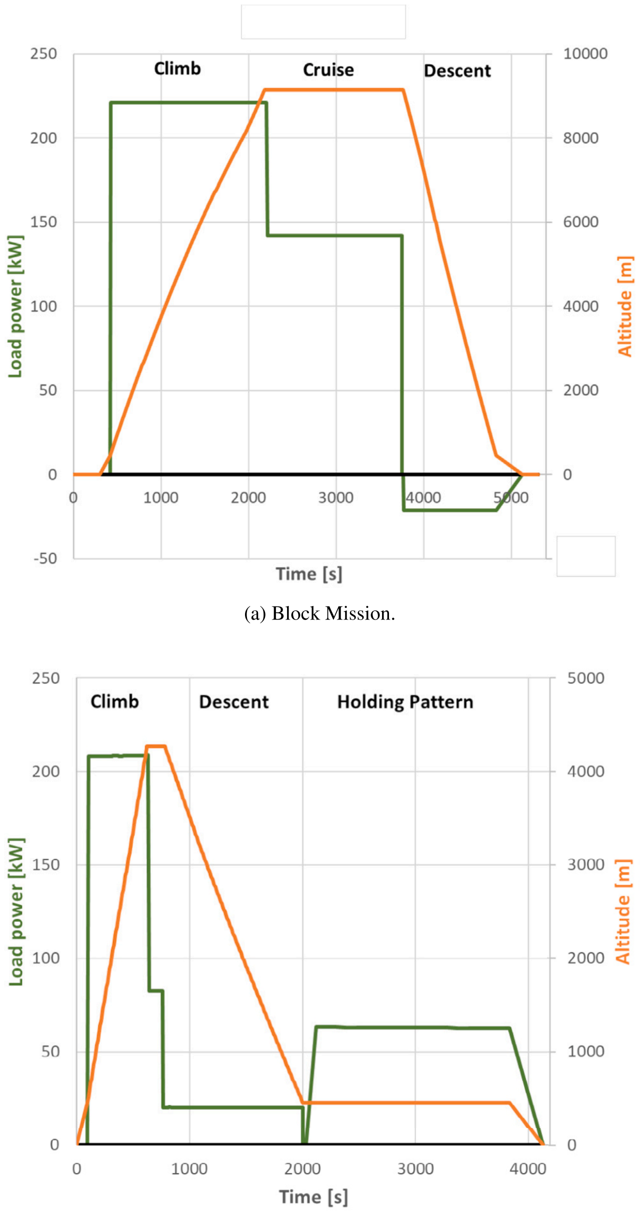
We use one test configuration of these detailed simulations for our experiments and simulations, the regional plug-in-hybrid design case. We can run the designed load profile of a typical flight mission on the test rig and test how the most promising operating modes prove themselves in practice. ProCoSim can accompany the experiment: The simulator shall reproduce the mission and provide information for advance planning and more safety for the experiment. ProCoSim shall provide data to analyze ramp up and ramp down maneuvers of the components as a pretest prediction. It shall also help with troubleshooting on the test rig. In addition to the experiments, we will assume altitude conditions for the components in ProCoSim. This shall give us a trend of how the drive would behave at flight level.

For the following simulations, we use data from our project partners that describe a flight mission over time: Altitude, distance, thrust force, flight phase and so on. From these data, we calculate the LUT data needed for the components: the power load profile, the split between mechanically and electrically provided power and the environmental conditions. We explain this in paragraphs *Mechanical Load*, *Power-Split Control* and *Boundary Condition* of the next Section 3.2.4. To study both operating modes from Section 3.2.2, we use two different types of flight missions, modified from Arzberger and Zimmer [35]: The Block Mission with climb, cruise and descent represents a typical standard flight mission with a fairly long cruise phase. It serves as an example for the Boosted Turboprop operation. The Reserve Mission contains a climb, a short cruise, the descent, a holding loop and the rest of the descent with landing. The focus is on the holding loop. We use this mission as an example for the Range Extender configuration. To match our test rig setup, we scaled down the power curve of both missions as shown in Section 3.2.4. And we adjusted the Block mission to our needs: With an extended cruise phase and the assumption of a slight negative load power P_{load} (descent) < 0 during the descent. This means potential energy is utilized as mechanical energy when the propeller is passively driven as the aircraft decreases. The environmental boundary conditions follow the International Standard Atmosphere as described in Eq. (37)

3.2.4. Modeling

We map the Boosted Turboprop configuration as a network of components and buses, as in the TPDE use case in Section 3.1.4. The system consists of a mechanical part as well as a direct current (DC) electric circuit. In the following paragraph, we show how both systems appear analogous for ProCoSim. The implementation details for the components and buses follow afterwards.

Analogy between hybrid-electric aircraft propulsion and decentralized energy supply. In the aircraft propulsion system, the energy considered is mechanical and electrical, while in the heating circuit system, the energy considered is thermal. The main energy source for the aircraft propulsion concept is the Gas Turbine, which provides mechanical power. Analogous to the MGT-CHP unit in the TPDE above, which provides thermal energy. Accordingly, the Load, i.e. the propeller or the load machine, uses mechanical power instead of thermal power as the buildings in the heating circuit above do. Mechanical Shafts connect the gas turbine and the load, Cables connect the electrical components; in the TPDE system, the Heating Water Pipes connect the components. The aviation application includes the Battery to store electrical energy; the distributed energy system includes the water tank to store thermal energy. Both systems include Boundary Conditions to describe the environment and specify the demand curves: here the mission profiles and above the heating water demand. Although the TPDE system is physically larger than the aircraft application, both examples shown here are similar in terms of power class size.



(a) Block Mission.

(b) Reserve Mission.

Fig. 12. The considered flight missions for the hybrid-electric aircraft propulsion. This is original data from our project partners shown in Section 3.2.5, which we modified to our needs as explained in *System Scaling* in Section 3.2.4.

Gas turbine. The Gas Turbine receives a power demand and the environmental conditions via a signal bus, see Fig. 10. It provides the power and determines the current speed of the output shaft from the given values using a stored look-up table.

We use our in-house quasi-stationary MGTS3 simulation data presented in [37]. They are validated for ground conditions and most temperature and pressure ranges. For altitude conditions, we assume that the maximum shaft power $P^{\text{max}}_{\text{gt}}$ the gas turbine can provide decreases with the decreasing environmental pressure p^{env}, as described in [38]: $P^{\text{max}}_{\text{gt}} = P^{\text{max}}_{\text{gt}}(p^{\text{env}})$. To achieve this in the model, we claim that the rotational speed does not exceed the realistic value range that the turbo components can deliver without damage. If we consider the temperature and pressure profiles $T^{\text{env}}(h)$ and $p^{\text{env}}(h)$ from Eq. (37) as well as the height profile $h(t)$ of the missions, we receive the correlations $P^{\text{max}}_{\text{gt}}(h)$ and $P^{\text{max}}_{\text{gt}}(t)$.

The gas turbine in use is a Rolls-Royce M250 C20B helicopter gas turbine with a ground maximum shaft power output of 314 kW, specified in [39] and described by Menrath [40].

Mechanical load. The Mechanical Load represents the propeller in the aircraft drive or the load machine in the ground test bed. It receives the current power demand or surplus via a signal bus and passes it on to the connected shaft. Power flow is possible in both directions: during climb, take-off, cruise or active acceleration, the load demands power from the system; during descent, it supplies power to the system while the propeller is driven by air resistance. The information flow of the set variables is identical in both cases: from the signal bus over the Mechanical Load to the connected shaft.

We calculate the required thrust power at the Mechanical Load from the flight mission data, which we describe in Section 3.2.5. The original required thrust power is

$$P_{\text{load, orig}}(t) = F_{\text{thrust}}(t)\frac{v_{\text{air}}(t)}{\eta_{\text{fan}}(t)}. \tag{12}$$

The time evolution of the thrust force F_{thrust}, the true air speed v_{air} and the propeller efficiency η_{fan} follow from the mission data.

Battery. The battery is part of the DC circuit in the system, compare Fig. 10. We model the Battery as a combination circuit of individual battery cells. X cells are connected in parallel to achieve the maximum current in the system. And Y parallel circuits are connected in series to achieve the required maximum voltage.

The battery cell has two bidirectional electrical inputs/outputs connected, where it receives an incoming or outgoing current request I on one side. It reads the potential difference U that is required for that from a stored look-up table, where

$$U = \Phi^{\text{out}} - \Phi^{\text{in}}. \tag{13}$$

This LUT contains the relationship between the battery's state of charge SOC, I and U, see Fig. 13, over all relevant conditions, so that no additional values were extrapolated. The battery compares the needed voltage to its maximum possible charge or discharge voltage. For $|U| <= |U^{\text{max}}|$, the battery is in the CC phase for 'constant current'. By convention, it sets the potential to zero on one side and to $\pm\Phi$ on the other. It delivers the required current:

$$I_2 = I_1 \tag{14a}$$

$$\Phi_1 = 0 \tag{14b}$$

$$\Phi_2 = \pm U(I_1, \text{SOC}). \tag{14c}$$

If $|U| > |U^{\text{max}}|$, the battery is in the CV phase for 'constant voltage'. It sets the potentials to zero and to $\pm U^{\text{max}}$. The actual current follows from the LUT:

$$\Phi_1 = 0 \tag{15a}$$

$$\Phi_2 = \pm U^{\text{max}} \tag{15b}$$

$$I_2 = I(U^{\text{max}}, \text{SOC}). \tag{15c}$$

The battery state of charge reflects the stored battery energy E:

$$\frac{\text{SOC}(t)}{100\,\%} = \frac{E_{\text{scal}}(t)}{E_{\text{scal}}^{\text{max}}} \tag{16}$$

It evolves according to the Coulomb Counting Method as

$$\text{SOC}(t) = \text{SOC}(t_0) - 1/C\int I(t)\,\mathrm{d}t, \tag{17}$$

where C is the battery capacity and the minus aligns with the definition of $I > 0$ as discharge. The sign choice of the potential is as follows: The battery sets the potential to $\pm U$ at the port where it receives a claim or delivery, and to zero at the port without a claim or delivery. That means: If the battery is being discharged, it receives a current claim at the outlet and sets

$$\Phi^{\text{in}} = 0 \tag{18a}$$

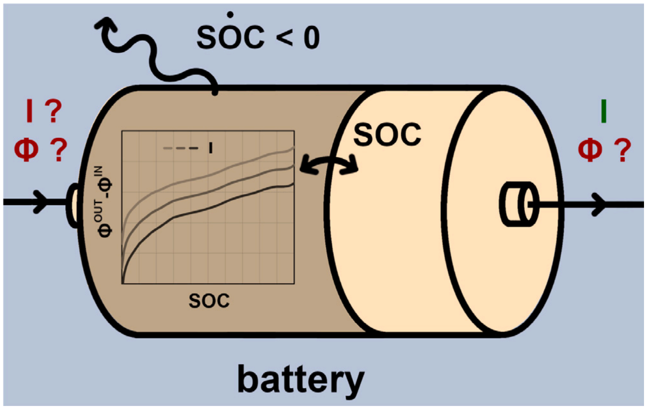


Fig. 13. The internal calculations (yellow) of the component battery: It receives an inflow or outflow current demand. It calculates the outflow or inflow current as well as the potential difference according to Eq. (14) or (15), depending on the phase. I and $\Phi^{\text{out}} - \Phi^{\text{in}}$ are related via the state of charge. The state of charge evolves according to the ODE (17) (blue). Compare Figs. 3 and 4 for color code. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\Phi^{\text{out}} = \Phi^{\text{in}} + U = U, \tag{18b}$$

according to Eq. (13). If the battery is being charged, it receives a current delivery at the inlet and sets

$$\Phi^{\text{out}} = 0 \tag{19a}$$

$$\Phi^{\text{in}} = \Phi^{\text{out}} - U = -U. \tag{19b}$$

The temperature influence on the battery was included in the experimental LUT data in use. For the modeled ground test rig we choose $T = 298\,\text{K}$.

We neglect battery degradation in this model since the test scenarios shown in Section 3.2.5 describe single flight missions from take-off to landing only, with two to three phases of charging and discharging phases at the maximum. In order to investigate a series of flight missions in the future, battery degradation over the charge and discharge cycles should be taken into account as well.

We use the LG 18650 MJ1 battery cell with a capacity of 3.5 Ah and the data from Albrecht et al. [41] with a slightly modified model.

Remember that with ProCoSim, we focus on the power distribution mainly in the ground test rig. For an actual flight application, the weight of the battery pack would also have to be taken into account. This means that the overall system would have to be optimized in terms of mission, operating mode and aircraft weight.

Converted power. The sources and loads carry an ODE for the total energy transferred: $\dot{W} = P(t)$. It adds up the work supplied or demanded during the simulation. In this way we calculate the total mechanical energy that the Gas Turbine or the load has processed up to the current time.

System scaling. We ensure that the simulation data matches the dimension of the experimental setup. For that, we adjust the mission data and choose the battery pack accordingly.

First, we scale the required load power (12) down to fit the dimensions of our ground test rig. For the Reserve mission it is:

$$P_{\text{load, scal}}(t) = k\,P_{\text{load, orig}}(t). \tag{20a}$$

For the Block mission, we also consider the power gain during the descent:

$$P_{\text{load, scal}}(t) = \begin{cases} P_{\text{load, scal}}^{\text{min}} < 0 & \text{for} \quad \text{descent} \\ k\,P_{\text{load, orig}}(t) & \text{else.} \end{cases} \tag{20b}$$

We use the scaling factor

$$k = \frac{P^{\max}_{\text{load, scal}}}{P^{\max}_{\text{load, orig}}}, \tag{21}$$

where $P^{\max}_{\text{load, orig}}$ and $P^{\max}_{\text{load, scal}}$ are the maximum power values that occur during the mission on the aircraft and on the ground test rig, respectively. We expect the highest load power within the mission at the beginning of the climb phase, where the most power is needed. The battery supports the gas turbine at this stage, it is being discharged. Thus, from the power split Eq. (31) follows:

$$\left|P^{\max}_{\text{load, scal}}\right| = P^{\max}_{\text{gt}} - \left|P^{\max}_{\text{bat, scal}}\right|. \tag{22}$$

With that, we choose $P^{\max}_{\text{load, scal}}$ so that both missions can be run with a battery pack of

$$\left.\begin{array}{l} P^{\max}_{\text{bat, scal}} \\ P^{\min}_{\text{bat, scal}} \end{array}\right\} = \pm\frac{1}{3}\,P^{\max}_{\text{gt}} = \pm\frac{1}{3}\,P_{\text{gt}}\left(\text{climb}, h^{\min}\right). \tag{23}$$

$P^{\max}_{\text{gt}}$ is the highest power the gas turbine can provide during the mission. This happens when the gas turbine runs at full load at lowest possible altitude, right at the start of the climb. To receive the value of $P^{\max}_{\text{gt}}$ for both missions, we use the maximum values of the gas turbine LUT data, described in the paragraph *Gas Turbine*, and interpolate it exponentially:

$$P^{\max}_{\text{gt}}\left(T^{\text{env}}\right) = P^{\max}_{\text{gt}}(0)\,\exp\left[c\,T^{\text{env}}\right], \tag{24}$$

with $P^{\max}_{\text{gt}}(0) = 1020.3\,\text{W}$ and $c = 0.0205\,1/\text{K}$. Finally, we lower the maximum gas turbine power in the mission by the discrete step width of the LUT:

$$\bar{P}^{\max}_{\text{gt}} = P^{\max}_{\text{gt}} - \Delta P^{\text{LUT}}_{\text{gt}}. \tag{25}$$

That way, we evade the boundary values $P^{\max}_{\text{gt}}(T^{\text{env}})$ of the LUT to avoid stepped simulation results for $P_{\text{gt}}(t)$ without having to extrapolate.

With Eq. (12) and Eqs. (20) to (25), the mission design for $P^{\text{scaled}}_{\text{load}}$ is complete.

Next, we assume a battery pack of $X = 23$ parallel cells and $Y = 178$ serial cell sets for both missions. It provides the maximum voltage, current and capacity needed to run the missions:

$$U^{\text{max, needed}}_{\text{pack, charge}} = 796\,\text{V} \tag{26a}$$

$$U^{\text{max, needed}}_{\text{pack, discharge}} = 739\,\text{V} \tag{26b}$$

$$I^{\text{max, needed}}_{\text{pack, charge}} = 158\,\text{A} \tag{26c}$$

$$I^{\text{max, needed}}_{\text{pack, discharge}} = 202\,\text{A} \tag{26d}$$

$$C^{\text{needed}}_{\text{pack}} = 21\,300\,\text{As}. \tag{26e}$$

We ensure that these missions can also be run on the test rig that has a maximum dis-/charge voltage of $U^{\text{max, limit}}_{\text{pack}} = 800\,\text{V}$, a maximum dis-/charge current of $I^{\text{max, limit}}_{\text{pack}} = 600\,\text{A}$ and a maximum electrical power of 300 kW. We set $\text{SOC}(t_0) \approx 80\,\%$ and choose the battery pack capacity large enough to keep SOC moderate during the mission:

$$5\,\% \lesssim \text{SOC} \lesssim 95\,\%. \tag{27}$$

To achieve that, we simulated the missions and integrated the charge/discharge current over time to estimate the needed battery capacity:

$$C^{\text{needed}}_{\text{pack}} >= \max\left[\left|\int I(t)\,\mathrm{d}t\right|\right]. \tag{28}$$

With Eqs. (14) to (19) and (26) to (28), the battery design is complete.

Power-split control. The Power-Split Control decides how the thrust power is split between the Gas Turbine and the Battery. It receives the required total thrust power designed in Eq. (20) via a signal bus and informs the Gas Turbine of its share accordingly. The rest comes from the Battery and follows from the system according to energy conservation and losses caused by friction and ohmic resistance.

Our control strategy for the Block Mission in the Boosted Turboprop scenario is as follows. The Gas Turbine should be at full load or off. This is expected to be more efficient than partial load operation. For the Reserve Mission in the Range Extender scenario, we require: The Battery should provide the power on its own whenever possible, the Gas Turbine only covers the peak demand during the short climb and cruise. Thus, we choose for both missions:

$$P_{\text{gt, scal}}(t) = \begin{cases} \bar{P}^{\max}_{\text{gt}} & \text{for} \quad \text{phase}_{\text{gt}} \\ 0 & \text{else} \end{cases} \tag{29}$$

with

$$\text{phase}^{\text{block}}_{\text{gt}} = \text{climb, acceleration, cruise} \tag{30a}$$

$$\text{phase}^{\text{reserve}}_{\text{gt}} = \text{climb, acc.}_{t<1000\,\text{s}}, \text{cruise}. \tag{30b}$$

The battery power measured at the load follows accordingly:

$$P_{\text{bat, scal}}(t) = P_{\text{load, scal}}(t) + P_{\text{gt}}(t). \tag{31}$$

The sign conventions are:

$$P_{\text{load}} \gtrless 0 \quad \text{power exits/enters the system} \tag{32a}$$

$$P_{\text{gt}} > 0 \quad \text{gas turbine provides power} \tag{32b}$$

$$P_{\text{bat}} \gtrless 0 \quad \text{battery being charged/discharged}. \tag{32c}$$

Note: Eq. (31) describes the balance of power flows right at the mechanical load. The actual battery power measured at the battery is higher due to power losses between the battery and the load:

$$\left|P_{\text{bat}}\right|(\text{at battery}) > \left|P_{\text{bat}}\right|(\text{at load}). \tag{33}$$

Generator/motor and inverter. The components Generator/Motor and Inverter are the interface between the mechanical and electrical parts of the system. The Generator/Motor can act as a generator or as a motor. For simplicity, we call it only Generator.

In the model system, the Generator converts mechanical power into electrical power or vice versa. The Generator has an Electric Power bus and a mechanical Rotation Contact connected to it. It receives the value for the mechanical power P_{mech} and calculates the value for the electric power P_{el}. It takes into account losses that depend on the rotation speed N and P_{mech}. If mechanical power is converted to electrical power, it calculates

$$P_{\text{el}} = \eta\left(N, P_{\text{mech}}\right)\,P_{\text{mech}}. \tag{34a}$$

If electrical power is converted to mechanical power, it calculates

$$P_{\text{el}} = \frac{P_{\text{mech}}}{\eta\left(N, P_{\text{mech}}\right)}, \tag{35a}$$

where η is its efficiency.

The Inverter in the model system transmits electrical power P_{el} into electrical current I and potential Φ or vice versa. Note: In the test rig, the inverter is also the interface between the system's DC circuit and the external alternating current (AC) mains power supply. We do not need to consider the AC/DC interface for our investigations. Thus, we only look at the DC circuit. The inverter then only acts as a lossy transmission between P_{el} and $\{I, \Phi\}$.

The Inverter receives P_{el} and calculates the corresponding current I^{in}, I^{out} and potential Φ. Its losses also depend on N and P_{mech} on the Generator shaft, which it receives by signal bus as soon as N and P_{mech} are known. The calculation $P_{\text{el}} = P_{\text{el}}(I, \Phi)$ in the other direction is therefore impossible because $\eta(N, P_{\text{mech}})$ would be unknown. Instead, we use an iterative calculation loop described in the next paragraph. The inverter passes the required power P^{wish} and the current actual power P^{is} to a signal bus. In the Boosted Turboprop and Range Extender configuration, with information flow from Generator to Inverter, it is:

$$P^{\text{is}}_{\text{el}} = I^{\text{in}}\,U_{\text{inv}} \tag{36a}$$

$$P_{\mathrm{el}}^{\mathrm{wish}} = \begin{cases} -\dfrac{P_{\mathrm{el,\ gen}}}{\eta(N, P_{\mathrm{mech}})} & \text{for} \quad P_{\mathrm{el,\ inv}\rightarrow\mathrm{gen}} \\ -\eta\left(N, P_{\mathrm{mech}}\right) P_{\mathrm{el,\ gen}} & \text{for} \quad P_{\mathrm{el,\ gen}\rightarrow\mathrm{inv}}, \end{cases} \tag{36b}$$

where U^{inv} is the voltage at the Inverter and $P_{\mathrm{el,\ gen}}$ is the electric power at the Generator. $P_{\mathrm{el,\ inv}\rightarrow\mathrm{gen}}$ means power flow from Inverter to Generator and $P_{\mathrm{el,\ gen}\rightarrow\mathrm{inv}}$ means the opposite.

Generator and Inverter hold look up tables for the dependency $\eta(N, P_{\mathrm{mech}})$ which follow from interpolated manufacturer's data.

Note: In the Turboelectric configuration with information flow from Generator to Inverter, as shown in Fig. 11, the roles of $P_{\mathrm{el}}^{\mathrm{wish}}$ and $P_{\mathrm{el}}^{\mathrm{is}}$ would be reversed in Eqs. (36).

Algebraic block and residual calculator. The components Generator and Inverter allow only unidirectional calculation: from the mechanical system part over Generator and Inverter to the electrical circuit. Nevertheless, ProCoSim can determine power and information flows in both directions. This is done using the iterative solution method described in Section 2.4.

The Generator and the Inverter cannot directly calculate $P_{\mathrm{mech}} = P_{\mathrm{mech}}(P_{\mathrm{el}}(I, \Phi))$. Because their losses depend on N and P_{mech} on the generator Shaft. This direction would be needed if the set values information flows from the electrical to the mechanical part of the system. For example, when the Battery requires or supplies a certain combination of I and Φ. We solve this case iteratively: The Inverter passes $P_{\mathrm{el}}^{\mathrm{wish}}$ and $P_{\mathrm{el}}^{\mathrm{is}}$ to the Residual Calculator. It compares the values and passes the residual to the Algebraic Block. It communicates with the external solver and receives a new value for I^{wish}, which the Inverter tests in the next iteration step. This cycle is repeated until the set Generator power $P_{\mathrm{el}}^{\mathrm{is}}$ actually matches the required or supplied power of the Battery $P_{\mathrm{el}}^{\mathrm{wish}}$ apart from the electric circuit losses.

Mathematically, this means: The given variables needed to solve the overall system are in some cases distributed across multiple components. If we wanted to solve this system exactly, we would have to set up and solve its system of equations globally. However, we want to keep the modular character of the simulator. To achieve that, we solve the system iteratively, which still allows us to perform the calculation locally in the components.

Shaft, electric cable and DC-converter. These are quite simple components that connect the other more complex components. They transmit certain quantities between them, taking losses into account.

The Shaft transmits mechanical power at a certain rotation speed. The Shaft component can be ideal or lossy, linear or branched, and generally allows power to flow in both directions. The Shaft sets P_{mech} and N in those connected buses where they are unknown: It distributes the value of N to all buses where N is unset. If P_{mech} is set in all but one connected bus, it calculates the unknown P_{mech} based on energy conservation and friction losses.

The Cable connects electrical components in a series connection. Current can flow in either direction. The Cable acts as an ohmic resistor. It passes I from one connection to the other and sets Φ according to the voltage drop it causes. The ohmic losses follow from a model including temperature dependence, cable length and material, or from an LUT.

The DC-Converter includes cables and switches. It is dominated by long cables and therefore acts as an ohmic resistance like the Cable component.

Boundary condition. The Boundary Conditions in this system are based on the data from the flight missions used. They provide the required thrust power $P_{\mathrm{mech}}(t)$ at the Load and the current flight phase, which gives the power split $P_{\mathrm{mech}}(t, \text{Gas Turbine})$ required by the Control Unit.

The environmental conditions $T^{\mathrm{env}}(t)$, $p^{\mathrm{env}}(t)$ of the Gas Turbine are constant on the ground; those at altitude follow from the flight altitude $h(t)$ of the mission:

$$T^{\mathrm{env}}(h) = T_0^{\mathrm{env}} + \frac{dT^{\mathrm{env}}}{dh} h \tag{37a}$$

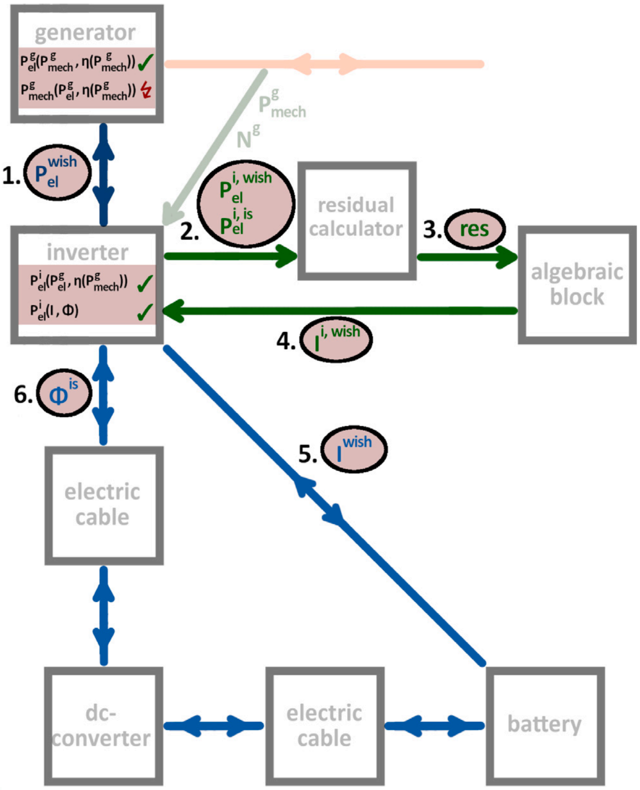


Fig. 14. The algebraic loop (red) in the hybrid-electric system: We iteratively determine the current that provides the required power. The numbers indicate the order of one iteration step: (1.) An electrical power is desired at the inverter and (2.) is compared to the actual power, then (3.) the algebraic block receives the residual and (4.) provides the corresponding electric current. (5.) It passes the circuit and (6.) results in an actual potential. (1.) The potential again results in an electric power at the inverter. If $P_{\mathrm{el}}^{\mathrm{i,wish}} \neq P_{\mathrm{el}}^{\mathrm{i,is}}$, a new iteration step starts. The components generator and inverter can only perform direct calculations for an information flow from the mechanical to the electrical part of the system. This is because of the efficiency η that depends on $P_{\mathrm{mech,\ g}}$ and that is part of the relation between $P_{\mathrm{mech,\ g}}$ and $P_{\mathrm{el,\ g}}$. So if the shaft power is given, the power at the battery can be directly computed. If the power at the battery is given, we use the algebraic loop to iteratively compute the shaft power. Compare Figs. 3 and 4 for calculation structure and color code. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$p^{\mathrm{env}}(h) = p_0^{\mathrm{env}} \left(1 + \frac{dT^{\mathrm{env}}/dh}{T_0^{\mathrm{env}}} h \right)^{5{,}255} \tag{37b}$$

Here, we assume an International Standard Atmosphere with $T_0^{\mathrm{env}} = T^{\mathrm{env}}(h = 0) = 288.15\,\mathrm{K}$, $p_0^{\mathrm{env}} = 1.013\,25 \times 10^5\,\mathrm{Pa}$, humid air of $f^{\mathrm{env}}(h) = f_0^{\mathrm{env}} = 60\,\%$ with $dT^{\mathrm{env}}/dh = -6.05 \times 10^{-3}\,\mathrm{K/m}$ and the Barometric Height Formula for $p^{\mathrm{env}}(h)$ [42].

Buses. The bus Rotation Contact describes a mechanical interface in the system, e.g. between a Shaft and the Gas Turbine. The bus carries the mechanical power P_{mech} as a directed quantity in the direction of the power flow and the scalar rotation speed N at which the power is transferred.

The bus Electricity describes the electrical current flow, e.g. between an Electric Cable and the Battery. The bus carries the current strength I as a directed quantity in the direction of the current and the scalar electrical potential Φ present at the connected components. To explain the meaning, we compare two buses that are connected to opposite sides of a component. A voltage source component, for example, creates a potential difference in the circuit; a resistive load, on the other hand, creates a voltage drop. Accordingly, the potential difference of the buses before and after the respective component is shown as $U = \Phi^{\text{out}} - \Phi^{\text{in}}$. The transmitted power is $P_{\text{el}} = U\,I$.

The bus Electric Power describes the interface between the Generator and the Inverter. There is neither a mechanical Rotation Contact nor an electrical connection. The bus only transmits P_{el}.

We also use multiple signal buses for boundary condition inputs and the algebraic solution loop.

Differences between hybrid-electric aircraft propulsion and decentralized energy supply. In addition to the similarities mentioned above, there are some slight differences between both applications.

The control unit in the TPDE application decides dynamically based on system observations. In contrast, the Power-Split Control here works deterministic: Each flight mission is fairly determined in terms of power requirements over time. The amount of power the control unit requires from the gas turbine and battery during the simulation is pre-set. A flexible control based on system observations is not needed for a standard mission. In the future, however, the flexible control will help us if we consider emergency scenarios with unforeseen situations.

Compared to the TPDE, the aircraft propulsion system is more complex. It includes a Generator and an Inverter to convert mechanical and electrical energy. Almost all electrical and mechanical power flows are bi-directional. This means that in some cases the direction of the power flow is opposite to the direction of the information flow of the set variables. At the Inverter and Generator, the calculation is only possible in one direction and the information flow might be opposite to that. In this case we need the additional algebraic solution loop in the system with the components Algebraic Block and Residual Calculator. We illustrate this correlation with an example in Fig. 14. Also see paragraphs *Generator and Inverter* as well as *Algebraic Block* and *Residual Calculator*.

3.2.5. Scenario

We use the hybrid-electric aircraft propulsion application to demonstrate a series of quantitative results that ProCoSim can deliver.

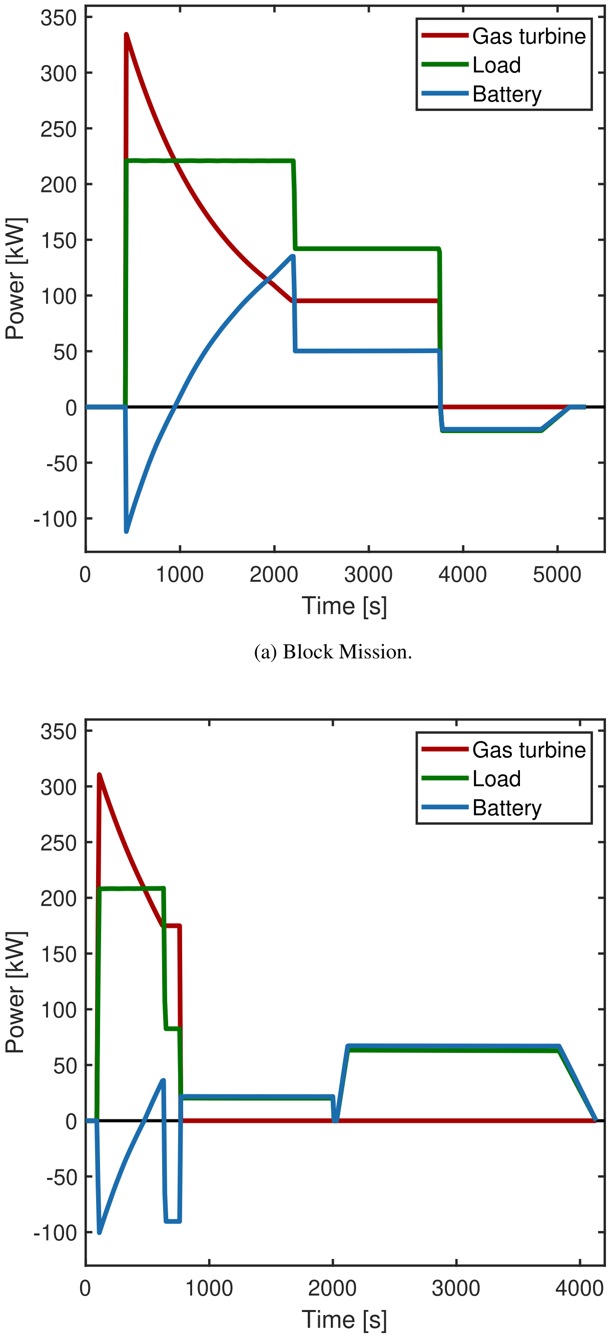
We simulate the system in the configuration shown in Fig. 10. We use the Block mission as an example for the Boosted Turboprop operation, where the battery supports a gas turbine that is designed to almost meet the maximum power needed within the mission. The Reserve mission represents the Range extender configuration, where an oversized gas turbine supports the otherwise battery driven mission during the climb. In contrast to other Range Extender concepts, the gas turbine supports at the beginning of the mission rather than at the end. Both missions are shown in Fig. 12. The reference aircraft for the mission is an interpretation of the ATR72-212 A model with a payload of 6650 kg and a distance of 228 NM ≈ 422 km, compare [43].

Note: We use the two missions to explore completely different modes of operation. A real aircraft would be designed specifically for one of the two modes. We scale the load power in both missions to use our ground test rig to capacity, as shown in paragraph *mechanical load* in the previous Section 3.2.4. The battery capacity is chosen accordingly, compare paragraph *Battery* of Section 3.2.4.

3.2.6. Results

First we consider the power claim from the mechanical load, the power at the source, i.e. the gas turbine, and the power provided by the battery for both missions, see Fig. 15.

For the Block Mission in the Boosted Turboprop operation, the load claim is constant and highest during the climb phase, constant on a lower level during the cruise and slightly negative for the descent. The gas turbine is required to run at full load during climb and cruise. The maximum power the gas turbine can provide decreases during the climb, when air pressure and air temperature decrease with altitude. The battery deals with the deviation $P_{\text{mech, load}} - P^{\max}_{\text{mech, gt}}$ between the power demand and the maximum power supply of the gas turbine at all times: After the take-off, the battery is being loaded by the excess gas turbine power. In the second part of the climb, it provides additional energy $P_{\text{el, bat}} > 0$. Finally, it absorbs energy $P_{\text{el, bat}} < 0$ during the descent. After the cruise phase, the mission is not expected to require more power than the battery alone can provide. Then, the gas turbine is shut down. Towards the landing, the descent charge power decreases with decreasing altitude and air speed.



(a) Block Mission.

(b) Reserve Mission.

Fig. 15. Simulation results of the hybrid-electric system: The scaled power converted at the gas turbine, the load, and the battery. For the load power, the absolute value is plotted. For the definition of the missions, see Fig. 12.

We use this example to demonstrate how the simulator works, compare Section 3.2.4. To achieve this simulation result, the system operates as follows: With the power given at the Gas Turbine and the Load, the Shaft GT calculates the power at the Generator via energy conservation, see Fig. 10. The Inverter converts this power into the appropriate voltage and current. The Battery learns from its Electricity buses how much power $P_{\text{el, bat}} = U\,I$ it must supply or absorb. This absolute amount is larger than $P_{\text{el, gen}}$ because from the Shaft GT to the Battery, the power passes lossy components. It experiences conversion losses at Inverter and Generator and ohmic losses at the Cables and the DC-Converter.

For the Reserve Mission where the gas turbine works as a Range Extender Fig. 15(b) shows: For the last two thirds of the mission, the battery provides the thrust power. During the climb, the gas turbine runs at full load and overcompensates the required thrust. The battery is being charged with the excess. The battery alone covers the rest of the flight, including the holding pattern where the airplane maintains a constant altitude. Towards the landing, the required power decreases with decreasing altitude and airspeed.

Note how Reserve Mission and Boosted Turboprop Mission differ in the landing characteristics. In the Boosted Turboprop Mission, the system gains power during the descent and landing while the airspeed decreases slowly and the propellers are driven by air resistance. In the Reserve Mission, the system must provide power during the reserve loop and the landing while the airspeed must drop quickly and the propellers work actively against the air resistance.

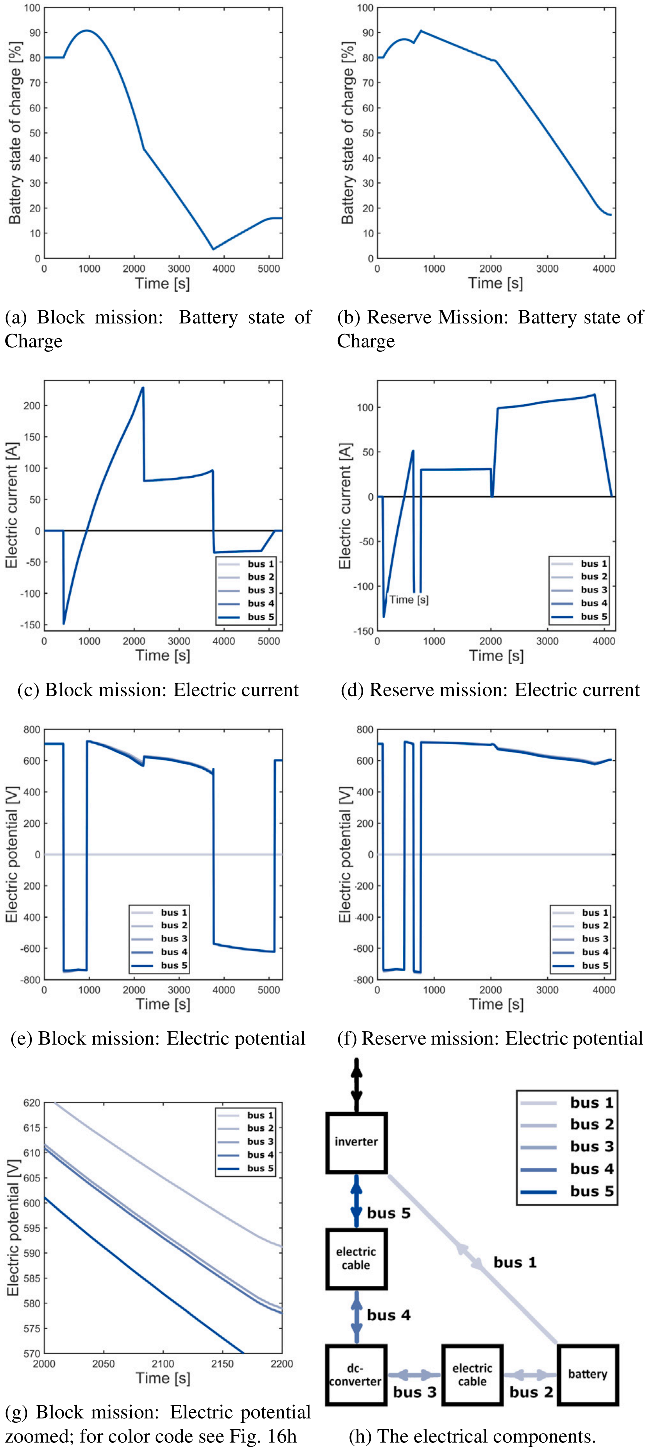
We have plotted the characteristic quantities of the electrical components in Fig. 16 to get an overview of the electrical part of the system.

We start both missions with a battery state of charge SOC of 80 %. Fig. 16(a) shows: For the block mission, the battery is being charged during the first part of the climb and during the descent. For the remaining mission, the battery is being discharged. After the landing, the battery would have to be recharged on the ground for a follow-up mission of this design. The reserve mission in Fig. 16(b) charges the battery in most parts during climb and cruise and discharges it during the descent. After that regular part of the mission, the battery state of charge is on a similar level as it was at take-off. The airplane might start another mission without ground charging the battery in between. However, if a holding pattern is required, the battery can cover that with the chosen capacity and initial state of charge.

Figs. 16(c)–16(e), 16(f) show the electric current I and the electric potential Φ between the electrical components shown in Fig. 16(h) over the missions. While the battery is being discharged and SOC decreases over time, the potential level from the battery decreases. It can only provide less when it is further discharged. At the same time, the current must increase to still achieve the desired power $P = U\,I$. This effect is strongest when the battery is almost empty as the steepening in the $I(t)$ curve towards 3750 s demonstrates in the Block mission. When the battery is being charged, I and Φ are negative which represents the opposite current flow in the circuit. The absolute potential level increases with SOC, the absolute value of the required current decreases accordingly. The sign change in Φ between the charge and discharge phase follows from the sign conventions we choose for the battery in Eqs. (18) and (19). Finally, the close-up in Fig. 16(g) compared with the buses in Fig. 16(h) shows: At a given time, the potential difference that is built up by the battery gradually decreases in the circuit via the individual ohmic resistors.

We simulated the system with ProCoSim for both ground conditions and ambient conditions at altitude, see Figs. 17(a) and 17(b). Both scenarios assume the same load and gas turbine power profile shown in Fig. 15, even though the gas turbine would be able to keep a constant power output on the ground. We want to run the actual flight mission on the ground to investigate the impact of the ambient conditions specifically. On the ground, we assume constant room temperature $T^{\text{env}} = 288.15\,\text{K}$ and standard pressure $p^{\text{env}} = 101\,325\,\text{Pa}$. For the altitude mission, T^{env} and p^{env} drop according to the height profile from Fig. 12.

The fuel consumption on the ground follows the gas turbine power profile. During the "climb" phase, the ground turbine lowers its load and therefore the required fuel mass flow. At high altitude the cause for the reduced fuel consumption is different: The gas turbine still runs at full load and achieves the same power profile due to the lower air pressure. That means, it processes less fuel to receive less power while still running efficiently at the design point. Therefore, the solid curve in Fig. 17(a) on the bottom left is steeper than the dashed curve.



(a) Block mission: Battery state of Charge

(b) Reserve Mission: Battery state of Charge

(c) Block mission: Electric current

(d) Reserve mission: Electric current

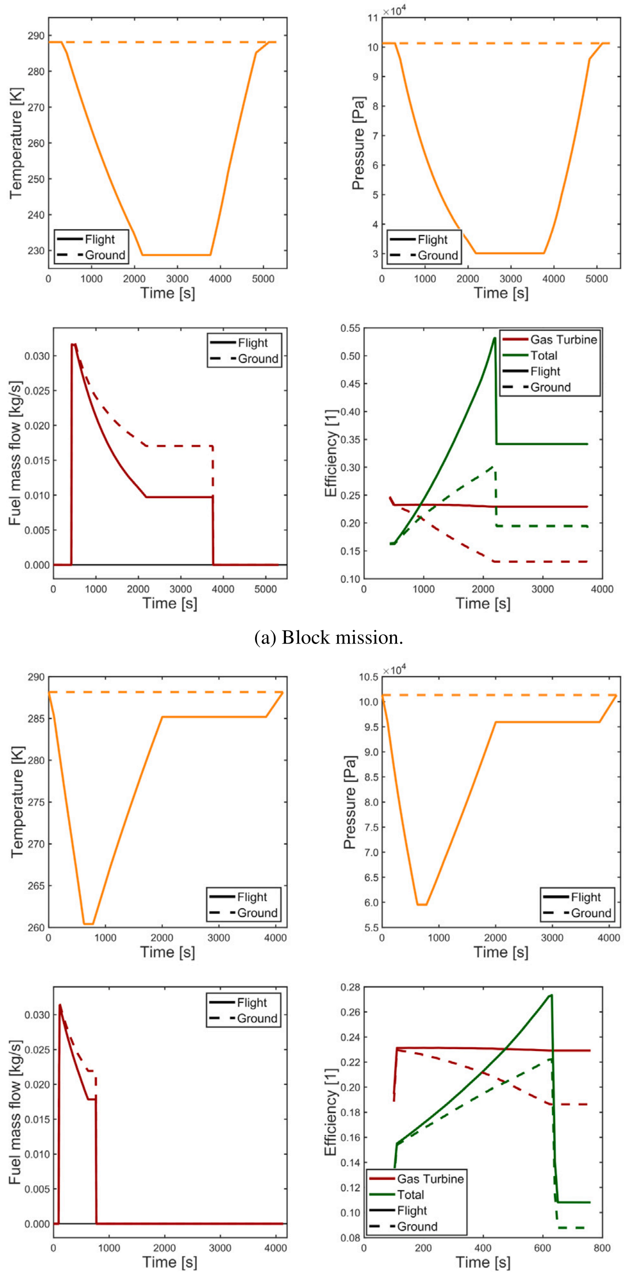
(e) Block mission: Electric potential

(f) Reserve mission: Electric potential

(g) Block mission: Electric potential zoomed; for color code see Fig. 16h

(h) The electrical components.

Fig. 16. Simulation results of the hybrid-electric system. For interpretation, see text and Eqs. (38a) and (38b). The color shades correspond to the buses defined in Fig. 16(h). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Block mission.

(b) Reserve mission.

Fig. 17. Simulation results of the hybrid-electric system: Comparison between ground level and flight level for temperature, pressure, fuel consumption and efficiency.

This also explains the gas turbine efficiency behavior, where

$$\eta_{\mathrm{gt}} = \frac{P_{\mathrm{gt}}}{P_{\mathrm{fuel}}} = \frac{P_{\mathrm{gt}}}{H_{\mathrm{fuel}}\,\dot{m}_{\mathrm{fuel}}} \tag{38a}$$

On the ground, the gas turbine efficiency drops with the decreasing power output during the climb. At high altitude, the proximity to the optimum operating point compensates this efficiency loss. The gas turbine efficiency stays almost constant until it is shut off after the cruise. The green curves show the total efficiency of the propulsion system

$$\eta_{\mathrm{total}} = -\frac{P_{\mathrm{load}}}{P_{\mathrm{fuel}}} \approx \frac{P_{\mathrm{gt}} + P_{\mathrm{bat}}}{P_{\mathrm{fuel}}} = \frac{P_{\mathrm{gt}} + P_{\mathrm{bat}}}{H_{\mathrm{fuel}}\,\dot{m}_{\mathrm{fuel}}}. \tag{38b}$$

Compared to the red curves of the single gas turbine efficiency we find: The total efficiency is lower than the gas turbine efficiency at times when the battery is loaded and higher at times when the battery contributes to the propulsion. This explains the high total efficiency in the Block mission that drops suddenly when the battery power drops to its cruise value as well as the low total efficiency in the Reserve mission. The total Block mission efficiency on the ground exceeds the gas turbine efficiency at $t = 945\,\mathrm{s}$ and stays higher by the amount of the battery contribution. If we bring the system to flight level, the efficiency is even higher as the gas turbine runs closer to its design point. This artificial advantage of flight over ground mode is most visible when the battery provides most power, i.e. at the end of the climb. The reserve mission mostly loads the battery while the gas turbine is running. Therefore, the total efficiency on the ground is lower than the gas turbine efficiency, except for the last part of the climb when the battery is being discharged. At flight altitude, the effect of lower fuel consumption appears additionally, see above. However, it is partly compensated by the gas turbine loading the battery for most times.

Please note two things: First, we only consider the output-input-balance in flight. This means that we start the flight mission with a charged battery and may end it with a lower state of charge. We do not take into account ground charging in Eq. (38b). Second, we did not design the battery capacity with a realistic aircraft in mind, for example limiting its weight. Both of these things would have to be done in a holistic consideration, that includes the airport infrastructure and aircraft efficiency. This is not our goal with ProCoSim.

The runtime of the simulation we performed was under $5\,\mathrm{min}$ or $4\,\mathrm{min}$ for the Block or Reserve mission with a time resolution of $\Delta t = 10\,\mathrm{s}$. The runtime drops to about $30\,\mathrm{s}$ for a time resolution of $\Delta t = 100\,\mathrm{s}$. This corresponds to a computation time per simulation time step of $0.5\,\mathrm{s}$ to $0.6\,\mathrm{s}$. The computations were performed on a 12 core Intel i7-1270P machine, [44]. Such a simulation provides all the overview results shown as well as a variety of detailed results, of which we have shown the electrical quantities in Fig. 16 as an example.

The results of the hybrid-electric aircraft propulsion prove that ProCoSim models the aircraft propulsion system as expected. The assumptions we made for the newly developed components are suitable to provide the results shown. The iterative solution of algebraic equilibria works smoothly. We have used it intensively in this application for each time step. The power comparison gives an overview of the system. We were able to simulate two basic operating modes and got a first impression: We expect to be able to realize the experimental investigation of operating modes with our setup. This shows: For our purposes, it was justified to simplify the system quite a bit compared to the spatially resolved detailed simulations of the project partners. As an example, we looked at the detailed results of the electrical components. We were able to show that such results give us an understanding of the individual components and their interdependencies. If unexpected events occur during the course of an experiment, ProCoSim can help to determine where the fault lies. With little effort, ProCoSim can calculate the expected behavior and the behavior if a certain input variable is faulty.

4. Conclusions

The design and implementation of the new simulation framework ProCoSim were presented. The approach to integrating ODE and algebraic solvers was described, together with the heuristic method used for the communication between component models and the solvers. This enables the integration of component models while imposing minimal requirements on the component model structure. Component models can be black-box models, equations describing their dynamics and algebraic constraints, or LUTs derived from other simulation programs and experimental data. They can have uni- or bidirectional connections to other component, while the component topology and information flow can change dynamically.

ProCoSim's functionality and flexibility were demonstrated by the implementation of two distinct application use cases: First, a decentralized building heating supply network including heat consumers, a thermal storage, and a micro gas turbine for combined heat and power production; and second, a hybrid-electric aircraft propulsion system, combining a gas turbine with a battery and an electric propeller to showcase system interactions during flight missions.

Hence, ProCoSim has demonstrated its applicability for different application cases, e.g. to derive load trajectories of a complex energy system, to emulate the demand of such a network in an experimental gas turbine test campaign, or to support the initial design of individual components, such as the optimal capacity of a storage component, or to analyze system interactions to improve operating strategies.

5. Outlook

We have created a fully functional simulator. But this is only the beginning. Our main task for the future is to apply the simulator to different systems and answer concrete questions. In addition, we have already planned some structural and programming enhancements.

We gave an idea what type of questions ProCoSim might answer in the future. For the TPDE application, we could add more MGT-CHPs to the system and optimize their efficiency and/or lifespan based on the operating mode and power split that we choose. We might also investigate more complex systems such as a combination of MGT-CHPs and heat pumps.

In the hybrid-electric aircraft propulsion application, we could systematically vary the power split between gas turbine and battery and investigate how different modes of operation suit different flight missions. We might analyze the total efficiency of the system when we assume altitude conditions and get an idea of how the gas turbine must be designed. We can also apply our test rig conditions to ProCoSim and determine the expected system behavior for a planned experiment beforehand. Besides this, we also want to investigate the Turboelectric configuration shown in Fig. 11 and emergency maneuvers as described in Section 3.2.4. For this, we might use a dynamic control that monitors the Gas Turbine efficiency and the Battery state of charge and divides the power accordingly. These are only a few of the many conceivable examples for fields of application for ProCoSim.

Beyond that, the simulator can be improved and extended. In the future, optimization problems shall be solved with automated parameter variation in the simulation process. Furthermore, ProCoSim shall be expanded by a graphical user interface in which the model system is visible. There, the user can design the system more intuitively and retrieve the results clearly assigned to the modules. The visual representation of the system quantities during the simulation supports the implementation and debugging stage and could also extend to online plotting options.

There are also new application areas for ProCoSim. These applications will extend the component and bus libraries and make the structure of the simulator even more flexible. It is conceivable to apply ProCoSim to systems that use hydrogen as an energy source or to other energy sectors such as maritime transport.

CRediT authorship contribution statement

Marlene Herbrik: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Martin Henke:** Writing – review & editing, Supervision, Software, Methodology, Funding acquisition, Data curation, Conceptualization. **Jan Zanger:** Writing – review & editing, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The original simulator development and the ProCoSim application to the TPDE system was supported by the Federal Ministry for Economic Affairs and Energy, Germany under the funding code 03ET7084. Further research such as the extension of ProCoSim for the hybrid-electrical application was financed by DLR, Germany internal funding. The authors are responsible for the content of this publication.

We thank our project partners and colleagues for their contributions to this work. Thank you to Dennis Kopljar and Max Henle from the DLR Battery Laboratory who kindly let us use their experimental battery cell data. Many thanks to Pascal Albrecht who introduced us to battery modeling and patiently answered any related question. Many thanks, Georgi Atanasov and Max Arzberger, for your kind input concerning aircraft configurations and mission data. Thank you to Anna Marcellan and Nils Jakobs who provided the gas turbine data with MGTS3, and to Thomas Krummrein for being an excellent and highly intelligent discussion partner in any issue. Finally, thank you, Timo Zornek, Simon Schuldt, Timo Lingstädt, Martina Hohloch and Oliver Kislat for your tips and contributions.

Data availability

The authors do not have permission to share data.

References

[1] Federal Ministry for Economic Affairs and Energy (BMWi). 7. Energieforschungsprogramm der Bundesregierung. BMWI Public Relations, 11019 Berlin: Federal Government of Germany; 2018, URL: https://www.publikationen-bundesregierung.de/pp-de/publikationssuche/7-energieforschungsprogramm-der-bundesregierung-1522080.

[2] The MathWorks I. Simulink. 2025, URL: https://mathworks.com/products/simulink.html.

[3] Modelica A. Modelica. 2025, https://modelica.org/.

[4] Lukaczyk TW, Wendorff AD, Colonno M, Economon TD, Alonso JJ, Orra TH, Ilario C. SUAVE: an open-source environment for multi-fidelity conceptual vehicle design. In: 16th AIAA/ISSMO multidisciplinary analysis and optimization conference. 2015, p. 3087.

[5] Reitenbach S, Schmeink J, Bröcker M, Nöthen M, Siggel M. GTlab. Zenodo; 2025, http://dx.doi.org/10.5281/zenodo.15129162.

[6] Klein S. TRNSYS 18: A transient system simulation program, solar energy laboratory, University of Wisconsin, Madison, USA. 2017, URL: http://sel.me.wisc.edu/trnsys.

[7] Siemens. Siemens amesim simulation framework. 2025, URL: https://plm.sw.siemens.com/de-DE/simcenter/systems-simulation/amesim/.

[8] GasTurb A. GasTurb 15. 2025, URL: https://www.gasturb.com/software/gasturb.html.

[9] Lang L, Jiang F, Cheng K, Liu Z, Wang S, Dang C, Qin J, Huang H, Zhang X. Dynamic characteristics analysis of supercritical CO2 closed brayton power generation system for hypersonic vehicles. Appl Therm Eng 2025;269:126016.

[10] Krien U, Schönfeldt P, Launer J, Hilpert S, Kaldemeyer C, Pleßmann G. Oemof.solph—A model generator for linear and mixed-integer linear optimisation of energy systems. Softw Impacts 2020;6:100028.

[11] Schönfeldt P, Schlüters S, Upadhaya A, Oltmanns K. Model template for residential energy supply systems (MTRESS). Zenodo; 2024, http://dx.doi.org/10.5281/zenodo.11205762, URL: https://doi.org/10.5281/zenodo.11205762.

[12] Wahler NF, Ma Y, Elham A. Conceptual design and aerostructural trade-offs in hydrogen-powered strut-braced wing aircraft: Insights into dry and wet ultra-high aspect ratio wings. Aerospace 2025;12(2):77.

[13] Krummrein T, Henke M, Kutne P. A highly flexible approach on the steady-state analysis of innovative micro gas turbine cycles. J Eng Gas Turbines Power 2018;140(12):121018. http://dx.doi.org/10.1115/1.4040855.

[14] Zimmer DD. Virtual physics equation-based modeling lecture. 2014, URL: https://rmc.dlr.de/rm/de/staff/extcms/images/rmc/users/zimm_di/lecture/Lecture4Handout%281%29.pdf.

[15] Henke M. Erstellung und Validierung eines modularen Simulationswerkzeugs zur Analyse des dynamischen Verhaltens von Mikrogasturbinen [Ph.D. thesis], Universität Stuttgart; 2018, URL: https://elib.dlr.de/123684/, VT-Forschungsbericht 2018-04.

[16] Agarwal S, Mierle K, Team TCS. Ceres solver. 2023, URL: https://github.com/ceres-solver/ceres-solver.

[17] Dawes B, Abrahams D, Rivera R. Boost C++ libraries. 1998, URL: http://www.boost.org/.

[18] The HDFGroup. Hierarchical data format, version 5. 2006, URL: https://www.hdfgroup.org/HDF5/.

[19] MATLAB. Version 9.14.0 (R2023a). Natick, Massachusetts: The MathWorks Inc.; 2023.

[20] ISO. ISO/IEC 14882:2020(E) – programming language C++. Geneva, Switzerland: International Organization for Standardization; 2020, p. 1853, URL: https://www.iso.org/standard/79358.html.

[21] Institute of Combustion Technology. Technology Platform Decentralised Energies (TPDE). 2019, URL: https://www.dlr.de/en/vt/research-transfer/research-infrastructure/technology-platform-decentralised-energies-tpde.

[22] Freepik. Pictures. 2025, URL: https://de.freepik.com/.

[23] Krummrein T, Henke M, Kutne P. A highly flexible approach on the steady-state analysis of innovative micro gas turbine cycles. J Eng Gas Turbines Power 2018;140(12):121018. http://dx.doi.org/10.1115/1.4040855.

[24] Henke M, Klempp N, Hohloch M, Monz T, Aigner M. Validation of a T100 micro gas turbine steady-state simulation tool. In: Turbo expo: power for land, sea, and air. Volume 3: coal, biomass and alternative fuels; cycle innovations; electric power; industrial and cogeneration, 2015, http://dx.doi.org/10.1115/GT2015-42090, V003T06A003.

[25] Hohloch M, Zanger J, Widenhorn A, Aigner M. Experimental characterization of a micro gas turbine test rig. In: Proceedings. 2010, URL: https://doi.org/10.1115/gt2010-22799, GT2010-22799.

[26] Ansaldo-Energia. AE-T100NG microturbine. 2024, URL: https://www.ansaldoenergia.com/fileadmin/Brochure/AnsaldoEnergia-Microturbine-AE-T100NG-20220907.pdf.

[27] Kleinbach E, Beckman W, Klein S. Performance study of one-dimensional models for stratified thermal storage tanks. Sol Energy 1993;50(2):155–66. http://dx.doi.org/10.1016/0038-092X(93)90087-5, URL: https://www.sciencedirect.com/science/article/pii/0038092X93900875.

[28] Kislat O, Römgens J, Schuldt S, Zanger J, Jakobs N, Henke M, Kraus C, Moosbrugger A, Asmi MA, Aigner M. Development and testing of a gas turbine test rig setup for demonstrating new aviation propulsion concepts. Aerospace 2024;11(7):534. http://dx.doi.org/10.3390/aerospace11070534.

[29] Hoidis J. A creative drive for the future of flight - bringing e-mobility to the sky requires whole-system thinking. DLRmagazine 2020;(164):16–9, URL: https://www.dlr.de/en/media/publications/magazines/2020_dlrmagazine-164/.

[30] Hoidis J, Hartmann J. Conceptual study for environment-friendly flight. 2020, URL: https://www.dlr.de/en/latest/news/2020/02/20200504_conceptual-study-for-environment-friendly-flight.

[31] Atanasov G. Comparison of sustainable regional aircraft concepts. In: DLRK 2022. 2022, URL: https://elib.dlr.de/193099/1/DLRK_2022_Atanasov_Sustainable_Regional_Aircraft.pdf.

[32] Atanasov G, Silberhorn D. EXACT sustainable aircraft concepts results and comparison. In: DLRK 2024. 2024, URL: https://elib.dlr.de/208693/.

[33] Atanasov G, Wehrspohn J, Kühlen M, Cabac Y, Silberhorn D, Kotzem M, Dahlmann K, Linke F. Short-medium-range turboprop-powered aircraft as a cost-efficient enabler for low climate impact. In: AIAA Aviation 2023 forum. 2023, http://dx.doi.org/10.2514/6.2023-3368, URL: https://arc.aiaa.org/doi/abs/10.2514/6.2023-3368.

[34] Balack P, Atanasov G, Zill T. Architectural trade-offs for a hybrid-electric regional aircraft. In: Aerospace Europe conference 2023. 2023, URL: https://elib.dlr.de/197732/.

[35] Arzberger MJ, Zimmer D. A modelica-based environment for the simulation of hybrid-electric propulsion systems. In: Proceedings of the 13th international modelica conference. Linköping electronic conference proceedings, Linköping University Electronic Press, Linköpings universitet; 2019, p. 471–80, URL: https://doi.org/10.3384/ecp19157471.

[36] Schröder M, Becker F, Kallo J, Gentner C. Optimal operating conditions of PEM fuel cells in commercial aircraft. Int J Hydrog Energy 2021;46(66):33218–40. http://dx.doi.org/10.1016/j.ijhydene.2021.07.099, URL: https://www.sciencedirect.com/science/article/pii/S0360319921027634.

[37] Marcellan A, Henke M, Schuldt S. A numerical investigation of the water-enhanced turbofan laboratory-scale ground demonstrator. In: AIAA SCITECH 2022 forum. 2022, http://dx.doi.org/10.2514/6.2022-0062, URL: https://arc.aiaa.org/doi/abs/10.2514/6.2022-0062.

[38] Walsh PP, Fletcher P. Gas turbine performance. Blackwell; 2004, URL: https://books.google.de/books?id=zxRdacCvjVsC.

[39] Rolls-Royce. M250/RR300 - 2024 first network directory. 2024, URL: https://www.rolls-royce.com/~/media/Files/R/Rolls-Royce/documents/civil-aerospace-downloads/helicopters/2024-m250-rr300-first-network-directory.pdf.

[40] Menrath M. Exterimentelle kennwertermittlung und systemanalyse bei hubschrauber-gasturbinen [Ph.D. thesis], Technische Universität München; 1989.

[41] Albrecht P, Bänsch C, Kopljar D. Empfindlichkeit der Auslegung von Hybrid-Elektrischen Flugzeugen gegenüber Verbesserungen in der Brennstoffzellen- und Batterietechnologie. In: DLRK 2023. 2023, URL: https://elib.dlr.de/200511/.

[42] ISO. Standard atmosphere. Geneva, Switzerland: International Organization for Standardization; 1975, URL: https://www.bibsonomy.org/bibtex/242453a3ba8ddd531d878706db3f2c80e/jrdn.

[43] Fritzsche F, Müller S, Atanasov G, Silberhorn D, Schneiders N, Zill T. Regional turboprop aircraft model as a benchmark for future concept studies. In: Deutscher luft- und raumfahrtkongress. 2023, URL: https://elib.dlr.de/208840/.

[44] Intel. Intel® core™ i7-1270p processor. 2022, URL: https://www.intel.com/content/www/us/en/products/sku/226255/intel-core-i71270p-processor-18m-cache-up-to-4-80-ghz/specifications.html.