1D SECONDARY AIR SYSTEM MODELING FOR APPLICATION IN ENGINE PREDESIGN AND MULTI-FIDELITY

D. Woelki*

* German Aerospace Center (DLR), Institute of Test and Simulation for Gas Turbines,
Am Technologiezentrum 5, 86159 Augsburg, Germany

Abstract

The secondary air system (SAS) is vital for the safe operation of aero engines and longevity of thermally loaded parts. The extraction of secondary air from the compressor, its transfer and provision as e.g. cooling and sealing air leads to manifold interactions on whole engine level. To enhance the evaluation of novel engine concepts with noticeable changes in secondary air requirements and supply concepts, it is obvious that the SAS must be integrated into the design process and in the mulfi-fidelity simulation processes.
With the focus on inter-disciplinary studies, three key features of SAS simulation are discussed: The first is the need for methods to derive a first SAS geometry at early design phases, which allows the realistic prediction of the mass flow distribution within the SAS. This particularly enhances cycle and turbine design. Second, the derivation of local pressures and temperatures in the inner SAS serves for the provision of boundary conditions for high-fidelity simulations. The third is full off-design capability, which allows the application of an SAS model within the scope of all relevant steady-state operating points. For this purpose, a 1D network simulation tool for holistic SAS modeling is going to be developed to extend the simulation capabilities.
After a brief review of existing approaches and software implementations for 1D SAS modeling, specific implementation requirements are discussed which can be categorized as follows. First of all, the 1D network simulation tool shall be generally applicable for studies with focus on preliminary engine design and providing boundary conditions for high-fidelity sub-models. It should also be applicable in processes like engine health monitoring. Further, all physics-based interfaces to adjacent components and simulation tools must be properly defined. The software implementation must provide flexibility of interfaces in order to allow multi-fidelity simulations. All of this results in special requirements for the software architecture which is presented and discussed. Finally, research questions for future enhancements of 1D SAS modeling are presented.

Keywords

secondary air system; engine predesign; multi-fidelity

NOMENCLATURE

$\dot{m}$	mass flow (kg/s)
$\underline{err}$	vector of residuals
h	specific enthalpy (J/kg)
p_s	static pressure (Pa)
p_t	total pressure (Pa)
$\dot{Q}$	heat flow (J)
r	radius (m)
SW	swirl (m^2/s)
T_t	total temperature (K)
v_{tan}	tangential velocity (m/s)
$\underline{x}$	vector of guesses

Subscripts:

i	index of network nodes
j	index of network elements
l	index of element fluid ports
$sink$	sink egress flow

Abbreviations:

B	Boundary
CFD	Computational fluid dynamics
CSM	Computational solid mechanics
DLR	German Aerospace Center
E	Element
I	Intermediate node
J	Junction
HPT	High pressure turbine
N	Node
OP	Operating point
SAS	Secondary air system
SG	Institute of Test and Simulation for Gas Turbines

doi: 10.25967/610187

1. INTRODUCTION

The SAS is a sub-system that is vital for the safe and long-lasting operation of aero engines. It supplies air which is extracted from the compressor to different consuming stations where it is primarily used for either cooling or sealing purposes. The requirements of secondary air set by these consuming stations will be subsequently referred to as *demands*. Cooling air demands are mainly defined by mass flow and fluid temperature. Sealing demands are particularly defined by the pressure in the sealing station. However, compressor bleed air, especially when extracted from the rear stages, implies losses and reduces the thermal efficiency. Therefore, all secondary air mass flows and pressure margins should be designed to the lowest possible level as long as compliance with all demands can be assured.
The operating points (OP) at which the demands become critical may differ for the individual consuming stations. Furthermore, the SAS boundary conditions are not scalable over the entire operating range. Hence, the design of the SAS strongly depends on the partly complex off-design behavior of the adjacent components, which define the thermodynamic boundary conditions of the SAS. At the same time, the condition of the SAS influences these components. In addition, the SAS forms large domains in the inner engine, connecting all interacting components.
The large number of flow paths and elements in the SAS, i.e. its configuration and geometry, as well as the variety of OPs make the design and evaluation of the SAS a complex endeavor. Thus, concept and design studies are accompanied by large parameter ranges and are usually part of a highly iterative process with the engine cycle and turbo components. This requires models which are suitable to represent the SAS holistically, but at low computational ressources - especially low simulation time. In this context, 1D models are state of the art. They connect parameterized SAS elements to networks of flow paths. Beyond the design process, those models are applied by the industry in different use cases of the product life cycle.

Although research with and further development of network models represent a niche, recent publications indicate a renewed interest in it. To provide some examples: Methods for automated design of SAS geometries have been continuously investigated for about the last decade, see e.g. [1] and [2]. Studies for the modulation of coolant in off-design have been performed in [3]. Developments for advanced methods are discussed in e.g. [4], here dealing with conjugate heat transfer in network models. The latest new development of a SAS network simulation tool, as to the author's knowledge, has been presented in [5]. Beyond, a review of currently implemented methods in another established tool has been provided in [6].

In their survey, Fuchs and Cordes [7] specified topics in SAS modeling, which are largely unexplored. Some of these topics were selected for experimental and numerical investigations at the Institute of Test and Simulation for Gas Turbines (SG) of the German Aerospace Center (DLR):

- de-swirled vortex reducers
- flow and heat transfer along co-rotating shafts
- flow and heat transfer in compressor drum cavities
- brush seals

The results of these investigations are, among other things, to be processed in the form of characteristics for use in 1D models. On the other hand, both experiment and high-fidelity methods such as CFD depend on realistic boundary conditions. In the cases presented, these correspond largely to local state conditions in the SAS, which in turn can be conveniently provided by a holistic 1D SAS model.

The development of the related simulation tool has already been announced in [7], the conceptual design of which is the central subject of this paper. After a description of the general functionality of network simulators and a short overview of available software, the most important fields of application at the Institute SG will be examined in more detail. This results in special requirements for the architecture and flexibility of the new software to be developed, which makes it possible to differentiate it from other tools. Based on these requirements, the main features of the concept for the implementation are presented. Finally, an outlook is given on further short-term development planned for the tool, first research applications and further research questions of interest.

2. OVERVIEW: NETWORK MODEL, SOLVER AND SOFTWARE

This section will first introduce the nature of network models as well as the differences to other common types of simulation. Subsequently, an overview to the common approach solving the numerical problem represented by the network model is given. In this context, this paper strictly differs between the term *simulation tool* describing the entire software, while the term *solver* is limited to the numerical part of the simulation. For completion, an overview of existing software is given.

2.1. 1D network model

In this paper, the term *holistic* refers to a model in which all relevant interdependent flow paths between the sources and sinks of the SAS are represented. Holistic refers mainly to the scope of the model, but also indirectly to the interactions with the adjacent components. Such a model does not necessarily have to include the complete SAS. Rather, it is common to break the SAS down into domains that can be delimited from one another as e.g. illustrated in Fig 1 with three domains: the high pressure turbine's

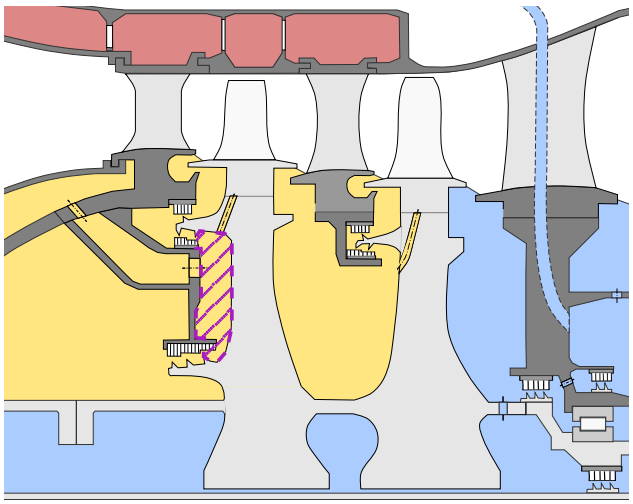


FIG 1. Suitable domains for 1D SAS modeling (colored) compared to limited sub-domains for preferred use of 3D CFD (hatched)

(HPT) casing (red), the HPT's rotor (yellow) and a part of the low pressure system (blue).
A common way of modeling SAS domains is using so-called network models. These are built from the secondary air flow paths which are merged and branched in junctions, cavities or slots. In turn, the flow paths are composed of one or more flow elements connected in series. The SAS contains various types of elements such as seals, orifices, nozzles, pipes or vortex generators. Most of these elements are modeled as *black boxes* with zero-dimensional character: Thermodynamic conditions at the inlet planes are computed to a set of corresponding outlet conditions by evaluating suitable characteristics or equations, which are specific for the different types of elements. The quantities at inlet and outlet planes are reduced to mean, scalar values. A discretization in meridional and circumferential direction is usually omitted, but would either way correspond to an internal sequence of black box models. Some elements or physical effects are also accounted for by surrogate models such for discrete losses, heat exchangers or vortices. More complex elements or cavities may be modeled by using a combination of these black box models.
As an example, Fig 2 shows the excerpt of a simplified network model of an HPT rotor. The flow paths, containing black box models, span the network between the sources or sinks of secondary air, subsequently both referred to as *boundaries* (marked with *B*). The resulting overall character of this modeling type is one-dimensional. This designation explicitly doesn't limit the model to a single spatial coordinate. For example, the consideration of swirl requires radial positions and injection angles. In general, the network model represents the geometry of the SAS or one of its specific domains. This also means that the modeled elements are usually parameterized geometrically. However, this is not mandatory and allows the use of surrogate models without a geometric

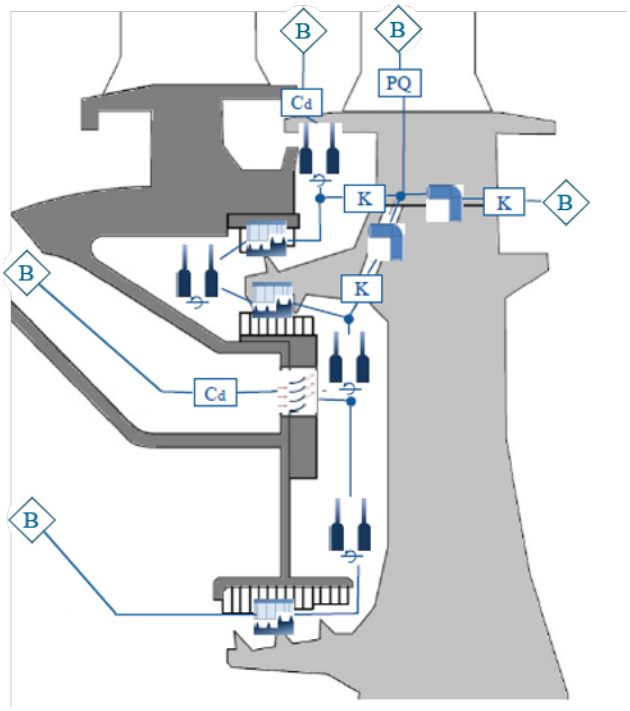


FIG 2. Simple SAS network model with flow paths and elements [3]

definition. In this case, no information on static flow quantities is provided. This assumption is particularly made in the modeling of rotor-stator cavities where the swirl pressure rise is calculated using models of free or forced vortices. Hence, the swirl pressure rise is attributed to the cavity's total pressure.
For many elements, characteristics or formulae are available in literature, e.g. [8] providing a still state-of-the-art compendium of hydraulic losses. Nevertheless, it is essential for complex configurations of elements or sub-domains to generate characteristics experimentally or with CFD - or to calibrate already existing. It also has to be noted that SAS geometry is often proprietary, making the modeling of a real engine's SAS difficult. An example for the generation of the network model of a generic stationary gas turbine is given in [9], providing both a guideline and information about further element characteristics.

2.2. Differentiation from other simulation types

Compared to high-fidelity models like in 3D CFD, the advantages of network models can be found in the size of modeled domains which may include several multiple cavities, interconnecting flow paths and flow restricting elements. The simulation times typically allow the numerical evaluation of the entire network for multiple OPs - or more generally: sets of boundary conditions - within one minute. Thus, network models are in particular effective when concept studies with large parameter ranges are conducted or when realistic physical quantities from a nearly arbitrary position within the SAS are required within seconds. The disadvantage compared to e.g. CFD is that no complex flow fields are analyzed. Overall, the network model

is strongly dependent on the existence of valid and accurate characteristics and computational methods for the elements' black boxes. Related to the typically modeled domains, CFD simulation on entire SAS domains like colored in Fig 1 will also remain too expensive for the time being. In contrast, the hatched area in Fig 1 represents a typical sub-domain of the SAS, for which the use of CFD is predestined.

Comparing SAS network models with other so-called low-fidelity models in the numerical landscape of gas turbine research and development, certain similarities with the synthesis models in engine performance are apparent. Performance synthesis are 1D models where the connected black boxes represent the engine's flow components, complemented by mechanical connections to e.g. shafts, see e.g. [10]. The flow connections are realized via stations in which the outlet quantities of a component are transferred to as inlet conditions to the adjacent, downstream component. Although prepared with special types of flow connectors, the complete SAS is modeled as black box as well. A highly detailed modeling of the flow paths within the SAS is not provided, nor is the consideration of specific, individual elements and their geometry.
In performance synthesis, the flow direction is clearly defined from the intake to the exhaust components like nozzles. Hence, the numerical consideration of flow reversal through the engine performance model is unusual. In contrast, in the SAS network model it is important to model local flow reversal as well. Whether a boundary on the consuming stations side, e.g., turbine annulus or bearing chamber, acts as a sink or source depends to some degree on the particular operating condition - so not every boundary can be considered a stable source or sink. One example is the common modeling approach of turbine rim seals setting two boundaries in the network to model the sealing quality by a local hot gas inflow (source) and a usually larger egress sealing air flow (sink). However, it is basically possible that due to a high driving pressure the local ingress will not occur, i.e. the source becomes a sink, or vice versa the egress flow collapses completely.

In summary, the SAS is investigated with all simulation types presented, with the level of detail required for the specific application being the major factor in the selection. The simulation types complement each other, but cannot be interchanged arbitrarily. For example, the integration of a simple network model into an engine performance model is basically possible and has been proposed in e.g. [11]. But for complex networks, the large number of elements and the detailed requirements as described above make the implementation in a stand-alone tool preferable - which is also pursued here.

2.3. Numerical problem

In order to derive a network model's state at a given set of boundary conditions, it is solved numerically. The common approach like suggested in [12] is to iteratively develop inner pressures and temperatures within the network in order to match the conservation equations for mass (Eq 1) and energy (Eq 2).

$$\sum_j \dot{m}_j = 0 \tag{1}$$

$$\sum_j \dot{m}_j h\left(T_j\right) + \dot{Q}_i = 0 \tag{2}$$

- $\dot{m}$: mass flow
- h: specific enthalpy
- T_t: total temperature
- $\dot{Q}$: heat input

The index j refers to the individual flow paths connected to a specific node. The index i denotes the nodes in which these flow paths are merged and branched. These nodes are typically assumed to represent plena with negligible flow velocity [13]. Thus, the total quantities $p_{t,i}$ and $T_{t,i}$ represent the solver's independent values which are arranged in the guess vector $\underline{x}$. It should be noted that $\dot{Q}$ is given in Eq 2 for completeness only. Adding heat directly within a node, unlike in elements, is not common.
If swirl (Eq 3) is considered, which is mandatory for the rotational domains, SW_i is an additional independent parameter type. Then, the conservation of momentum is an additional nodal dependent (Eq 4).

$$SW = v_{tan} r \tag{3}$$

$$\sum_j \left(v_{tan} r\right)_j \dot{m}_j = 0 \tag{4}$$

The numerical problem is non-linear. The right hand sides of the equations are therefore unequal to zero. They represent the residuals in each node i, which are reduced by the numerical solver and for this purpose are stored in a shared error vector $\underline{err}$.
Both $\underline{x}$ and $\underline{err}$ are properties of the numerical solver.

In the boundaries, pressure, temperature and swirl are fixed. Ultimately, both ends of every flow path are connected with nodes or boundaries which provide total pressures, temperatures and swirl at the inlet/outlet planes. The computation of the flow path - or strictly speaking: its elements - determines the mass flow as well as potential changes in swirl and temperature which are finally evaluated within the inner nodes.

Flow paths composed of multiple, serially arranged elements elements contain intermediate, unbranched nodes. These intermediate nodes can be either treated like common nodes. In this way, each ele-

ment would numerically represent an individual flow path. Alternatively, those flow paths can be treated as independent, embedded solution problems with the benefit that the intermediate nodes are no longer part of the superordinated system of equations.
Finally, it should be explained why p_t is chosen as boundary condition rather than $\dot{m}$. Within the SAS, it is likely that the critical pressure ratio is exceeded in certain flow cross sections. Then, the reduced mass flow would be maximal (hence constant), making the mass flow an unsuitable control variable.

2.4. Existing tools

Examples for different commercial tools are given in [4], [6] and [14]. Basically, all software provides full support of simulating arbitrarily defined steady-state OPs. To the author's knowledge, none of the referred tools supports a full transient simulation mode in terms of e.g. volume packing effects for a broad palette of integrated SAS elements. However, the selection from different incompressible and compressible fluids is state of the art. Partly, systems with individual circuits of incompressible and compressible fluids can be modeled and there are models which support interfaces of these circuits, like the bearing chambers in the SAS. Overall, the available software is suitable to analyze the usual conditions for the model of an existing SAS. This is adequate to verify the compliance with the demands in the consuming stations, to investigate interactions with other components, and also to perform simple concept studies without the need for major intervention in the configuration or geometry of the network.
However, the following options in particular are missing for the projects planned by DLR:

1) full flexibility and integrated methods for generating new configurations/geometries at
2) broad access to the code to extend the physics and numerics of the simulation tool if required, and
3) full flexibility in coupling with other simulation tools or dedicated frameworks, also independent of the workflow of coupling.

Based on these needs, the new network simulation tool shall be developed.

3. REQUIREMENTS

One of DLR's goals in aero engine research is the development of a virtual engine platform. The aerothermodynamic analysis of the SAS is intended to provide essential information for the design process, whereby the link with lifetime models is of particular importance. This requires new numerical process chains where different types of numerical models are considered. 1D network modeling as so-called low-fidelity method is therefore only one tool, which is, however, used among several high-fidelity methods like CFD or CSM in these process chains. The details of coupling 1D (network) with 3D (CFD) flow simulation is part of the larger topic multi-fidelity and will be discussed in another paper.
The two specific use cases preliminary engine design and engine health monitoring as one aspect of predictive maintenance are subsequently used to describe the general diversity of requirements of the new simulation tool. This is followed by an overview of interfaces, which are relevant for multi-fidelity modeling. Since the term interface is ambiguous, a strict distinction is made between

- model interfaces with physical *interface parameters* and
- *extension points* by means of software interfaces.

Because the selection of interface parameters and an effective approach for coupling the different simulation types is associated with particular challenges, this paper will only consider the extension points relevant for the 1D tool. Finally, a coarse overview of additional requirements is given.

3.1. Preliminary engine design

The major objective of 1D SAS modeling in preliminary design is to provide sound benchmarks for the amount and distribution of secondary air as well as coolant temperatures at disk surfaces and interfaces to turbine blades. This allows the early consideration of SAS associated losses on overall engine performance and supports part and component design.
Especially when investigating novel concepts, estimations based on empirical correlations will be outside the limits of their validity. Network models can be used flexibly. Furthermore, the distribution of secondary air to the individual sinks can be resolved at an appropriate level of granularity. Hence, providing a realistic flow path configuration that can be further approximated to a first SAS geometry is the implicit, albeit challenging, objective of preliminary SAS design. There are two basic methods for generating a SAS geometry, which are occasionally applicable depending on definite design constraints.

The first method is scaling an existing SAS geometry. Based on an already available SAS model, some parts are scaled while other geometries are set as degree of freedom. Examples for suitable variables are diameters of flow restrictors like orifices in order to adjust the associated mass flows. Of course, for all geometric variations, limitations resulting from design or manufacturing requirements must be taken into account, e.g. part collisions or minimum sizes.
This method is suitable when e.g. deviating a new engine with a different thrust requirement from an existing engine.
It is not necessary that the base geometry provides a high level of detail. Basically, the level of detail is low in conceptual design. Beyond, a highly detailed base geometry should not suggest high accuracy at an early design phase, where accuracy is neither required nor can be guaranteed. Finally, coarse geo-

metries are in tendency more suitable in early design phases because of their generic nature [15].

The second methodology is the configuration of a new network, nearly from scratch: For engine concepts, where no suitable base geometry is available, the generation of a new network of flow paths might be the only option. Given a desired distribution of secondary air, the most important flow paths could be generically linked to form an initial network. Similar to [2], the further modeling of the flow paths could be done with the help of parameterizable templates that represent common groups of SAS elements, cavities or even sub-domains. Again, the derivations of these templates must comply with certain geometrical restrictions and the approach explicitly needs no high level of detail.

Both suggested methodologies have a couple of specific requirements in common. First of all, they demand for element black boxes which must be as far as possible well parameterizable or scalable. Second, the network must not be a stiff model, hence providing options to dynamically re-configurate the flow paths and positions of elements at the study's runtime. Third, there must be a simulation mode which is suitable to modify arbitrary independent parameters - here quantities like diameters, radii, part numbers or even individual element characteristics.
The second approach or any kind of hybrid approach additionally requires a highly flexible solution to replace elements or entire sub-domains by alternative ones. Pre-defining the configuration and the detailed approach of modeling the individual flow paths, however, represents research questions on their own.

3.2. Engine health monitoring

One key aspect of engine health monitoring is the analysis of individual in-service engines and their components. It is a legitimate research question, which advantages the holistical consideration of the SAS - treated as engine component - can have in the scope of health monitoring.
Like every other flow component, the SAS is exposed to certain mechanisms of deterioration or damage. This applies to rotating seals, for example. The identification of reasonable and meaningful instrumentation within the SAS could be helpful to identify degraded parts. Above all, this enables the clearer differentiation of the current states of the turbo components with monitoring, since some effects caused by the SAS might to date have been assigned to other components, e.g. the turbine. In general, the precise localization and quantification of all part degradations increases the reliability and flexibility for scheduling shop visits, thus reducing off-wing time.
Vice versa, health monitoring can also be used to investigate the extent to which the SAS can comply with its demands at specific states of engine degradation.

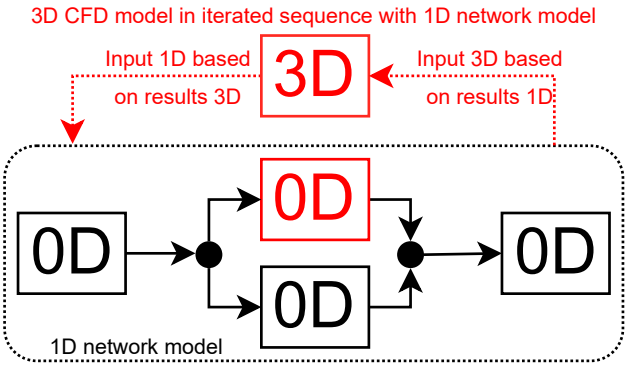


FIG 3. Network and CFD model running sequential, updating each others input until iteratively matched

In contrast to preliminary design, health monitoring depends on the existence of a detailed model which accurately describes the expected physical behavior in all flow paths. This results in a number of elements that can easily exceed 200 elements for the domain of a two-stage HPT alone, for example. On the one hand, this results in an increased demand on the robustness of the numerical solver. Furthermore and similar to preliminary design, studies for monitoring concepts will result in large parameter spans. This once again underlines the need for 1D SAS networks as fast simulation approach, though keeping in mind that broad parameter studies might come along with high overall simulation time. For this reason, it is important to also optimize the computation time of single simulations by means of one fixed geometry and one unique set of boundary conditions.

3.3. Multi-fidelity modeling

Network simulation and 3D CFD complement each other. While the 1D simulation can provide realistic boundary conditions for the CFD, the modeling of elements or sub-domains in the network can be improved by the results of dedicated investigations with CFD. Indeed, the coupling of a 1D network model with a 3D CFD model of a sub-domain, like sketched in Fig 1, is a perfectly illustrative use case of multi-fidelity.
There are two fundamental methods in which an externally simulated sub-domain can be considered by the network model. In this context, the term *external simulation* is always to be understood from the network simulation's point of view: any simulation that is from a different simulation type and thus performed with another simulation tool.

The first method is depicted in Fig 3 and represents the classic matching approach of two individual models. In the illustration, the 3D model is used to investigate the detailed physical behavior of one black box within the network model (marked red). Both models are arranged in a sequential process. The input parameters at the model interfaces - here the inlet and outlet planes of the marked box - are iteratively updated based on the results of the opposite model.

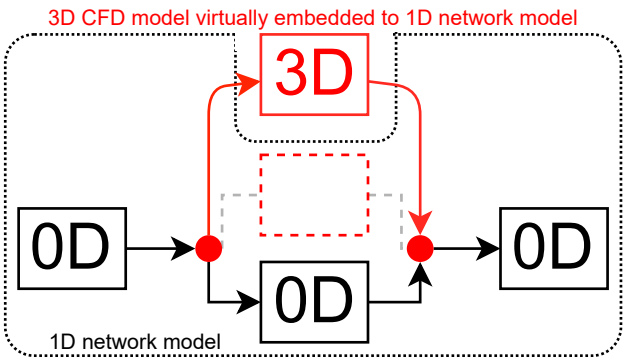


FIG 4. Blackbox in network replaced by CFD model and directly connecting via node

The requirement for the simulation tool is to provide extension points, which can

- transfer sets of network simulation results,
- receive result sets from external simulations to update the input of the network model and
- control simulation runs of the network model, especially by starting and waiting until receiving an appropriate return code.

In the second method, the 3D CFD model directly replaces the black box: The external simulation is embedded in the network model, acting like a virtual black box and thus keeping the interface parameters at direct control of the solver, Fig 4. Here, the requirement for the new tool is to design a flexible concept of extension points which allows the coupling of arbitrary external simulation software. At the same time, the basic structure of the network model must not be changed from the point of view of the numerical solver.

The variability of different use cases in terms of required external model types and approaches of coupling these models on process level makes the implementation of both methods mandatory. In this context, the requirements are not solely limited to the design of software extension points. Rather, the requirements must be reflected in the flexibility of the network model's structure as well.

3.4. Software extension points for external simulations

Extension points must be provided to allow both the transfer of interface parameters with and to control the network simulation from external tools. Such tools are primarily meant to be frameworks for collaborative projects. This requirement has to be initially implemented for coupling the tool with the DLR inhouse framework GTlab [16]. Embedding the network model into a serial work flow (Fig 3) is the minimal requirement and is realized by the implementation of specific interface-modules. For this purpose, the results of a network model simulation must be converted to the data model of GTlab and vice versa.

In an advanced phase, a generic extension point must be implemented which facilitates this way of coupling with arbitrary frameworks, too.
Furthermore, generic software extension points for simulation tools, which shall be embedded to the network model's solving process, must be provided (compare to Fig 4).

3.5. Additional requirements

The majority of simulations are expected to focus on steady-state OPs, compressible fluids and, of course, swirled flow. In common SAS, no supersonic flow is expected, so that gasdynamics can be reduced to subsonic and critical conditions. Full off-design capability must be ensured by means of arbitrary definition of boundary and ambient conditions as well as adaptable geometries and characteristics of elements.
While considering air as fluid is sufficient to start with, other fluids - incompressible, compressible, multiphase - are also important for future use cases. Also, extending models to capture more physical aspects must be feasible without sensitive deep intervention to the code architecture. For example, the implementation of selected transient effects such as volume packing is also conceivable.
The requirement for full off-design capability also means that a network model of a realistic SAS should feature a highly robust numerical behavior at all reasonable operating conditions. This also explicitly includes convergence in the case of local flow reversal. All convergence and termination criteria must be adjustable by the user while the return code follows established rules of numerical status indicators. Furthermore, the provision of suitable initial values for the numerical solver is crucial, i.e. the flexible integration of corresponding generators must be considered.
Nevertheless, it should be noted here that robustness cannot be guaranteed by the solver alone, but also depends on the characteristics of the elements used and thus ultimately on the user. Therefore, an extensive library of parameterizable SAS elements should be made available. This library should also be easily and arbitrarily extendable.
With regard to single flow paths or even sub-domains, it shall be tested, whether or not the solving as embedded problem (see Sec 2.3) provides benefits for robustness or computational performance. This in turn is related to another requirement to simply replace not only individual elements, but complete groups of elements with externally simulated flow paths or sub-domains, i.e. to exclude the associated black boxes from the network solving process (Fig 4). Since the models are potentially applied in broad concept studies, short simulation runtimes are mandatory, although a single simulation run is expected to only last a couple of seconds. When part of multi-fidelity studies, the tool should be also executable on high performance clusters, among other things.

4. CONCEPT AND ARCHITECTURE OF THE NEW SIMULATION TOOL

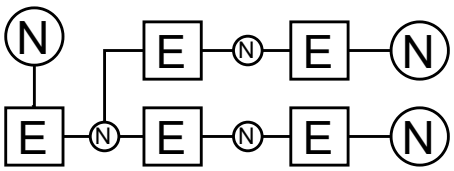
Although the suggested concept considers the complete list of requirements, this paper will focus on selected measures. The focus is subsequently placed on how the requirements for preliminary engine design and multi-fidelity can be met. The requirements for application in engine health monitoring are essentially covered by these measures.
For this purpose, the network is first abstracted to identify the basic object types as well as their interactions. Some examples are then used to further specify these types of network objects. On this basis, two independent concepts are presented: on one hand, the forming of interchangeable black box groups; on the other hand, the integration of extension points in the model. This results in a straight-forward class hierarchy.

4.1. Abstraction of and relation between network objects

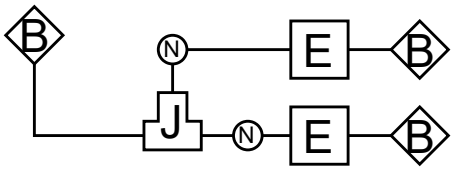
When looking at SAS networks, all objects can be broken down to two fundamental types.
The first type represents the *elements* (E) which are the already introduced black boxes containing specific characteristics. Nevertheless, these elements are all working in the same manner:

1) If there is the need to update specific input which is assumed constant for the simulated OP, it is set before the start of the simulation. After that the iterative part of the simulation starts.
2) The nature of fluid inlet and outlet planes is identical for all types of elements. Because the flow direction is undefined at simulation start, the terms inlet and outlet are subsequently avoided. Instead, the planes are for now referred to as *fluid ports* (index l).
3) The fluid ports are provided with the fluid composition and their recent local thermodynamic boundary conditions p_t, T_t, SW.
4) The computation is executed by means of evaluating the specific formulae or characteristics. This is also true when replacing the element by an external model like drafted in Fig 4. If p_s is required at the effective outlet, it is determined during the computation.
5) The major results, which are mass flow and the changes of temperature and swirl, are transferred to the adjacent network objects. If the latter are the nodes, in which flow paths are branched, these interface parameters are applied to the nodal equations of conservation.
6) Additionally, element specific results are available after element computation. These can be e.g. static quantities or non-dimensional quantities like Reynolds number or efficiencies.

The second base type of network objects represents the *nodes*. All types of nodes can be described as plena like introduced in Sec 2.3. The computational mode is not based on characteristics, but on the conservation equations only. This is implicitly true for the SAS boundaries as well - although not evaluated within the network model. Hence, p_t, T_t and SW are generally provided in another way than done for elements: Either they are part of the network model's input or they are provided by the numerical solver.
Figure 5a illustrates the radical abstraction of a network model to both types of network models (elements and nodes). For the sake of clarity, the nodes which are representing the SAS boundaries are depicted larger than the nodes within the network.



(a) Abstraction to base types element (E) and node (N)



(b) Extension with boundaries (B) and junctions (J)

FIG 5. Break down of network objects to certain types

Of course, it is reasonable to further specify the types of network objects. This is also necessary because of the various inputs required, which must be provided directly by the user or by another model. Furthermore, the system of equations is defined by the inner nodes, but not for the boundaries. Thus, the abstract node type is rather represented by boundary nodes (B) and the common, inner nodes (N).
The abstract element type is particularly the place holder for all types of SAS elements and substitute models (seals, pipes, swirl generators...). It is also important to consider that not all elements are limited to two fluid ports. For example, T-pieces are typical components in pipe systems and come with three fluid ports. In Fig. 5b, this is specified as junction type (J) with the special feature to branch flow paths while still representing an element. Although there are only a couple of common SAS elements which have more than two fluid ports, this feature is vital: An alternative modeling with three elements plus one inner node, as could be associated with the configuration in Fig 5a, neglects the dependence of the lateral branch on p_s at the junction point.
Furthermore, there are even elements with more than three fluid ports possible. For this reason, the simulation tool will provide not only the common element type with two and the junction type with three ports, but also a generic type with an arbitrary, user-defined

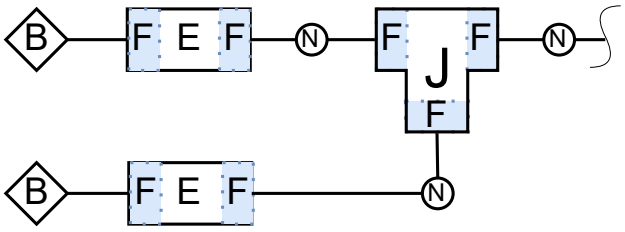


FIG 6. Uniqueness of network object interactions via fluid ports

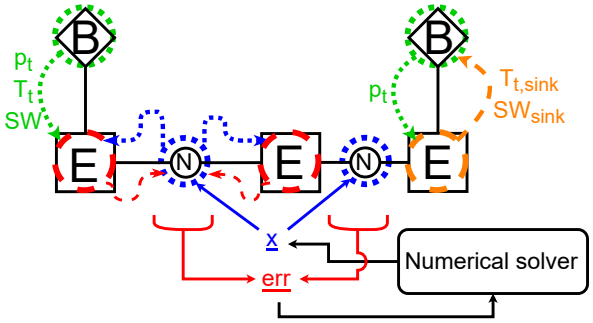


FIG 7. Parameter exchange between elements and node types

number of fluid ports that shall be called *generic cavity* element.

Finally, the fluid ports themselves are defined in more detail. In Fig 6, these are marked with F. Effectively, every fluid port is a non-replaceable part of an element and is hence a unique interface of the black box. For some elements, it can be important whether the flow through the port is inwards or outwards of the element, e.g. when comparing diffuser and nozzle. Furthermore, the concept of the fluid port only allows the connection to a single abstract node, which, however, can be arbitrary.
Hence, the fluid connections between all network objects - or in other words: the exchange of fluid parameters within the network - are clearly defined:

- Fluid ports are the interfaces of elements.
- Fluid ports establish an unique connection with one abstract node - no matter if a boundary or inner node.
- Nodes may establish connections to an arbitrary number of fluid ports - as long as not connecting to more than one fluid port of the same element.

Implicitly, the direct connection from one element to another (or one fluid port to another) is not allowed. Furthermore, there is no relationship between the actual elements and nodes. The existence of each other and therefore any interaction is unknown. On both sides, only the fluid ports are known.

4.2. Interaction with the numerical solver

Figure 7 illustrates, how the exchange of parameters between elements and internal/boundary nodes (Sec 2.3) is considered in the concept of the new tool. Of course, it applies to the figures and explanations that all transferred quantities still use the fluid ports as mediators.
The links between N and E visualize the transfer between elements and inner nodes. This also contains the interaction with the numerical solver, which provides the element's boundary conditions through the nodes ($\underline{x}$, blue). In fact, this is a query that starts from the elements via the fluid ports. This first assumes that $\underline{x}$ has been provided by the solver for the current iterative step. Then, for each element j, the steps introduced above can be performed: call the element's boundaries, perform the specific computation and finally return $\dot{m}$, T_t and SW to every node i (red). After all elements have been computed, the calculation of residuals (Eq 1, 2 and 4) and their storage to $\underline{err}$ is performed in the inner nodes. Finally, the numerical solver evaluates $\underline{err}$, checks for convergence (or other break criteria) and, if unconverged, prepares a new $\underline{x}$ for the next iterative step.
The links between B and E bring the flow direction into play for the first time. Like done with the inner nodes, the thermodynamic conditions in the boundary nodes are requested by all connected fluid ports at the beginning of each iterative step (green). Also, the fluid ports send $\dot{m}$, T_t and SW to the boundary nodes after element computation. This also underlines the equal processing for all specifications of the abstract node type. The only difference is that the network's thermodynamic boundary conditions must not be overwritten. Instead, additional result parameters of the egress flow (index *sink*) are written to the result list of boundary nodes which act as sink (yellow).

4.3. Groups of network objects: chains and subdomains

As introduced in Sec 2.1, some real SAS elements are modeled with a combination of different black boxes. Conversely, any flow path can be interpreted as an abstract component composed of several black boxes. At the same time, an entire flow path also satisfies one of the basic requirements of the basic element type, namely that fluid ports can be defined at both of its ends. Again, these fluid ports connect to each one node and mediate the parameter exchange. Furthermore, the order of elements is clearly defined within a flow path, so that the fluid ports can be identified to the outward facing ports of the end elements. It is now possible to express the flow path itself as abstract element, which will be referred to as *chain* in the following.
To meet the requirements of an element, the chain also needs a computation method. The idea of the chain is to be applied flexibly in the presented concept. Therefore, it cannot be assumed that a specific way of computation is known or that corresponding characteristics are available. The goal is also not necessarily to replace the associated flow path with a single autarkic component. Rather, the individual included elements are only virtually cut out, but remain

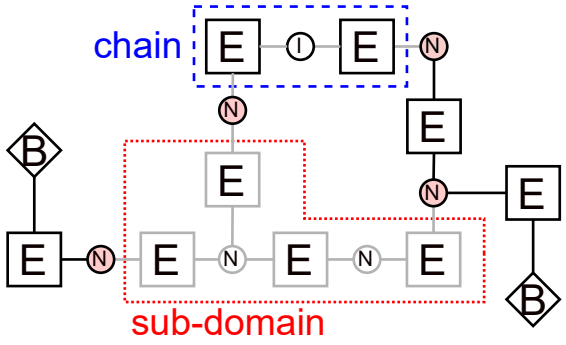


FIG 8. Groups of elements, replaceable in e.g. multi-fidelity simulations

available in the background. In this way, the nodes that are inside the chain, as well as the element black boxes, are not considered by the network's actual numerical solver, which is referred to as *global solver* in the following. The global system of equations is reduced thereby.
Now, the chain's inner nodes are specified to a third type of the abstract node: *intermediate node*. This has the same properties like the common inner nodes with the exceptions that they are limited to two fluid port connections and that they are ignored by the global solver. Instead, the chain now forms its own numerical problem to be solved. A separate numerical solver is used for this. Finally, the computation method of the chain involves performing the same iterative process as for the entire network, but limited to the elements and intermediate nodes of the chain.
This concept can be extended to more complex *sub-domains* in a similar way. Unlike chains, these sub-domains may also contain nodes where flow paths are branched. Thus, it is related to the generic cavity element with again a separate numerical solver.
Both types of element groups are depicted in Fig 8. They allow for the following options with benefits in certain use cases:

1) First of all, the groups are an elegant basis for being completely substituted in the context of multi-fidelity. This is explained in more detail hereafter.
2) For certain groups, characteristics could be derived in order to omit the simulation of the contained elements in the long-term, if necessary, i.e. to replace the group by an autarkic element.
3) Conversely, a single substitute model like a representative pressure loss can be used first, e.g. when generating a first geometry for a flow path in predesign. At a later stage, this can be easily replaced by a chain of adequate elements.
4) Finally, it is conceivable that the reduction of the global system of equations enhances robustness of the simulation and maybe even reduces simulation runtime.

4.4. Extension points and replacement of groups

For the realization of the multi-fidelity approach described in Fig 4, all inner nodes will be provided with extension points to external simulations. Those extension points are practically synonymous to the fluid ports of elements because they are a node's counterpart within the connection to an external model. On the side of the node, all processes remain unchanged. This means that the extension point requests the thermodynamic boundary conditions from the node in order to update inputs of the external simulation. At a certain point of time, depending on the exact coupling method, the extension point returns the appropriate parameters to the node where the residuals are calculated. It should be made clear that the extension points are themselves part of the 1D simulation tool and therefore also part of the network model.
In combination with the concept of fluid ports, which are also available in groups, embedding external simulations is not limited to autarkic elements. The extension point concept can be rather applied to defined groups while making the build-up of a second network unnecessary. For example, it is possible to build a new network model,which contains all elements as unique black boxes. The implementation of the group feature is now intended to label the elements contained in a chain or sub-domain by listing them - sort of like drawing a frame around the enclosed elements. This continuously ensures that the group feature can be deselected.
Finally, at the level of the network model, each group can be assigned an external simulation. The latter is embedded as a virtual element while the original black box or group is ignored by the network simulation process.

4.5. Class concept

For the purpose of implementing the concept at code level, Fig 9 illustrates the relationships between the configuration, the network objects and the base classes to be programmed.

- The configuration directly represents the real SAS with the network elements which are connected to flowpaths and finally connected through nodes to the network.
- The object layer clusters all network objects to abstract nodes and abstract elements, the latter including the fluid ports as implicit properties.
- On code level, the origin of each network object's implementation is one class. For all nodes, this is the AbstractNode class which is further specified to the dedicated classes of boundary nodes, inner nodes and intermediate nodes. For all elements, the abstract class NetworkEntity represents the base class.

At simulation runtime, after the network model has been initialized, all network objects are represented by each one instance of a specific class.
The introduced groups of elements (chains and sub-domains) have been described as abstract elements.

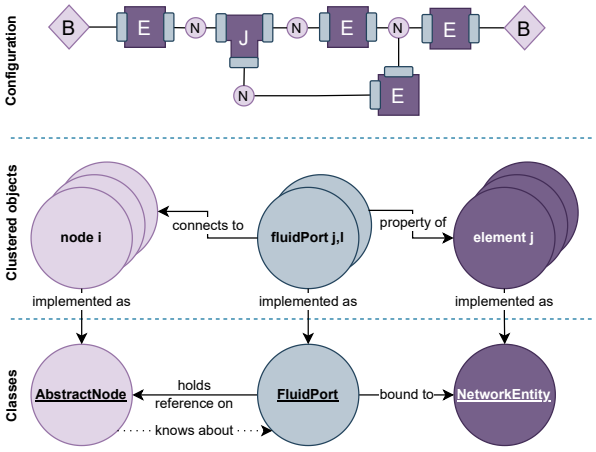


FIG 9. Relation between network configuration, clustering to network objects and classes

Finally, for the purpose of a clear class hierarchy, an adjusted terminology should be used. This will both highlight the common features as well as the differences between groups and single, autarkic elements:

- Autarkic elements are still named *elements*. Their specific classes inherit from the abstract class Element (Fig 10).
- The naming of chains and sub-domains will be shifted from elements to *groups* only. Their classes inherit from a suitable element, which fits best to the number of fluid ports. More important, they inherit from another abstract class EntityGroup which provides the special functionalities to integrate autarkic elements and nodes and which has the properties to be replaced by e.g. external simulations. If multiple inheritance is not supported by the programming language, one of the abstract classes would be replaced by an interface (expressed in the terminology of object oriented programming).
- Finally, both abstract classes - Element and EntityGroup - inherit from a new super-type, which is called NetworkEntity.

Figure 10 additionally provides insight to further specification originated from the class NetworkEntities. This new super-type puts more focus on the surrogative character of groups and has even more potential. For example, a network model could be configured on an initial level by only using placeholder objects which largely mimic the behavior of both abstract nodes and elements. In this way, a network can be modified by an external process (e.g. in preliminary design) or through a graphical user interface. All specific objects as well as groups are not finalized before starting the actual simulation of an OP. In principle, this concept even allows the transformation of entities in between OP simulations e.g. during broad concept studies.

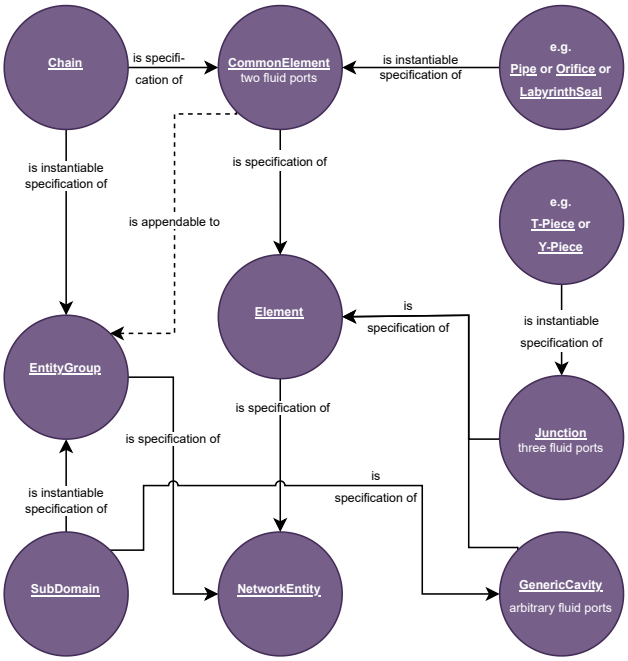


FIG 10. Class hierarchy for network entities

5. CONCLUSION AND OUTLOOK

The numerical assessment of the SAS will not only remain essential in almost all phases of the engine life cycle, but will also require the enhancement of the capabilities of corresponding simulation tools. In addition to 3D CFD, this includes 1D network simulations and the linking of both types of simulation. The presented concept for the definition of network models and the architecture of the associated simulation tool takes into account the currently set requirements and also remains flexible for future use cases and further development.

The implementation of this concept will immediately allow for the basic modeling of already existing SAS as well as for the design of simple SAS by hand. In combination with initial extension points to interface with the GTlab framework, collaborative, interdisciplinary studies will be feasible. At the beginning, these will likely be based on either known or - since the obtaining of original geometries is challenging - abstract, generic geometries.

Another important step towards the modeling of complex SAS will be the provision of a comprehensive library of SAS elements i.e. their characteristics or specific models. On the one hand, this will be based on literature. On the other hand, it will also include the findings from CFD studies running in parallel at the Institute SG. In addition, some of the sub-domains of the SAS which have been investigated with CFD and which were introduced at the beginning of this paper will be used as first demonstrators for multi-fidelity. This will also include the investigation of the different methodological approaches to link 3D CFD and network simulation.

Further research questions may arise in the mid-term from the possibilities of multi-fidelity, with the link to

lifetime models being of strategic importance as well. The application spectrum for the tool is thus explicitly not yet complete. Rather, it is also the intention to use this tool to identify new research questions in the context of SAS.

Contact address:

dominik.woelki@dlr.de

References

[1] T. Weiss, J. M. Rey Villazón, and A. Kühhorn. A parametric impact study of the geometry, air system and thermal boundary conditions on the life of a two-stage high pressure turbine. In *Proceedings of ASME Turbo Expo 2012*, 2012. DOI: 10.10115/GT2012-68453.

[2] D. Y. Kulkarni and L. di Mare. Virtual gas turbines part ii: An automated whole-engine secondary air system model generation. In *Proceedings of ASME Turbo Expo 2021*, 2021. DOI: 10.10115/GT2021-59720.

[3] V. D. Woelki. *Modulated Secondary Air Systems for Enhanced Off-Design Operation of Stationary Gas Turbines and Aero Engines*. PhD thesis, Technical University of Berlin, Berlin, Germany, Sept. 2022. DOI: 10.14279/depositonce-16063.

[4] D. Hunt and Y. Yuan. Modelling conjugate heat transfer within a gas turbine secondary air system using thermo-fluid system simulation. In *Proceedings of ASME Turbo Expo 2020*, 2020. DOI: 10.10115/GT2020-14202.

[5] A. Izadi, S. H. Madani, S. V. Hosseini, and M. Chizari. An integrated approach to simulate gas turbine secondary air system. In *Proceedings of ASME Turbo Expo 2020*, 2020. DOI: 10.10115/GT2020-14966.

[6] M. Kocagul, A. C. Arikan, O. Uyav, A. Ertas, J. Bruns, and A. Jayanthi. Gas turbine secondary air systems modeling. In *Proceedings of ASME Turbo Expo 2022*, 2022. DOI: 10.10115/GT2022-84308.

[7] F. Fuchs and L. Cordes. Secondary air system: A current review and future investigations. In *Proceedings of ASME Turbo Expo 2022*, 2022. DOI: 10.10115/GT2022-81262.

[8] I. E. Idel'chik. *Handbook of hydraulic resistance. Coefficients of Local Resistance and of Friction.* The U.S. Atomic Energy Commission, 1960.

[9] D. Woelki, J. Foret, L. Dittmar, M. Kanik, and D. Peitsch. Studies on part load controlled cooling air supplies in stationary gas turbines. In *Proceedings of Global Power and Propulsion Society Conference Beijing 2019*, 2019. DOI: 10.33737/gpps19-bj-221.

[10] R.-G. Becker, F. Wolters, M. Nauroz, and T. Otten. Development of a gas turbine performance code and its application to preliminary engine design. In *Deutscher Luft- und Raumfahrtkongress 2011*, 2011.

[11] A. Alexiou and K. Mathioudakis. Secondary air system component modelling for engine performance simulations. *J. Eng. Gas Turbines Power*, 131(3), May 2009. DOI: 10.1115/1.3030878.

[12] K. J. Kutz and T. M. Speer. Simulation of the secondary air system of aero engines. *J. Turbomach.*, 116(2), April 1994. DOI: 10.1115/1.2928365.

[13] Y. Muller. Secondary air system model for integrated thermomechanical analysis of a jet engine. In *Proceedings of ASME Turbo Expo 2008*, 2008. DOI: 10.1115/GT2008-50078.

[14] A. Spille-Kohoff, F. Hetze, and B. Du Toit. Transient cfd co-simulation of a 3d compressor model in its 1d system environment. In *Proceedings of ASME Turbo Expo 2019*, 2019. DOI: 10.1115/GT2019-90426.

[15] S. Bretschneider. *Knowledge-Based Preliminary Design of Aero-Engine Gas-Generators*. PhD thesis, Universität Stuttgart, Stuttgart, Germany, 2011.

[16] S. Reitenbach, M. Vieweg, R. Becker, C. Hollmann, F. Wolters, J. Schmerink, T. Otten, and M. Siggel. Collaborative aircraft engine preliminary design using a virtual engine platform, part a: Architercture and methodology. In *AIAA Scitech 2020 Forum*, 2020. DOI: 10.2514/6.2020-0867.