The Jet Propulsion Library: Modeling and simulation of aircraft engines

Michael Sielemann[1] Anand Pitchaikani[2] Nithish Selvan[2] Majed Sammak[3]

[1]Modelon Deutschland GmbH, Germany, michael.sielemann@modelon.com
[2]Modelon Engineering Pvt. Ltd., India, {anand.pitchaikani,nithish.selvan}@modelon.com
[3]Modelon AB, Sweden, majed.sammak@modelon.com

Abstract

The Jet Propulsion Library is a new Modelica library that provides a foundation for modeling and simulation of jet engines, and the model-based design of integrated aircraft systems. It provides a fully rigorous foundation for sizing and performance computations, and provides a number of advantages over existing domain-specific solutions due to the use of the Modelica language. This paper provides an introduction and overview of the library and describes an application in the design of a turbo fan engine.

Keywords: Turbo fan, turbo jet, turbo prop, turbo shaft, performance, model-based design, sizing, secondary power

1 Introduction

The prime mover of an aircraft, the jet engine, is one of the most important subsystem of an aircraft. Jet engines provide primary power (thrust) and secondary power (to drive flight control, air conditioning, cabin lighting and so on) to the aircraft. Recently, the improvements in the performance and efficiency of jet engines deliver a very large share in the overall platform improvements (for both commercial and military aircraft where for instance super cruise requirements were met). They therefore strongly affect aircraft value and, in case of commercial aircraft, eventually airline competitive edge. The latter is not only driven by costs, but even more so by environmental regulations.

However, these power plant improvements become increasingly difficult to achieve when focusing on the engine in isolation. The reason is that the local improvement potential has largely been leveraged in previous incremental design improvements, and only changes on global aircraft level remain to substantially improve the total aircraft package. For this reason aeronautical systems such as aircraft and their subsystems are becoming more and more integrated. This integration takes place along a number of trends. We mention two of these. The first one is the electrification of secondary power on-board aircraft (Provost, 2002). This trend is also called the "More Electric Aircraft" and has shaped industry road maps since more than two decades. Historically, three different types of secondary power were equal, namely, electric power, hydraulic power, and pneumatic power. With the "More Electric Aircraft" this is changing in favor of electric power. The main reason lies in the anticipated development potential of power electronics, which is all but exhausted (like that of pneumatic and hydraulic power).

The second trend is more recent and is the electrification of primary power. This is getting increasing interest due to intrinsic limitations in turbofan technology (Kyprianidis et al., 2014) (be it geared or ungeared). Following (Winter, 2013), the overall efficiency of a propulsion system can be considered to be proportional to the product of thermal and propulsive efficiency. To achieve thermal efficiency improvements, the Overall Pressure Ratio (OPR) and the Turbine Entry Temperatures (TET) of the cycles are being increased in an incremental way since the last few decades and are approaching peak values (approximately 1900-2000K TET and around 45-50 cycle OPR). Material limits, turbine cooling, emissions, and losses in the last stage of the high pressure compressor may now impose fundamental limits to the thermal efficiency. Improvements in propulsive efficiency are well achievable via reduction in fan pressure ratio and increases in bypass ratio. However, these improvements are deteriorated by losses through lowered transmission efficiency, increased nacelle weight and higher drag due to larger frontal area (Larsson et al., 2011). When these limits indeed turn out to become fundamental ones, different and more integrated concepts will become of interest. For instance electric ones where power is stored in one way or another on-board and possibly converted to electric power by gas turbines or fuel cells and used to drive distributed propulsion devices. Such aircraft with partially or fully electrified primary power systems are called hybrid or fully electric aircraft.

It is critical that the methods and tools supporting the design of such systems keep pace with the increasing integration on the product side. Only with efficient and robust prediction capabilities it becomes possible to establish model-based design for such solutions introducing new technologies, and cover all relevant "what if" scenarios.

It is therefore evident that if future propulsion systems and technology are to achieve the environmental challenges and performance targets set, rigorous mathematical

DOI
10.3384/ecp17132909

analysis of the component physics as well as integrated subsystem physics is important. There exists a critical requirement to develop and adapt models at an appropriate level of fidelity to specific components. One of the key requirement is also to have a generic framework where the user will be able to choose components and characteristics of his choice before integration of the subsystem.

Given the importance of propulsion system simulation as an academic and industrial engineering discipline, the literature on the state of the art is too extensive to be reviewed here. We therefore focus on selected references and tools. First, Gasturb (Kurzke, 1995) is a user-friendly and powerful domain-specific system simulation software for gas turbines. It is mature and provides extensive functionality, but also restricted to the scope defined by the authors. The model equations as implemented in the software can hardly be accessed and adapted by the user. Integration with other simulation and design models is possible but mostly requires process integration and design optimization (PIDO) solutions[1]. EnViRonmental Assessment (EVA) (Kyprianidis et al., 2008) is an example of a domain-specific simulation software with widened scope (engine system simulation and some aspects of aircraft sizing). Similar to Gasturb it is restricted to the application scope envisioned by its original programmers however. Numerical Propulsion System Simulation (NPSS) (Nichols and Chamis, 1991; Lytle, 1999; Jones, 2007, 2010) covers more than system simulation, as it also works as integration hub between system and field simulation. It can also be labeled as a domain-specific software but it relies on an object-oriented (yet causal) custom language in which component models are written. Model equations as implemented can be accessed and adapted by the user. Integration can also be established via PIDO solutions, but alternatively non-propulsion subsystems can be modeled in the native NPSS language and be integrated in a computationally more efficient and robust manner than via PIDO solutions. PRopulsion Object Oriented SImulation Software (PROOSIS) (Alexiou and Mathioudakis, 2005; Bala et al., 2007) is a similar system simulation software. It provides the same benefits in terms of access and customization (albeit using an acausal language), and also allows integration using non-PIDO approaches. A number of modeling libraries for non-propulsion sub-systems have been mentioned informally but not documented in the literature (according to the knowledge of the authors). In any case, a main limitation of this platform is the use of an in-house modeling language, which is not widely adopted or openly standardized via a non-profit organization.

While connecting a wide array of tools for multidisciplinary design optimization of aircraft and subsystems is feasible, integrating in a less fragile way based on open standards such as Modelica and FMI would increase flexibility and allowed to substantially increase manageable problem size due to higher computational efficiency. Other proposed interfacing standards such as ARP 4868 (SAE, 2001) are domain specific and work well for model-based efforts in their respective disciplines but not to couple analyses for unconventional designs. However, up to now nobody has proposed a plausible modeling library in Modelica for jet engines. Such a library could eventually be integrated one into a framework for modeling and simulation of aircraft and their components. This enabled time and resource efficient implementation of model-based design processes via reuse of such model assets; a key enabler for model-based design. Additionally, this improved consistency of results.

The objectives of this paper are

1. To suggest a library for modeling and simulation of aircraft jet engines and their sub-systems in Modelica for a broad range of applications ranging from engine and sub-system conceptual design to detailed analysis and design involving transient and real-time simulation[2].

2. To substantiate why the library can plausibly be applied to industrial-scale problems involving “complex” models and “sophisticated” analyses

3. To apply the framework to an engineering problem, namely, the computation of the full range of cycle performance.

4. To provide an outlook on how a Modelica implementation for modeling and simulation of aircraft jet propulsion provides additional value over existing discipline-specific tools in the design and analysis of unconventional systems.

2 Jet Propulsion Library: Overview and implementation

The Jet Propulsion Library is a modeling framework for gas turbines and jet propulsion of commercial and military aircraft. The comprehensive set of components enables cycle performance analysis and optimization of all types of aerospace gas turbines. On-design and off-design performance can be studied as well as steady-state and transient behavior based on a single model.

The physics of jet engines are governed by the balance equations of thermo-fluid dynamics, the conservation equations of mass, energy and momentum. Thus, some

[1] Process Integration and Design Optimization software typically contains numerous CAD/CAE integration adapters that allow the user to link different computation software in a GUI. They often also provide convergence, optimization, and surrogate modeling functionality, which allows to automate analysis and design processes.

[2] We restrict the scope however to only cover system simulation, i.e., all processes governed by ordinary differential equations (ODE) or differential-algebraic equations (DAE). Processes governed by partial differential equations are beyond the scope of the library, unless their partial derivatives have been suitably discretized to match the formal framework of an ODE or DAE (e.g., one-dimensional discretization of the balance equations of thermo-fluid dynamics).

DOI
10.3384/ecp17132909

of the underlying principles have been documented elsewhere (Elmqvist et al., 2003; Casella et al., 2006; Franke et al., 2009a,b). These will not be repeated here. However, given the flow velocities in such engines some assumptions commonly made in the mentioned articles are not appropriate; for instance the assumption that the flow velocity is small and that differences between static and total quantities can be neglected. The following presentation describes some of the differences to the established approaches, and gives a small example of the sophistication in its implementation to address the previously mentioned challenges.

2.1 Why Modelica

At a first glance, domain specific simulation solutions including sophisticated graphical user interfaces are very appealing. We believe however that the use of the generic modeling language Modelica provides advantages that may outweigh the benefits of the former.

First, this is due to the tool support to manage product and model complexity. This relies on the object-oriented nature of Modelica, and allows the tool to conveniently filter what implementations fit in a placeholder on a given model template. Manually choosing from a large library can be surprisingly difficult as industrial size problems are tackled. With Modelica, models can be built rapidly based on pre-configured templates. Additionally, a model architecture can be used once implemented across the system engineering V-cycle even as the user zooms into detailed modeling involving dynamic and real-time analyses. This facilitates creating and maintaining a holistic view even on challenging systems.

Second, given the declarative and symbolic problem description encoded in the Modelica language, a model compiler can transform the model description (equation system) into the form most suitable for a given analysis. This is based on automatic symbolic transformations, and allows executing the same model as dynamic simulation, steady-state simulation, optimization, real-time simulation and so on.

Additionally (and this has already been indicated above), using Modelica it is more straight forward to cover all domains based on first principles. After all, Modelica is one of the native languages of the aircraft sub-system industry. A large community/eco-system exists based on the Modelica language with many commercial and open source model libraries.

Furthermore, interactions become more productive. Based on the open standards, any given model can be made available on multiple tools. This enables model-based collaborations, independently whether based on Modelica or FMI.

Finally, this approach provides full access to the models. After all, while complete documentation of black box component models is great for many cases, reading the actual model code including the exact equations used for simulation in the engineering language Modelica enables deeper understanding and customization.

2.2 Thermodynamic properties

In the following sections, the distinction between so-called static and total quantities is very important. A static pressure p_s for instance is the actual pressure in the usual sense, which is associated not with fluid motion but with its state. Total and dynamic pressure in turn are closely related to fluid flow, and are a measure of flow velocity. For incompressible fluids, Bernoulli's equation states $p_t = p_s + p_d = p_s + 1/2\rho v^2$. Here, $p_d = 1/2\rho v^2$ is called the dynamic pressure, and p_t total pressure. Total quantities are sometimes also called stagnation quantities as they correspond to the value of the static or thermodynamic quantities if the fluid flow was brought to rest (zero velocity) in a reversible way (isentropically). As we are dealing with compressible fluids in the context of this paper, Bernoulli's equation does not hold. Instead, a compressible formulation has to be used. The details are described in the following sections.

The thermodynamic state is always defined by the static properties such as static temperature and pressure. These are the actual temperatures and pressure observed in the real world. In gas turbine performance computations it is however a tremendous simplification to express the component level equations mostly in total or stagnation quantities (Walsh and Fletcher, 2004). Like this, the exact flow cross section areas and velocities are not necessarily required. There are however also component models, in which the static quantities have to be computed such as mixers and nozzles (Walsh and Fletcher, 2004). In many cases the static quantities are also of interest and are therefore computed in the "output section" (using Modelica parlance).

In any case, the scope of the thermodynamic property computations in the Jet Propulsion Library therefore has to cover both static and total quantities. Additionally, to ensure accurate predictions, this has to be done in what is called the "fully rigorous" way (Kurzke, 2007). From text books, one is tempted use the following equation to relate the total temperature T_t and pressure p_t

$$T_t = T_s \frac{p_t}{p_s}^{\frac{\gamma-1}{\gamma}} \tag{1}$$

Or, likewise

$$\frac{p_t}{p_s} = \left(1 + \frac{\gamma-1}{2} M^2\right)^{\frac{\gamma}{\gamma-1}} \tag{2}$$

However, the isentropic exponent γ is not constant across larger temperature or pressure ranges. Therefore, equations (1) and (2) are strictly speaking not applicable. Following (Kurzke, 2007; Sethi, 2008), a fully rigorous approach based on the so-called entropy function Φ can be used instead.

$$\Phi(T) = \int_{T_{ref}}^{T} \frac{c_p}{R} \frac{dT}{T} \tag{3}$$

Then, the change of the entropy function in an isentropic process is equal to the logarithm of the pressure ratio,

$$\Phi_2 - \Phi_1 = ln\left(\frac{p_2}{p_1}\right) \tag{4}$$

Based on this approach, we can compute the complete set of the following six static quantities from any two of them plus the complete set of total quantities,

- Mass flow rate w
- Cross section area Ae
- Static pressure p_s
- Static temperature T_s
- Mach Number M
- Flow velocity v

Then, instead of using (2), we can compute the static pressure (and the complete set of static quantities) from the Mach Number as follows (Sethi, 2008) (note that this procedure requires the solution of implicit equation systems and additionally the mass flow rate as input). First, we compute the static temperature T_s from the following implicit equation.

$$M = \frac{\sqrt{2h_t - h_s\left(T_s\right)}}{\sqrt{\gamma_s\left(T_s\right)RT_s}} \tag{5}$$

Then, the static pressure can be computed explicitly in a fully rigorous way via the following equation

$$p_s = \frac{p_t}{exp\left(\frac{\Phi_t - \phi(T_s)}{R}\right)} \tag{6}$$

As written above, the complete set of six static quantities can be computed from any two of them (and the total quantities). Based on which set of two static quantities is given, between zero and two numerical solutions to implicit equations such as (5) are required to compute the full set of static quantities. Therefore, the thermodynamic properties involving rigorous computations of total and static quantities are somewhat different to the state of the art in Modelica (see references above), where the need for solution of implicit equation systems is considered a rare case, which can often be avoided by suitable model reformulations.

To provide convenient access to the computation of total and static thermodynamic properties we have therefore decided to use a package structure similar to Modelica.Media (Elmqvist et al., 2003) but tailored to the application specifics. First, we apply the concept of the thermodynamic state record to both total quantities (which, following the introduction to this section, are required in all component models) and static quantities.

A typical function to compute a total thermodynamic state record has the following interface.

```
replaceable partial function setTotal_pthtX
  "Return total state as function of pt, ht
       and composition X"
  input AbsolutePressure pt
          "Total pressure";
  input SpecificEnthalpy ht
          "Total specific enthalpy";
  input MassFraction X[nS]
          "Mass fractions";
  output TotalState total
          "Total state record";
end setTotal_pthtX;
```

Based on a given total thermodynamic state record any total quantity can be computed, for instance total temperature

```
  TtIn = Medium.totalTemperature(
      inlet_total);
```

Additionally, a static thermodynamic state record can be computed from a given total state record and any two quantities related to the static quantities (w, Ae, p_s, T_s, M, v as defined above). The rigorous procedure described with (5) and (6) is for instance implement in such a function conforming to the following interface

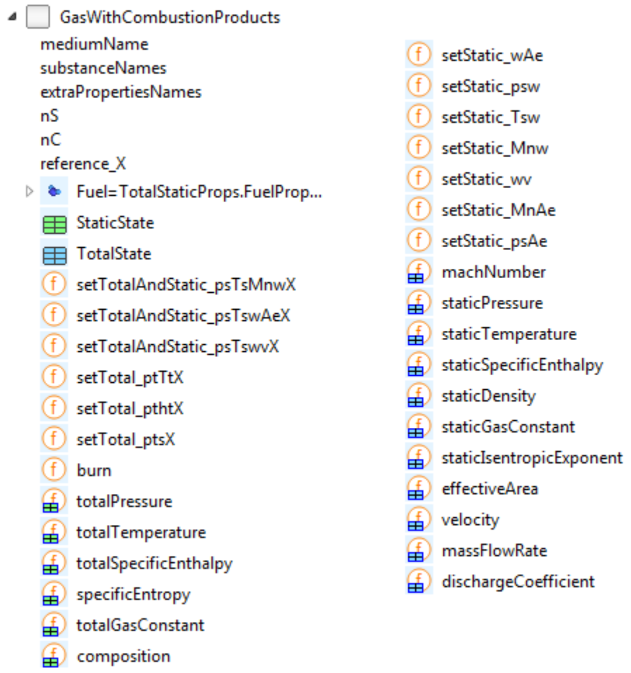
```
replaceable partial function setStatic_Mnw
  "Return static state as function of total
      state, Mach Number Mn and mass flow
      w"
  input TotalState total
          "Total state record";
  input Real Mn
          "Mach Number";
  input MassFlowRate w
          "Mass flow rate";
  input Types.FlowRegime regime
          = Types.FlowRegime.Subsonic
                "Flow velocity regime";
  output StaticState static
          "Static state record";
end setStatic_Mnw;
```

Enumeration FlowRegime is optionally used to constrain the solution interval to sub-sonic, sonic, or supersonic results.

Based on this code structuring concept fully rigorous thermodynamic properties are implemented in the Jet Propulsion Library. See figure 1 for an overview. The underlying model for the entropy function and other related quantities can be exchanged to allow different representations and fidelity levels. The first one implemented in Jet Propulsion Library utilizes the polynomial approach of (Walsh and Fletcher, 2004) and does not capture dissociation effects.

2.3 Connector definition

For the fluid connectors in the Jet Propulsion Library we adapt the concept of stream connectors as proposed in (Franke et al., 2009a,b). As defined there, the static pressure and the static specific enthalpy are used as key connector variables. Given the introduction to section 2.2 we instead opt for using the corresponding total quantities



GasWithCombustionProducts
mediumName
substanceNames
extraPropertiesNames
nS
nC
reference_X
Fuel=TotalStaticProps.FuelProp...
StaticState
TotalState
setTotalAndStatic_psTsMnwX
setTotalAndStatic_psTswAeX
setTotalAndStatic_psTswvX
setTotal_ptTtX
setTotal_pthtX
setTotal_ptsX
burn
totalPressure
totalTemperature
totalSpecificEnthalpy
specificEntropy
totalGasConstant
composition

setStatic_wAe
setStatic_psw
setStatic_Tsw
setStatic_Mnw
setStatic_wv
setStatic_MnAe
setStatic_psAe
machNumber
staticPressure
staticTemperature
staticSpecificEnthalpy
staticDensity
staticGasConstant
staticIsentropicExponent
effectiveArea
velocity
massFlowRate
dischargeCoefficient

Figure 1. Thermodynamic property functions: Static and total state records plus functions acting on total state records to the left and function acting on static state records to the right

on the connectors. Otherwise, the connector is identical to the well-established and widely adopted fluid connectors. The fluid connector carries flow and thermodynamic state information such as pressure, mass flow rate, specific enthalpy, and composition.

```
connector FluidPort "Fluid connector"
  replaceable package Medium =
    GasWithCombustionProducts
    annotation(choicesAllMatching=true);

  AbsolutePressure p
          "Total pressure";
  flow MassFlowRate m_flow
          "Mass flow rate into the
              component";
  stream SpecificEnthalpy h_outflow
    "Total specific enthalpy of exiting
        fluid";
  stream MassFraction X_outflow[Medium.nS]
    "Mass fractions of exiting fluid";
  stream ExtraProperty C_outflow[Medium.nC]
    "Properties c_i/m in the connection
        point";
end FluidPort;
```

Note how the Modelica naming convention is used for the variables on the fluid connector.

Unfortunately the connector is still not directly compatible with libraries using the standard fluid connector (e.g., from Modelica.Fluid) due to the use of a different package structure for the computation of the thermodynamic properties. Therefore, a simple adapter component was required if connections were to be made to the high speed gas flow path models; for all other interfaces standard connectors are used (fuel flow supply, shaft interfaces etc.).

2.4 Simulation modes: On-design, off-design, transient

Since the beginnings, jet engine performance computations have always considered two main computation problems, design point performance computation (also called on-design performance computations) on one hand and off-design performance computations on the other (Walsh and Fletcher, 2004).

For the design point performance computation, one set of operating conditions has to be imposed. Then, the component performance levels and sizes are selected. Additionally, top level requirements are implemented (e.g., based on cruise at altitude on an ISA day). The design point performance computation then allows to compute important cycle parameters, and to define a specific design. This includes a possibly abstract or estimated engine geometry, based on the fidelity of the analysis. Technically, the output of such a design point performance computation are however scaling parameters on component level.

Given a specific engine design (figuratively in terms of an estimated or abstract geometry, or, more technically, in terms of a complete set of component scaling parameters for a given engine topology), the off-design performance computation then allows to estimate the performance at other key operating conditions (different altitude, Mach Number, day type and so on). Here geometry is fixed and operating conditions are changing. While the literature typically describes off-design performance computation as steady-state analysis, this may as well involve transient simulation.

In order to provide complete functionality in relation to the established methods, these two kinds of computations were also implemented in the Jet Propulsion Library. They can be selected as “simulation modes”.

A closer look at the literature (e.g., (Walsh and Fletcher, 2004)) reveals that the notion of on-design computations is not directly compatible with the rules of balanced modeling in Modelica (Olsson et al., 2008). Typically, the bypass ratio or flow split is imposed on a three-way junction or splitter model. Following the rules of balanced modeling, such a component may however only impose both downstream pressures or impose one downstream pressure and the bypass ratio or a flow rate. In on-design computations it is however required for the scaling procedure to impose both downstream pressures *and* the bypass ratio. Off-design computations in turn are basically the computations classically done in the Modelica language. Therefore, the corresponding simulation problems (be it in steady-state or transient mode) are fully compatible with the concept of balanced modeling.

As the constraints of balanced modeling are imposed for very good reasons (for instance, to improve debugging messages and ensure “plug and play” compatibility when selecting specific implementations during system architecting for a given placeholder) it was not an option on the design of the Jet Propulsion Library to rely on locally unbalanced models. Instead, it was decided to restrict the use of on-design computations to initial time and initial equations. Like this, initial equations are used to compute corresponding values of the component parameters that have a fixed attribute equal to false. Based on this decision both on-design computations and off-design computations are in scope of the Library, and its component and system models always remain balanced.

2.5 Component models

With the exception in the connector definition described in section 2.3, the Jet Propulsion Library fully follows the variable naming convention suggested in ARP5571 (SAE, 2005).

2.5.1 Boundary conditions

The types of boundary condition models in the Jet Propulsion Library are similar to those in other thermo-fluid dynamics libraries. Most fundamentally, we distinguish boundary conditions imposing a given mass flow rate, and boundary conditions imposing pressure (obviously all boundary conditions also impose quantities transported by convection). These two kinds of boundary conditions are also required for modeling and simulation of jet engines. However, the prescribed variables may now change from on-design to off-design computations. To provide full flexibility to the user, four different flags are exposed on a boundary condition. These allow to switch on and off the prescription of pressure and mass flow rate for on-design and off-design models respectively. To improve ease-of-use, these four flags are only exposed to the user in the category of advanced component parameters; normally (in simple boundary condition parameterization mode), the user only decides whether the boundary condition is nominally a source or a sink.

- A nominal source prescribes both pressure and flow rate for on-design computations, and pressure for off-design computations, and
- A nominal sink prescribes neither pressure nor flow rate in on-design computation, and pressure in off-design computation.

Figure 2 shows a simple model diagram with such boundary condition instances. Color-coding is used to illustrate whether a component includes over-constrained initial equations (nominal source with green outline) or under-constrained initial equations (nominal sink with blue outline). As long as the number of blue components is equal to the number of green components the system model will be well-posed (actually, any system is well-posed by construction, the color-coding still helps users to double-check their model build-up). The color-coding is also used on other components such as the splitter mentioned in section 2.4 already. Quantities imposed for on-design and off-design have the corresponding letter written in opaque font on the boundary condition, quantities that are either imposed for on-design or imposed for off-design have their corresponding letter written in slightly transparent font.

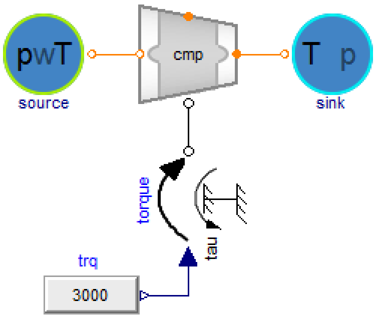


Figure 2. Single component experiment with two boundary conditions (the vectorized bleed port at the top is unconnected and has length zero)

2.5.2 Compressor

The compressor model is one of the component models that contains the scaling factors mentioned in section 2.4. For the compressor on-design performance computation, the user typically prescribes isentropic efficiency η_{des} and pressure ratio π_{des} at the design point. The corrected mass flow rate $w_{c,des}$ is not imposed directly as a parameter on the compressor model but on the system model as a whole, and then computed from boundary conditions or inlet as well as design bypass ratio BPR_{des} (the same holds for the corrected speed N_c).

The overall compressor performance in terms of isentropic efficiency η (or specific work) and pressure ratio π is encoded in performance maps (Walsh and Fletcher, 2004). Based on a particular point in the performance map that is marked as the design point, four scaling parameters are then computed as described by (Jones, 2007).

- Is. efficiency scaling factor $s_{\eta,des} = \frac{\eta_{des}}{\eta_{des,unscaled}}$
- Pressure ratio scaling factor $s_{\pi,des} = \frac{\pi_{des}-1}{\pi_{unscaled}-1}$
- Corrected flow scaling factor $s_{w_c,des} = \frac{w_{c,des}}{w_{c,unscaled}}$
- Corrected speed scaling factor $s_{N_c,des} = \frac{N_{c,des}}{N_{c,unscaled}}$

Here, quantities with index *unscaled* indicate the value in the original, unscaled performance map. Based on this procedure, one compressor map with its design point can be scaled to represent another compressor (as described by the target design point as prescribed by the user). As long as the design points are close enough, the scaling gives a reasonable approximation of the compressor behavior.

As indicated above, the compressor model requires that the off-design performance is captured in the format of a performance map. The format of this performance map has to be easy to handle in an computing environment (Walsh and Fletcher, 2004), and avoid vertical or horizontal lines in the table look-up (Jones, 2007). For this reason we use beta or R-line maps[3]. The method described by (Jones, 2007) in more detail. Basically, the performance maps relates the important thermodynamic variables like corrected mass flow rate, pressure ratio, corrected speed and the efficiency of the compressor. A compressor performance map using R-lines is shown in figure 3. R-lines are family of curves that are parallel to the surge line and evenly spaced among each other. The R-lines ensure unique result in the regions of low corrected air flow where pressure ratio is almost a constant and regions of constant air flow towards the highest air flow region for a given speed line (avoiding table look-up along vertical or horizonal tangents).

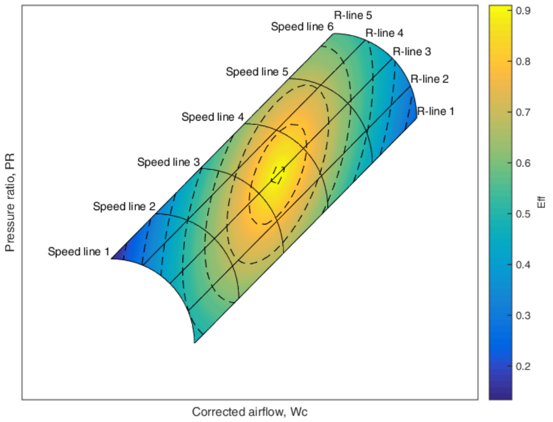


Figure 3. R-line based compressor map with speed lines (solid), r-lines (solid), and efficiency contours (dashed)

Other methods to capture the compressor performance characteristics are described in the literature. One example is the Map Fitting Tool (MFT) method (Sethi et al., 2013). While Jet Propulsion Library currently only implements the R-line or beta line methodology, the object-oriented structure allows for the convenient addition of such additional map format in the future.

Once the key component performance variables were read from the performance map, the component computations continue as known from other thermo-fluid dynamics libraries.

The compressor model also has a mechanical connector through which it can receive shaft power (for instance from the respective turbine models).

The compressor model (like all components in the Jet Propulsion Library) support the modeling of secondary air systems. For instance, bleed can be extracted from this compressor model, routed through an arbitrary network, and be supplied for turbine film cooling or for so-called customer purposes. A bleed mass flow rate through the bleed ports can be specified via constant or variable bleed mass flow fractions in the model. In order to capture the stage at which the bleed air is extracted, parameter corresponding to the relative bleed enthalphy and the pressure as a fraction of inlet and outlet conditions are used.

2.5.3 Turbine

The turbines models are built very similar to the compressor models based on the off-design turbine performance map and a set of scaling factors. For the on-design performance computation, the user typically prescribes isentropic efficiency η_{des} and (uncorrected) shaft speed N at the design point. The pressure ratio π_{des}, the corrected mass flow rate $w_{c,des}$ are again computed from boundary conditions and the system model (the pressure ratios at the design point are for instance solved for such that the power balances per shaft are fulfilled).

The turbine performance map used is as shown in figure 4. Given pressure ratio and speed, corrected flow and isentropic efficiency can be uniquely determined in a turbine map. This eliminates the need for R-lines as discussed in the compressor section. The format is again based on (Walsh and Fletcher, 2004; Jones, 2007). The four scaling parameters then computed from the performance map design point and the jet engine design point are similar to the ones used for the compressor and described in section 2.5.2.

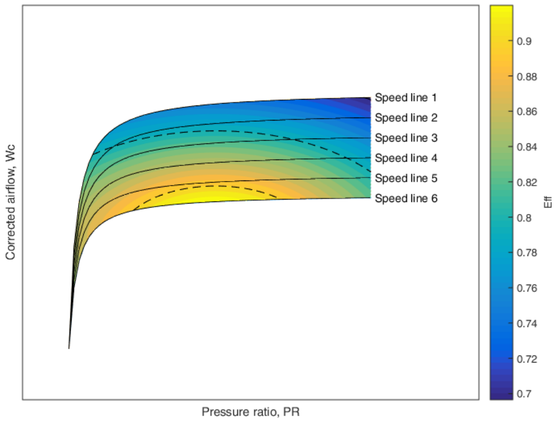


Figure 4. Turbine map with speed lines (solid) and efficiency contours (dashed)

Again, once the key performance variables were read from the map, the component computations basically continue as known from other thermo-fluid dynamics libraries. This means for instance that the turbine model also has a mechanical port through which expansion power is supplied to the compressor through shafts. As indicated before, the turbine model optionally provides bleed ports which can receive secondary cooling air from compressor stages or other sources. The resulting power

[3] The notion of using an auxiliary coordinate has at least two different names; beta lines (Walsh and Fletcher, 2004), and R-lines (Jones, 2007).

from the bleed air flows can be accounted for considering the tip velocity of the turbine blades.

2.5.4 Inlet

A critical part of the inlet models are parametric predictions of the ram pressure recovery η_{ram} and the spillage, bleed, and bypass drag (expressed via the corresponding drag coefficient $C_{d,install}$). These effects can be modeled using the correlations suggested by Kowalski (Kowalski and Atkins, 1979). These are available in two different flavors; a long (and more accurate) form of computation involving 14 tables, and a short form involving 2 compressed tables. The dimensional drag force due to capturing air D_{ram} is eventually computed from

$$D_{ram} = w \cdot v \tag{7}$$

where the ram pressure recovery η_{ram} indirectly influences mass flow rate. The drag force due to installation $D_{install}$ is

$$D_{install} = 1/2 \rho v^2 C_{d,install} \tag{8}$$

The given references contains more details about the implementation.

2.5.5 Nozzle

The ideal gross thrust $F_{g,ideal}$ of a nozzle can readily be computed from the following equation

$$F_{g,ideal} = w \cdot v + \left(p_{s,exit} - p_{s,amb}\right) Ae_{exit} \tag{9}$$

Here, $p_{s,exit}$ and $p_{s,amb}$ are the static pressures at the nozzle exit section and the ambient respectively. Again, the cruicial question for sound model-based design application is how much of the ideal results are achievable. This can be expressed via a number of correlations. One of the quantities to use for this purpose is the nozzle exit gross thrust coefficient C_{F_g}. This coefficient is also used by Kowalski (Kowalski and Atkins, 1979). Beyond C_{F_g} correlations to approximate gross thrust F_g, this methodology also estimates the aftbody drag coefficient $C_{d,ab}$. The former for instance is computed based on pressure ratio and area ratio. Two variations for an axisymmetric nozzle as well as 2-D nozzle exists. Eventually, the actual gross thrust can be computed

$$F_g = C_{F_g} F_{g,ideal} \tag{10}$$

The nozzle model in the library contains replaceable models to compute the contributions individually. This completes the short overview of exemplary component models.

2.6 Interface and template structure

Based on these component models, different cycles can be built up using an interface and template model structure. Different kinds of cycles such turbo fan, turbo jet, geared turbo fan, turbo prop or turbo shaft have been disassembled virtually into reusable sub-system and sub-assembly models. Based on the object-oriented interface and template structure they can be plugged together in a highly flexible and efficient manner. An unmixed turbo fan for instance consists of the inlet section, fan and compressors, the combustor, the turbines, and the primary and secondary nozzles. An exemplary break-down for such a two spool unmixed turbo fan thus is

- Inlet section
  - Inlet
  - Inlet frame duct
  - Inlet engine duct
- Fan and compressor
  - Fan
  - Splitter
  - Low pressure compressor
  - High pressure compressor
  - Fan duct
- Combustor
  - Diffuser duct
  - Burner
- Turbine
  - High pressure turbine
  - Low pressure turbine
- Primary and secondary nozzle sections
  - Exhaust frame duct and exhaust tailpipe duct, or bypass exhaust frame duct
  - Nozzle

Different to the state of the art described in section 1, this approach uses hierarchy and object-orientation to manage variants and system complexity. Previous art lays all element out on a flat level. With this approach, we can conveniently exchange inlet section models from regular inlets to inlets with inlet particle separator, based on available map data one can conveniently switch between average and split fan models (averaging the core and bypass fan flow, or modeling them via separate fan models using different maps), number of spools, as well as detailed section models for compressor and turbine sections (stage representation, inclusion or removal of case strut ducts, inlet guide vane ducts, transition ducts, exit guide vane ducts and so on).

An example breakdown of a turbo fan engine is illustrated graphically in figure 5. This figure shows the actual view presented to the user in the graphical user interface of a Modelica Integrated Development Environment (IDE).

Each type of component in the break-down above is represented through a class hierarchy of interfaces, templates, and implementations. The implementations use the atomic components described in section 2.5.

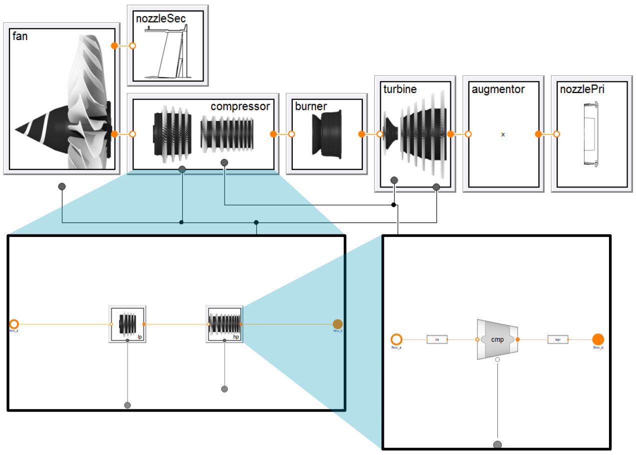


Figure 5. Top-level turbofan model breakdown shown on the top, compressor break-down on the lower left, high pressure compression section break-down on the lower right

3 Application example and results

A complete jet engine model was built using the Jet Propulsion Library of the Pratt & Whitney JT9D. It was created by configuring the two spool unmixed turbo fan template model. Each component starting from inlets, fans, compressors, turbines etc. is redeclared with parameterized models. The respective performance maps are adapted from the open source distribution of the Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS) as described by (Chapman et al., 2014).

Table 1 provides key cycle parameters at the design point. These parameters are approximate but consistent with the given source.

Table 1. Cycle design point parameters.

Parameter	*Value*
Design point	Sea level static
Day conditions	ISA+15 °C
Inlet flow	698 kg s^{-1}
Bypass ratio	5.2751
Turbine inlet temperature	1260 °C
Net thrust	223 kN

Figures 6, 7, and 8 show the compressor performance maps. Following the principles described in section 2.5.2, these maps are scaled based on the component design point data given in tables 2, 3, and 4.

Table 2. Fan design point parameters.

Parameter	*Value*
Efficiency	90.38 %
Pressure ratio	1.603 06

Table 3. Low pressure compressor design point parameters.

Parameter	*Value*
Efficiency	86.575 %
Pressure ratio	2.25

Table 4. High pressure compressor design point parameters.

Parameter	*Value*
Efficiency	86.2469 %
Pressure ratio	5.679 05

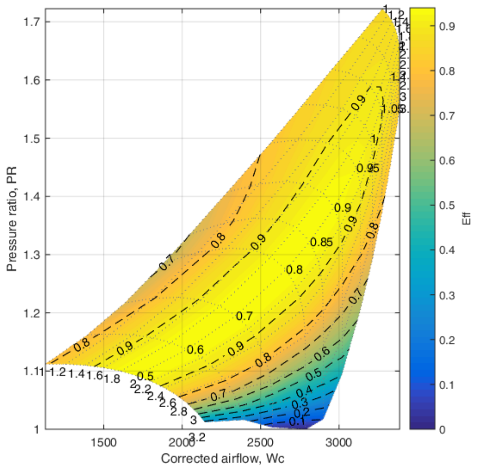


Figure 6. JT9D fan compressor map

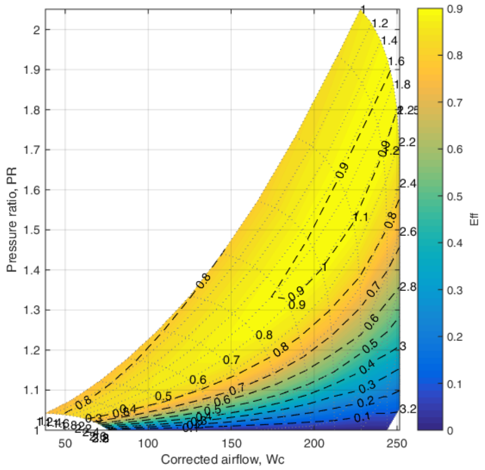


Figure 7. JT9D low pressure compressor map

Figures 9 and 10 in turn show the turbine performance maps. These maps are scaled based on the component design point data given in tables 5 and 6.

Table 5. High pressure turbine design point parameters.

Parameter	*Value*
Efficiency	91.445 %
Shaft speed	8000 / min

Table 6. Low pressure turbine design point parameters.

Parameter	*Value*
Efficiency	92.88 %
Shaft speed	3750 / min

In the following, exemplary simulation results are provided. For this purpose, two boundary conditions are imposed on this model, the aircraft Mach number and the fuel flow rate. The design point simulation runs for a sea-level static case. Basic sanity check results show a reasonably good match of the sea level static thrust and specific fuel consumption produced by the engine model with published data (Saarlas, 2007). Then, the boundary condition parameters are varied for off-design simulation. The model was simulated to conduct a full factorial experiment for inputs of inlet Mach numbers (0.5, 0.7 and 0.9) and fuel flow that varied ±50% from the nominal value in steps of 5%. The results of this full factorial experiment is summarized in the two figures below.

Figure 11 plots the relationship between the low pressure spool speed N_L and thrust for different Mach numbers. The thrust is divided by $\delta = p_t/p_{t,ref}$, the normalized inlet total pressure. The low pressure spool speed is divided by the square root of $\theta = T_t/T_{t,ref}$, the normalized inlet total temperature. These corrections are routinely done to normalize the data (Walsh and Fletcher, 2004). Higher Mach Numbers show lower corrected thrust due to the inlet ram drag.

The corrected thrust specific fuel consumption trends are shown in figure 12. Both plots are qualitatively very similar to the charts given in the relevant literature such as (Walsh and Fletcher, 2004; Saarlas, 2007). Illustrative results of the transient simulation mode will be given in a separate reference.

4 Conclusions

The Jet Propulsion Library provides a foundation for modeling and simulation of jet engines, and the model-based design of integrated aircraft system designs. It contains fully models for sizing and performance computations, and has a number of advantages over existing domain-specific solutions due to the use of the Modelica language.

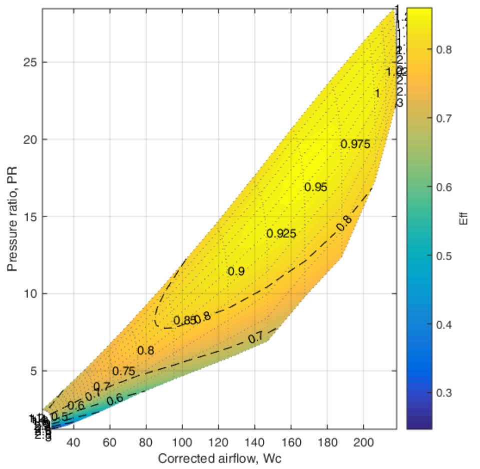


Figure 8. JT9D high pressure compressor map

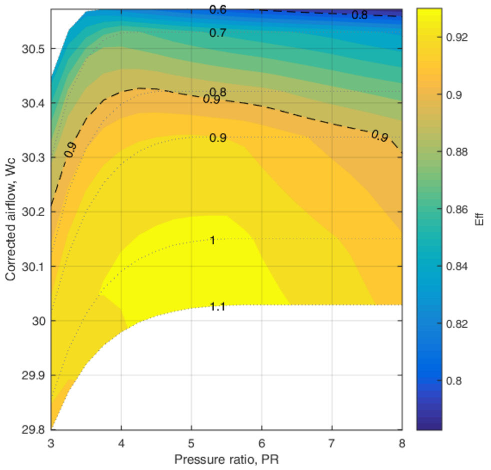


Figure 9. JT9D high pressure turbine map

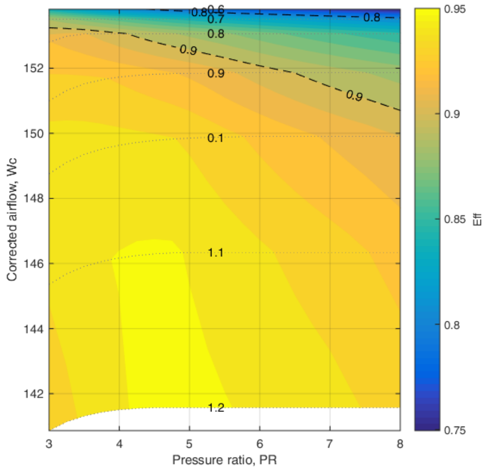


Figure 10. JT9D low pressure turbine map

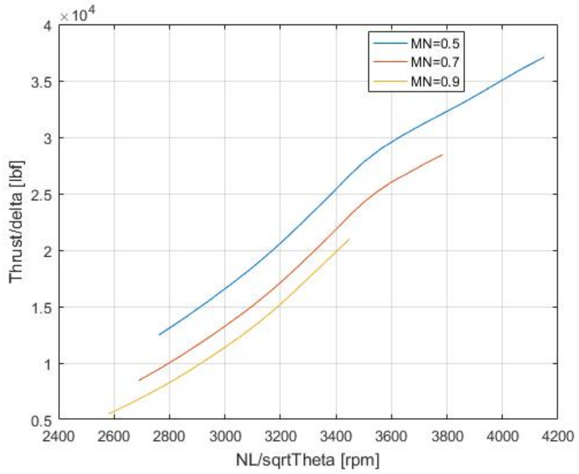


Figure 11. JT9D corrected thrust vs. corrected low pressure spool speed

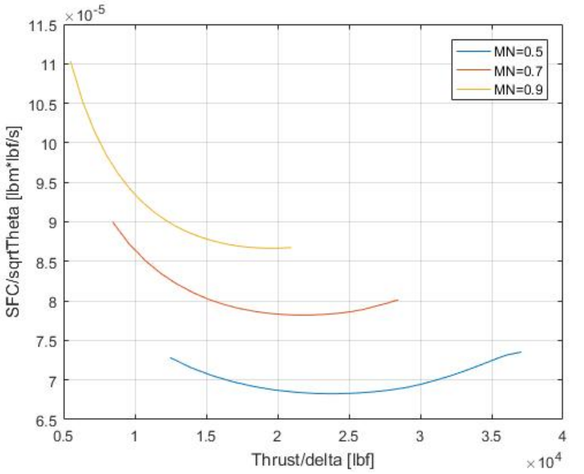


Figure 12. JT9D corrected thrust specific fuel consumption vs. corrected thrust

5 Acknowledgements

We acknowledge the contributions of Shashank Swaminathan who contributed to the interface and template structure described in section 2.6 as summer intern at Modelon, Inc in 2016.

References

A Alexiou and K Mathioudakis. Development of gas turbine performance models using a generic simulation tool. In *ASME Turbo Expo 2005: Power for Land, Sea, and Air*, pages 185–194. American Society of Mechanical Engineers, 2005.

Arjun Bala, Vishal Sethi, E Lo Gatto, Vassilios Pachidis, and Pericles Pilidis. ProosisŮa collaborative venture for gas turbine performance simulation using an object oriented programming schema. In *International Symposium on Air Breathing Engines*, 2007.

F. Casella, M. Otter, K. Proelss, C. Richter, and H. Tummescheit. The modelica fluid and media library for modeling of incompressible and compressible thermo-fluid pipe networks. In *Proceedings of the Fifth International Modelica Conference*, pages 631–640, 2006.

Jeffryes W Chapman, Thomas M Lavelle, Ryan May, Jonathan S Litt, and Ten-Huei Guo. Propulsion system simulation using the toolbox for the modeling and analysis of thermodynamic systems (t mats). In *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, 2014.

Hilding Elmqvist, Hubertus Tummescheit, and Martin Otter. Object-oriented modeling of thermo-fluid systems. In Peter Fritzson, editor, *Proceedings of the Third International Modelica Conference*, pages 269–286, Linköping, Sweden, 2003.

Rüdiger Franke, Francesco Casella, Martin Otter, Michael Sielemann, Sven-Erik Mattson, Hans Olsson, and Hilding Elmqvist. Stream connectors—an extension of Modelica for device-oriented modeling of convective transport phenomena. In Francesco Casella, editor, *Proceedings of the seventh International Modelica conference*, pages 108–121, Como, September 2009a.

Rüdiger Franke, Francesco Casella, Michael Sielemann, Katrin Proelss, Martin Otter, and Michael Wetter. Standardization of thermo-fluid modeling in Modelica.Fluid. In Francesco Casella, editor, *Proceedings of the seventh International Modelica conference*, pages 122–131, Como, September 2009b.

Scott M Jones. An introduction to thermodynamic performance analysis of aircraft gas turbine engine cycles using the numerical propulsion system simulation code. Technical Report TM—2007-214690, NASA, March 2007.

Scott M Jones. Steady-state modeling of gas turbine engines using the numerical propulsion system simulation code. In *ASME Turbo Expo 2010: Power for Land, Sea, and Air*, pages 89–116. American Society of Mechanical Engineers, June 2010.

Edward J. Kowalski and Robert A. Atkins, Jr. A computer code for estimating installed performance of aircraft gas turbine engines. Technical report, National Aeronautics and Space Administration, 1979.

Joachim Kurzke. Advanced user-friendly gas turbine performance calculations on a personal computer. In *ASME 1995 International Gas Turbine and Aeroengine Congress and Exposition*. American Society of Mechanical Engineers, June 1995.

Joachim Kurzke. About simplifications in gas turbine performance calculations. In *Proceedings of the ASME Turbo Expo*, volume 3, pages 14–17, May 2007.

Konstantinos G Kyprianidis, Ramon F Colmenares Quintero, Daniele S Pascovici, Stephen OT Ogaji, Pericles Pilidis, and Anestis I Kalfas. Eva: A tool for environmental assessment of novel propulsion cycles. In *ASME Turbo Expo 2008: Power for Land, Sea, and Air*, pages 547–556. American Society of Mechanical Engineers, 2008.

Konstantinos G Kyprianidis, Andrew M Rolt, and Tomas Grönstedt. Multidisciplinary analysis of a geared fan intercooled core aero-engine. *Journal of Engineering for Gas Turbines and Power*, 136(1):011203, 2014.

Linda Larsson, Tomas Grönstedt, and Konstantinos G Kyprianidis. Conceptual design and mission analysis for a geared turbofan and an open rotor configuration. In *ASME 2011 Turbo Expo: Turbine Technical Conference and Exposition*, pages 359–370. American Society of Mechanical Engineers, 2011.

John K Lytle. The numerical propulsion system simulation: A multidisciplinary design system for aerospace vehicles. In *International Symposium on Air Breathing Engines*, September 1999.

Lester Nichols and Christos Chamis. Numerical propulsion system simulation-an interdisciplinary approach. In *Conference on Advanced Space Exploration Initiative Technologies*, September 1991.

Hans Olsson, Martin Otter, Sven Erik Mattsson, and Hilding Elmqvist. Balanced models in modelica 3.0 for increased model quality. In *Proceedings of the 6th International Modelica Conference*, 2008.

Michael John Provost. The more electric aero-engine: a general overview from an engine manufacturer. In *Power Electronics, Machines and Drives, 2002. International Conference on*, number 487, pages 246–251. IEEE, 2002.

Maido Saarlas. *Aircraft performance*. John Wiley & Sons, 2007.

S-15 Gas Turbine Perf Simulation Nomenclature and Interfaces Committee of SAE. Application programming interface requirements for the presentation of gas turbine engine performance on digital computers (ARP4868). Technical report, Society of Automotive Engineers, 2001.

S-15 Gas Turbine Perf Simulation Nomenclature and Interfaces Committee of SAE. Gas turbine engine performance presentation and nomenclature for digital computers using object-oriented programming (ARP5571). Technical report, Society of Automotive Engineers, 2005.

Vishal Sethi. *Advanced performance simulation of gas turbine components and fluid thermodynamic properties*. PhD thesis, Cranfield University, April 2008.

Vishal Sethi, Georgios Doulgeris, Pericles Pilidis, Alex Nind, Marc Doussinault, Pedro Cobas, and Almudena Rueda. The map fitting tool methodology: Gas turbine compressor off-design performance modeling. In *Journal of Turbomachinery*. American Society of Mechanical Engineers, September 2013.

Philip P Walsh and Paul Fletcher. *Gas turbine performance*. John Wiley & Sons, 2004.

Michael Winter. A view into the next generation of commercial aviation (2025 timeframe). In *AIAA Aerospace Today and Tomorrow*, 2013.

DOI
10.3384/ecp17132909