DEVELOPMENT OF A GAS TURBINE PERFORMANCE CODE AND ITS APPLICATION TO PRELIMINARY ENGINE DESIGN

R. Becker, F. Wolters, M. Nauroz, T. Otten
German Aerospace Center - Institute of Propulsion Technology
Linder Höhe 51147 Köln, Germany

Abstract

The present paper describes the design and application of the new DLR gas turbine simulation environment GTlab which is an object-oriented performance synthesis program written in C++ and Python. Its modular layout is designed to be extensible and flexible in use and can seamlessly be integrated into other simulation and optimization environments. The new performance code provides means for steady state and transient performance simulation of arbitrary engine concepts and is used in DLRs design and assessment of future aero engines.

The purpose of this paper is twofold: The first part describes the object-oriented framework and software design in general as well as more detailed aspects of the modelling libraries representing the physical behaviour of engine components. In part two GTlab is applied to a preliminary design process. Subject of the exemplary investigation is a two shaft military aero engine designed to operate in a high subsonic unmanned combat aerial vehicle (UCAV). First the GTlab model setup for such an engine will be explained. Following is the presentation of a mission point based cycle optimization. Finally the outcomes of a single preliminary thermodynamic design are presented.

Nomenclature

BPR	Bypass Ratio
CO	Carbon monoxide
CRISP	Counter Rotating Integrated Shrouded Propfan
CROR	Counter Rotating Open Rotor
DLR	German Aerospace Center
FN	Net Thrust
FPR	Fan Pressure Ratio
HC	Hydrocarbon
HPC	High Pressure Compressor
HPT	High Pressure Turbine
ICR	Intercooled Recuperated Engine
IR	Infrared
LPC	Low pressure compressor
LPT	Low pressure turbine
LTO	Landing and take-off cycle
Ma	Mach number
MTF	Mixed Turbofan Engine
MTOW	Maximum take-off weight
NO_x	Nitrogen oxide
OEW	Operating empty weight
OOD	Object Oriented Design
OOP	Object Oriented Programming
OPR	Overall pressure ratio
RCS	Radar cross section
RF	Radio frequency
SEAD	Suppression of enemy air defence
SFC	Specific fuel consumption
SOFC	Solid oxide fuel cell
TOC	Top of climb
TO	Take-Off
T4	Turbine entry total temperature
T3	Compressor exit total temperature
T8	Nozzle outlet total temperature
UAV	Unmanned aerial vehicle
UCAV	Unmanned combat aerial vehicle
W2	Fan inlet flow

1 Introduction

To assess the feasibility and benefits of future aero engine concepts, preliminary design studies have to be performed. The preliminary design task has to consider multidisciplinary requirements and constraints including thermodynamics, aerodynamics and mechanics as well as weight, emissions and costs. To address this problem the German Aerospace Center (DLR) started the project EVITA (Evaluierung innovativer Turboantriebe) [1]. Within this project a variety of new software tools have been designed and applied. Besides aerodynamic and structural preliminary design tools for turbo components and combustors a new thermodynamic gas turbine performance program has been developed.

Nowadays performance synthesis software can be found in a wide range of applications. Besides classical thermodynamic cycle analysis, performance models accompany the engine life cycle from preliminary design studies over engine testing, certification, diagnostics, in service support and maintenance.

Within DLRs gas turbine performance group synthesis software is utilized in almost all fields of research. Historically a primary working area is the modelling and analysis of aero engine emissions to evaluate the potential effects of aviation emissions on the environment and the global climate. Therefore the operating parameters of individual combinations of engine and aircraft types are determined by aero engine synthesis and aircraft performance programs and analysed on flight mission basis. Another focus is set on the assessment and development of engine concepts and preliminary gas turbine design. Both aero engine related cycles as well as stationary gas turbines for power generation are evaluated. Last but not least the impact of alternative fuels on aero engine performance and emissions is investigated.

All of the above research topics pose requirements on a gas turbine performance simulation tool. The primary

demands are high flexibility and extensibility in engine concept modelling. The software tool has to provide means to simulate current engine technology concepts like turbojets, turbofans and turboshafts as well as future concepts such as counter rotating (open) rotors (CRISP/CROR) and intercooled recuperated engines (ICR). It needs to be extensible for the investigation of power generation related topics like combined gas and steam cycles, solar-hybrid gas turbines and solid oxide fuel cells (SOFC) hybrid cycle technology.
Besides the technological demands mentioned above software tools deployed in research environments face additional requirements. Most certainly the program is applied to variety of extrinsic tasks not directly linked to the developing performance group. The tasks range from aircraft preliminary design and flight mission analysis over the generation of boundary conditions and target functions for component design to future non foreseen applications. In many cases it is highly desirable to have the freedom to:

1. Directly integrate the performance tool into external software.
2. Introduce local modifications to any characteristic of tool behaviour.
3. Invoke an arbitrary number of program instances running at the same time.
4. Share the program with other facilities and partners.

Software licensing must allow for the points mentioned above at moderate costs while ensuring access, sustainability and control.
As for today a variety of commercial available synthesis software exists. To mention some of the most popular products that were evaluated in the forefront of the GTlab development efforts:

NPSS: The Numerical Propulsion System Simulation was originally designed at NASA GRC in cooperation with a consortium of other government and industry partners. It provides a modular object-oriented framework to simulate an extensive range of gas turbine and propulsion concepts [2]. NPSS is available on Microsoft Windows and Linux platforms and allows for the development of custom component extensions. Although initially restricted to the consortium members it is now available to the public under a commercial software license.

PROOSIS: The Propulsion Object Oriented Simulation Software is an object-oriented gas turbine performance tool based on the EcosimPro [3] framework initially developed at the European Space Agency (ESA). PROOSIS runs on MS Windows and comes with a graphical user interface (GUI). It offers a modular simulation build up to facilitate arbitrary engine concepts. The development of custom component extensions is supported. An overview of the software is given in [4].

GSP: The Gas turbine Simulation Program was originally developed at the Delft Technical University and was continued at the Netherlands National Aerospace Laboratory NLR. GSP provides a graphical user interface to build the simulation configuration from a library of pre-defined modules. It is commercially available for MS Windows only. Further information about the capabilities of the program is presented in [5].

GasTurb: The program offers performance simulation for a given set of predefined engine configurations. The GUI is optimized to allow for a convenient execution of the most frequent performance related tasks [6]. Custom component and engine configuration development is not supported. GasTurb is offered for MS Windows only.

Although highly professional none of the third party software mentioned above fulfilled all of the requirements, which lead to the decision to develop a new in house performance synthesis code.

2 GTlab Overview

DLRs new gas turbine performance code GTlab (Gas Turbine laboratory) is a component based simulation environment that is capable of simulating arbitrary gas turbine and turbo engine concepts. Both design and off-design simulations as well as stationary and basic transient simulation can be performed. An open interface software concept supports the integration of GTlab in other optimization and simulation software such as multi purpose multi objective optimization tools and flight mission analysis software. GTlab allows for the generation of customer decks which fit into the design and simulation process of the airframe research group of the German Aerospace Center. In the following sub sections the GTlab software concept, modelling methods, numerical scheme and model setup are described.

2.1 GTlab Software Concept

As most modern software projects today the architecture of GTlab is based on object-orientated design (OOD) methods. Object orientated programming (OOP) provides a set of mechanisms and principles that strongly support the nature of gas turbine synthesis modelling namely abstraction, encapsulation, polymorphism, inheritance and aggregation [7]. An excellent description of the application of OOD methods to gas turbine simulation can be found in [8] and [9].

GTlab utilizes third party software libraries and tools to keep the development efforts and resources at minimum. To cultivate sustainability and to ensure future access to external dependencies the GTlab software concept follows a free software and open standards approach. All of the included third party software complies with free software licensing requirements [10]. Interfaces and data exchange are designed to be simple and compatible to international industry standards such as [11], [12] and [13].

The GTlab software layout is depicted in figure 1. The core library libGTlab provides all classes and functions necessary to perform performance simulations. This library is subdivided into several sub sections:

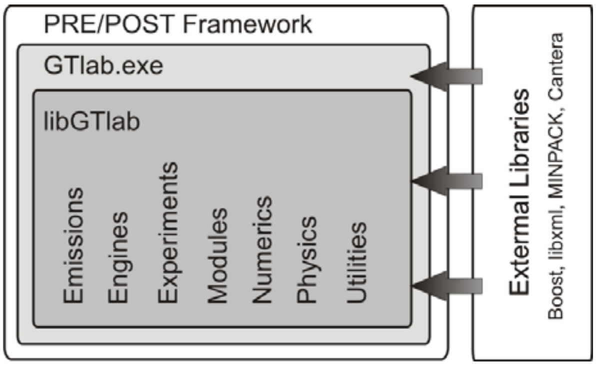

Figure 1: GTlab software layout

Engines: This section comprises all classes and methods necessary to build up, access and control the engine model. Functions range from model description parsing, module and connector data structure build-up, on- and off-design run control as well as public interfaces for external access to the engine data.

Modules: Within GTlab engine models are composed of single modules representing the physical engine components such as intake, compressor, shafts, etc. all component models are collected in the modules section of the software library.

Experiments: GTlab simulations are controlled by a set of commands which define the engine operating conditions during virtual test runs as well as the settings and parameters of the external non-linear solver. All related classes and methods are stored in the corresponding library section.

Numerics: This section contains tools for the external non-linear solver module. The solver is interfaced to GTlab by an abstract solver class in order to provide means for the integration of diverse solver libraries.

Physics: Set of functions and classes describing general physical models that are applicable to a broad number of modules. The sub library includes models for thermodynamics, gas dynamics, gas properties, combustion processes and the standard atmosphere.

Utilities: A collection of tools related to software techniques and general processing such as high level XML parsing, lexical and semantic analysis as well as output processing.

Emissions: For engine emission calculation a variety of prediction methods were implemented in the GTlab software framework. The library provides primarily semi-empirical correlation methods, based on a variable reference function, so that NO_x, CO, HC and Soot emissions can be derived.

Utilizing the library, the single executable file GTlab.exe performs all data input, computation and data output of a simulation run. So far GTlab is compiled and tested for Microsoft Windows™ and Linux. However, a deployment to further operating systems is viable.

Input and output data can be pre- and post-processed by a set of programs which are arranged in the PRE and POST module libraries. These scripts which are mainly based on python programming language provide means to perform performance map formatting, XML templating and output conversion as well as all graphical output available to the user.

2.2 Modelling methods

At the beginning of an engine modelling process the user has to decide what kind of engine concept is to be modelled. Possible concepts may range from civil turbofans over propeller engines to future aero engine concepts. Once the concept is decided engine modelling follows a modular build up process. From an abstract point of view a gas turbine engine is assembled from physical engine components such as intake, compressor, combustor, turbine and nozzle as well as its physical surroundings or ambient conditions. Figure 2 illustrates the process.

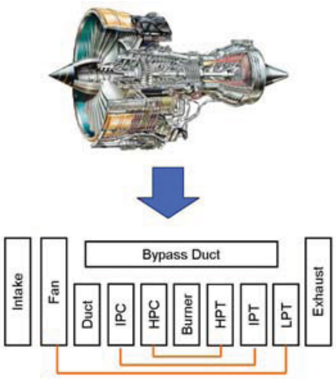

Figure 2: Modular build up process

The components are interconnected and allow for the exchange of mass, energy and momentum in form of fluid transport as well as mechanical shaft power and heat transfer. In this one-dimensional modelling approach the air and gas properties are thermodynamically averaged over the components inflow and outflow cross sections. A numerical synthesis scheme controls the validity of the collective physical model in terms of mass, momentum and energy conservation.

The simulation software GTlab utilizes OOD methods to support the software representation of the modular build up process described above. In OOP the previously described general behaviour of components is expressed by an abstract base class. All components are specializations of this base class and obtain its behaviour by inheritance. Figure 3 shows the fundamental inheritance tree employed by GTlab. Base class for all engine modules utilized in the synthesis process is the abstract class *Module*. It provides all interfaces necessary to access the module parameters via a common symbol table and therefore enables the user to work with named engine and component parameters during simulation setup and analysis. Two direct specializations are derived from the base class, *FluidModule* and *MechModule*, which separate gas loaded modules from their mechanical counterparts. All modules directly run in the synthesis loop by means of their design and off-design computation methods are abstracted by the *Component* base class. Non rotating components such as combustors, ducts, intakes, etc. directly inherit the interfaces of *Component*. Components connected to a shaft or gearbox are abstracted by the base class *TurboComponent* which is a specialization of the *Component* class.

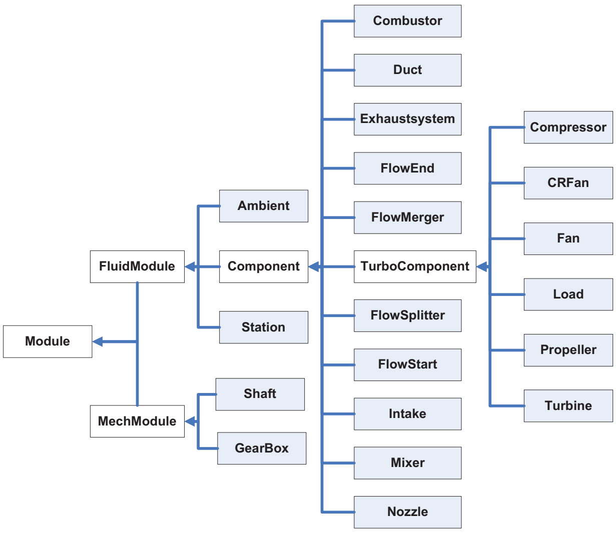

Figure 3: GTlab module inheritance tree

2.2.1 Component modelling

Generally a component retrieves a set of input information such as flows of a working fluid or energy transfer in form of mechanical shaft power or heat transfer. The interfaces for the information interchange between components are called *Ports*. To respect the distinct character of information different types of Ports exist. A *FluidPort* for example transports information about the thermodynamic state and physical properties of the represented fluid flow. In addition to that *BleedPorts* and *ShaftPorts* exist to interface the secondary air system and the mechanical components respectivley. After the information at the input ports is available to the component a functional relation is used to map the input information to a set of output variables. The relation may consist of algebraic equations directly representing a physical model or empirical correlations which are typically represented by interpolation tables or performance maps.
Figure 4 depicts an exemplary structure of an engine fan component.

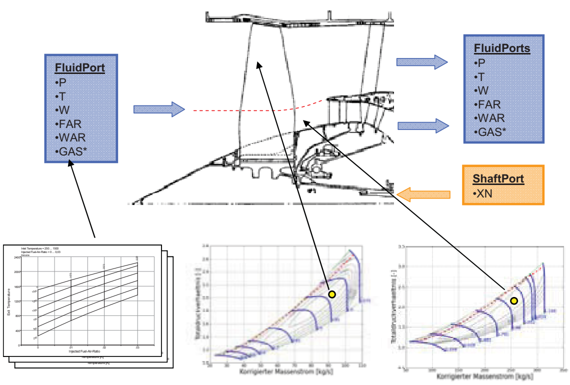

Figure 4: Exemplary computation process of a fan component

For the case of a standard fan module the input fluid flow is represented by single fluid port containing values for stagnation pressure and temperature, natural mass flow, fuel and water to air ratio. A pointer to a gas object common to all modules of the engine enables access to the gas properties and functions. Changes in thermodynamic state across components are calculated using the entropy function ψ , as defined in [14].

(1) $$\psi(T) = \frac{1}{R} \cdot \int_0^T \frac{c_p(T)}{T} dT$$

This has the advantage to do not rely on a simplification like mean or constant heat capacity or isentropic exponents, respectively.

A second form of information input is the shaft port which provides the mechanical speed of the low pressure shaft and transmits the power consumed by the fan after the computation is finished. Two separate compressor maps are employed to define the components off-design behaviour for the bypass flow as well as for the core flow. With the aerodynamic speed a first parameter to define the current operating points in both maps is fixed. With knowledge of two additional external variables beta the operating points are fixed and the map can be used to read stagnation pressure ratio as well as isentropic efficiency. With all input parameters for the two compression processes determined the output fluid port information is computed and transferred to the subsequent components.

2.2.2 Fuel and gas properties

The working fluid in GTlab is a perfect gas, whereas the heat capacity is a function of temperature only:

(2) $$c_p = f(T) \neq f(p)$$

For flexible use of arbitrary fuels within GTlab applications, gas properties and heat addition tables are used, combined in an exchangeable gas performance map. This map represents the data of thermodynamic functions like heat capacity, enthalpy, entropy function, isentropic exponent etc., as well as temperature rise (more precisely combustor outlet temperature) as a function of fuel to air ratio, water to air ratio, temperature and inlet temperature, respectively.

The data is currently generated using the free chemical kinetic software package Cantera [14]. Within Cantera gas composition is calculated assuming chemical equilibrium and heat release is defined by the equilibrium temperature, while considering dissociation effects and enthalpy of vaporization.
The thermodynamic data for heat capacity, enthalpy and entropy function is then derived by means of NASA 7 term polynomials as described in [16]. However, any other chemical calculation tool can be used just as well (e.g. CEA, CHEMKIN), since value generation is a stand-alone process and does not impact the use of the gas performance map.

2.3 Numerical scheme

To simulate the behaviour of the entire engine system GTlab needs to solve a series of non-linear equations. With respect to the synthesis of all component modules a numerical scheme is applied in order to keep the conservation of mass, momentum and energy as well as additional constraints satisfied.

The program goes along with the commonly accepted practice that an external solver rather than nested loops are employed to solve the equation system. The current software version utilizes the external solver library Minpack [17] although an abstract solver interface class allows for the integration of any other appropriate numerical library or function.

From a general point of view the equation system consists of a set of functional relations comprising vectors of independent and dependent variables of equal size. The independents are altered during an iterative process in such a way that all dependent variables are satisfied. In GTlab every module comes along with an arbitrary number of control methods which are directly accessible by the solver object by means of a symbol table. The methods allow for the alternation and enquiry of the modules internal state and behavioural parameters. During a solver setup process the named methods can be employed to form both independent and dependent objects. Due to the high flexibility and potential build-up of arbitrary engine concepts the equation system can not be determined by the program itself. However, for user convenience some of the engine components implicitly involve independent and dependent variables. A compressor component for example needs information about the location of its operating point in the corresponding compressor characteristics in order to model off-design behaviour. Typically the operating point location is determined by the compressor corrected speed and an auxillary coordinate commonly denoted as beta. In GTlab these parameters form default independent objects that can directly used within the solver without additional definition. A similar procedure applies for default dependent objects for example the mass flow error introduced during off-design computation of the compressor module.

A special set of dependent objects are maximum and minimum limiter variables that may additionally be employed. A limiter object allows for the specification of a set of confining parameters such as for example maximum or minimum speeds, temperatures and pressures that have to be maintained during engine operation. Within solver execution the limiter object behaves like a single dependent object. The error introduced by this object is calculated from the limit that is closest to be violated for the mementary iteration step.

2.4 Model setup and execution

A GTlab simulation run is performed by a single executable file named GTlab.exe and controlled by two input XML files: The so called model and experiment file. The former contains all data needed to describe the thermodynamic engine design whereas the latter provides the virtual testbed for a defined engine model.

2.4.1 GTlab model

Within the GTlab model file the engine layout is described and the engine design is defined by specifying the required components and design point parameters, their interconnection and the design point ambient conditions, see Figure 5, which shows the primary structure of the XML model file tree.

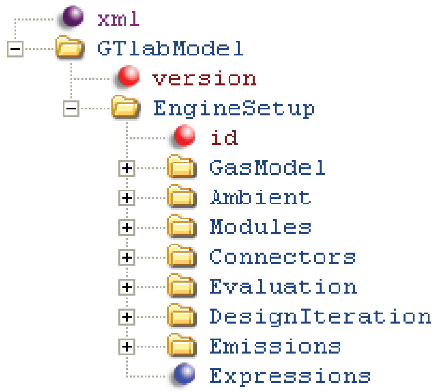

Figure 5: GTlab model file XML structure

The *EngineSetup* node contains several child nodes each providing information to a specific group of engine properties.

The engines general environment is described in the *GasModel* and *Ambient* nodes which contain information about the working fluid and fuel properties as well as the boundary conditions or flight condition respectively. All components and integral parts of the engine are collected as children of the *Modules* node. The modules itself contain information to determine the according design and off-design behaviour, e.g. design stagnation pressure ratio, isentropic efficiency and the performance map in case of a simplified compressor module. If all modules are specified, their interconnection must be determined, which is accomplished by the *Connector* nodes. Connectors of different types such as *FluidConnector*, *BleedConnector* or *ShaftConnector* exist according to the type of ports that are going to be connected. To complete a basic setup one has to specify the order of module evaluation for the problem setup. The *Evaluation* node comprises an ordered list of all modules to be executed during simulation run.

In addition to the mandatory nodes mentioned above several optional information nodes can be specified. It is possible to employ the external solver to iterate additionally desired parameters during design computation. Related information is stored in the *DesignIteration* section of the model file. The optional *Emissions* node contains data needed by the emission correlation methods mentioned in previous chapters. In certain cases it is helpfull to form composed expressions from the engine parameters.

2.4.2 GTlab experiment

To conduct design and off-design calculations with a predefined engine model the GTlab experiment environment is applied. The experiment XML file represents a virtual testbed configuration, in which an arbitrary amount of operating points can be simulated.

To create an experiment the model file of the engine which is going to be tested has to be identified. Moreover

the mode of operation – design or off-design - has to be declared, which applies to all operating points within the experiment. Finally a non empty list of operating points is required.
Operating Points may be defined solely and consist of an individual iteration scheme by means of independent and dependent variables. Additionally boundary conditions in terms of ambient conditions or parameter settings, which are not affected by the control system, and Limiters can be specified optionally. All parameters not described within the operating point settings are taken from the previous calculation, i.e. previous operating point if existent or the design point calculation.
An example of the experiment XML file and its principle layout is shown in Figure 6.

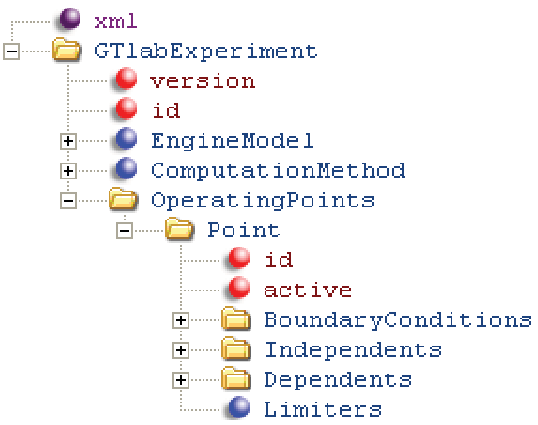

Figure 6: GTlab experiment file XML structure

For convenience operating point definition allows for the definition of loops where an arbitrary set of engine parameters can be automatically altered according to predefined value intervals in order to perform parameter studies, test the working line behaviour or compute engine flight envelopes. Thereby one iteration scheme applies to all operating points within the loop.

3 Application to preliminary design

As initially intended GTlab was successfully applied to the EVITA preliminary design process. In he following the general task, the solution methodology as well as the results of the application example are presented.

In preliminary design the principle engine layout is evaluated. Pre-design represents the initial step during an engine development process. At an early stage of the project preliminary design studies provide potential engine layouts to decision makers for a predefined engine specification. Therefore the scope of preliminary design studies in general is to specify engine type and cycle within a reasonable time rather than designing a single engine on a high level of detail. However, to provide sufficient, significant and dependable data a multidisciplinary and highly iteratively approach is mandatory to consider all aspects of the product, from the estimation of thermodynamic, aerodynamic and structural behaviour to the analysis of engine weight, costs and environmental impact (see figure 7).

In an initial step the engine specification has to be defined which can be derived from airframe configuration and aircraft mission requirements. Cycle optimisation parameters and constraints are identified considering aspects like production, maintenance and operation costs, engine size and weight, complexity and durability, technological innovation and performance requirements.
Once the engine specification is fixed the process chain starts with the design of the thermodynamic engine cycle. Within the thermodynamic design process both design and off-design calculations are performed so that the engine cycle meets all requirements of the specification at particular flight conditions within the flight envelope.

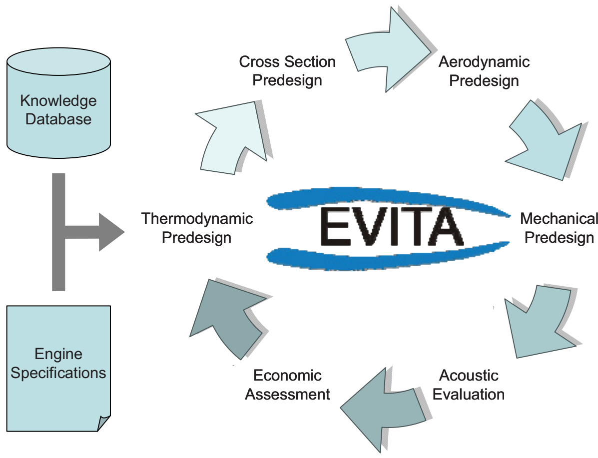

Figure 7: EVITA preliminary design process

Therefore the components pressure ratios and efficiencies at design conditions as well as their off-design behaviour have to be anticipated based on engineering judgment and estimated component characteristics. To keep time consuming iteration to a minimum an extensive knowledge base including data of component behaviour from previous engines or similar designs is of high importance.
The thermodynamic cycle is then optimised to improve a certain figure of merit, e.g. SFC, while not violating any boundary conditions for example maximum burner outlet temperature. Eventually, the thermodynamic cycle is designed providing mean line in- and outlet conditions in terms of mass flow, pressure and temperature, as well as overall component parameters, like pressure ratio or specific work, for the aerodynamic component design. Further on Mach numbers at component in and outlet stations are assumed to generate a design space by means of geometrical boundary conditions like hub, mean and tip radii. Consequently the initial engine cross section can be derived.

In the following the aerodynamic pre-design specifies component size, rotational speeds, amount of stages and blades, blades aerofoil design as well as component efficiencies. Iterations are necessary to adjust previously made assumptions regarding thermodynamic component behaviour to the performance calculated by the aerodynamic simulations. This procedure is repeated considering simplified mechanical stress and dynamics calculations and a rough design of disks and shafts. Finally, if the global iteration is converged, acoustic

evaluations and the economic assessment is performed. Where applicable this process has to be rerun to get reasonable data fulfilling all engine specifications requirements or to evaluate alternative designs.
Since the thermodynamic cycle calculations define all input data and boundary conditions for the aerodynamic design and all subsequent disciplines particular carefulness and proper engineering judgement as well as a sophisticated performance simulation program is required to do not overburden or underestimate potential technical innovations and predict accurate component behaviour. Moreover the synthesis approach during engine performance computation considers component interactions and has to assure to match overall engine performance and engine specification.

3.1 Design Case

In principal the design case is specified by an identified need issued through the customer requirements and/or market research. Typically the engine customer delivers a requirements document such as a request for proposal (RFP) to the engine manufacturer. The RFP contains information about what kind of aircraft is targeted as well as information about its mission and performance constraints. From both airframe and mission specifications the engine requirements are derived.

3.1.1 Aircraft and Mission Specification

The analysed aircraft configuration is a generic UCAV geometry called DLR F-17 developed within the DLR project UCAV-2010 [18]. The targeted thrust to weight ratio for take-off is 0.35 and the MTOW was determined to be 10000 kg as specified in the RFP of the UCAV-2010 project. The design is intended to reduce the aircrafts radar, infrared, visual and acoustic signatures to a minimal practical level. The DLR-F17 main dimensions are shown in Figure 8.

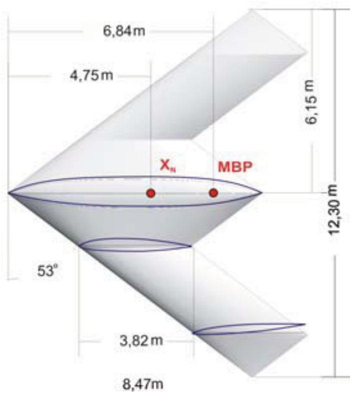

Figure 8: Main dimensions of the DLR-F17 airframe

With respect to UCAV engine design a wide range of possible mission requirements exist. Those are high manoeuvrability and thus high thrust for air to air combat, high endurance for surveillance tasks and high speed at low altitude for quick penetration into hostile terrain.
The mission scenario used for the present thermodynamic engine cycle design is a deep strike mission. It has the purpose of attacking stationary ground targets to suppress enemy air defence (SEAD). In general the mission follows a HI-LO-HI profile, consisting of a cruise phase to efficiently approach hostile terrain at high altitude, a low-altitude, high speed phase in vicinity of the operational area and a final cruise phase as the UCAV returns to its base. Figure 9 depicts the generic mission profile.

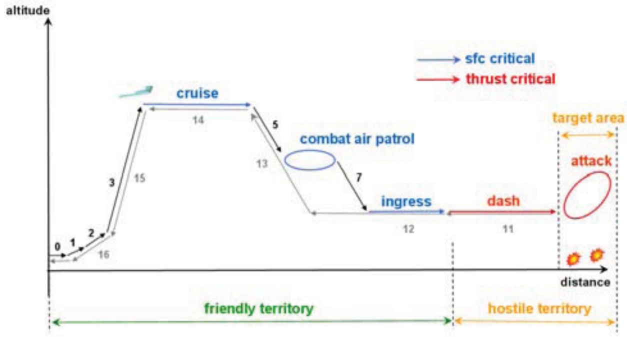

Figure 9: Generic deep strike mission profile

3.1.2 Derived Engine Requirements

With given aircraft performance and mission phases as specified in the RFP the minimum installed engine thrust for each mission point can be estimated by means of a mission analysis. In addition to that desired engine fuel efficiencies in form of SFC targets are derived to maximize the aircrafts mission range or payload which was the main objective defined by the aircraft engine user.
Besides these thrust and SFC constraints of the RFP one of the critical constraints of the engine design for the F-17 UCAV is the limited installation space. The stealth design favours low fuselage cross sections restricting the maximum engine- and thus fan diameter. From a constrained fan diameter a minimum specific thrust can be derived which affects the maximum achievable propulsive efficiency of the concept.
To ensure low radar and IR signatures a double s-shaped inlet as well as an s-shaped nozzle duct was selected. The special design of both ducts reduces the available axial installation space. A reduction of engine length can be achieved by minimizing the number of stages of the turbo components. Therefore high stage loading designs are required for the fan and the compressor. To ensure stable operation over the flight envelope and under all possible transient conditions it is essential that the fan and compressor are each provided with sufficient surge margin. Combining these two requirements poses a challenge for the aerodynamic design. An advanced compressor design method, taking into account these requirements using 3D-CFD multi-objective optimization was presented in [19]. The low pressure compressor characteristics facilitated in the present study are based on the optimized design but employed scaling on pressure ratio, corrected flow and efficiency.
The design of the s-shaped inlet has a major influence on compressor performance and thus on the thermodynamic design. The interaction of compressor performance with distorted inflow conditions from a double s-shaped inlet are analysed in [20]. Based on this study, a duct with a visibility of 30% and a total pressure loss of 4% was selected for the engine model.

3.2 Thermodynamic Cycle Design

Due to the embedded engine design for the UCAV propulsion, a two-spool-mixed-flow turbofan engine (MTF) design with a moderate bypass ratio was chosen. The modular GTlab build up as well as the considered engine components and station numbers are shown in Figure 10.

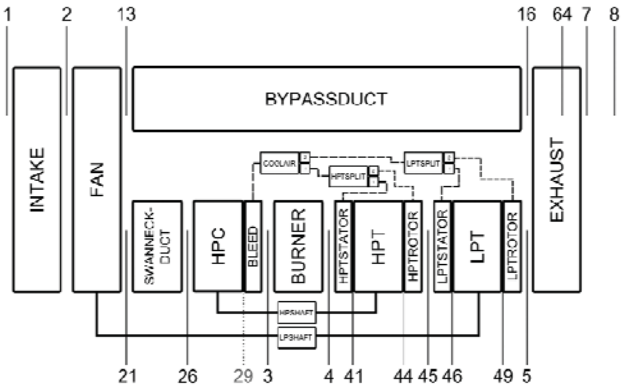

Figure 10: Engine layout and station numbering

3.2.1 Cycle Constraints

The thermodynamic cycle design is highly dependent on numerous practical constraints that have to be considered. These are the limitations of available component technologies and the operational requirements that are dependent on airframe application.

The temperature T4 and the parameter OPR are of fundamental importance for the main core cycle analysis because they limit the thermal efficiency and the energy transfer into the working fluid. A high OPR is favourable in terms of maximising the thermal efficiency of the core. The upper limits generally depend on the compressor and turbine materials as well as the assumed cooling technology and must be constrained for acceptable component life. Additional limits are the increased tip clearances relative to blade height for acceptable aerodynamic losses, which are encountered at the last stages of the HPC and therefore restrict the maximum OPR. With respect to the double s-shaped duct intake installation effects need to be considered in terms of higher required compressor surge margins and hence lowered operating lines and reduced efficiencies.
The constrained fan diameter in connection with a maximum axial inflow Mach number given by the low pressure compressor design restricts the maximal achievable airflow through the engine.
For a sophisticated thermodynamic modelling of the engine various pressure loss coefficients and mechanical efficiencies must be considered. Here their values are based on engineering judgment of the current level of technology.

3.2.2 Technology Assumptions and Initial Cycle

The component performance assumptions are based on state of the art technology. Peak component efficiencies and metallurgical limits are estimated from literature sources [21-24]. A high OPR and turbine entry temperature are targeted to achieve a low SFC.

The stage numbers of the turbo-components are initially derived from stage pressure ratio correlations and the thermodynamic design parameters are estimated by taking into account the specific geometric, stress and aerodynamic limits. Table 1 summarizes the most important assumptions.

Configuration	Two-spool-mixed-flow turbofan
Fan diameter	0.73 m
	Stages
Fan	1
HPC	6
HPT	1
LPT	2
	Isentropic efficiency [-]
Fan bypass	0.84
Fan core	0.89
HPC	0.85
HPT	0.85
LPT	0.89
	Pressure Loss [%]
Intake	4
Burner	6
Bypass duct	2
Nozzle	3
	Component efficiency [-]
Burner	0.99
LP-Spool	0.99
HP-Spool	0.99
Mixer	0.7

Table 1: Engine technology assumptions

3.2.3 Optimisation

An optimization process chain based on the commercial software framework ModelCenter™ [25] was developed and applied to the thermodynamic cycle design. ModelCenter allows users to bring commercial or self-developed software tools into a common environment. Once an integrated model has been built, the design analysis and optimization tools can be used to perform optimization and trade-off studies and to compare different design options. The optimizer used for the present thermodynamic cycle design is the gradient-based optimization DOT [26] code. This DOT optimizer was supplied as part of the ModelCenter software. Figure 11 shows the ModelCenter analysis view with the optimisation configuration and the associated GTlab graphical output.

The UCAV engine cycle has to provide a low installed SFC to improve the mission distance and to reduce the required fuel mass. Contrary to that a high specific thrust is required to maintain the engine fan diameter and thrust limits. The allowed exhaust gas temperatures are restricted in order to reduce the IR-signature. Therefore the thermodynamic engine cycle is optimised around five operating conditions:

1) Of major importance in the mission profile is the cruise segment because the UCAV is designed as an agile bomber to carry a given payload over long distances. Therefore the SFC is a critical parameter for this phase and has to be minimized.

2) For this study the hot day take-off condition (ISA + 25K) is chosen as the engine design point. Hot day conditions are considered in order to meet the take-off thrust requirements for a majority of runway conditions around the globe. This point defines the maximum cycle temperatures and the required cooling mass flows to achieve sufficient metal temperatures in the hot engine sections.

3) The top of climb (TOC) is the condition that determines the maximum OPR, maximum corrected spool speeds and maximum corrected flows for the compressors. Therefore this condition was used to define the maximum engine inlet mass flow for a given fan diameter and a maximum estimated axial Mach number at the fan inlet plane.

4) A thrust-critical operating point in the mission profile is the attack phase. For this manoeuvre conditions high thrust for acceleration is required. Any decrease would make the aircraft vulnerable to enemy attack.

5) The high-speed low altitude dash condition is of particular interest for the survivability of the vehicle. The ability allows to approach the target area within the radar ground clutter and to avoid detection until the airplane nears the target. For stealth purposes, the minimisation of infrared signature and therefore the decrease of the exhaust gas temperature are essential.

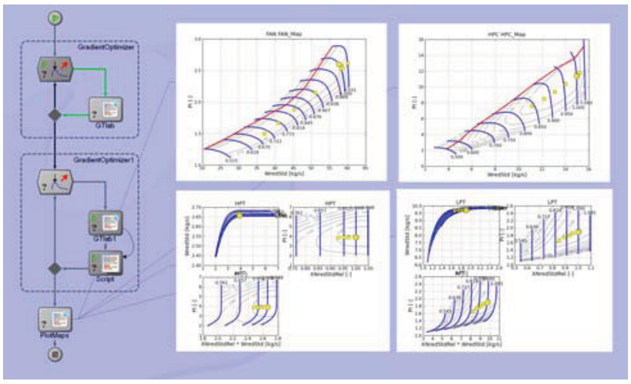

Figure 11: ModelCenter optimization loop and integrated GTlab graphical output

Three design variables are considered in the optimization process and determined by the optimizer:

- Intake mass flow W_2,
- Bypass ratio BPR,
- HPC pressure ratio Π_{HPC}.

The values are restricted in a predetermined range defined by upper and lower bounds. The design range is additionally limited by other physical limitations and technology constraints. These are:

- The thrust limits at all considered engine operating points
- A minimum surge margin at all operating points
- The maximum combustor outlet temperature T4 at mission point 2
- The maximum compressor exit temperature T3 at mission point 2,
- The maximum corrected mass flow at mission point 3
- The maximum design pressure ratio of the HPT at mission point 2 for a single stage design.
- Maximum exhaust gas temperature at point 4

The figure of merit to be optimized by the cycle design is the SFC at cruise condition with the restriction that all off design requirements be achieved within the constraints.

3.3 Results

For the present design case and derived engine requirements a set of feasible thermodynamic cycle design solutions exist. Due to the selection of the cycle layout and constraints as well as the parameters to be optimized the design space has been confined implicitly. Therefore the optimiser is able to identify a single solution, representing the minimum SFC at cruise conditions while not violating any other constraints.

In the optimisation process all design parameters have to be increased in order to minimize SFC by means of decreasing engine specific thrust. Consequently the intake mass flow W_2 and the compressor pressure ratio Π_{HPC} have been set to the maximum available limit, defined by the TOC and take-off conditions respectively. The BPR increase by the optimizer was restricted by the minimum thrust requirements in all discussed operating points. In this case the hot day take-off condition was the most thrust-critical operating point, due to the maximum combustor outlet temperature T4 at that mission point, resulting in a moderate BPR of the engine cycle. The maximum allowed compressor exit temperature T3 could be maintained at any operating condition during cycle optimisation.

A summary of important thermodynamic parameters of the cycle design for the different operating points is shown in Table 2.

Operating Point	TakeOff	TOC	Attack	Cruise	Dash
W2 [kg/s]	54.1	21.3	51.9	19.2	62.8
FN [N]	21	25	27	24	32
OPR [-]	24.7	25.5	24.1	21.0	15.0
BPR [-]	0.84	0.83	0.84	0.84	0.83
SFC [g/kNs]	21.0	24.7	27.4	24.3	31.6

Table 2: Key thermodynamic parameters at investigated mission points

4 Conclusion

GTlab is a capable simulation software for gas turbine performance analysis. The tool is based on an object oriented software concept and provides the means to simulate arbitrary aero engine and power generation cycles. The decision to conform to open standards and software libraries ensures sustainable development and high cooperation abilities.

The attained high flexibility makes GTlab a very suitable tool for gas turbine performance related investigations. Besides the assessment of aero engine and power generation concepts GTlab is presently applied to variety of research applications.

As an application example the program was successfully integrated into the EVITA preliminary design process. The thermodynamic cycle of an UCAV mixed flow turbofan engine was designed to meet the requests posed by the specifications. The synthesis software was seamlessly integrated into ModelCenter where the modelled cycle was improved by a basic optimizer task. The resulting thermodynamic engine model formed the basis for the subsequent preliminary design disciplines.

References

[1] EVITA Project Website, http://www.dlr.de/at/en/desktopdefault.aspx/tabid-7137/

[2] Lytle, J.K., "The Numerical Propulsion System Simulation: An Overview", NASA/TM-2000-209915, 2000

[3] EcosimPro Website, http://www.ecosimpro.com/

[4] Alexiou, A., Mathioudakis, K., "Development of Gas Turbine Performance Models using a Generic Simulation Tool", ASME GT2005-68678, 2005

[5] Visser, W.P.J., Broomhead, M.J., "GSP, a Generic Object-Oriented Gas Turbine Simulation Environment", ASME-2000-GT-0002, 2000

[6] Kurzke, J., "Advanced User-friendly Gas Turbine Performance Calculations on a Personal Computer", ASME 95-GT-147, 1995

[7] Stroustrup, B., "The C++ programming language", Addison-Wesley, 2000

[8] Reed , J. A., Afjeh A. A., "Computational Simulation of Gas Turbines: Part 1-Foundations of Component-Based Models", J. Eng. Gas Turbines Power Vol. 122, 2000

[9] Reed , J. A., Afjeh A. A., "Computational Simulation of Gas Turbines: Part 2- Extensible Domain Framework", J. Eng. Gas Turbines Power Vol. 122, 2000

[10] Open Source Initiative (OSI) Website, http://www.opensource.org/licenses/index.html

[11] SAE Aerospace Standard AS681J, "Gas Turbine Engine Performance Presentation for Computer Programs", 2008

[12] SAE Aerospace Standard AS755D, "Aircraft Propulsion System Performance Station Designation and Nomenclature", 2009

[13] Extensible Markup Language (XML) Website http://www.w3.org/XML/

[14] Münzberg, H.G., Kurzke J., "Gasturbinen – Betriebsverhalten und Optimierung", Springer-Verlag, 1977

[15] Goodwin, D.G., "An open-source, extensible software suite for CVD process simulation", Proc. of CVD XVI and EuroCVD Fourteen, Electrochem. Soc., pages 155–162, 2003

[16] McBride, B.J. et al, "Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species", NASA-TM-4513, 1993

[17] Moré J. J., Garbow B. S., Hillstrom K. E., "User Guide for MINPACK-1", Argonne National Laboratory Report ANL-80-74, Argonne, Ill., 1980

[18] UCAV-2010 Project Website, http://www.dlr.de/as/desktopdefault.aspx/tabid-3574/5578_read-8080/

[19] Siller, U., Voss, C., Nicke, E., „Automated Multidisciplinary Optimization of a Transonic Axial Compressor", AIAA 2009-863, Florida, 2009

[20] Siller, U., Voss, C. " Automated Optimization of a Double S-Shaped Inlet for Minimum Loss and Reduced Sight onto the Engine Face", ISROMAC13-2010-22, Hawaii, 2010

[21] Grieb, H., „Projektierung von Turboflugtriebwerken", Birkhäuser Verlag, Basel, 2004

[22] Grieb, H., „Verdichter für Turbo-Flugtriebwerke", Springer Berlin Heidelberg, 2009

[23] Schaber, R.: Numerische Auslegung und Simulation von Gasturbinen, Dissertation TU München, 2000

[24] Mattingly, J. D. et al., "Aircraft Engine Design (Second Edition)", AIAA Education Series, 2002

[25] ModelCenter™ Website, http://www.phoenix-int.com/software/phx_modelcenter.php, 2011

[26] DOT Optimizer Website, http://www.vrand.com/DOT.html