ISABE2015-20008

Benchmark Criteria to Evaluate 'Expert Systems' for First Preliminary Engine Design

Reinhold W. Schaber, & Fabian Donus
MTU Aero Engines

Abstract

In the last decade many ideas for preliminary gas turbine design expert systems have evolved, allowing the very first design of a gas turbine engine. This design includes the thermodynamic cycle, a first flow path, Fig. 1, and basic aerodynamic design of the engine components, a weight assessment etc.

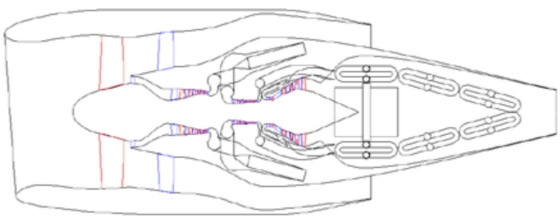

Fig. 1: Example of an Engine Sketch of an Intercooled Recuperated Aero- Engine created by the MTU Pre-Design Tool MOPEDS

Those expert systems reach from relatively simple, commercially available performance programs to complex OEM in-house engine design systems. Most of today's expert systems feature modern graphical user interfaces, Fig. 2 /1/,

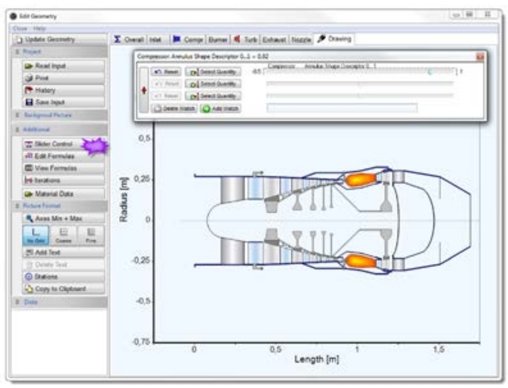

Fig. 2: Example of a modern Graphical User Interface of a Pre-Design Tool /1/

a modular program design, interdisciplinary calculation modes and the option to evaluate the engine component of interest with different methods (1 D, 2 D, …).

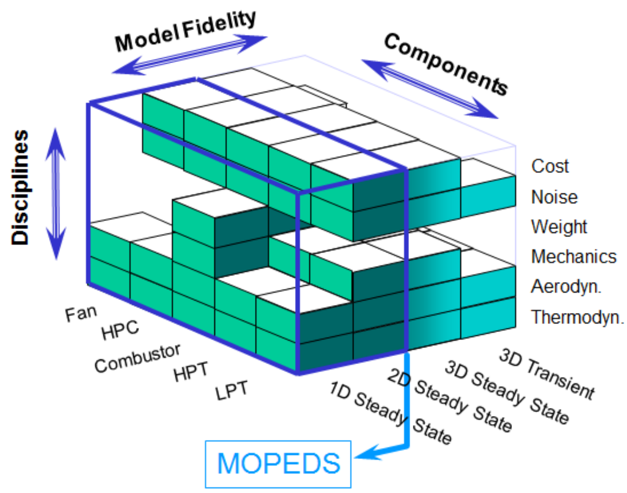

Fig. 3: Concept of Different Levels of Model Fidelity adopted from Claus /2/

However, there are many more features that have to be taken into account to adequately solve a given task.

This paper tries to give an overview how to benchmark such gas turbine design programs. An extensive list of features grouped in main categories will be provided, e.g.

- general requirements
- application of the system
- program environment
- program modules
- gas dynamics
- technology
- program development environment

The features will be described and it will also be explained why and when they are important. This list will allow a thorough check of the program's abilities.

In an industrial environment the time needed to perform required tasks often is crucial. This paper will give some recommendations about which type of tool to use for a given task. It will also illustrate some "standard tasks" where time efficiency is a crucial evaluation criteria.

These tasks comprise:

- building a new propulsion system model
- performing an interdisciplinary parameter study including various operating points
- creating a TSFC map (off-design operating points covering the flight envelope)
- introducing new algorithms

The given catalogue can also be regarded as a basis for a specification for new engine design program systems. The presented paper might not only be interesting for pre-design tool users but also for developers.

Abbreviations

CAD	computer-aided design
CAE	computer-aided engineering
GUI	graphical user interface
MCL	maximum climb thrust
MCT	maximum continuous thrust
MOPEDS	Modular Performance and Engine Design System
MOPS	Modular Performance Synthesis
MTO	maximum take-off thrust
OEM	Original Equipment Manufacturer
TSFC	thrust specific fuel consumption

Foreword

The description of typical tasks and procedures in this paper as well as the evaluation of program features are based on the authors' own experience and partially inspired by work flows in MOPEDS, the MTU Aero Engines in-house propulsion system predesign tool. The given standard tasks are examples only and do not claim to be universally representative.

1 Basic Requirements of Computer Programs for First Preliminary Engine Design

During the conceptual and preliminary design phase of gas turbines, the main parameters are fixed, leading, in return, to already predetermined risks and financial resources associated with the development, manufacture and operation of the engine under concern. The preliminary design process, on the other hand, must be carried out very quickly so that engine suppliers are able to evaluate numerous concepts with respect to the market requirements at a given short notice. This brings up the need for adequate software tools that accurately and quickly guides the designer through the preliminary design process. Main requirements of such a tool are

- assessment of all major engine components and their interrelations;
- inclusion of all relevant disciplines;
- ability to design over several operating points;
- model fidelity zooming at least for individual components

Since the ordinary user of a program system as described above will not be an expert in all fields, it is mandatory that the program system semi-automatically applies rules and makes decisions on how to design the desired propulsion system. This can be considered a - even if sometimes primitive - form of an expert system. Here an expert system is defined as a computer program which helps users solving complex tasks emulating the human decision making process by using a knowledge based databank /15/.

2 Overview of Computer Programs for First Preliminary Engine Design

Most engine design programs of a new generation started to evolve in the 1990s in line with the break-through of object oriented computer program languages. Ambitions always have been high to significantly improve the way propulsion systems are designed. Hereafter some examples for such expert systems for preliminary engine design are given (alphabetic order):

2.1 DLR. PEGASUS

PEGASUS (preliminary gas turbine assessment and sizing) is DLR's attempt to create a brand new multidisciplinary program for the evaluation of existing and future engine concepts /14/.

2.2 GASTURB

The GASTURB development started in the mid-1990s with the objective of creating a performance program for mainly educational purposes. Today GASTURB is a worldwide distributed commercial program used in all fields of engine performance assessment, performance synthesis and performance analysis. Latest versions also feature a flow path assessment and provide the means to assess engine masses /1/, /8/. There are plenty of publications in connection with or based on GASTURB results (given here only a selection: /9/-/12/).

2.3 MTU Aero Engines: Modular Performance and Engine Design System MOPEDS

MTU Aero Engines started the development of a new propulsion system design program as early as in the mid-1990s and had a functional demonstrator in 2000 /3/. To avoid having to start the development from scratch, MOPEDS (Modular Performance and Engine Design System) was based on other programs. The numerical solver as well as the program control parts of the program were shared with the performance program MOPS (Modular Performance Synthesis). Physical components, e.g. mean line programs, were integrated /4/. Apart from reducing the amount of work (data transfer from one program to another etc.) and the probability of making errors MOPEDS was intended to support closer cooperation between the various specialist departments such as thermodynamics, aerodynamics, structures, acoustics, costs already in the early phases of engine programs. MTU has constantly improved and further developed the program system /6/, /7/. Today MOPEDS is one of the main tools in MTU's advanced programs department.

2.4 NPSS

In 1999 Lytle, working with NASA, published information about the Numerical Propulsion System Simulation (NPSS).
This program aims to enable accurate information about propulsion system parameters such as performance, operability and life to be determined early in the design process before any hardware is built and tested. Therefore NPSS provides modeling techniques and data standards to couple the relevant disciplines such as aerodynamics, structures, heat transfer, combustion, acoustics, controls and materials. The system is developed by NASA, industry, other government agencies and universities and consists of engineering models, a simulation environment and a computing environment. Contradictory to most of the other design systems NPSS is commercially available. /20/

2.5 Pratt & Whitney Canada

In 2010 Brophy, working with Pratt & Whitney Canada, published an introduction into the inhouse preliminary multidisciplinary optimization tool PMDO-Lite. Similar to Rolls Royce' Genesis PMDO-Lite is based on a thermodynamic cycle calculation, which provides the input to the preliminary sizing tool. This sizing tool is meant to be the core of PMDO-Lite. The gas path as well as key mechanical dimensions, parametric masses and costs are calculated. The major objective of this part of the tool is the quick turn from cycle data into geometry. This was found to be one of the most critical time paths existing in the preliminary design process. /17/, /18/, /19/

2.6 Rolls Royce:

In 2003 Jones /13/ published details about the military engine preliminary design process used by Rolls-Royce to support 'capability vs. cost' trades conducted at the weapon system level. In order to accomplish that Rolls Royce makes use of the preliminary multi-disciplinary design system Genesis, which has been developed since the early 1970s. This program is able to make basic assumptions about key aerodynamic and mechanical parameters to enable the calculations to be started. Genesis uses Rolls Royce Analysis Program (RRAP) to represent the thermodynamic behavior of the engine.

3 Benchmark Criteria

Benchmark criteria are useful to identify the best suited program for a given task or a portfolio of tasks. They can also serve as a guideline for the development of such programs. Following main features and typical applications are given to help select the required features.

Features can be grouped into the following categories /3/:

- general requirements
- application
- environment
- program modules
- gas dynamics
- technology
- program development environment

3.1 General Requirements

The most crucial general requirements will be explained in detail hereafter.

- Availability of the Program

Engine design systems can be grouped in two categories: in-house codes, usually not accessible to the public and commercially available programs.

The advantages of commercially available programs are:

- No in-house effort to create the program
- Documentation available
- Error prove by multitude of users
- Hotline (potentially)
- No permanent in-house support/programmer necessary

On the other side there are also disadvantages related with commercial software:

- Sales price
- Source code normally not accessible
- Dependency to the seller:
  - o User wanted features / needed modifications cannot be demanded, good will of seller required
  - o Even if modifications are possible, long lead times
  - o Confidential changes to the program generally speaking not possible
  - o Platform changes only possible with cooperation of the seller (if not already implemented)
  - o Realistically seen, updates must be purchased, e.g. long-term financial commitment
  - o No or limited influence on program development direction / philosophy
  - o Depending on license agreement no flexible usage of the program (platform / computer change, varying users, etc.)

Apart from access to the program, the program's maturity should also be very important to the user. Programs which have been around for several years usually feature more user friendliness and less bugs. Here publications can help the user to get an idea.

- Calculation of Random Engine Configurations

Typically, there are two fundamentally different approaches in engine design programs to provide the information about the engine configuration to be simulated (mixed 2-spool turbofan, unmixed 3-spool engine, turboprop, etc.). In the first case predefined engine configurations can be selected by the user (e.g. GASTURB). These configurations usually differ only in the main characteristics such as number of spools, mixed or unmixed, propeller engine or turbofan, etc.

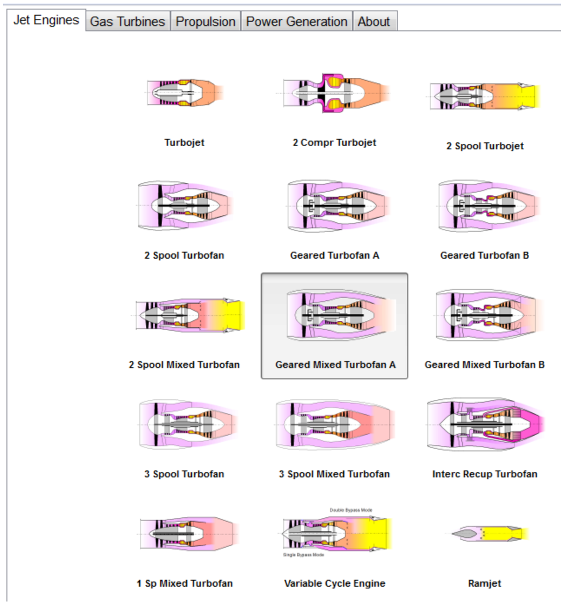

Figure 4: GASTURB 12 Engine Selection /1/

Advantages:

- o Tested and reliable configurations immediately available and ready to use
- o Available methods tailored to the specific configuration

Disadvantages:

- o Only given configurations can be used
- o Modifications of configurations cannot be introduced by the user

The second approach allows the user to provide all information about the configuration manually.

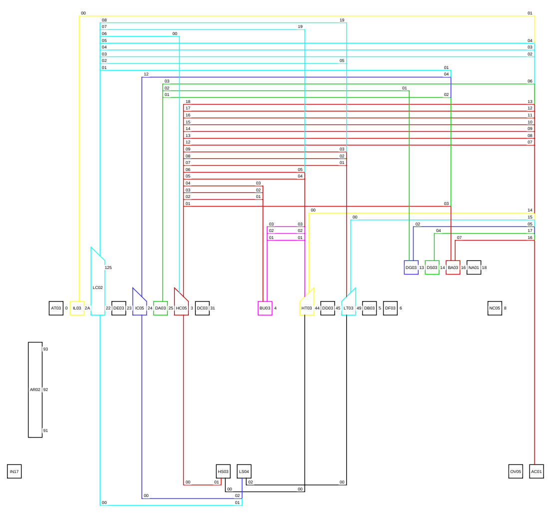

Figure 5: Example for a user defined engine configuration

Advantages:

- Random configurations (especially also unconventional configurations) can be built according to the actual requirements
- Unusual components can be added (pistons, electrical components, …) Communality between different models can be created and hence simplify program handling
- Once a set of different configurations is present and automatically provided to the user the program can also be used as in the “predefined configuration” mode

Disadvantages:

- Building a new configuration often is a tedious task which might require a certain expertise – usually this task is not done frequently

To overcome the disadvantages model libraries, i.e. predefined thoroughly checked models of certain engines, can be used.

- Multi-disciplinary Design Capabilities

Modern expert systems for the preliminary engine design have to comprise a vast range of aspects to be looked at, taking into account thermodynamics, aerodynamics, mechanical design, mass and cost assessment. Here some examples:

Baseline usually is the performance calculation which provides the fundamental cycle data, i.e. temperatures and pressures for all stations (module entry planes). Today’s performance programs, in addition, assess mass flows and provide station areas in design and off-design condition. For the cycle off-design calculations it should be possible to use any type of standard component map.

Based on the performance results, a first flow path can be derived. With the flow path aerodynamic mean line, analyses can be performed leading to a first aerodynamic assessment resulting in component efficiency and aerodynamic loading. Based on the engine flow path, the nacelle contour and its associated drags can be assessed. With this information, gas and acoustic emissions at component and at system level can also be determined. Typical missions can be calculated and the fuel burn assessed.

With flow path and mechanical speeds, a first mechanic design of disks and casings can be performed. For this task a material data base is very helpful.

Mass and cost – both production cost and maintenance cost - can be estimated based on the geometry and cycle information. Statistical and analytical methods are both possible.

With the engine design the effects on the aircraft can be assessed. This might allow an optimization of the overall aircraft system.

- “Zooming”

Modern expert systems for the first preliminary engine design usually do not only provide one single method to assess a particular piece of information. The user can choose between different methods to obtain the level of depths wanted ('zooming'). E.g. there can be a pure mean-line analysis, a multi-line analysis up to a CFD analysis to assess the component’s aerodynamics.
This can be useful e.g. within large parameter studies. When time is crucial, it is often advantageous to use simple methods in order to narrow down the area of interest. Subsequently more detailed methods can be used to either strengthen the findings or for special investigations of a selected component (e.g. 2D calculations with HPC only).

- Design Traceability

Crucial for any work with such a mighty tool as described is the traceability of the calculation. It is important to be aware of the “heritage” of a result (starting point, design laws, etc.). Typical features are:

- the use of standard plots to evaluate the physical soundness, e.g. organized in a plot catalogue, including special plots as operating lines in the component maps, t,s-diagrams, smith diagrams, velocity triangles etc.,
- suitable means to inform about the calculation progress (convergence etc.)

- Extensive documentation

The following features are important supplements to the program:
- user Manual
- documentation of knowledge and rules
- documented Source Code
- tutorials
- online-help
- additional documentation

3.2 Application

The main objective of a program system for conceptual engine design is to allow the user to as quickly as possible fulfil his task at the desired quality of the results. A simple tool application is key. The risk of application errors and calculating physical non-sense should be minimized. Consequently the data processing for input and output is highly important. Here the program system can assist the user in multiple way:

- Graphical User Interface

Especially commercially available programs like GASTURB often feature excellent graphical user interfaces (GUI). The importance of the GUI is reciprocal to the usage of the program. Occasional users certainly require a more sophisticated interface compared to regular users. For regular users (e.g. daily 4 h) speed to achieve a certain objective might be a by far more relevant feature of the program system. The optimum is – of course – a mixture of user friendliness and an interface enabling extremely fast operations.
GUIs have to be divided in those controlling the calculation process, facilitating the data input process and those providing output. A special case is the batch mode which e.g. enables the program execution within optimization environments. GUIs are particularly useful if data quantities are small. On the other hand large data quantities (e.g. the calculation of a set of spot points containing several thousands of operating points) are usually easier handled by file based input and not GUI supported.

- Suitable Program Operation Modes

Depending on the task an interactive or a batch mode is preferable. An interactive GUI can significantly support to build the design point of a new engine – quicker access to the required input data by input data categories, graphical checks of input parameters, etc. The user can better concentrate on the physical content of the task. On the other hand a batch mode is required if large quantities of data have to be repeatedly processed, e.g. when generating TSFC maps. Here it is important that changes in the input data do not require a new set-up of the whole job.

- Automatic calibration

In case a parameter study has to be performed, a standard procedure is to base the calculation on a known reference engine. Here, the geometry and component efficiencies predicted from mean line codes, mass and cost assessments, etc. will be calibrated. Calibration can be performed manually or – faster and more convenient - supported by the system.

- Easy data input and output

Easy data input and output is key to successfully solve a given task. The following features can help:

- nomenclature acc. to standard
- task oriented input / output features reasonable input data structure
- reasonable output data structure
- existence of comprehensible log-files
- database with calculation input and output parameters

- Nomenclature acc. to a standard

The more complex the task, the more important, not only for the individual user but also when teams of different institutions cooperate, is to name parameters according to standards (e.g. SAE AS755 /16/).

- Reasonable input data structure
    - adequate number of input parameters
    - adequate structure of input parameters
    - readable input file
    - parameter names adaptable by user
    - extensive usage of graphics for data input
    - only needed input can be identified / is shown
    - input parameter unit can be independently selected

- Reasonable output data structure
    - adequate structure of output parameters
    - output after each operating point
    - output parameter unit can be independently selected
    - only output is created which is relevant for the selected methods
    - creation of a binary database
    - adequate number of output parameters
    - extensive usage of graphics for data output

- Existence of comprehensible log-files

It is absolutely crucial that the user knows in detail how a calculation was performed, what program version was used, which input was used, what kind of problems / errors occurred, how iterations advanced, etc.
This is especially important if the calculation terminates in an unexpected way.

- Database with calculation input and output parameters

Often it is important that the input and results can be retrieved at any time after a calculation without performing the calculation itself again. So e.g. the evaluation of certain calculated parameters can be done later at any point in time.

- User Guidance

There are many fields the program can ease tasks where the program can speed and lighten up the otherwise time consuming and cumbersome activities. Two examples for such activities are:

- automated checks of engine configurations, models and calculation results
- automated set-up of parameter studies

In the first case, the program can execute checks regarding the model structure, used data, but, also, to a certain extent, physical soundness of the model.
In the second case, the system should provide the complete input for a whole parameter study. The user then only has to provide the variable parameters with their respective range.

- Availability of auxiliary tools

With today's high performance computers, the pure computing time, especially when using simple 1D or 2D-methods, is often not crucial. Instead the set-up of the model and rather the validation of the simulation's results as well as the preparation of the documentation take a large portion of the overall time. Therefore auxiliary programs are needed to most effectively support the user in these time consuming tasks. The following features offer assistance:

- Plot software with direct access to calculation results
- Creation of user defined parameter listings
- libraries with component maps
- libraries with predefined documented and checked engine models and components

- Availability of Help Features

The following help features shall be available:

- online-help describing the physical content and standard tasks
- help to recognize errors and troubleshooting
- predefined plot libraries to graphically check results
- assistance to implement program extensions

3.3 Environment

Often, there exist boundary conditions which demand certain features. Following the main features are described.

- Independency of platform

The program should run on Microsoft and Unix operating systems.

- Provision of standard interfaces

To accelerate data handling the following interfaces are helpful:

- CAD and CAE interfaces
- Microsoft interfaces (Excel, Powerpoint)
- plot software interfaces

- Program security

It should be possible to especially restrict the access to input and output data. In case computer decks could be transferred to non-in-house customers, it should be possible to “scramble” the computer code (i.e. make it unreadable).

- User environment

Instead of a prescribed, frozen environment a flexible one will allow the environment to be optimized according to the users' preference.

3.4 Program Modules

The most important features are:

- Modular design

The following modules (or options to calculate) should be available:

- turbo machines (compressors and turbines; propellers)
- inlets, ducts, nozzles
- heat exchangers
- aircraft related calculations (nacelle etc.)

- Numerical stability of applied methods and programs
  - Starting Values can be prescribed at any operating point
  - Starting Values can be prescribed automatically

- Clearly arranged computer source code
  - communication between different calculation modules should be possible
  - documentation of the source code

- Uniform subroutines

3.5 Gas Dynamics

Regarding the actual physics, there are also some important requirements:

- Convenient design and off-design calculations

  - possibility to design the engine in any operating point
  - off-design calculations using component maps
  - off-design: various component map representations
  - off-design: calculation without design point possible
  - access to arbitrary operating point data
  - enabling / disabling of component specific design / off-design calculations (mixed design / off-design among various components)
  - automatic calculation of multi point methods such as ICAO noise or gas emission certification

- Possibility of transient performance calculations

3.6 Technology

The following features significantly ease the program handling:

- User defined iterations

For the convenient adaptation of user prescribed models it is crucial that the user can implement arbitrary iterations according to his needs, including iterations involving more than one operating point .

- Special calculation modes

- optimization (already built in or by external means); different optimization strategies should be available
- optimization of the complete propulsion system and portions of it
- Monte Carlo Analysis

- User defined algorithms can be implemented without a program change

- Use of arbitrary tables

Often it is required to introduce new dependencies in a model. An easy way is the implementation of tables. Those tables should be accessible without any code modification.

3.7 Program Development Environment

In commercial applications the program development environment plays an important role. Following a list with important features:

- update process and responsibilities clearly defined
- downward compatibility
- minimized test time and effort for updates
- version controlled program storage
- possibility of arbitrary extensions
- possibility to include specialists' programs and algorithms
- selectable calculation accuracy
- possibility to create computer decks, DLLs etc.

4 Selection of an adequate Expert Systems for First Preliminary Engine Design

The selection of a suitable tool for a given task often is not easy. The user's requirement in general is to accomplish the job

- as fast as possible
- without errors
- in an adequate quality.

Also the tool has to cover the entire range from providing results in a few minutes up to extensive studies of several month.

The selection of an adequate expert system mainly depends on the task the engineer has to perform. Extensive tasks where many users are involved may certainly require more sophisticated tools than the quick "first glance assessments" of an individual user. Consequently the requirements for the tool are totally different.

For quick tasks, the engineer cannot afford to build up new model configurations, create huge new data sets etc.. Following several examples of typical tasks engineers in the very early phase of a new engine program are faced with.

4.1 Standard Task 1: Building a new propulsion system configuration and set up of a new simulation model

In the case of predefined configurations, a substantial part of the task is already provided by the program, given an appropriate configuration exists. In this case, "only" the cycle parameters have to be provided. The capability of the implementation of user driven adaptations is very useful in case the data structure or rather the data set-up does not meet the user's necessities. Usually, templates for input data are provided which eases the task. The downside is the inflexibility of the configuration which might prevent that special requests needed by the user can be implemented.

On the other side, modular program architectures require already substantial effort to build the engine configuration. Here suitable tools may help, e.g. graphical interfaces where the configuration can be built interactively, configuration libraries from which the user can select a suitable configuration as a starting point. This configuration then only has to be adjusted.

Along with the configuration a library can also provide sample input and output data for various models. In case reference data is available it can be used for calibration. The calibration logic can either be implemented in the tool or needs to be provided by the user e.g. by defining iterations.

To fulfil this kind of tasks it is important that graphical configuration tools exist which help to select or build new configurations, easy data input including user support to select adequate component characteristics, automatic calibration of a model to a set of reference data, clear organization / implementation of "laws" for iterations, support of a quick result check (graphical and numerical).

4.2 Standard Task 2: Building a parameter study to evaluate a new optimum cycle for a given set of boundary conditions

One of the main tasks preliminary design tools are used for is the identification of the optimum engine design for a given set of boundary conditions such as thrust, power off-take, ECS bleed, fuel burn target, etc.. Typically a baseline engine is modeled, which serves as a starting point for studies in which parameters like bypass ratio, overall pressure ratio and combustor exit temperature are varied. The design rules, which are applied by the program, can either be integrated in the tool, e.g. scaling rules for flow path geometry, mean line codes for the evaluation of component efficiencies, or be provided by the user, which is usually done using iterations, e.g. adapt spool speed for constant reduction gear ratio, etc. The typical scope of such a study is 30 engine designs incorporating 3 to 10 off-design operating points (MCL, MTO, noise and gas certification points). For this task the following steps have to be executed:

- set-up and calibration of a reference engine model (preferably using available templates or retrieved from existing models) as a starting point (see standard task 1)
- creation of the control laws to be applied during the study:
  - o which parameters should be varied (e.g. BPR, OPR, T4)
  - o which physical parameters should be adapted (e.g. flow path, component efficiencies, cooling air flows, etc.)
  - o how should those parameters be adapted?

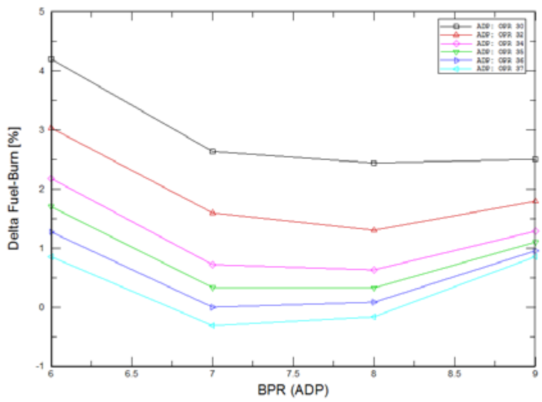

Figure 6: Example of a parameter study

This task especially requires fast and robust methods, the ability of iterations over different operating points and extensive predefined but adoptable graphical checks.

4.3 Standard Task 3: Provision of a TSFC Map

A typical task is the provision of so called TSFC-maps for customers, often aircraft manufacturers or research institutes.
These maps are used for the design and optimization of new or modified aircrafts and incorporate a large quantity of scans covering all significant operating points of the flight envelope: maximum take-off operating points at different altitudes and ISA conditions, maximum climb and maximum continuous at differ-

ent flight Mach numbers, altitudes and anti-icing bleed conditions, cruise at different thrust ratings (from max. cruise to flight idle). A typical number of operating points for a TSFC-map is in the order of 5000.

The special “problem” of this task is the huge amount of data which has to be created.

The following steps have to be executed to create a TSFC-map:

- the basic engine model has to be set up; key performance parameters have to be fixed (see also chapter 4.2)
- the “rating structure” has to be defined, i.e. the rules according to which the engine is controlled:
    - how is “MTO” defined
    - how is MCL and MCT defined?
    - definition of the handling bleed schedule
- the “rating structure” has to be implemented
- input for all operating points to be calculated has to be provided
- the results have to be checked. Due to the huge amount of data adequate utilities have to be used, e.g. graphical checks
- the data sets (tables) for the customer have to be created.

4.4 Standard Task 4: Introduction of a new Algorithm

Sometimes, a project requires that the way certain parameters are calculated is modified or that completely new methods are introduced to the program source code. Here of course, those programs have an advantage where the user directly and immediately can

- modify the source code
- create a new executable
- run the modified program
- repeat the above described steps as many times as requested

without interfering with the program usage of other users. Useful features here are:

- separation of program control and physical modules; users only have to modify the modules containing physical algorithms
- direct access of users to the modules containing physical algorithms
- source code in a widely spread computer language (e.g. C++, Fortran)
- version control and release management to ensure the implementation of user driven changes for the benefit of all program users

This task requires clear rules and processes for the update of the program system provided that changes shall be accessible for all users later.

The rules should comprise:

- nomenclature of parameters
- programming style
- downward capability

5 Selection of an adequate tool for a standard task

According to the recommendations in Chapter 4 in the Annex extensive lists are provided, trying to give hints which features are needed to fulfil which task, based on the above described features and tasks. Categories are grouped in 5 levels:

0: not necessary
1: nice to have
2: useful
3: recommended
4: needed

Please refer to the Annex.

6 Conclusions

Performance and engine design programs for preliminary gas turbine engine design have been around for more than a decade. However, there are plenty of tasks such a program has to cope with. Accordingly, the requirements for such a program vary a lot.

There are crucial basic requirements a program has to fulfil and others more dependent on the individual task. Here four “standard tasks” have been introduced and the related features have been explained. The importance of individual features not only depends on the main tasks the program has to fulfil but also on the frequency the program is used. Occasional usage clearly requires different features than intensive daily usage.

This paper gives guide lines which features are beneficial for the individual tasks to allow a potential user or developer of such a program to determine the most important features tailored to his individual needs.

<u>References</u>

/1/ Kurzke, Joachim
GASTURB 12 - Design and Off-Design Performance of Gas Turbines
Germany, 2012

/2/ Claus, R.W. et al.,
Numerical Propulsion System Simulation,
Computing Systems in Engineering, Vol. 2, No. 4, 1991

/3/ R. Schaber
Numerische Auslegung von Gasturnen
Dissertation, München, 2000

/4/ R. Schaber, P. Jeschke, J. Kurzke
Konzept eines multidisziplinären Vorauslegungsprogramms für Gasturbinen
DGLR-Jahrestagung 2001, Hamburg

/5/ Jeschke, Kurzke, Schaber, Riegler
Preliminary Gas Turbine Design Using the Multidisciplinary Design System MOPEDS
ASME Journal of Engineering for Gas Turbines and Power, Vol. 126, Jan. 2004

/6/ F. Donus, R. Schaber, Schmidt, Staudacher
Accuracy of Analytical Engine Weight Estimation during the Conceptual Design Phase
ASME 2010, Glasgow

/7/ Donus, Fabian; Bretschneider, Stefan; Schaber, Reinhold; Staudacher, Stephan
The Architecture and Application of Preliminary Design Systems
ASME 2011, Paper GT2011-45614, Vancouver

/8/ Donus, Fabian
Ermittlung der Genauigkeit und Erarbeitung von Verbesserungsvorschläge bei der Gewichtsabschätzung mit Gasturb 11
Diploma Theses, Munich, 2008

/9/ J. Kurzke
Advanced User-Friendly Gas Turbine Performance Calculations on a Personal Computer
ASME 95-GT-147, Houston, 1995

/10/ J. Kurzke
Gas Turbine Cycle Design Methodology: A Comparison of Parameter Variation with Numerical Optimization
Journal of Engineering for Gas Turbines and Power January 1999

/11/ J. Kurzke
About Simplifications in Gas Turbine Performance Calculations
ASME GT2007-27620, Montreal, 2007

/12/ J. Kurzke
Transient Simulations During Preliminary Conceptual Engine Design
ISABE 2011-1321, Gothenburg, 2011

/13/ Jones, M.J., Bradbrool, S.J., Nurney, K
A Preliminary Engine Design Process for an Affordable Capability
Defense Technical Information Center
Compilation Part, Notice, ADP014191

/14/ DLR
http://www.dlr.de/at/desktopdefault.aspx/tabid-1547/2182_read-3654/,
down loaded 19.02.2015

/15/ Jackson, Peter
Introduction To Expert Systems
1998 (3 ed.)

/16/ SAE
Aircraft Propulsion System Performance Station Designation and Nomenclature
AS755 Rev. C, 1997

/17/ Pachenko,Y.,Moustapha,H.
Preliminary multi-desciplinary optimization in turbomachinery design
2002

/18/ Prado,P., Panchenko,Y.
Preliminary multi-desciplinary design optimization system: a software solution for early gas turbine conception
2005.

/19/ Brophy,F., Mah,S.
Preliminary multi-desciplinary optimization (PMDO) an example at engine level
2010

/20/ Lytle, J. K.
The Numerical Propulsion System Simulation - A multidisciplinary design system for aerospace
1999

/21/ Irwin, K.
Multi-Disciplinary Design Optimization *using WAVE*
2000

Annex

A: General requirements

Feature	Standard Task			
	1	2	3	4
	Model Set-up	Parameter Study	TSF-Map	New Algorithm
Affordable price level	4	3	2	2
Possibility of implentation of user needed features	3	3	2	4
Hotline	3	3	2	2
Error-proof, mature program	2	3	4	2
Source code accessible	2	2	2	3
Eninge configurations can be defined by the user	4	3	3	3
Library with configurations available	3	3	3	1
Ability to create new designs and to assess given geometries	4	3	3	0
Cycle assessment	4	4	4	0
Flow path assessment	4	4	0	0
Aerodynamics (mean line assessment)	4	4	0	0
Mechanics (compoenent sizing [disks, casings, …)	3	3	0	0
Weight assessment	4	4	0	0
Center of gravity assessment	4	4	0	0
Assessment of moments of inertia	4	4	0	0
Production Cost assessment	3	3	0	0
Maintenance Cost assessment	3	3	0	0
Nacelle geometry assessment	1	1	0	0
Nacelle drag assessment	1	1	4	0
Component noise assessment	3	3	2	0
System noise assessment	3	3	0	0
Emissions	3	3	2	0
Data base of material properties	3	3	0	0
Aircraft design	3	3	0	0
Aircraft mission calculation	3	3	0	0
Special investigations with a selected component	3	2	0	0
Consistency of methods of different profoundness	3	1	0	0
Plot catalogue	3	4	4	0
Plot of h-s-Diagram and T-s-Diagram	2	1	0	0
Component map plots incl. Operating line	4	4	4	0
Smith-Diagrams, velocity triangles	4	3	0	0
User Manual	3	3	3	1
Documentation of knowledge and rules	4	4	3	4
Documented Source Code	4	3	2	4
Tutorials	2	2	2	1
Online-Help	4	4	2	0

B: Application

Feature	Standard Task			
	1	2	3	4
	Model Set-up	Parameter Study	TSF-Map	New Algorithm
GUI (graphical user interface)	4	1	1	0
Suitable program operation modes: interactive	3	2	1	0
Suitable program operation modes: batch	1	3	3	3
Automatic calibration (Tool vs. Model calibration)	3	0	0	0
Nomenclature acc. to standard	3	3	3	3
Task oriented input / output features: interactive	3	2	0	0
Task oriented input / output features: batch	1	3	4	3
Resonable Number of Input Parameters	3	4	2	0
Structured Input Paramters	4	4	3	0
Readable Input File	1	3	3	0
Adaptable Names in Input File by user?	2	1	1	0
Extensive usage of graphics for data input	3	2	1	0
After selecting desired calculation methods only needed Input is shown	3	3	3	0
Output after each operating point	3	3	3	0
Output independed of desired property unit	3	3	3	0
Output can be user defined	3	3	3	0
Only output is created which is relevant for desired selected methods	3	3	3	0
Output is binary	1	3	3	0
Output Paramters structured?	3	3	3	0
Extensive usage of graphics for data output	3	4	4	0
Existence of comprehendible log-files	4	3	3	4
Database with input and output parameters	3	3	3	0
Autmated checks of resuts	3	3	3	3
Autometed set-up of parameter studies	0	3	0	0
Plot software with direct access to calculation results	3	3	3	0
Libraries with component maps	3	3	2	0
Libraries with predefined engine models and components	3	3	1	0
Online-Help	4	4	2	0
Help to recognize errors and troubleshooting	4	4	4	0
Predefined plot libraries to graphically check results	3	3	3	0
Assistance to implement program extensions	1	0	0	3

C: Environment

Feature	Standard Task			
	1	2	3	4
	Model Set-up	Parameter Study	TSF-Map	New Algorithm
Platform indipendent	3	3	3	3
CAD and CAE interfaces	3	3	3	3
Microsoft Office interfaces	2	2	2	0
Plotsoftware interfraces	3	4	4	0
Scrambeld Decks possible	0	0	4	0
Data Safety (access limitations can be implemented)	4	4	4	4
Adjustable by user	2	2	2	0

D: Program Modules

Feature	Standard Task			
	1	2	3	4
	Model Set-up	Parameter Study	TSF-Map	New Algorithm
Full Set of components	4	4	3	0
Arbitrary secondary air, power and heat connections	4	3	3	0
Starting Values can be prescribed at any operating point	3	4	4	0
Starting Values can be prescribed automatically	2	3	3	0
Nomenclature acc. to standard	3	3	3	3
Documentation of the source code	4	4	3	4
Uniform subroutines	0	0	0	3

E: Gas Dynamic

Feature	Standard Task			
	1	2	3	4
	Model Set-up	Parameter Study	TSF-Map	New Algorithm
Possibility to design the engine in any operating point	4	3	3	0
Off-DS: various component map representations available	3	3	3	0
Off-DS: calculation w/o Design Point possible	0	0	2	0
access to arbitrary operating point data	4	4	4	0
Enabling/Disabling of component specific design/off-design calc	4	4	4	0
User selectable gas composition	3	3	3	0
Transient calclulations	1	1	0	0

F: Technology

Feature	Standard Task			
	1	2	3	4
	Model Set-up	Parameter Study	TSF-Map	New Algorithm
User defined iterations	4	4	4	0
Iterations over operating points	4	4	0	0
Optimization of the complete propulsion system and portions of it	3	3	0	0
Monte Carlo	4	3	0	0
New algorithms w/o program changes	3	3	2	0
Use of arbitrary tables	4	4	4	0

G: Program Development Environment

Feature	Standard Task			
	1	2	3	4
	Model Set-up	Parameter Study	TSF-Map	New Algorithm
Responsibility clearly defined	2	2	2	4
Update process clearly defined	2	2	2	4
Program Version Control	4	4	4	4
Automated porcedures	3	3	3	4
Downward compatibility	4	4	4	4
External programs can be implemented	4	4	1	0
Possibility of arbitrary program extensions	4	4	4	4
Creation of computer decks, DLLs, etc.	3	2	4	0