ASME/IGTI TURBOEXPO Conference 2004
June 14-17, Vienna, Austria

GT – 2004 – 53770

FRAME, A TOOL FOR PREDICTING GAS TURBINE CONDITION AS WELL AS RELIABILITY, AVAILABILITY PERFORMANCE

Timot Veer, Andreas Ulvestad, Olav Bolland
Norwegian University of Science and Technology
Department of Energy and Process Engineering
N-7491, Trondheim, Norway
E-mail: veer@maskin.ntnu.no, Tel.: +4773598462, Fax: +4773598390

ABSTRACT

A complex software tool (FRAME) for assessing lifing parameters as well as reliability and availability performance of gas turbines is presented in this paper.

The herein described tool is developed by the authors, whereas the implemented know-how originates from field experience gained by condition and health monitoring of gas turbines operating offshore in oil industry. The authors focus on structure, functionality as well as possible benefits of its use. In the second part of the work results of several simulation runs and parameter variations are presented.

Although the primary goal is to test the utility and functionality of the presented tool, interesting and relevant results are obtained. Due to space limitations we restrict ourselves to discuss washing/time interdependence when presenting the findings herein.

INTRODUCTION

Within the energy industry, there is a strong focus on increasing availability of the systems. Besides, maintaining reliability performance becomes an issue of primordial importance too. Stops, outages due to failures as well as control over derating of the systems are therefore key topics when it comes to economic concern within operation of the machinery. Purely theoretical approaches or idealized cycle models of turbomachinery are sometimes insufficient in order to accurately map the effects of external factors acting upon the systems. On the other hand, in case of long term analysis, forecast or analysis of multiple superposed effects, these approaches regularly fail to give accurate answers [1,10,19,22,23,27].

The authors suggest an alternative to solve this problem. A numerical tool of a relatively complex structure is presented: thermodynamic modeling is the core of the system. An intelligent frame is built around this core. The function of this latter frame is to simulate the environment around the machine: generates the time-dependant frame conditions (inputs), monitors the key operation parameters, simulates the decisions that would occur during operation (e.g. corrective maintenance actions) and last but not least performs archiving of outputs after each run. The whole package undergoes a series of runs, so simulating the operation of a machine between to time coordinates. The archived data is used in similar manner as it would have been obtained by measurement. Prediction and harmonizing of the maintenance actions can be successfully performed, as well.

When is the right time to perform crank washing? What is the suitable operation and maintenance policy? What is the current remaining lifetime of the gas generator? What shall the remaining lifetime be under given conditions at a time in future? Answers are sought for these and other typical operational questions under *given or variable frame conditions.* In order to give the forecast a better accuracy, the thermodynamic response of the machine and the surrounding framework (ambient conditions, operation and maintenance policies) are modeled in detail. Close cooperation with offshore gas turbine operators allowed us a continuous validation of the approach against field data. The know-how incorporated in the software is also based on data originating from field measurements.

THE GOAL OF THE PAPER

A short insight in motivation of the current work is given as well as the international framework is overviewed. The software tool is then presented. Structure, operation algorithms and utility are the main concerns. The flexibility and functionality is tested then results are presented and commented.

The overall conclusions are finally summarized and possible development directions outlined. Both methodological as well as technical aspects are handled herein.

BACKGROUND

Interesting interpretations of results or proven correlations between operation, personnel, maintenance and the resulting R&A data can be found in literature [3,5,6,7,19]. Manufacturers bring also an important contribution to the published information regarding impact of novel solutions or improvements upon reliability and availability, [11,12,30,16,].

Different theories, methodologies and statistical considerations were also developed to prognosticate degradation, failure, remaining lifetime, creep and crack propagation [2,7,15,17,25,34]. This represents a new group of publications in the field of lifing assessment at subcomponent level. Statistics is heavily involved in this effort to formulate methodologies which comprise these phenomena of a dominant stochastic feature. Simultaneously, detailed creep, stress and temperature effect models at subunit level are published by Sampath [26] as results of research in lifing and fault analysis. No reference is found to prediction as well as possibility to run different operating scenarios, age, maintenance policies.

As it regards the derating and degradation analysis of the machinery as a unit, important results are published by Perkavec [22] and Schepers [27]. Detailed discussion of employing basic computational tools in assessing fouling, and fouling-washing optimization for given machinery is presented. These discussions are mainly based on history data processing, thus a certain lack of flexibility is reported. Most recently, Silva [28] presents a macro view (system level) upon corrective maintenance planning in a CC plant. Linear fouling and degradation models are used in order to optimize washing intervals. It is not clear whether or not more complicate operating scenarios are possible to be modeled and thus assessed. In conclusion, the operational behavior of a machine, subject to different operating scenarios deserves an increasing attention unrevealed by international literature, yet.

Current work is aiming to overcome the lack of such tool which primarily enables analysis and prediction of reliability, availability and remaining life of gas turbines. Numeric, software based approach is suggested within this work. It is emphasized the statistical nature of both the inputs and outputs.

The authors introduce the following important operational parameters EOuH, lim_w and EOpH. The following subsections give a detailed definition of these criteria.

Equivalent Outage Hours (EOuH)

Equivalent Outage Hours criterion is defined by the authors based on [4,12,14]. It reflects equivalent loss in production due to both outages and progressive performance loss caused by incipient or degraded failures. EOuH helps comparing the non-availability of different operating or maintenance scenarios. The model used for calculating EOuH in this project is:

$$EOuH(\tau) = OH + \int \left(\frac{D_f(\tau)}{P_{0,f}}\right) d\tau + \int \left(\frac{D_g(\tau)}{P_{0,g}}\right) d\tau \qquad (1)$$

Where:

- EOuH = Equivalent Outage Hours, [h]
- OH = Outage hours due to maintenance or washing
- D_f = Deviation from reference power (base line) caused only by fouling
- $P_{0,f}$ = Relative base line with respect to fouling, *clean state*.
- D_g = Deviation from reference power (base line) caused only by degradation
- $P_{0,g}$ = Relative base line with respect to degradation, *new state*

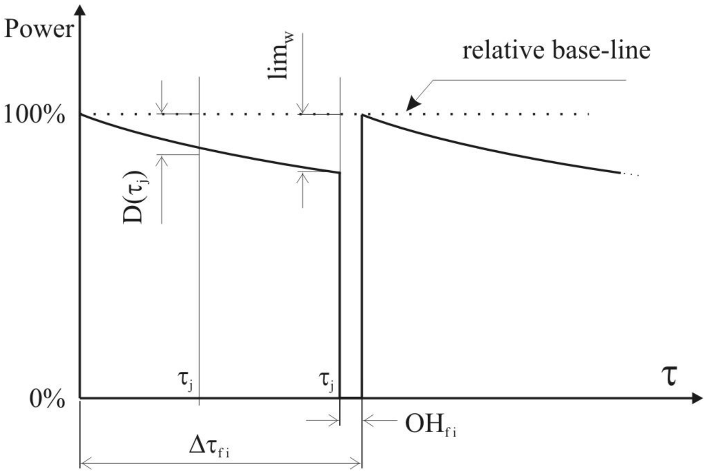


Figure 1., explanatory scheme for Equivalent Outage Hours criteria (only fouling)

Limit of fouling before crank wash, *lim_w*

It is necessary to introduce this term in order to evaluate the current state of the machine with respect to fouling. ***lim_w*** is the maximal amount of derating (due to fouling) the operator allows before performing washing of compressor. The mathematic form of the criteria is described by:

$$\mathrm{lim}_w = \max[\, D_f(\tau)] \qquad (2)$$

Whereas :

$D_f(\tau)$ is derating at time τ, caused only by fouling; as in Fig 1.

Equivalent Operating Hours (EOpH)

Equivalent Operating Hours (EOpH) criterion illustrates the *real usage* of the machine. With other words, the operation manner defines how the machine life is consumed for the same number of *fired* hours. This parameter is mainly employed in machinery lifing assessments.

The authors define an incremental definition especially suitable for the tool presented in this paper.

$$EOpH_j = EOpH_{j-1} + \Delta EOpH \ \ \text{[h];} \qquad (3a)$$

Where:

$$\Delta EOpH = f_f \cdot \mathrm{f}_C \cdot \left(f_S + f_A + f_L \cdot \sum_j f_{P,j} \right) \cdot \Delta\tau_{j,\,\mathrm{j\text{-}1}} \qquad (3b)$$

And, consequently:

- EOpH_j is EOpH corresponding to moment ***j***,
- EOpH_{j-1} is EOpH corresponding to moment ***j-1***,
- ΔEOpH is the incremental EOpH in time interval $\Delta\tau_{j,\,\mathrm{j\text{-}1}}$
- $\Delta\tau_{j,\,\mathrm{j\text{-}1}}$ time interval between moments ***j-1, j***,[1].

---

[1] in our case the time step of the calculations repetitions *1 hour*

- f_F = Factor depending on type of fuel
- f_C = Factor for wet or dry control
- f_S = Factor, that takes into account number of starts
- f_A = Factor, that takes into account the number of excessively steep accelerations, decelerations
- f_L = Location factor
- f_P = Load level factor

the factors ***f*** above are further on described in ***Appendix***, based on [7,11].
More detailed analysis and interpretation of EOpH and EOuH can be found in [33].

Predefined set of general requirements

A survey of the most important expectations towards accomplished beforehand. The main purpose for that was to map requirements towards such a system in concordance with the previously formulated needs. Nevertheless, we intended to delimit the set of frame conditions for choosing means (softwares, algorithms) in order to achieve these goals.

A quick glance produced a fairly great number of aspects that were going to be considered. The authors would classify them as follows: software issues and technical issues, respectively.

Software issues: the most important requirement was to set up a system that is stable, quick[2], flexible and has minimal hardware/software requirements. Additionally, the memory management was an important factor, as long as it was foreseen a great number of data to be conveyed.

Due to the character of academic working teams, special concern had to be given to ease in further development and user friendliness in setting new runs and/or different calculation scenarios. Thus, a module based structure was a must in order to allow independent modular development.

Technical issues: in order to facilitate parameter variation type of research, high flexibility and ease in set up of input-generating routines are the most important viewpoints. Functions, constants as well as on/off setting of several input parameters are a must; all this without *any* programming intervention. On the other hand, the thermodynamic modeling *subsystem* has to enable access/modification of a great extent of internal parameters like: settings (fuel, TIT, VGV angle, fouling), efficiencies, PT speed, etc… The possibility to flexibly modify the thermodynamic modeling part without *any* intervention in the core software (and vice-versa) is a cardinal need.

The software package had to be able to accept at programming level more complex calculation processes as accounting of several operating parameters (function of other thermodynamic or operating parameters) as well as logical routines (***if*** *Condition A* ***then*** *Action B* ***)***. Whereas, *condition A* and *action B* respectively, have to easily be interpretable by other subsystems accordingly; thus, have to handle *numeric* inputs/outputs. The most important operating and thermodynamic (internal) data were going to be archived for post-run analyses.

Premises

In this section we give an overview of most important features of the software package and a few hints about its architecture. As a general remark, it has to be stated that the design process started "from scratch", that is, no preliminary work was taken as basis or imported as functional subunit. This fact was allowing us a very flexible approach and the possibility of choosing *any* solution we would foresee as optimal. In current subchapter only programming aspects are going to be discussed.

Thermodynamic calculations subsystem. We introduce this entity first, because it is the core of entire system. It is meant to summarize the set of routines performing thermodynamic calculations, i.e. is a *simulation tool* of the thermodynamic behavior of the machinery[3]. IPSEpro[4] was chosen. The evaluation based on discussions with previous users and the designers revealed this being a choice able to fulfill the demands enumerated in the previous section; this version was provided with *Component Object Model* (COM) extension, more than necessary for the assessing and exchange of the variables with external software.

Subsystem performing other calculations and data processing. IPSE is not designed for flexible mathematic, logic control as well as accounting and input/output management actions. These tasks had to be accomplished by "*auxiliary software modules*" (briefly *modules)*. In order to assure the modular construction and flexibility in development and maintenance, further dividing of the tasks is needed. Accordingly, all mathematic, accounting, logical and controlling duties are subject of a subsystem (*module*) each. The best choice turned out to be using *Dynamic Library Links* (DLLs). These are little programs in fact, providing practically unlimited possibility in implementing desired functions or more complex duties. Further on, the communication back and forth toward any other software is a routine task. Last but not least, a DLL can be written, maintained as well as upgraded *independently* of the *entire* system as long as the variables are synchronized. These units can be written in any software language. Thus, in case of development process in typical academic workshops, this means flexibility in having no pre-requisite knowledge in a *specific program language*. The authors chose to use FORTRAN compiler for *each* of the DLLs.

The "spine" of program group. The remaining multitude of tasks, mainly the start/stop, data management, archiving, hierarchy setting, user interface, etc… are done by the spine of the software package itself. Further on, it is shortly called as FRAME. Although a multitude of programming languages are able to fulfill these needs, Visual Basic (VB) had the advantage of simplicity, ease in implementing COM objects and it is proven as flexible.

Degradation typologies

In this paragraph, a brief description of degradation typologies is given. The aim is to introduce the models and the empirical relations used later on in the work. A detailed discussion, description of the methodologies and further references can be found in [32, 33]. Basically, the degradation phenomenon is devided into recoverable and non-recoverable degradation.

---

[2] We counted on having simulations of 10-20000 runs. While these would incorporate several iterations each.

[3] It is outlined that the tool was concepted for *gas turbine* analyses, but due to its architecture little modification enables analyses of *any* other system too.

[4] IPSEpro simulation software package, by SIMTECH ltd., version 2003.

Off-line washing is presumed as being regularly performed on site. The amount of cyclically eliminated performance derating represents the “recoverable” share of degradation. Consequently, the authors substitute this with the term “fouling”. Although, the two terms are not fully equivalent, it makes in our case little difference from both quantitative and qualitative point of view. As it regards the “non-recoverable” degradation (onwards “degradation”), we denominate with this term the surface roughness increase, erosion, clearance augmentation as well as other minor non-washable deposits and clogging within the machinery.

It is found that both fouling and degradation can be expressed as time dependant term evolving after logarithmic form:

$$r_f = \mathbf{1} - A\cdot\log(B\cdot\tau^C + \mathbf{1});$$

Whereas:

$r_f = \left.\frac{\partial P}{\partial \tau}\right|_{fouling}$ [-] the effect of time on power derating when only fouling is considered.

τ time, fired hours,

A, B and C are constants.

The more detailed analysis gave: A=0.006, B=0.8, C=1.3 for *fouling*. NB, the qualitative feature is in great extent possible to be generalized, while the constants (A, B, C) are machinery, site and operation dependant. Further research revealed the time dependency of parameter C. The empirical relation above can thus be written as follows:

$$r_f = 1 - A\cdot\log(B\cdot\tau^{C(\tau\_GG)} + 1);$$

whereas results that ***C*** is function of ***τ_GG*** , the number of fired hours since the gas generator was changed, that is its “age”

For *degradation*, the mathematic form of relative derating factor (r_g) is identical. ***A*** and ***B*** were kept constant, while the value for ***C*** was found *constant* and ***C=1.55***.

DETAILED PRESENTATION OF THE SYSTEM

In the current chapter, some of the main structural and operational details of the previously mentioned subsystems are presented. During the following descriptions, only technical issues are discussed. Accordingly to previous sections, the structure is: IPSEpro (thermodynamic calculus), DLLs (additional data processing) FRAME (control of entire simulation process, input/output management).

Simulation Tool – IPSEpro

A gas turbine model has to be created in IPSEpro environment. The basic knowledge being incorporated comprises of well-known basic aero-thermal theoretic considerations Traupel [29]. As a general approach, a dual fuelled two spool gas turbine (gas generator and power turbine) was chosen. We only outline herein the most relevant features of the model employed.

The system is in the greatest extent flexible regarding the machinery. The FRAME communicates with the model via addresses and this gives the opportunity to call, write and read independently from the actual hardware. If the model of a different name/path is called it is a simple name/path modification in the user interface necessary, see fig. 5. If the interface names of variables are intended to be e accomplished in either the FRAME or the IPSEpro model. Change in the physical background of the modules is easily done by rewriting and recompiling the DLLs.

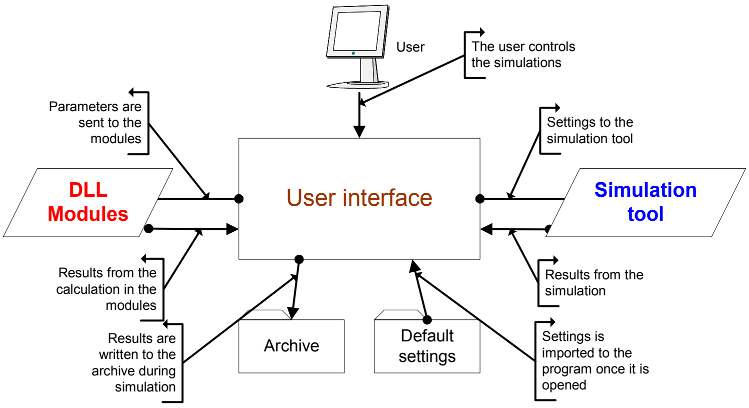


Figure 2., Block diagram of program package

The turbine is a two stage cooled HP expander and an un-cooled LP expander, the power turbine (PT). In order to have a better control upon calculations, cooling model, and pressure-flow equations the cooled stages[5] are modeled row by row. Thus, in fact, there are five thermodynamic and calculation entities in expander section: first stage vanes, first stage rotor, second stage vanes, second stage rotor and finally the power turbine. Pressure-flow-temperature correlation was set according to Stodola’s Cone Law. Cooling air is extracted in an intermediary compressor stage[6] and after the last compressor stage, respectively.

The compressor is divided in two spools as a direct consequence of cooling air being extracted after an intermediary stage. The pressure-flow-speed calculations use numerical relations based on a digitalized compressor map obtained from the manufacturer. The results were extensively validated for *new and clean* state against part-load performance calculations using the vendor’s data.

Controlling routines. The most challenging family of tasks was implementing the controlling routines. A multitude of interaction possibilities with the machinery was to be provided when planning analysis of gas turbines in most different off-design operational conditions. The main factors causing off-design operation of real machineries are: load, turbine inlet temperature (TIT), variable guide vane (VGV) angle failure, pressure drop increase in inlet/outlet, variable ambient temperature and pressure, variation of PT speed as well as of other intrinsic performance factoring parameters, like relative efficiency or flow factor. Progressive degradation and fouling is to be considered by modifying the efficiency and flow factoring terms provided. One can read more details about correlations between fouling process and the above mentioned parameters in [32]

General comments. A certain number of parameters (mainly those described in the *Controlling Routines* section above) are fully accessible in read/write mode. With the help of COM these are also available for external use, thus, can be set by the user via FRAME to any preset or calculated value (e.g. to the output of a calculation that is done by a *module*), see Fig. 2-3. This means that each of the internal parameters as well as the ambient parameters related to thermodynamic modeling is possible to be

---

[5] The first two stages of the gas turbine. The GG stages.

[6] Determined according to manufacturers information.

extracted or overwritten by main program (FRAME) and consequently used by *modules* (DLL) if needed.

The *Modules* – the DLLs

These modules (DLLs) have a multitude of duties. We assigned a single task for each of these. Basically, there are several classes or types of such modules, as follows.

Input generation for thermodynamic calculations This means, in fact, that these modules are primarily standing for generating input datasets preliminary to the very oncoming calculation. The only entering variable is *calendar time* (step), whereas outputs are current ambient conditions, desired load level, fuel quality, wet/dry control. By demand, extension is very easily possible in the future by simply implementing new DLL modules.

Accounting modules The next branch of tasks assigned to DLLs is accounting type of jobs. The authors suggest the Equivalent Outage Hours (EOuH) criteria (see Eq. (1)) as measure for the loss as direct consequence of a change in operating conditions or maintenance policy. EOuH criteria is extensively used by the authors in assessing the impact of different operating policies on the availability of the machine. Further on, it is of cardinal importance for the appraisal of remaining life, as well as for assessing the actuality of any maintenance action to accurately keep track of Equivalent Operating Hours (EOpH) criteria. This relatively complex accounting is based on very simple arithmetic calculations as previously described (see corresponding section before).

Modules performing logical decisions DLL modules are entrusted other very important duty type, namely to *interpret* results. As pattern, these are logical "***if*** *Condition A* ***then*** *Action B*" kind of tasks. Input data are originating from other modules and/or directly from thermodynamic modeling. Output is always numerical.

Typical example for a module performing logical tasks is the one responsible for crank wash stops. The authors refer to a well-known fact that gas turbines are subject to fouling. This negative effect is generally eliminated by *washing* the compressor part. This action causes a downtime, (outage, see Fig. 1.) as well. Consequently, such a module has to perform a multitude of steps in order to simulate the multiple events related to this kind of set of actions. Hereafter the authors intend to shortly present the logical steps as incorporated in FRAME.

We refer to knowledge and notations introduced in chapter *Background.* Additional notations are:

- $D_f(\tau)$ momentary derating caused by fouling, (1)
- lim_w the allowed amount of power derating due to fouling before a wash is scheduled, (Eq. 2). Its value is usually predefined by operator, e.g. 5%.
- τ_{wb} time of beginning of crank wash. Moment, when the machine is stopped for washing.
- $\Delta\tau_w$ duration of performing one crank wash, ca. 8-10 hours depending on machinery, site and previous operation.
- τ the current time, independent variable.

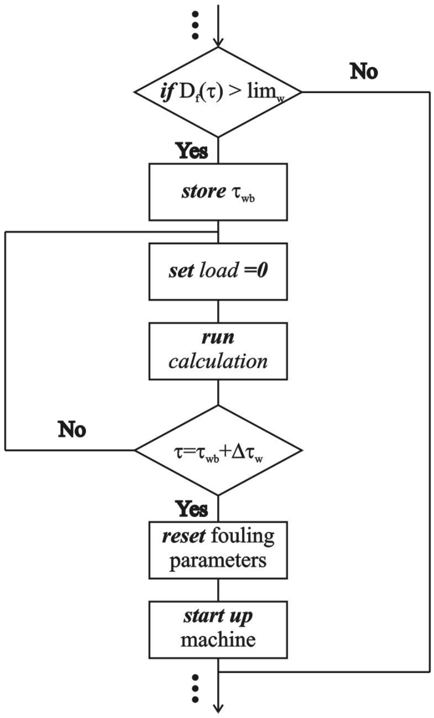


Figure 3., the logic flow sheet diagram for module simulating wash stop

The *module* monitors the momentary value of *corrected power* [32] that is, after each run of the thermodynamic calculation. The value of momentary derating due to fouling $D_f(\tau)$ is then calculated. The logic check is performed, see Fig. 3., $D_f(\tau) > lim_w$., i.e. is the derating exceeding the predefined limit lim_w.or not? If *no,* subsequent runs follow and no action is taken. If *yes*, ***stop*** is necessary and a downtime of $\Delta\tau_w$ hours follows. As soon as, the *calendar time*, as independent variable reaches $\tau = \tau_{wb} + \Delta\tau_w$ the machine returns to its previous regime; a start as well as run in *clean state* is modeled, meaning also that factors responsible for accounting the fouling are reset to null. These are efficiency and flow reducing terms employed in compressor model.

Other auxiliary modules Finally, we can simulate failures as well. These "accidental" failures are prescribed in concordance with field data: that is, the frequency, consequences, as well as the duration of the corrective maintenance action is based on real measurements. Examples are: fouling, non-recoverable degradation, inlet filter clogging, VGV angle failure, decrease efficiency of compressor, turbine and combustion chamber.

Failures, causing immediate stop/trip are easily modeled if data is available. These can also be useful for certain assumed parameter variation purposes, i.e. different failure frequency or failure mode scenarios. The authors did not implement these latter features, yet.

More detailed description about the modules operation is to be found in [33] as well as in the subsequent chapters.

User interface – the FRAME itself

Description. This is the only executable entity; the *master* tool. The FRAME itself is considered the spine of the system. It is designed to perform a series of callings of different procedures. These could be data/file handling routines, DLLs (as previously described), archiving and last but not least communication from/to user (read data from keyboard and write data on the screen and memory).

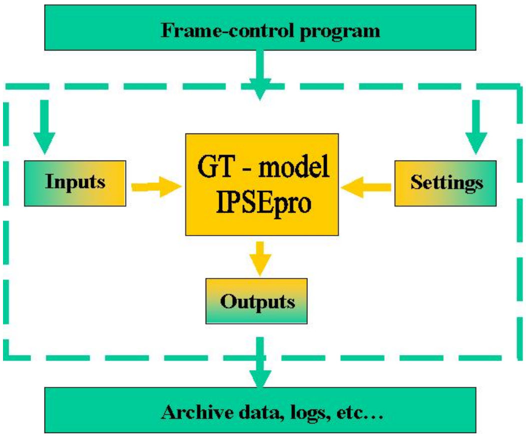
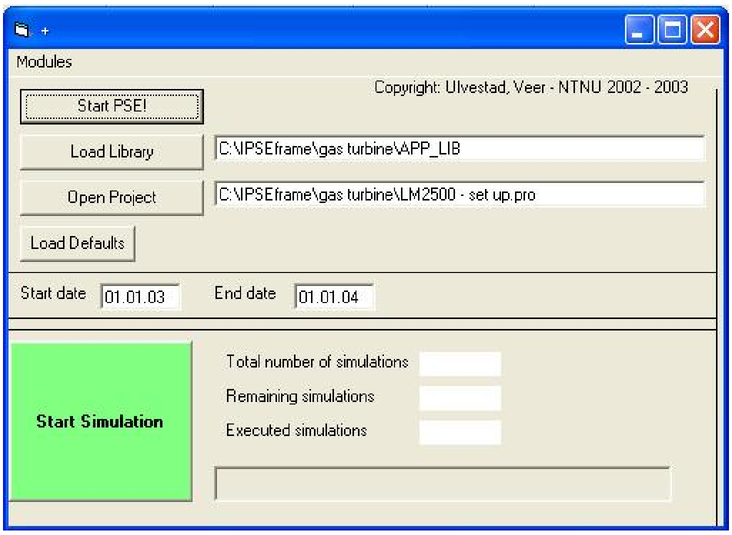


Figure 4., the functionality scheme of the software package - FRAME

Visual Basic programming language was chosen for this purpose. The program model consists of several visual forms with respective separate codes. The forms communicate with each other and update the running initiation database when enabled or closed by the user. The main form is the window the user can see when opening the program. There is communication of values between the forms.

The authors' main concern was to facilitate parameter variation kind of analyses. At start of FRAME, the last runs setting are automatically loaded from an initiation file. This way, the user has a minimal duty, mainly to *modify* and not to fill in unnecessary repetitive information. This means that the user is not requested to give inputs before starting a new simulation process, but only modify the changing parameter if desired. So, typing errors typically occurring by running series of a parameter variation research are minimized.

File information, start/stop dates, model paths as well as start of the system are part of the main form. All other information are hidden from the main form, see Figure 5. These can easily be accessed though, via the menu system, ***Modules***, (upper left corner). The modification opportunities of the modules' input are made in separate sub forms.



Copyright: Ulvestad, Veer - NTNU 2002 - 2003

Load Library: C:\IPSEframe\gas turbine\APP_LIB

Open Project: C:\IPSEframe\gas turbine\LM2500 - set up.pro

Start date: 01.01.03

End date: 01.01.04

Figure 5., main / starting window, FRAME

The authors designed the system so that most possible settings or operating parameters can be accessed directly from FRAME menus or forms. As it is now, the following modules are ready made and tested: ambient temperature, ambient pressure, load variation, fuel type, fouling, washing and degradation. Turning modules on or off is also possible directly from the forms of the spine program that makes testing isolated influences, i.e. modules one by one. Tha is, ticking on only one of modules will enable testing the influence of only this one factor upon the operation.

Archiving of both the settings of the FRAME as well as the results of the simulation is provided. The files are plain text (ASCII) files. Data is being separated by semi-column. We chose the simplest format, in order to avoid any incompatibility.

SAMPLE OF CALCULATION RESULTS, FINDINGS

Preliminaries

The main goal was to develop a tool that would enable the user a time-marching of a given system, in this case a gas turbine. The core thermodynamic calculations would be repeatedly performed while an intelligent frame is governing the input/output management, decisions as well as data processing is simultaneously ran and data is continuously archived for more detailed post-run analysis.

Inputs. Data is provided by offshore gas turbine operators. Knowledge, reflecting realistic operational behavior of a machine is implemented as series of empirical relations in the *modules* (DLLs), as previously described. Ambient conditions are provided by empirical data or prognoses. Based on these, the authors performed relatively realistic simulations, and useful results should be produced.

Expected results. A better understanding of how fouling, degradation and off-line washing of compressors affects EOuH is desired. Focus is on long term analysis. The authors look into prediction of Equivalent Operating Hours (EOpH) with different operating conditions. A brief sample of results is presented as follows.

An Example of Simulation Algorithm

The authors will now demonstrate the usage and the course of the program. A simulation run will be explained.

Initiate When FRAME is initiated a program window with a few buttons will show, see Fig. 5. Three choices are available at this point:

1. Modify modules
2. Start PSE
3. Set dates

These choices can be taken in random order. As follows, the example is shown for running simulations in order to find an optimal lim_w. for given operating conditions and machinery.

Setup before running

The approach was to keep all values constant and vary lim_w (parameter variation), in order to investigate the effect on the EOuH. Thus, we set the coefficients for fouling and degradation models, by opening the ***Module*** menu. We then decided on what lim_w to start on, and for how much outage each off-line wash would cause.

When the modules were modified the start PSE button was pushed. This starts the ***IPSEpro*** application with the specified library and project. The library and the project can be changed by entering the path in the respective windows.

The last thing that needs to be done is to set the dates for the simulations; we simulated for one year (01.01.03-01.01.04, see Fig 5.

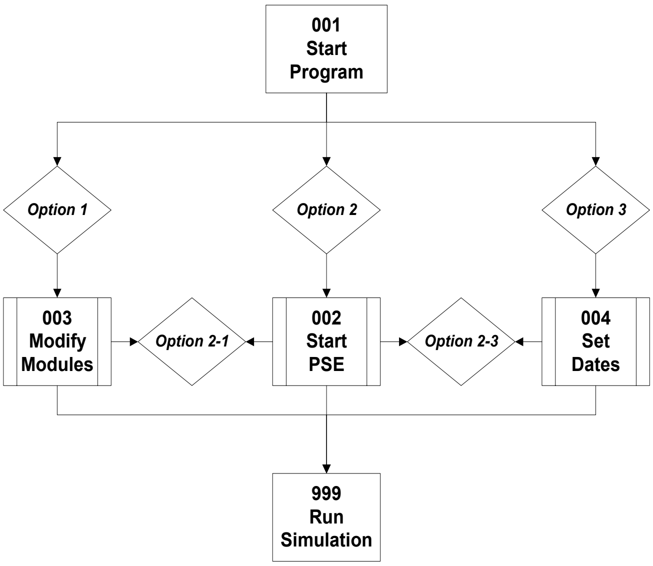


Figure 6., Logical scheme, overview of functions of FRAME

Simulation Now the "Start simulation" button can be pushed. This will initiate the simulation loop, which sends parameters to the DLLs, gets the results of the calculations, sends the appropriate settings to IPSEpro and writes the desired results to an archive.

Processing results When a simulation run is finished the desired output is gather from the archive and processed in the desired way.

Next set of simulation When starting the next run (i.e. the next value of lim_w) all we did was to open the module menu, change the value of lim_w and push the ***Start Simulation*** button in the main window, Fig. 5. It is advised to delete the archive before the nest run. The results of these runs are presented in the oncoming chapters.

Power loss due to fouling.

The EOuH is suggested as a measure for power loss due to fouling, degradation and outage (wash stop). We also define lim_w (Eq. (2) as a limit of power loss due to fouling before washing is performed. Firstly, the optimal value of lim_w It is sought with respect to EOuH.

Primarily, we investigate how EOuH varies for different lim_w with time span as independent parameter, see Fig. 7.

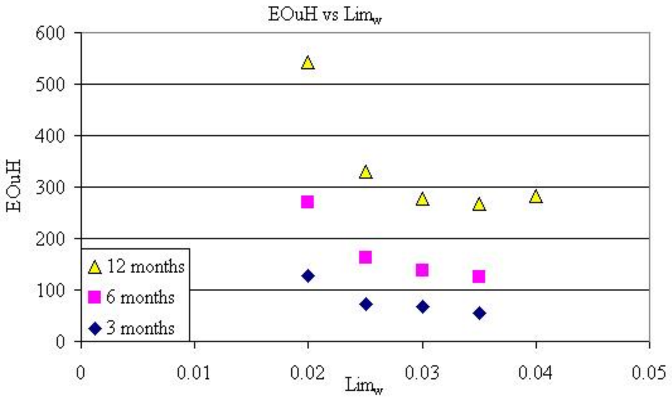


Figure 7., EOuH as a function of power loss before washing limit over three periods.

The abscissa is lim_w . Simulations were carried out for 3, 6 and 12 months period keeping constant lim_w (values of 0.02-0.04 are considered). Lower values mean more frequent wash. In the case of the 3 and 6 months time span, the limit of 0.04 meant *no washing*. In 3 and 6 month cases this last point is also where the least amount of EOuH is, and therefore no washing is the optimal strategy. and this is interesting and helpful information when planning for off-line compressor washing. When investigating the 12 month period one sees that the optimal lim_w occurs at a point before the point of no washing, ca. 0.033, i.e. 3.3% percent power output derating.

Power loss due to fouling and degradation.

The results presented and described in the previous paragraph are found by neglecting *non-recoverable* losses occurring due to degradation. To complete the picture of expectable EOuH variation with time for different washing strategies, we consider power loss due to general degradation, as well. The model for power loss due to degradation is more in detail described in [32]. The objective in this simulation was to investigate the combined effect of fouling and degradation on EOuH for the same time spans as previous. We investigate the 12 months period closer, see Figure 8.

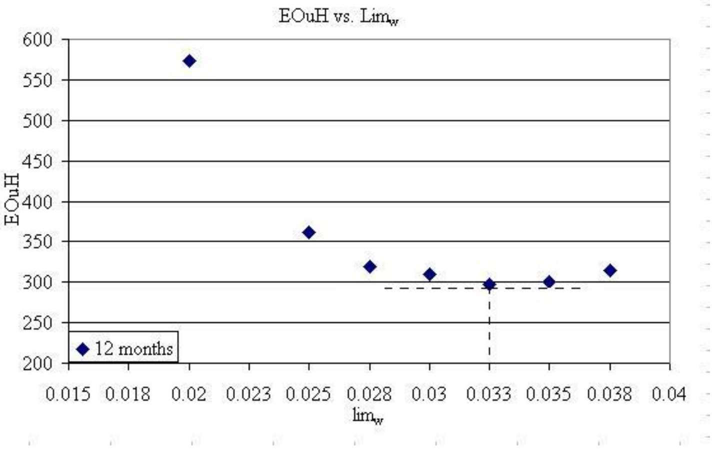


Figure 8., Equivalent Outage Hours (EOuH) versus limit of power loss before washing (Lim_w) for 12 months.

The curves present a similar pattern to the previous simulations. Although, for the same washing frequency EOuH values are higher due to imminently higher power loss due to the additional derating caused by degradation. The minimal value for EOuH versus time has the same value as when degradation was not considered. This has to be interpreted as consequence of the fact that degradation has relatively weaker effect on EOuH compared to fouling.

Age Dependency

It is common practice in maintenance planning to decrease the washing frequency as operation time increases. The frequency is known to decrease with a factor of two each year. This is not necessarily the optimal frequency. The objective in this simulation was to use FRAME to investigate how the optimal lim_w would modify, as age of machinery changes. Age dependency of development of fouling is found [32].

Fouling models are implemented as time dependant for three different ages, 7000, 10000 and 16000 operating hours[7], see Figure 9. As previously, the abscissa is ***lim***$_w$.

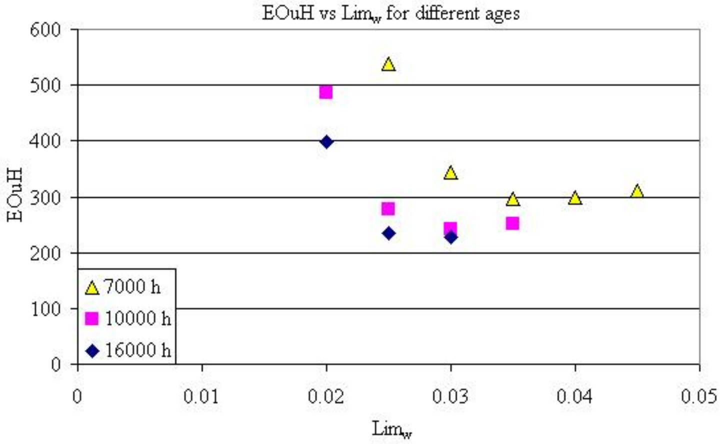


Figure 9., Equivalent Outage Hours (EOuH) vs. limit for different ages of the gas generator.

Discussion of results: EOuH versus time

It is shown that the limit value of ***lim***$_w$ is very much of an age dependant. On the other hand, it turned out to be relatively un-sensible to the fact, whether degradation is additionally considered or not. This finding facilitates calculation and prediction of the moment of the oncoming washing by possibly neglecting effect of degradation.

It is noted that the EOuH versus ***lim***$_w$ curve around the optimal limit point is quite flat, i.e. a relative low sensibility is found around the optimum. Nevertheless, additional parameter variation done by the authors reveal that this curve is becoming steeper when exceeding 2% of power loss compared to the optimum, ***lim***$_w$=0.033 in our case. This makes sense, since the consequence of moving from the optimal limit will have more effect for longer periods.

A limit of 3 % power loss due to fouling is a well known in the industry as power derating limit before crank wash. A thorough overview of state of the art is given in [27]. Our calculations give the same value for a ca. 1 year old gas generator, 7000 operating hours at the beginning of fouling cycle. This is a useful hint concerning applicability of the tool presented herein. It is expected that optimizing of washing intervals with respect to EOuH, hence cost, are done.

Easily, other object functions and hence cost assessment is definable, e.g. in a scenario, where power has to be produced, even though washing ideally should take place. Further on, the operator can be interested in estimating how much outage (revenue loss) will result as direct consequence of a performed maintenance action. Together with adequate risk assessment tools, FRAME facilitates advanced and accurate reliability and availability analysis too. Great variety of optional operation and maintenance strategies can be appraised by answering questions like “what is the consequence of scenario A upon RAM[8] parameter X”.

Accounting EOpH

It is of special importance to monitor and assess/predict the remaining life of a system. The EOpH is defined in literature [4, 11, 21] as a function of multiplicative and additive factors weighted by machinery dependant coefficients. The authors ran tests of how successfully FRAME responds to modified operating and/or ambient conditions with respect to EOpH (as described in Eq.(3),) Start ups, load levels (TIT[9]), type of fuel and wet control for emissions are considered. The model was implemented by writing a simple code for calculating EOpH, compiling it as a Dynamic Link Library (DLL) and attaching it to the FRAME. The tests were done as parameter variation, and the applicability for FRAME for this kind of assessments is fully demonstrated. However, since the authors did not have the opportunity to validate results against field operation data, we would avoid publishing these, as for now.

Further on, the tests also pointed out the flexibility of the tool as it concerns implementing modules that perform rather complex parameter accounting. Thus, the authors foresee great potential in extending and refining the facilities of software package described herein by rewriting or adding a new module as DLL.

CONCLUSIONS AND FUTURE WORK

The authors presented an overview of a newly developed tool for assessing the effect of different operating and maintenance scenarios on performance as well as reliability, availability performance of gas turbines. Main concern of the authors was to present the structure of the tool, describe the functionality of the employed computational and programming techniques as well as logical inter-relation between subsystems. Then, the operation of the governing algorithm of a typical simulation run is explained.

In the second part of the work, typical results are presented in order to illustrate the applicability of FRAME. As it concerns the technical aspect of the results, only effect of fouling and degradation on the equivalent operating hours is detailed due to space limitations. A short discussion follows the results.

Based on the result samples and the auxiliary test runs performed by the authors it can be stated that FRAME fulfills the requirements of enabling accurate, flexible and useful performance, availability and reliability analysis. It is outlined the wide potential of adding or refining the know-how incorporated by simple DLL programming or re-programming tasks.

The authors plan to extend the set of modules available for the user. Additionally, setting up new object functions and testing in cooperation with the industrial partner operating gas turbines offshore is expected

ACKNOWLEDGMENTS

Norsk Hydro, Norway as well as Statoil, Norway are deeply acknowledged for supporting this work.

APPENDIX

Description / definition of Factors – EOpH

In this project it is used a selection of influent factors, based on literature and knowledge.

---

[7] These amounts of operating hours correspond the period length that the gas generator was operating from its change till the *last washing*, i.e. the beginning of the monitored period.

[8] RAM stands for reliability, availability and maintainability

[9] Turbine inlet temperature

- $f_F = k_1 \cdot G + k_2 \cdot (1 - G)$
- $f_C = k_3 \cdot W.C. + k_4 \cdot (1 - W.C.)$
- $f_S = a_s \cdot n_s$
- $f_A = a_{lch} \cdot n_{lch}$
- $f_L = c$
- $f_P = \sum_i b_i \cdot t_i$

G = % of time/100 where gas is used as fuel.
k_1 = Weighting factor for natural gas as fuel (1.0)
1-G = D = % of time/100 where distillate is used
k_2 = Weighing factor for distillate used as fuel (1.1 – 1.6)
W.C. = % of time/100 where wet control is used.
k_3 = Weighting factor for wet control
1- W.C. = D.C. = % of time/100 dry wet control is used.
k_4 = Weighting factor for dry control
n_s = Number of starts
a_s = Weighting factor for a start (6 – 10)
n_{lch} = Number of load changes
a_{lch} = Weighting factor for load change (2 – 6)
c = location factor (1.0 in this project)
t_i = Different load levels (60%, 70%, 80%, 90%, 100%)
b_i = Different weighing factors for different load level (e.g.: 0.9, 1.3, 2.0, 4.0, 6.0)

The concrete values of these terms are suggested bases literature [7,11] and oral communication with the operators. The figures would differ from hardware to hardware, but a good qualitative overview of the processes is enabled by using these above.

REFERENCES

[1] Anuskiewicz R. J., 1994, An Operation and Maintenance Strategy to Maximize Performance of a CC Power Plant, ASME TurboExpo Proceedings, The Hague, 94-GT-121.

[2] Bowler D. J., 1994, Reliability-Centered Maintenance: a Methodology for Economic Justification, ASME TurboExpo Proceedings, 94-JPGC-PWR-22.

[3] Bruns D. P., 1990, The influence of Maintenance on Gas Turbine Reliability-Availability-Maintainability Performance, PowerGen '90, Proceedings, pp. 1185.

[4] BS4778-3.2, 1990, Quality Vocabulary – Availability, Reliability and Maintainability Terms, British Standardization Inst..

[5] Byington C. S. et al., 2002, Prognostic Enhancements to Diagnostic Systems for Improved Condition-Based Maintenance, IEEE Proceedings Journal.

[6] Ceschini G. F., 2002, Gas Turbine Maintenance Policy: a Statistical Methodology to Prove Interdependency between Number of Starts and Running Hours, ASME TurboExpo 2002 Proceedings, 02-GT-30281.

[7] Cooke, Lloyd A., 2001, Component management programs for reliable life extension, ASME Turbo Expo Conference Proceedings, 01 – GT - 0431.

[8] Corr J., 1991, The Maintenance Experience of Frame 9 Gas Turbine Operating in Combined Cycle, PowerGen '91, Proceedings pp. 1517.

[9] Ekström T., 1995, Reliability/Availability Guarantees of Gas Turbine and Combined Cycle Generating Units, IEEE Transactions on Industry Applications, Vol. 31.

[10] Hale P., 2001, Survey of Reliability and Availability Information for Power Distribution, Power Generation and HVAC Components, IEEE Transactions on Industry, Vol. 37.

[11] Hoeft R., 2000, Heavy-Duty Gas Turbine Operating and Maintenance Considerations, GE Power Systems Reports, GER-3620G, GE Energy Services, Atlanta, GA.

[12] IEEE 1987, Standard Definitions for Use in Reporting Electric Generating Unit Reliability, Availability and Productivity, ANSI/IEEE 762/87.,

[13] ISO 14224, 1999, Collection and Exchange of Reliability and Maintenance Data for Equipment.

[14] ISO 3977-9, International Standard, 1999

[15] Khan, Iftikhar et al., 2003, Technical, financial and engineering management of CCGT overhauls, Proceedings of ASME Turbo Expo 2003, Power for Land, Sea, and Air, June 16-19, Atlanta, Georgia, USA, GT2003-38874

[16] Koob H., 1993, ABB service focus on trouble-free gas turbine and CC operation, ABB Reviews 10/93, Mannheim,

[17] Krockow W., 1996, Predicting Gas Turbine Reliability to Assess Risks for Purchased Power Agreements, ASME Conference Proceedings, Birmingham, UK, 96-GT-266.

[18] Lampert D., 1990, The Effect of the Structure of Combined Gas/Steam Turbine Plants upon their Availability, BBC Review, CH- T 040 143.

[19] Müller E., 1992, 10 Jahre Betriebserfahrungen mit einer Gas-/Dampfturbinen-Heizkraftanlage, VGB 72. Heft 8,

[20] NERC, 2002The Generating Availability Data System, Generating Availability Reports, www.nerc.com

[21] OREDA 92, 1992, Offshore Reliability Data Handbook, Det Norske Veritas, Hovik, Norway.

[22] Perkavec, 1992, Uberwachung der Leistungsparameter der Gasturbine in Abghängigkeit vom Verschmutzungsgrad, VGB KRaftwerkstechnik, Juni.

[23] Pinelli M., Spina, P. R., 2002, Gas Turbine Field Performance Determination: Sources of Uncertainties", ASME J. for Eng. for Gas Turbines and Power, **124**, pp. 155-160.

[24] Ricketts R., 1994, Organize to Manage Reliability, The Journal of Hydrocarbon Processing, **12/1994**.

[25] Roemer M. J., 2000, Advanced Diagnostic and Prognostic Technologies for Gas Turbine Engine Risk Assessment, ASME Turbo Expo 2000 Proceedings, 00-GT-30

[26] Sampath, Singh, 2003, Managing Maintenance and Operational Availability of Gas Turbine Propelled Naval Ships Trough Advanced Engine Fault Diagnosis, ASME Turbo Expo 2003, Atlanta, 03-GT-38456

[27] Schepers, 1999, Optimierung der On-line Wäsche an einer 26MW-Gasturbine unter besonderer Berucksichtigung der Leistungssteigerung, VGB Kraftwerkstechnik, March, 1999.

[28] Silva et al., 2003, Optimization of Operating Conditions and Compressor Cleaning Time Intervals of Combined Cycles in a Liberalized Market, ASME Turbo Expo 2003, Atlanta, 03-GT-38455

[29] Traupel, 1985, Thermische Turbomachinen, Textbook from Springer Verlag Berlin, IIIrd Edition, 1985

[30] Ulbrich, Arthur, 2003, High steam turbine operating flexibility coupled with service interval optimization, International Joint Power Generation Conference Proceedings, Atlanta, 2003, IJPGC2003-40072

[31] Ulvestad, 2003, Advanced Software Tools for Operational Analysis of Gas Turbines, Semester work, submitted to Norwegian University of Science and Technology, Trondheim, November 2003.
[32] Veer, Bolland, 2004, Measured data correction for improved fouling and degradation analysis of offshore gas turbines, ASME Turbo Expo 2004, Vienna, (under publication) 04-GT-53760
[33] Veer, Bolland, 2003, Using Probabilistics and Advanced Software Tools for Reliability and Availability Assessment and Lifing, ASME TurboExpo 2003, Atlanta, 03-GT-38474
[34] Yau J., 2002, An Empirical Model for Predicting Crack Growth Behaviour of Gas Turbine Stage 1st Nozzles, ASME TurboExpo 2002 Proceedings, 02-GT-30302
[35] Z-016, 1998, Regularity Management and Reliability Standard, Norsok, 1998.
[36] Zirkelback Ch. E., 1990, Cogeneration Plant Machinery Design for Maintenance and Reliability, ASME TurboExpo 1990, Proceedings Brussels, Belgium, 1990, 90-GT-104