Proceedings of ASME Turbo Expo 2011
GT2011
June 6-10, 2011, Vancouver, Canada

GT2011-45298

DISTRIBUTED NETWORK SYSTEM FOR REAL-TIME MODEL BASED CONTROL OF INDUSTRIAL GAS TURBINES

V. Panov
vili.panov@siemens.com

Siemens Industrial Turbomachinery Ltd, PO Box 1, Waterside South, Lincoln, LN5 7FD, UK

ABSTRACT

This paper describes the development of a distributed network system for real-time model based control of industrial gas turbine engines. Distributed control systems contribute toward improvements in performance, testability, control system maintainability and overall life-cycle cost. The goal of this programme was to offer a modular platform for improved model based control system. Hence, another important aspect of this programme was real-time implementation of non-linear aero-thermal gas turbine models on a dedicated hardware platform. Two typical applications of real-time engine models, namely hardware-in-the-loop simulations and on-line co-simulations, have been considered in this programme. Hardware-in-the-loop platform has been proposed as a transitional architecture, which should lead towards a fully distributed on-line model based control system. Distributed control system architecture offers the possibility of integrating a real-time on-line engine model embedded within a dedicated hardware platform. Real-time executing models use engine operating conditions to generate expected values for measured and non-measured engine parameters. These virtual measurements can be used for the development of model based control methods, which can contribute towards improvements in engine stability, performance and life management. As an illustration of model based control concept, the example of gas turbine transient over-temperature protection is presented in this study.

ABBREVIATIONS

CIF	Communication Interface
CPU	Central Processing Unit
DCS	Distributed Control System
GT	Gas Turbine
H-i-L	Hardware in the Loop
HMI	Human Machine Interface
I/O	Input/Output
MBT	Model Based Temperature
MGT	Measured Gas Temperature
PCS	Process Control System
PLC	Programmable Logic Controller
PROFIBUS	Process Field Bus
PROFIBUS DP	Decentralized Peripherals
PROFIBUS PA	Process Automation
RTOS	Real-Time Operating System
TET	Turbine Entry Temperature
Top	Turbine Operating Temperature
UCP	Unit Control Panel

INTRODUCTION

Distributed network architecture has already been successfully implemented into the control system of industrial gas turbines. This architecture was chosen primarily because of simpler integration with large plant distributed control systems (DCS). Benefits of implementing distributed architecture into a gas turbine control system are multiple [1-4].

Maybe the most obvious benefit of distributed network architecture is the potential to reduce wiring within the turbine package, and hence to contribute towards reduction of weight and cost of product. Modular design supports addition of new software and hardware functionality even after the system has been commissioned. This system flexibility allows extension of system capability and improvement of system maintainability. Composability is another attribute of distributed control systems, which allows different modules to be used across different engine platforms, and therefore to contribute towards reduction of system testing.

The aim of this project was to exploit the benefits of existing gas turbine DCS and to offer a platform for more efficient control system design. High fidelity non-linear gas turbine models were developed and implemented onto a dedicated hardware platform. It has been proposed to integrate these real-time engine models into distributed gas turbine control system, thus offering a modular platform for model based control system.

To demonstrate the proposed concept and to reduce development time of fully distributed on-line system, H-i-L (hardware-in-the-loop) technology was proposed as a development facility. Using this approach, testing of the development platform, which can be considered as an additional I/O's node in distributed network system, can be performed in isolation prior to integration. Hardware-in-the-loop technology is an ideal test facility where simulations and hardware components can be integrated for development of model based control methods and software validation [5].

DISTRIBUTED CONTROL SYSTEM

The control system used in this study is based on SIMATIC S7-400 PLC hardware. Networked I/O's in this DCS are used to reduce system cabling between the turbine package (gas turbine & driven unit) and the unit control panel (UCP).

The UCP contains the PLC and communications systems. Through the communications systems all other systems are interfaced either directly (ET200M modular I/0 station) or via protocol converters, thereby allowing integration of specific systems on different engine platforms (Fig.1.).

The PLC consists of the central processing unit (CPU) and two Profibus interfaces, which are used to integrate on-package (local to the turbine) and off-package systems (remote from the turbine). These two Profibus networks (DP1 and DP2) are utilised to collect gas turbine and its associated auxiliary systems I/O's.

The turbine and driven unit package are built with all sensors and actuators wired to junction boxes on the end of the package. Some of the equipment is connected directly into distributed I/O in the junction boxes, whereas others have to be wired back to the UCP due to their zoning or power requirements.

The PLC connects to an operator station, i.e. human\machine interface (HMI) via an Ethernet link, which can be used for accommodation of additional stations and protocol converters. The Ethernet link can also be extended to customer DCS, where remote monitoring of the system is possible via a broadband internet connection.

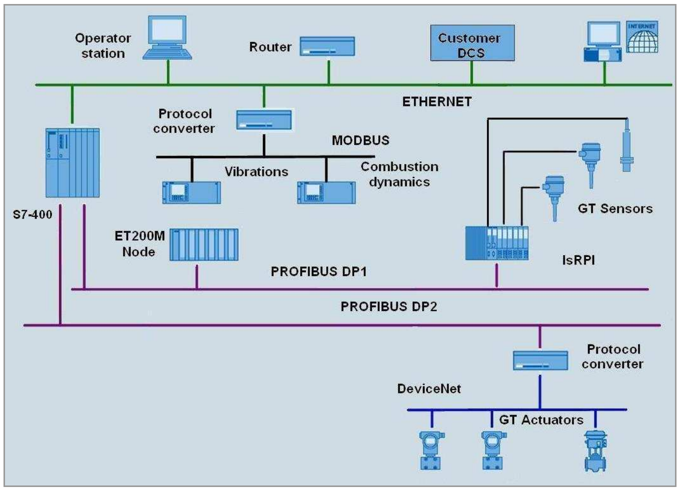

Fig. 1. Gas turbine DCS

SIMATIC PCS7 is an engineering tool for working with distributed control system. PCS7 software suite is used for creating the control program, configuring hardware of the system and creating the user interface. This tool allows a structured approach to creating and managing software, which is downloaded to PLC & HMI, and also secures easy configuration of distributed system hardware.

REAL-TIME ENGINE SIMULATIONS

Gas turbine models were developed using a Simulink specialized block-set library called GasTurboLib [6]. Different engine configurations can be generated using this generic simulation tool and these models can be used for real-time simulations.

Using Simulink add-on tool xPC Target, it is possible to deploy real-time models using standard PC hardware. It is an environment that uses a target PC, separate from a host PC, for running real-time applications. This tool is used to automatically generate and compile a C/C++ code representation of Simulink models. The target application is then downloaded via an Ethernet connection from the host computer to the target computer, where Simulink models can run in standalone mode using real-time xPC kernel.

Real-time operating system

Real-time xPC operating system is a computing platform that ensures a timely execution of all scheduled tasks for the Simulink application. Because a real-time application must be ready to process a task when the moment arrives for the task to begin execution, only fixed-step solvers are supported. In this way the execution time of a task is prohibited from exceeding the sample rate period, i.e. overrun. An overrun postpones the start of the next scheduled task, so an accumulation of such occurrences may eventually overload the processor.

The Simulink environment employs an “auto-solver” facility, which during the simulation run provides integration of differential equations using numerical method chosen by user. A fourth order Runge–Kutta explicit one-step integration algorithm has been used for implemented gas turbine real-time simulations. A fixed integration step time of 1 ms has been used to secure numerical stability of solution at every time step. It has been concluded that engine simulations executing with an integration step size of 1 ms are sufficient to accurately calculate the critical transient effects during typical engine acceleration / deceleration maneuvers [6].

Further optimization of real-time models has been achieved by carefully selecting appropriate computational schemes. In real-time environment computational tasks could be grouped and implemented as single-task and multi-task scheduling schemes. The single-task approach formulates a single thread of computations that runs the entire model at the base rate and conditionally updates sub-rate blocks. The multi-tasking approach formulates multiple threads of computations with each tread running blocks of a specific rate group. Implemented real-time simulations execute with typical task rates ranging from 100 ms to 1 ms at the most demanding.

Interfacing

Real-time models executing on dedicated hardware platform have been integrated into gas turbine control system via communication interface. Real-time hardware platform must be easily reconfigured for different engine configurations and customized for different applications (H-i-L or On-line). Connecting at a communications level rather than a physical I/O level offers this flexibility.

Documented methods of communication interfacing include different memory-mapped techniques, such as duel-port memory and reflective memory. Those memory-mapped techniques have been used for development of various data bus communication protocols such as CAN bus, DeviceNet, Profibus, Ethernet etc.

There is a range of commercial off-the-shelf communication interface (CIF) I/O boards, which are developed for PC-compatible hardware. To communicate with the control system, the real-time application must perform a sequence of read and write operations with the help of driver routines that are specific to the installed communication interface board.

Using communication interface PC card for fieldbus communication, the dual port memory on the card serves as a process image, which can be accessed from both sides at the same time. Simulation application need not concern itself with the actual fieldbus communication, because this task is automatically undertaken by the CIF card thanks to its own communication processor. The communication interface handles the complete data exchange between the xPC platform and connected fieldbus devices.

Hence, generating engine models with customized communication driver library, and deploying these Simulink models onto PC with CIF hardware, real-time simulation platform can be easily integrated into distributed network system.

Real-time network communication

The fundamental prerequisite of distributed control systems is the real-time communication. The communication network must not contribute to control instability and hence network data throughput must be deterministic [7]. The most important task to consider when distributing control over a communication network is synchronization of the I/O to the control application software. Typically 100 ms update rates are used for slow control loops, and 10 ms update rates for more demanding control loops.

Within industrial communication systems, fieldbus networks are especially developed for the interconnection of process controllers, sensors and actuators. In this study Profibus protocol has been used to support real-time communications between xPC and PLC platform. Profibus is deterministic time-triggered communication protocol, and it is based on a simplified Time Token protocol [8]. This is a well-proved solution for real-time communication systems with several thousands of installations worldwide. Several studies on the ability of fieldbus networks to cope with real-time requirements have been carried out, and Profibus was recently considered as one of the general-purpose solutions for the fieldbus communication systems by European standard.

The main function of the Profibus is cyclical transmission of process data from the control system to the peripherals and in the reverse direction. Access is by the Master-Slave principle. A Master controls the assigned Slave device on the bus in polling operation. Data exchange is initiated by poling message and terminated by an acknowledgment message from the Slave addressed. Each Slave therefore only becomes active when required by the Master, and simultaneous bus access is thus avoided.

There are two variants of Profibus in use today, the most commonly used Profibus DP, and the lesser used, application specific Profibus PA. Profibus DP (Decentralized Peripherals) is used to operate sensors and actuators via a centralized controller in plant automation applications. Profibus PA (Process Automation) is used to monitor measuring equipment via a process control system in process automation applications.

SYSTEM CONFIGURATION

Distributed control system under consideration in this study is based on SIMATC S7-400 process controller. This control system allows the uniform configuration of the communication relationship between Master and Slave fieldbus devices. PLC in this system is configured as a mono Master device in Profibus DP network.

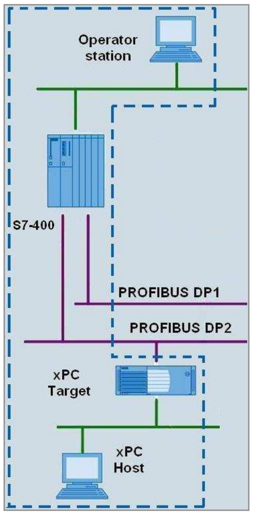

Fig. 2. H-i-L configuration

The configuration and setup of xPC platform takes place through the system configuration tool. This configuration tool is located on the remote PC and communicates with the CIF device driver via Ethernet link. The configuration is stored on the CIF and therefore the CIF is immediately ready after the start of system.

In H-i-L configuration (Fig.2.), network is configured as single Master – single Slave Profibus network. In On-line configuration (Fig.3.), PLC is configured as a single Master device, where xPC platform is just one of the I/O Slave devices.

In the H-i-L configuration, just as in normal process, operator station is connected with Ethernet to the PLC, into which the control program is loaded. In the gas turbine setup there are DP-Slaves connected via Profibus to the PLC. The DP-Slaves handle I/O’s, connected to measurement points in the gas turbine, managing the communication to the control program in the PLC. In the H-i-L configuration, the CIF interface card, connected via Profibus to the PLC, is simulating I/O’s. The CIF interface sends and receives signals to and from the simulator program and forwards them to the control program, which receives them and acts as if they come from the real gas turbine. In On-line configuration xPC platform is connected as an additional I/O slave device on Profibus network.

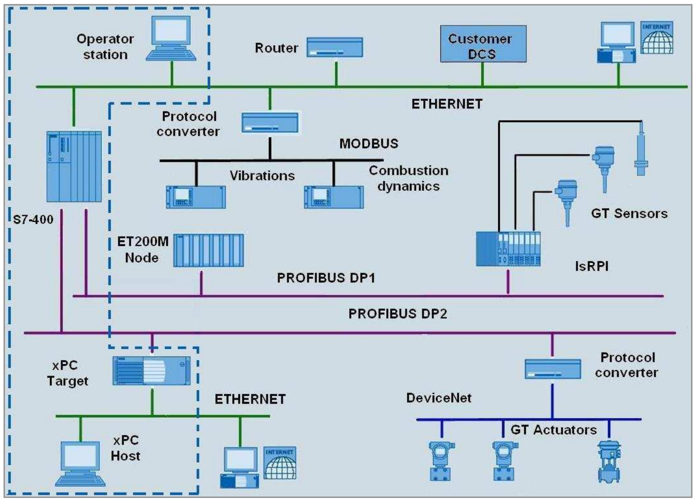

Fig. 3. On-line configuration

GAS TURBINE DYNAMIC MODELLING

The integration of an on-line engine model into the control system enables the use of virtual measurements in the control logic. Modelling techniques for real-time dynamic simulations and methods for on-line model adaptation are presented in this section.

The detailed dynamics model of gas turbine engine can be expressed with a system of differential equations in state space [9]:

$$\begin{aligned} \dot{X} &= F_x\left(X,U,V\right) \\ Y &= F_y\left(X,U,V\right) \end{aligned} \tag{1}$$

where X is state coordinate vector, U is control vector, V is vector of operating conditions and Y is vector of output observable coordinates.

Control vector $U = \left[\dot{m}_f, \alpha_{VGV}, \alpha_{BOV}\right]$ in adopted modelling approach consists of following control variables:

$\dot{m}_f$ - fuel flow demand,

α_{VGV} - variable guide vane demand and

α_{BOV} - bleed/blow-off valve demand

This vector $\boldsymbol{U}$ is a subset of vector Λ, which contains full set of controlling variables used in real plant, i.e. gas turbine engine. (Fig.4.)

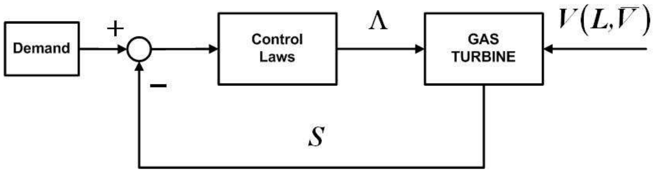

Fig. 4. Structure of GT system

Vector of observable coordinates $Y = \left[\bar{S}, \bar{\bar{S}}\right]$, consists of two sub-vectors, $\bar{S}$ and $\bar{\bar{S}}$. Vector $\bar{S}$ represents measured engine parameters, and vector $\bar{\bar{S}}$ non-measured engine variables:

$$Y = \left[T_{amb}, P_{amb}, n_{gg}, n_{pt}, T_{TLT}, \ldots\right] \tag{2}$$

A particular dynamic condition of the engine is described by the derivative vector in Eq.1., where the state vector $X = \left[n_i, \bar{X}_j\right]$ can be formed arbitrarily to some degree. In proposed modelling framework, vector of state coordinates consists of following elements:

n_i - rotational speed, where i is number of shafts and

$\bar{X}_j$ - sub-vector, where j corresponds to number of gas path components used to model required engine configuration.

Each sub-vector $\bar{X}_j$ is represented with vector of gas properties, which corresponds to the specific gas path component:

$$\bar{X}_j = \left[P_j, T_j, \rho_j, \dot{m}_j\right] \tag{3}$$

where elements of vector correspond to total values of pressure P_j, temperature T_j, density ρ_j and mass flow rate $\dot{m}_j$ at the specific engine station.

In steady-state conditions, all derivatives are equal to zero, and Eq.1. transforms into system of following equations:

$$\begin{aligned} &f_x(X, U, V) = 0 \;;\; F_x \equiv f_x \equiv 0 \\ &Y = f_y(X, U, V) \;;\; F_y \equiv f_y \end{aligned} \tag{4}$$

The nonlinear operators F_x and F_y are reduced to the operators f_x and f_y at the steady state conditions, where operator f_x implicitly includes operator f_y. If $\dot{X} \neq 0$ then solution of the system $f_x \neq 0$ does not satisfy static model, and such solution corresponds to a dynamic point, i.e. not steady-state condition.

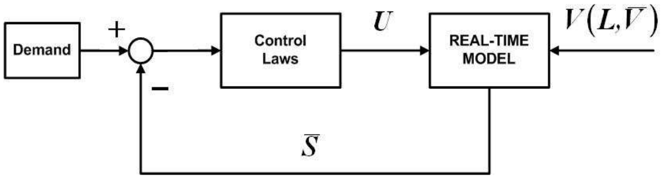

Fig. 5. Structure of H-i-L architecture

Vector of operating conditions $V = \left[\nu, \bar{V}\right]$ consists of sub-vector $\bar{V} = \left[P_{amb}, T_{amb}, \rho_{amb}\right]$, which is ambient conditions vector and scalar ν. In case of H-i-L configuration [Fig.5.] scalar ν is represented with gas turbine loading L, and in on-line configuration [Fig.6.] this scalar is substituted with rotational speed of loading shaft n_{pt}.

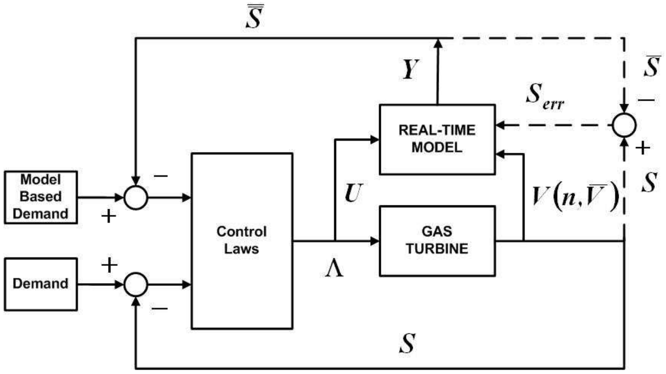

Fig. 6. Structure of On-line architecture

In on-line configuration, models over the time can deviate enough from the actual engine being monitored so that the model estimation is inaccurate and misleading. Various engine component characteristics change during the life of the engine and this can be recognized as degradation of different component efficiencies.

To mitigate this problem, tuning of engine model can be performed so that model aligns to actual engine being monitored using model tracking approach [10-12]. The idea behind this approach is to minimize the deviations or modelling errors of model that runs in a parallel to real plant, i.e. gas turbine, by correcting parameters of “nominal” engine’s behaviour. This approach effectively removes the uncertainty introduced with engine to engine variability, different disturbances, unknown initial conditions and modelling simplifications.

The tracking methods are usually based on the errors between the measured engine variables and the corresponding simulated variables, where generated errors are used to correct engine component characteristics. The measured variables S are compared with simulated variables $\overline{S}$, and then corresponding simulation error vector S_{err} is multiplied by a matrix, where resulting vector is used to correct the model state variables X. Therefore dynamic model described with system of equation Eq. 1, in on-line configuration takes following form:

$$\begin{aligned} \dot{X} &= F_x(X,U,V) + F_z(S - \overline{S}) \\ Y &= F_y(X,U,V) \end{aligned} \tag{5}$$

where function F_z represents gain matrix, which determines model correction and sensitivity to measurement noise.

MODEL BASED CONTROL

A wide range of engine sensors covering high and low bandwidth signals are used for controlling the gas turbine during steady state and transient operations. Some of these engine variables are of crucial importance for the safe operation of the engine from perspective of components life and stable operation. Not all of the engine variables can be measured, or they can be measured only with reduced accuracy and reliability. Traditionally control system transforms real engine limits into limits, which are based on measured engine variables. As a result of that, engines operate with increased safety margins and thus with non-optimal performance.

To overcome this problem model based control concept has been proposed [13-15]. Model based approach exploits real-time engine models to estimate control feedback signals, enabling the implementation of novel control and diagnostic methods [16]. Using real-time on-line engine models it is possible to obtain non-measured engine variables, such as temperature loading of high pressure turbine component or surge margin of the compressor.

Real-time engine models generated by GasTurboLib tool can be used to calculate temperatures, pressures, mass flow and gas properties at relevant engine stations. This particularly applies to stations for which no measured data are available. Using these models it is possible to accurately calculate the dynamic responses of parameters, which are not available or data are affected with high measurement lags or low update frequencies. These virtual measurements can be used for the development of model based control methods. Proposed concept offers the possibility to contribute towards improvements in engine reliability, e.g. combustor stability [17], component life extension [18] and performance diagnostics [19].

Transient turbine over- temperature protection

As an illustration of model based approach, the example of gas turbine transient over-temperature control is presented in this section. To prevent turbine component damage induced by excessive, prolonged combustor outlet gas temperature, the engine is operated at a turbine peak temperature that is several degrees below the turbine vane's critical temperature. Typically, the turbine component is protected by the engine control parameter, which is based on the measured gas temperature.

The temperature of the combustion gases leaving the combustor is not directly controlled because measurement of the extremely high temperature at the inlet of the high pressure turbine is very difficult and impractical. Hot gas temperature is usually measured by a plurality of thermocouples disposed either at the outlet of turbine section or between high pressure and low pressure turbine. At these engine stations, energy has been already extracted from hot gas, and gas temperature is correspondingly reduced to a suitable level, which may be practically measured.

With the engine operating at steady-state operating condition, adequate engine life can be assessed by limiting peak temperature based on the measured gas temperature. During engine acceleration, however, peak temperature may be exceeded transiently because the response of the measured gas temperature does not reflect the true critical turbine temperature.

Because the thermocouple probes at the turbine exit are constructed for accuracy and durability, but not for quick response, temperature measurement results in a lag with relatively slow response as compared to that of the critical turbine hardware. Although consideration of this temperature lag is not critical for engine accelerations of long duration, the delay becomes most significant when attempting to accurately compensate for thermocouple dynamics during rapid accelerations of short duration.

In order to prevent transient temperature overshoot from damaging turbine, model based temperature limiting control parameter has been devised. Proposed model based parameter can adequately reflect unmeasured critical turbine temperature during rapid engine transients [20].

In presented example rapid transients of single shaft gas turbine engine have been considered. During full load acceptance simulation, the response of VGV's (variable guide vanes) has been varied. VGV demand is determined using a least select from the speed and temperature schedule. Speed schedule is based on gas generator rotational speed, and temperature schedule is based on the turbine operating temperature. On fast load changes, however, the VGV's are limited against fuel demand. Rapid application of load and increase of fuel demand during load acceptance causes VGV's to move to fully open position. Several scenarios have been simulated using different VGV opening rates, and these cases are presented in Fig. 7.

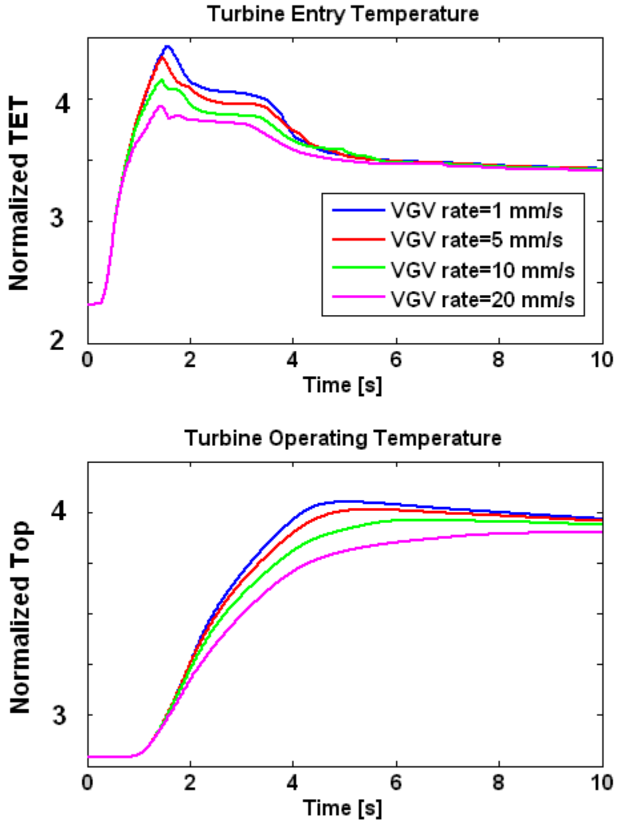

Fig. 7. Temperature profiles during rapid transient

Turbine Entry Temperature (TET) and Turbine Operating Temperature (Top) are presented and compared in this figure. Turbine Entry Temperature is model based temperature (MBT) generated using real-time model, and Turbine Operating Temperature is measured gas temperature (MGT) obtained by engine thermocouples. Top is defined with following relation:

$$T_{OP} = T_{pte} - T_{inlet} + K_{T_{op}} \left(T_{inlet} - T_{amb} \right) \qquad (6)$$

where T_{pte} is average power turbine exit temperature , T_{inlet} is average inlet temperature, T_{amb} ambient reference temperature and $K_{T_{op}}$ is constant dependant on the fuel type.

One can observe that by reducing VGV opening rate, transient thermal loading of engine is increased. Also it can be clearly observed that Measured Gas Temperature (Top) exhibits substantial lagging compared to Model Based Temperature (TET).

Proposed transient over-temperature control parameter is based on permissible thermal load to which turbine is subjected during rapid transients. Turbine thermal load can be derived from heat flow that is transferred from hot gas to metal, and is defined as follows:

$$\dot{\theta}_h = h_h A_h \left(T_{hg} - T_m \right) \qquad (7)$$

where h_h is averaged heat transfer coefficient for hot side, A_h is contact surface for hot side, T_{hg} is hot gas temperature and T_m is metal temperature.

Introducing expression for cooling effectiveness:

$$C_{eff} = \frac{T_{hg} - T_m}{T_{hg} - T_{coolant}} \qquad (8)$$

into expression for heat flow (Eq.7.) one can obtain following relation:

$$\frac{d\theta_h}{dt} = h_h A_h C_{eff} \left(T_{hg} - T_{coolant} \right) \qquad (10)$$

where coolant temperature can be substituted with measured compressor discharge temperature $T_{coolant} = T_{cd}$, and hot gas temperature can be equated with model based turbine entry temperature $T_{hg} = TET$.

Assuming that heat transfer coefficient h_h and cooling effectiveness C_{eff} are constant during engine transients for the conditions above nominal turbine entry temperature, it is possible to rewrite heat flow equation in the following form:

$$\theta_h = h_h A_h C_{eff} \int \left(TET - Tcd \right) dt \qquad (11)$$

If we limit heat that can be transferred during transients from hot gas to metal, when $\Delta T = TET - T_{cd}$ is above threshold value $\Delta T_{\lim}$, we can deduce transient turbine over-temperature limit as follows:

$$\overline{\theta_h \lim} > \int \Delta T dt \quad \text{for} \quad \Delta T = TET - T_{cd} \geq \Delta T_{\lim} \qquad (12)$$

where normalized thermal load $\overline{\theta_h \lim} = \dfrac{\theta_h \lim}{h_h A_h C_{eff}}$ is proposed as a new limiting parameter.

Proposed model based over-temperature control parameter has been evaluated for different rapid transients and shown in figure Fig.8. Transient over-temperature control parameter is obtained by integration of the normalized thermal load when the model-based temperature parameter ΔT is in excess of the threshold limit $\Delta T_{\lim}$.

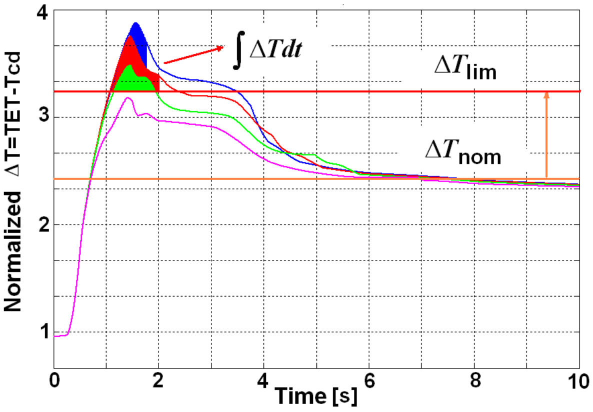

Fig. 8. Normalized thermal load parameter

Using control parameter based on MBT instead of MGT, control system can adequately reflect unmeasured critical turbine temperature during rapid engine transients. This clearly illustrates the advantage of real-time models integrated into gas turbine DCS, which can be used as a basis for development of model based control and protection algorithms. The presented example has shown that the model based temperature parameter, generated by real-time on-line model has the potential to contribute towards better over-temperature protection and component life extension.

CONCLUSIONS

Implementation of aero-thermal engine models into gas turbine control system has been discussed in this study. Modular flexibility of DCS architecture offers easy extension of system capability through integration on-line gas turbine models executing on dedicated hardware platform. It has been demonstrated the current state-of-art industrial DCS can successfully support development of real-time model based control concepts for industrial gas turbine engines.

Real-time execution of non-linear models and their integration through network communication within distributed system have been addressed in this paper. To overcome limitation of potential model obsolescence, flexible modular modelling framework has been proposed.

Hardware-in-the-loop facility has been considered as a transitional architecture in this development programme. Modelling techniques of real-time dynamic simulations for H-i-L and On-line control applications have been presented.

This paper has laid out the benefits associated with implementation of real-time engine models into industrial gas turbine distributed control systems. Discussed capabilities range from different model based control and protection algorithms to engine model based diagnostic and prognostics methods. The concept of model based control has been demonstrated using the example of gas turbine transient over-temperature control.

REFERENCES

[1] Schulke, W. W., 2004, “Redundant networked controls for industrial turbines”, ASME Paper No. GT2004-53946.

[2] Culley D., 2010, “Transition in gas turbine control system architecture: modular, distributed and embedded”, ASME Paper No. GT2010-23226.

[3] Berner A.W., 2010, “Challenges for implementing a distributed-control system for turbine engines”, ASME Paper No. GT2010-23640.

[4] Bourne D., Dixon R., Horne A., 2009, “Distributed control systems for aero gas turbine engines: a wicked problem for systems engineering”, Proc. 7th Annual conference on systems engineering research, Loughborough, UK.

[5] Kulikov G. G., Ar’kov V. Yu., Abdulnagimov A. I., 2008, “Hardware-in-the-loop testing technology for integrated control and condition monitoring systems of aircraft gas turbine engines”, Russian Aeronautics, Vol 51, No. 1, pp. 47-52.

[6] Panov V., 2009, “GasTurboLib – Simulink library for gas turbine engine modelling”, ASME Paper No. GT2009-59389.

[7] Culley D. E., Behbahani A. R., 2008, “Communication needs assessment for distributed turbine engine control”, NASA/TM-2008-215419.

[8] Tovar E., Vasques F., 1999, “Real-time fieldbus communications using profibus networks”, IEEE Transaction on industrial electronics, Vol.46, No.6, pp.1241-1251.

[9] Kulikov G. G., Thomson H. A., 2004, “Dynamic modeling of gas turbine”, Springer.

[10] Brunell B. J., Viassolo D. E., Prasanth R., 2004, “Model adaption and nonlinear model predictive control of an aircraft engine”, ASME Paper No. GT2004-53780.

[11] Volponi A. J., 2005, “Use of hybrid engine modeling for on-board module performance tracking”, ASME Paper No. GT2005-68169.

[12] Borquet S., Dewallef P., Leonard O., 2006, “A way to deal with model-plant mismatch for a reliable diagnosis in transient operation”, ASME Paper No. GT2006-90412.

[13] Lietzau K.,Kreiner A., 2001, “Model based control concepts for jet engines”, ASME Paper No. 2001-GT-0016.

[14] Turevskiy A., Meisner R., Luppold R.H., Kern R.A., Fuller J.W.,2002, "A model-based controller for commercial aero gas turbine", ASME Paper No. GT-2002-30041.
[15] Tagashira T., Mizuno T., Koh M., Sugiyama N., 2009, "ATF test evaluation of model based control for a single spool turbojet engine", ASME Paper No. GT2009-59854.
[16] Fuller J. W., Kumar A., Millar R. C., 2006, "Adaptive model based control of aircraft propulsion systems: status and outlook for naval aviation applications", ASME Paper No. GT2006-90241.
[17] Morgans A.S., Dowling A. P., 2005, "Model based control of combustion instabilities", ASME Paper No. GT2005-68897.
[18] Guo T., Chen P., Jaw L., 2005, "Intelligent life-extending controls for aircraft engines", NASA/TM-2005-213373.
[19] Borguet S., Dewallef P., Leonard O., 2005, "On-line transient engine diagnostics in a kalman filtering framework", ASME Paper No. GT2005-68013.
[20] Panov V., 2010, "Modelling and simulation of transient thermal loading in a gas turbine", Proc. 11th European Fluid Machinery Congress, Edinburgh, UK.