Model-based Reasoning Approach for Automated Failure Analysis: An Industrial Gas Turbine Application

Gulnar Mehdi[1], Davood Naderi[2], Giuseppe Ceschini[3], Mikhail Roshchin[4]

[1,4]*Siemens AG, 81739, Germany*

gulnar.mehdi.ext@siemens.com
mikhail.roshchin@siemens.com

[2]*Siemens Industrial Turbomachinery AB, 61241, Sweden*

davood.naderi@siemens.com

[3]*Siemens AG, 90001, Germany*

giuseppe.ceschini@siemens.com

ABSTRACT

Keeping up with the technological advances, turbo-machinery industry aspires to integrate manufacturing, servicing and maintenance of their plants. Typically, these objectives may be accomplished by adoption of condition monitoring services and diagnostic solutions, resulting in improved plant operations, lower maintenance cost, and impart safety and reliability. Specifically, failure analysis, within systematic diagnostics, is a fundamental feature of design and maintenance phase, as it allows fault identification, and its causes and effects that propagate at different system levels. With the large number of subsystems and process flows, failure analysis for industrial gas turbines is non-trivial, and requires expertise of system mechanics, aerodynamics, thermodynamics, etc. Consequently, in order to realize an efficient system analysis, we device an automated model-based approach to failure analysis for industrial gas turbine applications. This paper presents context-independent qualitative models of key turbine components, which are most error-prone, together with their potential failure mode descriptions, and their impact at different system levels. Using an existing reasoning engine, we present behavior models and results for two most vulnerable turbine subsystems i.e. Lubrication Oil System and the Core Gas Turbine Engine. Finally, we evaluate the practical use-cases of this model-based solution implemented for diagnostic services at Siemens AG.

1. INTRODUCTION

Over the decades, the turbo-machinery industry has been operating complex and expensive machines, with a long history of providing quality products and services to their customers. In addition, this industry has been successful in utilization and implementation of various degrees of diagnostics, prognostics and health management capabilities, which has helped the entire turbo-machinery industry to manage and keep up with the desired efficiency of their massive systems, and most importantly, gain customer loyalties. Nevertheless, the automation curve is still pretty steep, and the plant performance is highly dependent on diverse and time-variant technical, operational, environmental and financial conditions (Siemens AG, 2014).

From the customer perspective, large process units produce daily revenue in excess of 5 million US dollars. In this context, component availability, reliability assessment and optimization are an important part of plant revenue and profits, as stated by Forsthoffer (2011). Consequently, industrial communities remain concerned to maximize reliability and product throughput, and at the same time minimize the maintenance and operating cost. This can be achieved by adopting a new business model that integrates manufacturing, service and maintenance, and furthermore, employs intelligent diagnostics and prognostics technologies. This integration would aim to boost the industrial plant operations, its longevity, impart safety, reliability, asset integrity, mitigation of risks, and help to continuously improve plant's compatibility to variable conditions. According to the current trend, following are the key areas being explored as a part of automation and reliability improvement programs (Forsthoffer, 2011):

- Site reliability audits
- Assessment methods
- Availability improvement plans
- Condition monitoring techniques
- Diagnostics Solutions
- Prognostics and maintenance plans

Focusing on the practical implementation of aforementioned solutions, numerous industry leaders emphasize on failure analysis as an input to diagnostic framework. Failure analysis serves as a baseline to identify and analyze the most error-prone units, and strengthen the tangible problem solving capabilities. This approach serves to determine system and component or sub-component failures, and its impact across the subsystems. Therefore, failure analysis is a key enabler and attributes to "enlighten" the diagnostic capabilities. It also improves the design, service and maintenance decisions by anticipating required actions, and provide unprecedented insight into the system's health. Thus, it is widely adopted as a successful approach.

From stakeholder's perspective (including senior managers, end-users, service engineers, design engineers etc.), failure analysis is a way to quantify reliability, and improve the quality of the plant. These stakeholders are a part of design and maintenance cycle and contribute in their own capacity. The motivation and role of the computer scientist is to provide next-generation technology tools, in order to match the diverse requirements set by the stakeholders (see Fig. 1).

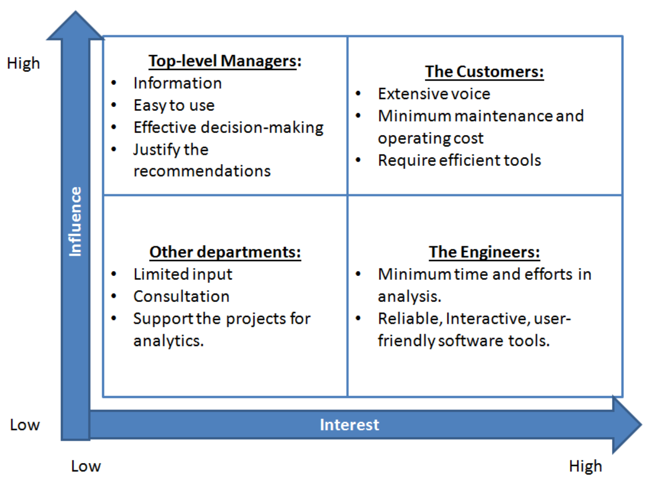

Figure 1. Stakeholder Interest and Influence Matrix

Fig. 2 shows the current environment and infrastructure of the turbo-machinery industry that provides opportunities to enhance the software services for diagnostics and prognostics. These services can be data-driven or knowledge-based expert systems. The services are mainly translated by using the available sensor technology, central database technology and feature requirements from a large group of data consumers.

Considering the different perspectives described above i.e.; i) the reliability management and improvement process, ii) adoption of failure analysis approach, iii) requirements from the stakeholders, and iv) opportunities within the existing infrastructure (software services), the experts themselves have to decide a methodology and tools that best fits for them for failure analysis and which also align with the standards. This is indeed a task because it requires them to understand the properties of the system failures, the standard requirements and how to achieve it.

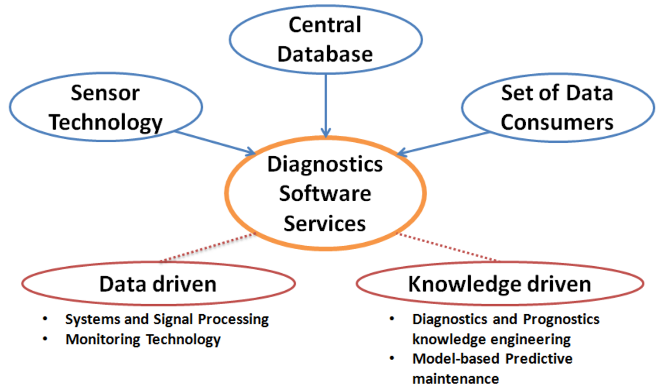

Figure 2. Environment and Infrastructure for Turbo-industry

Another constraint to the practical implementation of failure analysis for turbines is a diverse set of configurations of every unit. Every power generating plant has different operating and process requirements and thus, often differs in its design. In addition to this, it is non-trivial to capture the behavioral properties and dependencies of critical units in the rotating equipment because it requires greater expertise of mechanics, thermodynamics and aerodynamics as discussed by Ceschini and Saccardi (2002). Currently, several off-the-shelf approaches are available that conduct failure analysis manually and/or take support from semi-automated tools. The results from these approaches are high in efforts and costs, while still compromise the quality with respect to the completeness and accuracy of results. This identifies the demand for a systematic and innovative solution as an addition to the available software services. The solution should be reliable, easy to use and cost effective.

To address the above mentioned challenges, this paper presents a model-based solution to automate the task of failure analysis for diagnostic purposes. The high demand of a sophisticated tool would justify to the design and service recommendations, especially when the changes in the system design happen on yearly basis. We adopt a software engineering approach at a very abstract level in order to make a context-free solution that is independent of a fixed structure or architecture design.

The core of our solution task is devising a software system to identify faults and their impact. Elements for this task, are:

- specifications of the faults modes of the components, turbine situations and ambient conditions;
- the failure modes, which are violations of system functions;
- and impacts such as turbine trip or low shaft speed. The impact can be monitored at different system-levels such as component-level, sub-system level and in the entire system.

Our software solution follows the knowledge-based systems and software engineering principles for problem solving and is based on so-called qualitative deviation models (Werthne, 1994) to capture the domain application. These models can capture how significant deviations from nominal behavior are generated and propagated by components models. By using an automated model-based reasoned along with an existing constraint-based predictive algorithm (Raz'r OCC'M, 2014), we provide a model-based generation of failure analysis results (which has been developed for the physical components of the system, and also extended to include electrical control units) along with its effects. Our solution has been successfully introduced at Siemens AG and this paper presents some industrial use-cases of our implementation.

The paper follows with Section 2 describing the application task at hand with an overview of industrial gas turbines. Section 3 presents the proposed model-based solution architecture and its foundations. Section 4 presents the case study and two use-case scenarios for industrial gas turbines along with the results. Finally, we conclude in Section 5 with discussion and future outlook.

2. THE APPLICATION BACKGROUND

2.1. Reliability Perspective of Industrial Gas Turbines

In the context of rotating equipment engineering, gas turbines moves product i.e. gases; either for power generation or mechanical drive applications. In general, every unit of a plant consists of a driven machine, driver, transmission device and is supported by auxiliary equipment as discussed by Forsthoffer (2011). Fig. 3 shows topology of an industrial plant.

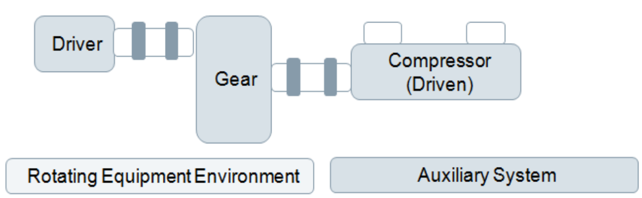

Figure 3. Industrial Plant Landscape

Each of the equipment mentioned above can be classified further and have different configurations. For example: drivers can be classified as steam turbines, gas turbines, motor (Induction, synchronous or vari-speed) or engines (Internal combustion, Diesel or Gas). The key is to understand the functionality of its critical components in order to effectively monitor and maximize plant safety and reliability. Reliability is commonly defined as the amount of time equipment operates in one year. It is an ability of the equipment unit to perform its specified function without a forced (unscheduled) outage in a given period of time (Forsthoffer, 2011). In case of an outage, the loss of revenue can exceed a million U.S. Dollars a day as shown by Forsthoffer (2011). The cause of an outage is usually the shutdown of a critical component. Many leading companies including our industrial partner recognize the reliability management of the critical component and adopt following strategies (Ceschini and Saccardi, 2002):

- Involve the end-user in the specification, design and installation phase of the plant.
- Determine the life span of the plant and its component which is extremely long compared to development phases.
- Analyze the instrumentation and location of the plant that directly impact the equipment's reliability.
- Focus the design and installation because it has a substantial influence on the maintenance requirements, its cost and availability of particular piece of machinery.

2.2. Failure Analysis

Failure analysis fulfills the reliability requirement by predicting what could go wrong in the system. It determines the severity and probability of a component's failure mode that can occur in a given system and is considered to be a bottom-up inductive technique which starts from faults/failure modes and ends at the resultant effects (Dobi, Gleirscher, Spichkova, & Struss, 2013).

2.2.1. Requirements on fault identification and impact determination

The pre-requisite of performing model-based failure analysis is to check the system design for completeness and consistency of the models. The analysis could go wrong when the rules and component models are wrongly designed. Furthermore it can be applied to many different levels in a hierarchy of an industrial plant shown in Fig. 4. The effects are produced at the boundaries of the systems and subsystems, and for this reason it is necessary that the intermediate effects keep track of fault/failure modes at each level (Dobi et al, 2013). The analysis can be performed by using parameters such as severity, probability of occurrence, and detect ability. In our system, we consider the effects of (single) faults on the system behavior.

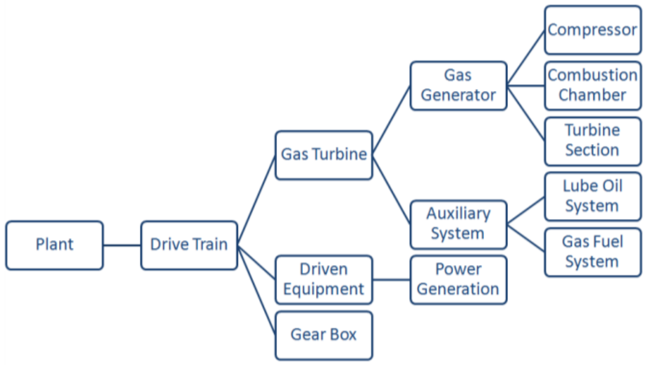

Figure 4. Decomposition of an Industrial Plant

In Fig. 5 we draw an example of an axial bearing using a traditional failure analysis template as presented by Abilla (2011). The process starts by defining failure modes as a first step, the functional aspect, type, failure impact, causes and detection mode. The criteria i.e. severity, occurrence and detection levels are calculated to quantify the decisions, setup priorities and corrective measures.

Functions	Failure Type	Potential Impact	Severity	Potential Causes	Occurrence	Detection Mode	Detection	RPN
Axial Bearing: maintain bearing temperature	bearing temperature increases	Shaft speed decreases	10	Lube oil temperature, cooling air temperature, compressor flow rate	2	Check flow rate during startup and so on	3	60

Figure 5. Axial Bearing Failure Mode Analysis

The tool is acquired to address the reliability and quality aspects of the system. Though, manual adoption of failure analysis can be very expensive. But with automation, it can be cost effective in terms of design changes and can increase satisfactory level of manufacturers and customers. In our presented solution, we automate by using behavior models and check for implication or entailment to the functions.

3. A SYSTEMATIC MODEL-BASED APPROACH TO FAILURE ANALYSIS FOR INDUSTRIAL GAS TURBINES

The proposed approach has been applied to a specific product line of Siemens industrial gas turbines. The following section describes the application details and solution implementation.

3.1. High-level System Design

Fig. 6 shows the high level system design of our solution. Using remote monitoring service database (i.e. MS SQL database in our case), first we formulate interesting turbine scenarios by adopting sensor signal processing techniques. These scenarios are presented as set of complex event processing rules using physical parameters that define different states of the turbine. In the next step, we instantiate complex event processing for each unit under consideration. Few examples of these events are "*modelX01 startup condition*", "*modelX02 turbine operating high ambient conditions*", "*modelX03 operating low ambient conditions*" etc. In parallel to this, we develop a component library to model various critical components of the turbine system. The nominal (OK mode) and faulty behaviors (failure mode) of each component is captured as qualitative constraints along with its impact on the system level as presented by Struss (2004). Structure for a given physical system is defined separately as interface variables that will connect components together. Once the component library and structural description is made available, we construct a system model for analysis. System model comprises of different configurations supported by connecting the components in various styles. Once the system model is ready, we run an existing automated model-based reasoner for failure analysis task. Part of this algorithm is presented by Struss and Fraracci (2014) which solves for the finite constraint satisfaction problem. The reasoner considers the complex events as scenarios and a specific system model description for each plant to check if the propagation of a failure would entail the local or system level impact or not. Finally the model-based failure analysis results are presented as recommendation and alert messages. The solution determines the impact that may occur under a particular failure mode and predicts whether it can lead to a critical situation or violate any reliability requirement. These results serve as in input to the diagnostic framework. They are useful together with other methodologies to strategize and follow for root-cause analysis and other diagnostic tasks. The qualitative model-based system development and its foundation concepts are described in the following sections.

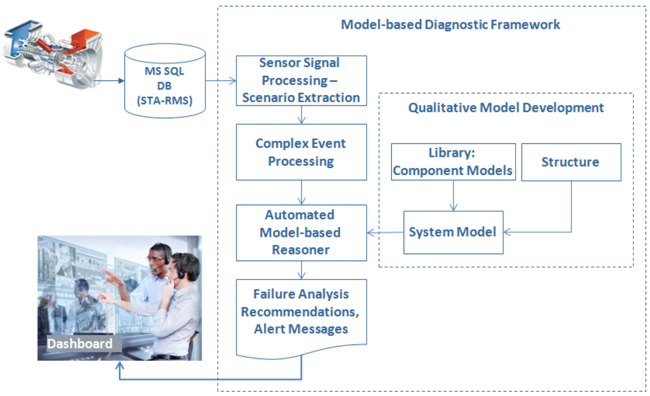

Figure 6. High-level System Design

3.2. Model-based Solution Foundations

The key feature of model-based approach is the re-use of models and easy adaption to new structures/topologies and variants (physical system, software architecture). Some key features described by Struss (2008) of the models used in our work to produce successful results are:

- The models are context-free and compositional.
- The models should realize how the faults in one component of the system propagate to the rest of the system.
- The model's qualitative deviations from their nominal behavior serve as a basis for detecting faulty components.

The failure analysis formalization as shown by Struss and Fraracci (2014) considers a set of scenarios and a set of relevant component failures and checks whether they can lead to an unwanted effect (violations of the system functionality). If we consider one component and one faulty mode $\mathrm{MODEL_F}$, for a given input scenario SCEN, we need to check whether it entails the specified effect EFFECT, or that they are all consistent with each other. The check can be performed by a constraint satisfaction algorithm (Dobi et al, 2013).

$$\mathrm{SCEN} \cup \mathrm{MODEL_F} \vdash \mathrm{EFFECT}$$

$$\mathrm{SCEN} \cup \mathrm{MODEL_F} \cup \mathrm{EFFECT} \nvdash$$

E.g. consider the scenario where the gas turbine is in operating phase, compressor is on and the rotor shaft is active. If the compressor falls into the high pressure faulty mode, then it is possible that the rotor shaft speed will turn less than it should which is also the effect. Formally the inference of the system is:

(GasTurbine_Demand = "*Operating*") ∪ (Compressor FaultMode = "*High_Inlet_Pressure*") ⊢ *Low_ShaftSpeed*

(GasTurbine_Demand ="*Operating*") ∪ (Compressor FaultMode = "*High_Inlet_Pressure*") ∪ *Low_ShaftSpeed* ⊬

3.3. Tasks

The task is to make the failure analysis of the turbine system, the causal relationships between faults which occur in the system and their effects which are unintended behavior of the different components such as bleed valve stuck opened etc. These effects are part of series of situations such as turbine is in operating condition, coasting-down, and stopping, with high ambient conditions or normal ambient and so on. The overall impact is either the automatic trip or effects on exhaust pressure, temperature and mass flow. The purpose of this work is to identify possible faulty components that can lead to trips of the turbine or high exhaust conditions that can cause high $C0_2$ emission, with the objective to reduce these risks by maintenance, redesigning the existing components, or adding others in some cases e.g. more sensors. Information provided from the industrial partner includes information about the modes of operation of the turbine, system functions, list of faults, turbine situations, and impacts.

Our task is achieved by modeling two sub-systems components that is of the core engine and lube oil system, identifying the failure modes and their effects along with the overall impact on the turbine system.

4. CASE STUDY: AUTOMATED FAILURE ANALYSIS OF AN INDUSTRIAL GAS TURBINE

The case study was conducted to demonstrate the feasibility of the approach described in section 3 for one of the product-line of Siemens industrial gas turbines. The turbine system in general has a number of sub-systems that work together to perform a specific task such as power generation or mechanical drive. Fig. 7 gives an overview of turbine model at sub-system level. These subsystems are functional and can be configured differently for every model/design of the industrial plants. In this paper, we present our solution for two of the most vulnerable sub-systems i.e. lubricating oil system and the core gas turbine engine. It is important to note that the main system which is evaluated is the gas turbine at system level, which has: turbine driver, the physical subsystems containing both electrical and mechanical subsystems; the software controlling part and the electrical subsystem. The turbine driver gives commands to the turbine like start up, coast-down and stop. The physical system contains all the necessary mechanisms to allow the physical phenomena like gas flow path, combustion, etc., to occur. The electrical system offers the platform to send commands initiated from the software package of the turbine, which in the end are needed for the mechanical system and its components.

In the following sections, we present the component models and results for Siemens use-cases respectively, including fault modes, effects, the turbine conditions (e.g. turbine driving situations), and the impacts.

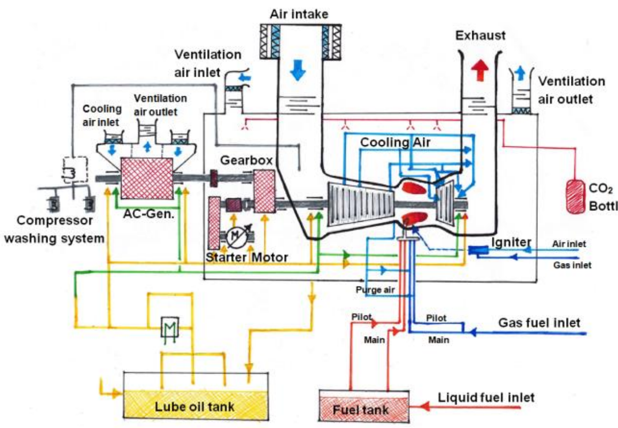

Figure 7. Schematic view of an Industrial Gas Turbine

4.1. Use-case #1: The Core Gas Turbine Engine

The core gas turbine engine is the heart of any industrial gas turbine. Its purpose is to generate a flow of pressurized hot gas which is converted into mechanical energy, which drives the load (e.g. an electric generator) via a gear box.

The specified model under consideration operates in an open cycle with straight air and gas flow through the turbine. The core engine can be divided into three major sections: namely the compressor, the combustor and the turbine section.

4.1.1. The Physical Model

The main mechanical, thermo-dynamical, hydro-dynamical and software (control unit) components considered in the study of the core engine are presented in the Fig. 8. The ambient air is captured and either cool down or heated up by the heat exchanger component. Later the compressor draws this air and compresses it by using an adiabatic process of thermo-dynamics. The compressor is dependent on startup motor in the turbine startup phase and uses variable guided vanes and bleed valves to control the pressure ratio and prevent surge. The compressed air enters the combustor where it is heated up. The burner mixes the gas fuel coming from the fuel system with the compressed air in the combustor and maintains stability of the main and the pilot flame. Finally, the hot gas from the combustor enters the turbine section. The turbine section expands the air and drives the compressor and the generator. The gearbox transmits power from turbine to the generator. Ultimately the generator is being operated to generate electricity for the power grid and the hot gas is exhausted by the diffuser to the air exhaust system. The rotor assembly is associated with the rotor shaft speed and considers the rotor welded on the shaft. It consists of casing, blades, discs and bearings. Here we only consider the radial and thrust bearing that affects the shaft speed when faulty. The cooling system maintains the temperature of the bearings. These mechanical components are controlled from specialized Electronic Control Units (ECUs) which controls the heat exchanger, start-up motor, variable guided vanes, bleed valves, rotor assembly and the gas fuel system.

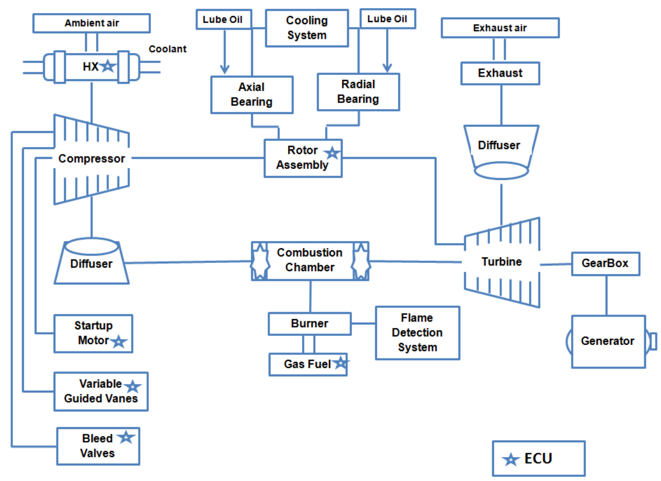

Figure 8. Physical Model: The Core Gas Turbine Engine

4.1.2. Component Models

In this section, we show the basic examples of component models, their physical quantities, domain types, connecting terminals and conventions that we have modeled so far for the physical system with the intention to connect them as in Fig. 8. The components exchange variables which represent physical quantities through the interfaces (terminals). The physical quantities exchanged between them are: temperature (T), pressure (P), flow rate (F), position (pos), Speed (V), Active power (A), signal/commands etc. and their deviations from nominal values expressed as Δ"Physical_Quantity", e.g. for pressure it would be ΔPressure. Most models variables and all deviations have values from the domain Sign = {-, 0, +}, whereas he commands and states have Boolean values {0, 1}.

The core purpose of the core engine model is to determine if the pressure ratio in the compressor is sufficient enough, temperature in combustor is nominal; rotor speed is up to the setting point and power output of the turbine can synchronize with the load (e.g. generator).

Example Component: Variable Guided Vanes

The purpose of variable guided vanes, VGV, is to control the compressor inlet mass flow. It is controlled by ECU, which send signals (command = {0, 1}) to **position** the VGV depending on the inlet temperature and rotor shaft speed. The qualitative behavior model of VGV is described as:

Position = function(Command)

The faulty modes:

- "***Stuck_at_PositiveSwirl***" is when Position is greater than the function(Command)

- and "***Stuck_at_NegativeSwirl***" is when Position is less than the function(Command).

The position of the VGV depends on the signal received by the ECU which can either be correct or wrong. . This will have an effect on the whether the VGV is positioned correctly or wrongly also what effects it will produce. If the command received from the ECU is wrong, e.g. the ECU.Command = False (meaning the VGV should be closed) but there is a deviation in the command ECU.ΔCommand = True, and in the mean time the VGV is actually Open (i.e. Position > function(Command)), even though there is no deviation in the physical component (ΔPosition =0) due to the SW error, the valve still results to be in the faulty mode: *Stuck_at_PositiveSwirl*. This faulty mode leads to a ΔPosition different from zero, as the VGV will transfer increase the inlet mass flow when it should have not.

Example Component: Compressor

The compressor model captures the pressure balance, heat flow and mass flow equations in a qualitative fashion. These equations are dependent on quantities exchange by rotor assembly, bleed valves, VGVs, heat exchanger, motor, and turbine section. The model for Pressure Balance is described as:

Pressure_from_Compressor = pressure_ratio ⊗ Pressure_from_heatexchanger

Where pressure ratio is a function of rotor shaft speed:

Pressure_ratio = function(Speed_from_Rotor)

The temperature rise is modeled as:

Temperature_from_Compressor = Temperature_from_HX ⊗ (Pressure_ratio) ⊗ (gamma_constant -1 / gamma_constant).

where (gamma) is ratio of specific heat at constant pressure to volume.

The Heat transfer function under the adiabatic process is seen as Q = 0 whereas the mass flow is function of the pressure ratio, inlet mass flow and VGV angle.

Flow_from_Compressor = Flow_from_HX ⊕ Flow_from_VGV ⊕ pressure_ratio.

The faulty mode captured as deviations are:

- ***"SurgeDetected"*** happens if ΔPressure_ratio and the ΔFlow_from_Compressor is positive
- "***HighDifferentialPressure***" occurs when ΔPressure_from_Compressor is positive;
- Similarly ***"LowDifferentialPressure"*** occurs when ΔPressure_from_Compressor is negative.

Example Component: Rotor Assembly

The rotor assembly defines the rotor shaft speed of the turbine and the compressor. The speed is dependent on the inlet temperature and it highly affected by the temperature of the bearings. The qualitative nominal behavior can be described as:

RotorSpeed = M^+*(inlet_temperature, bearings_ temperature)*

and ΔRotorSpeed = 0; where M+ is a monotonic function.

The fault modes captured as deviation models are:

- ***"UnderSpeed"*** if ΔRotorSpeed is negative
- **"*OverSpeed*"** if ΔRotorSpeed is positive.

4.1.3. Failure Analysis Results for Core Engine

Table 1 shows the model-based generation of failure analysis results for the core turbine engine produced automatically by reasoning engine. The result does not outreach the expert knowledge but only supports in analysis by automating the mechanistic part of their work. Given the current system models, operating scenarios and failure modes, the tool is cost effective, correct and complete to capture various settings of a complex system, whereas presents qualitative results that facilitate the human reasoning. These results have been evaluated against the real industrial scenarios and fits best to their knowledge. For example: under the turbine startup scenario, the motor failure may abort the operation while in operating mode its failure has no impact. Likewise, under operating with normal ambient conditions may trip due to compressor's failure of surge or low pressure ratio or diffuser leakage etc.

Table 1. Partial Failure Analysis Results of Core Turbine Engine

Turbine Scenario	**Turbine Component**	**Failure Mode**	**Impact**
Startup	Startup Motor	Broken	:startup abort
Operating Normal Ambient	Startup Motor	Broken	>>no system-level effect<<
Operating Normal Ambient	Variable guided vanes	Stuck at Positive Swirl	:increase compressor pressure
Operating Normal Ambient	Variable guided vanes	Stuck at Negative Swirl	>>no system-level effect<<

Operating Normal Ambient	Bleed Valves	Stuck at Open	:reduce compressor pressure
Operating Normal Ambient	Compressor	Surge Detection	: turbine trip
Operating Normal Ambient	Compressor	High Pressure	:increase turbine pressure
Operating Normal Ambient	Compressor	Low Pressure	>>no system-level effect<<
Operating Normal Ambient	Diffuser	Leakage	:reduced turbine pressure
Operating Normal Ambient	Rotor Assembly	Over Speed	:increase turbine power
Operating Normal Ambient	Radial Bearings	High Temperature	:reduce rotor speed
Operating Normal Ambient	Radial Bearings	High Vibration	:reduce rotor speed

4.2. Use-case # 2: Lubricating Oil System

The second use-case conducted at Siemens AG is the lubricating oil system. This is a part of an auxiliary system. Its purpose is to supply oil of correct pressure and temperature to the gas turbine engine. The pressure and temperature of the oil is continuously monitored to secure safe operation of the turbine.

4.2.1. The Physical Model

The main components of the lubricating oil system are shown in Fig. 9. The heater maintains the temperature of the oil tank, whereas the pressure in the tank is controlled by the fan component. The oil tank is a reservoir of oil and supplies the oil to the pumps. There are three pumps; primary, backup and emergency pump that operate in different conditions. These pumps provide sufficient pressure to the oil and transfer to the check valve. The check valve component transfers some oil to the cooler and some to the temperature control valve. The oil temperature is cooled down in the cooler. The temperature control valve mixes the cold oil and the hot oil in a specified ratio to achieve required temperature of the oil. The filter component cleans the oil and transfer the oil to the gas turbine components. The electrical control unit (ECU) controls the heater, fan, temperature valve and the electric motors.

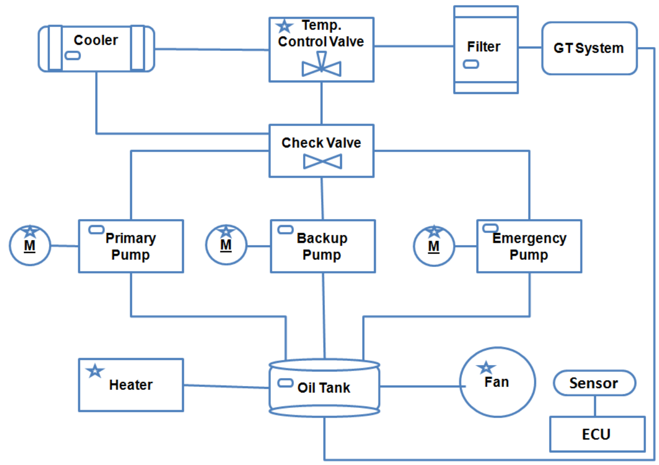

Figure 9. Physical Model: The Lubricating Oil System

4.2.2. Component Model

In this section, we show the examples of components of the lubricating oil system as mentioned in Fig. 10. The physical quantities exchanged between the components are same as described above for the core engine.

The purpose of the lubricating oil system is to determine deviations in pressure, temperature and flow rate of the lube oil so that the gas turbine system can operate in a safe way.

Example Component: Heater

The heater keeps the oil in the tank at a minimum temperature required for the turbine. It is controlled by ECU and operates in all the turbine situations except the stopping. During standstill when the temperature is above 15 Celsius, the heater can be turned off in order to extend the oil life. Under OK mode, the heater is described as:

Heater_State = function(Command);

and If (State = 'On') then Temperature_from_heater is positive; Else Temperature_from_heater is zero;

The faulty mode is described as:

- ***"OverHeating"*** if State = *'On'* and ΔTemperature_from_heater is positive;
- ***"LowHeating"*** if State = 'Off' and ΔTemperature_from_heater is negative;

Example Component: Oil tank

The oil tank is a reservoir of oil and one of a main component of this system. Sufficient level of lube oil is required in order to provide sufficient flow of lube oil to the gas turbine. The oil tank interacts with the turbine system that drains the oil into the tank. The nominal behavior of the tank is model as:

For the flow out of the tank is related with the state of level in the oil:

Flow_from_Tank = Oil_level;

The temperature and pressure output from the tank is simply propagated as a sum of gas turbine, heater and fan quantities. For example:

Temperature_from_Tank = Temperature_from_bearings ⊕ *Temperature_from_heater;*

The faulty modes are:

- ***"Leakage"*** if ΔFlow_from_tank and Oil level is negative;

Example Component: Electric Motor

The electric motor provides active power to its next component e.g. the pump. It is controlled by the ECU and given the command ({0, 1}) it either turns 'on' or 'off' in order to drive the pump or shut it off. Such that if (Command = 1) then ActivePower_from_Motor = 1 else zero.

- Faulty mode "***ElectricDriveFault***" is identified where active power stays at zero irrespective of the command.

4.2.3. Failure Analysis Results of Lubricating Oil System

Table 2 shows partial failure analysis result for lubricating oil system. The analysis of auxiliary system is significant in the workings of the core engine. It is observed that most of the time the fault exists in the auxiliaries whereas the engineers waste their efforts in analyzing the core components. Therefore, it is important to model auxiliary components and their failures to derive complete and correct results. Our results produced by reasoning engine comply with the specifications provided by Siemens. For example: during turbine operation, continuous supply of lube oil is required to the bearings with required temperature and pressure. Therefore, faulty components of lube system will directly affect the pressure, flow rate and temperature in the bearings. If all three pumps or motors are faulty during operation, the turbine ultimately trips.

Table 2. Partial Failure Analysis results of Lubricating Oil System

Turbine Scenario	**Turbine Component**	**Failure Mode**	**Impact**
Startup	Electric Motor	Broken	:startup abort
Startup	Primary Pump	Broken	:startup abort
Operating Normal Ambient	Heater	Over Heating	:increase bearing temperature
Operating Normal Ambient	Primary Pump	Broken	:turbine trip
Operating Normal Ambient	Vacuum Fan	High Pressure	:increase bearing pressure
Operating Normal Ambient	Oil Tank	Leakage	:reduce oil flow

5. CONCLUSION

In this paper, we focus on automating the inference of failure modes and its impact for diagnostic tasks. The analysis is performed on the physical components and electronic control units (software). The component models are represented as qualitative relations, and hence, maintain their abstraction. We built models using compositional and context-independent modeling approach. The primary goal of these models is the analysis of component failure and propagation of its effect at turbine system level. The failure modes and its impact are captured as qualitative deviations. The turbine conditions, fault models and effects are represented by sets of constraints. The fault analysis iterates over the Cartesian product of turbine situations and faults checking whether they entail the defined effect via the constrain solver. The results we have obtained can also be adopted for sensor fault analysis and in particular for root-cause analysis. Our model-based solution is of greater interest for further work on automated failure analysis in the turbo-machinery industry (combining model-based approaches from artificial intelligence (AI) and software engineering). We are currently preparing to extend the models to capture more complex behaviors, include other sub-systems and integrate all systems to derive results at a high level of abstraction that can further improve the decision-making as a part of integrated diagnostic framework at Siemens AG.

ACKNOWLEDGEMENT

We would like to acknowledge Mr. Michal Skubacz, Mr. Alexey Fishkin and Mr. Sebastian Brandt for the support and encouragement. Special thanks to our colleague Lina Panchyrz for helping with thermodynamic models and the industrial plant landscape. Sincere thanks to Prof. Peter Struss and Oskar Dressler (OCC'M) for providing software support.

REFERENCES

Energy Siemens AG (2014). [Online] Available: ttp://www.energy.siemens.com/br/en/services/industrial-applications.

Forsthoffer, W. E. (2011). *Forsthoffer's Best Practice Handbook for Rotating Machinery*. Elsevier.

Ceschini, G. F & Saccardi, D. (2002). *Availability centred maintenance (ACM), an integrated approach*. In Reliability and Maintainability Symposium, 2002. Proceedings. Annual (pp. 26-31). IEEE.

Werthne, H. (1994). *Qualitative reasoning: modeling and generation of behavior*, New York, Springer-Verlag Wien, 1994.

OCC'M, (2014). OCC'M Software GmbH. Raz'r Model Editor Version 3. Interactive Development Environment for Model –based Systems. http://www.occm.de/, ©, 1995-2011.

Dobi, S., Gleirscher, M., Spichkova, M., Struss, P. (2013) *Model-based Hazard Analysis and Risk Assessment Technical Report TUM-I1333*, Technische Universität München.

Abilla, P. (2011), *FMEA Template and Example*, [Online]. Available: http://www.shmula.com.

Struss, P. (2004) *Models of Behavior Deviations in Model-based Systems*. Proceeding of the 16th Europoean Conference on Artificial Intelligence.

Struss, P. and Fraracci, A. (2014), *Modeling Hydraulic Components for Automated FMEA of a Braking System* in: Proceedings of the Annual Conference of the Prognostics and Health Management Society 2014, Fort Worth, Tx, USA.

Struss, P. (2008). *Model-based Problem Solving*, in Handbook of Knowledge representation (Foundations of Artificial Intelligence), Elsevier, 2008, pp. 395-465.