Gas-turbine condition monitoring using qualitative model-based diagnosis

Louise Travé-Massuyès, Robert Milne

TIGER™: GAS TURBINE CONDITION MONITORING USING QUALITATIVE MODEL BASED DIAGNOSIS

Dr. Louise Travé-Massuyès

LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse Cedex 4, France

Dr. Robert Milne

Intelligent Applications Ltd, 1 Michaelson Square, Livingston, Scotland UK. EH54 7DP

1. INTRODUCTION

The TIGER system integrates several artificial intelligence technologies, including qualitative model based reasoning to perform condition monitoring of gas turbines. The effectiveness of its capabilities makes TIGER a very innovative tool which may revolutionise the way in which gas turbines are monitored.

Given the critical nature of gas turbines in most industrial plants, their availability is of prime importance. Associated maintenance costs can also be extremely high and hence, it is a high priority to find ways of reducing maintenance costs and increasing the availability of the gas turbine. Routine preventative maintenance techniques have been used for many years to minimise major problems by routinely checking and taking care of small problems.

There is a major move towards condition based maintenance and condition monitoring of gas turbines, the goal being to monitor the turbine on a regular basis in order to establish when maintenance actions need to be performed based on the condition of the turbine rather than a fixed number of operating hours. The TIGER system [1] performs gas turbine condition monitoring by using the information available from the gas turbine control system.

TIGER runs continuously on-line to the turbine, sampling the key operating parameters at once per second intervals, it performs a wide variety of checks including limit checking, dynamic response and consistency with model based predictions. Not only is TIGER able to recognise pre-anticipated fault situations and identify the presumed fault(s) in an associative manner, but it has the unique capability to « discover » new faulty situations, and to propose the malfunctioning parts of the turbine and to isolate the faulty physical component(s). This is possible because TIGER includes a diagnosis component built with qualitative and model based reasoning techniques. These emerging AI technologies fill a gap in the approach to diagnosis of industrial plants. The model based diagnosis technique allows one to decouple the knowledge about the physical system on the one hand, and the knowledge about the diagnosis task on the other hand. The qualitative modelling approach then provides a way to structure the knowledge about the physical plant in a decomposed manner, each fragment model being generic enough to represent the devices in a class. Altogether this also results in a diagnosis system significantly more robust and maintainable.

In order fully to support an engineer, TIGER also provides extensive graphical user interface displays to help view the state of the turbine as well as extensive trending support system to allow the user to select sets of data and view long term trends.

The first TIGER condition monitoring system has been in continuous use at the Exxon Chemical Fife Ethylene Plant since 1992. It has been instrumental in identifying the underlying problems for a number of situations, one of which resulted in considerable financial benefit to Exxon [2].

The TIGER system has been installed on several gas turbines in the UK to date (1996) and is proving itself an invaluable tool. For each new turbine that TIGER has been installed on, it has identified several previously unknown, but significant problems within the first few days of operation. It transforms the engineer's understanding of the turbine, providing a continuous inside in the health of the gas turbine, as compared to the traditional check every 10,000 operating hours. The feedback from the gas turbine engineering community is that TIGER will make a significant change in how gas turbines are monitored, reducing costs, fuel consumption and pollution.

1.1 The Fife Ethylene Plant Application

The initial installation of TIGER was at the Fife Ethylene Plant which is a 650,000 tonnes a year gas cracking facility located in South East Scotland, and jointly owned by Exxon Chemical company and Shell Chemical company. The major product of the facility is high grade Ethylene for use in the plastics and butyl rubber industries in both the UK and on the continent. The feed stock is ethane gas obtained from the Shell/Esso offshore facilities in the North Sea. The process is continuous with the Ethylene product being transported by both ship and pipeline to end users in the UK and on the continent. The gas turbine is a 28 mega-watt General Electric Frame Five with two shafts supplied by John Brown Engineering (See Figure 1), this is used to drive the primary compressors for the Fife Ethylene Plant. It is a vital item of equipment for the plant. If the turbine fails the entire plant must shut down. This turbine is typical of large scale industrial gas turbines.
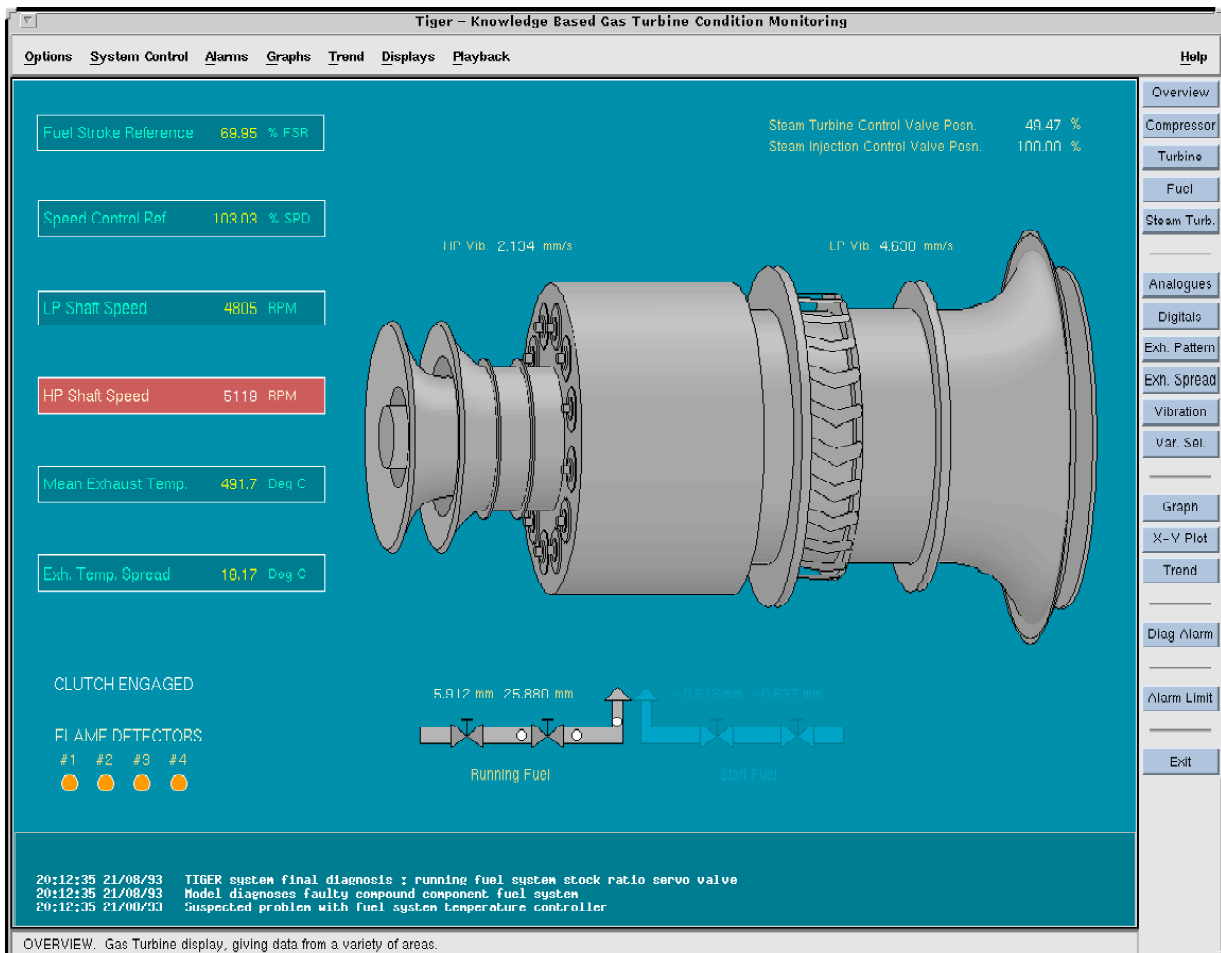

Figure 1: TIGER Overview of the Exxon FEP Turbine

1.2 The Users View of TIGER

Since 1992, TIGER has been cycling once per second and monitoring for faults 24 hours a day. The senior engineers and the mechanical engineering section check the TIGER system several times a day. For example, each morning when they arrive at work, they check the TIGER system to see if there are any faults that have been identified over night.

TIGER works continuously with a direct on-line data feed from the gas turbine control system. This data input provides all the key operating parameters for the turbine. TIGER works continuously to analyse this data and evaluate the state of the turbine. At Exxon Chemical, TIGER is located in the Chief Engineer's office. There is a variety of colour mimic diagrams that give a view of the overall turbine or sub-system of the turbine. The Chief Engineer selects the colour view that he considers most interesting for his routine monitoring, normally this is an overview of the total turbine operation. All the main operating parameters are colour coded and at a glance he can identify whether there are any major problems or not. Most importantly, there is a small window known as the alarm window, which indicates whether the monitoring and diagnosis system has detected any problems.

When a turbine is working well, the engineer can quickly browse through the colour displays to reassure himself that the turbine is operating according to his own intuitive feel. If at any time TIGER believes something is not right it displays a message in the alarm window. The engineer can then go to the alarm page to see what faults are being detected. The amount of information on this page is kept very low.

A major task of TIGER is to consolidate a wide variety of faults and problems detected from different parts of the turbine and by different tools into a smaller number of high level fault conditions. This lets the engineer rapidly assess the overall state of the turbine and its problems without having to wade through a large number of fault conclusions. If he wants to know more about any faults on the alarm page, he merely selects that fault with the mouse. A new window is presented that shows the derivation of the fault based on the fault conclusions from the various diagnostic tools (See Figure 2).
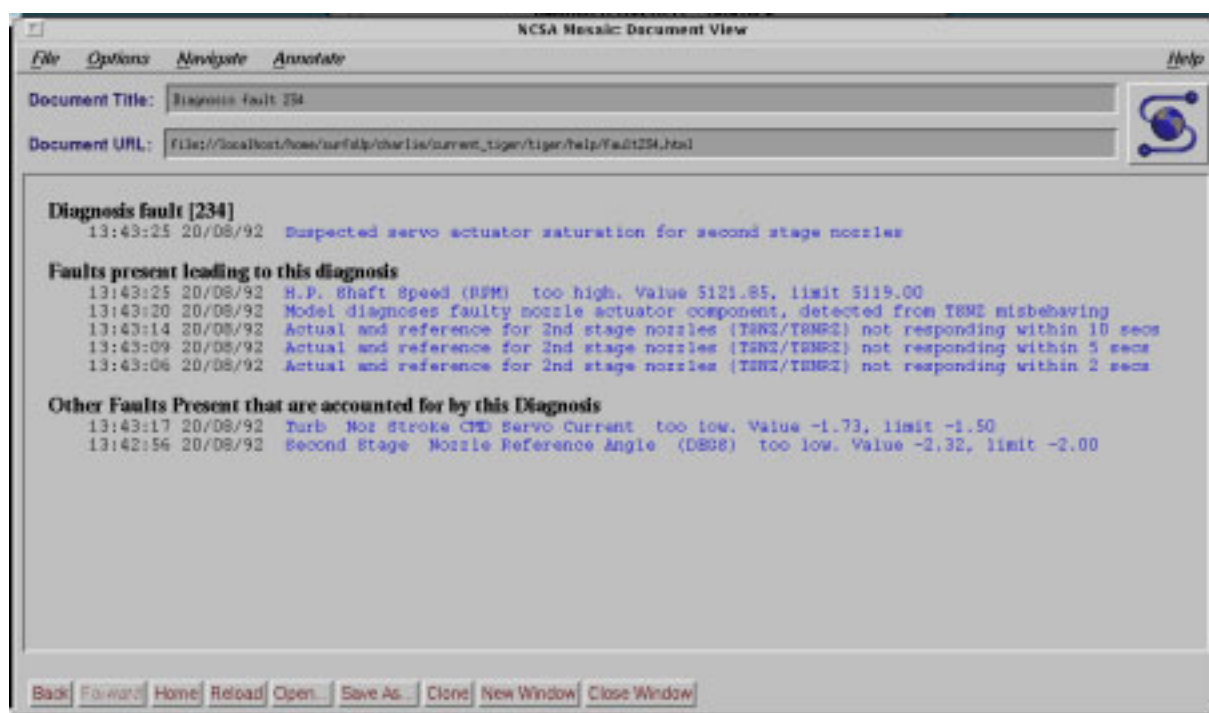
Figure 2: TIGER's Fault Manager Display

This shows the hierarchical way in which problems are identified within TIGER and the groupings that can be created to bring fault categories together. Each of these faults can also be selected and the engineer receives further information about the precise cause. In addition, for each root cause there is a link through Hypertext to the manual and documentation relevant for that fault, so very quickly he is able to transition from a high level display of the turbine with a fault indication to the high level fault, a more detailed breakdown of the fault, and finally the relevant manual or supplemental information for each problem (See Figure 3).

The gas stop valve functions in the fuel gas system to provide a positive fuel shutoff when required by either normal or emergency conditions. A trip valve is operated by trip oil acting on the piston end of a spool. A hydraulic trip solenoid is located in the trip oil line to the trip valve when the trip oil pressure is normal and the solenoid valve is energised to close, the spool is held in a position that allows hydraulic oil to flow between the control servovalve and the hydraulic cylinder. In this position, normal control of the stop ratio valve is allowed. In the event of a drop in trip oil pressure below a pre-determined limit, a spring in the trip valve shifts the interrupt the flow path of oil between the control servovalve and the hydraulic cylinder. Hydraulic oil is dumped and the stop valve closes, shutting off fuel gas flow to the Gas Turbine.
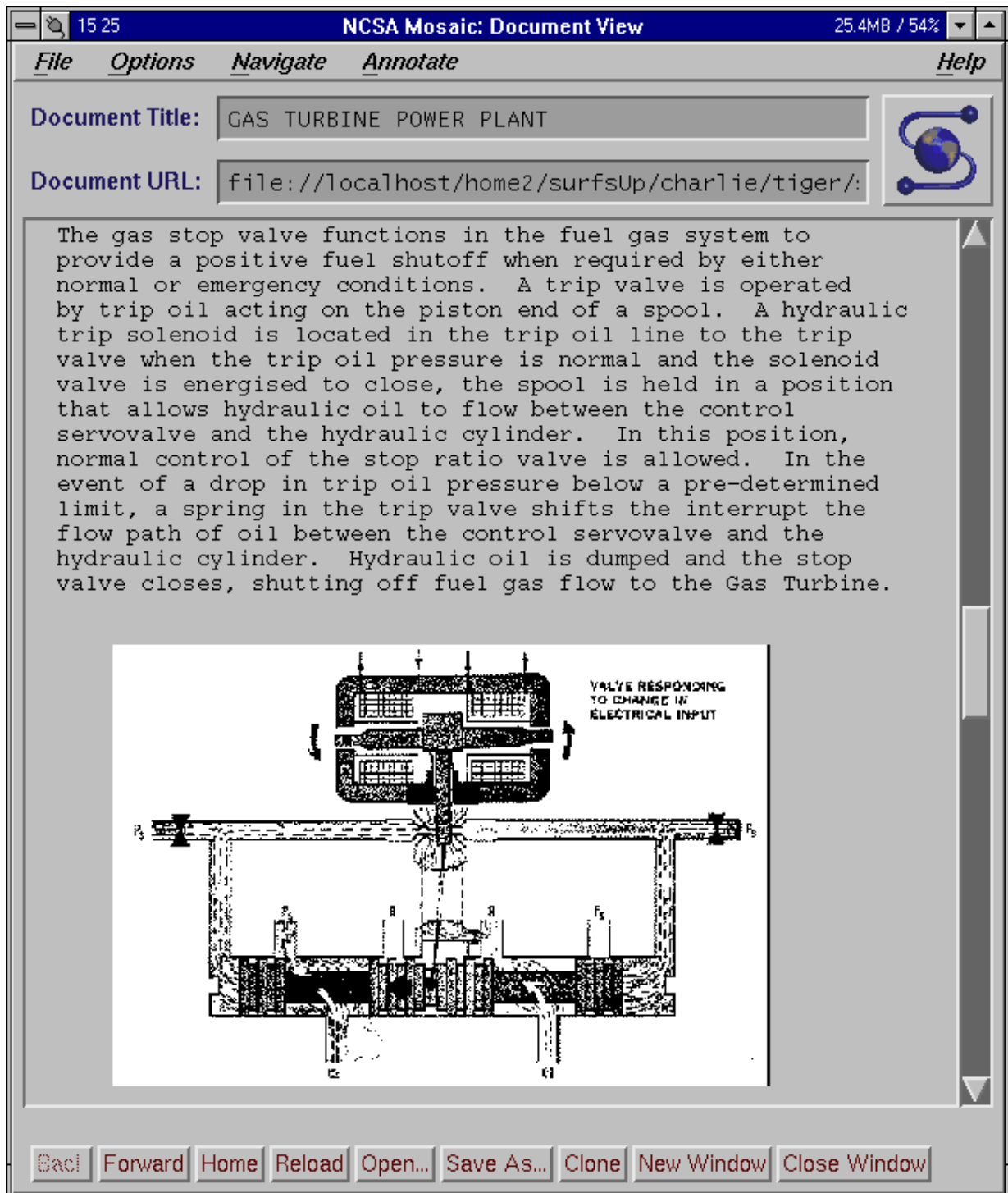
Figure 3: TIGER's Extended Help

If one would like to investigate the raw data, by clicking on the time stamp for the fault a graphical display of the relevant parameters is presented to the engineer. He is then able to browse through the raw data and investigate the exact performance of the turbine. This capability is quite important as it allows the engineer to use his extensive knowledge and experience to investigate more deeply, complex or unusual situations.

The TIGER capability is even more important and useful to the user when you realise that it runs 24 hours a day and the faults that are reported might occur during the night-time or when the engineer is not available. Through TIGER, he is able to go back and see the precise fault situation, even though it may have occurred many hours before. This is a particularly important aspect of TIGER, its ability to continuously monitor for faults, even though engineering staff or operations people are not monitoring the turbine closely. If the engineer so wishes, it is possible to save the relevant data for the fault for later analysis.

The Exxon turbine is a two shaft turbine with a set of nozzles, known as 'second stage nozzles' that balance the energy between the two shafts. This allows the compressor to run at its optimum speed while providing for variable load on the turbine. When the turbine was running very near full power, occasionally they would have a problem where the second stage nozzles appear to become stuck. Using TIGER, the engineer was able to show that the problem was that the servo was not strong enough to move the actuator against the full power of the turbine. This was very important for Exxon. It meant that when the nozzles did stick, there was no risk of a serious incident. The operators were then advised that they could run the plant to full capacity and ignore the nozzles sticking problem. It is important to realise that this problem had been effecting Exxon for 3 years but was diagnosed within one month of introducing TIGER.

The month following the change of restriction to the operators, the plant set its first production record for several years. This increase in production is worth a considerable amount to Exxon. The real proof of the benefit to Exxon however was the fact that all the employees at the plant received a special gift as a congratulations for setting the production record. This is considered a real benefit [3].

Since 1992, TIGER has helped to identify and provide diagnostic support for dozens of problems.

2. THE DIAGNOSTIC TOOLS

2.1 Diagnostic Tools

The TIGER diagnostic mechanism uses 3 independent tools; the fault manager is used to co-ordinate the conclusions of these tools. It should be noted that each tool runs independent of the others. This means that they can all work in parallel, each examining different aspects of the turbine for an early detection of problems. This also means that for each application, the appropriate subset of these tools can be selected. The tools are as follows:

2.1.1 KHEOPS

KHEOPS [4] is a high speed rulebase system. It is used to express diagnostic rules in a classic rule based formalism. It also allows the user to set oriented pre-alarm limits for each parameter. These limits are adjusted by the user by a simple menu driven interface. Although the primary usage of KHEOPS is as a limit checking system, it is in fact a full powered rulebased expert system language.

The KHEOPS limits used within TIGER go beyond normal limit checking. This is because of the ability within KHEOPS to express different conditions for the limits. Hence, in many cases, for example monitoring the fuel values, different limits apply depending on the operating regime of the turbine. In addition, any specific fault rules that want to be inserted can be used through KHEOPS. The KHEOPS system is capable of very high speed execution. Response time less than 15 milliseconds is possible.

2.1.2 IxTeT

The IxTeT [5] functionality is a very powerful capability to monitor the dynamic reaction of the gas turbine. It is also a unique capability in TIGER that is not found in other condition monitoring systems. Essentially within IxTeT the user is able to specify a sequence of events, called *chronicle*. The system then monitors to ensure that the chronicle is properly followed. For example, when the load set-point increases on the turbine, we expect to see the fuel valves open and shortly after the temperature increase, resulting in an increase in shaft speed. This causal sequence can be encoded easily in IxTeT situations. The tool then detects that the starting incident has occurred and ensures that the cascade of subsequent events is maintained properly.

Time limits can be specified to ensure the system response is adequate. This is helpful in detecting a wide range of problems. For example, if a servo is partially blocked and resulting in slower movement of the actuator, the IxTeT system can detect this by the delayed time response. A fundamental element of any dynamic system is its causal reaction to events. IxTeT is designed to monitor these situations.

In addition, specific situations can be described that can recognise faults. For example, a specific chronicle can be implemented in IxTeT to detect nozzle sticking or dirty servos. Hence, IxTeT can be used to either describe the normal causal reaction or look for specific patterns resulting from known faults.

2.1.3 CA-EN

CA-EN [6] is a model-based supervision system devoted to complex dynamic systems. CA-EN's representation formalism allows one to combine empirical causal knowledge and first principles of the domain. The CA-EN formalism is based on a two-level representation scheme for the description of a physical system:

- an analytical equation level which allows one to represent algebra-differential equations.

- a causal graph in which the links represent the causal paths along which a perturbation is propagated.

Both levels can manage imprecise knowledge. A CA-EN program represents a formal model of the physical system built from knowledge about the physics underlying the behaviour of the system. This model can be viewed as an implicit behavioural model of the physical system. CA-EN has two processing modules:

- a simulation module which produces the explicit behaviour of the physical system in terms of the values of the internal variables across time according to the behaviour of the exogenous variables. Imprecision is managed with interval values, which implies that predicted graphs are curve envelopes.

- a diagnosis module which accounts for fault detection and isolation of faulty components. Fault detection is based on models of normal behaviour. The on-line simulation of these models provides a way of implementing a discrepancy detection procedure. This enables monitoring of the behaviour of the system and detection of early deviations from the nominal behaviour. The diagnosis algorithm falls into Reiter's model-based diagnosis framework [7] and uses the CA-EN causal graph as the System Description (SD). It relies on the collection of conflict sets, i.e. sets of components such that the observations indicate that at least one of the components in a set must be behaving abnormally, and the use of an incremental hitting set algorithm.

The diagnoses are given as sets of faulty components labelled by their corresponding time of failure.

The CA-EN diagnosis system conclusions rely on a reasoning based on the physics underlying the behaviour of the system, i.e. physical laws and empirically known causal interactions. When a fault is detected, it is viewed as the violation of some of these physical principles which then guide the isolation of the faulty component(s). As a consequence, there is no need to anticipate the faults, which is highly valuable in most complex engineering domains. On the other hand, as variables and parameters take interval values, one can easily adapt the model's granularity to the requirements of the faults. Hence CA-EN has a wide coverage of faults, from those radically changing the behaviour of the physical system to those causing smooth deviations.

2.2   The Fault Manager

The three primary tools in TIGER work in parallel, each has a particular strength and a particular class of faults and problems that it monitors for. Depending on the exact behaviour of the turbine and the sensitivity chosen for some of the fault conditions, these tools can detect a wide variety of minor fault conditions. The fault manager is used to filter the minor fault conditions and only present to the engineer those conditions that he considers interesting. In addition, it provides a means of consolidating the conclusions of the various tools.   This provides two important capabilities:

The first is that in some cases it can eliminate minor faults as being unimportant or accounted for by some other action. For example, if KHEOPS detects that the compressor shaft speed is too high, but IxTeT concludes that the second stage nozzles are reacting slowly but still within acceptable response times, then the fault manager will suppress the high compressor shaft speed message as it is likely that the shaft speed will go back within limits very quickly. The fault manager is one of the unique capabilities of TIGER and is a major advance in the use of multiple knowledge based tools to filter information to only that which is relevant for the engineer.

The second aspect is that it provides a mechanism for the reinforcement of fault conclusions, for example the KHEOPS tool may detect that there is a high shaft speed, a short time later the IxTeT tool may conclude that the second stage nozzles are not responding in their expected response time. At approximately the same time the CA-EN model based prediction would conclude that the nozzles are not following the expected response giving the operating parameters of the turbine. These three together both explain the cause of the high shaft speed for the compressor but also give confirmation that something is not quite correct with the nozzle movement system.

3.   THE CA-EN QUALITATIVE MODEL BASED PREDICTION AND DIAGNOSIS

3.1   The Prediction Procedure

3.1.1   The CA-EN Knowledge Representation Formalism

Time is an essential variable in the representation of dynamic systems. In CA-EN, we have chosen to deal with time explicitly, abstracting it to a logical clock-based sampled time. Among other things, having a linear temporal scale allows us to easily evaluate state durations and event dates and to be synchronous with the observations coming in real time from the physical system [8].

CA-EN is able to handle numeric (real-valued) and symbolic variables. However, only numeric variables were used in the TIGER applications [9]. For these variables, we refer to their *quantity space* as their value set which is assumed to be a closed interval of reals. The value of the variable

may be any subinterval of its quality space. We call *variation* their value change within a unit of time. The variation value may be a real number (if precisely known) or any interval of reals.

One of the original aspects of the CA-EN formalism is being based on a two-level representation scheme for the description of the relationships between the process variables: *a local constraint level* and a *global constraint level.*

The local constraint level is in agreement with our perception of a physical process as a net of interacting variables influencing one another. This is represented by a directed graph in which the paths presume the perturbation flow causality. The influences supported by the edges of the graph allow for representing causal dependency type knowledge. The CA-EN formalism allows for two kinds of causal relations : (1) influence-based relations which are concerned with cause-effect interactions among numeric variables; (2) information-based relations which are concerned with direct information about variable states. These latter relations were not useful in the TIGER applications and they are not described in this paper.

An influence-based relation between a variable X and a variable Y is assumed to represent a linear first order type relation, i.e. it corresponds to a linear first order differential equation. Since any differential equation can be brought back to a set of first order differential equations by defining a set of intermediate variables, influence-based relations allow one to represent any type of linear dynamic phenomena. These relations are described by means of the following predicates:

$$I^+(X,Y,c,K,T_d,T_r), \quad or \quad I^-(X,Y,c,K,T_d,T_r)$$

X, Y are the influencing and the influenced variable, respectively;

c is the activation condition of the causal relation (the influence from X to Y is said to be activated whenever c holds);

K is a positive interval or real number representing the *static gain* of the influence;

T_d is a positive real number or interval representing the *delay* of the influence, i.e. the time needed by Y to react to a variation of X;

T_r is a positive real number or interval representing the *response time* of the influence, i.e. the time needed by Y to get to a new equilibrium state after having been perturbated.

Note that if K is not known numerically, it can be given by an order of magnitude relation between the variation of Y and the variation of X. An order of magnitude formalism [10] then allows us to interpret K as a numeric interval. This option provides good expressiveness to the formalism. Very important is that the formalism allows one to cope with *imprecise temporal knowledge* (since T_d and T_r can be defined as intervals), which is highly useful in real applications. Also note that a weight may eventually be associated to the influence to account for non linearities and distortion phenomena.

The global constraint level is composed of functional numeric constraints associated with interval domains, e.g. constraints arising from physical laws. In other words, a global constraint is any mathematical equation, which may be non linear as well, in which each unknown is assumed to take on interval values. This allows us to manage imprecise knowledge at this level as well. The global constraints are expressed by means of the four following predicates which are interpreted in the interval algebra :

```
(ideriv, x, y)              for x=dy/dt
(isum, x_{n+1}, x_1,..., x_n)    for x_{n+1}=x_1+...+x_n
(imult z, x, y)             for z=xy
```

```
(ipower, x, y,n)                    for x=y^n
```

3.1.2 The CA-EN Prediction Algorithm

The prediction algorithm performs a simulation, i.e. prediction across time. The input data are the causal model - including initial conditions - and the evolution of the measured variables of the causal graph over time. The output of the system is the behaviour of each process variable which is displayed as the process runs [11]. Before starting the simulation itself, the causal model is first compiled. The compilation mainly consists of generating the tables containing influence delays and response times, the structure of the causal graph and eventually, the fuzzy weights of influences and the numeric values of the static gains. The temporal unit of the simulation is that of the logical clock that is set according to the swiftness of the process. At each clock tick, the exogenous variables' values and variations are first updated. Then these new values and variations are used to estimate the other variables' values and variations, exploring the causal network in a breadth-first manner. The following steps are executed :

At the local constraint level :

1.1- *Computation Of The Net Variation*

The computation of the net variation that a variable Y undergoes consists of processing a weighted sum of the activated influences having exerted on the variable during the last time-interval. If X_i, $i=1,...n$, are the active variables influencing on Y, we have :

$$\delta y(t) = S(_{i=1,...,}n) w_i \delta y^i(t) \tag{1}$$

where w_i is the weight associated to the influence from X_i, $\delta y(t)$ ($\delta x_i(t)$) is the net variation of Y (X_i) from instant $t-1$ to instant t, and $\delta y^i(t)$ is the variation induced by $\delta x_i(t)$ on Y assuming that only X_i influences Y, which is called the *marginal variation* of X_i on Y. Marginal variations are computed taking into account that an influence represents a first order linear differential equation. For the influence-based relation $I+(X_i,Y,c,K,T_d,T_r)$ between X_i and Y, this is formally expressed as:

$$a_1 dy(t)/dt + a_2 y(t) = b_1 x_i(t-T_d) \tag{2}$$

where coefficients a_1, a_2 and b_1 are real numbers or intervals.

If T_s is the sampling period, the values of Y and those of X_i at the sampling instants can be directly related by a recurrent equation which is equivalent to (2) in a sampled temporal scale.

$$y(t+1) = ay(t) + bx_i(t-d) \tag{3}$$

where:

t		stands for the logical time (an integer)
d		is the least integer greater than or equal to T_d / T_s
$a = e^{-T_s/\tau}$ $b = K(1-e^{-T_s/\tau})$		if T_r is finite (τ is the *time-constant* of the influence which corresponds with a good accuracy to $T_r/3$)
$a=1$ $b=KT_s$		if T_r is infinite (integration relation)

10

If we had a negative influence $\Gamma^-(X_i, Y, c, K, T_d, T_r)$ instead of $\Gamma^+(X_i, Y, c, K, T_d, T_r)$, then the coefficient a would remain unchanged, but b would have a negative sign.

The recurrent equation relating the variation of Y to that of X_i (the marginal variation) at each sampling instant is directly obtained from (3):

$$\delta y^i(t+1) = a_i \delta y^i(t) + b_i \delta x_i(t-d_i) \qquad t=0,1,2,... \qquad i=1,...,n \qquad (4)$$

where a_i, b_i and d_i are the coefficients related to the influence of X_i on Y and are defined in the same manner as in equation (3). The delays and response times are automatically taken into account.

The net variation of Y is then obtained by equation (1) which sums up all the marginal variations weighted by the fuzzy weights of every influence w_i, $i=1,...n$. These fuzzy weights are updated according to the set of active influences at hand [6].

1.2- Computation Of The Updated Variables' Values[1]

For each variable Y the net variation is summed with the value of the variable at the previous clock tick.

$$y(t) = y(t-1) \oplus dy(t) \qquad (5)$$

The result is an interval.

At the global constraint level :

2.1- Refinement Of The Updated Variables' Values

The numeric intervals obtained for the updated values (Equation (5)) are refined with the global constraints by performing a tolerance propagation algorithm [12] on the set of variables. The effect of the tolerance propagation algorithm is to filter (reduce) for consistency the values $y(t)$ using the global constraints.

The simulation results produced by the CA-EN prediction module are envelopes (See Figure 4) for the variables of interest. The envelopes provide the upper and lower bound of the variable values at each sampled instant. As a consequence of the interval-based reasoning used in CA-EN, the results are complete but not precise.

---

[1] *If the set of influences arises from a linear relation, the updated value of Y is directly computed by summing up the values obtained from (3) for every X_i, i=1,...,n. This gives better results than equation (5) — although consistent — because of well known problems of the interval algebra. The presented procedure is used if the set of influences arises from a non linear relation which has been linearised.*
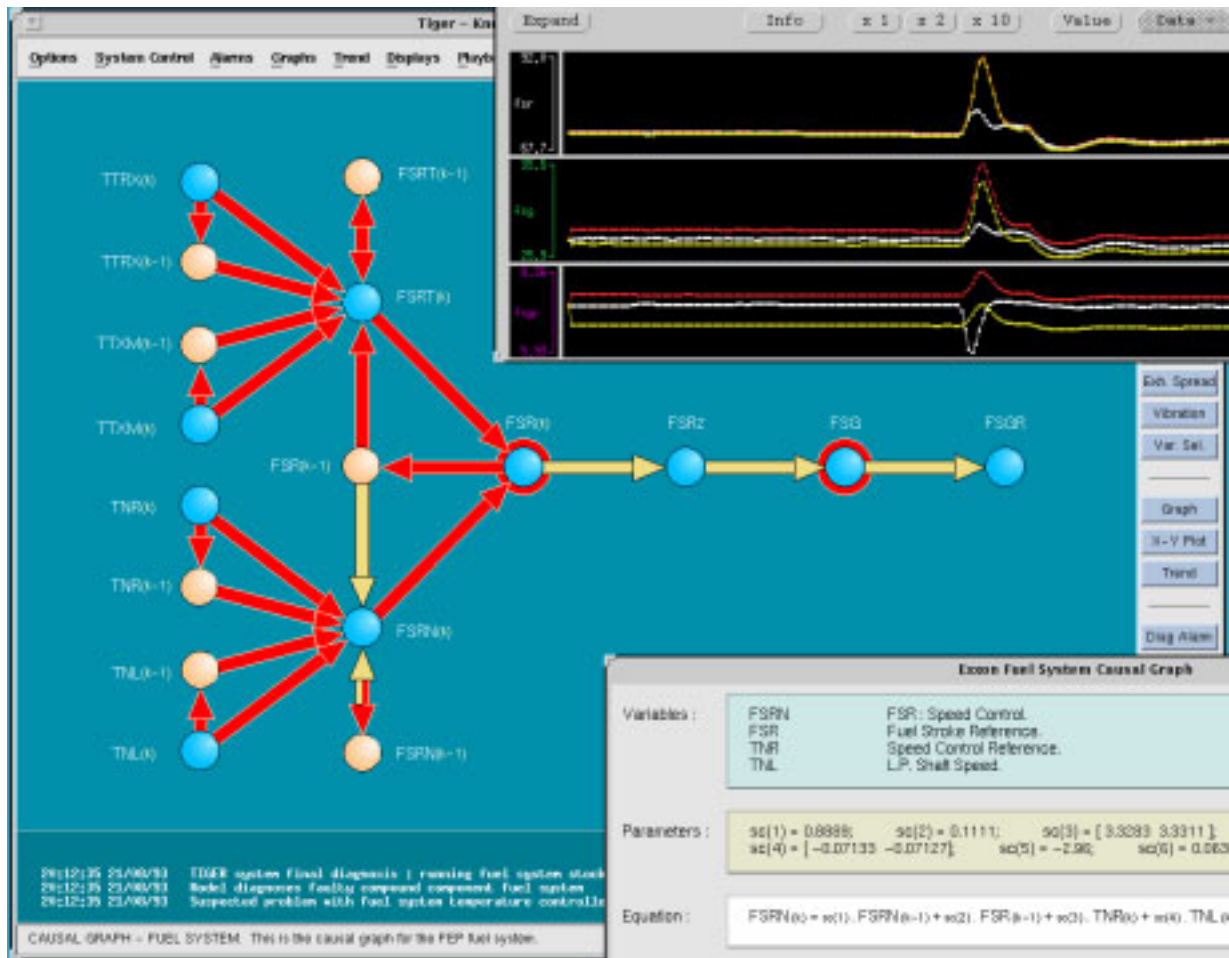
Figure 4: CA-EN Causal Graph and Interval Predictions for the Fuel System

3.3   The Diagnosis Procedure

The CA-EN diagnosis algorithm, like most model based diagnosis algorithms, includes two procedures : fault detection which is based on the results of the prediction mechanism, and conflict and diagnosis generation which follow Reiter's diagnosis methodology [7]. This algorithm is incremental in the sense that it revises the set of suspected components as new observations and eventually new conflicts are outlined [16][18].

General Diagnosis Algorithm

At each clock tick do:

1. « Reset » the measured variables which behave normally and « reset downstream » the measured variables which do not behave normally (see following comment).

2. Run the prediction procedure,

3. Run the detection procedure,

4. **If** abnormal variables are detected,

    Then run the conflict generation procedure,

5. **If** new conflicts are generated,

    Then run the diagnosis generation procedure.

The algorithm has been specifically designed for dynamic systems. In particular, the difference between *« reset »* and *« reset downstream »* in step 1 has considerable importance. Consider a measured variable Y influenced by a variable X and influencing a variable Z (with zero delays for the sake of simplicity) and assume that $Y_o(t)$ and $Y_p(t)$ are the observed and predicted value of Y at instant t, respectively. To *« reset Y »* at t means to give the value $Y_o(t)$ to Y (and ignore $Y_p(t)$). On the other hand, to *« reset Y downstream »* at t means :

(1) to give the value $Y_o(t)$ to Y to be propagated downstream in the graph. The value of Z is hence predicted as : $Z_p(t+1) = f(Z(t), \mathbf{Y_o(t)})$ (cf. §3.1.2, equation (3) applied to the influence between Y and Z) ;

(2) to consider $Y_p(t)$ for predicting the value of Y as : $Y_p(t+1) = f(\mathbf{Y_p(t)}, X(t))$ (cf. §3.1.2, equation (3) applied to the influence between X and Y).

In other words, when a measured variable Y is detected as abnormal $(Y_o(t) \notin Y_p(t))$, we use its observed value $Y_o(t)$ for predicting the values of the variables which are downstream in the causal graph but we still use its predicted value $Y_p(t)$ to update the value of the variable itself (the dynamic aspect of the influences makes that we need the value of a variable at instant t for predicting its value at instant $t+1$). If $Y_o(t)$ was used to predict future values of Y, the dynamics would be ignored and the predictions would erroneously « follow » the measures since only one step predictions would be performed.

This was tested on several fault scenarios of the APU fuel system : if the abnormal measured variables were simply reset, the fault would not be detected. On the other hand, if these variables were not reset at all, it would make it difficult to decide whether an abnormal behaviour detected on some variable is really abnormal or just the consequence of an abnormal behaviour of some upstream variable. Using the « reset downstream » mechanism allows us to decouple the physical system at the measured variables while preserving at the same time the dynamic prediction ability.

3.3.1 Fault Detection

The CA-EN fault detection procedure is based on models of normal behaviour. The on line simulation of these models is the basis of a discrepancy detection procedure which allows us to track the physical system. This is performed by comparing the predicted and observed values of variables across time so that static as well as dynamic discrepancies are detected. This is essential for controlled systems such as turbines. The controller indeed tends to compensate for the faults in such a way that the fault is only observable, and hence detectable, during the transient response of the turbine. The variables then stabilise at normal values. A classic limit checking diagnosis system is often inefficient in this kind of situation.

The comparison consists at each instant t and for every observed variable X, in checking whether the observed value $X_o(t)$ (a real number) belongs or not to the predicted value $X_p(t)$ (an interval). If not, variable X is said to be alarming at time t. From the graphical point of view, this is interpreted as the observed trajectory going out of the predicted curve envelope at time t [17].

In practice, the measurements may be noisy and the turbine may be disturbed by all kinds of perturbations that have not been taken into account into the models. Consequently, a local incompatibility between prediction and observation at some instant t does not necessarily mean that the system is faulty. In real applications, it is very important to have a robust fault detection system for a system which would untimely report faults would rapidly lose the confidence of the operator and engineer staff. Hence, we use a more robust indicator than just alarming variables. A fault is reported when a variable has remained alarming during a whole temporal interval T of length significantly greater than the sampling period. The variable is then said to be *misbehaving*.

The length of *T* may be regarded as a multiple of the sampling period and it should be adjusted according to the technology of the sensors.

As an illustration, we consider the sticking problem of the second stage nozzles. The graphs in (See Figure 5) outline the observed and the predicted values for two key variables : *TSNZ* (nozzles angle position) and *TANZ* (servo current). The first graph clearly shows how the nozzles should move, in contrast to the real behaviour. In consequence, the servo-current tries to force the behaviour of the nozzles whereas, as shown by the prediction, it should remain constant in a non faulty situation. The detection module detects *TANZ* and *TSNZ* as misbehaving variables at a very early stage. It can be noticed that even smooth deviation-like discrepancies are easily detectable, CA-EN being able to detect inconsistencies in temporal as well as magnitude aspects.
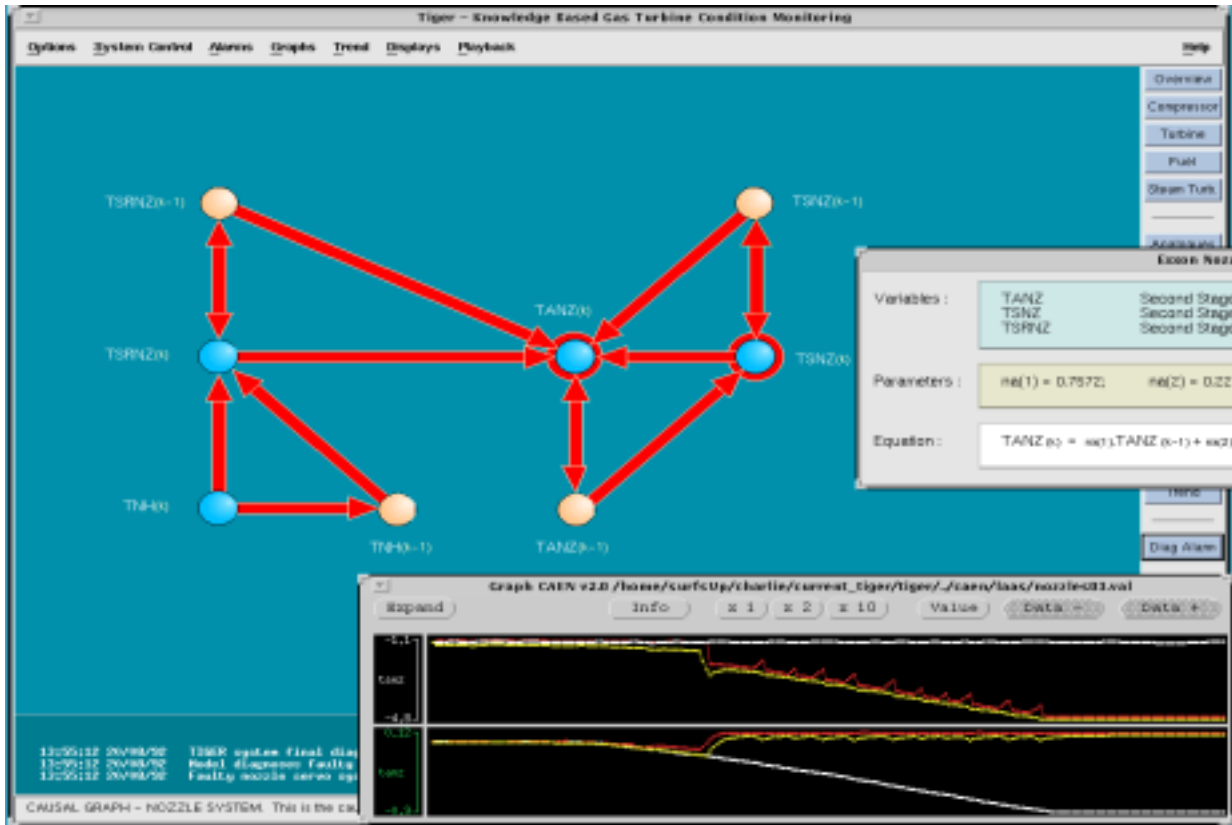

Figure 5: CA-EN Causal Graph and Interval Predictions for the Second Stage Nozzles

3.3.2 Fault Diagnosis

Having detected a discrepancy between predictions and observations, our system searches for the original possible cause(s) and elaborates a list of potential diagnoses. A diagnosis is a minimal set of components for which the invalidation of the normal behaviour assumption yields *(SD, COMP, OBS)* consistent, where *SD* is a formal description of the system including assumptions of normal behaviour for the set *COMP* of components and the components in *COMP* are the elementary diagnosis units. In the CA-EN diagnosis approach, the causal graph acts as the *SD* and the influences themselves are the elements of *COMP*. Faulty influences are internally turned back to faulty components. Diagnoses are computed from the collection of *conflict sets*, i.e. sets of components such that the observations indicate that at least one of the components in a set must be behaving abnormally, by an incremental hitting set algorithm. They are given as sets of faulty components labelled by their corresponding time of failure.

The diagnosis process is initiated as soon as a variable is reported as misbehaving. For this variable, say *X,* the conflict generation procedure traces backward in the causal graph, following the intuition that the influences which may be at the origin of the misbehaviour of *X* are those related to the edges belonging to the paths going from the measured variable nodes to the node of *X*. This set of influences, called the *ascendant influences* of *X*, is recorded as a conflict set.

Another type of conflict set can also be outlined by accounting for the non-misbehaving variables. Knowing that some variable is not misbehaving may be very informative for refining the diagnosis. However, not all the non-misbehaving variables are useful; intuitively, only the variables which have at least one causally upstream unmeasured variable in common with a misbehaving variable must be considered. Lets call *Y* such a non-misbehaving variable and *X* such a misbehaving variables. Then, the symmetric difference of the set of the ascendant influences of *Y* and that of *X* is also a conflict set. This proposition can be extended for the case in which there are more than one misbehaving variables, say a set *S*, which share a causally upstream unmeasured variable with the non-misbehaving variable *Y*. A conflict set can be generated for every subset of *S*, as the symmetric difference of the set of the ascendant influences of *Y* and that of this subset.

The conflicts obtained by tracing back in the causal graph from misbehaving variables are called *hard conflicts* whereas the conflicts accounting for non-misbehaving variables are called *soft conflicts*. The reason for this explicit distinction comes from the fact that our prediction procedure gives complete but not correct predictions. In consequence, when a variable is detected as misbehaving, there is no doubt about it and the corresponding conflict is said to be « hard ». Now for non-misbehaving variables, although their observed value is within the envelope of predicted values, this envelope may include some spurious values and it might be that the variable is actually misbehaving. The probability of this being so is however very low. This is why we still generate a conflict but we distinguish it as « soft ». As will be explained later, this distinction provides the basis for defining a preference criterion for the generated diagnosis.

The diagnosis generation is based on generating the minimal hitting sets of the collection of conflicts generated by the above algorithm. As new symptoms for a given fault can appear across time, it is important that the diagnosis procedure is incremental. In CA-EN, we use Levy's algorithm [18] which is an incremental revised version of Reiter's original one [7]. The diagnoses are classified according to whether their elements belong solely to hard conflict sets or not. Let us define a *soft conflict element* as one that belongs to a soft conflict and does not belong to any hard conflict. Then, the *preference class* of a diagnosis set is defined as the number of soft conflict elements that it contains. The smaller the preference class of a diagnosis set, the more the concerned diagnosis is preferred.

As an illustration, consider the FEP nozzles system for which the causal graph is the following :
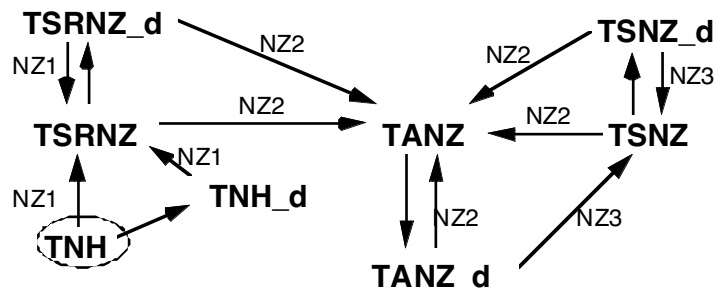
Causal Graph Of The FEP Nozzles System

The diagnosis conclusions provided for the nozzles sticking scenario outline the following fault messages which show the misbehaving variables as well as the proposed diagnosis candidates.

```
FAULT MESSAGES
1     Time 18:36:42
      Variable tanz misbehaves
      CURRENT DIAGNOSIS
            {tanz sensor, nozzles servo, nozzles actuator}

2     Time 18:37:09
      Variable tsnz misbehaves
      CURRENT DIAGNOSIS
            {nozzles servo, nozzles actuator}

      END OF DIAGNOSIS
```

As the servo and the actuator run in closed-loop, it is normal that the system cannot discriminate among those two components. In this real application example, the preference class of both the generated diagnosis at step 1 and at step 2 are zero.

4.   DISCUSSION

Perhaps the key benefit of TIGER is that it acts as a gas turbine engineer monitoring the system every second, 24 hours a day, every day of the year. Most engineers make a regular check of the operation of the turbine ensuring that key variables are within their limits and looking at trends and recent patterns of the turbine to ensure that everything is okay. TIGER performs this task automatically, thus being able to detect problems at an early stage before they become significant. This provides a revolutionary change in how a company can monitor their turbine and opens up a much more precise level of equipment management.

Further benefit results because TIGER is able to show the trend of parameters of short or long periods of time, enabling more accurate condition monitoring. The TIGER system captures the key data for an incident and allows the engineer to go back and examine this in more detail.

The TIGER system monitors the dynamic response of the system. This is a unique capability. This can be used to ensure that the servo actuator systems are reacting properly. It can also be used to ensure that the control system, and more importantly, the key transducers affecting the control system are correct. It has a very powerful user interface environment to develop graphical displays of the state of the turbine giving a quick look analysis of the state. It supports a fault log and data log to enable long term analysis of problems.

Given that the focus of this work and publication is the artificial intelligence aspects, it is interesting to think about the question of how much of the benefit derive from the artificial intelligence aspects as compared to more mundane computer software capabilities. It is not easy to make a percentage split in the benefit from the artificial intelligence techniques relative to other software techniques. The provision of benefits by a software package is sometimes more a binary concept, either it has no benefit or it is highly beneficial. For TIGER it is highly beneficial because of the combination and the sum of all the functionality provided by the system. It is the range of functionality provided to the user and the integration of that functionality that makes TIGER so useful.

During the development period, considerable thought was given to the wide range of functionality that a user would need. Traditional software such as the user interface and trending represent vital aspects of the system, and on their own could justify the use of TIGER. However, the real

achievement of the step change in how turbines can be monitored, is by also having the addition of the fault detection and diagnostic components, but these alone would have no value without a properly integrated environment for the user. No one component, including the various AI fault detection and diagnosis modules would provide the benefit of TIGER, but each component is an essential part of the whole. This is consistent with the feedback received during the many demonstrations of TIGER to gas turbine engineers. They see the fault detection and diagnosis modules as very exciting and highly useful, but without the other functionality they would not be interested in the system.

Related to this, it is interesting to analyse the type of supplementary work implied by the qualitative model based fault detection and diagnosis component. Putting aside the effort required by the algorithms themselves which can be viewed as general fault detection and diagnosis engines, let us examine more closely the model building task. How are the models built? Can part of the work be automated? How much of the models are reusable?

CA-EN models can be built out of deep knowledge and/or empirical knowledge based on the physics underlying the behaviour of the physical system. Empirical knowledge should state cause-effect interactions among variables (e.g. when the fuel stroke reference increases, then the first stage shaft speed rotation increases as well) supplemented by order of magnitude knowledge (e.g. increases much more or increases by a certain numerical coefficient) as well as temporal knowledge like the delay of propagation and response times which are not necessarily exact but can be given as intervals. This is easily implemented at the causal level. Deep knowledge can be in the form of algebraic equations (linear or non linear), first order linear differential equations or recurrent equations (linear or non linear). This needs to be implemented both at the global constraint level and at the causal level. The difficult part of translating analytic knowledge into causal relations is performed automatically by a program called CAUSALITO.

CAUSALITO uses causal ordering concepts for automatically generating the CA-EN causal graph as well as some of the influence attributes from a set of equations [13]. Compared to other methods, the algorithm implemented in CAUSALITO has the advantages that it can deal with highly interconnected sets of equations and handle systems with multiple operating modes thanks to an incremental approach.

Let us now analyse where the knowledge came from for the two applications in the TIGER project. In the case of the Exxon turbine application, it came from two sources, the controller (Speedtronic Mark IV) documentation on the one hand and recorded data of the turbine in real operation on the other hand. The first source provided equations relating control and controlled variables with precise structure and parameter values. The second source provided relationships whose structure and parameter values could be automatically obtained by identification and parameter estimation methods (we used the System Identification toolbox MATLAB [14]). Parameter values were estimated as interval values to account for imprecision, noisy data and dispersion. The set of relationships came out in the form of a set of algebraic (linear and non linear) and recurrent (linear) equations.

The knowledge available for the other application, the Auxiliary Power Unit (APU), came from completely different sources. It was provided by the experts who specified the turbine on the one hand and the experts who designed the turbine on the other hand. The knowledge available was then in the form of:

- relations derived from physical laws (flow conservation law, pressure decrease through a valve, etc.).

- numerical knowledge: several parameters of the relations were numerically specified within a tolerance interval.

- characteristic graphs recorded on the test bench.

One of the difficulties here came from the fact that the knowledge in the form of equations was often too precise (and therefore mathematically complex) for building a model dedicated to diagnosis. Several meetings were necessary with the experts to decide, for every component, on the phenomena which could be neglected and to obtain appropriate approximated equations. Then, numerical intervals had to be estimated for the non specified parameters in the equations. This was done by using the recorded data. The whole analysis phase required deep involvement into the electro-mechanics of the turbine domain. However, it resulted in a significant set of reusable generic models.
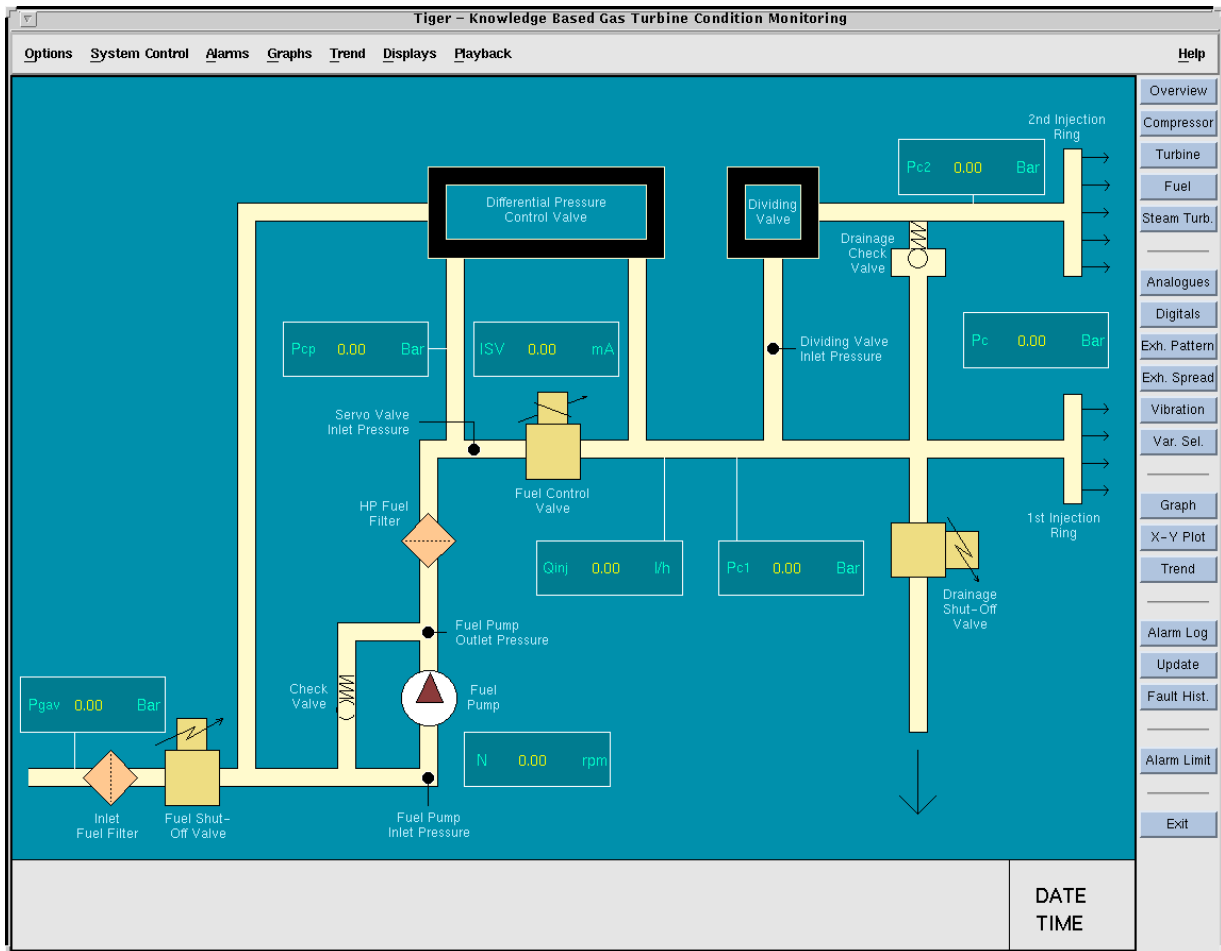

Figure 6: Auxiliary Power Unit Block Diagram

For example, the APU fuel system shown in Figure 6 was composed of ten hydraulic components of five different types. Generic behavioural models were built for the different types of hydraulic components and the general laws applying to this class of systems were outlined. The assembled APU fuel system equation model contained 22 equations for 4 input variables and 18 internal variables (8 measured variables and 14 non-measured ones), and 22 constant parameters[2]. The CA-EN causal graph which was generated automatically by CAUSALITO consisted of 43 interconnected influences [15].

---

[2] *This model had a restricted zone of validity in the neighbourhood of a stabilised pump speed of 6100 rpm and for the following environmental conditions: pressure P0=1 bar, temperature T0=15°C, altitude Z=0m.*

We found two major difficulties in the model-based approach that we believe are not specific to our applications but rather related to the complex dynamic systems domain in general. The first one is that the knowledge available is not necessarily at the right level of abstraction and that it might be difficult to abstract it without falling into too trivial of models. The second one is that most of the deep knowledge concerns nominal operating conditions. This means that it is extremely difficult to get a model which would be valid over a wide operation range (for the TIGER turbines, this meant that the model might not be usable to monitor the starting up phase) and even more difficult to get realistic fault models (this is why our method relies on normal behaviour models only).

The TIGER system is continuing to evolve. Although TIGER is a framework for monitoring a wide range of dynamic systems and also has the diagnostic core for a wide range of gas turbine models, the major focus of TIGER after the initial developments were completed was a version packaged as a complete solution for monitoring General Electric Frame 6 gas turbines. This version uses over 500 inputs defined as a standard data input file. It includes a more detailed set of diagnostics for Frame 6 gas turbines used in power generation. It also has a more extensive set of on-line manuals and supporting displays. Because of the standard configuration of GE Frame 6 gas turbines for power generation, this is now a standard product that can be installed very rapidly on a gas turbine that has not been monitored before. This is consistent with our thinking that the best way to make applications successful is to focus on providing a complete vertical solution that is easy to install and to use.

5. ACKNOWLEDGEMENTS

6. REFERENCES

[1] Milne, R. et. al. '*TIGER: real time situation assessment of dynamic systems*'. Intelligent Systems Engineering Journal, vol. 3, no. 3, Autumn 1994.

[2] Milne, R. '*On-Line Diagnostic Expert System For Gas Turbines*'. ES'92 Conference, Cambridge, 15th-17th December, 1992.

[3] Milne, R. et. al. '*TIGER: Knowledge Based Gas Turbine Condition Monitoring*'. AI Communications Journal, pp1-17, Vol 9, No. 3. Published by IOS Press, September 1996.

[4] Ghallab, M., & Philippe, H. '*A Compiler for Real-Time Knowledge-Based Systems*'. Proc. IEEE International Symposium on AI for Industrial Applications, 287-293, 1988.

[5] Dousson, C., Gaborit, P., & Ghallab, M. '*Situation recognition: representation and algorithms*', Proc. 13th IJCAI, Chambery, France, 1993.

[6] Bousson, K., & Trave-Massuyes, L. '*Fuzzy Causal Simulation in Process Engineering*'. IJCAI-93, Chambery, France. August-September 1993.

[7] Reiter, R. '*A Theory of Diagnosis From First Principles*'. Artificial Intelligence 32, p 57-95, 1987.

[8] Trave-Massuyes, L. "*Qualitative Reasoning for Dynamical Systems Simulation*". Systems and Control Encyclopaedia (Ed. M. Singh), 2nd Supplementary volume, Pergamon Press, 1992.

[9] Trave-Massuyes, L. & Milne, R. '*Diagnosis of Dynamic Systems Based on Explicit and Implicit Behavioural Models: An Application To Gas Turbines In ESPRIT Project TIGER.*

Applied Artificial Intelligence, Volume 10, No. 3, pp257-277.   Published by Taylor & Francis. May-June 1996.

[10]   Mavrovouniotis M.L. & Stephanopoulos G. *'Formal Order Of Magnitude Reasoning In Process Engineering'*, *Computer Chemical Engineering,* 12, pp. 867-880.

[11]   Bousson, K. & Trave-Massuyes, L. *'Putting more numbers in the qualitative simulator CA-EN'*. Proceedings of the Second International Conference on Intelligent Systems Engineering, 5-8 September, 62-69, Hamburg, Germany, 1994.

[12]   Hyvönen E. *'Constraint Reasoning With Incomplete Knowledge; The Tolerance Propagation Approach'*. Doctoral Dissertation, Technical Research Center of Finland, VTT Publications, Espoo, Finland.

[13]   Travé-Massuyès L. & Pons R. *'TIGER Deliverable D320.36, Methodology Report-Models 2'*, ESPRIT Project 6862: TIGER.

[14]   Ljung, L. '*User guide of system identification toolbox for use with Matlab'*, The ManWorks, Inc., 1991.

[15]   Travé-Massuyès L. & Escobet T. *'TIGER Deliverable D523.36, APU Fuel System Model'*, ESPRIT Project 6862: TIGER.

[16]   Bousson, K., Zimmer, L. & Trave-Massuyes, L. '*Causal Model-Based Diagnosis of Dynamic Systems'.* In DX-94 Conference.

[17]   Bousson K., Travé-Massuyès L. & Zimmer L. *'TIGER Deliverable D330.36, Methodology Report-Diagnosis Methodology 2'*, ESPRIT Project 6862: TIGER.

[18]   Levy, F. '*ATMS for default theory'*, Internal Report LIPN n.91-8, Universite de Paris-Nord, Villetaneuse, France.