A Multi-Agent Fault Detection System for Wind Turbine Defect Recognition and Diagnosis

A. S. Zaher and S.D.J. McArthur
Institute for Energy and Environment, University of Strathclyde,
Glasgow, United Kingdom. Email: azaher@eee.strath.ac.uk

***Abstract*-This paper describes the use of a combination of anomaly detection and data-trending techniques encapsulated in a multi-agent framework for the development of a fault detection system for wind turbines. Its purpose is to provide early error or degradation detection and diagnosis for the internal mechanical components of the turbine with the aim of minimising overall maintenance costs for wind farm owners. The software is to be distributed and run partly on an embedded microprocessor mounted physically on the turbine and on a PC offsite. The software will corroborate events detected from the data sources on both platforms and provide information regarding incipient faults to the user through a convenient and easy to use interface.**

I. Introduction

Wind turbines (Wind Energy Converter or WEC) are becoming more established as an economically viable alternative to fossil-fuelled power generation. Currently, wind farms consisting of hundreds of units are being built in various locations around the country adding a significant amount of electrical generating capacity. As the size of wind farms continues to increase, business economics dictate the need for effective condition monitoring (CM) systems that allow for careful asset management to minimise downtime and maximise availability and profits. Current CM systems essentially provide the necessary sensor and data capture capability required for monitoring. The data is then collated and stored via a SCADA (Supervisory Control and Data Acquisition) system which an operator must then examine in order to deduce the overall health of the turbine as well as its internal components. This task rapidly becomes increasingly difficult since the onerous volumes of data can become rather tedious to inspect, meaning faults are not always detected as early as possible.

Early detection of incipient faults within a turbine can prevent major component failures and allows scheduling of efficient condition based maintenance repair strategies. Although modern WEC's have reached high technical standards there is still a strong potential for further development especially with large megawatt size machines, where enhancements in availability, reliability and overall lifetime are all viable factors for improvement.

Online fault detection/diagnosis systems (FDS) offer an opportunity for improved maintenance and failure prevention strategies. They allow for early warnings of mechanical and electrical defects to prevent major component failures and to keep side effects on other components low. In this way they can be used to estimate the state of degradation or remaining life of a specific component in a WEC. Therefore replacement of parts that are still operational is avoided without the risk of unforeseen breakdown.

This paper will describe the development of a FDS which aims to complement the original CM approach already in use with WECs. It will make use of a combination of anomaly detection and data trending techniques that attempt to provide early fault detection and diagnosis.

It is believed that anomaly detection will provide detection of the fault as early as possible, while minimising the dependence on information of how defects are reflected in the data. The alarms raised by the incipient anomalies detected can easily be traced back to the data parameter in which it was identified, leading to an opportunity for the identification of the degrading component in time before it fails and is required to be replaced. Finally, a front end will gather and present the output of these detection modules in a comprehensible manner to wind farm operators.

II. Requirements of a Fault Detection System

The general functional requirements of a FDS can be summarised by the system having the following capabilities [1].

- Automatic capture and conditioning / formatting of relevant data
- Automatic interpretation of the conditioned / formatted data, to identify incipient and serious defects
- Discrimination between a sensor failure and an actual plant failure
- Provision of clear and concise defect information and advice supplied to the users
- Extensibility and flexibility to include further interpretation techniques and monitoring technologies.

In the research literature, Artificial Intelligence (AI) techniques have grown more popular in CM systems over recent years [1-6]. The process of CM has generally become far more complex with constant advances in sensor technology. A number of AI techniques can be used to model or evaluate relations between the measured signals and current operating conditions of machinery.

Within the context of FDS, the main use of AI techniques is the interpretation and diagnosis activity mentioned above in the requirements. As can be expected some techniques are better suited to identifying specific failures more so than others, and so in any one system, a combination of techniques are used in order to interpret the data from the sensor streams. This essentially leads to a hybrid architecture [1] shown in Fig. 1, which results in a process where the proposed conclusions from each analysis require rationalisation into one single conclusion. Systems that incorporate AI techniques are often referred to as intelligent systems.

In order to develop such a system, a framework which is capable of integrating multiple data sources and interpretation techniques is required. Multi-agent systems (MAS) provide a flexible and extensible structure for designing these systems, allowing different tasks to be encapsulated in separate modules (agents). The flexibility and extensibility of this framework allows new agents to be introduced into the system in real-time i.e. without having to stop and start all of the other agents that are running. A “yellow pages” like facility known as the directory facilitator or DF allows each agent to register with it the services it provides along with the information or services it is interested in. In this way each of the other agents can learn about the newly introduced agent’s existence and the services it can provide that may be of use to them. This proves extremely useful in the development of a FDS, especially if new diagnosis techniques are later discovered to be of benefit if they were to be added to the system. MAS also permit the development of more intelligent and automated diagnostic functions given that each agent can independently exhibit the following characteristics: social ability, reactivity and pro-activeness [7]. Social ability means that each agent can communicate and cooperate with other agents, supporting data and information exchange. Reactivity and pro-activeness suggest that agents have the ability to react to their surroundings and pro-actively take the appropriate action required to solve any problems they encounter.

The number and variety of sensor technologies applicable for turbine monitoring is somewhat diverse [8]. Modern turbines typically utilise a number of different types of sensor which inevitably leads to a dramatic increase in the volume of data collected. Successful interpretation of this data is predicated upon knowledge and understanding of the links between actual defects and the data gathered as a result. If this understanding is non-existent then often detailed data mining and analysis activities are required to derive the relationships. As a result of this, existing CM approaches require one of two prerequisites to ensure their success [3].

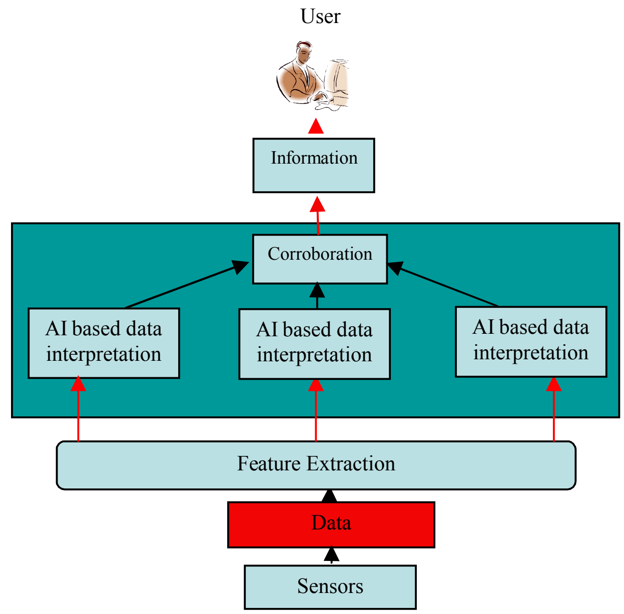

Fig. 1. Hybrid architecture using multiple techniques

1. detailed knowledge of how defects manifest themselves in monitored data; or

2. The availability of extensive historical data along with records of the actual defects in order to allow data mining activities to be used to extract defect knowledge for CM.

Since this information is not always readily available, a selection of techniques that do not depend on the presence of this knowledge would be extremely useful. This factor essentially dictated the detection techniques proposed for the system.

III. SYSTEM ARCHITECTURE

This research is part of a UK EPSRC funded consortium project (“PROSEN”) in which multiple institutions are taking part. The aim of the project is to enhance overall monitoring methods through the development of a monitoring system that can be retro-fitted to mechanical machinery. This monitoring system consists of wireless sensors that communicate with an embedded microprocessor mounted on a FPGA. Its purpose is to allow diagnostic and fault detection algorithms to be deployed down at the sensor / hardware level. This will decrease the volume of data that must be transmitted and stored via the more traditional centralised SCADA system approach. The results output from the algorithms will already provide preliminary information regarding the state of the components meaning only this data has to be sent instead of the actual raw data for each of the monitored components.

It is hoped that this system will actually be fitted to an operational turbine, however for testing purposes it will be used on a number of bearings which will be purchased for initial experiments. To put this into perspective for the FDS, there will be two main sources of data that it must take into account.

1. The traditional 10 minute averaged data acquired from the SCADA systems which adopt the centralised approach to collating and storing data from all of the turbines present in the wind farm in a database; and
2. The more frequent raw data collected via the microprocessor / sensor system physically retro-fitted to the turbine.

Both of these data sources reside on two separate platforms, distinguished by where they exist physically. The SCADA data resides on a server / PC in the form of a database, and the raw data resides on the hard-drive of the microprocessor / sensor system mounted on the turbine. The important aspect to note here is the fact that these data sources are independent of one another and will be referred to as the *high-level* and *low-level* platforms respectively for the remainder of this paper. The details of the microprocessor system and its internal workings are out with the scope of this paper and it is suffice to say that it will be capable of running computationally light software i.e. utilising minimal memory and processing in order to limit battery consumption. There will also be some form of communication link that will enable simple messages to be relayed between the two platforms. The number of messages that can be sent however is again restricted due to the power hungry nature of communications.

In order that the FDS utilises both the sources of data for its processing and analysis, it must therefore be partially distributed over the two platforms, corroborating events that are identified in the *low-level platform* with those identified in the *high-level* via the messages that can be sent between them. This corroboration will offer a more robust and accurate fault detection since similar faults that are identified in both data sources will offer greater confirmation and reassurance that the fault detected is genuine.

The overall system architecture is shown in fig 2. The data to be utilised along with the processing and its outputs is shown for each platform. The data for the *low-level* platform is yet to be acquired and so the software for this platform therefore will be developed at a later stage. Once this is available however, a similar process of learning normal behaviour models for the bearings will allow deviations to be detected. Diagnosis of specific faults will aim to be achieved through the process of destructive testing and recording of the results. This will allow for the acquisition of fault signatures which can be used as a reference for classifying the different bearing faults.

IV. HIGH-LEVEL SOFTWARE ARCHITECTURE

The *high-level* software is being developed using a Multi-Agent framework. The system will be composed of five types of agents consisting of three detection modules, an interface agent with which the user can interact and receive information and, finally data parser agents to handle the data.

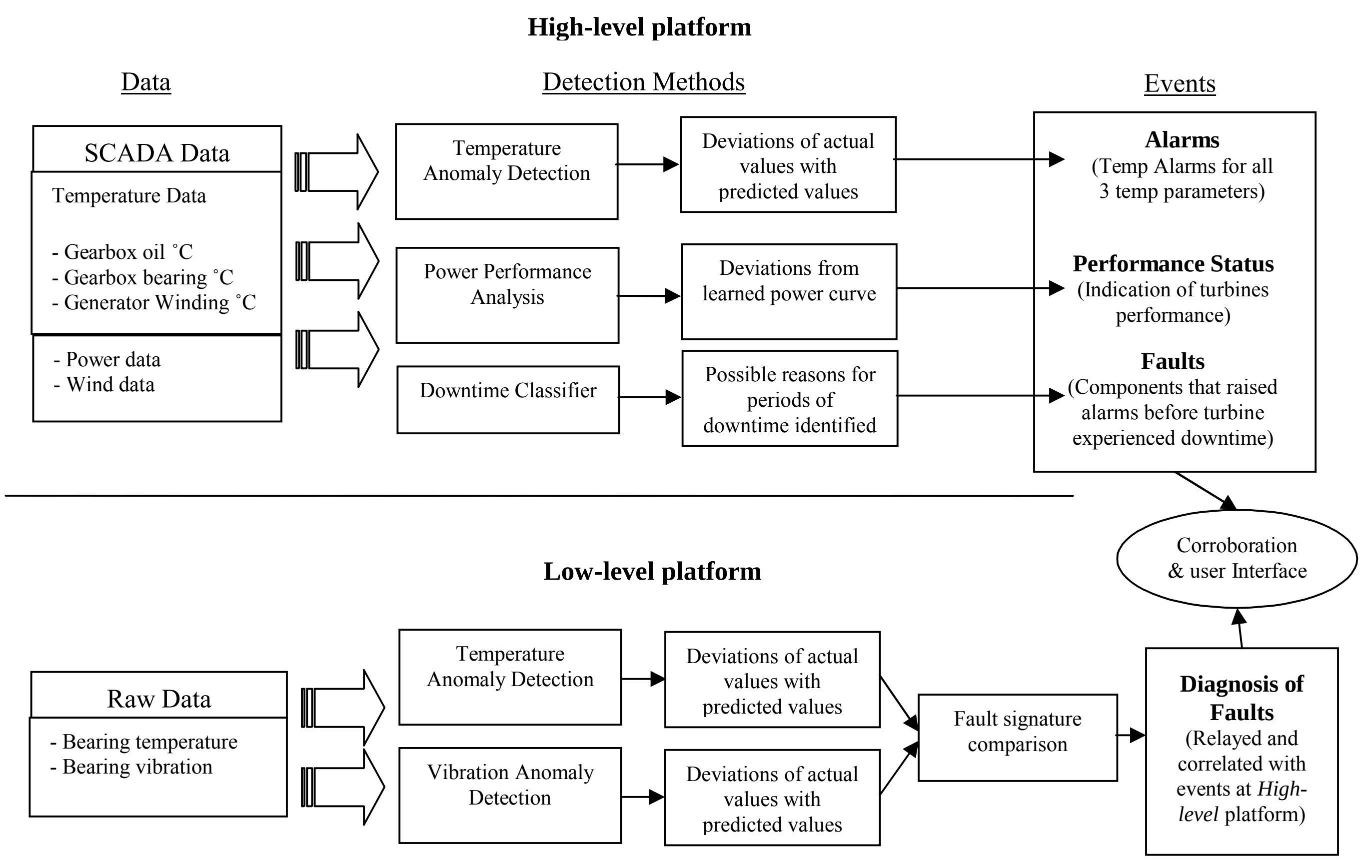

Fig. 2. Overall System architecture

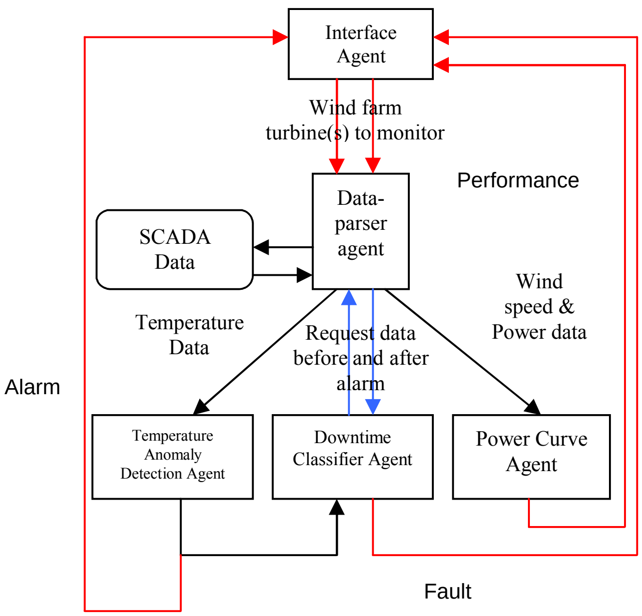

Fig. 3. High-level software agent architecture

In order to simplify the agent architecture diagram shown in Fig 3. only one data parser agent is shown; however a separate agent will be used to parse the different data parameter types stored in separate files leading to a total of five data parser agents for the five types of data to be analysed by the *high-level* system. By simply adding in a new data parser agent, this gives the system the capability of reading data files of different formats from different wind farms.

The purpose of the interface agent is to allow the user to interact and view the information presented by the system. It will also provide the user with the option of selecting the turbines and alarms they are interested in. This will give the user the control of further reducing the volume of data presented to them by selecting only what is important to them.

The three detection agents are composed of two anomaly detection modules and a downtime identification module. Section V goes on to describe the internal workings of the detection agents.

In order that the agents can exchange information and data with one another, a language supporting the concepts and vocabulary which they share in common must be designed. This is known as an ontology. An ontology defines the concepts and ideas that the agents communicate between them. To design an ontology suited to any particular domain, thorough knowledge of the sort of messages and information that will be exchanged between the agents is necessary.

Protégé v3.3 beta was used to create and design the ontology. Fig 4. shows the class browser view with the hierarchy of concepts determined. Concepts that extend from other concepts inherit the attributes of its parent concept, for example the individual temperature types all have in common the fact that they are temperatures. The difference is in which component of the turbine the temperature is associated with. Therefore, the individual temperatures would have one extra attribute detailing the component which the temperature reading belongs to. A turbine is composed of the three main components, the gearbox, generator and rotor blades. In this way a temperature reading for a specific component can be traced back to the turbine which it belongs to. This gives an agent an “understanding” of the significance a simple numerical value holds. The existence of these concepts also gives the agents within the system a common ground regarding the information they can send, request or query.

V. Detection and Diagnosis Techniques

There are three main components of a wind turbine whose failure leads to a very costly and lengthy period of downtime simply due the impracticalities of their replacement. These are the gearbox, the generator and the rotor blades. Monitoring the health of these components and maximising their lifetime potential is vital in order to increase reliability, availability and decrease overall maintenance costs of the turbine. The FDS described will focus on the monitoring of these components only.

A. Anomaly Detection agents

Anomaly detection, as the name suggests, is a technique that uses the profile of normal behaviour expected from some piece of machinery or system for comparisons in order to detect anomalous behaviour that may become apparent in the system. Any deviation from the normal profile or normal behaviour model as they are sometimes referred to is flagged as an anomaly. The normal behaviour profile is learned over a

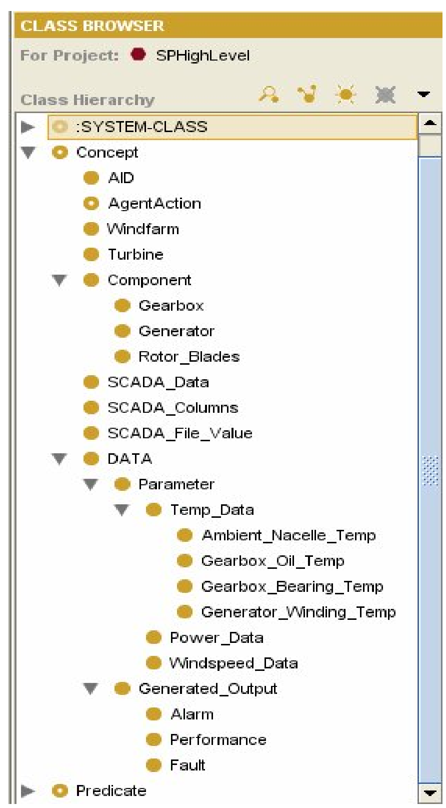

Fig. 4. The ontology viewed in the class browser created in protégé 3.3

period of time. This technique has the potential to detect all kinds of failures since it assumes all deviations from the normal as errors or problems manifesting themselves and proves helpful in cases where there is no underlying information of the kind of data expected to be generated by the system. The two modules to be used are described below:

1. *Power Curve Agent.*

The power characteristic is essentially the turbines' typical power output values plotted across a range of the wind speeds that generated those power values. This is also known as the power curve of the turbine due to the shape the graph forms. By keeping track of both wind speed and power output parameters, the overall health of the turbine can be supervised. Each turbine model has a set power output performance level which it is expected to sustain in its everyday normal operation. By comparison of the actual power output and the expected output, an indicator of the turbines' generation performance is feasible. Any degradation in this value will automatically reflect degradation in the turbines' components. References [9] & [10] State that degradation in this value reflects deterioration in the turbines' rotor blades. This can be due to many factors ranging from pollution and dirt building up on the surface of the blades to mass imbalance of one blade leading to increased oscillation and vibration of the tower, which again in turn can lead to a reduction in the power output. Aside from this, its main purpose in this FDS is to provide the user with an indicator as to how well the turbine is performing. Used in conjunction with the *temperature anomaly detection* agent described below, it will also serve the purpose of determining the time it takes from when an anomaly is detected in one of the turbines' internal components (i.e. the generator or gearbox) to the time it takes for this to be reflected in the turbines' overall performance. This will give an indication as to how long it takes faults to manifest in the turbine to the point where they actually begin to impair the turbines generation ability.

The task of the power curve agent is to learn the turbines' averaged power curve under healthy operation. This is carried out by selection of data from a number of healthy turbines. The wind speeds are separated into groups of 0.5m/s. The associated power values are then sorted into these groups. The average power value for each group is then calculated to give the learned power curve. Fig 5. shows the learned power curve as well as the inner and outer alarm limits. The inner alarm limits are calculated through the standard deviation of each of the groups and then added to either side of the averaged power curve. The outer alarm limits are chosen by the developer through the study of a number of turbines operating under normal conditions. A variety of methods for calculating alarm limits will be tested in order to observe a strategy that minimises or eliminates any false alarms generated. Performance alarms will only be generated if a number of points consequently cross the alarm thresholds.

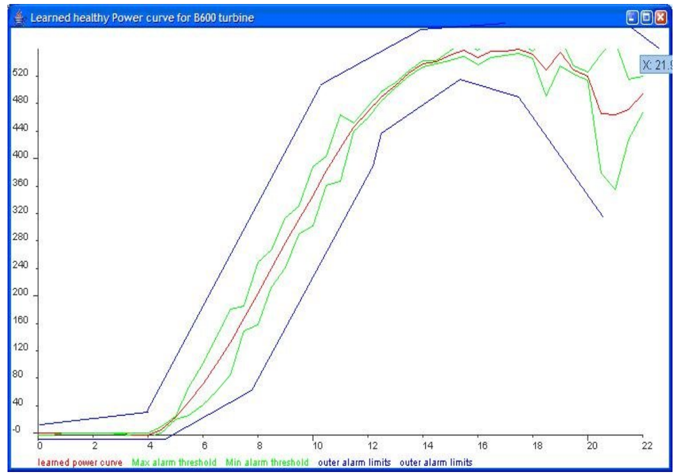

Fig. 5. Learned average power curve with alarm limits for a Bonus 600KW turbine

2. *Temperature Anomaly Detection Agent*

This agent will employ normal behaviour models [5] in order to predict the "normal" behaviour expected for each of the monitored data types according to its current working and environmental conditions. This is achieved through the use of a back propagation neural network. Neural networks have the ability to model dynamic non-linear processes and work well in situations where there is no algorithmic solution but plentiful examples of the kind of behaviour required. It exploits a supervised learning process where it will be fed the current power output and wind speed as inputs and given the three associated temperatures in the SCADA data as an output to learn. In this way correlations can be made between the level at which the turbine is generating and the expected temperatures of its components for that level of output. When training a neural network it is necessary to consider the number of inputs and outputs in the training data set when determining the number of nodes in the hidden layers in the network [11]. Typically networks consist of 3 layers, an input layer, output layer and a hidden layer. If the number of inputs and outputs are known then the number of hidden layer nodes can be calculated using the given equation (1).

$$J = \sqrt{IK} \tag{1}$$

Where I is the number of network inputs, J is the number of hidden network nodes and K is the number of network outputs. According to this equation with 2 inputs and 3 outputs the number of hidden layer nodes should be 2.449, this means either 2 or 3 nodes should be used in the hidden layer.

Once the models are built, they can be used in real-time to predict future temperature values for the components. In this way anomalies can be detected by means of a comparison between the predictions of the model and the actual values being monitored. Any anomalies detected will raise alarms of differing severity dependent on the extent of deviation from the normal behaviour. These alarms will be passed on to the downtime classifier agent and also the interface agent if the user so wishes to be notified of alarms of a particular severity level.

B. Downtime Classifier Agent

The downtime classifier agent will serve two main purposes. Its first use is essentially the output of its main data analysis operation. It is activated upon receiving alarms from the temperature anomaly detection agent. Once an alarm is received it identifies downtime periods by monitoring the active power output trends of the turbine. If a period of downtime is identified, an attempt to provide some form of justification for each period found can be established through the information provided in the alarms and also by analysing the associated temperature trends of the components. Any unexpected peaking of a component's temperature before the period of downtime is a good indicator of which component was responsible for the experienced downtime. This processing also proves beneficial since it can help towards a better understanding of how defects manifest themselves in the data. The second use for the output is to build a probability distribution of the various downtime periods and utilise this information for developing a model of the turbine's operational behaviour.

VI. Conclusion

This paper has illustrated the architecture of a fault detection system which adopts a selection of techniques that can be used for the identification / detection of incipient failures in turbines. The notion of a distributed FDS has been presented along with the benefits that are to be gained from this particular approach. At present the high-level system is currently under development using a multi-agent framework. Multi-agent systems prove to match the requirements of an FDS detailed in section II and so have been justified as a suitable framework for development. The specific agent architecture along with the internal workings of each detection method has been discussed and commented as well as the method of communication between the agents.

Future work will consider the implementation of both platforms and how they will corroborate their results. The system will then be tested in order to quantify its success at fault detection. The extensibility of the system also allows for the possibility of developing further diagnosis methods for example gaining access to expert knowledge from industrial partners may be captured and encapsulated in a knowledge-based agent at any later stage.

VII. References

[1] S. D. J. McArthur, S. M. Strachan, and G. Jahn, "The Design of a Multi-Agent Transformer Condition Monitoring System," *IEEE Trans. Power Syst.,* vol. 19, no. 4, pp. 1845-1852, Nov. 2004.

[2] S. D. J. McArthur, V. M. Catterson, and J. R. McDonald, "A Multi-Agent Condition Monitoring Architecture to support Transmission and Distribution Asset Management," in *3rd IEE International Conference on Reliability of Transmission and Distribution Networks 2005,* London, UK, Feb. 2005.

[3] McArthur, S.D.J. Booth, C.D. McDonald, J.R.; McFadyen, I.T, An agent-based anomaly detection architecture for condition monitoring," *IEEE Trans. Power Syst.,* Volume 20, Issue 4, Nov. 2005 pp:1675 - 1682

[4] Jason R. Stack, *Member, IEEE,* Thomas G. Habetler, *Fellow, IEEE,* and Ronald G. Harley, *Fellow, IEEE,* "Fault Classification and Fault Signature Production for Rolling Element Bearings in Electric Machines", IEEE Transactions on industry applications, Vol. 40, NO. 3, MAY/JUNE 2004

[5] Garcia, Mari Cruz (Universidad Pontificia Comillas, Instituto de Investigacion Tecnologica, IIT); Sanz-Bobi, Miguel A.; del Pico, Javier "SIMAP: Intelligent System for Predictive Maintenance. Application to the health condition monitoring of a windturbine gearbox," in *Computers in Industry*, v 57, n 6, August, 2006, p 552-568.

[6] J. Yang, M. Montakhab, A. G. Pipe and T.S. Davies, "Application of Multi-Agent Technology to Fault Diagnosis of Power Distribution Systems", in *Proceedings of Engineering of Intelligent Systems (EIS 2004)* Feb 29 – Mar 2 2004,

[7] M.J. Wooldridge and N. R. Jennings, "Intelligent agents: theory and practice," *Knowl. En. Rev.,* vol. 10, no. 2, pp.115-152, 1995.

[8] Y. Han, Y.H. Song "Condition Monitoring Techniques for Electrical Equipment—A Literature Survey", in *IEEE Transactions on Power Delivery*, Vol 18, NO.1 January 2003

[9] P. Caselitz, J. Giebhardt, "Rotor Condition Monitoring for Improved Operational Safety of Offshore Wind Energy Converters", ISET in *Journal of Solar Energy Engineering* -- May 2005 -- Volume 127, Issue 2, pp. 253-261

[10] Caselitz, P., Giebhardt, J., and Kewitsch, R., 1999, "Advanced Condition Monitoring System for Wind Energy Converters," in *Proceedings of the EWEC* 1999,

[11] Tarassenko, L., "A Guide to neural Computing Applications", ISBN 0 340 70580 2, Arnold, 1998.