Proceedings, 16th International ISSAT on Reliability and Quality in Design, 5-7 August, 2010,ISBN: 978-0-9763486-6-5.

# A Review of Prognostics and Health Monitoring Techniques for Autonomous Ocean Systems

Randall Wald, Taghi M. Khoshgoftaar, Pierre-Philippe Beaujean, John C. Sloan
Florida Atlantic University, Boca Raton, Florida, USA
Contact author: khoshgof@fau.edu

Keywords: prognostics and health monitoring, machine condition monitoring, autonomous ocean systems, neural networks.

***Abstract*—Prognostics and health monitoring (PHM) is an important and growing focus in the design and maintenance of complex systems. It is being applied to a wide range of problems, from industrial machines to avionics systems to batteries. In this paper, we review existing work in the context of a novel application: repair and maintenance of autonomous ocean systems, one example being a fleet of ocean turbines. Various approaches to PHM are considered, including model- and data-driven techniques as well as hybrid strategies.**

## I. Introduction

Increasingly complex equipment is used in a variety of fields. These machines and systems are composed of many parts, often in a nested hierarchy. There are many reasons to monitor these systems: to verify that they are performing their tasks properly, to recognize faults before they can cause severe damage, and to predict faults which may arise in the future. Unfortunately, the complexity of these systems makes it impractical for humans to perform these monitoring tasks. There are too many separate subsystems to examine, and it may not be obvious what changes are predictive of future problems. To resolve this, prognostics and health monitoring (PHM) systems exist to computationally assess the state of a system and determine how it will evolve.

PHM can be divided into four main stages [1]: identifying that a system is in a faulty state, determining what component has a fault, determining the exact nature of the fault, and predicting future faults. These are called fault detection, fault isolation, fault identification, and fault prognostics. The first stage determines whether the system is currently working properly. While in principle this can be determined by any PHM system, most implementations employ a separate, dedicated system for this stage, to allow for better responsiveness in the event of a fault. The next two stages, collectively known as fault diagnostics, require ascertaining the current state of the system: finding not only whether or not it is healthy, but what specifically is making it not healthy. In addition to informing the user about the present state, this information is also necessary for the last stage, prognostics. Without being able to determine the system's current state, there is no way to use sensor data to determine its future state. This last stage is the most difficult. It must predict what will happen to the system in the future, in order to provide the user with warnings and advisories. There are no longer clear answers, only probabilities, so the algorithms must carefully weigh the data.

Prognostics has been applied to a variety of fields, ranging from paper-making machines [2] to metal-cutting machines [3] to avionics [4] to battery health [5]. In general, if a system is composed of many parts, and these parts need to each be operational for the system as a whole to function, PHM can be applied. Further surveys examining different applications of PHM can be found in [1], [6], [7].

One application domain which has not yet been fully explored by PHM research is the field of autonomous ocean systems, such as a fleet of submerged turbines. The closest well-studied fields are nuclear submarines (sometimes called "electric boats") and wind turbines, but there differ from unattended submerged systems in important ways. While nuclear submarines (and oceangoing vessels in general) face some of the same potential failure modes, such as bearing fatigue, gear breakage, and corrosion, there are additional challenges with moored unattended systems, including an increased cost in verifying potential problems and the risk of sea life beginning to grow on the equipment (biofouling). Likewise, while wind turbines and ocean turbines both possess a propeller-driven generator, their differences (including a significant difference in typical angular velocity and momentum flux, aside from the obvious effects of being submerged in seawater) require that different PHM solutions be applied.

In this paper, we propose strategies which can be employed to perform PHM on autonomous ocean systems, including a fleet of ocean turbines. In Section II, we outline an approach to choosing a PHM system. In Section III, we apply this approach to our case study, autonomous ocean systems. In Section IV, we review a number of different techniques used to perform PHM on systems in general. In Section V we look at how to combine multiple algorithms into a single system. In Section VI, we examine how these algorithms have been used in the past to optimize maintenance schedules, the specific application we have in mind for ocean systems. Finally, Section VII contains our conclusions and suggestions on specific approaches.

## II. Designing a PHM system

Before designing a PHM system, one must consider all the aspects of the system in order to make the best choice at each stage of PHM design. This problem is dealt with in [7]. Using their procedure, the first step is to determine whether or not PHM is justified for the system in question. If

there is no business case for prognostics, there is no reason to deploy it. Once it is known that PHM is relevant to the system, the designer must find the most severe potential faults in the system and determine the root causes of these faults, employing an understanding of the underlying mechanics of the system. Once these causes are found, the designer must decide if existing sensors are sufficient to detect the warning signs for these causes, or whether physical redesign of the system is necessary to support PHM.

Once the system design itself is finalized, the testing platform must be constructed. A mock-up of the system is used to generate data representing normal and abnormal behavior. Using this data, a model-driven approach (using the lab data to create a physics model or expert system which will examine the real-world data), data-driven approach (comparing real-world data to data generated in the lab and finding differences), or hybrid of the two can be built and tested.

One problem which must also be considered when designing a PHM system is how to evaluate its effectiveness. One approach is a cost-benefit analysis. If the PHM system is employed, how much money will be saved by repairing machines before they become severely damaged, compared with the extra cost of inspecting potentially-healthy systems? By computing the net effect of PHM, one can produce a bottom-line result that is easily understood by all stakeholders [8]. This is not the only possible metric, however; there are a variety of methods to evaluate PHM approaches, including examining the effectiveness of the algorithm as well as how computationally efficient the algorithm is [9]. As in many fields, the choice of metric often determines which algorithm will be declared most suitable.

## III. Case Study: Ocean Turbines

Autonomous ocean turbines face many hazards, including biofouling, saline leakage, and surface damage. Early detection of these problems can allow for quicker recovery or even mitigation. For example, if there is evidence that seaweed or monofilament line is inhibiting propeller rotation, the turbine can be shut down to prevent further damage until a recovery expedition can be dispatched. Thus, the economic case for PHM is clear: detect problems before they cause too much damage.

The primary faults and the sensors needed to detect them have been established [10]. A suite of sensors, including water, oil, temperature, strain, and vibration sensors, has been planned, each able to detect a different set of potential faults. Many of these sensors are somewhat binary in their interpretation: either the pressure vessel is leaking water, or it is not; either the system is within an acceptable temperature range, or it is not. For other sensors, notably the vibration sensors, the interpretation of what constitutes "normal" is more difficult. Nonetheless, this collection of sensors is believed to span all of the major potential faults in the system.

Additionally, a test model of the turbine drive train has been completed. This dynamometer (so called because it is used to test the forces of the running system) consists of a motor connected via a shaft to a generator. The former represents the propellers being driven by the ocean current, while the latter represents the generator of the turbine. This system allows for the simulation of data during the normal state of the turbine, as well as some simulated abnormal data (for example, if the turbine is running faster or slower than it should be). The data from this physical testbed will serve as the basis for further prognostics work: it will show what signals are most indicative of the system's health, as well as allow for a direct test of the appropriateness of different proposed PHM algorithms, whether they be model- or data-driven, to autonomous operation.

## IV. Well-known PHM algorithms

PHM approaches can be divided into model- and data-driven strategies. The former are generally hand-tuned descriptions of the systems, attempts to translate what is known by experts into something a computer can directly apply. These range from fault trees (descriptions of possible faults, what sensors would indicate those faults, and which faults in combination can lead to potential problems with the system), to life cycle load flow charts (state machines describing how degraded a system is and what to do when it fails), to physics models (which incorporate a formalized description of the system and systematically determine the effects of different faults), to full expert systems (with complex rules that attempt to mimic what a human would do in a given situation) [11]. The downside of these approaches is that they must be built by an expert in the field, and at best only match the knowledge possessed by that expert. With especially complex systems, it is difficult to foresee all potential interactions among components, let alone encode those interactions into terms a computer can understand. To avoid these problems, data-driven approaches may be used.

A number of different algorithms have been applied to data-driven PHM. In principle, any existing data mining algorithm may be employed; they all solve the underlying problem of "use a set of data to predict future instances." Not all algorithms function equally well, however. PHM data comes with a unique set of challenges: reliance on a variety of different sensors, each of which might have different weights; examining data across a number of different time scales, from chronic to acute; and varying possible urgencies of response. Neural networks are especially well-suited to manipulating this data, with the multilayer perceptron, radial basis function, and adaptive neuro-fuzzy interface system variants having received particular focus.

### A. Multilayer Perceptrons

Neural networks simulate the function of biological nervous systems, incorporating a network of artificial neurons. Each of these neurons takes values from one or more inputs, applies some function to them, and uses this to produce its output. Typical networks apply appropriate weightings to their inputs, find their sum, and then apply a thresholding function such as a sigmoid. Some variants replace the sigmoid with a radial basis function, or (in the case of adaptive neuro-fuzzy systems) use different functions for different neurons. While in principle a

variety of topologies could be employed, most research uses a feedforward architecture, where neurons are arranged into $n$ layers with the neurons of each layer only receiving inputs from the previous layer and only supplying their outputs to the next layer. Moreover, research shows that for traditional (e.g., threshold-based) networks, using just one or two so-called "hidden layers" (layers which are neither directly supplied with data from the system nor directly used to find the conclusion of the algorithm) are sufficient for most applications. Such networks, sometimes called multilayer perceptrons, have been used in a number of PHM applications [12], [13], [14], [15], [16], [17], [18].

A major challenge with using multilayer perceptrons is selecting a training method. The weights that each neuron assigns to its various inputs are crucial; they determine whether or now the network is able to perform its task. One popular learning algorithm for multilayer perceptrons is called backpropagation [19], [20]. In this technique, the data is first run on an untrained network. The output is not expected to be very accurate, but it is a starting point. The difference between the output generated by the network and the expected output given the input is found; for each output node, this is that node's error. The algorithm then determines how much of each output node's erroneous value came from each of its input nodes; that is, how much each of those nodes is to blame for the error. The output node's error is propagated backwards to these input nodes. These nodes then iterate the process, propagating the error further back into the network until it hits the original input nodes (those which received their data directly from the system being modeled). After each node knows its error, new weights are computed by changing the weights slightly in order to reduce the errors. At this point, the data is run again on the newly-weighted network and the learning continues. The choice of how much to shift each weight per iteration determines the pace of learning, as well as the risk of settling for local minima which are non-optimal solutions.

While backpropagation is a popular learning technique for neural networks, it is not the only choice. Evolutionary algorithms involve randomly generating a population of networks and then breeding them to generate those which have the best performance. This is straightforward if the topology of all the networks are identical since only the weights need to be crossed over when mating the most successful networks. There is, however, a risk that two networks with similar weights but different node orderings will not have meaningful results when bred. More complexity is needed if the topology of the network itself can vary between individuals in the pool, but as this generally pertains to networks more complex than feedforward networks, these variations of evolutionary algorithms aren't widely used in the field of PHM.

### *B. Radial Basis Functions*

Radial basis function networks differ from traditional multilayer perceptrons in that nodes in the single hidden layer use radial basis functions (RBFs) for collecting their inputs and creating their outputs, rather than a sum-and-threshold approach [21]. An RBF is a function which calculates how far each of its inputs is from a chosen "center" value, then sums these "differences from center" results (using their absolute values) and plots this sum on a Gaussian distribution. In particular, if all of the inputs are precisely on their centers, the differences will be zero and the Gaussian (measured at zero) will return a value of one. On the other hand, as the inputs tend to get farther and farther from their centers, the overall difference increases, and the Gaussian will return smaller and smaller values.

In a sense, each of the hidden nodes represents a cluster: inputs closer to the center of that cluster will score higher, while those farther away are lower. Thus the number of hidden nodes is critical: it must be great enough to account for all the expected clusters in the data. One of the benefits of RBF networks is that they allow for simplified training by using a mixed approach. There are only three sets of values which must be trained: the centers of each of the hidden neurons, the width of the Gaussian distributions for each of those neurons, and how much weight each hidden neuron will have when combined to create the final, output neuron. Frequently the widths are fixed, and the centers and weights are trained in separate phases. The centers for the hidden neurons can be found using standard clustering approaches, while the weights for each of the output neurons can be discovered using delta learning, a simplified form of backpropagation applied to a single layer of a network. RBF networks are especially well-suited to working with time series data, and thus may be of particular interest for some classes of PHM.

### *C. Adaptive Neuro-Fuzzy Inference Systems*

Adaptive neuro-fuzzy inference systems (ANFIS) are specialized neural networks designed to couple the expert-system-like flexibility of fuzzy logic with the trainability of neural networks [22], [23]. Fuzzy logic is an extension of fuzzy set theory, which permits elements to exist in more than one set at a time, each element having a confidence factor for membership in any given set. In the context of PHM, this means saying that the system isn't in any one state, but has a certain confidence of being in multiple overlapping states. Depending on the state of the system, one might perform different calculations to find its expected time until failure; ANFIS allow all of these calculations to be performed in tandem, weighted by the confidence in each state. These work by employing a feedforward network with separate hidden layers to perform each stage of calculation. The first layer (which receives its inputs directly from sensor values) tries to determine all of the possible states of the system (possibly employing radial basis functions or other approaches), while the following stages carry out the functions deemed necessary to predict the end value (e.g., expected time to failure) for each of the states. While the downstream neurons may have a variety of functions (summation, product, etc.) based on the recommendations of domain experts, there will often be unknown weights applied to some of their inputs; these can be learned through other neural network approaches. Thus, ANFIS allows for fuzzy logic to be applied with only incomplete knowledge of the underlying system.

## V. Combining PHM algorithms

A wide range of algorithms have been applied to PHM; collectively, these approach the problem of prognostics from a variety of different perspectives. It is not good to rely too heavily on any one strategy. There are always edge cases where that specific approach will fail, while another might have caught a potential problem. Thus, whenever possible, it is good to employ a hybrid approach, running multiple algorithms in parallel or in sequence to achieve better results.

One example of hybrid prognostics is found in [24]. Here, the researchers combine a physics-based model with a condition-based one. The first type of model uses data from the system to determine the state of each component and then uses preprogrammed knowledge about component interactions to predict the overall lifetime of the system. The second model uses that same data to define an overall state of the system and then uses past experimental data to interpret the survival of systems found in a given state. Each of these approaches has their drawbacks: a physics model assumes that the experts designing the model have a perfect understanding of the system, and that no unexpected problems will arise. The condition model, on the other hand, assumes that there is data from all possible conditions, and that no new state will be found which doesn't have a match in the training data. Neither can be used alone.

To overcome these difficulties, both their predictions and the uncertainties in their predictions must be fused. How this is to be done depends on the amount of processing power available, as well as the specific type of problems (short-term or long-term) being sought. The researchers in [24], examining aeronautic data, consider prognostics both during an actual mission (while the plane is in flight) and between missions (while the plane is in a hangar). For the purposes of remote oceanic systems, there is no sense of "during" or "between" missions, but the same principles show that processor-intensive prognostics can be run less frequently, because they are meant to see further into the future.

When performing short-term, low-processor predictions, the researchers perform prediction fusion in a fairly rudimentary way: information from sensors and a physics model are fused in a diagnostic reasoner, and when the amount of predicted damage exceeds a given threshold, this information loops back to the physics model which changes into a different state. Between flights, when planning future missions, more elaborate models can be run. For example, the physics model can be used in conjunction with a mission profile for future flights to predict what sort of damage would be expected on that specific mission, based on the current state of the system. A condition-based model can also produce the same information. Each of these models will have known uncertainties, both in terms of numeric fuzziness and specific areas of the input domain known to be especially hard to predict.

Prognostic models can combine these results using simple weighted averaging or adaptive neuro-fuzzy inference systems, but combining the uncertainties is just as important. Dempster Shafer regression provides one method to do this [25]. In essence, the data is considered as a phase space: each sensor, metric, or other feature is one dimension, and $\mathbf{x}$ is a vector which represents a new instance which is to be classified. The class itself is $y$. To perform this classification, each piece of training data of the form $(\mathbf{x}_i, y_i)$ is considered; the closer that a given $\mathbf{x}_i$ is to the $\mathbf{x}$ being classified, the more its prediction $y_i$ is believed. These predictions are then combined, based on their respective belief values, and the results (along with their belief values) are returned. Thus, multiple modalities can be integrated while incorporating their individual uncertainties to inform the final result.

An alternate strategy is the agent-based approach discussed in [26]. This approach incorporates a number of separate databases and agents, covering both experience- and model-based information. In this system, the databases for the case library and the expert-written failure library are used by independent agents to generate predictions. In addition, a separate reinforcement learning agent scours the case library and attempts to discover new failure modes, which it can then add to the failure library. This allows the system to work even with minimal starting information. Over time, the case library will grow as more data is collected, and this growing case library is used to expand the failure library. Reinforcement learning can also be employed within the case-based agent to improve its predictions. Once both agents have come to their conclusions, they are combined using an additional agent, the knowledge fusion agent. While the fusion agent is not discussed extensively in [26], it could employ similar techniques as discussed earlier. Even without fusion at this stage, the use of the case library to improve the failure library allows for the best of both types of learning.

For our premiere application, ocean turbines, we have a lot of information which can be employed to create an expert-driven model. The parts are well understood in isolation: gears, bearings, and shafts have known failure modes, rated capacities, and lifespans. Collectively, these possible failure states have been assembled into a database, according to ISO-13374, the ISO standard for machine condition monitoring. This standard describes a multilayer architecture, with each layer describing progressively more complex types of PHM, progressively larger pieces of the system. Thus, it provides an outline for model-based PHM: at each level, known interactions of problems on the level below are integrated into a single state. These models can be combined with the data-driven and hybrid approaches discussed above to result in more accurate predictions.

## VI. Applying PHM to machine maintenance

As outlined earlier, PHM has been applied to a wide variety of fields. Not all of these applications are directly relevant to the specific task of predicting faults for the purpose of maintaining machines before they break. Fortunately, this is a particularly well-studied subset of the field; much research has been conducted examining how to optimize repair schedules. Some examples follow.

A straightforward method is discussed in [27]. Here, the assumption is that a single organization is handling PHM

and maintenance planning for a large, distributed network of systems, all of which are sending data back to a headquarters. After the data from the various systems are integrated into a single database which can be mined, features are selected in a two-phase process which first removes irrelevant features and then builds an entropy-based decision tree (which, since it iteratively selects those features which produce the best reduction of entropy, contains embedded feature selection). Though this decision tree classifies various instances as "in need of preventative maintenance," "in need of corrective maintenance," or "not in need of any maintenance," this tree is not used directly on live systems to decide their fates. Instead, the most significant feature is used alone, and a threshold is established which indicates that maintenance is essential. Based on historic data as well as a weighted mean slopes model (a type of generalized linear regression) for analyzing live data, the prognostic unit predicts when that feature will cross the threshold, and reports this as the time left until maintenance mus be performed. Though the use of a single feature is extremely limiting, the idea of setting a threshold and using historic data to determine how long until it is surpassed has many potential applications.

A more unusual approach to taking maintenance into account when planning a PHM system is found in [28]. This paper views maintenance as one part of an overall prognosis system, with the ultimate goal of producing a probabilistic behavior model of the system. This model combines a physics understanding of how the system works as a whole with data-derived decay information on the system's parts: events (data from a live system) are incorporated as predictors of the decay state of various components. In this framework, maintenance can be added as a type of event: rather than observing that a given part is failing, the part is actively being fixed. To determine the value of maintenance, one simply runs the model with and without maintenance; if the predicted benefits outweigh the costs, it is worth doing.

It is especially difficult to plan maintenance for remote ocean systems. Land-based systems are often relatively easy to repair in-place; either they are located in a factory which has staff available to perform the maintenance, or they are large fixed structures which have appropriate access systems to allow for workers to open up the system. Even mobile units such as aircraft and vehicles (and unmanned versions thereof) can be driven or flown into a garage. The same applies to manned aquatic systems: boats and submarines can sometimes be repaired by their crew, and even if the damaged section is outside the vessel, it can at least be examined by an expert before the entire system is hauled back to shore. This is not the case with remote ocean systems: repair missions are all-or-nothing, requiring a full expedition and recovery trip just to see if there's a problem in the first place. This make predicting needed maintenance especially important for such systems: there is a much lower margin of error, since mistakes in either direction will lead to significant costs.

## VII. Conclusions and Further Discussion

Prognostics and health monitoring is a very important part of any large-scale deployment. Simply getting the machines working and getting them running is not enough. Without monitoring, they will deteriorate, and the only sign of damage will be their complete failure. To avoid this, a comprehensive strategy must be developed to find out what the problems are before they happen. Important parts of this strategy must include:

- Expert input on possible failure modes of the system, as well as potential precursors to failure
- Choice of sensors to detect all possible problems with the system
- Construction of a model based on failure modes, including the connections between failures and symptoms (sensor data)
- Collection of data from the system, both in a healthy state and in various failure and pre-failure states
- Approach to integrate information from both model- and data-driven perspectives to understand system trajectory
- Strategy to use system information to proactively find potential failures and perform maintenance to prevent them

By employing the above outline, the resulting system will be more robust and lower cost, with repairs being performed when and only when they are necessary.

## VIII. Acknowledgment

The work discussed here grew from collaborations within the Prognostics and Health Monitoring (PHM) working group of the Center for Ocean Energy Technology (COET) at Florida Atlantic University and was funded through COET by the State of Florida.

## References

[1] M. Schwabacher and K. Goebel, "A survey of artificial intelligence for prognostics," in *AAAI Fall Symposium*, 2007.

[2] P. P. Bonissone and K. Goebel, "When will it break? a hybrid soft computing model to predict time-to-break margins in paper machines," in *Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation V* (B. Bosacchi, D. B. Fogel, and J. C. Bezdek, eds.), vol. 4787, SPIE, 2002, pp. 53–64.

[3] A. Muller, M.-C. Suhner, and B. Iung, "Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system," *Reliability Engineering & System Safety*, vol. 93, no. 2, 2008, pp. 234 – 253.

[4] V. Skormin, V. Gorodetski, and L. Popyack, "Data mining technology for failure prognostic of avionics," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, Apr 2002, pp. 388–403.

[5] K. Goebel, B. Saha, A. Saxena, J. Celaya, and J. Christophersen, "Prognostics in battery health management," *Instrumentation & Measurement Magazine, IEEE*, vol. 11, August 2008, pp. 33–40.

[6] M. A. Schwabacher, "A survey of data-driven prognostics," *AIAA InfoTech Aerospace*, 2005.

[7] S. Uckun, K. Goebel, and P. Lucas, "Standardizing research methods for prognostics," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, Oct. 2008, pp. 1–10.

[8] C. Yang and S. Letourneau, "Model evaluation for prognostics: Estimating cost saving for the end users," in *ICMLA '07: Proceedings of the Sixth International Conference on Machine Learning and Applications*, (Washington, DC, USA), IEEE Computer Society, 2007, pp. 304–309.

[9] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, and M. Schwabacher, "Metrics for evaluating performance of prognostic techniques," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, Oct. 2008, pp. 1–17.

[10] J. C. Sloan, T. M. Khoshgoftaar, P.-P. Beaujean, and F. Driscoll, "Ocean turbines – a reliability assessment," *International Journal of Reliability, Quality and Safety Engineering*, vol. 16, October 2009, pp. 413–433.

[11] N. Vichare and M. Pecht, "Prognostics and health management of electronics," *Components and Packaging Technologies, IEEE Transactions on*, vol. 29, march 2006, pp. 222 – 229.

[12] R. B. Chinnam and P. Mohan, "Online reliability estimation of physical systems using neural networks and wavelets," *International Journal of Smart Engineering System Design*, vol. 4, no. 4, 2002, pp. 253–264.

[13] N. Gebraeel, M. Lawley, R. Liu, and V. Parmeshwaran, "Residual life predictions from vibration-based degradation signals: a neural network approach," *Industrial Electronics, IEEE Transactions on*, vol. 51, june 2004, pp. 694 – 700.

[14] J. Kozlowski, M. Watson, C. Byington, A. Garga, and T. Hay, "Electrochemical cell diagnostics using online impedance measurement, state estimation and data fusion techniques," in *IECEC01: 36th Intersociety Energy Conversion Engineering Conference*, American Society of Mechanical Engineers, July–August 2001.

[15] J. Lee, "Measurement of machine performance degradation using a neural network model," *Computers in Industry*, vol. 30, no. 3, 1996, pp. 193 – 209. Computer Integrated Manufacturing.

[16] M. Roemer, J. Ge, A. Liberson, G. Tandon, and R. Kim, "Autonomous impact damage detection and isolation prediction for aerospace structures," in *Aerospace Conference, 2005 IEEE*, march 2005, pp. 3592 – 3600.

[17] Y. Shao and K. Nezu, "Prognosis of remaining bearing life using neural networks," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 214, Jan 2000, pp. 217–230.

[18] R. Sharda, "Neural Networks for the MS/OR Analyst: An Application Bibliography," *INTERFACES*, vol. 24, no. 2, 1994, pp. 116–130.

[19] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, vol. 1 of *Santa Fe Institute Studies in the Sciences of Complexity*. Westview Press, a Member of the Perseus Books Group, Boulder, Colorado / Oxford, 1991.

[20] J. A. Freeman and D. M. Skapura, *Neural Networks: Algorithms, Applications and Programming Techniques*. Computational and Neural Systems, Addison-Wesley Publishing Company, Pasadena, California, 1991.

[21] L. Zhang, L. He, K. Ben, N. Wei, Y. Pang, and S. Zhu, *Advances in Neural Networks – ISNN 2005*, vol. 3498 of *Lecture Notes in Computer Science*, ch. Identification of the Acoustic Fault Sources of Underwater Vehicles Based on Modular Structure Variable RBF Network, pp. 567–573. Springer, Berlin / Heidelberg, 2005.

[22] R. B. Chinnam and P. Baruah, "A neuro-fuzzy approach for estimating mean residual life in condition-based maintenance systems," *International Journal of Materials and Product Technology*, vol. 20, 10 May 2004, pp. 166–179(14).

[23] L. Studer and F. Masulli, "On the structure of a neuro-fuzzy system to forecast chaotic time series," in *Neuro-Fuzzy Systems, 1996. AT'96., International Symposium on*, aug 1996, pp. 103 –110.

[24] K. Goebel, N. Eklund, and P. Bonanni, "Fusing competing prediction algorithms for prognostics," in *Aerospace Conference, 2006 IEEE*, 2006, pp. 1–10.

[25] S. Petit-Renaud and T. Denoeux, "Nonparametric regression analysis of uncertain and imprecise data using belief functions," *International Journal of Approximate Reasoning*, vol. 35, no. 1, 2004, pp. 1 – 28.

[26] L. Tang, G. Kacprzynski, J. Bock, and M. Begin, "An intelligent agent-based self-evolving maintenance and operations reasoning system," in *Aerospace Conference, 2006 IEEE*, 2006, pp. 1–12.

[27] A. Bey-Temsamani, M. Engels, A. Motten, S. Vandenplas, and P. Agusmian, "A practical approach to combine data mining and prognostics for improved predictive maintenance," in *15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-2009)*, June – July 2009.

[28] A. Muller, M. Suhner, and B. Iung, "Maintenance alternative integration to prognosis process engineering," *Journal of Quality in Maintenance Engineering*, vol. 13, no. 2, 2007, pp. 198–211.