Chapter 6

Basic Principles

George Vachtsevanos, School of Electrical and Computer Engineering, Georgia Institute of Technology

Kai Goebel, National Aeronautics and Space Administration

"*Ibis Redibis Non Morieris In Bello"*— Pythia

6.1 Introduction

Integrated Vehicle Health Management (IVHM) must rely on accurate and robust detection of incipient failures (faults) for critical components / systems and estimation of the Remaining Useful Life (RUL) of such failing components. Recently, stringent diagnostic, prognostic, and health management capability requirements are being placed on many new aircraft and spacecraft applications to enable the benefits of logistic support concepts, but also to assist in avoiding catastrophic events. While effective diagnostics with low false alarm rates continue to improve on these new applications, prognostic requirements are even more ambitious and present very significant challenges to both the system design and program management teams.

The "reasoning" performed by the diagnostics and prognostics tasks is done using sophisticated algorithms that process sensor information, historic use, design and materials data, alarm thresholds, real-time usage information, future load, and environmental conditions. (These algorithms, called "reasoners," are discussed in more detail in Chapter 7.) Before any of the reasoning can be carried out, a host of algorithmic steps has to be completed to ensure that the algorithms receive the correct information. In particular, there are checks that the sensor information, upon which much of the reasoning hinges, are working properly; that noise is removed to the largest degree possible; that the information is brought into focus through various transformations; etc. All this has to be done in an architectural framework that supports these tasks in the optimal fashion. This chapter reviews the fundamental principles that form the foundation for Prognostics Health Management (PHM) and Condition-Based Maintenance (CBM) technologies.

6.2 The OSA-CBM Framework

This chapter presents the general outline of a systems engineering approach that facilitates the integration and interchangeability of computational components from a variety of sources. An open systems standard should consist of publicly available descriptions of component interfaces, functions, and behaviors. The Open Systems Architecture for Condition-Based Maintenance (OSA-CBM) has been developed specifically to meet these requirements **Error! Reference source not found.**].

Figure 6.1 is a schematic representation of the major modules of the OSA-CBM architecture. The figure suggests the progression of components from data acquisition, data manipulation (processing), condition monitoring (state detection), health assessment, prognostics, and decision support (advisory generation). The components are not application specific; they are scalable and upgradeable.

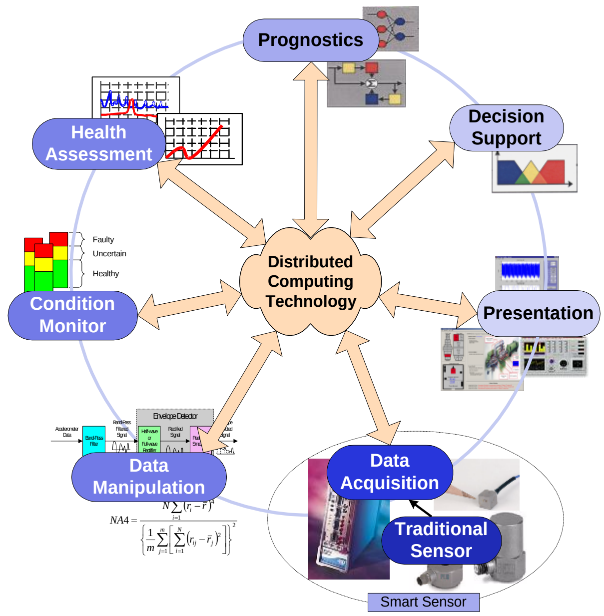

Figure 6.1 The OSA-CBM components.

MIMOSA is a standard for data exchange between asset management systems, whereas OSA-CBM is a specification for transactions between components within a Condition-Based Maintenance system **Error! Reference source not found.**]. The core of the OSA-CBM standard is the Object Oriented data model, defined using Unified Modeling Language (UML) syntax. It is a mapping of key concepts from the MIMOSA Common Relational Information Schema (CRIS) with extensions for diagnostics, prognostics, and data transactions. This common architecture has been demonstrated in a variety of application domains. In reviewing fundamental IVHM technologies, we pursue the same basic structure of the OSA-CBM architecture.

6.3 An Integrating IVHM Architecture

Figure 6.2 lists the major modules of a generic architecture in sequential order. The modules include Sensor Validation, Data Pre-Processing, Feature Extraction, Data Fusion, Anomaly Detection, Diagnostic Analysis, Prognostic Analysis, and Contingency Management. This chapter covers Sensor Validation, Data Pre-Processing, Feature Extraction, and Data Fusion. Anomaly Detection, Diagnostic Analysis, Prognostic Analysis, and Contingency Management are covered in Chapter 7, “Algorithms and their Impact on IVHM.”

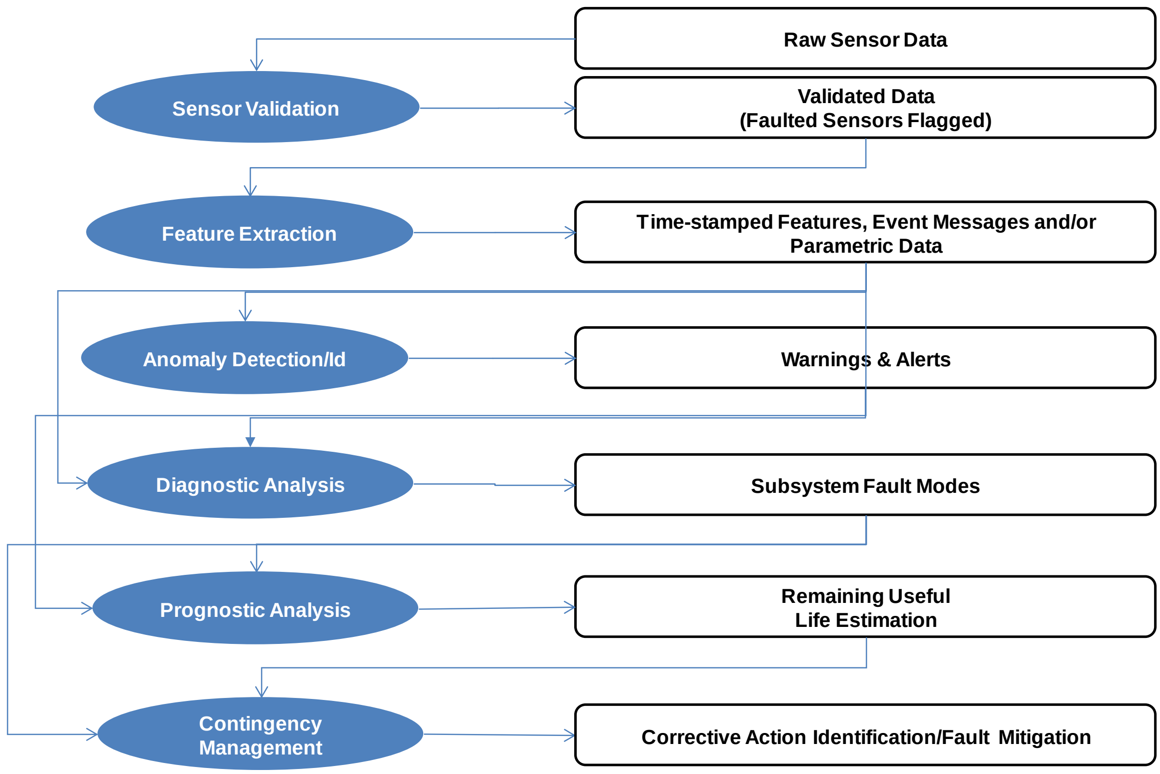

Figure 6.2 Integrated Vehicle Health Management functions.

Figure 6.3 depicts a more specific architecture for implementation of fault diagnosis and failure prognosis algorithms [Vachtsevanos et al. 2006; Roemer et al. 2005] on board a helicopter. In this example, some of the functions mentioned in Figure 6.2 are performed on board, while others are performed off board (it should be noted that in other applications, this separation is not being made). A further distinction is made in that training of the models and algorithms is performed offline, while analysis of data from the system is made online; i.e., while the system is operating. For the example case considered in Figure 6.3, the online modules perform raw data pre-processing, feature extraction, fault diagnosis, and failure prognosis that exploit available ground truth fault data, noise models, experimental data, system models, and other tools offline to tune and adapt online parameters and estimate suitable mappings. The architecture suggests a hybrid and systematic approach to sensing, data processing, fault feature extraction, fault diagnosis, and failure prognosis that may lead to a system hardware/software configuration implementable online in real time. The integrated architecture, when augmented with V&V (Verification and Validation) studies, may be optimized to facilitate its eventual on-platform transition.

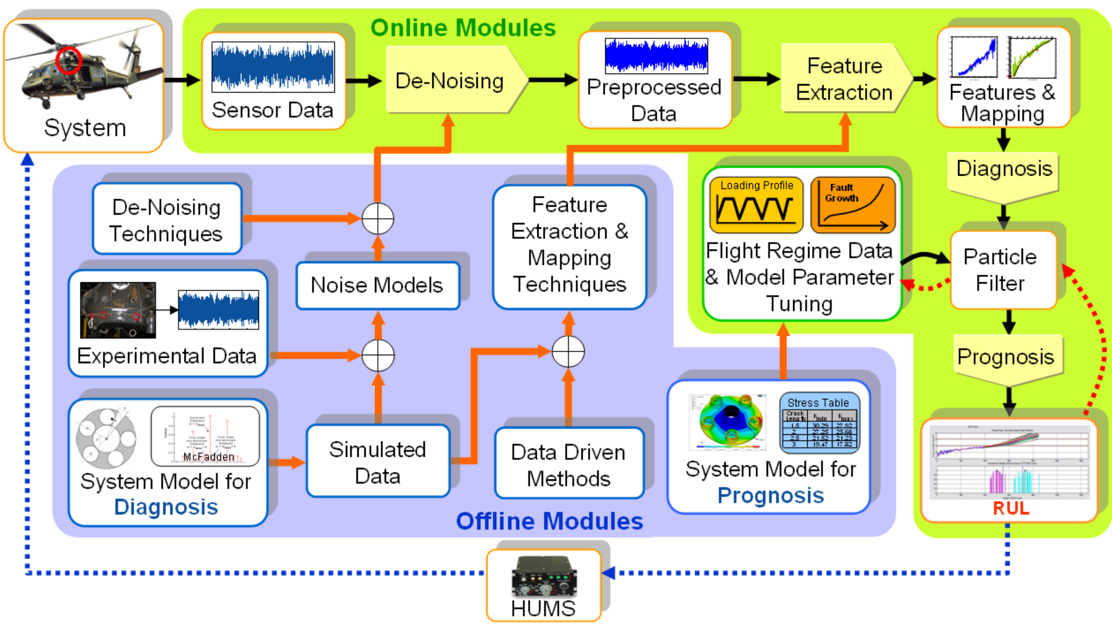

Figure 6.3 An architecture for development and implementation of fault diagnosis and failure prognosis algorithms.

The online modules are designed to perform in sequence: data pre-processing, feature extraction, diagnosis, and prognosis. The architecture suggests also the possibility of closing the loop and providing corrective action to maintain a degree of acceptable system performance for the duration of an emergency.

6.4 Sensing and Data Processing

Much of the reasoning in IVHM hinges on data obtained from sensors. Therefore, sensors and sensing strategies constitute the foundational basis for fault diagnosis and prognosis. Strategic issues arising with sensor suites employed to collect data that eventually will lead to online realization of diagnostic and prognostic algorithms are associated with the type, number, and location of sensors (see Chapter 8 for more information on this topic); their size, weight, cost, dynamic range, and other characteristic properties; whether they are of the wired or wireless variety; etc. Data collected by transducing devices rarely are useful in their raw form. Such data must be processed appropriately to enable extraction of useful information that is a reduced version of the original data but preserves, as much as possible, those characteristic features or fault indicators that are indicative of the fault events we are seeking to detect, isolate, and predict the time evolution of. Thus, such data must be preprocessed, that is, filtered, compressed, correlated, etc., in order to remove artifacts, and reduce noise levels and the volume of data to be processed subsequently. Furthermore, the sensors providing the data must be validated; that is, the sensors themselves might be subjected to fault conditions. Once the preprocessing module confirms that the sensor data are “clean” and formatted appropriately, features or signatures of normal or faulty conditions must be extracted. This is the most significant step in the Condition-Based Maintenance/Prognostics Health Management (CBM/PHM) architecture whose output will set the stage for accurate and timely diagnosis of fault modes. The extracted-feature vector will serve as one of the essential inputs to fault diagnostic algorithms.

6.4.1 Sensor Validation

Raw sensor data are a measurement of operational and environmental quantities. Before they are further processed, the sensor itself must be assessed to determine its integrity in a step called "sensor validation." Here, data are acquired from sensors (and possibly from other sources) to be validated. Next, the output of each sensor is estimated using analytical relationships with other sensors. For example, the pressure at a particular location in the flow of a system is related to the pressure at a different reading via laws of physics that can be expressed as mathematical equations. One can then build entire networks of relationships ("Analytical Redundancy Relationship Network" **Error! Reference source not found.**]) between sensors for which the readings are all related in some way via mathematical equations.

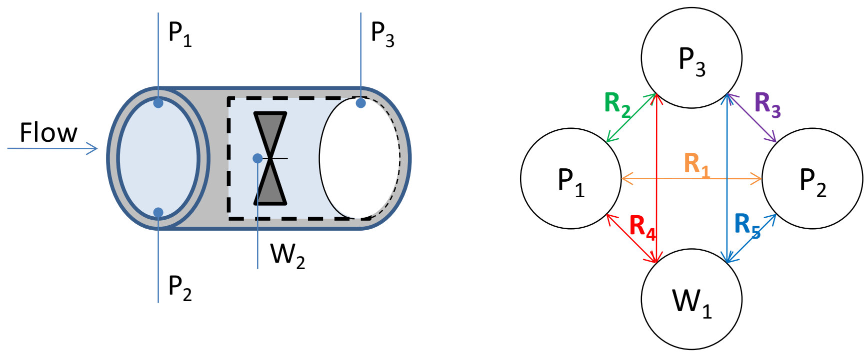

Figure 6.2 Analytical Redundancy Relationship Network **Error! Reference source not found.** et al. 2009].

As an illustrative example, consider a medium flowing in a pipe (Figure 6.2). One can then describe physical relationships between the sensor measurements. Next, one would determine whether any of these relationships are violated. That can be done by checking whether the difference between the measurement and the estimate is larger than some predefined threshold **Error! Reference source not found.**]. For example, if the difference between the estimate of the measurement (as established through the physical relation) and the actual measurement is less than the threshold, then the relation is declared qualified; otherwise, it is declared failed. As a safeguard, and to contain the number of false positives (i.e., the number of cases where a sensor is declared failed although it is still operational), one can require more than one of these relationships to fail or one can wait for repetitions of condition violation to declare the sensor failed. In addition, one can employ health management principles on the sensors themselves and when a sensor fault is suspected, have the system call for maintenance on itself before it fails.

6.4.2 Data Pre-Processing

Raw sensor data (vibration, temperature, etc.) must be pre-processed to reduce the data dimensionality and to improve the (fault) Signal to Noise Ratio (SNR). Typical pre-processing routines include data compression and filtering, Time Synchronous Averaging (TSA) of vibration data, Fast Fourier Transforms (FFTs), etc. Pre-processing methods, which improve the SNR (de-noising), are particularly valuable in aircraft situations where significant noise levels tend to mask the real information. As an example, a de-noising methodology, based on blind deconvolution, for a helicopter application is outlined below.

The process of blind deconvolution attempts to restore the unknown vibration signal by estimating an inverse filter, which is related to partially known system characteristics. This is an active field of current research in image processing, speech signal processing **Error! Reference source not found.**], but rarely applied in mechanical vibration signals

Error! Reference source not found.]. Vibration and other high-bandwidth signals are typically corrupted by multiple noise sources. An iterative de-noising scheme may be constructed starting with an initial estimate of the inverse of the modulating signal, which demodulates the observed signal to give a rough noise-free estimate in the time-domain. Its Fourier transform is passed next through a nonlinear projection, yielding the ideal characteristics of the vibration signal. The estimate is iteratively refined via an optimization scheme. Figure 6.5 shows the blind deconvolution de-noising scheme. Note that the proposed scheme is implemented in the frequency domain, and the nonlinear projection, which is derived from a nonlinear dynamic model, is also given in the same frequency domain **Error! Reference source not found.**2010].

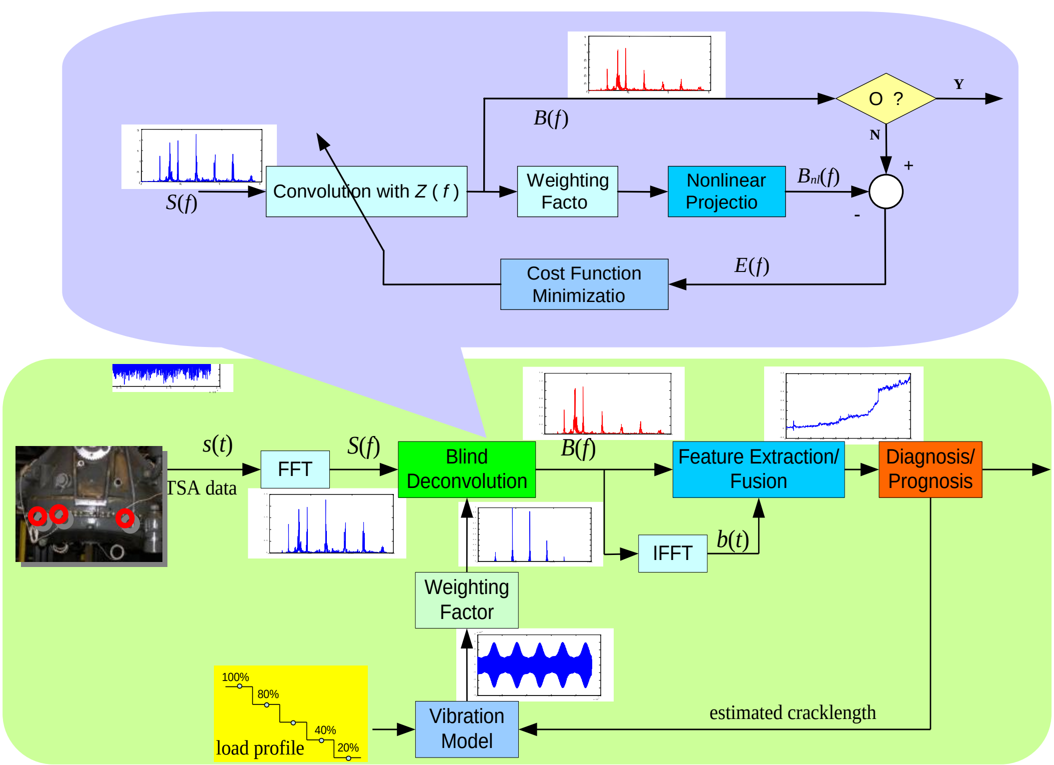

Figure 6.3 A blind deconvolution data de-noising scheme.

6.4.3 Feature Extraction and Selection

Feature or Condition Indicator (CI) selection and extraction constitute the cornerstone for accurate and reliable fault diagnosis. The classical image recognition and signal processing paradigm of *data* → *information* → *knowledge* becomes most relevant and takes central stage in the fault diagnosis case, particularly since such operations must often be performed in a real-time environment. Irrespective of whether the function has to be performed online or offline, the objective is to transform high-dimensional raw data into a tractable lower-dimensional form without loss of useful information.

Fault diagnosis depends mainly on extracting a set of features from sensor data that can distinguish between fault classes of interest, and can detect and isolate to a particular fault at its early initiation stages [Wu et al. 2005; **Error! Reference source not found.**]. These features should be fairly insensitive to noise and within fault class variations. "Good" features must have the following attributes. They must be:

- Computationally inexpensive to measure
- Mathematically definable
- Explainable in physical terms
- Characterized by large interclass mean distance and small interclass variance
- Insensitive to extraneous variables
- Uncorrelated with other features

Researchers have relied on ad hoc or empirical methods to define a feature vector for a particular application domain. Knowledge of the system structure and function, modeling, and heuristics is called upon to arrive at the "best" features or CIs.

Past research has focused on feature extraction, whereas feature selection has relied primarily on expertise, observations, past historical evidence, and understanding of fault signature characteristics. In selecting an "optimum" feature set, the following questions need to be addressed: Where is the information? How do fault (failure) mechanisms relate to the fundamental "physics" of complex dynamic systems? How do fault modes induce changes in the energy, entropy, power spectrum, signal magnitude, etc.? Is the feature selection application-dependent?

When seeking those features for a particular class of fault modes from a large candidate set that possesses properties of fault distinguishability and detectability, a reliable fault classification must be determined in the minimum amount of time. Feature extraction, on the other hand, is an algorithmic process where features are extracted in a computationally efficient manner from sensor data, while preserving the maximum information content. A hybrid methodology for feature selection and extraction may rely on physics-based modeling of the fault modes in combination with sensor data as the latter are streaming into the processor. The physics-based models employ a finite element analysis technique jointly with a nonlinear dynamic model of the failing component's behavior to guide the selection process. Figure 6.6 depicts a typical scheme for feature extraction from raw vibration data. The data in each Ground-Air-Ground (GAG) cycle is reduced to one feature value **Error! Reference source not found.** et al. 2009].

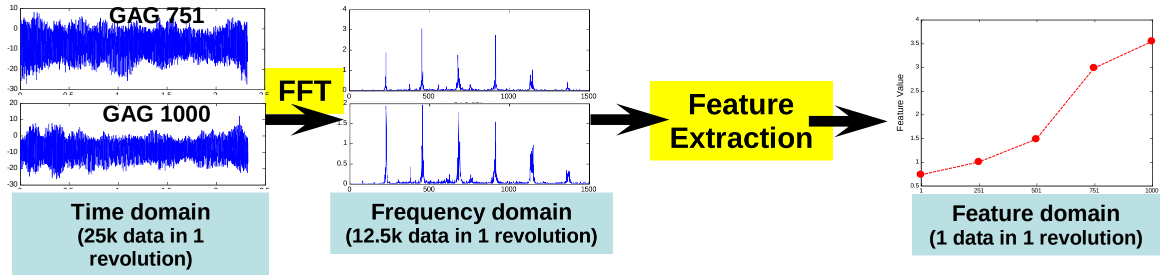

Figure 6.4 A typical illustrative example of a feature extraction scheme.

Feature evaluation and selection metrics include the similarity (or linear correlation) between the feature and the true fault (crack) size, based on the linear dependency between them. A feature is desirable if it shows a similar growth pattern to that of the ground truth data. A correlation coefficient, the covariance between the two signals divided by their standard deviations, may be employed as a metric of similarity for feature evaluation. When multiple features are extracted for a particular fault mode, it might be desirable to combine or fuse uncorrelated features to enhance the fault detectability. Genetic Programming algorithms may assist to define an appropriate fitness function by using genetic operators to construct new feature populations from old ones.

6.4.4 Sensor Data Fusion

Although significant achievements have been reported in the recent past, the processing of sensor data intelligently still requires development, testing, and validation of new techniques to manage and interpret the increasing volume of data, and to combine them as they become available from multiple and diverse sources. Sensor data fusion describes a set of techniques that can contribute significantly toward a better understanding and a more efficient utility of raw data by reducing it to useful information. Information is synthesized to higher informational levels. A typical sensor data fusion process incorporates several levels of abstraction: fusion at the data level, feature (characteristic signature of the collected data) level, sensor level, and knowledge level. At the data level, a variety of filtering, data compression, and data validation algorithms are employed to improve such indicators as signal to noise ratio, among others. The enabling technologies at the feature level may borrow from Dempster-Shafer theory [Shafer 1976], soft computing, and Bayesian estimation to fuse features while meeting specified performance metrics. At the sensor level, we rely upon concepts from information theory while multiple sensors are gated and coordinated spatially and temporally to minimize their number while maximizing the probability of detection. Significant reduction of the computational burden is always a desired objective. The top level of the fusion hierarchy, i.e., the knowledge fusion module, is designed to reason about the evidence provided by the lower echelons, aggregate the available information in an intelligent manner, resolve conflicts, and report to the end-user the finding of the fusion architecture. Artificial Intelligence (AI) tools and methods from Dempster-Shafer theory, Bayesian estimation techniques, and soft computing may find utility as the reasoning enablers at this level.

6.5 Diagnostics and Prognostics

Fault diagnostics and, more recently, prognostics have been the subject of in-depth study. Researchers in such diverse disciplines as medicine, engineering, the sciences, business, and finance have developed methodologies to detect fault (failure) or anomaly conditions, to pinpoint or isolate which component in a system / process is faulty, to decide on the potential impact of a failing or failed component on the health of the system, and to determine a component's remaining life. Diagnostics and Prognostics are covered in more detail in Chapter 7.

6.6 Performance Metrics

Generally, metrics to measure the performance and effectiveness of the CBM/IVHM system are laid out during the requirements phase as part of the systems engineering process. Low-level performance metrics are used to verify that algorithms meet stated performance goals. For fault detection, typical metrics include detection rates. Fault isolation is the ability to determine the exact fault root cause from a set of possible root causes. Here, false negatives (the inability to detect a fault), false positives (incorrectly determining the presence of a fault), and false classified (identifying the wrong root cause) are important metrics. An exhaustive set of metrics for diagnostics is presented by Kurtoglu et al. **Error! Reference source not found.**et al. 2009]. In the context of prognostics, the prognostic horizon (the first prediction of remaining life within acceptable uncertainty bounds) is an important metric. Similarly, $\alpha-\lambda$ performance quantifies prediction quality by determining whether the prediction falls within specified limits at particular times with respect to an accuracy or precision performance measure. The topic is discussed in more detail by Saxena et al. [2010].

6.7 Database Management

Central to a successful and efficient health management architecture that enables some of the functionality outlined above is a well-designed database and a flexible database management schema. There are OSA-CBM and MIMOSA efforts to establish database standards for IVHM. The purpose of the database is to store and facilitate exchange of the different types of information that come from the sensors, the feature extraction, and the reasoning module (abnormal condition detection, diagnostics, prognostics, and contingency management) as well as static information. Demands on such a database can be quite high since it must collect dynamic information that arrives at various, possibly non-synchronous, instances. Thus, the database management system must be able to provide the ability to organize large amounts of data in linked tables to facilitate ease of understanding. It also needs to provide a complete language for data definition, retrieval, and update.

6.8 Closing Thoughts

This chapter described at a very high level some of the considerations that need to be made when designing algorithms for a vehicle health management application. The choices made here affect the quality of the diagnosis and prognosis (covered in Chapter 7). Therefore, the algorithmic design choices are made in conjunction with the design choices for diagnostics and prognostics to optimally support these tasks. Furthermore, additional considerations imposed by computational constraints, resource availability, algorithm maintenance, need for algorithm re-tuning, etc. will impact the solutions.

It should also be noted that technological advances, both in hardware and software, impose the need for new solutions. For example, as new materials and new sensors are being developed, the algorithmic solutions will need to follow suit.

In general, there seems to be a trend to have more sensor data available. While this is potentially a good thing, sensor data provides value only when it is being processed and interpreted properly, in part by the techniques described here. Testing of the methods, however, requires the “right” kind of data. Generally, there is a lack of seeded fault data which are required to train and validate algorithms. It is also important to migrate information from the component to the subsystem to the system levels so that health management technologies can be applied effectively and efficiently at the vehicle level. It may be required to perform elements described in this chapter between different levels of the vehicle architecture.

References

Brown, D., G. Georgoulas, B. Bole, H. L. Pei, M. Orchard, L. Tang, B. Saha, A. Saxena, K. Goebel, and G. Vachtsevanos. 2009. “Prognostics enhanced reconfigurable control of electromechanical actuators.” In *Proceedings of 2nd International Conference on Prognostics and Health Management (PHM)*, San Diego, CA, USA, September 27 – October 1, 2009.

Kundur, D. and D. Hatzinakos. 1996. “Blind Image Deconvolution.” *IEEE Signal Processing Magazine*. 13 (3), 1996, pp. 43-46.

Kurtoglu, T., S. Narasimhan, S. Poll, D. Garcia, L. Kuhn, J. de Kleer, A. van Gemund, and A. Feldman. 2009. “A Framework for Evaluating and Comparing Diagnosis Algorithms.” *20th International Workshop on Principles of Diagnosis (DX ’09)*. Stockholm, Sweden. 373-381.

Lianghua, X., R. Liqing, Z. Mei, W. LiXin, M. Qiang. 2010. "Intelligent Computing and Intelligent Systems (ICIS), " 2010 IEEE International Conference. Issue Date 29-31, Oct. 2010.

Maul, W., R. Bickford, and K. Melcher. 2009. Use of Analytical Redundancy Relationship Networks for Sensor Data Qualification. " NASA Technical Memorandum 2009-216067.

Mimosa. 2011. Website: www.mimosa.org.

Patrick, R., A. Ferri, and G. Vachtsevanos. 2007. "Detection Of Carrier-Plate Cracks Using Vibration Spectra," ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2007. Las Vegas, Nevada, USA, September 4-7, 2007.

Peled, R., S. Braun, and M. Zacksenhouse. 2000. "A Blind Deconvolution Separation of Multiple Sources, with Application to Bearing Diagnostics." *Mechanical Systems and Signal Processing*, 14 (3), 2000, pp. 427-442.

Roemer, M., C. Belington, G. Kacprzynski, and G. Vachtsevanos. 2005. "An Overview of Selected Prognostic Technologies with Reference to an Integrated PHM Architecture." Proceedings of NASA Integrated Vehicle Health Management workshop (Invited), Napa, CA, November 7-10, 2005.

Saxena, A., J. Celaya, B. Saha, S. Saha, and K. Goebel. 2010. "Metrics for Offline Evaluation of Prognostic Performance," *International Journal of Prognostics and Health Management*, 001, 2010.

Shafer, G. 1976. *A Mathematical Theory of Evidence*. Princeton University Press.

Vachtsevanos, G., F. Lewis, M. Roemer, A. Hess, and B. Wu. 2006. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley & Sons, Inc.

Wu, B., A. Saxena, R. Patrick, and G. Vachtsevanos. 2005. "Vibration Monitoring for Fault Diagnosis of Helicopter Planetary Gears." Proceedings of the 16th IFAC World Congress, July 2005.

Zhang, B., T. Khawaja, R. Patrick, G. Vachtsevanos, M. Orchard, and A. Saxena. 2010. "A novel blind deconvolution de-noising scheme in failure prognosis." *Transactions of the Institute of Measurement and Control* 32, 1, pp. 3-30, 2010.