ABSTRACT

Title of Document: AGENT AUTONOMY APPROACH TO PROBABILISTIC PHYSICS-OF-FAILURE MODELING OF COMPLEX DYNAMIC SYSTEMS WITH INTERACTING FAILURE MECHANISMS

Katherine Emily Gromek, Doctor of Philosophy (Ph.D.) in Reliability Engineering 2015

Directed By: Professor Mohammad Modarres, Department of Mechanical Engineering

A novel computational and inference framework of the physics-of-failure (PoF) reliability modeling for complex dynamic systems has been established in this research. The PoF-based reliability models are used to perform a real time simulation of system failure processes, so that the system level reliability modeling would constitute inferences from checking the status of component level reliability at any given time. The "agent autonomy" concept is applied as a solution method for the system-level probabilistic PoF-based (i.e. PPoF-based) modeling. This concept originated from artificial intelligence (AI) as a leading intelligent computational inference in modeling of multi agents systems (MAS).

The concept of agent autonomy in the context of reliability modeling was first proposed by M. Azarkhail [1], where a fundamentally new idea of system representation by autonomous intelligent agents for the purpose of reliability modeling was introduced. Contribution of the current work lies in the further development of the agent anatomy concept, particularly the refined agent classification within the scope of the PoF-based system reliability modeling, new approaches to the learning and the autonomy properties of the intelligent agents, and modeling interacting failure mechanisms within the dynamic engineering system. The autonomous property of intelligent agents is defined as agent's ability to self-activate, deactivate or completely redefine their role in the analysis. This property of agents and the ability to model interacting failure mechanisms of the system elements makes the agent autonomy fundamentally different from all existing methods of probabilistic PoF-based reliability modeling.

AGENT AUTONOMY APPROACH TO PROBABILISTIC PHYSICS-OF-FAILURE MODELING OF COMPLEX DYNAMIC SYSTEMS WITH INTERACTING FAILURE MECHANISMS

By

Katherine Emily Gromek

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor Mohammad Modarres, Chair
Emeritus Professor Ali Mosleh
Professor Abhijit Dasgupta
Professor Eyad H. Abed, Dean's Representative
Associate Professor Enrique Lopez Droguett

Dedication

Commitment, effort, and perseverance were fundamental elements for the completion of my doctoral dissertation, but even more was the support of my family. I dedicate this work and give special thanks to my loving parents, Ella and Igor, and my wonderful daughter Michelle, who have always been by my side at every step of the doctorate program. Without their unconditional love, patience, and understanding this professional achievement would not have been possible. Mom and Dad, thank you for being always available to take care of my daughter when I had to write the paper. Your prayers, good wishes, and interest are specially treasured in my heart. To my daughter Michelle, thank you because just your presence was enough to keep me on the right track. I love you all.

Acknowledgements

First and foremost, I wish to extend my deepest gratitude to my advisor, Dr. Mohammad Modarres, for his continual guidance, caring, support and patience throughout the entire process. The joy and enthusiasm he has for the research is truly motivational and commendable. To Dr. Mohammad Modarres, I genuinely thank you for your time, for the knowledge you shared and the encouragement you gave me. It has been an honor to be your Ph.D. student. From the very beginning of my educational journey at University of Maryland, you have been a role model for me. To the rest of the professors throughout the courses of the curriculum of the Reliability Engineering Program, I want to thank you for your dedication, commitment to education, research, and practice in reliability and risk analysis. Special thanks to Dr. Ali Mosleh and Dr. Jeffrey Herrmann for being truly inspirational teachers.

To my mentor and friend, Dr. Vasiliy Krivtsov, you were very special; you have encouraged me to join Reliability Engineering Program at University of Maryland, and that has become a turning point in my professional carrier. The knowledge I gained at UMD is unprecedented.

Finally I would like to thank all my colleagues at Bombardier Aerospace who assisted me with this project. Their help and willingness to share expertise made the completion of this research an enjoyable experience. The last but not least, many thanks to my manager at Bombardier Aerospace, Brian Alphonso, for his continued support during these challenging years of the new aircraft development program.

Table of Contents

List of Tables

List of Figures

Chapter 1: Background, Motivation and Contribution

1.1. Introduction

As technological advancements have been rapidly progressing over the last few decades, engineering systems are becoming increasingly complex, and as a result, they experience new failure modes because of complex and often interdependent physical failure mechanisms. New technology has required adapted approaches to ensure that engineering systems meet desired reliability goals in a cost-effective and timely manner.

Since the 1990's, reliability engineering faced a strong challenge to eliminate the need to rely solely on life tests and historical failure data for reliability assessment of electronic devices, mechanical components, and complex engineering systems[1]. As life tests are becoming more costly and time consuming, the classical way of reliability assessment is mostly dependent on the availability of extensive field data for empirical modeling and component libraries having reliability measures for similar parts. The failure rate prediction made by this approach often results in

[1] In this research, the terms "part", "component" and "system" used in hardware system description, are defined as follows:
1) Part (or piece part) is a simplest constituent element of more complex items, further defined as components, so that the only constituent element of the part is material it is made of.
2) Component is more complex item comprised of parts, it is a combination of parts having a specific function, which can be installed or replaced only as an entity.
3) System is functionally, physically, and/or behaviorally related group of regularly interacting or interdependent elements, such as hardware (components), software (programs) and human elements.
4) Note that a complex component may be considered a system, and the terms "component" and "system" are sometimes used interchangeably to the user's discretion, often depending on the levels of system hierarchy and complexity of the item. In contrary, a piece part cannot be treated as a component in the context of this research, even though the term "component" is often used in the literature referring to a "part". Also, the term "sub-system" is not defined here, but can be found in the literature, meaning "component" in the context of this research.

inaccurate reliability estimates due to only partial relevance of historical data to a new system with different design subjected to non-identical operating conditions.

As an alternative to fully empirical reliability modeling not rooted in the underlying physical process, the physics-of-failure (PoF) approach has emerged in the past 50 years as a powerful method of component reliability analysis based on modeling and simulation that relies on understanding the underlying physical processes contributing to the degradation, damage and occurrence of critical failures. This concept has been in use in fracture mechanics and by structural engineers for many years, but in the 1990's it has been more extensively used in reliability engineering. Sufficient computational tools and technological advancements in testing technologies also supported more practical modeling of the underlying physical phenomena.

The PoF-based reliability models, also known as probabilistic-mechanistic life models, have proven to deliver the most comprehensive representation, capable of bringing many influential factors into the reliability models of the engineering system. As such, the PoF approach was shown to bring more reality into reliability models. In addition, these models have wider applications because they are more flexible to accommodate variation of item characteristics and usage profiles.

Due to inevitable variations of many factors involved in failure processes and limited information about them, the probabilistic physics-of-failure (PPoF) methodologies were developed for assessing the reliability of parts and components by involving variations of environmental and operational stresses, mission profiles, and manufacturing processes. Taking the PPoF approach to the modeling of a complex dynamic system, however, can be very challenging, if not impossible, due to the diversity of components and their failure mechanisms, the complexity of system logic, and various types of dependencies at all levels of system hierarchy.

The traditional static techniques of system reliability (e.g. fault tree, event tree, reliability block diagram, Bayesian belief network), as well as dynamic methods (e.g. Markov chains, stochastic Petri nets, dynamic event trees) demonstrated a limited capability to incorporate physics of failure into the system level assessment and offer no or very limited ability to model quantitative causal relations between interdependent and interacting failure mechanisms of the system elements during the system evolution over time.

New methodologies are required and currently being developed by several researchers to introduce PoF into system reliability in a robust, structured, but also flexible manner to capture the dynamics of system evolution. The intelligent agent-oriented approach is introduced in this research as a framework for efficient use of the PoF information and models in developing reliability assessment approaches for highly reliable mechanical and electronic systems, structures and components with interacting failure mechanisms.

1.2. Reliability Modeling of Complex Engineering Systems

The PoF modeling approach, driven by the first principles of degradation and failure, continues standing as the dominant method for reliability modeling of mechanical, electromechanical, and electronic components. While significant achievements have been made with the PoF application for component reliability assessment, a formal structure of the PoF approach for the system level analysis is not yet defined [2]. Widely used static methodologies of system reliability modeling, specifically event tree, fault tree and Bayesian belief network [3] - [7], do not explicitly treat the time-dependent interactions between operational variables (i.e. environmental factors and operational parameters) and triggered or stochastic events (e.g., degradation and failure of components) that may lead to the coupling between these events during system operation. Poor treatment of such dynamic interactions implies that potentially significant dependencies between failures events may not be identified or properly quantified with current methods.

The dynamic methodologies of system reliability modeling, specifically Markov chain [6], [8] - [13], Petri net [6], GO-FLOW [14], [15], Dynamic Master Logic Diagram [16] and various types of dynamic event trees [6], [14], [17] - [36], are intended to address system dynamics, but their application is limited to specific aspects of reliability engineering and Probabilistic Risk Assessment [3] and has a limited capability to capture interacting degradation processes and interdependent failure events. Also, both static and dynamic traditional methods of system modeling are lacking a robust framework for aggregation of available data from various sources, such as material level fatigue models, material coupon tests, component tests, field data, in-service inspection findings, health monitoring measurements, data from generic sources, expert opinion, and partially relevant data.

In summary, both static and dynamic traditional methods of system modeling impose several challenges when applied to modern engineering systems:

1. Modeling degraded states of the system

Modern systems have three basic elements: hardware, software, and human elements. Reliability engineering was originally developed to handle failures of hardware components [37]. A commonly adopted assumption underlying the quantitative analysis of hardware failures by the system analysis methods is that systems are made up of binary components (i.e., devices that can be in two states, either fully functional or failed). Yet, there are many components which could operate in a degraded state as a result of the wear-out / aging process. As a result, the overall performance of a system can settle at different levels (compared to the initial performance level) depending on the operational states of its constituent elements. As the degradation of system components proceeds, the level of system performance becomes a function of time until a complete loss of system functionality occurs. In literature, such systems are referred as degraded systems. Their analysis requires the development of new techniques of system state representation, modeling and quantification.

Degraded systems reliability analysis is not currently supported by the classical static methods of system modeling. Fault tree and event tree methods relate the states of the components to the occurrence of the top event of interest (system failure condition) and do not explicitly make use of the state changes (dynamics) of the system. In this case, the top event is defined without knowing the sequence of state changes that leads from a good functioning state to the system failure. Consequently, the scenarios that lead to the system failure condition cannot be deduced from a Fault Tree or an Event Tree. The same is true for Bayesian Belief Network.

While several dynamic modeling methodologies, such as Markov Chains or Dynamic Master Logic Diagrams, can be used to handle the degradation process of a specific multi-state system that consists of more than one multi-state component, these methodologies are limited to special cases of the system degradation scheme [16], [38] - [42]. No generic framework, however, exists to perform explicit PoF-based reliability analysis of a continuously degrading system.

2. Modeling system dynamics

System components usually operate in highly varied dynamic environments in which operational conditions of each hardware component strongly depend on the nearby components, usage stresses and environmental conditions, as well as software and human elements. It means that in a dynamic system, not only the properties of parts, components, and other system elements may change over time, but also the system configuration and failure logic. The probability of each event is conditional on the physical states of system elements, the operational and environmental conditions, and system configuration.

Conventional (static) methods of system modeling do not have the capability to depict system reliability in the context of multi-state dynamics of the system elements evolving over time. Fault trees, event trees and Bayesian belief networks generally show a snap-shot reliability, which is not dynamically sensitive to the variation of operational and environmental conditions. In order to create a dynamic model of the system, a new fault tree model is required for every scenario of the relationship between the components with each other and their environment as the system dynamics progresses over time. As a result, it becomes difficult (or almost impossible) to list all potential paths of system degradation and failure within the framework of conventional risk assessment methods.

Dynamic methods of system modeling provide some ability to quantitatively capture the realistic features of the stochastic behavior of dynamic systems. Even these approaches, however, cannot address all necessary aspects of the physical evolution of the system. In addition, they face significant limitations due to the size of the analysis when a complex system is considered.

3. *Modeling interacting and interdependent failure mechanisms*

Interdependency of failure events at all levels of system hierarchy increases system unavailability compared to the case when the system is modeled as a sequence of independent failure events. The most common sources of component dependency in a system are usually related to the operational and environmental conditions (such as temperature, pressure and other influential stresses) that may affect the life of the components or other coupling factors leading to common cause failures of several components. Both traditional (static) and dynamic system reliability modeling approaches provide some ability to model these types of dependencies (with a higher capability for dynamic methods).

In addition to the numerous links between different components by means of their operational characteristics and environmental conditions, there is another type of interdependency often overlooked due to its complexity. In the study of system behavior, there are situations when failure progress in one component may activate or accelerate failure mechanisms of other components. Also, one failure mechanism of a certain component may activate or accelerate another failure mechanism of the same component.

For example, if an ordinary gear box consists of two bearings, a shaft and one gear, the bearings and the gear have a one-way interaction with the shaft, while the shaft is able to exchange information with the gear and bearings. A change in operational characteristics of the gear passes through the shaft and impacts the operation of the bearings. Changes in operational parameters of the bearings as well as the shaft due to age related degradation result in an additional impact on the degradation process of the gear. These direct and indirect interactions are examples of interdependency between failure mechanisms of several components. At the component level, two failure mechanisms can accelerate each other, such as fatigue and corrosion.

The existing methods of system reliability analysis, static or dynamic, have limited capability of providing quantitative causal relations between several competing mechanisms that cause failures [43], [44]. The same applies to modeling complex interactions between hardware, software, and human elements within a dynamic process of system degradation and failure processes. There are very few studies that deterministically model the interactions between failure mechanisms by means of the finite element analysis (FEA) method, and there are no probabilistic models that consider the interdependency and interactions of the competing failure mechanisms [43].

As such, PoF-based reliability modeling techniques, similar to those employed for stress or temperature analysis, are needed for accurate reliability analysis of an engineering system. New methodologies are currently being developed to introduce PoF into system reliability in a robust, structured yet flexible manner to capture the dynamics of system evolution. The Dynamic Hybrid Bayesian Network approach has been recently introduced by C. Iamsumang [45] as a computational algorithm for reliability inference in real-time System Health Management (SHM) of complex engineering systems considering the underlying physics of failure. Another methodology of PoF-based system reliability modeling is proposed in the current research.

1.3. Distributed Artificial Intelligence

Agent autonomy, a methodology of distributed artificial intelligence, was the benchmark for the methods and concepts which were used to represent the evolution of system reliability over time. The agent-oriented distributed modeling approach originated from computer science and artificial intelligence (AI), where intelligent computer agents are software programs designed to act autonomously and adaptively to achieve goals defined by the developer or runtime users [46] - [62]. The following aspects of computer agent autonomy helped to build a foundation of the PoF-based agent autonomy for system reliability modeling that is proposed in this research:

1. Decentralization of control and decision making is considered a paradigm shift and a future direction of research in computer science. Decentralization capability is critical because it provides flexibility and adaptability of

computer software in runtime in order to react accordingly to structural modifications in runtime. It also considers all time-dependent interactions and interdependencies between the system elements. A system of computer-based agents as autonomous decision makers is the most efficient solution proposed to date.

2. Each computer-based agent is capable of settling on its state evolution autonomously and without interference of the environment or other agents.
3. Each computer-based agent is able to sense the environment and collect the information which is critical for its internal processes.
4. Each computer-based agent shares its properties and the current state with other agents.

The key focus of multi-agent system development is on the flexible behavior of computer agents related to the above listed expectations. It is agent flexibility which makes the agent-oriented approach a valuable choice of modeling methodology for dynamic systems. Although there is no universally accepted definition of a computer agent, most authors agree on the following concepts: each agent is autonomous, has a set of goals, has a local model of the part of the world that affects the achievement of its goals, and has a way of communicating with other agents. The following properties of computer agents, described in the literature sources [46], [49], [50], [51], [57], [59], were considered in this research in development of the PoF-based agent autonomy:

1. Autonomy in Action
2. Intelligence

2.1. Internal Knowledge (Rules of Behavior, Memory, Goals, Plans)

2.2. Reactivity

2.3. Reasoning / Learning (Adaptability)

2.4. Proactivity, Goal Orientation

3. Communication, Social Activity, Cooperation

4. Mobility

This generally accepted categorization of agent properties was summarized by M. Azarkhail in his dissertation [1] and will be further addressed within the current research.

A probabilistic aspect of computer-based agents within agent autonomy was considered in several publications [53] - [56], [60], [61]. The Bayesian belief network (BBN) methodology was utilized to introduce a probabilistic aspect into sharing raw data and analysis results (probability distributions) among the computer agents. Another work [53] proposed Bayesian updating framework to support agent learning and decision making tasks in multi-agent computer systems used for control of complex industrial processes. Bayesian formalism was applied to incorporate the uncertainty through agent representation by probability density functions used in Bayesian inference.

1.4. Motivation

Based on the discussion in previous sections, a new approach to system reliability modeling should be able to incorporate physics-of-failure knowledge about the systems elements, including the following capabilities:

1. The ability to capture physics of degradation and failure and describe the system degradation processes according to failure mechanisms of hardware parts and components. This includes software reliability, human interactions with the system and the resulting scenarios of the system evolution in time.
2. The capability to include interactions between the failure mechanisms of the system elements (i.e. two or more simultaneous failure mechanisms affecting each other's propagation rate and resulting in one complex mechanism at the hardware part, component or system level).

The above objectives and the following baseline concepts defined the choice of a novel approach to system reliability modeling developed in this research:

1. In any engineering system, hardware components and sub-systems are physically distributed, have their own properties and rules of behavior, and an ability to influence the final state of the system. One may consider it as an intelligence within the component that autonomously responds to changes (in usage conditions and in adjacent components) by managing its properties and behaviors and making appropriate decisions on its final state (i.e. success, degraded performance or failure). Evolution of hardware components, sub-systems and the entire system is affected by software reliability and interactions with human elements. The latter possess certain properties and rules of behavior when acting upon the given system. As such, the modeling procedure implies a distribution of the failure knowledge (i.e. intelligence) among all elements of the system.

2. Knowledge of the first principles and physics of failure fundamentals allows the modeler to set rules and conditions for the behavior of system elements (hardware, software, human elements) and anticipate the resulting degradation and failure events.

1.5. Agent Autonomy

In this research the agent autonomy is used as a solution method for probabilistic physics-of-failure modeling of reliability of complex engineering systems with interacting failure mechanisms. The agent-oriented distributed modeling approach originated from computer science and artificial intelligence (AI), where intelligent agents were software programs designed to act autonomously and adaptively to achieve goals defined by the developer or runtime users. In system reliability modeling, however, agent hierarchy, classification, and properties of agents are different from those in computer science and artificial intelligence.

In this research an agent is a computer replica of any parameter, characteristic or feature of the system element (hardware part or component, software, human element) or usage profile (environmental or operational conditions, mission attributes, inspection and maintenance program, etc.). This piece of software contains all properties of the respective element, mimics how that element evolves over time, and shares information with other agents. The agent structure of engineering systems is tailored to the dynamics of physical and chemical degradation and failure processes of system elements under variable usage conditions.

1.6. Contributions of this Research

The contributions of this research can be summarized as follows:

1. Introduced a new classification of agents and agent hierarchy within the scope of system reliability modeling: Type I Micro-Agents, Type II Macro-Agents, and Type III Monitoring Agents. Three classes of agents are defined according to the different types of entities within the physical processes of degradation and failure at all levels of system hierarchy, from materials and piece parts to components and the entire system, also considering software and human elements. This also included the identification of the properties of each class of agents, considering the physical characteristics of their counterparts in the real system and their role in system evolution.
2. Developed agent representation which presents a new approach to modeling complex interdependency and interactions between failure mechanisms of different elements of a system, specifically where the degradation process in one element (part, material or component) activates or accelerates the failure mechanisms of other elements. The associated terms are explicitly introduced into the PoF or empirical models of the system elements and their respective agents. This allows for bidirectional communication between agents where interacting failure mechanisms exist.
3. Proposed a new definition of the learning property of intelligent agents. Developed guidelines for the selection of the agent learning algorithm depending on the agent class, availability of the PoF or empirical model of the

system element represented by the agent, and the types of data used for agent learning.

4. Introduced a new definition of the key property of intelligent agents - autonomy in action. Autonomy was defined as agent's ability to activate and deactivate itself during system evolution. There is also a proposed algorithm of agent activation and deactivation based on the most appropriate methods of uncertainty importance and sensitivity analysis. This new definition of the autonomy is a "core" contribution of this research.
5. Identified the key distinctions between the agent autonomy and the existing methods of system reliability modeling. The autonomy property of intelligent agents and the capability of the agent autonomy to model interacting failure mechanisms of system elements make the agent autonomy fundamentally different from all existing methods of probabilistic PoF-based reliability modeling and simulation. These features bring more reality into reliability models, giving the agent autonomy an advantage over the traditional methods of system reliability modeling.

1.7. Outline of this Dissertation

The remainder of this dissertation is organized into eight chapters. Chapter 2 defines research objectives and the choice of research approach. In Chapter 3, the definition of agents and agent classification is developed. Next, agent classes and their role in PPoF system model are described. Chapter 4 outlines the agent properties, while Chapter 5 and Chapter 6 are focused specifically on the learning property of agents. Chapter 7 provides detailed definition of the autonomy property of agents along with considerations of uncertainty characterization within the agent autonomy modeling. Chapter 8 presents a case study of a gas turbine aircraft engine structures as an application example. The conclusions and suggested future research are summarized in Chapter 9.

Chapter 2: Research Approach

2.1. Concept Definition and Objectives

Consider the hardware system of several interconnected components. Figure 2-1 depicts a hierarchy of hardware system elements in the context of PoF, as proposed in [44]. Failure of such system relates to the failures of its components, which comprise several piece parts and materials. In turn, piece parts and materials fail due to specific failure mechanisms which are driven by several stress-strength and degradation-endurance factors influenced by operational and environmental conditions of the given system.

Operational and environmental conditions and subsequently stress-strength and degradation-endurance factors (Figure 2-1) could be introduced deterministically or probabilistically depending on the degree of uncertainty. They are further used to build PPoF models of the individual failure mechanisms at the piece part/material level as a combination of the scientific knowledge of degradation processes and the uncertainties of operational variables, material properties and environmental conditions. Every item and feature in the system hierarchy in Figure 2-1 is further replaced by an intelligent autonomous software agent acting according to the system logic towards the final states of the system. Such agents are autonomously evolving over time, having access to the status of other agents and ability to intelligently react to any circumstances that may occur during the course of system operation within the given environment.

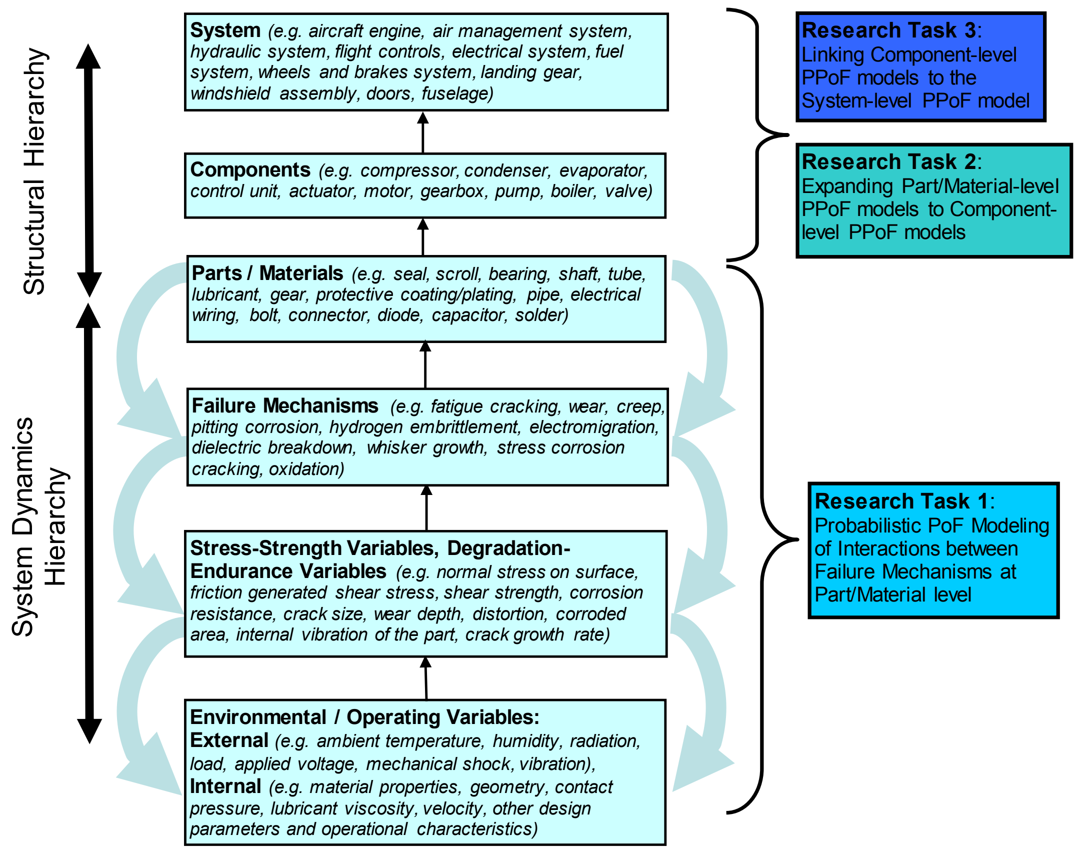


Figure 2-1: Conceptual System Hierarchy and Failure Dynamics

The objective of this research is to develop an agent-oriented approach to system reliability modeling as a hierarchy of intelligent autonomous agents powered to accomplish the following three tasks, as depicted on Figure 2-1:

1. Develop representation of interactions between failure mechanisms at the piece part/material level within the PPoF modeling scheme of item degradation over time.
2. Expand piece part/material-level PPoF models of degradation and failure processes to the component-level PPoF modeling framework, accounting for interactions between failure mechanisms of various piece parts / materials and components.

3. Link component-level PPoF models into the system-level PPoF representation of degraded states of the system and system failures, considering interactions between failure mechanisms at all levels of system hierarchy.

Monte Carlo based sampling will be used to combine PoF knowledge about piece parts/material and components within direct simulation to assess system reliability. Using this approach, the probabilities of possible system states are sampled based on state probabilities of all different variables included in PoF models of piece parts / materials and components, yet consistent with dynamic configuration of the system. As such, system level reliability modeling becomes as simple as checking the status of system elements at any given time. This way, the most relevant system model will be available at every "snapshot" in time. The same concept applies to any engineering system containing not only hardware components, but also software and human elements.

2.2. Research Approach

The concept of the intelligent agent-oriented approach to reliability modeling was first introduced by M. Azarkhail [1]. In that research, a direct and efficient intelligent agent-oriented simulation is proposed to model the reliability of a long-term complex dynamic system. In this system simulation, every component of the system is replaced by an intelligent piece of software that represents the properties and behaviors of its real counterpart from the system. These software agents act autonomously to mimic their counterparts in real system. The failures are simulated using Monte Carlo-type methods applied to a system of intelligent computer agents.

That computer-based direct simulation is made to account for the dynamic failure logic, design, and control characteristics of the system. The software agents are meant to support the failure logic of the system that evaluates the final state of the complex system, given the final state of the components. The agent-oriented approach also assumes that there are always enough PoF models that explain the underlying failure phenomena of the system components and piece parts.

The current research is intended to expand M. Azarkhail's approach [1] by developing a universal, structured framework of agent autonomy supporting the PoF-driven modeling of complex engineering systems with competing and interdependent failure mechanisms. The scope of this research includes the definition of agents, the development of the multi-agent system hierarchy, and system reliability modeling techniques. As identified in Section 2.1, the developed agent autonomy should allow modeling interdependent and interacting failure mechanisms within the dynamic system.

Consider a system built of hardware parts and components (Figure 2-2) where each component contains one or more piece parts. Each piece part is decomposed into failure mechanisms. Failure mechanisms as real causes of failures can be linked to certain physical or chemical degradation processes developing within the piece part over its life. Stress-strength and degradation-endurance variables are responsible for the progression of failure mechanisms and linked to the operational and environmental conditions (coupling factors) by PoF equations of physical or chemical degradation processes (enablers), as shown on Figure 2-2. Part reliability measure is life to failure (or time to a certain degree of degradation) given by physical model of stress/strength or degradation/endurance vs. life which obtains its probabilistic representation where uncertainty exists.

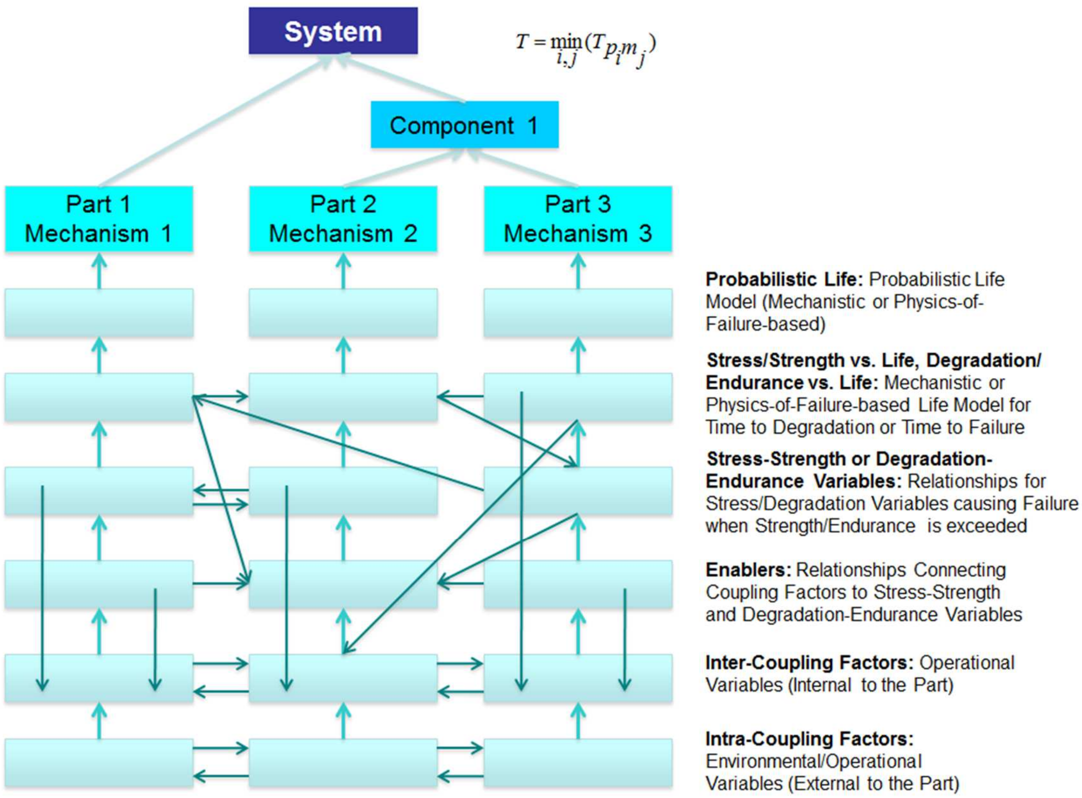


Figure 2-2: Probabilistic-Mechanistic Framework of System Reliability

Part interdependency arises from the exposure to the same environmental and operational conditions, as well as due to the same design, materials, components, location in the system or other factors that lead to common cause failures. Interdependency also means that the degradation process in one piece part could be influenced by the failure mechanisms of other piece parts, resulting in cascading failures and possibly altering the physical nature of the progressing failure mechanisms of one or more piece parts due to load redistribution or other changes in the operational conditions. This is especially important in modeling complex components or systems with several competing failure mechanisms which are progressing at all levels of the system hierarchy, as shown in Figure 2-2. The elements of the probabilistic-mechanistic framework of system reliability, from coupling factors to the PPoF life model, are defined on the right side of the diagram in Figure 2-2, while the arrows depict interdependent elements.

Depending on the depth of engineering knowledge about the degradation and failure processes, failure mechanisms could be defined at the component or at the sub-system level, while PoF equations of degradation and the PoF life model could be replaced by an alternative empirical representation. In addition, the same concepts as described above for a hardware system (shown in Figure 2-2) apply to any engineering system containing software and human elements in addition to hardware parts and components.

Figure 2-3 provides an example of a probabilistic-mechanistic framework of system reliability for an aircraft cargo door system. Reliability of the positioning mechanism, for instance, is driven by degradation and failure measures of three components: life to fatigue failure for the roller guide fitting, fatigue crack size for the roller guide, and damage due to wear for the roller. Dashed lines in Figure 2-3 show interactions between the failure mechanisms of the system elements. For example, the progression of the roller wear leads to the increased vibration of the roller, which in turn, impacts the shear stress of the roller as well as the cyclic stress of the roller guide. The PoF model of roller guide fatigue mechanism (equation of fatigue crack size) and the PoF model of roller wear (equation of wear depth) should include additional terms corresponding to the roller vibration.

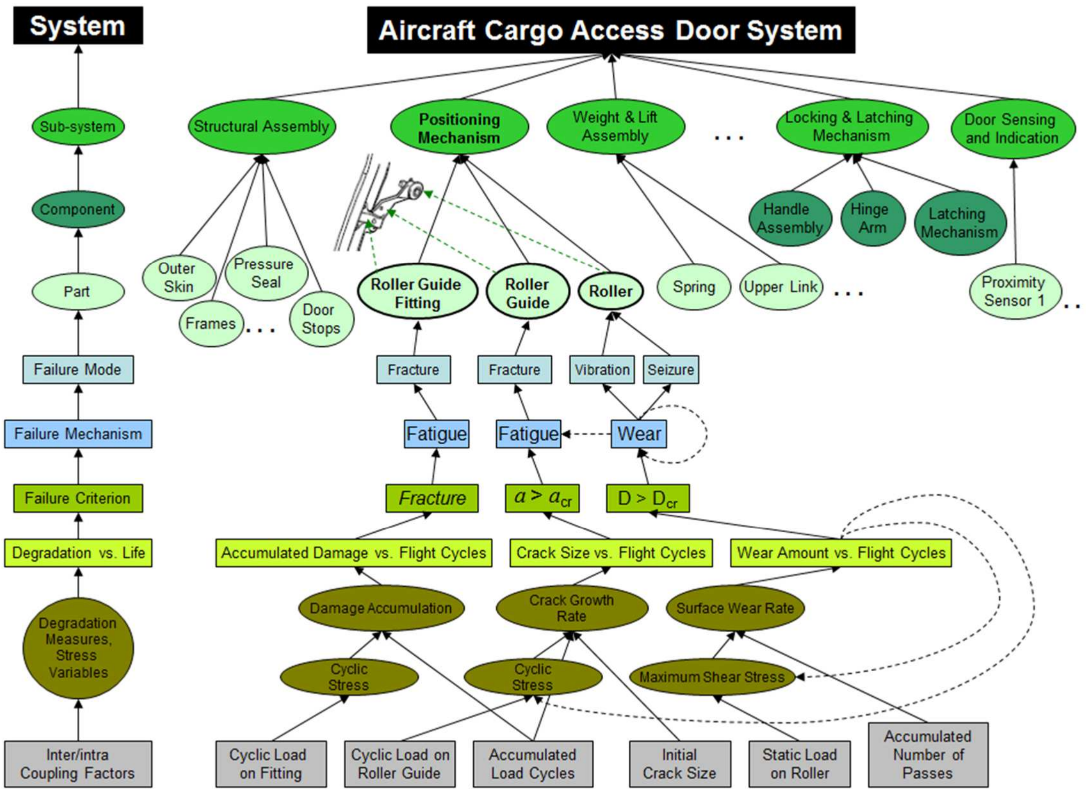


Figure 2-3: Example of System Hierarchy for Positioning Mechanism of Aircraft Cargo Door

An example of the probabilistic-mechanistic life model of a hardware part, particularly time-to-failure model of a ball bearing, is given by Figure 2-4. This PPoF model was developed from physical principles considering critical variables that contributed to the failure of ball bearings. The obtained PPoF model was then converted into a PPoF agent-oriented representation of bearing reliability.

In order to integrate a PPoF representation of system reliability into an agent-oriented framework, each element of the system hierarchy is replaced by an agent designed with a level of intelligence that allows it to handle its needs in the system.

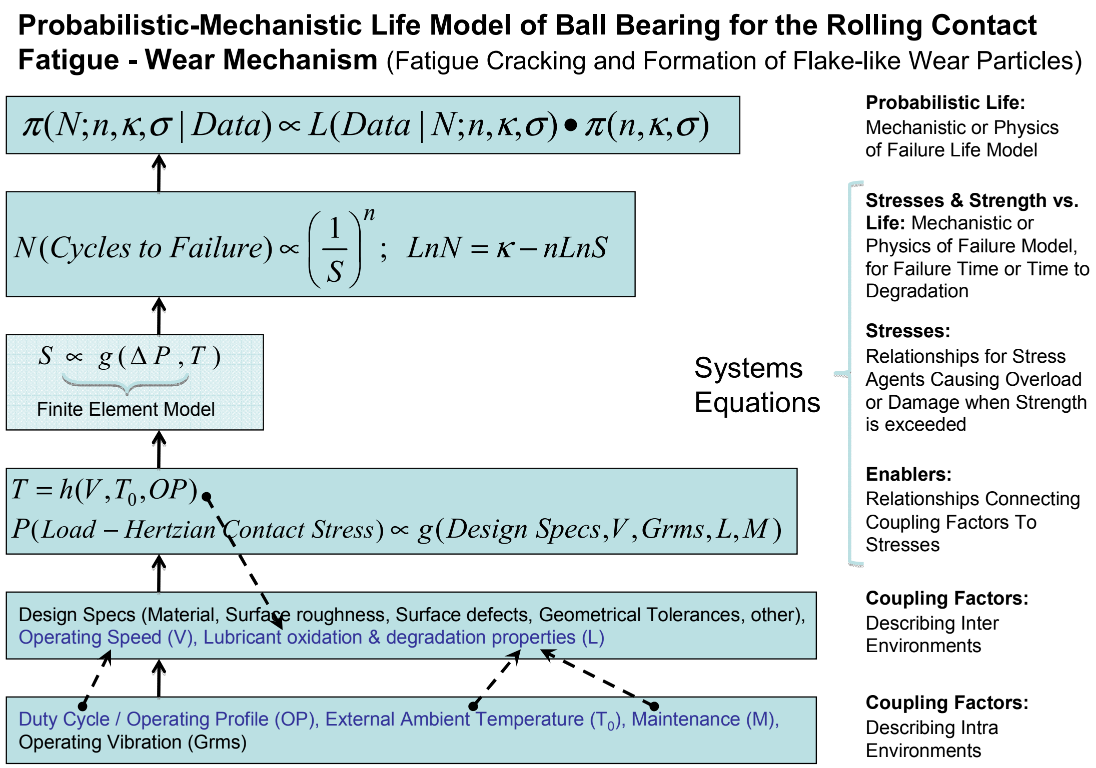


Figure 2-4: Probabilistic-Mechanistic Reliability Life Model of Ball Bearing

Similar to computer science and artificial intelligence, in system reliability modeling, agents are viewed as intelligent autonomous entities that are capable of effective operation over time in dynamic environments. One of the main reasons for introducing the agent-oriented approach in both areas is the necessity to model all aspects of system dynamics. This means that a complex system comprised of many interacting elements in changing environments should be able to react accordingly to any changes in the components and environments in runtime as well as consider all time-dependent interactions and interdependencies between the system elements, and to do so autonomously and based on agent roles and the rules of system evolution. One of the key tasks of this research, therefore, is modeling two-way interactions (shown by dashed lines in Figure 2-4) as “feedback loops”: while the lower level simple variables (such as coupling factors) are the inputs to much more complex variables at higher levels of the hierarchy (such as stress-strength and degradation-endurance variables), the latter may impact the state of lower level variables.

A multi-agent approach to reliability modeling is explained in the next chapter and the structure of agents replacing each element of the engineering system is discussed along with definition and classification of agents and their properties. The hierarchy of autonomous intelligent agents and agent properties has been developed in this work based on the concepts of computer agents described in Chapter 1, the definitions of agent autonomy for reliability modeling introduced by M. Azarkhail [1], and the objective of the PoF representation of degradation and failure processes of dynamic engineering systems with interacting failure mechanisms.

It must be noted that, while agent definition and classification developed in this research makes provisions for representation of all elements of the complex engineering system (hardware, software and human factors), the concepts of agent autonomy are demonstrated only for hardware parts and components in order to limit the scope to a reasonable size.

Chapter 3: Definition and Classification of Agents

3.1. Definition of Agents in the Context of System Reliability

This research defines an agent as a computer replica of any element of a system operational profile, such as characteristic or feature of a hardware part, component, software program, or human element, environmental factor or operational parameter, an attribute of inspection and maintenance program or mission profile. This computer replica is developed by a modeler based on the physical principles of degradation and failure of system elements operating in the context environment. It contains all the properties of the respective variable, mimics changes of that variable over time, and communicates with other agents. To accomplish these tasks, each agent is structured to have a single output variable as a function of one or more input variables. Each input variable represents another agent, specifically its output variable. The general form of the agent output variable can be written as:

$$Y = f(X_1, X_2, \dots, X_n)$$

Equation 3-1

where Y denotes a single output variable of an agent as a function of input variables $X_1, X_2, \dots, X_n$. The function f(·) is a PoF-based or empirical model of the agent output variable, which can be either probabilistic or deterministic. Such agents are also called "multi-agents" because their evolution depends on the inputs from several other agents and inter-agent interactions during system degradation over time. There could also be agents which do not obtain any inputs from other agents. These are "self-sufficient agents", agents that are not dependent on other agents for their own development. Such agents have their independent output variables in the form of probability density functions.

It must be noted that an output of any multi-agent is always probabilistic due to uncertainties related to a function, f(·), the input variables X_i, or both. An output of any self-sufficient agent is also probabilistic in accordance to the applicable probability density function (with fixed or uncertain parameters). Any characteristic or parameter given by a constant value should be treated as a constant and should not be assigned with an agent. If a certain quantity is deterministic but changes during system evolution in a known fashion (e.g. time variable) or according to the deterministically defined function of time (or other non-randomly varying quantity), such quantity is identified as an agent.

Agent classes and their relationship to the elements of a physical system in the context of system reliability modeling are defined further in this chapter.

3.2. Classification of Agents in Reliability System Modeling

Different levels of agents can be defined, depending on the area of application of that intelligent agent autonomy. In order to allow modeling the progression of interdependent and interacting failure mechanisms at all levels of system hierarchy (Figure 2-2 and Figure 2-4), it is logical to identify types of agents according to the degree of dependency between various elements of the probabilistic-mechanistic framework of system reliability. This would include everything from coupling factors to the PPoF life model. Three classes of intelligent agents are proposed in this research for the agent-oriented reliability modeling of engineering systems: Type I

Micro-Agents, Type II Macro-Agents and Type III Monitoring Agents. Sections 3.2.1 to 3.2.4 provide detailed definitions of three classes of agents and include a discussion on why this classification is important and novel for PoF-based system reliability modeling.

3.2.1. Type I Micro-Agents

Agent classification starts with the highest granularity of agent autonomy – Type I Micro-Agents. Agents are assigned not at sub-system or module level, not even at component level, but to every feature and internal characteristic of a piece part, material, component, software, human element, and each external parameter affecting the degradation and failure mechanisms of system elements. In the context of the probabilistic-mechanistic framework of system reliability shown in Figure 2-2, Type I Micro-Agents represent both inter- and intra-coupling factors.

Type I Micro-Agents include five subclasses, Group A to Group E, in order to distinguish coupling factors of different natures, such as environmental and operational factors, material properties, design characteristics, performance parameters, mission attributes, software design features, and various aspects of human elements.

Type I Micro-Agents are introduced to represent the physical variables that lead to degradation and failure (such as environmental and operational conditions) or impact the nature of life limiting failure mechanisms and rate of degradation/failure (such as material properties, design characteristics, duty cycle and usage profile). These physical variables affect various elements of the system during system evolution. This implies that each Type I Micro-Agent may be used as an input to several next level agents, specifically Type II Macro-Agents described in Section 3.2.2. Such representation allows expressing the interdependency between the system elements which arises from the same design features, same functional requirements, or the exposure to the same environmental and operational conditions. Type I Micro-Agents are needed to identify the "drivers" of degradation and failure processes within the PoF-based system reliability model.

Some examples of Type I Micro-Agents from each of five groups are shown in Table 3-1, Table 3-2, Table 3-3, Table 3-4 and Table 3-5. These tables are intended to serve as Type I Micro-Agents classification guidelines and do not provide a complete list of the elements as potential agents or impose any strict rules regarding agent assignment within the Type I Micro-Agents class.

Table 3-1: Classification of Agents – Type I Micro-Agents, Group A

Type I. Micro-Agents
Group A. Usage Stress Variables *(Intra-Coupling Factors)* ***Variables External to Hardware, Software, Human Elements***
Group A1. Environmental Factors
1.Temperature
2. Thermal Cycling Range
3. Humidity
4. Moisture/Water Ingress
5. Icing/Fog/Rain
6. Concentration of reactive substances (salt, acids and bases)
7. Dust, Dirt, Sand
8. Grease, Oil, other contaminants
9. Radiation
10. Lightning
11. Atmospheric Pressure
12. Wind Speed
13. Earthquake Strength
14. Environment Factor Rating (Environmental Designation) for Hardware Examples: *GB (Ground Mobile), AA (Airborne Attack), AIF (Airborne Inhabited Fighter)*

Group A2. Operational Conditions, Maintenance Characteristics, Logistics
1. Voltage
2. Power
3. Pressure
4. Pressure Cycling/Pressure Impulse
5. Vibration, Random or Sinusoidal
6. Mechanical Load (e.g. Static Load, Amplitude of Dynamic Load)
7. Mechanical Shock
8. Acceleration
9. Electromagnetic Impact
10. Operating Speed
11. Altitude
12. Preventive/Corrective Maintenance (interval, goodness of repair/restoration factor)
13. Software Application Type (Airborne, Strategic, Tactical, Process Control, other)
14. Software Development Environment
15. Software Test Coverage
16. Number of Software Test Cases
17. Software Fault Exposure Ratio
18. Goodness of Repair (Repair Effectiveness)
19. Inspection Interval
20. Frequency of Preventive Maintenance
21. Elements and Parameters of Preventive Maintenance Activities (tests, measurements, adjustments, application of lubrication solution, parts replacement)
22. Repair Rate
23. Weight of Passengers, Cargo

Table 3-2: Classification of Agents – Type I Micro-Agents, Group B

Type I Micro-Agents
Group B. Exposure Time and Mission Parameters *(Intra-Coupling Factors)* ***Mission Variables External to Hardware, Software, Human Elements***
1. Accumulated Missions since Installation, Entry into Service, Overhaul, other milestone
2. Accumulated Mission Hours or Operational Hours since Installation, Entry into Service, Overhaul, other milestone
3. Calendar Time or Number of Cycles since Installation or since Operation Start
4. Mission Profile Parameters (Mission Duration, Phases, Duration of each Phase)
5. Duty Cycle of Hardware, Equipment Utilization Frequency, Duration of On/Off Cycle Examples: *Duty Cycle is 50% (1 Operating Hour = 2 Mission Hours); Duty Cycle is 100% during Cruise Phase and 0% during Landing Phase*
6. Software Operational Profile/Usage Frequency
7. Number of Cycles per Mission Phase
8. Time in Storage/Inspection/Transportation/Assembly

Table 3-3: Classification of Agents – Type I Micro-Agents, Group C

Type I Micro-Agents
Group C. Hardware Characteristics *(Inter-Coupling Factors)* ***Variables Internal to Hardware***
Group C1. Design Parameters and Manufacturing Characteristics
1. Material properties Examples: *Material Grade, Strength characteristics, Surface roughness, Surface defects, Hardness, Elongation, Microstructure, Lubricant type, Fluid viscosity, Additives, Material Constants (various)*
2. Shape/Geometry/Dimensions and Tolerances
3. Design characteristics (various)
4. Item Type, Grade, Style, Category, Quality Factor, other factors related to the item Examples (from MIL-HDBK-217F): *Resistor Type (Fixed, Film, Insulated), Resistance Factor, Capacitor Type (Paper, By-pass, Filter, Blocking, DC), Capacitance Factor, Quality Factors*
5. Configuration Attributes Examples: *Number of parts/components of a certain type, location of parts/components in the assembly*
6. Manufacturing Process Attributes Examples: *Manufacturing technology type, process grade, stress screening sample size, end of line testing applied, inspection frequency, category of acceptance test procedure*
Group C2. Performance Parameters and Functional Characteristics
1. Voltage/Voltage Range
2. Power
3. Pressure (gas, fluid)
4. Pressure Impulse
5. Vibration
6. Mechanical Load
7. Torque
8. Transition Rate (e.g. Heat Release Rate, Reaction Rate, other)
9. Thermal Expansion
10. Heat Dissipation
11. Operating Speed
12. Acceleration
13. Processing Time
14. Efficiency
15. Capacitance
16. Resistance
17. Current/Current Density
18. Impedance
19. Noise
20. Distortion
21. Dimensions (Linear, Angular)
22. Cycling Frequency
23. Intensity (of Signal, Light, etc.)
24. Flammability

Table 3-4: Classification of Agents – Type I Micro-Agents, Group D

Type I Micro-Agents
Group D. Software Design Parameters *(Inter-Coupling Factors)* ***Variables Internal to Software***
1. Number of LOC (Lines of Code)
2. Modularity (Number or Complexity of Modules)
3. Language Type
4. Instruction Rate
5. Number of Object Instructions in the Program
6. Number of Errors Fixed in Time Interval
7. Number of Faults Experienced in Test Case(s)

Table 3-5: Classification of Agents – Type I Micro-Agents, Group E

Type I Micro-Agents
Group E. Human Factors *(Inter-Coupling Factors)* ***Variables Internal to Human Elements***
1. Performance Shaping Factors (PSF) of Human Behavior Examples: *Experience/Knowledge, Psychological Stress, Safety & Quality Culture, Non-task Related Load, Shift Handover, Fatigue, Procedure Availability, Procedure Quality*
2. Organizational Factors Examples: *Structural Factors (related to organization structure and resource allocation), Behavioral Factors (related to responsibilities on the job, objectives, management commitment to reliability and safety of the product, methods and processes, training, performance measures, work compensation, etc.)*
3. Technology Factors Examples: *Technological Complexity Factors (related to the degree of technological advancements of product commodity), Engineering Knowledge and Expertise Factors (related to engineering experience with the given commodity in the organization)*

According to the above classification, Type I Micro-Agent is the simplest type of agents. Type I Micro-Agent may have no input from other agents. For example, if an ambient temperature agent is normally distributed, $T \sim N(\mu, \sigma)$, then temperature, T, is the output variable of this agent, but there are no agents providing input variables to this ambient temperature agent. Otherwise, Type I Micro-Agents are dependent on other Type I Micro-Agents and Type II Macro-Agents as inputs and have a functional form defined by Equation 3-1.

3.2.2. Type II Macro-Agents

More complex agents at higher abstraction level are Type II Macro-Agents. In the context of the probabilistic-mechanistic framework of system reliability, shown on Figure 2-2, Type II Macro-Agents represent stress-strength and degradation-endurance variables and life to failure (or time to degradation). Type II Macro-Agents can have any number of inputs from other agents. The simplest Type II Macro-Agent represents an independent random variable which has no inputs from other agents (for example, time to failure developed from pass/fail test data as Weibull probability distribution). More complex Type II Macro-Agents are expressed as a combination of two or more Type I Micro-Agents via the PoF model or the empirical function generally defined by Equation 3-1. The most complex Type II Macro-Agent may combine several Type I Micro-Agents, Type II Macro-Agents and Type III Monitoring Agents in a similar manner. In order to support the objective of PoF-based modeling of system reliability, input agents should be combined into a Type II Macro-Agent by means of a PoF model as a mathematical relationship which is derived from the physical principles of degradation and failure of the associated element (such as hardware part or component). Empirical functions should only be used where PoF model is not available or where it cannot be utilized due to lack of knowledge.

General examples of Type II Macro-Agents are given in Table 3-6. Note that this table is not a complete list of potential Type II Macro-Agents, but rather the modeler's guideline for assignment of Type II Macro-Agent classification.

Table 3-6: Classification of Agents - Type II Macro-Agents

Type II Macro-Agents *(Stress and Strength Variables, Time to Degradation or Damage Accumulation, Life to Failure)* ***Variables describing Failure Mechanisms of Hardware, Software Program or Human Elements***
1. Life to Failure or Time to Critical Degradation
2. Degradation Measures - direct (*crack size, distortion, wear depth, corroded area, etc.)*
3. Degradation Measures - indirect (*various failure precursors*)
4. Rate of Degradation (*wear rate, crack growth rate, creep rate, etc.*)
5. Strength Characteristics (*tensile strength of structural part, creep strength, rated maximum operating temperature, rated voltage, corrosion resistance, etc.*)
6. Stress Parameters (*mean or amplitude of tensile/bending/torsional stress, stress intensity factor, current density, etc.*)
7. Failure Rate or Failure Probability of Hardware Part or Component (*failures per operating hour, probability of failure on demand, cumulative failure probability, etc.*)
8. Rate or Probability of Software Faults (*probability of software faults within time interval, number of remaining software errors at a given time, etc.*)
9. Rate or Probability of Human Errors (*probability of human error in a certain action (task) upon a part, component or system, accumulated number of human errors by a given point in time, etc.*)

The fundamental difference between Type I Micro-Agents and Type II Macro-Agents lies in the type of variables that these agents represent. Type I Micro-Agents are assigned to the physical variables that lead to degradation and failure. Type II Macro-Agents are needed to accomplish the main objective of the research – to introduce physics-of-failure knowledge into the system reliability model. Type II Macro-Agents are assigned to the degradation and failure characteristics of the system parts and components, such as mechanical stress, wear depth or life to failure. Each degradation or failure characteristic is a physics-of-failure-based function of several other degradation or failure characteristics, environmental and operational conditions, and/or other variables. This implies that several Type II Macro-Agents may share the same inputs from Type I Micro-Agents and Type II Macro-Agents, and may become inputs to other Type II Macro-Agents or Type I Micro-Agents. Such representation allows linking the interdependent elements of system hierarchy and modeling interactions between failure mechanisms.

As such, a commonality between the multi-agents of Type I and Type II is in the representation of their output variables as functions of the output variables of other Type I Micro-Agents and Type II Macro-Agents. Similarly, the self-sufficient agents of Type I and Type II have their independent output variables given by probability distributions. This implies that the probability distribution of the element represented by an agent depends on the level of detail selected by the modeler and the uncertainty about the functional form of the agent output model and about each input variable (for the dependent agents, or multi-agents). The dependence of agent output uncertainty on the granularity of the agent representation via other agents was the main driver of the agent classification process presented in this research. This classification is intended to be flexible enough to model engineering systems of any composition and complexity.

As an example of the definition of Type II Macro-Agent, consider the Type II Macro-Agent life to fatigue failure of structural part, L_F, given by Equation 3-2 as a function of cyclic stress:

$$L_F = A(\Delta S)^n \Rightarrow \ln(L_F) = A + n \cdot \ln(\Delta S)$$

Equation 3-2

The model parameters, A and n, could be represented by probability distributions or as constant values. Applied alternating stress, ΔS, is the input variable (from the cyclic stress agent), and life to fatigue failure, L_F, is the output variable. In turn, ΔS

could be a Type I Micro-Agent, defined deterministically (e. g., constant amplitude during each phase of the mission) or probabilistically (if variation is present). The agent ΔS could also be structured as a Type II Macro-Agent if it depends on other operational factors as inputs (e.g. pressure amplitude, vibration, temperature, thermal cycles, etc.) or if it is a function of time.

3.2.3. Type III Monitoring Agents

Type III Monitoring Agents are intended to "monitor" the status of each hardware part, component, and system and communicate it as an output variable in a form of probabilistic measure of degradation or failure, such as:

- Degree of damage accumulation (DA),
- Remaining useful life (RUL),
- Probability of reaching critical degradation limit,
- Probability of failure.

Type III Monitoring Agents are assigned to each part and each component of the system and to the system itself, as summarized in Table 3-7. Each Type III Monitoring Agent is expressed as a combination of one or more Type II Macro-Agents and/or other Type III Monitoring Agents. These Type II Macro-Agents and Type III Monitoring Agents are the input agents to the Type III Monitoring Agent of interest. The output variable of the Type III Monitoring Agent has a functional form defined by Equation 3-1, where the function $f(\cdot)$ is a deterministic or probabilistic empirical model of the agent output variable which is developed according to degradation and failure logic of the item it represents (hardware part, component, or the system). For example, deterministic failure logic of the item could be depicted by a fault tree or Bayesian belief network, and the function, f(·), becomes a logic equation for a failure event (or reaching critical degradation limit). Degradation logic could also be given by a deterministic or probabilistic expression of the degree of damage accumulation or the remaining useful life according to the particular application.

Table 3-7: Classification of Agents – Type III Monitoring Agents

Type III Monitoring Agents
Group A. Part Monitoring Agents
1. Remaining Useful Life (RUL) at a given point in Time
2. Degree of Damage Accumulation (DA) by a given point in Time
3. Probability of Reaching Critical Degradation Threshold at a given point in Time
4. Probability of Failure, Operational Availability, Reliability (Probability of Success) at a given point in Time
5. Mean Time To Failure (MTTF), Mean Time Between Failures (MTBF), Mean Time To Repair (MTTR) at a given point in Time
6. Failure Count within a given Time Interval
Group B. Component Monitoring Agents
1. Remaining Useful Life (RUL) at a given point in Time
2. Degree of Damage Accumulation (DA) by a given point in Time
3. Probability of Reaching Critical Degradation Threshold at a given point in Time
4. Probability of Failure, Operational Availability, Reliability (Probability of Success) at a given point in Time
5. Mean Time To Failure (MTTF), Mean Time Between Failures (MTBF), Mean Time To Repair (MTTR) at a given point in Time
6. Failure Count within a given Time Interval
Group C. System Monitoring Agent
1. Remaining Useful Life (RUL) at a given point in Time
2. Probability of Reaching Critical Degradation Threshold at a given point in Time
3. Probability of Failure, Operational Availability, Reliability (Probability of Success) at a given point in Time
4. Mean Time To Failure (MTTF), Mean Time Between Failures (MTBF), Mean Time To Repair (MTTR) at a given point in Time
5. Failure Count within a given Time Interval

Type III Part Monitoring Agents are structured as a combination of Type II Macro-Agents associated with failure mechanisms of the given hardware part, aggregating information from Type II Macro-Agents according to the part degradation and failure logic. Type III Component Monitoring Agents collect information from Type II Macro-Agents and Type III Part Monitoring Agents associated with PoF of the given hardware component, and aggregate the input agents information according to the component failure logic or according to the definition of degraded states of the component. Type III System Monitoring Agent collects information from Type III Component Monitoring Agents, Type III Part Monitoring Agents and Type II Macro-Agents, and aggregates this information into the system reliability measure according to the system degradation and failure logic.

Type I Micro-Agents and Type II Macro-Agents representing software and human elements are the inputs into the associated Type III Monitoring Agents, providing information about software programs or human elements that interact with system hardware and impact the status of hardware parts, components and the entire system reported by Type III Monitoring Agents.

A probabilistic representation of the output variable of Type III Monitoring Agent is obtained by means of simulation using deterministic or probabilistic function $f(\cdot)$ and probability distributions of the input agents.

To summarize, Type III Monitoring Agents are introduced to render the system reliability assessment (RUL, DA) that is an ultimate objective of the agent-oriented modeling. This assessment relies on the input from degradation and failure characteristics of the parts and components (represented Type II Macro-Agents) under the influence of physical variables that lead to degradation and failure (defined by Type I Micro-Agents).

3.2.4. Agent Hierarchy

It is shown that the proposed agent classification supports representation of all three main elements involved in the operation of engineering systems: hardware parts and components, software programs and human actions. Multilevel agent classification presented in this section was also developed to allow modeling of any level of system complexity, from a system consisting of one piece part to a complex combination of multiple hardware parts and components controlled by software programs and subjected to human actions. Further, each element of an engineering system could be represented at any level of detail preferred by the modeler, starting from the lowest level of system hierarchy, such as environmental and operational conditions, manufacturing process parameters, maintenance schedule, material properties and geometrical dimensions of hardware piece parts, number of lines in a software code, characteristics and behavioral patterns of human actions. Type I Micro-Agents are used to represent such variables. Every aspect of the degradation and failure mechanisms of hardware, software faults and human actions are modeled by Type II Macro-Agents which represent complex, dependent characteristics of system elements. Interactions and interdependency between system elements are explicitly expressed within the structure of the associated Type I Micro-Agents and Type II Macro-Agents via the Type I Micro-Agents and Type II Macro-Agents they share as inputs. This shows that the proposed agent structure and classification scheme allows bidirectional communication between Type I Micro-Agents and Type II Macro-Agents where complex interdependencies between failure mechanisms of the system components exist. Type III Monitoring Agents define the degradation or failure status of the hardware parts, components and the entire system, considering the evolution of all system elements and their interactions.

The resulting agent hierarchy may combine agents of all classes for an efficient mechanistic representation of complex engineering systems that comprise hardware parts and components in combination with software and human elements, as shown on Figure 3-1. The arrows indicate that agents exchange information by communicating the required inputs to each other. This information exchange is a key feature of agent autonomy because it allows modeling of interactions and interdependency between system elements. Note that Type I Micro-Agents exchange information with other Type I Micro-Agents and with Type II Macro-Agents, while Type II Macro-Agents exchange information with agents of all three classes. Type III Monitoring Agents exchange information with other Type III Monitoring Agents and with Type II Macro-Agents.

It can be concluded that the classification of agents, introduced in this chapter, covers various elements of engineering systems and usage profiles to support probabilistic-mechanistic reliability modeling of the dynamic behavior of system elements in a continuously changing environment.

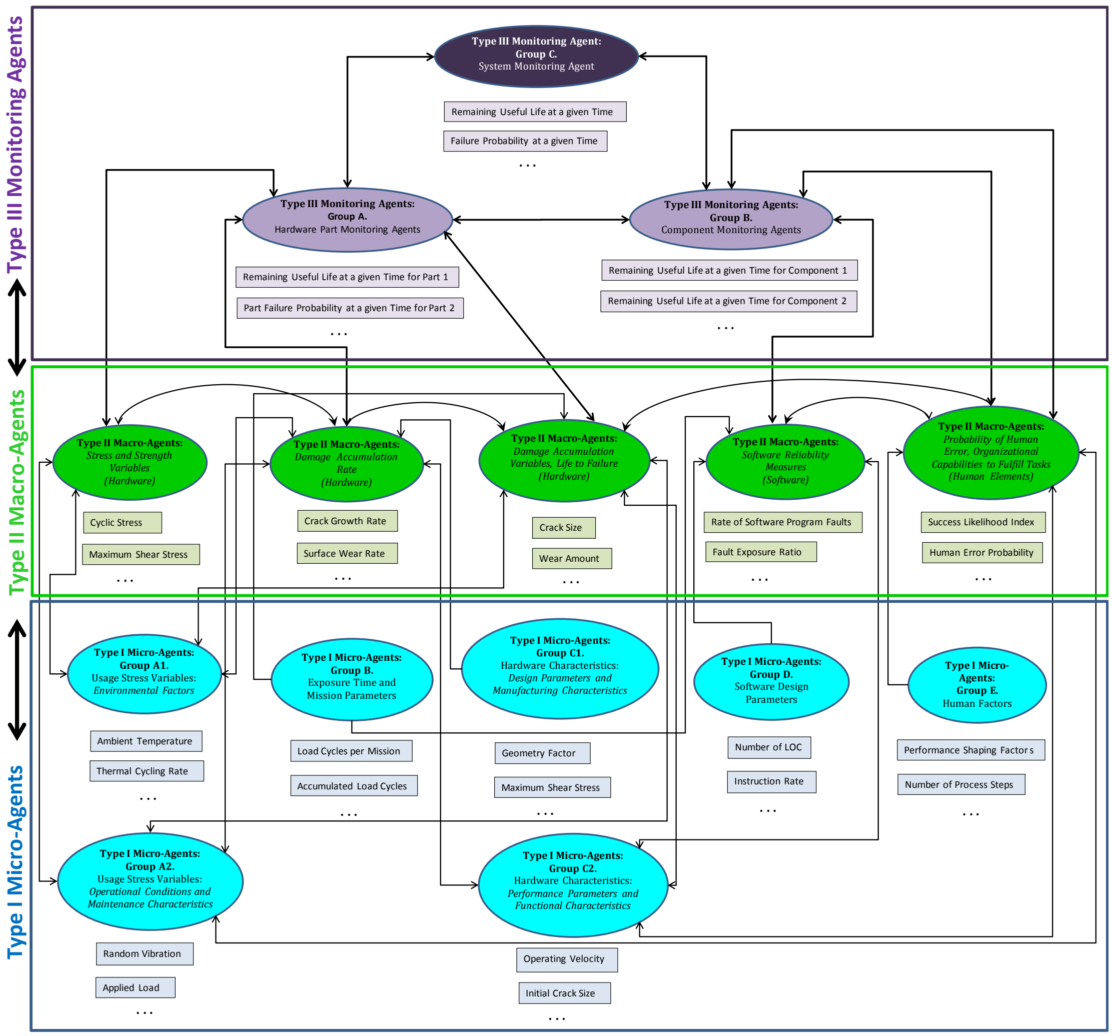


Figure 3-1: Agents Classes and Information Exchange

Figure 3-2 shows an example of system architecture in terms of the elements of degradation and failure processes and their interactions. Each element of the system hierarchy on Figure 3-2 will be further assigned with an intelligent agent, either a simple Type I Micro-Agent (such as for coupling factors), or a complex Type II Macro-Agent, comprised of several Type I Micro-Agents and Type II Macro-Agents of the lower levels of the system hierarchy (such as for stress-strength and degradation-endurance variables, life to failure or time to degradation at failure mechanism level), or a Type III Monitoring Agent at the part level or at the system level of the hierarchy.

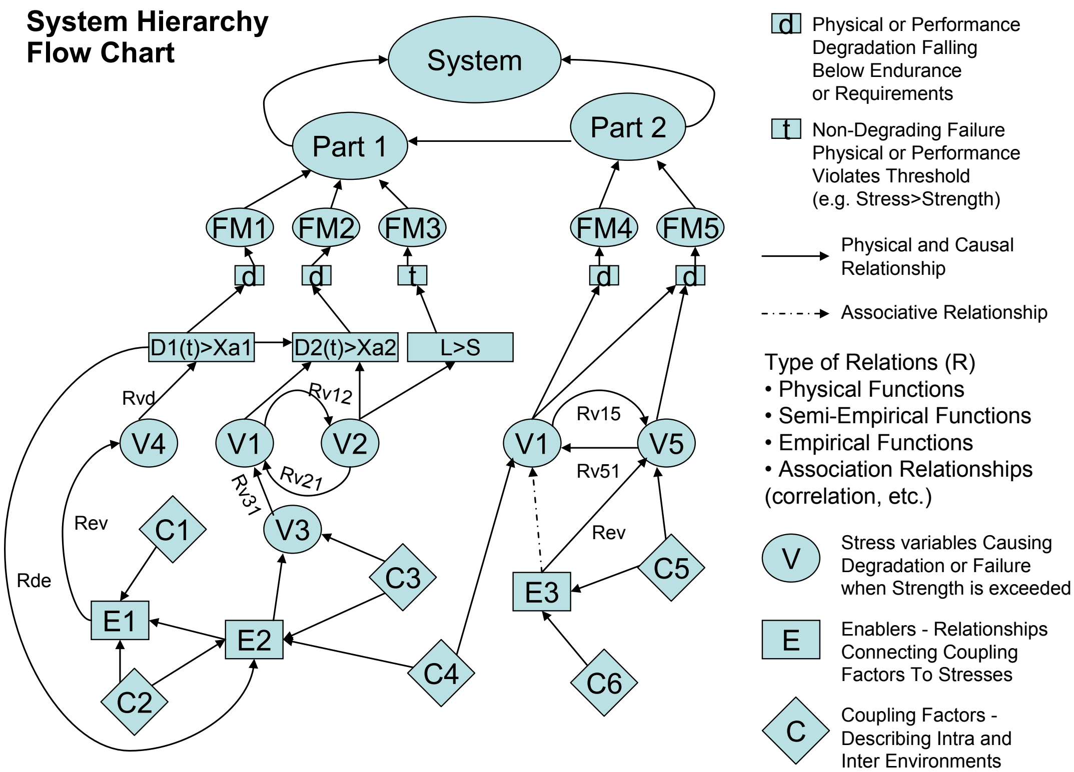


Figure 3-2: Flowchart of System Hierarchy

As shown on Figure 3-2, an output of one agent in the system hierarchy becomes an input to others. A change in the output variable of one agent propagates through the system hierarchy to update the output variables of other agents. Furthermore, two Type III Part Monitoring Agents collect information from five failure mechanisms (FM), combined further by a Type III System Monitoring Agent into the reliability model of the system. Properties of agents reflecting the roles of various elements within system hierarchy are defined in the next chapter.

Chapter 4: Agent Properties

Properties of intelligent agents in the context of agent autonomy for reliability modeling were identified by M. Azarkhail [1]. While the same definitions of agent properties are used in current research, some properties, specifically agent autonomy and learning, were further developed in this research and described in this chapter.

4.1. Internal Knowledge of Agent

This property was introduced and briefly explained in [1] as part of the reasoning/learning characteristic. In the current research, internal knowledge of an agent, also called "agent's beliefs", is defined as the information about an agent output variable, Y, according to the model function, $f(\cdot)$ (per Equation 3-1, given input agents X_i), or according to the probability density function (PDF) of an agent output variable with no inputs from other agents. Agent's beliefs are updated when new data become available. This process is called learning and is explained further in this section (see definition of learning property of agents in Section 4.3).

A history of beliefs comprises an agent's memory about the past events that occurred to the agent during system evolution. Agent's memory is preserved upon every update as the agent evolves over time, following the degradation process of the system. It is due to the nature of the agent learning process that agents can preserve what they have learned from previous experiences and upgrade this knowledge during further updates when new data arrive. An agent's beliefs also include the agent's status as active or inactive, according to autonomous properties of agents (described below). Agent's beliefs are shared with other agents within agent autonomy (as explained in communication/cooperation property of agents) and therefore, are categorized as the public knowledge of an agent.

In addition to an agent's beliefs, internal knowledge of an agent may include special rules of agent behavior during system evolution as a set of goals that the agent is intending to pursue in addition of the main goal of self-evolution (described in Section 4.4). Special rules of agent behavior are formulated by the modeler and built into the computer program of a given agent. They are only accessible by the agent itself and are not shared with other agents, thus categorized as private knowledge of the agent. Special rules of behavior may include, but are not limited to the following examples:

- Special conditions causing a change in the failure logic equation for a complex item represented by the agent, such as a hardware component or a system modeled by a Type III Monitoring Agent (for example, failure of oxygen supply is removed from the aircraft failure logic for the flight phases at altitudes below 12,500 feet).
- Change of the critical degradation threshold for a hardware part or component represented by the Type III Monitoring Agent due to changes in environmental or operational conditions (such as a reduction in the critical crack size when an aircraft is switching to a different mission type with more severe operational conditions).

- When more than one failure mechanism is working within a component at different times of system evolution, the Type III Component Monitoring Agent is designed and programmed to consider the appropriate one based on the circumstances (for example, an agent representing erosive wear mechanism due to sand blasting becomes an input agent to Type III Monitoring Agent only when the aircraft mission takes place in a sandy environment).

4.2. Reactivity

The reactivity property of agents was identified in [1] as agents' ability to perceive their environment by responding to changes that occur. This includes both the sensing and the reaction stages of the action. It is due to this property that the agents remain alert about the changes to the system without the need to modify the agents.

In the context of PoF-based agent autonomy within this research, reactivity is defined as an agent's ability to gain information about the status of other agents, particularly the input agents associated with the input variables X_i (per Equation 3-1). The sensing capability is the agent's ability to track the current beliefs of the input agents, particularly the status of their output variables in a functional form defined by Equation 3-1. It is triggered when an update of the input agents' beliefs is requested upon the availability of new data. The agent reacts to the changes of the input agents' beliefs by updating the probability distribution its output variable, Y, by means of simulation using the updated input variables, X_i ($i = 1, 2, \ldots$).

According to the above definition, the reactivity property is not applicable to Type I Micro-Agents and Type II Macro-Agents which have no inputs from other agents.

4.3. Learning/Reasoning

The learning property was defined in general terms by M. Azarkhail [1] as an agent's ability to learn from previous experiences in order to be able to continuously adapt its behavior to the environment. This current research develops a detailed definition of the learning property and introduces methods of learning for three classes of agents:

1. The process of updating an agent's beliefs using new data is called learning. The learning property supports the development of the internal knowledge of an agent and involves updating the agent's beliefs about the agent output variable, Y, according the model function, f(·) (per Equation 3-1), given the input variables, X_i ($i = 1, 2, \ldots$), specifically:
    a. Update the functional form of the PoF or empirical model, where a function f(·) is probabilistic or deterministic, as applicable.
    b. Update the parameters of the PoF or empirical model given by a function f(·), if f(·) is probabilistic function.

c. Obtain the updated distribution of the agent output variable, Y, by the simulation over the updated model function, f(·), given the current (updated) beliefs about input variables, X_i (i = 1, 2, …).

For Type I Micro-Agents and Type II Macro-Agents that do not have inputs from other agents, the learning property involves updating the functional form and/or parameters of the PDF of the agent output variable.

2. The essence of the agent learning task is to infer posterior knowledge about the agent output function, f(·), or about PDF of the agent output variable (for agents with no inputs) from prior knowledge and the observed data (evidence), and to do it recursively as new data become available. This posterior knowledge (updated beliefs) is obtained by combining the prior knowledge (past beliefs) and new observations (data) by means of the learning methods. The choices of learning methods for Type I Micro-Agents and Type II Macro-Agents depend on the nature of the physical characteristics represented by the agent and the available data. A detailed definition of the learning methods for Type I Micro-Agents and Type II Macro-Agents, their applicability and conditions of use are outlined in Chapter 5. The learning property of Type III Monitoring Agents is discussed in Chapter 6.

4.4. Proactivity/Goal Orientation

According to [1], proactivity is defined (for an agent that has a complex goal) as a collection of several goals that could be switched depending on the circumstances. An example of this, given in [1], is when more than one failure mechanism is involved and the agent is capable of the activation of the appropriate mechanism based on the circumstances. Proactivity of an agent is seen as a goal oriented judgment based on the agent's knowledge of goal preferences.

In this research, all agents have a goal of self-evolving. Each agent follows the dynamics of the environment and uses new data to update the agent's beliefs about the status of the agent output, *Y,* according to the updated model function, f(·) (per Equation 3-1), or according to the updated PDF of the agent output variable (for agents with no inputs). This goal is supported by the reactivity and learning properties of an agent.

Any additional goals may be set for an agent by the modeler by setting the special rules of agent behavior (as part of an agent's internal knowledge property described in Section 4.1). In this case, the proactivity of an agent will also include an agent's capability to choose the correct behavior in the given circumstances.

4.5. Communication/Cooperation

Social activity or the communication property of agents was developed in [1], and defined as an agent's ability to interact with other agents when appropriate. In addition, the cooperation property was defined for the agents that share their goals and knowledge while providing a solution for a common task. In order to cooperate successfully, agents need to communicate their goals, tools and status, and do so using one of the two main approaches: the blackboard approach (indirect communication by accessing a common location for information posted by other agents) and the message passing approach (agents exchange messages directly with each other using defined protocols).

The blackboard approach was used to establish the communication property of agents in this research. In the context of PoF-based agent autonomy, agents communicate by sharing their beliefs with other agents. Information about the agent output variable, Y, is expressed by either a function f(·) (per Equation 3-1, given input agents X_i), or by the PDF of the agent output variable (for agents with no inputs), and is accessible for all other agents within the agent autonomy. This sharing allows an agent to cooperate with other agents during the system evolution by providing a required input to some agents and using others as inputs.

4.6. Autonomy

Agent autonomy was generally defined in [1] as one of the key characteristics of agents; the agents are not only capable of evolving over time with no supervision, but also have some degree of control over their own actions (e.g. self-activation and self-deactivation). This current research extends the definition of the autonomy property and proposes methods of autonomy execution.

In the context of PoF-based agent autonomy developed in this research, an agent may change its status with respect to the form of the input it provides to the other agents. According to the definition of agents in Section 3.1, the output variable of any agent is always probabilistic (i.e. represented by probability distribution of a certain kind). An agent is said to have an active status if the probabilistic form of the agent's output variable is used by other agents as an input to their model functions, $f(\cdot)$ (per Equation 3-1). As an alternative, a constant value, representing the agent's output variable, could be used as an input to other agents. This constant value could be defined as the mean, median, certain percentile, or other numerical value associated with the probability distribution of the agent's output variable. In this case, the agent is said to have inactive status. Change of status from active to inactive is called agent deactivation, and the reverse is called agent activation. Agents activate and deactivate themselves according to defined criteria. For example, if we assume that the agents "A" and "B" are the two inputs for the agent "C", then activation/deactivation criteria for the agents "A" and "B" will be based on the contribution of the uncertainty (variability) of their output variables into the uncertainty (variability) of the output variable of the agent "C". This contribution can be quantified using methods of uncertainty importance (also known as sensitivity analysis). Recommended methods of uncertainty importance for the modeling of agent autonomy along with the considerations of activation/deactivation criteria are discussed in Chapter 7.

Agent autonomy reduces the computation time, brings only the most relevant elements into the system reliability simulation, and allows achieving a better quality of prior information that is needed for the future use of agents in the agent autonomy of similar systems. It is the autonomy property that makes the agent-oriented approach fundamentally different from all existing methods of reliability modeling.

4.7. Mobility

The mobility of an agent is described in [1] as the ability to navigate within a communication network, specifically when a program needs to perform a task on a distributed network.

The current research defines agent mobility in the context of PoF-based agent autonomy for system reliability modeling as the ability to reuse agents in other system applications. As agents learn and execute their autonomy during system evolution, they become richer in their knowledge about the elements they represent. Specifically, each agent continuously improves its “expertise” about the model of the agent output variable and inputs from other agents. In addition, each agent gains the knowledge about its “importance” to other agents (i.e. about the contribution of the uncertainty in the agent output variable into the uncertainty in the output variable of other agents which are using a given agent as an input). As such, the agents become “experts” about the associated elements of the system hierarchy. These “experts” could be reused as unrelated but relevant experience for other applications to reduce computational effort, minimize data requirements, and provide “mature” prior information for further learning.

In order to reuse an agent in similar applications, the agent should be relevant and have an active status. The relevance of an agent stems from a similar design and functionality of the associated element of the system in the new versus the baseline application. If an agent is relevant but inactive, it is introduced into the new application as a constant (not an agent).

4.8. Summary of Agent Properties

Figure 4-1 provides a summary of agent properties. It can be seen that agent properties are linked together to support the agent's Goal of self-evolution. Reactivity, Learning and Autonomy ensure that internal knowledge of an agent is updated when new data become available. Agents communicate by sharing their internal knowledge with other agents. Agents are mobile (i.e. can be used within other agent systems in similar applications), which is especially beneficial when agents become "experts" after several rounds of updates of their beliefs and they deliver "mature" information about a system element. The data types for updating an agent's beliefs could include fully or partially relevant data for the given system (such as test results or operational records) or generic data (such as published industry data or reliability prediction standards).

Agent properties for each class of agents are shown in details on Figure 4-2, Figure 4-3, Figure 4-4 and Figure 4-5. The main distinction between the three classes of agents is in the nature of their communication, reactivity, learning, autonomy and mobility properties, specifically:

1. Agents share their beliefs (communicate) with other agents laterally within their class and vertically with agents from one class above and one class below (as applicable). For example, Type II Macro-Agents provide inputs to other Type II Macro-Agents, to Type I Micro-Agents, and to Type III Part and Component Monitoring Agents. Type I Micro-Agents share their beliefs with other Type I Micro-Agents and with Type II Macro-Agents. Type III System Agent exchanges information only with Type III Part and Component

Monitoring Agents, while the latter also share their beliefs with each other and Type II Macro-Agents.

2. The reactivity property applies to all agents except Type I Micro-Agents and Type II Macro-Agents that don't receive any inputs from other agents. All other agents receive information from their input agents, specifically the updated beliefs of the input agents.
3. The learning process of Type I Micro-Agents and Type II Macro-Agents that don't have any inputs from other agents (i.e. self-sufficient agents) involves updating the type and parameters of the PDF of the agent output variable using data. The dependent Type I Micro-Agents, Type II Macro-Agents and Type III Monitoring Agents (or multi-agents) with probabilistic model of agent output variable learn in three-step process: 1) updating the functional form of the agent output model using data (if applicable), 2) updating the agent output model parameters using data, and 3) updating the probability distribution of the agent output variable by simulation over the updated agent output model given the latest beliefs of the input agents. The learning process of Type I Micro-Agents, Type II Macro-Agents and Type III Monitoring Agents with deterministic model of agent output variable involves a two-step process: 1) updating the functional form of the agent output model using data (if applicable), and 2) updating the probability distribution of the agent output variable by simulation over the (updated) agent output model given the latest beliefs of the input agents. For any class of agents, the parameters of the probability density function of the agent output variable obtained by

simulation could be defined by selecting the best fit probability distribution to the simulated sample.

4. Autonomy and mobility properties apply to all agents with the exception of the Type III System Monitoring Agent which remains in an active state at all times during system evolution and typically is not mobile (unless reused as a Type III Sub-System Monitoring Agent representing a sub-system within another system).
5. Agents update their beliefs and the status (by learning and autonomy) upon availability of new data. It is recommended that the update frequency is defined by the modeler considering the value of information. The value of information principle implies that agent beliefs and status update occurs if the new information noticeably changes the uncertainty of the agent's output variable (for example, coefficient of variation (CV) changes by 10%). Alternatively, the agent beliefs and status update frequency could be based on practical considerations of the amount of new data, frequency of data arrival from different sources and for different agents within the agent hierarchy, other criteria.

Chapter 5, Chapter 6 and Chapter 7 provide detailed definitions of learning and autonomy properties proposed for three agent classes introduced in this research.

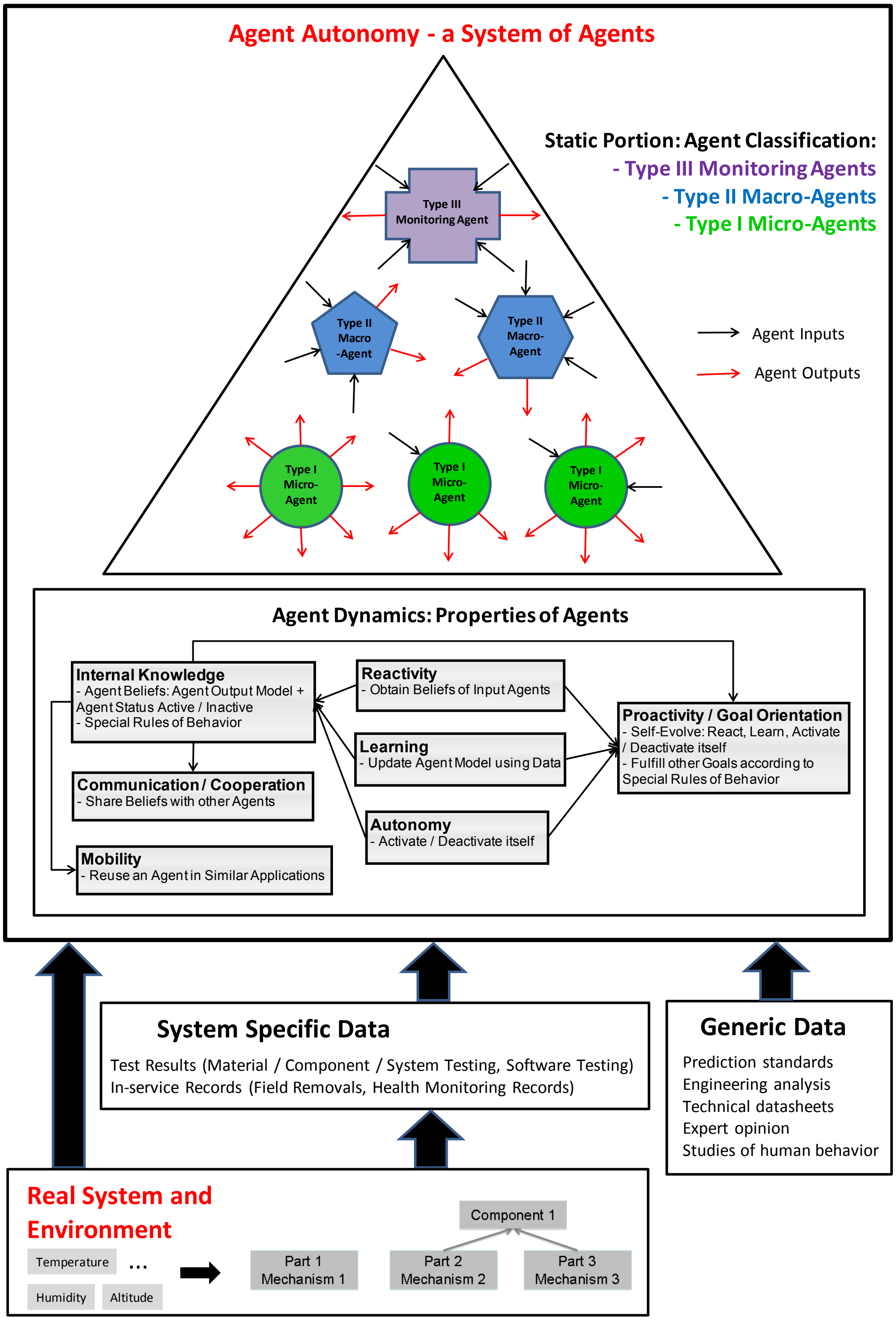


Figure 4-1: Summary of Agents Properties

Type I Micro-Agents (with One or More Inputs)

Associated Elements of System Hierarchy
- Inter and Intra Coupling Factors

Internal Knowledge
- Agent Output Model (PoF or Empirical Function, probabilistic or deterministic)
- Probability Distribution of Agent Output Variable (Simulated over Output Model given Input Agents)
- Agent Status Active / Inactive
- Special Rules of Behavior (if applicable)

Reactivity
- Obtain Beliefs of Input Agents

Learning
- Update Functional Form of Agent Output Model using Data
- Update Parameters of Agent Output Model using Data
- Update Probability Distribution of Agent Output Variable by Simulation via Agent Output Model (given Input Agents)

Data for Learning
- Test Data, Data Collected during Field Operation and Maintenance, Generic Data
- Fully or Partially Relevant

Autonomy
- Activate / Deactivate itself

Proactivity / Goal Orientation
- Self-Evolve: React, Learn, Activate / Deactivate itself
- Fulfill other Goals according to Special Rules of Behavior

Communication / Cooperation
- Share Beliefs with other Type I Micro-Agents (with One or More Inputs) and Type II Macro-Agents

Mobility
- Reuse an Agent in Similar Applications

Figure 4-2: Properties of Type I Micro-Agents (Multi-Agents)

Type II Macro-Agents (with One or More Inputs)

Represent Elements of System Hierarchy
- Stress and Strength Variables, Degradation-Endurance Variables, Time to Degradation or to Damage Accumulation, Life to Failure

Internal Knowledge
- Agent Output Model (PoF or Empirical Function, probabilistic or deterministic)
- Probability Distribution of Agent Output Variable (Simulated over Output Model given Input Agents)
- Agent Status Active / Inactive
- Special Rules of Behavior (if applicable)

Reactivity
- Obtain Beliefs of Input Agents

Learning
- Update Functional Form of Agent Output Model using Data
- Update Parameters of Agent Output Model using Data
- Update Probability Distribution of Agent Output Variable by Simulation via Agent Output Model (given Input Agents)

Data for Learning
- Test Data, Data Collected during Field Operation and Maintenance, Generic Data
- Fully or Partially Relevant

Autonomy
- Activate / Deactivate itself

Proactivity / Goal Orientation
- Self-Evolve: React, Learn, Activate / Deactivate itself
- Fulfill other Goals according to Special Rules of Behavior

Communication / Cooperation
- Share Beliefs with Type I Micro-Agents, other Type II Macro-Agents and Type III Part and Component Monitoring Agents

Mobility
- Reuse an Agent in Similar Applications

Figure 4-3: Properties of Type II Macro-Agents (Multi-Agents)

Type I Micro-Agents (with No Inputs)
Type II Macro-Agents (with No Inputs)

Associated Elements of System Hierarchy
- Inter and Intra Coupling Factors (Type I Micro-Agents)
- Stress and Strength Variables, Degradation-Endurance Variables, Time to Degradation, Life to Failure (Type II Macro-Agents)

Internal Knowledge
- PDF (Probability Density Function) of Agent Output Variable
- Agent Status Active / Inactive
- Special Rules of Behavior (if applicable)

Reactivity
- Not Applicable

Learning
- Update Functional Form of the PDF of Agent Output Variable using Data
- Update Parameters of the PDF of Agent Output Variable using Data

  Data for Learning
  - Test Data, Data Collected during Field Operation and Maintenance, Generic Data
  - Fully or Partially Relevant

Autonomy
- Activate / Deactivate itself

Proactivity / Goal Orientation
- Self-Evolve: Learn, Activate / Deactivate itself
- Fulfill other Goals according to Special Rules of Behavior

Communication / Cooperation
- Share Beliefs with Type I Micro-Agents (with One or More Inputs) and Type II Macro-Agents

Mobility
- Reuse an Agent in Similar Applications

Figure 4-4: Properties of Type I Micro-Agents and Type II Macro-Agents (Self-Sufficient Agents)

Type III Part Monitoring Agents
Type III Component Monitoring Agents

Represent Elements of System Hierarchy
- Parts, Components

Internal Knowledge
- Agent Output Model (Empirical Function, deterministic or probabilistic)
- Probability Distribution of Agent Output Variable (Simulated over Output Model given Input Agents)
- Agent Status Active / Inactive
- Special Rules of Behavior (if applicable)

Reactivity
- Obtain Beliefs of Input Agents

Learning
- Update Functional Form of Agent Output Model using Data
- Update Parameters of Agent Output Model using Data
- Update Probability Distribution of Agent Output Variable by Simulation via Agent Output Model (given Input Agents)

Data for Learning
- Test Data, Data Collected during Field Operation and Maintenance, Generic Data
- Fully or Partially Relevant

Autonomy
- Activate / Deactivate itself

Proactivity / Goal Orientation
- Self-Evolve: React, Learn, Activate / Deactivate itself
- Fulfill other Goals according to Special Rules of Behavior

Communication / Cooperation
- Share Beliefs with Type II Macro-Agents, other Type III Part and Component Monitoring Agents, and Type III System Monitoring Agent

Mobility
- Reuse an Agent in Similar Applications

Type III System Monitoring Agent

Represents Element of System Hierarchy
- System

Internal Knowledge
- Agent Output Model (Empirical Function, deterministic or probabilistic)
- Probability Distribution of Agent Output Variable (Simulated over Output Model given Input Agents)
- Special Rules of Behavior (if applicable)

Reactivity
- Obtain Beliefs of Input Agents

Learning
- Update Functional Form of Agent Output Model using Data
- Update Parameters of Agent Output Model using Data
- Update Probability Distribution of Agent Output Variable by Simulation via Agent Output Model (given Input Agents)

Data for Learning
- Test Data, Data Collected during Field Operation and Maintenance, Generic Data
- Fully or Partially Relevant

Autonomy
- Not Applicable (always Active)

Proactivity / Goal Orientation
- Self-Evolve: React, Learn
- Fulfill other Goals according to Special Rules of Behavior

Communication / Cooperation
- Share Beliefs with Type III Part and Component Monitoring Agents

Mobility
- Reuse an Agent in Similar Applications (if applicable)

Figure 4-5: Properties of Type III Monitoring Agents

Chapter 5: Learning Property of Type I and Type II Agents

5.1. Introduction

The essence of agent learning was described in Chapter 4 as the recursive updating of agent's beliefs every time new data became available. During recursive (sequential) updating, new data are added to the current beliefs of an agent.
To further define agent learning, consider an agent with an output variable given by Equation 3-1. The first step of the update of agent's beliefs involves updating the parameters of the agent output model given by a function f(·) (per Equation 3-1), if the function f(·) is probabilistic (has uncertain parameters).

In the case if a function f(·) is deterministic, the agent's beliefs update involves the change of parameters and/or a functional form of the agent output model if the model error does not satisfy acceptability criteria defined by the modeler. In such case the agent output variable is given by deterministic function, f(·), as follows:

$$Y = f(X_i) + \varepsilon(X_i), \quad i = 1, \dots, n$$

Equation 5-1

In the above equation *Y* denotes a single output variable of an agent as a function of the input variables, $X_1, X_2, \dots, X_n$, and $\varepsilon(X_i)$ is an additive error term. If necessary, multiplicative error model could be used as an alternative to the additive error model given by Equation 5-1. Multiplicative error model (also called ratio model) implies that the magnitude of error is proportional to the value of the output variable. In the additive error model (also called absolute difference model) the magnitude of error does not depend on the value of the output variable. Multiplicative error model could be transformed into an additive error model by a non-linear transformation (such as logarithm transform). The suitability of either type of error term is typically evaluated based on several criteria such as the model fit to the data, predictive capability of the model, the degree of separation of the systematic and random errors (the uncertainties resulted from the difference between the model and the average system responses vs. the uncertainties causing the data variations between experimental runs), applicability to the wide range of variation of the output [63]. Additive errors are normally distributed, while multiplicative errors are highly skewed and follow gamma or lognormal distribution. More complex error models could be used, but there is the risk of over-fitting, and such models quickly lose their predictive capability. A complex error term also implies poor choice of measurement system or inadequate model of the output variable, $f(\cdot)$.

In the second step of an agent's beliefs update, once the functional form and the parameters of the agent output model are updated, the probability distribution of the agent output variable is updated by simulation over the updated model function (probabilistic or deterministic) using the latest beliefs of the input agents.

For Type I Micro-Agents and Type II Macro-Agents that do not have inputs from other agents, the agent's beliefs update involves only one step, changing the type and/or parameters of the PDF of the agent output variable by finding parametric distribution function having the best fit to the new data.

This chapter introduces the learning methods for Type I Micro-Agents and Type II Macro-Agents which can be used to define the model of the agent output variable, if the model is not available, and update the parameters of the identified or already known model. Changing (updating) the functional form of the model during agent learning does not require any special methods in addition to the common criteria of model error evaluation (e.g. normality of additive error, a comparison of the error value to the value of the output variable) and the modeler's engineering judgment. Special reasons for model type change could be programmed into the agent's internal knowledge (particularly the special rules of behavior). Updating the probability distribution of an agent output variable by means of simulation will not be discussed further in this chapter because simulation algorithms are well known and widely applied [64], [65].

With respect to the format of the data required for agent learning, discrete measurements are obtained when the system elements are sampled at certain time points during system evolution. Continuous measurements must be discretized. Where indirect measurements of the output quantity, *Y,* are used, a reasoning algorithm (causal relationship) is required to correlate a change in the measured variable (also called precursor variable) with a change in agent output variable, *Y*. It is desirable that such correlation is based on the PoF model rather than a statistical function because use of purely statistical transfer function implies that the first principles are omitted and extrapolation to new conditions is at risk to be invalid if non-monotonic relationships or complex interacting failure mechanisms are present.

Several existing methods of data analysis were chosen as learning methods for Type I Micro-Agents and Type II Macro-Agents, as summarized in Table 5-1 and discussed further in this chapter.

Table 5-1: Applicability of Learning Methods for Type I and Type II Agents

Learning Method	**Type I Micro-Agents (with No Inputs)** **Type II Macro-Agents (with No Inputs)**	**Type I Micro-Agents (with One or More Inputs)** **Type II Macro-Agents (with One or More Inputs)**
Parametric Distribution Analysis (classical)	✓	
Model-based Parametric Distribution Analysis (cumulative damage)		✓
Time Series and Trend Analysis for Degradation and other Trend Modeling	✓	✓
Machine Learning and Pattern Recognition Methods (excluding Bayesian Fusion methods)	✓	✓
Bayesian Inference	✓	✓
Bayesian Fusion		✓

Applicability of each learning method to a specific Type I Micro-Agent or Type II Macro-Agent is identified according to the criteria that comprise the following considerations, as a minimum:

1. *Dependency of the agent output variable on other variables*
    a. Independent variable Y, or
    b. Variable $Y(t)$ is a function of one or more input variables t_i ($i = 1, \ldots$).
2. *Availability of PoF or empirical model of the agent output variable*
    a. A model is not available, or
    b. A model is available, the model is probabilistic or deterministic (i.e. model parameters are uncertain or fully defined), linear or nonlinear.

3. *Time dependency of the agent output variable*
    a. Time dependent (time trend in the output variable *Y(t)* exists, one of the input variables, t_i, is time variable), or
    b. Not time dependent (no time trends in the output variable *Y(t)* exist).
4. *Type of agent output variable according to the continuity of data/information represented by the variable*
    a. Continuous variable (represented by continuous distribution model), or
    b. Discrete variable (represented by discrete distribution model).
5. *Rate of change in the agent output variable over time (for time dependent variables)*
    a. Slow change over time (gradual increase in degradation measure, accumulation of damage, such as crack growth to critical size), or
    b. Rapid change over time (acute change in degradation measure, rapid increase in accumulated damage, e.g., as brittle fracture, rapid crack propagation to rupture, crack arrest).
6. *Monotonicity of change in the agent output variable over time (for time dependent variables)*
    a. Monotonic change over time (entirely non-increasing or non-decreasing), or
    b. Non-monotonic pattern (e.g., cyclic changes, seasonal patterns).
7. *Types of data available for agent learning*
    a. Direct measurements of the agent output variable, *Y*, or

b. Direct measurements of the agent output variable, *Y(t)*, and the input variables t_i ($i = 1, \ldots$).

8. *Data sources*

   a. Inspection and maintenance records.

   b. Data from sensors and monitors.

   c. Generic data.

   d. Expert opinion.

The next sections of this chapter include the guidelines for choosing an agent learning method based on the above criteria and provide application examples.

5.2. Parametric Distribution Analysis (Classical)

Classical parametric distribution analysis [8], [66] - [70] can be used as an agent learning method for the Type I Micro-Agent or the Type II Macro-Agent assigned to a time independent, random variable with no inputs from other variables (i.e. no PoF or empirical model of the agent output variable exists). Classical methods of parameter estimation, least squares regression (LSR) or maximum likelihood estimation (MLE) method, are used to estimate the parameters of the best fitted distribution for the data. Each time new data emerge, it is added to the previously available data and the analysis is repeated for the updated data set. Any discrete or continuous parametric distribution can be used to represent the agent output variable. An extensive list of parametric distributions is given in publication [71]. The data that are required for agent learning include the direct measurements of the agent output variable, such as environmental or operational characteristic, time to failure/success, etc. Typical data sources are the testing, inspection and maintenance records from field operation, and generic data sources.

For example, ball bearings are tested in a lab and the ball bearing temperature is considered a critical variable within physical failure model of the bearing. Even under a constant ambient temperature, the internal temperature of a ball bearing fluctuates depending on the operating conditions. If the relationship between the ball bearing temperature and the operating stresses cannot be reasonably established, then ball bearing temperature is defined as a Type I Micro-Agent with no input agents. The probability distribution model for the output variable of this agent is established using initial lab test measurements as normal distribution, $N(\mu_0,\sigma_0)$. As the test continues, new measurements of bearing temperature arrive and are added to the initial data set, then the parametric distribution analysis is performed to obtain an updated normal distribution, $N(\mu_1,\sigma_1)$, having the best fit to the updated data set.

The multivariate parametric distribution function can be viewed as a generalization of agent representation and learning through parametric distribution analysis for Type II Macro-Agents. For example, the bivariate normal distribution of Type I Micro-Agents *X* and *Y* forms a representation of the output variable of the Type II Macro-Agent, *Z*:

$$f(Z) = f(X,Y) =$$

$$= \frac{1}{2\pi\sqrt{1-\rho^2}\sigma_X\sigma_Y} exp\left[-\frac{\frac{(X-\mu_X)^2}{\sigma_X{}^2}+\frac{(Y-\mu_Y)^2}{\sigma_Y{}^2}-2\rho\frac{(X-\mu_X)(Y-\mu_Y)}{\sigma_X\sigma_Y}}{2(1-\rho^2)}\right]$$

Equation 5-2

As new data become available, the distribution parameters are revaluated using all available data for the variable *Z* versus variables *X* and *Y*.

Table 5-2 provides two examples of the learning process of Type I Micro-Agents: current density and temperature of the conductor. These Type I Micro-Agents could be used further as input agents to a certain Type II Macro-Agent, for example life to electromigration failure of electronic part. The type of model of the agent output variable (i.e. PDF of the current density, *J*, and PDF of the conductor temperature, *T*) did not change during the updating of agent's beliefs, but the model parameters were updated using new data.

Table 5-2: Examples of Agent Learning by Parametric Distribution Analysis

Elements of Agent Learning Process	**Type I Micro-Agent**	
	Current Density	**Temperature of the Conductor**
Input Variables from Input Agents	Not applicable	Not applicable
Output Variable	Current Density, *J*	Temperature of the Conductor, *T*
Model of Agent Output Variable - Past Beliefs	Lognormal probability distribution, $LN(\mu_0, EF_0)$, *obtained by selecting best fit distribution for the measurements collected during the past test (6 data points), where* μ_0 *is median and* EF_0 *is Error Factor of the distribution*	Normal probability distribution, $N(\mu_0, \sigma_0)$, *obtained by selecting best fit distribution for the temperature measurements collected during the past test (6 data points), where* μ_0 *is mean and* σ_0 *is standard deviation of the distribution*
Parameters of Agent Output Model - Past Beliefs	$\mu_0 = 500{,}000$ A/cm^2, $EF_0 = 2$, *distribution parameters are obtained by MLE method*	$\mu_0 = 60$°C, $\sigma_0 = 3$°C, *distribution parameters are obtained by MLE method*
New Data	Current density measurements obtained during accelerated test (3 data points)	Temperature values obtained from the thermal model (3 data points)

Elements of Agent Learning Process	Type I Micro-Agent	
	Current Density	**Temperature of the Conductor**
Model of Agent Output Variable - Updated Beliefs	Lognormal probability distribution, $LN(\mu_1, EF_1)$, *obtained by selecting best fit distribution for the measurements collected during the past test and the current accelerated test combined (total of 9 data points), where* μ_1 *is median and* EF_1 *is Error Factor of the updated distribution*	Normal probability distribution, $N(\mu_1, \sigma_1)$, *obtained by selecting best fit distribution for the temperature measurements collected during the past test and obtained from the thermal model combined (total of 9 data points), where* μ_1 *is mean and* σ_1 *is standard deviation of the updated distribution*
Parameters of Agent Output Model - Updated Beliefs	μ_1 = 600,000 A/cm^2, EF_1 = 1.5, *distribution parameters are obtained by MLE method*	μ_1 = 65°C, σ_1 = 2°C, *distribution parameters are obtained by MLE method*

5.3. Model-Based Parametric Distribution Analysis (Cumulative Damage Model)

The output variable of the Type I Micro-Agent or the Type II Macro-Agent with one or more input agents can be represented by a parametric distribution if one or more parameters of that distribution are expressed by the PoF or empirical function of agent input variables (associated with the input agents). The agent output variable is not time dependent (i.e. no time trends in agent output variable exist, and none of the input variable is time variable). In this case the agent model will be given by the PDF as a combination of the probability distribution model (such as Weibull, lognormal, exponential, etc.) and the life-stress PoF relationship (such as Arrhenius model, inverse power law (IPL), exponential model, etc.) with uncertain parameters. The examples are Eyring-Weibull, IPL-exponential, Arrhenius-lognormal probability distribution functions. The MLE method is used to estimate model parameters. Some of the input variables could be time-varying, so that the life-stress relationship and the maximum likelihood estimation of model parameters will take into account the cumulative effects of the applied stresses [72]. That's why the described type of model is also referred to as a cumulative damage or cumulative exposure model. Each time new data set emerges, it is combined with all the past observations into one data set for a re-estimation of the parameters of the model-based probability distribution. The data set that is required for agent learning includes the direct measurements of the agent output variable (such as time to failure/success) and direct measurements of input variables (such as environmental or operational parameters, etc.). Typical data sources include test results, inspection and maintenance records from field operation.

For example, the Type II Macro-Agent representing time to fatigue failure of a component due to cyclic stress, *S*, could be modeled by Weibull-IPL probability distribution function, $Y \sim W(t, S|\beta, K, n)$, as follows:

$$f(t,S) = \beta K S^n (K S^n t)^{\beta-1} exp\left[-(K S^n t)^{\beta}\right]$$

Equation 5-3

The letter *t* in the above equation denotes time to failure, letters *β, K* and *n* denote the model parameters, specifically, *β* is Weibull shape parameter, while Weibull scale parameter, *α,* is expressed as IPL function of the cyclic stress, *S*:

$$\alpha = \frac{1}{K S^n}$$

Equation 5-4

The probability density function given by Equation 5-3 is a combination of the empirical Weibull distribution model of the time to failure and life-stress PoF model defined by Equation 5-4. Distribution parameters *β*, *K* and *n* could be obtained by the MLE method. As new data for time to failure versus cyclic stress become available, the distribution parameters are revaluated using all available data combined.

5.4. Time Series and Trend Analysis

The learning property of the Type II Macro-Agent assigned to a time dependent degradation variable could be modeled using various methods of time series and trend analysis, stationary or non-stationary [73], [74], [75]. Time series is defined as sequence of measurements of a numerical quantity collected at a regular interval. The available data recorded in chronological order are used to develop a time series plot which is further used to detect trends in the data over time and compare the trends across several data groups. The data are plotted on the vertical y-axis versus time on the horizontal x-axis. Trend analysis is used to fit a general trend model to the time series, and to provide forecasts (extrapolations of the trend model fits). The elements of the fitted equation are: t, representing the time variable (calendar hours, operational hours, cycles, expended life or an index of age, etc.), and $Y(t)$, representing the value of the measured variable at time t. All other terms are the coefficients, i.e. numerical constants that are used to express the variable under consideration as a function of time. Some examples of trend models which can be assigned to the output variable of a Type I Micro-Agent or a Type II Macro-Agent are the following:

- Regression model (cubic)

$$Y(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3$$

Equation 5-5

- Exponential model

$$Y(t) = \beta_0 {\beta_1}^t + \beta_2 exp(\beta_3 t + \beta_4)$$

Equation 5-6

- Sinusoidal model

$$Y(t) = \beta_0 + (\beta_1 + \beta_2 t + \beta_3 t^2) \sin(2\pi\beta_4 t + \beta_5)$$

Equation 5-7

The time variable, *t,* in trend equations is an output variable of the Type I Micro-Agent representing time (e.g. cycles, hours, months), being a single input to another Type I Micro-Agent or Type II Macro-Agent with output variable *Y(t).* The coefficients β_i (i = 1, 2, ...) are the model parameters. Classical methods of parameter estimation, LSR or MLE, are used to obtain the model parameters.

A general form of the output variable of a Type I Micro-Agent or a Type II Macro-Agent is defined by the time series or trend model with a single input variable, which can be written as:

$$Y(t) = f(t, \theta) + \varepsilon(t)$$

Equation 5-8

where *Y(t)* is the agent output variable, $f(t,\theta)$ is a mathematical function of the input variable, *t*, with the vector of model parameters $\theta = \{\beta_0, \beta_1, ..., \beta_k\}$, and $\varepsilon(t)$ is an additive error term, which is assumed to be a normally distributed, random variable with zero mean and standard deviation σ, as follows:

$$\varepsilon(t) = Normal\ (0, \sigma)$$

Equation 5-9

The variable t may represent not only a time variable, but also any other variable serving as an input to the stochastic variable $Y(t)$. If necessary, multiplicative error model could be used instead of the additive error model given by Equation 5-8.

The learning properties of Type I Micro-Agents or Type II Macro-Agents that are assigned to a time dependent variable could also be modeled using trend analysis in cases when the agent has more than one input variable. Some examples of trend models for complex variables, $Y(t_i, i=1,\dots,n)$, representing the output variables of Type I Micro-Agents or Type II Macro-Agents as a combination of two or more input variables, $t_1, t_2, \dots, t_n$, (associated with input agents), are the following:

- Linear model

$$Y(t_1, t_2, \dots, t_n) = \beta_0 + \beta_1 t_1 + \beta_2 t_2 + \cdots + \beta_k t_n$$

Equation 5-10

- Nonlinear model

$$Y(t_1, t_2) = \beta_1 + \beta_2 \cdot Ln(t_1 + \beta_3) + \beta_4 exp(t_2 + \beta_5) + \beta_6 t_3^2$$

Equation 5-11

The coefficients β_i ($i = 1, 2, \dots$) are the model parameters. A general form of the output variable of a Type I Micro-Agent or a Type II Macro-Agent with the output variable defined by a trend model with multiple input variables can be written as:

$$Y(T) = f(T, \theta) + \varepsilon(T)$$

Equation 5-12

$Y(T)$ is agent output variable, $f(T,\theta)$ is a mathematical function of n input variables $T = \{t_1, t_2, \dots, t_n\}$ with the vector of model parameters $\theta = \{\beta_0, \beta_1, \dots, \beta_k\}$, and $\varepsilon(T)$ is an additive error term, which is assumed to be normally distributed random variable with zero mean and standard deviation σ, as follows:

$$\varepsilon(T) = Normal\ (0, \sigma)$$

Equation 5-13

In order to choose the appropriate model, a model fit to the data is subject to graphical and quantitative techniques of model validation. For example, residual plots could be used to determine if the ordinary least squares assumptions are being met for the additive error term, such as the constant variance for different data groups, independence of variables and the normality of the distribution of residuals with zero mean. If these assumptions are satisfied and the value of standard deviation, σ, is relatively small, then the chosen model will have unbiased coefficient estimates with minimum variance. Otherwise, the modeler could decide to update the functional form of the model as part of the agent learning process or choose to use multiplicative error term.

Another method of time series analysis, decomposition, is used to separate the time series into the linear trend, seasonal components, and the error term, and to provide forecasts. Decomposition is used when the data exhibit either no trend or a constant trend, have constant seasonal pattern, and the seasonal component is either additive or multiplicative with the trend. The decomposition method determines the seasonal indices used to seasonally adjust the data and fits a trend line to the seasonally adjusted series. The trend and seasonal indices are further used to determine the predicted values and forecasts as a sum (additive case) or a product (multiplicative case) of the trend and seasonal components. Figure 5-1 shows the example of decomposition analysis for the temperature variable.

Fitted Trend Equation

```
Y(t) = 345.4 + 4.25*t
```

Seasonal Indices

Period	Index
1	-82.097
2	-69.597
3	-50.431
4	-40.847
5	0.403
6	94.569
7	170.653
8	64.986
9	34.569
10	15.819
11	-45.931
12	-92.097

Figure 5-1: Results of Data Analysis by Decomposition

The data decomposition procedure produces a graph containing a trend line, observations, predicted values, and forecasts versus time (Figure 5-2). The predicted values and the forecasts are obtained from the fitted trend line multiplied by (multiplicative model) or added to (additive model) the seasonal indices. Each predicted value in this example of the additive model is a sum of the trend value and the corresponding seasonal index. Same applies to the forecasts. For example, for the second period, the trend component, obtained from the trend equation, equals 353.9, the seasonal index is -69.597, and their sum gives the predicted value for the second period as 284.3.

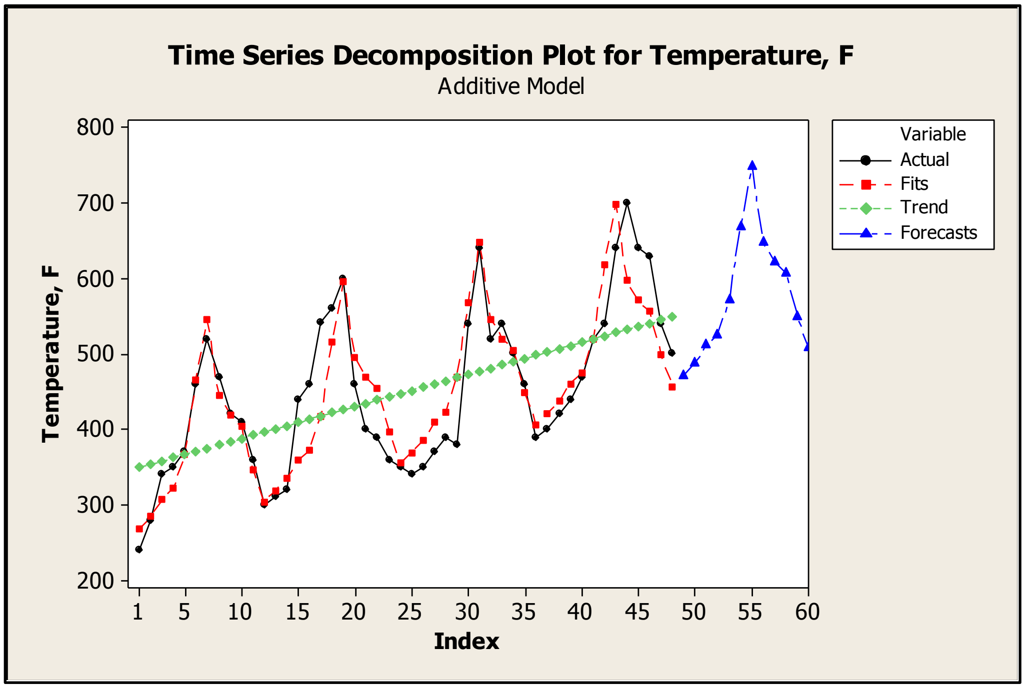


Figure 5-2: Graphical Output for Decomposition Example

The coefficients of the linear function, *Y(t)*, and the seasonal indices (shown on Figure 5-1) comprise parameters of the model of the agent output variable, given in general form by equation:

$$Y(t, Season_i) = Y(t) + Seasonal\ Index_i,\ \ i = 1, 2, \ldots, 12$$

Equation 5-14

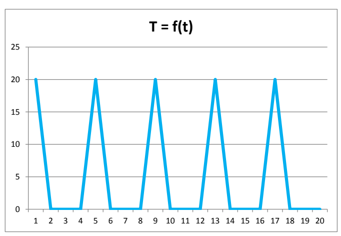
Such model applies to the output variable of the Type I Micro-Agent or the Type II Macro-Agent with the time variable as a single input variable.

Empirical mode decomposition (EMD) method of breaking down a signal can be viewed as a generalization of the decomposition techniques described above [75]. EMD is used, along with other analysis methods such as Fourier transforms and wavelet decomposition, to decompose non-linear and non-stationary time dependent variable into a finite number of components.

The time series and trend analysis models described in this chapter can be used as agent learning methods for Type I Micro-Agents or Type II Macro-Agents, both independent agents (with no inputs from other agents) and agents with one or more input agents (one of which could represent time variable). These agents could represent time dependent degradation measures (such as crack size) or other time-dependent or time-independent variables when the PoF or empirical model of agent output variable is not available. Every time new data become available, it is combined with all past observations into one data set for re-estimation of the time series or trend model parameters. The data set that is required for agent learning includes the direct observations of the agent output variable (such as degradation measure, environmental or operational characteristic, etc.) and direct measurements of input variables (such as time variable, environmental or operational parameters, etc.). Typical data sources are testing, inspection and maintenance records from field operation, and data recorded by sensors and monitors. Examples of Type II Macro-Agent learning by trend analysis are given in Table 5-3.

It must be noted that several supervised machine-learning learning and pattern recognition procedures could also be used for time series and trend analysis. Application of machine-learning learning and pattern recognition methods of data analysis for agent learning is discussed in Section 5.5.

Table 5-3: Examples of Agent Learning by Time Series and Trend Analysis

Elements of Agent Learning Process	Type II Macro-Agent	
	Operating Temperature, *T*	**Current Density, *J***
Input Variables from Input Agents	Flight Time, *t*	Voltage, *V*
Output Variable	Device Operating Temperature, *T*	Diode Current Density, *J*
Model of Agent Output Variable - Past Beliefs	$T(t) = \beta_0 + \beta_1 T(t-1) + \beta_2 T(t-2) + \beta_3 T(t-3)$ The above equation was obtained by selecting best fit model based on the physical nature of temperature change during flight cycle and using temperature measurements collected during three flight cycles. 	The PoF fundamentals dictate the following exponential model: $J(V) = \frac{I_0}{A}\left[exp\left(\frac{V}{V_T}\right) - 1\right]$ where: I_0 is the diode reverse saturation current (diode material constant), A/cm^2 A is the junction area, cm^2 V is applied voltage across the terminals of the diode (forward bias, $V > 0$), Volts V_T is thermal voltage, Volts $V_T = kT/q$, where q is absolute value of electron charge, k is Boltzmann's constant, T is absolute temperature (K), V_T = 25.85 mV at room temperature (T=300K). The model is simplified for room temperature operation, as follows: $J(V) = J_0\left[exp\left(\frac{V}{0.02585}\right) - 1\right]$ where $J_0 = I_0/A$, A/cm^2.
Parameters of Agent Output Model - Past Beliefs	Iterative algorithm was used to obtain model parameters, as follows: $\beta_0 = 20,\ \beta_1 = \beta_2 = \beta_3 = -1$	The model parameter J_0 was obtained by LSR method based on experimental measurements of the current density vs. voltage (6 data points), as follows: $J_0 = 25.0$ μA/cm^2
New Data	Temperature measurements collected during ten flight cycles (10 data points).	Additional experimental measurements of current density vs. voltage (3 data points)
Model of Agent Output Variable - Updated Beliefs	Time model for operating temperature *T* is unchanged: $T(t) = \beta_0 + \beta_1 T(t-1) + \beta_2 T(t-2) + \beta_3 T(t-3)$	Exponential model for the current density *J* remains unchanged: $J(V) = J_0\left[exp\left(\frac{V}{0.02585}\right) - 1\right]$

Elements of Agent Learning Process	**Type II Macro-Agent**	
	Operating Temperature, *T*	**Current Density, *J***
Parameters of Agent Output Model - Updated Beliefs	The iterative algorithm was used to obtain the updated model parameters using the temperature measurements collected during the total of six flight cycles (i.e. total of 6 data points): $\beta_0 = 15,\ \beta_1 = \beta_2 = \beta_3 = -1$	The updated parameter J_0 was obtained by LSR method based on the current density vs. voltage measurements collected during the two rounds of experimentation (total of 9 data points): $J_0 = 24.5\ \mu A/cm^2$

5.5. Machine Learning and Pattern Recognition Methods

Machine learning, a branch of artificial intelligence, deals with data representation and generalization. Representation of data instances and functions (patterns) evaluated on these instances are part of all machine learning methods. Generalization is the property that the measured variable will perform well on unseen data instances, allowing to make predictions of future behavior that is based on the known properties learned from the data. These approaches are based on statistical learning techniques from the theory of pattern recognition, and include supervised and unsupervised learning methods. Statistical models and algorithms of machine learning comprise the core of the data-driven approaches to prognostics and health management (PHM). Figure 5-3 shows the most common machine learning methods employed in PHM, detailed in publications [76] - [88].

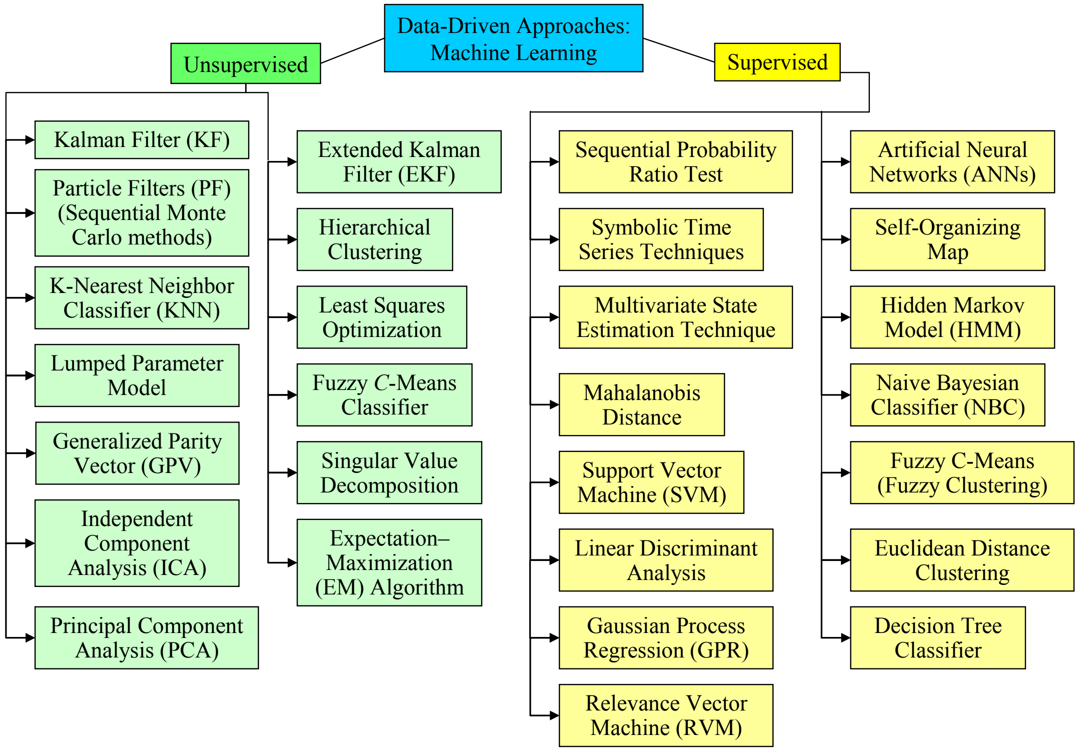


Figure 5-3: Machine Learning Methods

Supervised learning is defined as the machine learning algorithm of inferring a function from labeled data called training data. The training data is a set of training examples, where each example is a pair of input object measures and a desired output value (also called a supervisory signal). A supervised machine learning algorithm analyzes the training data set and produces an inferred function for the output quantity. This function is used to map new data points to obtain a predicted value of the output. To be effective, the training data for machine learning algorithms must span the universe of system failures and operational conditions, which is impractical when dealing with complex engineering systems.

The task of unsupervised learning is finding hidden structures in unlabeled data. Since the data given to the learner are unlabeled, no error could be obtained to evaluate a potential solution. Unsupervised learning is similar to the problem of probability density estimation in statistics (described in Sections 5.2 to 5.4). Unsupervised learning, however, also encompasses many other techniques that explain key features of the data.

While the machine learning approaches are suitable for all levels of system hierarchy from the piece part to the system, the above considerations suggest that unsupervised algorithms are preferred for the agent learning process. Furthermore, considering properties and capabilities of the available unsupervised methods, Kalman filter (KF) and extended Kalman filter (EKF) are generally recommended as learning methods of Type I Micro-Agents and Type II Macro-Agents where PoF or empirical model of the agent output variable exists. Applicability of KF and EKF depends on the model type: KF is suitable under the assumptions of the model linearity and Gaussian distribution of the agent output variable and the noise (error term), while EKF performs local linearization of the non-linear model. Both KF and EKF, along with several other unsupervised algorithms, are categorized as Bayesian fusion methods and discussed in details in Section 5.8. Other machine learning methods (supervised and unsupervised) could still be used for agent learning to the discretion of the modeler. For a detailed description of machine learning methods, the interested reader is referred to the literature. One of the recommended machine learning methods from a supervised category, Gaussian process regression, is described below in Section 5.5.1.

Similar to parametric distribution analysis and trend models, machine learning methods make no reflection on fault sites, operating conditions, and physical mechanisms of failure (i.e. the first principles are omitted). With exception of Bayesian Fusion methods, machine learning algorithms are used for agent learning when the physics-of-failure model of the agent output variable is not available.

Machine learning methods are suitable for Type I Micro-Agents or Type II Macro-Agents, both independent agents (with no inputs from other agents) and agents with one or more input agents (one of which could represent time variable). The required data for agent learning include direct measurements of the agent output quantity, $Y(T)$, $T = \{t_1, t_2, \ldots, t_n\}$, and the input quantities, t_i ($i = 1, \ldots, n$), where $Y(T)$ is could be a degradation measure (e.g. crack size), environmental or operational characteristic. Typical data sources are sensors and monitors (usually in PHM applications). New data, when become available, are combined with all past measurements into the updated data set for a re-estimation of machine learning or pattern recognition model parameters.

Application examples include situations when a Type I Micro-Agent or a Type II Macro-Agent represents a failure precursor used as an indicator of part degradation in lieu of the physical model of degradation or failure [89]. A precursor is a random time dependent variable which could also be a function of other variables. Examples of failure precursors for electronic devices are reverse leakage current, forward voltage drop and power dissipation for diodes, leakage current/resistance for capacitors, and impedance changes for cables and connectors. Examples of failure precursors in mechanical systems are vibration of a gearbox [90], noise and vibration of ball bearings [91]. Failure precursor measurements are collected at certain intervals during the experiment or in field operation. Machine learning methods can be used to recursively update the time trend model of such agent as it evolves over time, due to continuous change in the precursor following the degradation process of the item. A failure condition would be considered to occur when the precursor value has reached a known threshold.

5.5.1. Gaussian Process Regression

Gaussian process regression (GPR) is non-parametric regression technique that provides an alternative solution to the model selection problem commonly seen in parametric models [78], [86], [87], [92]. GPRs are able to model complex non-linear relationships that are often present in failure rate data. This model has received considerable attention recently due to the inherent flexibility provided in its Bayesian framework. The technique also provides a straightforward approach for modeling dependencies within the data.

Within a framework of agent autonomy, the GPR model could be used to represent the output variable of Type I Micro-Agents or Type II Macro-Agents having multiple input variables from other agents. The problem of learning in a Gaussian processes is finding suitable properties for the covariance function, specifically hyperparameters of the covariance function [78].

An example of GPR application to agent learning is the modeling of a certain reliability characteristic, such as failure rate, as a function of time (or accumulated mileage), aircraft or vehicle model (type), and usage conditions. Modeling the failure rate of a fleet of vehicles on a monthly basis using the GPR method is described in publication [92]. An example of Type II Macro-Agent learning by means of the GPR is given in Table 5-4.

Table 5-4: Example of Type II Macro-Agent Learning by GPR

Elements of Agent Learning Process	**Type II Macro-Agent**
	Automotive Tire Tread Wear Rate
Input Variables from Input Agents	Type I Micro-Agents: 1. Accumulated Mileage Agent, *M* 2. Vehicle Model Agent, *V (3 vehicle models, assigned with 2 indicator variables due to qualitative nature)* 3. Predominant Road Conditions Agent, *R (3 road types, assigned with 2 indicator variables due to qualitative nature)*
Output Variable	Tire Tread Wear Rate, *WR*
Model of Agent Output Variable - Past Beliefs	*WR = f(X)* is the observed tread wear rates for each vehicle during the month, defined as follows: 1. Input vector $X = (x_1, x_2, x_3)$, where continuous variable x_1 represents agent *M*, indicator variables x_2, x_3 represent agent *V*, indicator variables x_4, x_5 represent agent *R* 2. Distribution of the log of the fleet rate of tire tread wear: $\log(f(x)) \sim GP\left(m(x), k(x_i, x_j)\right), \; i, j = 1, 2, 3$ $\log(f(X)) \sim N(m(X), K(X, X))$ 3. The data outputs will be centered to have zero mean on the training data set, *m(X) = 0*. 4. Possible periodic trend in tread wear rate is expected, leading to a periodic kernel function component. A smoothed model that could handle the noise present within the data is also required, leading to the addition of squared exponential and noise components. The resulting kernel function consists of the sum of squared exponential, periodic, and general noise components: $k(x_i, x_j) =$ $\theta_6^2 exp\left(-\frac{(x_i - x_j)^T P (x_i - x_j)}{2\theta_7^2}\right) + \theta_8^2 exp\left(-\frac{(x_i - x_j)^T P (x_i - x_j)}{2\theta_9^2} -\right.$

Elements of Agent Learning Process	Type II Macro-Agent
	Automotive Tire Tread Wear Rate
	$\left.\frac{2sin\left(\pi^2(x_i-x_j)^T P(x_i-x_j)\right)}{\theta_{10}^2}\right) + \sigma_n^2\delta_{ij}$ where δ_{ij} is a Kronecker delta which is one iff $i = j$ and zero otherwise, P is diagonal 5-dimensional matrix with hyperparameters θ_1 through θ_5 along the diagonal and zeros elsewhere, each hyperparameter in matrix P corresponds to the respective term in the input vector $X = (x_1, x_2, x_3)$. 5. The log-likelihood of data (X, Y) (reasoning property of the agent) is defined as: $\log p\,(Y\|X,\theta) = -\frac{1}{2}Y^T[K(X,X) + \sigma_n^2 I]^{-1}Y - \frac{1}{2}\log[K(X,X) + \sigma_n^2 I] - \frac{n}{2}\log(2\pi)$ where θ is the vector containing all hyperparameters and noise level parameter σ_n^2, $\theta = (\{P\}, \theta_6, \theta_7, \theta_8, \theta_9, \theta_{10}, \sigma_n^2)^T$.

<table>
<tr><th rowspan="2">Elements of Agent Learning Process</th><th>Type II Macro-Agent</th></tr>
<tr><th>Automotive Tire Tread Wear Rate</th></tr>
<tr><td>Parameters of Agent Output Model
- Past Beliefs</td><td>The vector containing all hyperparameters and noise level parameter σ_n^2: $\theta = (\{P\}, \theta_6, \theta_7, \theta_8, \theta_9, \theta_{10}, \sigma_n^2)^T$
The optimal values of the hyperparameters and noise level parameter were computed from the past training data in GPML Matlab code using conjugate gradient optimization [92]:

<table>
<tr><th>Parameter</th><th>Value</th></tr>
<tr><td>θ_1</td><td>7.734</td></tr>
<tr><td>θ_2</td><td>0.245</td></tr>
<tr><td>θ_3</td><td>0.390</td></tr>
<tr><td>θ_4</td><td>716.394</td></tr>
<tr><td>θ_5</td><td>325.983</td></tr>
<tr><td>θ_6</td><td>30.285</td></tr>
<tr><td>θ_7</td><td>3.764</td></tr>
<tr><td>θ_8</td><td>10.324</td></tr>
<tr><td>θ_9</td><td>1.120</td></tr>
<tr><td>θ_{10}</td><td>5.506</td></tr>
<tr><td>σ_n</td><td>0.025</td></tr>
</table>

Note:
The chosen kernel function $k(x_i, x_j)$ allows for relevance determination, such as the optimal values of the hyperparameters within matrix P define the relevance of the contribution of the specific input within the overall model structure: large hyperparameter values indicate high relevance to the model, while small values indicate input covariates of lower importance to predicting the rate of tire tread wear. For example, hyperparameter θ_1 (corresponding to the odometer reading) is relatively small indicating low impact of accumulated mileage on tire tread wear rate. Relatively high values of hyperparameters θ_4 and θ_5 indicate high importance of road conditions on tire tread wear rate.</td></tr>
</table>

<table>
<tr><th rowspan="2">Elements of Agent Learning Process</th><th>Type II Macro-Agent</th></tr>
<tr><th>Automotive Tire Tread Wear Rate</th></tr>
<tr><td>New Data</td><td>The new set of training data is obtained from the collection of unscheduled maintenance actions on a fleet of similar vehicles that are used for various jobs in different locations. The data collection is performed on a monthly basis, and the objective is to understand any differences in the underlying tread wear rate that may exist across the fleet. The values in the input vector X are the various factors that may impact the tread wear rate for a vehicle tire in the fleet during a given month: accumulated mileage, M, vehicle model, V, and predominant road conditions, R.

The new training data points were used to develop the Log-GPR model with the X inputs associated with the month number and the $Log(Y)$ being the log of the fleet tire tread wear rate each month.</td></tr>
<tr><td>Model of Agent Output Variable
- Updated Beliefs</td><td>Same as “Representation of Output Set - Past Beliefs”</td></tr>
<tr><td>Parameters of Agent Output Model
- Updated Beliefs</td><td>The optimal values of the updated hyperparameters and noise level parameter were computed, as follows:
<table>
<tr><th>Parameter</th><th>Value</th></tr>
<tr><td>θ_1</td><td>24.629</td></tr>
<tr><td>θ_2</td><td>0.297</td></tr>
<tr><td>θ_3</td><td>0.405</td></tr>
<tr><td>θ_4</td><td>987.541</td></tr>
<tr><td>θ_5</td><td>412.974</td></tr>
<tr><td>θ_6</td><td>22.113</td></tr>
<tr><td>θ_7</td><td>5.365</td></tr>
<tr><td>θ_8</td><td>8.174</td></tr>
<tr><td>θ_9</td><td>3.156</td></tr>
<tr><td>θ_{10}</td><td>3.704</td></tr>
<tr><td>σ_n</td><td>0.003</td></tr>
</table></td></tr>
</table>

5.6. Bayesian Inference

In Bayesian inference, the initial belief about the distribution of the parameters of the model of agent output variable (a priori distribution) is systematically updated according to Bayes' theorem based on new data (evidence) to obtain a posteriori distribution of the model parameters [68], [69], [93], [94]. New data are recursively added to the existing estimates, so that there is no need to combine new data with all past observations and to re-run the analysis. This reduces the computational time by building on the agent's "memory" about the history of all past updates. Bayesian formalism is flexible enough to combine data from different sources, including partially relevant data. For these reasons, Bayesian inference is recommended as one of the primary methods of agent learning for Type I Micro-Agents and Type II Macro-Agents. Bayesian inference applies when the PoF or empirical model of the agent output variable is or is not available, and can be used as parameter estimation method for the majority of functional forms of the agent output variable including standard parametric distributions, model-based parametric distributions (cumulative damage models), time series and trend functions (degradation or other trends), and some machine learning algorithms (e.g., Gaussian process regression [92]).

The required data for agent learning via Bayesian inference include direct measurements of the agent output quantity $Y(T)$, $T = \{t_1, t_2, \ldots, t_n\}$, and the input quantities t_i ($i = 1, \ldots, n$), where $Y(T)$ is could be a degradation measure (e.g. crack size), time to failure/success, environmental or operational characteristic. Any type of data are useable, such as test results, inspection and maintenance records in field operation, records from sensors and monitors (usually in PHM applications), data from generic sources, and expert opinion.

The examples shown in Table 5-5, Table 5-6, Table 5-7 and Table 5-8 demonstrate different cases of Bayesian updating in the context of agent learning. Bayesian formulation takes a different form depending on the type of problem and the available data (evidence). When dealing with test or field data collected under time-varying stresses, the life-stress relationship must take into account the cumulative effect of the applied stresses. Cumulative damage (cumulative exposure) models are described in publication [72], which presents a derivation of the model parameters by classical estimation methods. Bayesian formulation for the cumulative damage model parameters is demonstrated by the example in Table 5-7. It must be noted that Bayesian updating of the cumulative damage model parameters cannot be done recursively due to the nature of cumulative damage equations, and new data must be combined with all past observations in order to re-run the analysis. In this exceptional case, the only advantage of the Bayesian inference over the classical methods of parameter estimation is the ability to use the first set of prior estimates of model parameters available from the past before any data were obtained within the current study of system reliability.

A disadvantage of Bayesian inference as an agent learning method is significant computational effort. Software programs existing today (WinBUGS [120], [121], R-Dat, R-Dat Plus and BRASS [95]) have a limited ability to handle complex non-conjugate Bayesian formulations typical for most applications of homogeneous Bayesian inference (including Bayesian inference for cumulative damage model). A software program R-Dat Plus also handles non-homogeneous Bayesian inference [95], but has similar limitations when complex non-conjugate Bayesian formulations are concerned. Significant developments in this area are necessary.

Table 5-5: Example of Type I Micro-Agent Learning by Bayesian Inference

Elements of Agent Learning Process	**Type I Micro-Agent**
	Operating Temperature
Input Variables from Input Agents	Not applicable
Output Variable	Operating Temperature of the Component, T
Model of Agent Output Variable - Past Beliefs	Variation of operating temperature of the mechanical component is defined by normal distribution with parameters μ and σ, as follows: $T \sim N(\mu,\sigma)$
Parameters of Agent Output Model - Past Beliefs	The vector of model parameters: $\theta = (\mu, \sigma)^T$ Prior beliefs about the agent output model parameters are established according to engineering knowledge and historical field data as follows: $\mu \sim N(m_1, s_1), \sigma \sim N(m_2, s_2)$ where m_1, s_1, m_2, s_2 are known values.
New Data	Temperature measurements $D\{T_1, T_2, \ldots, T_n\}$ obtained from the field are used as an evidence to update prior beliefs of Operating Temperature Agent into the posterior distribution of the agent output model parameters, μ and σ.
Model of Agent Output Variable - Updated Beliefs	The updated model of the agent output variable, T, is developed, as follows: 1. Priori distributions of the agent output model parameters μ and σ are defined in "Parameters of Agent Output Model - Past Beliefs" section of this table: $\mu \sim N(m_1, s_1), \sigma \sim N(m_2, s_2)$ where m_1, s_1, m_2, s_2 are known values. 2. Temperature measurements $D\{T_1, T_2, \ldots, T_n\}$ are used as data (evidence) for Bayesian updating.

Elements of Agent Learning Process	**Type I Micro-Agent**
	Operating Temperature
	3. The Likelihood function of the evidence (reasoning property of the agent): $$L(D\|\mu,\sigma) \propto \frac{1}{\sigma^n} exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(T_i-\mu)^2\right]$$ 4. Priori distribution of the agent output model parameters is the following joint distribution of μ and σ (assuming their independence): $$\pi_0(\mu,\sigma) \propto \frac{1}{s_1 s_2} exp\left[-\frac{(\mu-m_1)^2}{2(s_1)^2}\right] exp\left[-\frac{(\sigma-m_2)^2}{2(s_2)^2}\right]$$ 5. Bayesian formulation for posteriori probability of the agent output model parameters according to Bayes' theorem: $$\pi(\mu,\sigma\|D) = \frac{L(D\|\mu,\sigma)\pi_0(\mu,\sigma)}{\iint L(D\|\mu,\sigma)\pi_0(\mu,\sigma)d\mu d\sigma}$$ 6. Posterior predictive distribution of Operating Temperature variable resulted from the learning process becomes: $$\bar{f}(T) = \iint f(T\|\mu,\sigma)\pi(\mu,\sigma\|D)\, d\mu d\sigma$$ This formulation provides the new (updated) probability distribution of the agent output variable Operating Temperature, *T*. The term $f(T\|\mu,\sigma)$ represents a formulation of normal distribution of the variable *T*, $N(\mu,\sigma)$: $$f(T\|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(T-\mu)^2}{2\sigma^2}\right)$$
Parameters of Agent Output Model - Updated Beliefs	Bayesian formulation for posteriori probability of the model parameters provides the updated beliefs about the agent output model parameters (as obtained in "Model of Agent Output Variable - Updated Beliefs" section of this table): $$\pi(\mu,\sigma\|D) = \frac{L(D\|\mu,\sigma)\pi_0(\mu,\sigma)}{\iint L(D\|\mu,\sigma)\pi_0(\mu,\sigma)d\mu d\sigma}$$

Table 5-6: Example of Type II Macro-Agent Learning by Bayesian Inference

Elements of Agent Learning Process	**Type II Macro-Agent**
	Life to Fatigue Failure
Input Variables from Input Agents	Type I Micro-Agents: Amplitude of Applied Alternating Stress, *S* Note: *Life to Fatigue Failure Macro-Agent is not time dependent (no time trends exist).*
Output Variable	Life to Fatigue Failure of Structural Part, *N* Note: *The structure of this Macro-Agent and the respective Bayesian formulations are also shown on* Figure 2-4.
Model of Agent Output Variable - Past Beliefs	1. PoF life-stress model of time to fatigue failure: $N = \frac{1}{K \cdot S^n}$ where *S* is stress amplitude, *K* is proportionality constant and *n* is power parameter. 2. The agent output variable *N* is assumed to follow lognormal distribution with shape parameter σ and median defined according to PoF life-stress model of time to fatigue failure shown above: $N \sim LN(\mu, \sigma)$ where log-mean μ is assigned as $\mu = \log(Median) = \log\left(\frac{1}{K \cdot S^n}\right)$ 3. Life-stress multivariate lognormal distribution of time to fatigue failure variable *N*: $f(N\|S, K, n, \sigma) = \frac{1}{\sigma N \sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{log(N) + log(K) + nlog(S)}{\sigma}\right)^2\right]$

<table>
<tr><th rowspan="2">Elements of Agent Learning Process</th><th>Type II Macro-Agent</th></tr>
<tr><th>Life to Fatigue Failure</th></tr>
<tr><td>Parameters of Agent Output Model
- Past Beliefs</td><td>The vector of model parameters: $\theta = (K, n, \sigma)^T$
Priori distribution of agent output model parameters K, n, σ is given by joint probability distribution derived from uninformative (uniform) prior distributions established based on engineering judgment:
$$\pi_0(K, n, \sigma) = \pi_0(K) \cdot \pi_0(n) \cdot \pi_0(\sigma)$$
where
$$\pi_0(K) = U(K_{min}, K_{max})$$
$$\pi_0(n) = U(n_{min}, n_{max})$$
$$\pi_0(\sigma) = U(\sigma_{min}, \sigma_{max})$$</td></tr>
<tr><td>New Data</td><td>The data are obtained from accelerated fatigue test where stress amplitude S remains constant during the test.
The test data D comprising F complete failure observations t_i and S right censored observations T_j are used as evidence for Bayesian updating: $D\ \{(t_i, T_j),\ i = 1, \ldots F,\ j = 1, \ldots, S\}$.</td></tr>
<tr><td>Model of Agent Output Variable
- Updated Beliefs</td><td>The updated model of the agent output variable N is developed, as follows:
1. Priori distribution of the agent output model parameters $\pi_0(K, n, \sigma)$ is defined in “Parameters of Agent Output Model - Past Beliefs” section of this table.
2. The test data $D\ \{(t_i, T_j),\ i = 1, \ldots F,\ j = 1, \ldots, S\}$ are used as an evidence for Bayesian updating.
3. The Likelihood function of the evidence becomes:
$$L(D|K, n, \sigma) = L(t_1, \ldots, t_F, T_1, \ldots, T_S|K, n, \sigma)$$
$$= \prod_{i=1}^{F} f(t_i|K, n, \sigma) \prod_{j=1}^{S} R\left(T_j\middle|K, n, \sigma\right)$$
Reliability function in the above equation is given by:
$$R\left(T_j\middle|S, K, n, \sigma\right) = 1 - \int_0^{T_j} f(x|S, K, n, \sigma)dx$$
4. Bayesian formulation for posteriori probability of the agent output model parameters according to Bayes’ theorem:
$$\pi(K, n, \sigma|D) = \frac{L(D|K, n, \sigma)\pi_0(K, n, \sigma)}{\iiint L(D|K, n, \sigma)\pi_0(K, n, \sigma)dKdnd\sigma}$$</td></tr>
</table>

Elements of Agent Learning Process	Type II Macro-Agent Life to Fatigue Failure
	5. Posterior predictive distribution of Life to Fatigue Failure agent *N* is an outcome of the agent learning process: $$\bar{f}(N,S) = \iiint f(N\|S,K,n,\sigma)\pi(K,n,\sigma\|D)\, dKdnd\sigma$$ This function provides the new (updated) probability distribution of the agent output variable Life to Fatigue Failure *N*. The term $f(N\|S,K,n,\sigma)$ represents a formulation of lognormal distribution of the variable *N*, as defined in "Model of Agent Output Variable - Past Beliefs" section of this table.
Parameters of Agent Output Model - Updated Beliefs	Bayesian formulation for the posteriori probability of the model parameters represents the updated beliefs about the agent output model parameters (as obtained in "Model of Agent Output Variable - Updated Beliefs" section of this table): $$\pi(K,n,\sigma\|D) = \frac{L(D\|K,n,\sigma)\pi_0(K,n,\sigma)}{\iiint L(D\|K,n,\sigma)\pi_0(K,n,\sigma)dKdnd\sigma}$$

Table 5-7: Example of Type II Macro-Agent Learning by Bayesian Inference

Elements of Agent Learning Process	Type II Macro-Agent Life to Failure (Cumulative Damage)
Input Variables from Input Agents	Type II Macro-Agents: Stress Variable, *x(t)* Type I Micro-Agents: Operational or Test Time, *t* (indirect input to Life to Failure agent via Stress Variable agent *x(t)*) Note: *Life to Failure Macro-Agent is time dependent via Stress Variable agent x(t)*
Output Variable	Life to Failure, *L*
Model of Agent Output Variable - Past Beliefs	1. The PoF life-stress model of time to failure *L*: $$L\big(x(t)\big) = \eta(t,x) = \left[\frac{K}{x(t)}\right]^n$$ where stress variable *x* is a function of time t, x(t), parameter *K* is proportionality constant, *n* is power parameter, β is shape parameter and η is scale parameter (characteristic life) of Weibull

Elements of Agent Learning Process	Type II Macro-Agent
	Life to Failure (Cumulative Damage)
	time-to-failure distribution $W(\beta,\ \eta)$. 2. Weibull reliability function is given by: $$R(t,x) = exp\left[-\left(I(t,x)\right)^{\beta}\right]$$ where the integral $I(t,x)$ is cumulative damage equation: $$I(t,x) = \int_0^t \left[\frac{x(u)}{K}\right]^n du$$ 3. Weibull probability density function (pdf) is given by: $$f(t,x) = \frac{\beta}{\eta(t,x)}\left(I(t,x)\right)^{\beta-1} exp\left[-\left(I(t,x)\right)^{\beta}\right]$$
Parameters of Agent Output Model - Past Beliefs	The vector of the agent output model parameters: $\theta = (K,\ n,\ \beta)^T$ Priori distribution of model parameters $K,\ n,\ \beta$ could take any form depending on the available engineering knowledge. For example, it could be joint probability distribution derived from uninformative (uniform) prior distributions established based on engineering judgment: $$\pi_0(K,n,\beta) = \pi_0(K)\cdot\pi_0(n)\cdot\pi_0(\beta)$$ where $$\pi_0(K) = U(K_{min}, K_{max})$$ $$\pi_0(n) = U(n_{min}, n_{max})$$ $$\pi_0(\beta) = U(\beta_{min}, \beta_{max})$$
New Data	The data are obtained from accelerated testing where stress x is a function of test time, $x(t)$. Test data D comprising F complete failure observations t_i and S right censored observations T_j are used as evidence for Bayesian updating: $D\ \{(t_i, T_j),\ i = 1,\ \ldots F,\ j = 1,\ \ldots,\ S\}$
Model of Agent Output Variable - Updated Beliefs	Same as "Model of Agent Output Variable - Past Beliefs"
Parameters of Agent Output Model - Updated Beliefs	The updated beliefs about parameters of the agent output model are developed, as follows: 1. Priori distribution of model parameters $\pi_0(K,n,\beta)$ is defined in "Parameters of Agent Output Model - Past Beliefs" section of this table.

Elements of Agent Learning Process	Type II Macro-Agent
	Life to Failure (Cumulative Damage)
	2. The test data *D* $\{(t_i,T_j), i = 1, \ldots F, j = 1, \ldots, S\}$ are used as an evidence for Bayesian updating. 3. The Likelihood function of the evidence is the following: $$L(D\|K,n,\beta) = \prod_{i=1}^{F} f(T_i,x_i) \prod_{j=1}^{S} R(T_j,x_j)$$ where $x_i = x(T_i)$, $x_j = x(T_j)$, T_i is the exact failure time of the *i*-th failure observation, T_j is the running time of the *j*-th suspension. Considering the equations of reliability function and probability density function (pdf), shown in "Model of Agent Output Variable - Past Beliefs" section of this table, the likelihood becomes: $$L(D\|K,n,\beta) = \\ = \prod_{i=1}^{F} \left\{ \frac{\beta}{\eta(t_i,x_i)} \left(I(t_i,x_i)\right)^{\beta-1} exp\left[-\left(I(t_i,x_i)\right)^{\beta}\right]\right\} \\ \times \prod_{j=1}^{S} exp\left[-\left(I(T_j,x_j)\right)^{\beta}\right]$$ Upon substitution of the equation of PoF life-stress model of time to failure *L(x(t))* and the integral *I(t,x)* (given in "Model of Agent Output Variable - Past Beliefs" section of this table) the final expression of the likelihood is obtained as: $$L(D\|K,n,\beta) = \\ = \prod_{i=1}^{F} \left\{ \beta \left[\frac{x(T_i)}{K}\right]^n \left[\int_0^{T_i} \left[\frac{x(u)}{K}\right]^n du\right]^{\beta-1} exp\left[-\left[\int_0^{T_i} \left[\frac{x(u)}{K}\right]^n du\right]^{\beta}\right]\right\} \\ \times \prod_{j=1}^{S} exp\left\{-\left[\int_0^{T_j} \left[\frac{x(u)}{K}\right]^n du\right]^{\beta}\right\}$$ 4. Bayesian formalism for the posterior distribution of agent output model parameters *K, n* and β can be expressed as: $$\pi(K,n,\beta\|D) = \frac{L(D\|K,n,\beta)\pi_0(K,n,\beta)}{\iiint L(D\|K,n,\beta)\pi_0(K,n,\beta)dKdnd\beta}$$ This posterior distribution of model parameters provides the new (updated) beliefs about the agent output variable Life to Failure, *L*.

Table 5-8: Example of Type II Macro-Agent Learning by Bayesian Inference

<table>
<tr><th rowspan="2">Elements of Agent Learning Process</th><th>Type II Macro-Agent</th></tr>
<tr><th>Crack Size</th></tr>
<tr><td>Input Variables from Input Agents</td><td>Type I Micro-Agents:
1. Applied Stresses, S:
1a) Cyclic Mechanical Stress, σ
1b) Frequency of the Applied Cyclic Load, v
1c) Corrosion Current, I_p
2. Test Time (Number of Cycles of Load Application), N
Note:
Crack Size agent is time dependent via test time N.</td></tr>
<tr><td>Output Variable</td><td>Crack Size (Crack Depth) in a Structural Part, a</td></tr>
<tr><td>Model of Agent Output Variable
- Past Beliefs</td><td>In the works [96] and [97], a combined probabilistic physics-of-failure-based model for pitting and corrosion-fatigue degradation mechanisms is proposed to estimate the reliability of structures (such as pipes and steam generator tubes in power plants and oil pipelines). Bayesian updating formalism, described in [96] and [97], is shown in this table as an example of learning method for Type II Macro-Agent representing a degradation attribute of an item (specifically crack size of a structural part).
1. PoF model connecting the environmental degradation factors with accumulated damage (i.e., crack depth, a) is the following:
$$a(A,B;S,N) = \\ = A(\sigma^{0.182}\nu^{-0.228}Ip^{0.248})N^{1/3} + B(\sigma^{3.24}\nu^{-0.377}Ip^{0.421})N^2 exp[(4 \times 10^{-10}\sigma^{2.062}\nu^{-0.024})N]$$
where the applied mechanical and environmental stress agents are cyclic load frequency, ν, cyclic mechanical stress, σ, and corrosion current, I_p, and N is number of cycles of load application. The model parameters, A and B, depend on material properties only (i.e., they are independent of the effects of applied mechanical and environmental stresses).
2. It is assumed that lognormal distribution represents the variability of crack size a, where μ and σ are the log-mean and log-standard deviation of the crack size distribution:
$$f(a) = LN(\mu, s)$$
3. Using PoF model of the crack length, the log-mean of the crack size distribution is expressed as follows:</td></tr>
</table>

Elements of Agent Learning Process	Type II Macro-Agent
	Crack Size
	$\mu(A,B;S,N) = = Ln\big[A\sigma^{0.182}\nu^{-0.228}Ip^{0.248}N^{1/3} + B\sigma^{3.24}\nu^{-0.377}Ip^{0.421}N^{2}exp(4 \times 10^{-10}\sigma^{2.062}\nu^{-0.024}N)\big]$ 4. The corrosion-fatigue model is separated into pitting-corrosion and corrosion-fatigue parts, dividing the above equation into the following two: $\mu_{pc}(A;S,N) = Ln\big[A\sigma^{0.182}\nu^{-0.228}Ip^{0.248}N^{1/3}\big]$ $\mu_{cf}(B;S,N) = Ln[B\sigma^{3.24}\nu^{-0.377}Ip^{0.421}N^{2}exp(4 \times 10^{-10}\sigma^{2.062}\nu^{-0.024}N)]$ 5. Substituting above two equations into lognormal distribution model of crack size *a* yields the conditional lognormal distribution functions of the crack size *a* given stress conditions and cycles of load, for the pitting corrosion and the corrosion-assisted fatigue parts: $f\big(a_{pc,i}\big\|\sigma_i,\nu_i,Ip_{i,}N_i\big) = = \frac{1}{s_{pc}a_{pc,i}\sqrt{2\pi}}exp\Big\{-\frac{1}{2s_{pc}^2}\big[Ln\big(a_{pc,i}\big) - Ln\big(A\sigma_i^{0.182}\nu_i^{-0.228}Ip_i^{0.248}N_i^{1/3}\big)\big]^2\Big\}$ $f\big(a_{cf,i}\big\|\sigma_i,\nu_i,Ip_{i,}N_i\big) = = \frac{1}{s_{cf}a_{cf,i}\sqrt{2\pi}}exp\Big\{-\frac{1}{2s_{cf}^2}\big[Ln\big(a_{pc,i}\big) - Ln\Big(B\sigma_i^{3.24}\nu_i^{-0.377}Ip_i^{0.421}N_i^{2}exp(4 \times 10^{-10}\sigma_i^{2.062}\nu_i^{-0.024}N_i)\Big)\big]^2\Big\}$ where s_{pc} and s_{cf} are the standard deviation of the log-normal distribution of crack size when pitting-corrosion or corrosion-fatigue is dominant, respectively.
Parameters of Agent Output Model - Past Beliefs	Vector of the agent output model parameters: $\theta\,(A, B, s_{pc}, s_{cf})^{-1}$. Parameters *A*, *B*, s_{pc}, s_{cf} were initially estimated using the generic data produced from a benchmark model and used as prior estimates of the PoF model parameters within Bayesian updating framework (shown in "Model of Agent Output Variable - Updated Beliefs" section of this table).

Elements of Agent Learning Process	**Type II Macro-Agent**
	Crack Size
New Data	The data $D\{S_i, N_i, a_{pc,i}, a_{cf,i}\}$ are obtained during pitting-corrosion and corrosion-fatigue lab testing conducted under the constant stresses. The data include number of cycles N_i, applied stresses S_i (pitting-corrosion current, applied mechanical loads, load frequency) and crack size measurements.
Model of Agent Output Variable - Updated Beliefs	Same as "Model of Agent Output Variable - Past Beliefs"
Parameters of Agent Output Model - Updated Beliefs	Updated beliefs about parameters of the agent output model are developed, as follows: 1. The likelihood functions for the pitting corrosion and the corrosion-assisted fatigue parts are: $L(a_{pc,i}, S_i, N_i \| A, s_{pc}) = \prod_{i=1}^{N_{tr}} f(a_{pc,i} \| \sigma_i, \nu_i, Ip_{i,} N_i), \ \ for\ 0 \leq N_i \leq N_{tr}$ $L(a_{cf,i}, S_i, N_i \| B, s_{cf}) = \prod_{i=1}^{N_f - N_{tr}} f(a_{cf,i} \| \sigma_i, \nu_i, Ip_{i,} N_i), \ \ for\ N_{tr} \leq N_i \leq N_f$ where, N_{tr} and N_f are transition and final number of cycles in the corrosion-fatigue experiment, respectively. 2. Test data $D\{S_i, N_i, a_{pc,i}, a_{cf,i}\}$ are used as an evidence for Bayesian updating. 3. Bayesian formulation for posteriori probability of the crack size model parameters θ is the following: $\pi(\theta \| D) = \frac{L(D \| \theta)\pi_0(\theta)}{\iiint L(D \| \theta)\pi_0(\theta) d\theta}$ Upon substitution of the expressions of data likelihood functions for the pitting corrosion and the corrosion-assisted fatigue parts into the above equation will obtain updated form of posterior probability of crack size model parameters θ:

Elements of Agent Learning Process	Type II Macro-Agent
	Crack Size
	$\pi(\theta\|D) = \pi\left(A, B, s_{pc}, s_{cf} \middle\| a_{pc,i}, a_{cf,i}, S_i, N_i\right) = = \frac{L\left(a_{pc,i}, S_i, N_i \middle\| A, s_{pc}\right) L\left(a_{cf,i}, S_i, N_i \middle\| B, s_{cf}\right)\pi_0(\theta)}{\int L\left(a_{pc,i}, S_i, N_i \middle\| A, s_{pc}\right) L\left(a_{cf,i}, S_i, N_i \middle\| B, s_{cf}\right)\pi_0(\theta) d\theta}$ As an outcome of agent learning, this probabilistic function provides the updated beliefs of the agent crack size a, specifically the updated distribution of the agent output model parameters, vector θ.

5.7. Overview of Data Fusion Methods

The ultimate goal of the agent-oriented PoF modeling of system reliability is to evaluate the progression of the degradation processes within a system in order to predict the long-term evolution of damage accumulation and the failure time based on the anticipated future usage profile. The necessary data for agent learning are often obtained from various sources. It is critical to be able to combine all independent sources of information in order to achieve a more accurate assessment of system reliability. Further discussion about data fusion pertains to hardware parts, but conceptually similar approaches may also apply to software and human elements of the system.

Data fusion methods are especially useful when data are collected during in-field health monitoring in addition to the available data from other sources, such as reliability testing, in-service inspection and maintenance records, published generic data, and engineering analysis. If an automated health management system is the source of data for agent learning, then the fusion of data and information can happen at different levels [98], as shown in Figure 5-4. At each level, the outcomes of the previous levels are fused together with the objective to improve the overall prognostics of time to failure.

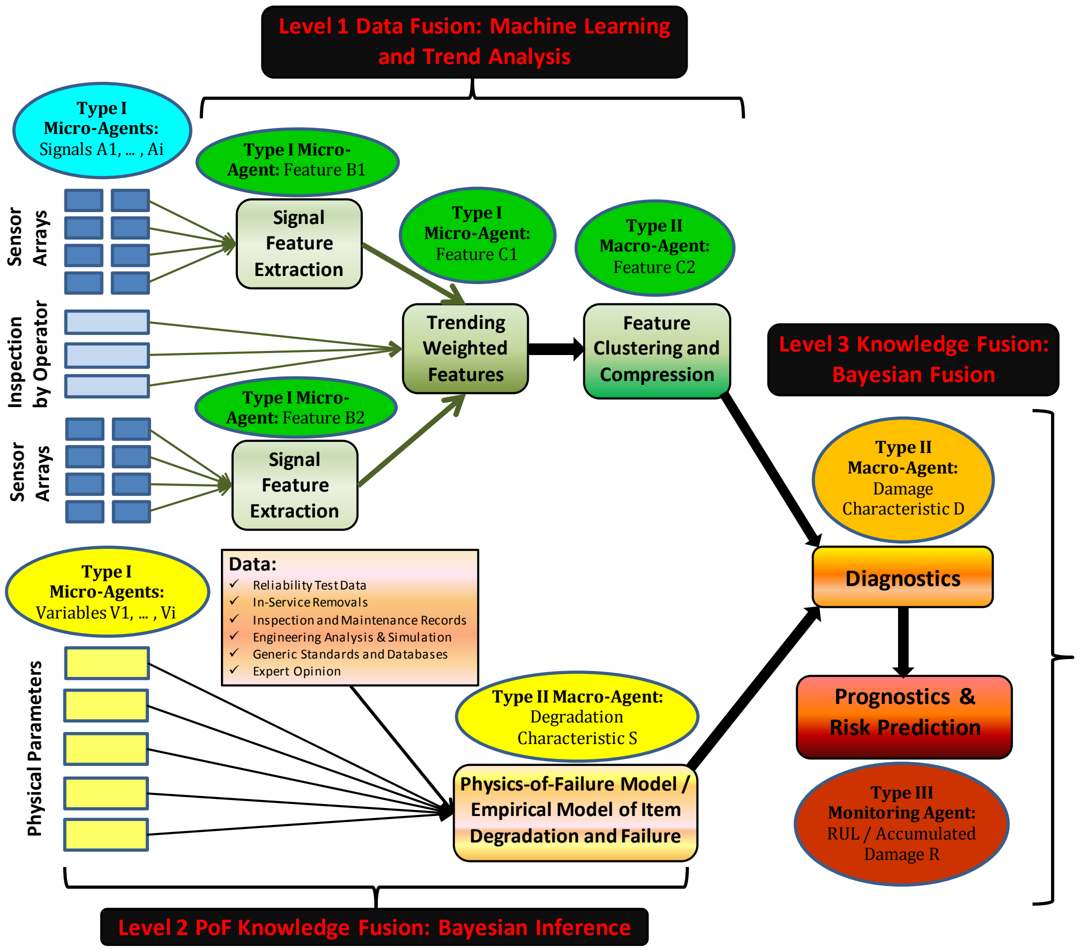


Figure 5-4: Data Fusion Architecture for Type II Macro-Agents

At the lowest level (Level 1 on Figure 5-4), data coming from an array of sensors or inspection by an operator can be combined to validate the signals or to create new features. At this level, one must use signal-processing or decision making techniques that can read the available sensor outputs and provide information about the quantity or quality of interest. Machine learning methods (described in Section 5.5) and trend analysis techniques (described in Section 5.4) can be utilized to combine (fuse) the signals from multiple sensors into an output feature. The combination of available sensors and signal processing routines is also known as a virtual sensor (soft sensor/analytic sensor). A schematic representation of artificial neural network as a virtual sensor is shown in Figure 5-5 [98]. A virtual sensor may be implemented as an artificial neural network, a Kalman filter, a look-up table, a fuzzy-logic expert system, or another similar mapping tool of machine learning or trend analysis. In terms of agent-oriented system modeling, the lowest level of data fusion means that output variables of multiple Type I Micro-Agents (variables measured by each sensor or by human operator) will form another Type I Micro-Agent or Type II Macro-Agent (a new feature formed by fusion) which learns by means of the chosen fusion method (from the available machine learning or trend analysis methods described in Sections 5.4 and 5.5).

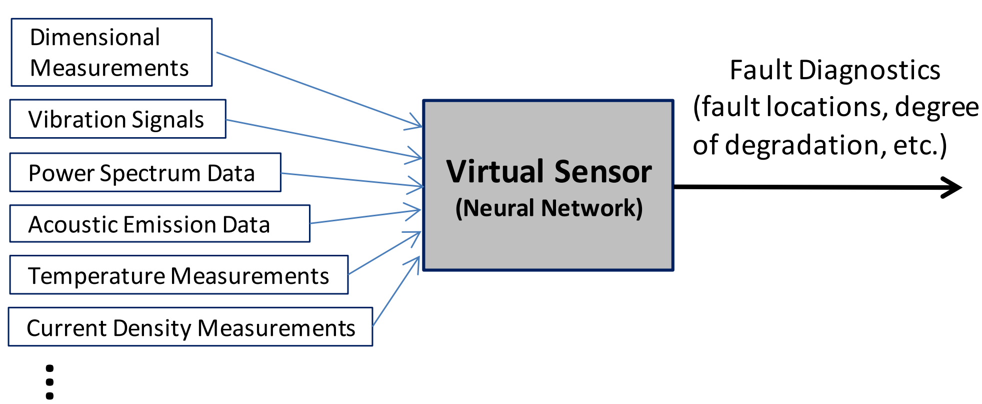


Figure 5-5: Artificial Neural Network as a Virtual Sensor

Once feature extraction is performed on signals from the individual sensors, the extracted features are combined to obtain better information about the item of interest. For instance, in acoustic emission (AE) monitoring described in [99], the AE count rate feature is calculated by first extracting the AE count feature directly from the raw signals, and then calculating its rate of change with respect to elapsed fatigue cycles. Another example is when a feature related to the particle count and size in a bearing's lubrication oil was fused with a vibration characteristic, such as kurtosis [98]. The combined result yielded an improved confidence about the bearing's health. Several fusion architectures exist for processing the multi-sensor data during different stages within Level 1 data fusion, the most common are described in [98]. An agent-oriented model developer will choose the appropriate architecture based on the sensor network size, amount of data collected, required accuracy, and available computational resources.

Data fusion at Level 2 is performed by means of the Bayesian inference described in Section 5.6 through the PoF or empirical model of failure for the item of interest. The most complex knowledge fusion occurs at Level 3, where experience-based information such as legacy failure rates and physics-based modeling architectures are incorporated with signal-based data. The most common fusion approaches are [98]:

1. *Bayesian Fusion*

   Bayesian fusion is based on the recursive Bayesian estimation technique for the fusion of information from multiple sources. Recursive Bayesian estimation (Bayes filter) is a probabilistic approach for estimating an unknown probability density function recursively over time using incoming uncertain observations (noisy measurements) and a mathematical process model that describes the evolution of the state variables over time. This method is recommended for agent learning when data fusion is required, as shown on Figure 5-4 for "Type II Macro-Agent Damage Characteristic D". A detailed description of the Bayesian fusion methods and their application to agent learning is outlined in Section 5.8.

2. *Dempster-Shafer Fusion*

The Dempster-Shafer reasoning method generalizes the Bayesian inference to support not only a single hypothesis but also the union of hypotheses, which could contain any possible hypotheses including a nested hypothesis (a hypothesis that is a subset of another hypothesis), non-mutually exclusive (overlapping) hypotheses, and an ignorance hypothesis. The Dempster-Shafer theory of evidence and Bayesian inference produce identical results if all the hypotheses in the study are mutually exclusive and are not nested.

The Dempster-Shafer methodology is often used to combine the sensor outputs where subjective judgments are present. The Dempster-Shafer fusion approach lacks a well-established decision theory, whereas the Bayesian decision theory maximizes the expected utility and is almost universally accepted. The above definitions suggest that Dempster-Shafer fusion may be used as an agent learning method in cases where the data are coming from various sources (different sensors, inspection records), some of the available data are subjective, and human interactions are present. Specifically, different types of sensors may use different physical principles, cover different data space, provide data in different formats, and the generated data can have a different resolution and accuracy. In addition, human inspection records may contain subjective judgments and impose the same challenges as noted above for the sensors data. The Dempster-Shafer theory of evidence could be used as a generalized fusion solution to overcome data context difficulties and efficiently combine the outputs from multiple sensors and human inspection data.

3. *Neural-Network Fusion*

   A well-accepted application of artificial neural networks (ANNs) is data and feature fusion. This method however, requires large amount of training data, which is not expected to be available when the fusion of signal-based data and physical model predictions is required. This type of data fusion is not recommended as an agent learning method; however, it could be used when applicable, to the discretion of the modeler.

5.8. Bayesian Fusion Methods

Bayesian fusion, also known as Bayes filtering or recursive Bayesian estimation, is one of the most common data fusion methodologies. It involves the recursive estimation of the unknown probability density function over time [98]. Recursive Bayesian estimation has two elements: a time update and a measurement update [100]. In the context of agent learning, the time update comes from the understanding of previous values of the agent output variable and how they relate to the current point in time according to the PoF or empirical model of the agent output variable. The measurement update involves inferring information from observations made at the current point in time. Recursive Bayesian estimation combines both elements to improve the systemic understanding of state evolution of the associated system element. For a comprehensive review of the concept of recursive Bayesian estimation and the existing methodologies of Bayesian fusion, the interested reader is referred to publications [100], [101].

Bayesian fusion is suitable as a learning method for Type I Micro-Agents or Type II Macro-Agents with one or more input agents. The PoF or empirical model of the agent output variable should be available and model parameters must be fully defined (i.e. all model parameters are assigned with known constant values). In addition, all input agents should be set in "inactive" status (i.e. constant values are assigned to their output variables). The required data for agent learning include measurements of the agent output quantity $Y(T)$, $T = \{t_1, t_2, \ldots, t_n\}$, and the input quantities t_i ($i = 1, \ldots, n$), where $Y(T)$ is a degradation measure (e.g. crack size), environmental or operational characteristic. Typical data sources are sensors and monitors (usually in PHM applications). When new data set becomes available, it is combined with the model-based prediction for the current point in time to update past estimates of the Bayesian fusion model parameters obtained during the previous learning cycle. Several examples of agent learning via Bayesian fusion are provided in Sections 5.8.1 and 5.8.2.

5.8.1. Discrete-Time Kalman Filter

A discrete-time Kalman filter (KF) can be used if discrete measurements are taken, whether they come from a discrete or a continuous system. A continuous-time Kalman filter is used when the measurements are continuous functions of time. This section describes the applications of the discrete-time Kalman filter as a learning method, considering that only discrete data are used for agent learning.

Discrete-time Kalman filter generates closed formed recursive solutions as it makes three simplifying assumptions [100]. The first assumption is that both the state transition function and the observation function are linear. The second is that each state variable is normally distributed. The third is that the 'noise' factors are normally distributed. In order to use the discrete-time Kalman filter as a method of agent learning, the following must be defined for the respective Type I Micro-Agent or Type II Macro-Agent:

1. The state variable to be estimated – this is the agent output variable.
2. A state transition function that defines the evolution of state variables over time – this is the PoF or empirical model of the agent output variable.
3. An observation function that defines how various observations over time (data) are related to the state variables – this is the reasoning algorithm (causal relationship) used to correlate the measured quantity (e.g. precursor values) with the agent output variable.
4. The noise (uncertainty) in both the process model and the observation model – this is assumed model error.

An example of Type II Macro-Agent learning by the discrete-time Kalman filter is given in Table 5-9.

Table 5-9: Example of Type II Macro-Agent Learning by Kalman Filter

<table>
<tr><th rowspan="2">Elements of Agent Learning Process</th><th>Type II Macro-Agent</th></tr>
<tr><th>Roller Damage due to Wear (Wear Depth)</th></tr>
<tr><td>Input Variables from Input Agents</td><td>Type I Micro-Agents:
1. Accumulated number of passes, p
2. Material shear yield point, τ_{yp}
3. Stress concentration factor, k_e
4. Roller load (due to weight of the supported structure), W
5. Roller diameter, d
6. Width of the contact surface of the roller, w
7. Friction coefficient, f

Type II Macro-Agents:
1. Wear rate, $\dot{D}$
2. Maximum shear stress in the vicinity of the surface, τ_{max}
3. Normal stress on the surface, σ_n
4. Friction generated shear stress, τ_f

Notes:
1. Wear Depth agent is time dependent via Accumulated Number of Passes Agent, p
2. All input agents have “Inactive” status (i.e. constant values are assigned to their output variables).</td></tr>
<tr><td>Output Variable</td><td>Roller Wear Depth, D</td></tr>
<tr><td>Model of Agent Output Variable
- Past Beliefs</td><td>1. The PoF model of roller wear depth is defined as
$$D = \dot{D} \cdot p$$
where wear rate $\dot{D}$ is considered to be a constant value calculated as
$$\dot{D} = B\left[\frac{\tau_{max}}{\tau_{yp}}\right]^k$$
where B and k are model parameters, and the other terms are:
$$\tau_{max} = k_e\sqrt{\left(\frac{\sigma_n}{2}\right)^2 + \left(\tau_f\right)^2}$$</td></tr>
</table>

Elements of Agent Learning Process	Type II Macro-Agent
	Roller Damage due to Wear (Wear Depth)
	$$\sigma_n = \frac{W}{d \cdot w}$$ $$\tau_f = f \cdot \sigma_n$$ 2. The roller wear depth *D* is considered a system state variable. According to the Kalman filter assumptions, the state variable D_k is Gaussian, specified by its first and second moments: $$D_k \sim Normal(\widehat{D}_k, P_k)$$ where $\widehat{D}_k$ and P_k denote the first and second moments of the wear depth distribution (i.e., posterior mean and posterior covariance) at time step k. 3. A general form of *process model* equations is: $$\hat{x}_{\bar{k}} = A\hat{x}_{k-1} + Bu_k$$ $$P_{\bar{k}} = AP_{k-1}A^T + Q$$ where $\hat{x}_{\bar{k}}$ is the mean of the *prior* estimate of x_k, which is solely based on the process model of the state defined by system dynamics, before updating with observation y_k, $\hat{x}_{k-1}$ is the mean of the *posterior* state density at time *k–1* after it has been updated with observation y_{k-1}, $P_{\bar{k}}$ is *prior* estimate of covariance matrix of the state density at time k, solely based on the process model defined by system dynamics and before updating with observation y_k, P_{k-1} is covariance matrix of the *posterior* state density at time *k–1* after it has been updated with observation y_{k-1}, *A* is state transition matrix, *B* is the matrix mapping external inputs to system state variables x_k, *Q* is a process error. These equations are written for roller wear depth according to PoF model defined in step 1 above: $$\widehat{D}_{\bar{k}} = \widehat{D}_{k-1} + \dot{D} \cdot p_k$$ $$P_{\bar{k}} = P_{k-1} + Q$$ where $= 1,\ B = \dot{D}$, variables vector u_k represents the number of passes p_k accumulated since step *k-1*, *Q* is a process error (a scalar). These equations establish the 1st step of Bayesian fusion via Kalman filter, Model Based Prediction (Time Update). 4. General form of Kalman filter equations for the posterior

Elements of Agent Learning Process	Type II Macro-Agent **Roller Damage due to Wear (Wear Depth)**
	mean $\hat{x}_k$ and the posterior covariance P_k of the state variable x_k: $$\hat{x}_k = \hat{x}_{\bar{k}} + K_k(y_k - H\hat{x}_{\bar{k}})$$ $$P_k = (1 - K_k H)P_{\bar{k}}$$ $$K_k = P_{\bar{k}}H^T(HP_{\bar{k}}H^T + R)^{-1}$$ where H is the measurement matrix (i.e., the matrix mapping the observable evidence y_k to system state x_k), K_k is Kalman Gain and R is a measurement error. These equations take the following form for roller wear depth variable: $$\hat{D}_k = \hat{D}_{\bar{k}} + K_k\left(y_k - \hat{D}_{\bar{k}}\right)$$ $$P_k = (1 - K_k)P_{\bar{k}}$$ $$K_k = P_{\bar{k}}(P_{\bar{k}} + R)^{-1}$$ where R is a measurement error (a scalar), y_k is roller wear measurement at time step k. It is assumed that the measurement y_k is the exact same scale as system state estimate x_k, i.e. $H = 1$. These equations establish the 2nd step of Bayesian Fusion via Kalman filter, Data Driven Correction (Measurement Update). 5. The initial conditions for the system state variable (roller wear depth) $\hat{x}_0$ and P_0 are assigned with zero mean and high variance as it is completely unknown. Initialization with more meaningful starting values results in faster convergence. For example, wear rate of the roller material is 1×10^8 μm^3/km. Roller size is: 30 mm outside diameter, length 30 mm. Roller pass is 50 cm. Wear rate $\dot{D}$ becomes 7.1E–5 μm/pass. Initial conditions prior to test start (at $p_0 = 0$) are assigned as: $$\hat{D}_0 = 0\ \mu m$$ $$P_0 = 1E - 6\ \mu m^2$$
Parameters of Agent Output Model - Past Beliefs	1. State transition parameter, A: $A = 1$ 2. External inputs to system state conversion parameter, B: $B = \dot{D} = 7.1E - 5\ \mu m/pass$ 3. Measurement to state variable conversion parameter, H: $H = 1$ 4. Process error, Q: $Q = 1E - 12\ \mu m^2$ 5. Measurement error, R: $R = 1E - 8\ \mu m^2$

<table>
<tr><th rowspan="2">Elements of Agent Learning Process</th><th>Type II Macro-Agent</th></tr>
<tr><th>Roller Damage due to Wear (Wear Depth)</th></tr>
<tr><td>New Data (1st Data Set)</td><td>The new measurement of roller wear depth y_1 is obtained during the test after p_1 roller passes.</td></tr>
<tr><td>Model of Agent Output Variable
- Updated Beliefs 1st Update</td><td>The 1st update uses the following equations below:
1. Time Update:
$\widehat{D}_{\bar{1}} = \widehat{D}_0 + \dot{D} \cdot p_1$
$P_{\bar{1}} = P_0 + Q$
2. Measurement Update:
$\widehat{D}_1 = \widehat{D}_{\bar{1}} + K_1\left(y_1 - \widehat{D}_{\bar{1}}\right)$
$P_1 = (1 - K_1)P_{\bar{1}}$
$K_1 = P_{\bar{1}}(P_{\bar{1}} + R)^{-1}$</td></tr>
<tr><td>Parameters of Agent Output Model
- Updated Beliefs 1st Update</td><td>Same as “Parameters of Agent Output Model – Past Beliefs”</td></tr>
<tr><td>New Evidence (2nd Data Set)</td><td>New measurement of roller wear depth y_2 is obtained during the test after p_2 roller passes. Total number of passes since test start is $p = p_1 + p_2$.</td></tr>
<tr><td>Model of Agent Output Variable
- Updated Beliefs, 2nd Update</td><td>The 2nd update uses the following equations below:
1. Time Update:
$\widehat{D}_{\bar{2}} = \widehat{D}_1 + \dot{D} \cdot p_2$
$P_{\bar{2}} = P_1 + Q$
2. Measurement Update:
$\widehat{D}_2 = \widehat{D}_{\bar{2}} + K_2\left(y_2 - \widehat{D}_{\bar{2}}\right)$
$P_2 = (1 - K_2)P_{\bar{2}}$
$K_2 = P_{\bar{2}}(P_{\bar{2}} + R)^{-1}$</td></tr>
<tr><td>Parameters of Agent Output Model
- Updated Beliefs, 2nd Update</td><td>Same as “Parameters of Agent Output Model – Updated Beliefs 1st Update”</td></tr>
</table>

5.8.2. Discrete-Time Extended Kalman Filter

If the first of the three limiting assumptions of the discrete-time Kalman filter formulation, discussed in Section 5.8.1, does not hold (i.e. the state transition and observation models are not linear functions of the state variable), the discrete-time extended Kalman filter (EKF) is used as the nonlinear version of the discrete-time Kalman filter [101]. Discrete-time EKF utilizes the first-order Taylor series approximation (around its current state) instead of the linear transition and observation functions.

An example of Type II Macro-Agent learning by means of the discrete-time EKF is given in Table 5-10. This example is based on the fatigue crack propagation study described in [99]. The two agents, crack growth rate and crack size, are learning simultaneously via Bayesian fusion by discrete-time EKF.

Table 5-10: Example of Type II Macro-Agent Learning by Extended Kalman Filter

Elements of Agent Learning Process	**Type II Macro-Agent** **Fatigue Crack Size** **Crack Growth Rate**
Input Variables from Input Agents	Type I Micro-Agents: Loading cycles elapsed since previous time step k, ΔN_k Type II Macro-Agents: Stress intensity factor range at time step k, ΔK_k Notes: *1. Crack Size agent and Crack Growth Rate agent are time dependent via Elapsed Loading Cycles Agent* ΔN_k. *2. All input agents have "Inactive" status (i.e. constant values are assigned to their output variables).*
Output Variable	Size of Fatigue Crack in Metallic Structure, a Crack Growth Rate, $\dot{a}$

Elements of Agent Learning Process	Type II Macro-Agent **Fatigue Crack Size** **Crack Growth Rate**
Model of Agent Output Variable - Past Beliefs	1. The PoF model of crack growth is defined as $$a = \sum_k \Delta a_k$$ $$\Delta a_k = \dot{a}_{k-1} \Delta N_k$$ $$N = \sum_k \Delta N_k$$ where a is crack size at accumulated load cycles N, Δa_k is the amount of crack extension at time step k ($k = 1, 2, \ldots$), ΔN_k is the number of elapsed loading cycles since time step $k{-}1$, and $\dot{a}_{k-1}$ is crack growth rate at time step $k{-}1$. Paris equation is used to relate the crack growth rate $\dot{a}_k$ to the stress intensity factor range ΔK_k at every time step k: $$\dot{a}_k = C(\Delta K_k)^m$$ where ΔK_k is stress intensity range at time step k, C and m are constants that depend on material properties and a set of test conditions, such as loading ratio, frequency and environment. The stress intensity factor range at every time step k, ΔK_k, is defined as a function of the geometry of the structure, applied loading cycle and the crack size a_k. For example, ΔK_k for a standard specimen (per ASTM E647-08 2008) is defined as follows: $$\Delta K_k = \frac{\Delta P_k}{B\sqrt{W}} \frac{(2+\alpha_k)}{(1-\alpha_k)^{3/2}} (0.886 + 4.64\alpha_k - 13.32{\alpha_k}^2 + 14.72{\alpha_k}^3 - 5.6{\alpha_k}^4)$$ where α_k is the dimensionless crack size a_k/W, B and W are the thickness and the width of the specimen, respectively, and ΔP_k is the amplitude range of applied load at time step k. 2. According to the EKF *process model* equation, the matrix of system state variables is specified in general form as: $$x_k = f(x_{k-1}, u_k, w_k)$$ where x_k is the vector of system state variables at time step k, x_{k-1} is the vector of system state variables at time step $k{-}1$, u_k is a vector that contains all external inputs, w_k is the ‘noise’

Elements of Agent Learning Process	Type II Macro-Agent Fatigue Crack Size Crack Growth Rate
	vector that is normally distributed with zero mean and covariance Q_k: $w_k \sim Normal\ (0, Q_k)$, where Q_k is process variance matrix (i.e., process model error). Crack size a_k and crack growth rate $\dot{a}_k$ are considered system state variables: $$x_k = \begin{pmatrix} a_k \\ \dot{a}_k \end{pmatrix}$$ External input u_k is used to map the time steps of the state-space model to the elapsed loading cycles, ΔN_k, and stress intensity factor range at time step k, ΔK_k. The process error is defined as: $$w_k \sim (0, Q_k)$$ where Q_k is a 2×2 covariance matrix Q at time step k. Considering PoF model described in step 1 above, the system state function is defined as follows: $$f(x_{k-1}, u_k, w_k) = \begin{pmatrix} a_{k-1} + \dot{a}_{k-1}\Delta N_k \\ C(\Delta K_k)^m \end{pmatrix} + w_k$$ where ΔK_k is a function of the crack size a_k at time step k, $a_k = a_{k-1} + \dot{a}_{k-1}\Delta N_k$. Crack growth *process model* equation becomes: $$\begin{pmatrix} a_k \\ \dot{a}_k \end{pmatrix} = \begin{pmatrix} a_{k-1} + \dot{a}_{k-1}\Delta N_k \\ C(\Delta K_k)^m \end{pmatrix} + w_k$$ 3. According to the EKF *observation model* equation, the vector of observable evidence is defined as $$y_k = h(x_k, \upsilon_k)$$ where y_k is a vector of the observable evidence, υ_k is the 'noise' vector that is normally distributed with zero mean and covariance R_k: $\upsilon_k \sim Normal\ (0, R_k)$, where R_k is measurement variance matrix (i.e., measurement error). An observation vector y_k includes two types of observations at time step k: the observation of crack size and the observation of crack growth rate. An observation vector is defined assuming a linear form for the observation function $h(\cdot)$:

Elements of Agent Learning Process	Type II Macro-Agent **Fatigue Crack Size** **Crack Growth Rate**
	$$y_k = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_k \\ \dot{a}_k \end{pmatrix} + \upsilon_k$$ The measurement error υ_k is defined as $$\upsilon_k \sim (0, R_k)$$ where R_k is a 2×2 covariance matrix R at time step k. 4. If there are times when only crack growth rate observation is available, but not crack size measurement, a rule could be programmed onto the agent's internal knowledge (specifically into special rules of agent behavior) to change an observation vector to the following when only one observation is available: $$y_k = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} a_k \\ \dot{a}_k \end{pmatrix} + \upsilon_k$$ where covariance matrix R_k becomes a scalar representing variance of the observation noise. 5. Initial conditions for the system state variables (crack size and crack growth rate) $\hat{x}_0$ and P_0 are assigned based on the known distribution of initial crack size, $a_0 \sim N(\hat{a}_0, P_0)$, as follows: $$\hat{x}_0 = \begin{pmatrix} \hat{a}_0 \\ \hat{\dot{a}}_0 \end{pmatrix}$$ where $$\hat{\dot{a}}_0 = C\left(\widehat{\Delta K}_0\right)^m$$ and $\widehat{\Delta K}_0$ is evaluated for crack size $\hat{a}_0$.
Parameters of Agent Output Model - Past Beliefs	1. Process error matrix time step k, Q_k 2. Measurement error matrix time step k, R_k 3. Constants C and m (obtained from fatigue tests performed on standard components with similar material and in the same testing condition)
New Data	The new measurement of the crack size, a_1, and the corresponding measurement of the crack growth rate, $\dot{a}_1$,are obtained during the test after ΔN_1 roller passes: $$y_1 = \begin{pmatrix} a_1 \\ \dot{a}_1 \end{pmatrix}$$

<table>
<tr><th rowspan="2">Elements of Agent Learning Process</th><th>Type II Macro-Agent</th></tr>
<tr><th>Fatigue Crack Size
Crack Growth Rate</th></tr>
<tr><td>Model of Agent Output Variable
- Updated Beliefs</td><td>

1. The 1st step of Bayesian fusion via the EKF is Model Based Prediction (Time Update):

$$\hat{x}_{\bar{k}} = \begin{pmatrix} \hat{a}_{\bar{k}} \\ \hat{\dot{a}}_{\bar{k}} \end{pmatrix} = \begin{pmatrix} \hat{a}_{k-1} + \hat{\dot{a}}_{k-1}\Delta N_k \\ C\left(\widehat{\Delta K}_{\bar{k}}\right)^m \end{pmatrix}$$

$$P_{\bar{k}} = A_k P_{k-1} A_k^T + W_k Q_k W_k^T$$

where

A_k and W_k are the partial derivative matrices:

$$A_k = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1}}$$

$$W_k = \left.\frac{\partial f}{\partial w}\right|_{\hat{x}_{k-1}}$$

$\hat{x}_{\bar{k}}$ is the mean of the *prior* estimate of x_k, which is solely based on the process model of the state defined by system dynamics, before updating with observation y_k,

$\hat{x}_{k-1}$ is the mean of the *posterior* state density at time *k–1* after it has been updated with observation y_{k-1},

$P_{\bar{k}}$ is *prior* estimate of covariance matrix of the state density at time k, solely based on the process model defined by system dynamics and before updating with observation y_k,

P_{k-1} is covariance matrix of the *posterior* state density at time *k–1* after it has been updated with observation y_{k-1},

$\widehat{\Delta K}_{\bar{k}}$ is evaluated for the crack size $\hat{a}_{\bar{k}}$, where $\hat{a}_{\bar{k}} = \hat{a}_{k-1} + \hat{\dot{a}}_{k-1}\Delta N_k$.

For the first update, using initial conditions, the above equations take a form:

$$\hat{x}_{\bar{1}} = \begin{pmatrix} \hat{a}_{\bar{1}} \\ \hat{\dot{a}}_{\bar{1}} \end{pmatrix} = \begin{pmatrix} \hat{a}_0 + \hat{\dot{a}}_0\Delta N_1 \\ C\left(\widehat{\Delta K}_{\bar{1}}\right)^m \end{pmatrix}$$

$$P_{\bar{1}} = A_1 P_0 A_1^T + W_1 Q_1 W_1^T$$

2. The 2nd step of Bayesian fusion via the EKF, Data Driven Correction (Measurement Update), is performed to obtain the posterior mean $\hat{x}_k$ and the posterior covariance P_k of the state variable $x_k = \begin{pmatrix} a_k \\ \dot{a}_k \end{pmatrix}$:

</td></tr>
</table>

Elements of Agent Learning Process	**Type II Macro-Agent**
	Fatigue Crack Size **Crack Growth Rate**
	$\hat{x}_k = \hat{x}_{\bar{k}} + K_k[y_k - h(\hat{x}_{\bar{k}}, 0)]$ $h(\hat{x}_{\bar{k}}, 0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{a}_{\bar{k}} \\ \hat{\dot{a}}_{\bar{k}} \end{pmatrix}$ $P_k = (1 - K_k H_k) P_{\bar{k}}$ $K_k = P_{\bar{k}} H_k^T (H_k P_{\bar{k}} H_k^T + M_k R_k M_k^T)^{-1}$ where y_k is the vector of crack size and crack growth rate measurements at time step k, K_k is Kalman Gain, H_k and M_k are the partial derivative matrices: $H_k = \left.\frac{\partial h}{\partial x}\right\|_{\hat{x}_{\bar{k}}}$ $M_k = \left.\frac{\partial h}{\partial v}\right\|_{\hat{x}_{\bar{k}}}$ For the first update using the new data, the above equations take a form: $\hat{x}_1 = \hat{x}_{\bar{1}} + K_1[y_1 - h(\hat{x}_{\bar{1}}, 0)]$ $P_1 = (1 - K_1 H_1) P_{\bar{1}}$ $K_1 = P_{\bar{1}} H_1^T (H_1 P_{\bar{1}} H_1^T + M_1 R_1 M_1^T)^{-1}$ where $y_1 = \begin{pmatrix} a_1 \\ \dot{a}_1 \end{pmatrix}$ $h(\hat{x}_{\bar{1}}, 0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{a}_{\bar{1}} \\ \hat{\dot{a}}_{\bar{1}} \end{pmatrix}$
Parameters of Agent Output Model - Updated Beliefs	Same as "Parameters of Agent Output Model – Past Beliefs"

5.8.3. Other Nonlinear Approaches

More refined linearization techniques than discrete-time EKF (Section 5.8.2) can be used to reduce the linearization error in the discrete-time EKF for highly nonlinear systems. They include iterated EKF (IEKF) and second-order discrete-time EKF (SOEKF) [101], both recommended as agent learning methods.

Other approaches also exist to handle nonlinear systems. These techniques typically provide better estimation performance for highly nonlinear systems, but do so at the price of higher complexity and significant computational effort. The most common methods of nonlinear filtering are Gaussian sum filters, grid filters and unscented Kalman filter (UKF) [101]. Another method, called the particle filter, is a simulation-based estimation technique that models the probability density function of state variables using a set of discrete points called particles. No assumptions (such as linearity or normally distributed uncertainty) need to be made. One approach that has been proposed for improving particle filtering is to combine it with another filter such as the EKF or the UKF. Due to the high computational intensity, the combined approaches may not be practical for the agent learning application, at least until further advancements in computational sciences occur.

For a system that is nonlinear and/or has non-Gaussian noise, the non-linear Kalman filters can be used for state estimation, although the particle filter may give more accurate estimates at the price of higher computational expense. The unscented Kalman filter provides a good balance between the moderate computational effort of the non-linear Kalman filters and the high performance of the particle filter.

5.9. Summary of Learning Methods for Type I and Type II Agents

This chapter introduced the learning methods of Type I Micro-Agents and Type II Macro-Agents, specifically those methods which are used to define the model of the agent output variable, if the model is not available, and update the parameters of the identified or already known model. The guidelines for applicability of these learning methods are summarized in Table 5-11. This summary suggests that Bayesian methods of agent learning (Bayesian inference and Bayesian fusion) are superior to other methods because of their ability to incorporate physics-of-failure into the agent representation model as well as to recursively update the existing estimates of model parameters with new information from various sources. This recursive (sequential) update of agent's beliefs using all available data maximizes the system reliability knowledge and minimizes computation time. The ability to incorporate physics-of-failure into agent modeling was highlighted in the previous chapters as a key aspect of agent structure and evolution. Bayesian inference and Bayesian fusion, therefore, are considered the primary methods of agent learning within the scope of intelligent agent autonomy.

Bayesian fusion is the most comprehensive method of agent learning in situations when incoming uncertain observations from in-field health monitoring or life testing are available in addition to a mathematical model that describes the evolution of the variables over time. Where two or more learning methods are applicable for an agent, Bayesian methods of agent learning (i.e. Bayesian inference or Bayesian fusion) should be preferred due to the efficient data updating scheme.

It is known, however, that the physical model of failure may not be available for some agents due to the lack of engineering knowledge, complexity of failure process, or a nature of some random variables external to the system under study (such as environmental factors, for instance). Empirical models of the agent output variable can be used within the agent learning scheme in lieu of the PoF relationship. If there is no model available for the agent output variable, several learning methods may apply, such as parametric distribution analysis (classical), time series and trend analysis methods, machine learning and pattern recognition methods, or Bayesian inference.

Table 5-11: Learning Method Selection Criteria for Type I and Type II Agents

Agent Learning Method	Type of Agent Output Variable	Agent Output Model (PoF or Empirical)	Time Dependency of Agent Output Variable	Required Data Types	Data Aggregation within Agent Learning Process	Examples
Parametric Distribution Analysis (classical)	Independent Output Variable Y, continuous or discrete	Model is not available	Not time dependent (no time trends in output Y exist)	Direct measurements of the quantity Y, such as environmental or operational characteristic, time to failure or time to success, other independent variable. Typical data sources: inspection, maintenance records, generic data.	New evidence is combined with all past observations into one data set for re-estimation of parametric distribution parameters	Probability distributions: - Normal - Lognormal - Weibull
Model-based Parametric Distribution Analysis (Cumulative Damage)	Output Variable $Y(t)$, continuous, is a function of Input Variables t_i ($i = 1, ..., n$), none of which is time variable	Model is available, the model is probabilistic, linear or nonlinear	Not time dependent (no time trends in output $Y(t)$ exist)	Direct measurements of the quantity $Y(t)$ and input quantities t_i ($i = 1, ..., n$), where $Y(t)$ is time to failure or time to success. Typical data sources: inspection, maintenance records.	New evidence is combined with all past observations into one data set for re-estimation of model-based parametric distribution parameters	Cumulative Damage models: - Eyring-Weibull - IPL-Exponential - Arrhenius-Lognormal

Agent Learning Method	Type of Agent Output Variable	Agent Output Model (PoF or Empirical)	Time Dependency of Agent Output Variable	Required Data Types	Data Aggregation within Agent Learning Process	Examples
Time Series and Trend Analysis for Degradation and other Trend Modeling	- Independent Output Variable *Y*, continuous or discrete - Output Variable *Y(t)*, continuous or discrete, is a function of Input Variables, t_i ($i = 1, \ldots, n$), one of which could be time variable	Model is not available	- Time dependent (time trends in *Y(t)* exist), monotonic or non-monotonic, slow change over time - Not time dependent (no time trends in *Y(t)* exist)	Direct measurements of the quantity *Y(t)* and input quantities t_i ($i = 1, \ldots, n$), where *Y(t)* is degradation measurement (e.g. crack size), environmental or operational characteristic, other variable. Typical data sources: inspection, maintenance records, sensors, monitors.	New evidence is combined with all past observations into one data set for re-estimation of time series and trend equation parameters	Time Series and Trend Analysis methods: - Regression - Sinusoidal - Exponential - Logarithmic - Decomposition
Machine Learning and Pattern Recognition Methods (excluding Bayesian Fusion methods)	- Independent Output Variable *Y*, continuous or discrete - Output Variable *Y(t)*, continuous or discrete, is a function of Input Variables, t_i ($i = 1, \ldots, n$), one of which could be time variable	Model is not available	- Time dependent (time trends in *Y(t)* exist), monotonic or non-monotonic, slow change over time - Not time dependent (no time trends in *Y(t)* exist)	Direct measurements of the quantity *Y(t)* and input quantities t_i ($i = 1, \ldots, n$), where *Y(t)* is degradation measurement (e.g. crack size), environmental or operational characteristic, other variable. Typical data sources: sensors, monitors (PHM applications).	New data are combined with all past data into one data set for re-estimation of machine learning or pattern recognition model parameters	Machine Learning methods: - Particle Filter - Artificial Neural Networks - Gaussian Process Regression

Agent Learning Method	Type of Agent Output Variable	Agent Output Model (PoF or Empirical)	Time Dependency of Agent Output Variable	Required Data Types	Data Aggregation within Agent Learning Process	Examples
Bayesian Inference	- Independent Output Variable *Y*, continuous or discrete - Output Variable $Y(t_i)$, continuous or discrete, is a function of Input Variables, t_i ($i = 1, \ldots, n$), one of which could be time variable	- Model is not available - Model is available, the model is probabilistic, linear or nonlinear	- Time dependent (time trends in $Y(t)$ exist), monotonic or non-monotonic, slow or rapid change over time - Not time dependent (no time trends in $Y(t)$ exist)	Direct measurements of the quantity $Y(t)$ and input quantities t_i ($i = 1, \ldots, n$), where $Y(t)$ is degradation measurement (e.g. crack size), time to failure or time to success, environmental or operational characteristic, other variable. Typical data sources: inspection, maintenance records, sensors, monitors, expert opinion, generic data.	New data are used to update the existing estimates of model parameters obtained in the past using previously available data	Applied to: - Parametric Distributions - Model-based Parametric Distributions (Cumulative Damage models) - Time Series and Trend models - Machine Learning models (e.g. Gaussian Process Regression)

Agent Learning Method	Type of Agent Output Variable	Agent Output Model (PoF or Empirical)	Time Dependency of Agent Output Variable	Required Data Types	Data Aggregation within Agent Learning Process	Examples
Bayesian Fusion	Output Variable $Y(t_i)$, continuous or discrete, is a function of Input Variables, t_i ($i = 1, ..., n$), one of which could be time variable	Model is available, the model is probabilistic or deterministic, linear or nonlinear	- Time dependent (time trends in $Y(t)$ exist), monotonic or non-monotonic, slow change over time - Not time dependent (no time trends in $Y(t)$ exist)	Direct measurements of the quantity $Y(t)$ and input quantities t_i ($i = 1, ..., n$), where $Y(t)$ is degradation measurement (e.g. crack size), environmental or operational characteristic, other dependent variable. Typical data sources: sensors, monitors (PHM applications).	Model based prediction and new data are combined to recursively update current estimates of Bayesian fusion model parameters obtained in the past using previously available data and model based prediction	Fusion methods (discrete time): - Kalman Filter (KF) - Extended Kalman Filter (EKF) - Unscented Kalman Filter (UKF) - Iterated EKF (IEKF) - Second-Order EKF (SOEKF) - Gaussian Sum Filters - Grid Filters

Chapter 6: Learning Property of Type III Agents

Type III Monitoring Agents learn and update their status by aggregating information from the input agents (Type II Macro-Agents and other Type III Monitoring Agents) according to the failure logic of the system elements they represent. The failure logic of an item represented by the Type III Monitoring Agent (hardware part, component or the entire system) could be given by:

1. A fault tree or an event tree [3]
2. Bayesian belief network (BBN) [4], [5]
3. Any mathematical expression $f(X_i, \theta)$ representing the Type III agent output variable $Y(X_i)$ as a function of n input variables provided by the input agents, X_i ($i = 1, \ldots, n$), with deterministically or probabilistically defined vector of model parameters $\theta = \{\beta_0, \beta_1, \ldots, \beta_k\}$:

$$Y(X_i) = f(X_i, \theta)$$

Equation 6-1

The learning process of Type III Monitoring Agents with deterministic model of the agent output variable, a function f(·), involves: 1) updating the functional form of f(·) using data, if applicable, and 2) updating the probability distribution of the agent output variable, Y, by means of simulation over the (updated) model of the agent output variable given the probability distributions of the input variables, X_i ($i = 1, \ldots, n$), according to the latest beliefs of the respective input agents.

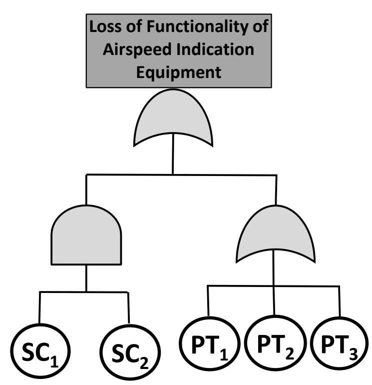
The learning process of Type III Monitoring Agents with probabilistic model of the agent output variable, a function f(·), involves: 1) updating the functional form of f(·) using data, if applicable, 2) updating the agent output model parameters, $\theta = \{\beta_0, \beta_1, \ldots, \beta_k\}$, using data, and 3) updating the probability distribution of the agent output variable, Y, by means of simulation over the updated model of the agent output variable given the probability distributions of the input variables, X_i ($i = 1, \ldots, n$), according to the latest beliefs of the respective input agents.

In many cases the output variable of the Type III Monitoring Agent is given by a deterministic function of the input variables, i.e. the agent output model parameters are fully defined constant values which do not get updated during agent learning. In such cases there is no uncertainty associated with the failure logic equation that is developed strictly based on the rules of probability according to the definition of the item failure within the given application. The failure logic of an item represented by the Type III Monitoring Agent, however, may change over time (e.g. the probabilistic equation P(A) = P(B) x P(C) could change to P(A) = P(B) x P(C) x P(D), where P(A) stands for probability of the event A, and so on). The functional form of the deterministic model of the agent output variable would change over time if the change criteria are programmed onto the agent's internal knowledge (specifically, into the special rules of agent behavior).

In some cases the output variable of the Type III Monitoring Agent is given by probabilistic function of the input variables, which implies that the failure logic of an item or the contribution of some input events probability into the output event probability is uncertain. In such cases data are used to update the uncertain parameters of the model of Type III Monitoring Agent output variable.

Examples of the learning process of Type III Monitoring Agents with deterministic model of the agent output variable by means of fault tree and BBN are given in Table 6-1 and Table 6-2, respectively.

Table 6-1: Example of Type III Monitoring Agent Learning via Fault Tree Logic

Elements of Agent Learning Process	**Type III Monitoring Agent**
	Functionality of Airspeed Indication Equipment
Input Variables from Input Agents	Type II Macro-Agents: - Three Failure Mechanisms of Pitot Tube, PT_i, $i = 1, 2, 3$ - Two Failure Mechanisms of Static Circuits, SC_j, $j = 1, 2$ Failure Mechanism probabilities $f(PT_i)$, $i = 1, 2, 3$, and $f(SC_j)$, $j = 1, 2$, are given by probability distributions representing output variables of the respective Type II Macro Agents.
Output Variable	Loss of Functionality of Airspeed Indication Equipment, *ASI*
Model of Agent Output Variable - Past Beliefs	Probability of Airspeed Indication Equipment failure (loss of functionality), $P(ASI)$, is modeled by the Fault Tree:  Probability $f(ASI)$, is expressed through the probabilities of ASI Failure Mechanisms according to the Fault Tree logic, as follows: $$f(ASI) = f(SC_1) \cdot f(SC_2) + f(PT_1) + f(PT_2) + f(PT_3)$$ Probability distribution $f(ASI)$ is obtained from probability distributions probabilities $f(PT_i)$, $i = 1, 2, 3$, and $f(SC_j)$, $j = 1, 2$, via simulation.

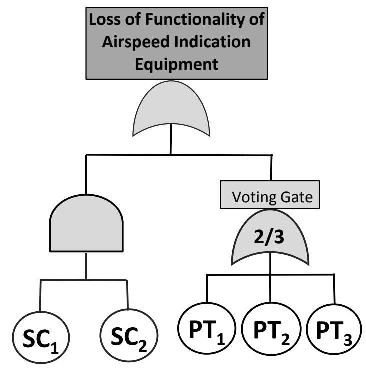
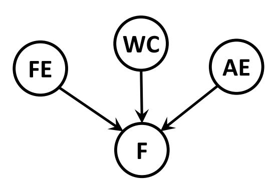
Elements of Agent Learning Process	**Type III Monitoring Agent** **Functionality of Airspeed Indication Equipment**
Parameters of Agent Output Model - Past Beliefs	All model parameters are equal one (unity).
New Data	No data are used to update agent output model parameters. New information, however, may become available to update the Fault Tree structure according to the criteria defined in the Special Rules of agent behavior (such as design changes or other reasons). For example, consider the following new structure: 
Model of Agent Output Variable - Updated Beliefs	Probability *f(ASI)* is updated to the following: $f(ASI) = f(SC_1) \cdot f(SC_2) + f(PT_1) \cdot f(PT_2) + f(PT_2) \cdot f(PT_3) + f(PT_1) \cdot f(PT_3)$ Probability distribution *f(ASI)* is obtained from probability distributions probabilities $f(PT_i)$, $i = 1, 2, 3$, and $f(SC_j)$, $j = 1, 2$, via simulation.
Parameters of Agent Output Model - Updated Beliefs	All model parameters are equal one (unity).

Table 6-2: Example of Type III Monitoring Agent Learning via Bayesian Belief Network

Elements of Agent Learning Process	**Type III Monitoring Agent**
	Pilot Error
Input Variables from Input Agents	Type I Micro-Agents: - Flight Experience, *FE (binary state: Sufficient, Low)* Type II Macro-Agents: - Weather Conditions, *WC (binary state: Normal, Severe)* Type III Monitoring Agents: - Functionality of Airspeed Indication Equipment, *AE (binary state: Normal, Partial Loss of Function)* Binary states are assumed for all nodes: $FE = \{Sufficient, Low\} = \{fe, \overline{fe}\}$ $WC = \{Normal, Severe\} = \{wc, \overline{wc}\}$ $AE = \{Normal, Partial\ Loss\ of\ Function\} = \{ae, \overline{ae}\}$ $F = \{Aircraft\ In\ Control, Aerodynamic\ Stall\} = \{f, \bar{f}\}$ Probability distributions of the input variables *FE*, *WC* and *AE*: $P(fe) = \mathrm{p}_{fe}, P(\overline{fe}) = 1 - \mathrm{p}_{fe}$ $P(wc) = \mathrm{p}_{wc}, P(\overline{wc}) = 1 - \mathrm{p}_{wc}$ $P(ae) = \mathrm{p}_{ae}, P(\overline{ae}) = 1 - \mathrm{p}_{ae}$ Probability p_{fe} is known fixed value (the output of Type I Micro Agent Flight Experience). Probability p_{wc} is a distribution obtained based on physical model and weather data for the flight region (the output of Type II Macro Agent Regional Weather). Probability p_{ae} is a distribution obtained by simulation via Fault Tree logic of functional failure condition for the Airspeed Indication Equipment (the output of Type III Equipment Monitoring Agent described in Table 6-1).
Output Variable	Pilot Error resulting in Aerodynamic Stall, *F (binary state: Aircraft in Control, Failure to Avoid Aerodynamic Stall)*
Model of Agent Output Variable - Past Beliefs	Probability of Pilot Error is modeled by the following BBN: 

<table>
<tr><th>Elements of Agent Learning Process</th><th>Type III Monitoring Agent
Pilot Error</th></tr>
<tr><td></td><td>

Probability distribution of the output variable F, $P(F)$, is defined by the unconditional probability of F computed as a sum of conditional probabilities of the states of F given the states of the input variables FE, WC and AE:

Node FE	Node WC	Node AE	Probability of Combination, p_i	$Pc(F=f)_i$	$Puc(F=f)_i$
fe	wc	ae	$p_1 = p_{fe}p_{wc}p_{ae}$	r_1	$P_1 = p_1r_1$
fe	wc	$\overline{ae}$	$p_2 = p_{fe}p_{wc}(1-p_{ae})$	r_2	$P_2 = p_2r_2$
fe	$\overline{wc}$	ae	$p_3 = p_{fe}(1-p_{wc})p_{ae}$	r_3	$P_3 = p_3r_3$
fe	$\overline{wc}$	$\overline{ae}$	$p_4 = p_{fe}(1-p_{wc})(1-p_{ae})$	r_4	$P_4 = p_4r_4$
$\overline{fe}$	wc	ae	$p_5 = (1-p_{fe})p_{wc}p_{ae}$	r_5	$P_5 = p_5r_5$
$\overline{fe}$	wc	$\overline{ae}$	$p_6 = (1-p_{fe})p_{wc}(1-p_{ae})$	r_6	$P_6 = p_6r_6$
$\overline{fe}$	$\overline{wc}$	ae	$p_7 = (1-p_{fe})(1-p_{wc})p_{ae}$	r_7	$P_7 = p_7r_7$
$\overline{fe}$	$\overline{wc}$	$\overline{ae}$	$p_8 = (1-p_{fe})(1-p_{wc})(1-p_{ae})$	r_8	$P_8 = p_8r_8$

where $r_i=Pc(F=f)_i$ is conditional probability of F being in state f given the i-th combination of the states of three input variables, and $P_i=Puc(F=f)_i$ is the respective unconditional probability. The resulting probability of the agent output variable F being in state f is defined by the following sum:

$$P(F = f) = \sum_{i=1}^{8} P_i = \sum_{i=1}^{8} p_i r_i$$

Note, that conditional probabilities r_i do not sum up to unity in general case:

$$\sum_{i=1}^{8} r_i \neq 1$$

</td></tr>
<tr><td>Parameters of Agent Output Model
- Past Beliefs</td><td>The model parameters are conditional probabilities of F being in state f given the i-th combination of the states of the input variables: $\theta = (\{r_i\}, i=1,\ldots, 8)$. These conditional probabilities are known constant values.</td></tr>
</table>

Elements of Agent Learning Process	Type III Monitoring Agent
	Pilot Error
New Data	No data are used to update agent output model parameters. New information, however, may become available about the degree of influence of various states of the input variables *FE*, *WC* and *AE* on the states of the output variable *F*. This information would be used to update the values of conditional probabilities of *F* being in state *f* given *i*-th combination of the states of the input variables, r_i, according to the criteria defined in the Special Rules of agent behavior (such as design changes, different mission profile, or other reasons).
Model of Agent Output Variable - Updated Beliefs	Same as "Model of Agent Output Variable - Past Beliefs"
Parameters of Agent Output Model - Updated Beliefs	New information are utilized to obtain updated values of conditional probabilities of *F* being in state *f* given *i*-th combination of the states of the input variables, r_i. The new values of r_i, $i=1,\ldots, 8$, are used as agent output model parameters instead of the past values.

Figure 6-1 provides an example of agent autonomy in order to demonstrate the difference in the learning property of the three types of agents:

- Type I Micro-Agent cyclic load has an independent output variable, mechanical load amplitude, defined probabilistically as a lognormal probability distribution LN(μ,σ). As a new set of field data (for use load amplitude) becomes available, parameters of probability distribution of the mechanical load amplitude are updated through Bayesian formalism. This update is the only step of the learning process for the mechanical load amplitude agent that has no inputs from other agents.
- Type II Macro-Agent has the output variable, fatigue life of structural component, as a function of the mechanical load amplitude. This agent's 1st step of learning is

updating the parameters of the PoF model of fatigue life to failure (Equation 3-2 in Section 3.2.2) by Bayesian inference using the available test data. This Type II Macro-Agent has Type I Micro-Agent cyclic stress (mechanical load amplitude) as an input agent. In the 2^{nd} step of learning, the probability distribution of fatigue life to failure is obtained by simulation over the PoF model with the updated parameters and using the latest (updated) probability distribution of the mechanical load amplitude.

- The Type III Monitoring Agent has the output variable, remaining life to failure (called remaining useful life, or RUL), quantified by simulation according to the RUL equation (as life to failure less total accumulated load cycles, *N*) based on the input from the Type II Macro-Agent fatigue life.

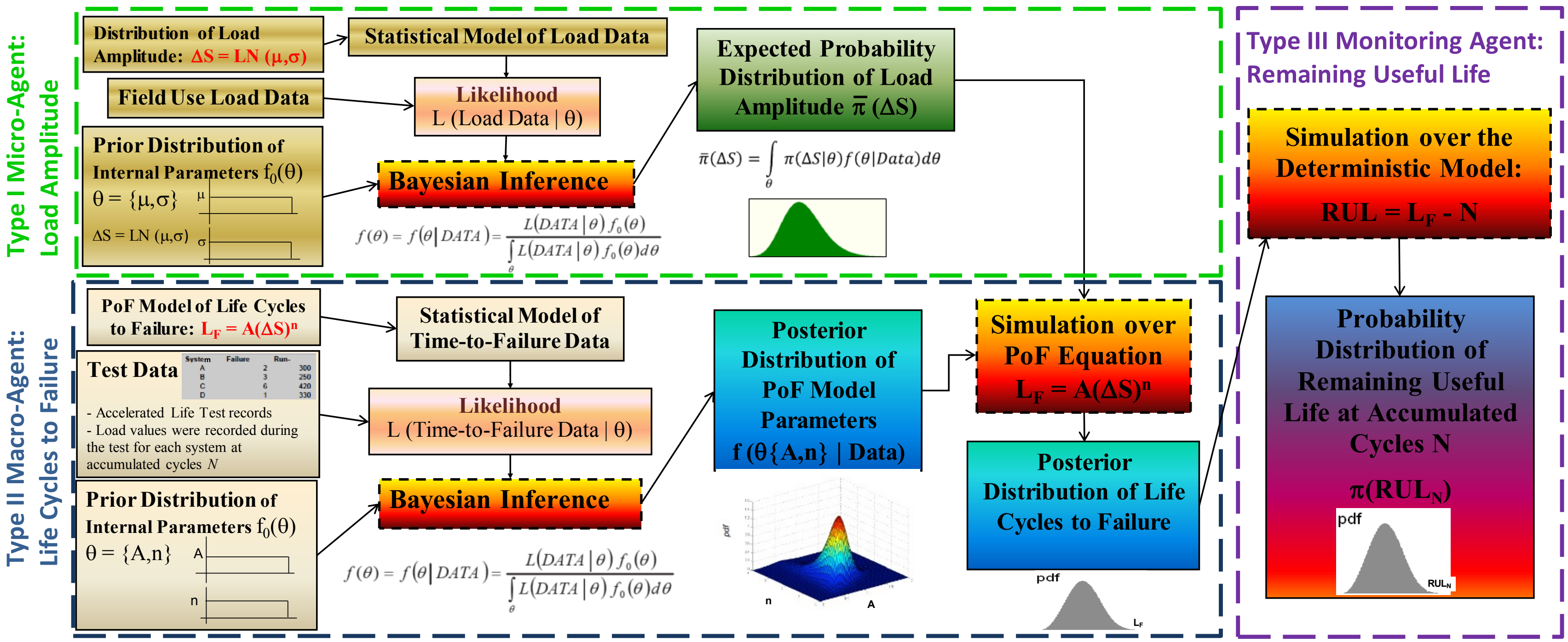


Figure 6-1: Agent Learning Demonstration Example

Chapter 7: Autonomy Property of Agents

The ability of intelligent agents to activate and deactivate themselves during system evolution is what makes an agent autonomous. Autonomy means that an agent is not only capable to act without supervision by recursive learning, but also has a degree of control over its participation in system evolution, specifically by changing its status between “active” and “inactive”. An agent’s ability to activate and deactivate itself is achieved by means of uncertainty importance analysis performed upon the agent output variable after each learning cycle.

7.1. Uncertainty Importance Measures for Autonomy Execution

Uncertainty importance measures are intended to identify the contribution of the uncertainty of input variables to that of the output response [102]. Based on the rankings of the input variables, one can give more priority to the variables with high importance, and neglect the variables with low importance, depending on the objective of the study. Uncertainty importance measures have been extensively introduced by sensitivity analysis [103] - [105].

By general definition, model sensitivity analysis determines the impact that changes in model inputs have on the model outputs. Model inputs include primarily model variables, but could also consider the initial conditions, boundary conditions, etc. Considering the n dimensional vector, X, as a vector of independent random variables, x_i $(i = 1, 2, \ldots, n)$, the output, $Y = f(X)$, is also a random variable as a function of n random variables. Uncertainty importance aims to determine the part of the total unconditional variance, *Var(Y),* of the output, *Y,* resulting from each input random variable, x_i. Within the scope of agent-oriented modeling, uncertainty importance analyzes whether or not sensible changes in the input variables (given by the input agents) would induce noticeable changes in the agent output variable.

Sensitivity approaches can be categorized into two main groups - local methods and global methods [106] - [112]. Local methods represent the simplest approach to sensitivity analysis (one-at-a-time analysis), where sensitivity measures are determined by varying only one parameter, while all others are held constant. The local sensitivity analysis methods have the advantage of being straightforward to implement while maintaining modest computational demands. The major drawback of these methods is their inability to account for input parameter interactions, making them prone to underestimating true model sensitivities. Alternatively, global parameter sensitivity analysis methods vary all of a model's inputs in predefined regions to quantify their importance and the importance of input parameter interactions. This is critical for the agent-oriented modeling of complex dynamic systems.

The choice of methods of sensitivity analysis for agent autonomy is dictated by the following considerations:

1. *Nonlinearity in output variables, non-monotonic output variables*

   The local sensitivity analysis methods cannot provide accurate sensitivity measures when the model response is nonlinear and/or non-monotonic with respect to its inputs. Variance-based global sensitivity measures should be used.

2. *Input parameter interactions*

    If interactions between the model inputs are present, varying two or more inputs simultaneously causes greater variation in the output than that upon varying each input alone. Such interactions are present in any non-additive model, but they are neglected by the local sensitivity methods that are based on one-at-a-time perturbations.

Based on the above considerations, the desirable properties of sensitivity analysis for agent autonomy execution are the following [112]:

1. *The ability to incorporate the effect of the input parameter mean, variation, and the form of its probability density function*

    It is important to consider the type of probability distribution of each input variable and the distribution parameters. The method should work regardless of the linearity and monotonicity of the model. The local sensitivity methods do not have these capabilities, and therefore global methods should be preferred.

2. *The ability to perform multivariate analysis*

    Local sensitivity methods consider the effect of the variation of one input parameter while all others are kept constant at the mean (or nominal) value. A global method should be used in order to evaluate the effect of one input parameter while all others are also varying. This would allow capturing the input parameter interactions, which are said to occur when the effect of changing two or more input variables is different from the sum of their individual effects.

From the above discussion, global sensitivity analysis methods are the most appropriate for modeling agent's activation and deactivation capability. Sobol's method of global sensitivity analysis is recommended because it is capable to deal with simultaneous variation in all qualitative and quantitative inputs, model nonlinearity, input interactions, and non-monotonic models, and it can also yield robust sensitivity rankings. To explain the essence of Sobol's method, let us denote a single output variable, *Y,* as a function of an input variable, *X*, or $Y = f(X)$, where *X* is defined by individual elements as $X = \{x_1, x_2, \ldots, x_n\}$. Sobol's method is based on variance decomposition where the variance, *V(Y),* of the output is a finite sum, and each term corresponds to the contribution of one input variable, x_i, or to the interaction of several input variables [108], [109], [112]. According to Sobol's method, two sensitivity indexes should be calculated: the first-order effect index for the variance of the input variable, x_i, and the total effects index for the variance of x_i. The first-order effect index of the variance of x_i is given by:

$$S_i = \frac{V_i}{V(Y)} = \frac{V_{x_i}\left(E_{x_{\sim i}}(Y|x_i)\right)}{V(Y)}$$

Equation 7-1

where *V(Y)* is the total variance of the output, *Y*, V_i is a contribution of variance of x_i into *V(Y)*, $E_{x_{\sim i}}(Y|x_i)$ denotes the expectation of the output, *Y,* by fixing the variable x_i at a particular value and considering random variations of all other variables (denoted by $x_{\sim i}$). Equation 7-1 calculates the variance of this expectation by considering the random variation in x_i and fixing it at random values. Monte Carlo simulation is used to calculate S_i by solving the respective multidimensional integral for V_i:

$$V_i = \int\limits_{\chi_i} \left[\int\limits_{\chi_{-i}} f(X) \prod_{\substack{k=1 \\ k \neq i}}^{n} p(x_k) \prod_{\substack{k=1 \\ k \neq i}}^{n} dx_k \right]^2 p(x_i) dx_i - E^2(Y)$$

Equation 7-2

where $p(x_k)$ is the probability density function of variable x_i, $E(Y)$ denotes the expected value of the output variable, Y, considering random variations of all the elements of the input variable, $X = \{x_1, x_2, \ldots, x_n\}$, χ_i is the sample space for x_i, and χ_{-i} corresponds to the reduced sample space defined by:

$$\chi_{-i} = \prod_{\substack{k=1 \\ k \neq i}}^{n} \chi_k$$

Equation 7-3

The first-order effect calculated in Equation 7-1 gives an estimate of the contribution of the variable x_i to the uncertainty in the output, Y, without consideration of the effects of the other variables, $x_{\sim i}$, since their contribution is averaged. The contribution of the input variable x_i in combination with all other input variables is known as the total effects index and can be calculated as:

$$S_{Ti} = \frac{V_{Ti}}{V(Y)} = \frac{V(Y) - V_{x_{\sim i}}\left(E_{x_i}(Y|x_{\sim i})\right)}{V(Y)} = \frac{E_{x_{\sim i}}\left(V_{x_i}(Y|x_{\sim i})\right)}{V(Y)}$$

Equation 7-4

where $V_{x_i}(Y|x_{\sim i})$ denotes the variance of the output, Y, when all variables other than x_i (denoted by $x_{\sim i}$) are fixed at a particular quantity to calculate the variance by considering variation in x_i. Equation 7-4 calculates the expectation of this variance by considering the variation in all other quantities ($x_{\sim i}$). Monte Carlo simulation is used to calculate S_{Ti} by solving the respective multidimensional integral for V_{Ti}:

$$V_{Ti} = V(Y) - \left\{ \int_{\chi_{-i}} \left[\int_{\chi_i} f(X) p(x_i) dx_i \right]^2 \prod_{\substack{k=1 \\ k \neq i}}^{n} p(x_k) \prod_{\substack{k=1 \\ k \neq i}}^{n} dx_k - E^2(y) \right\}$$

Equation 7-5

In summary, the first-order effect, S_i, describes the contribution of a particular input variable, x_i, alone to the uncertainty in the output variable, whereas the total effect, S_{Ti}, describes the overall contribution of a particular input variable, x_i, to the uncertainty in the output variable, in combination with all other variables. The determination of S_i and S_{Ti} is a problem in the evaluation of multidimensional integrals. In practice, this evaluation is carried out with sampling-based methods (such as Monte Carlo sampling). The concept of Monte Carlo sampling for the evaluation of sensitivity indexes is discussed in [5], [112] and [113], one of the sampling algorithms is also shown in [108]. The Monte Carlo sampling procedure involves sampling the input variables from given probability distributions for the evaluation of the first order and total effect Sobol's sensitivity indices. The sensitivity indexes calculation method, proposed in [114] and discussed in [115], formulates Sobol's indices in terms of the Pearson correlation coefficients and subtracts the spurious correlations (correlations caused by finite sample size in an ideally uncorrelated data) for improved accuracy. Both approaches, direct Monte Carlo sampling and correlation coefficients method, could be used to calculate the sensitivity indexes as a measure of uncertainty importance in establishing the activation/deactivation property of intelligent agents.

While statistical (probabilistic) sensitivity methods are the most appropriate for modeling the activation/deactivation capability of intelligent agents, they are mathematically comprehensive and involve significant computational effort. Their use is inefficient for large models and their results, in some cases, are comparable to those obtained from simpler techniques of local sensitivity measures. The local methods of sensitivity analysis are suitable for agent autonomy modeling if there are no interactions between the input variables and the model of the output variable is linear. If any of the above two conditions are not met, the use of a local sensitivity analysis method becomes a compromise between the reduction in computational effort and partial loss of information.

For Type II Macro-Agents and Type III Monitoring Agents which have a large number of inputs from other agents, the number of simulations required for a global sensitivity analysis becomes too large to be practical. In such case, the local methods of sensitivity analysis could be suitable given no interactions between the input variables are anticipated (each input variable is statistically independent of any other input variable) and a nonlinear model of the output variable can be linearized by transformation. Where the latter cannot be achieved (no transform function can be defined), graphical sensitivity analysis method called a scatter plot could be used to help identify relationships between individual inputs and a model output. Each point on a scatter plot represents a pair of an input value and the corresponding output value generated by simulation. A relationship could be linear or nonlinear. If the relationship is close to linear for all input variables, the local methods of sensitivity analysis could be considered a suitable compromise when the agent output model is large. If the relationship is highly nonlinear, the use of local sensitivity analysis methods may need to be ruled out.

Where a local sensitivity analysis is intended, the local method called Importance Index (II) is recommended for modeling of agent activation and deactivation capability. This measure of importance provides an indication of each input parameter's contribution to the variability in the model output. The Importance Index uses random sampling techniques to evaluate the input parameter's fractional contribution to the amount of uncertainty in the model output when varying each input parameter, one-at-a-time, according to its probability density function while the effect of all other variables is averaged out. This sensitivity index is the ratio of the output variance, obtained by considering random variation in the input parameter of interest, x_i, and fixing all other parameters at their mean values, to the total variance of the model output, *Y,* upon random variation of all input parameters. The Importance Index is calculated as:

$$I_i = \frac{V_{x_i}(Y|x_{\sim i})}{V(Y)}$$

Equation 7-6

where $V_{x_i}(Y|x_{\sim i})$ denotes the variance of the output, *Y,* when all variables other than x_i (denoted by $x_{\sim i}$) are fixed at a particular quantity (usually the mean value) to calculate the variance of *Y* by considering variation in x_i, and *V(Y)* is the total variance of the output parameter $Y = f(X)$, considering random variations of all elements of the input variable $X = \{x_1, x_2, \ldots, x_n\}$.

7.2. Agent Status Update Process

As discussed in Chapter 4, Type I Micro-Agents, Type II Macro-Agents, and Type III Monitoring Agents (except the Type III System Monitoring Agent) have activation and deactivation properties. These properties stem from the fact that an influence of some physical characteristics and constituent parts of the system on a physical processes of degradation and failure may change over time. For example, the variation of environmental temperature in the avionics bay of an aircraft was considered a critical input to the thermal model of the electronic control unit (ECU) which provided inputs to the reliability model of the ECU. After several years in service, upon degradation of some heat dissipation measures inside the ECU, variation of the bay temperature does not have any significant effect on the thermal model results and, consequently, on the reliability model of time to failure of the ECU because of other significant contributors to the board temperature rise. Changes in relative importance of various attributes to system degradation over time imply that the sensitivity of the model output to the input variables also changes with time, therefore, it is necessary to calculate the sensitivity indices as a function of number of the cycles or other measure of time.

As discussed in Section 7.1, uncertainty importance (sensitivity) analysis calculates the effect of the variance of an input quantity on the variance of the output quantity for a generic model given by:

$$Y = f(X), \;\; X = \{x_1, x_2, \dots, x_n\}$$

Equation 7-7

where x_i, $i = 1, \ldots, n$, denote the input variables and Y is an output variable. In the context of agent autonomy, Y represents the output variable of the Type II Macro-Agent or the Type III Monitoring Agent, and x_i represents output variables of Type I Micro-Agents, Type II Macro-Agents or Type III Monitoring Agents serving as inputs to the respective Type II Macro-Agent or Type III Monitoring Agent. For example, the output of Type II Macro-Agent, Y, is the crack size at the end of a particular number of cycles, then the inputs, x_i, $i = 1, \ldots, n$, include all Type I Micro-Agents and Type II Macro-Agents that affect the crack size prediction based on the physical model of crack growth (e.g. such Type I Micro-Agents as initial crack size, loading parameters, geometry, material constant, etc.).

A rule restricting the activation/deactivation property of certain agents could be programmed into the agent's internal knowledge (particularly special rules of behavior) to the discretion of the modeler. For example, a rule may be added that restricts the activation/deactivation property of all agents within the agent hierarchy except for Type I Micro-Agents and Type II Macro-Agents that have no inputs from other agents. This would imply that only the independent variables can change their input from probabilistic to deterministic and vice versa, while all dependent variables in the system hierarchy (with one or more inputs) will remain active at all times. Other examples are setting a permanent "active" status for the Type II Macro-Agent which serves as an input to the Type I Micro-Agent, and setting a permanent "active" status for all Type III Part Monitoring Agents and Type III Component Monitoring Agents.

The modeler defines the activation/deactivation criteria for each agent that has the autonomy property. These criteria may have a form of relative importance rank of the output variable of an agent, x_i, compared to the output variables of other agents forming an input set $\{x_1, x_2, \dots, x_n\}$ for a certain higher level agent with the output variable, Y. Such rank would be assigned based on first order and/or total sensitivity indexes for all the input variables, $\{x_1, x_2, \dots, x_n\}$. In the above example of Type II Macro-Agent crack size, the input agents (initial crack size, loading parameters, geometry and material constant) will be ranked based on their sensitivity indexes and the agents with the lowest rank may be deactivated per activation/deactivation criteria set by the modeler. Considering these four input agents and using the Importance Index as sensitivity measure, the agent deactivation criteria could be set as, for example, Importance Index (II) < 0.10, meaning that the input agent is deactivated if not more that 10% of the crack size variance is explained by variation of this input agent.

"Inactive" status of an agent implies that the agent output variable, x_i , has no effect on the next level agent, Y, and so x_i can be fixed at any value over its uncertainty range, preferably at mean value, changing the representation of the respective agent from probabilistic to deterministic (see Sections 3.1 and 4.6).

If a certain agent serves as an input agent for two or more other agents, this input agent could have an "active" status with respect to one of the next level agents and an "inactive" status with respect to the others. For example, if a Type I Micro-Agent mechanical load is an input to two Type II Macro-Agents, fatigue crack size and wear depth, sensitivity analysis may render this Type I Micro-Agent as "active" for Type II Macro-Agent fatigue crack size and “inactive” for Type II Macro-Agent wear depth.

The learning property of an agent is not affected by the agent status. The agent learns with the same intensity and according to the same rules, defined in previous chapters, regardless of its status (active or inactive).

The agent activation/deactivation procedure can be summarized, as follows:

1. All agents are set as active at time zero when the agent hierarchy is developed based on PoF knowledge of the system and prior (initially available) data.
2. Upon the first update of the agents’ beliefs based on new data (through the agent learning process), sensitivity analysis is performed for each Type I Micro-Agent, Type II Macro-Agent and Type III Monitoring Agent (except Type III System Monitoring Agent) to evaluate the effect of the variance of the agent output variable on the variance of the output variables of other agents that use aforementioned agent as an input. Sensitivity indexes are evaluated only for probabilistic agents comprising inputs to the next level agent, while uncertain parameters of the output model of the latter are set at fixed values, preferably at their mean values.
3. If the effect of the variance of the output variable of a given input agent on the variance of the output variable of the respective agent exceeds a threshold identified by the modeler, the input agent’s status remains as “active”.
4. If the effect of the variance of the output variable of a given input agent on the variance of the output variable of the respective agent is below a threshold

identified by the modeler, the input agent's status changes to "inactive" (i.e. this input agent is deactivated).

5. Upon the next update of the agents' beliefs based on new data (through the agent learning process), sensitivity analysis is performed for each agent within the agents hierarchy (except Type III System Monitoring Agent), including inactive agents, to re-evaluate the effect of the variance of the agent output variable on the variance of the output variables of the respective agents that use aforementioned agent as an input.
6. If the effect of the variance of the output variable of a given input agent on the variance of the output variable of the respective agent exceeds a threshold identified by the modeler, the input agent's status remains as "active" if the agent is currently active, or changes to "active" if the agent is currently inactive (i.e. this input agent is reactivated).
7. If the effect of the variance of the output variable of a given input agent on the variance of the output variable of the respective agent is below a threshold identified by the modeler, the input agent's status remains as "inactive" if the agent is currently inactive, or changes to "inactive" if the agent is currently active (i.e. this input agent is deactivated).
8. Steps 5 to 7 are repeated every time the agents in the hierarchy complete their learning process upon the availability of the new data.

As an example of uncertainty importance for agent autonomy, consider a Type II Macro-Agent, life to failure of the electromechanical component. The PoF model of the agent output variable, life to failure, is given by the following life-stress equation:

$$L = A\,(\Delta T)^{-k} exp^{\left(\frac{B}{T}\right)} (V)^{-l} H^{-m} (\Delta P)^{-n}$$

Equation 7-8

where *A*, *B*, *k*, *l*, *m* and *n* are the model parameters, *T* is ambient temperature, *ΔT* is thermal cycling amplitude, *V* is applied voltage, *H* is ambient humidity, and *ΔP* is pressure load cycling amplitude. Random variables *T*, *ΔT*, *V*, *H* and *ΔP* are represented by the Type I Micro-Agents, which are the inputs to the Type II Macro-Agent, life to failure, *L*. Life test data were used to update the beliefs of the Type I Micro-Agents and the Type II Macro-Agent. Upon each update, uncertainty importance (sensitivity) analysis was performed for the random variables *T*, *ΔT*, *V*, *H* and *ΔP* to evaluate their "importance" for the dependent variable *L*. The Importance Index (II) method was used to assess the contribution of the uncertainty (variability) of the input variables, *T*, *ΔT*, *V*, *H* and *ΔP*, into the uncertainty (variability) of the output variable, *L*. All five Type I Micro-Agents were set to have an active status prior to the first update. The results of two updates are summarized in Table 7-1. The contribution of each input agent was ranked according to the normalized importance index. It can be seen that Type I Micro-Agents representing thermal cycling and humidity levels remained active upon the first update and deactivated themselves after the second update, while Type I Micro-Agent pressure cycling remained active at all times. The Type I Micro-Agents temperature and voltage deactivated themselves upon the first update and remained inactive upon the second update. It was concluded that pressure cycling is the most important contributor to the variability of the life to failure. Pressure cycling should be represented by a Type I Micro-Agent in a future study, while temperature and voltage may be set as constant values. In order to confirm the status of thermal cycling and humidity agents, the modeler may need to obtain additional data so that the respective agents could continue their learning process. Another observation is related to the chosen method of sensitivity analysis, the Importance Index. The fractional contribution of each input variable to the uncertainty in the model output was evaluated by varying each of five input parameters, one-at-a-time, while holding all other variables at their average values. Since the life to failure, L, is a nonlinear function of the input variables, T, ΔT, V, H and ΔP , the total effect of all five input variables was close to 50% after each update (as shown by the importance index for life to failure variable in Table 7-1). This suggests that global sensitivity analysis methods, specifically Sobol's method of variance decomposition, should be preferred to improve the accuracy of the analysis.

Table 7-1: Example of the Agent Status Update by Uncertainty Importance (Sensitivity) Analysis

	Type I Micro-Agent Output Variables					**Type II Macro-Agent Output Variable**
First Update						
Variable Name	**Temperature**	**Thermal Cycle**	**Voltage**	**Humidity**	**Pressure Load Cycle**	**Life to Failure**
Variable Letter ID	**T**	**ΔT**	**V**	**H**	**ΔP**	**L**
Distribution	Normal	Lognormal	Normal	Lognormal	Lognormal	Lognormal
Mean	4.89	59.06	55.01	0.56	5.44	61,192
Variance	454.74	536.47	37.54	0.03	5.72	4.837E+09
Model Parameter ID	**B**	**k**	**l**	**m**	**n**	**A**
Model Parameter (Mean)	100	1	2	1	2	1.00E+10
Importance Index	**0.000**	**0.117**	**0.011**	**0.069**	**0.261**	0.457
Importance Index Rank	**0**	**26**	**2**	**15**	**57**	100
Agent Status	**Inactive**	**Active**	**Inactive**	**Active**	**Active**	-
Second Update						
Variable Name	**Temperature**	**Thermal Cycle**	**Voltage**	**Humidity**	**Pressure Load Cycle**	**Life to Failure**
Random Variable ID	**T**	**ΔT**	**V**	**H**	**ΔP**	**L**
Distribution	Normal	Lognormal	Normal	Lognormal	Lognormal	Lognormal
Mean	5.12	56.34	55.08	0.56	5.81	64,627
Variance	462.19	148.62	60.83	0.01	11.57	1.063E+10
Model Parameter ID	**B**	**k**	**l**	**m**	**n**	**A**
Model Parameter (Mean)	500	2	1	1	2	1.50E+10
Importance Index	**0.002**	**0.024**	**0.003**	**0.003**	**0.547**	0.580
Importance Index Rank	**0**	**4**	**1**	**1**	**94**	100
Agent Status	**Inactive**	**Inactive**	**Inactive**	**Inactive**	**Active**	-

7.3. Key Features and Advantages of the Autonomy Property

Considering the agent status update procedure described in Section 7.2, the next step is to define the algorithm of the autonomy execution within agent-oriented reliability modeling. The following example demonstrates the process. Consider the agent Y_1 having an output variable y_1 as a function of the input variables x_{1i} (*i = 1, 2, ..., n*), where the variables x_{1i} represent the respective input agents X_{1i}:

$$y_1 = f(x_{11}, x_{12}, \dots, x_{1n})$$

Equation 7-9

Assume that the agent Y_1 provides an input to another agent Z with an output variable *z*, where the latter is a function of the variable y_1 and several other variables y_j (*j = 2, ..., m*) associated with the agents Y_j:

$$z = g(y_1, y_2, \dots, y_m)$$

Equation 7-10

The agents X_{ji} (*j = 1, 2, ..., m, i = 1, 2, ..., n*) are the input agents of the respective agents Y_j (*j = 1, 2, ..., m*). The autonomy execution algorithm is defined as the follows:

1. As new data/information becomes available, the agent Y_1 learns from it by updating the functional form and the parameters of the model function f(·).
2. The input agents X_{1i} (*i = 1, 2, ..., n*) also learn from the new data/information and update their beliefs about: a) the functional form and the parameters of the model of their output variables, x_{1i}, and b) the resulting probability distributions of their output variables, $\pi(x_{1i})$, given the respective input variables (as applicable).

3. The input agents X_{1i} (*i = 1, 2, ..., n*) update their status by means of uncertainty importance (sensitivity analysis) to evaluate the effect of the variance of the agent output variables, x_{1i}, on the variance of the output variable of the agent Y_1 (this process is described in Section 7.2).
4. The input agents X_{1i} (*i = 1, 2, ..., n*) communicate the updated believes about the probability distributions of their output variables, $\pi(x_{1i})$, and about their status with respect to other agents (active or inactive) to all agents within the agent hierarchy, including the agent Y_1.
5. The agent Y_1 reacts to the updated beliefs of the input agents X_{1i} (*i = 1, 2, ..., n*) by updating the distribution of its output variable, $\pi(y_1)$, by means of simulation over the updated model function, f(·), and using the updated probability distributions $\pi(x_{1i})$ of those and only those input variables x_{1i} that represent the agents X_{1i} with an active status with respect to the agent Y_1. Probability distributions $\pi(x_{1i})$ of the input variables x_{1i} associated with inactive agents X_{1i} with respect to the agent Y_1 are not used in the simulation of the probability distribution $\pi(y_1)$. For the purpose of this simulation, the output variables of the inactive input agents X_{1i} are assigned with constant values derived from their respective probability distributions, $\pi(x_{1i})$, such as distribution mean or median.
6. The agents Y_j (*j = 2, ..., m*) and their respective input agents X_{ji} (*j = 2, ..., m, i = 1, 2, ..., n*) learn and update their beliefs in the same manner as the agent Y_1 and its input agents X_{1i} (*i = 1, 2, ..., n*) in accordance to steps 1 to 5 above.

7. The agent Z learns from the new data/information by updating the functional form and the parameters of the model function g(·).
8. The agents Y_j ($j = 1, 2, \ldots, m$) update their status with respect to the agent Z by means of uncertainty importance (sensitivity analysis) to evaluate the effect of the variance of their output variables, y_j, on the variance of the output variable of the agent Z (this process is described in Section 7.2).
9. The agents Y_j ($j = 1, 2, \ldots, m$) communicate their updated believes about the probability distributions of their output variables, $\pi(y_j)$, and about their status with respect to other agents (active or inactive) to all agents within the agent hierarchy, including the agent Z.
10. The agent Z reacts to the updated beliefs of the input agents Y_j ($j = 1, 2, \ldots, m$) by updating the distribution of its output variable, $\pi(z)$, by means of simulation over the updated model function, g(·), and using the updated probability distributions $\pi(y_j)$ of those and only those input variables y_j that represent the agents Y_j with an active status with respect to the agent Z. Probability distributions $\pi(y_j)$ of the input variables y_j associated with inactive agents Y_j with respect to the agent Z are not used in the simulation of the probability distribution $\pi(z)$. For the purpose of this simulation, inactive input agents are assigned with constant values derived from their respective probability distributions, $\pi(y_j)$, such as distribution mean or median.
11. Steps 1 to 10 above are repeated every time new data/information becomes available.

The above process of the autonomy execution suggests that the agent autonomy uses only the most relevant elements for the system reliability assessment since only the active agents are included in the simulation. This brings several benefits which distinguish the PoF-based agent autonomy approach from other existing methods of system reliability modeling and makes the autonomy property of agents a “core” contribution of this research:

1. Agent autonomy offers more efficient algorithm due to less frequent updates and reduced computational effort. In resemblance to the well-known 80/20 rule, the concept could be described as “80% of the answer is delivered by 20% of the agents and achieved it with 20% of the modeler’s efforts”.
2. Agent autonomy delivers higher quality of prior information for the future use of the mobile agents in the agent autonomy of similar systems.
3. Agent autonomy provides stronger guidance for uncertainty reduction by pointing to the multiple elements of the system which are not “important”.
4. Since any agent could have an “active” status with respect to one agent and an “inactive” status with respect to the other while constantly updating its probabilistic model by learning from new data/information, the agents are said to be active globally but active/inactive locally. It means that all agents remain to be probabilistic at all times during the system modeling and share the probabilistic information about their output variables with other agents in the hierarchy (i.e. being “active globally”), however become deterministic when deactivate themselves only with respect to some selected agents and only until the uncertainty importance defines so (i.e. become “inactive locally”).

7.4. Uncertainty Characterization within the Agent Autonomy

Uncertainty characterization within agent autonomy has some advantages as well as some challenges. Both epistemic and aleatory uncertainties are present within the agent autonomy model as they emerge from various sources, such as:

1. Uncertainty in the functional form of the agent output model (primarily epistemic uncertainty due to lack of engineering knowledge about the physical phenomena, human reliability or software errors, or due to lack of data for the right choice of the best fit model).
2. Uncertainty in the agent output model parameters (primarily epistemic uncertainty due to lack of data for parameters estimation, but aleatory uncertainty could also take place for complex dynamic systems).
3. Uncertainty in the agent input variables, which includes both epistemic and aleatory uncertainties related to the input variables as well as initial and boundary conditions.
4. Data uncertainty due to partial relevance, conflicting pieces of information within the data set, subjectivity of engineering judgment and expert opinion, and/or measurement errors.
5. Uncertainty in the reasoning algorithm (causal relationship model) where indirect measurements of a certain quantity are used as information (data) for agent learning.
6. Calculation uncertainties as a result of simplification, approximation or rounding errors, and/or the use of simulation or numerical techniques in lieu of closed form solution.

7. Data discretization related uncertainties.

These uncertainties propagate through the agent autonomy modeling structure and form the uncertainties about the output variables of all agents up to the Type III Monitoring Agent providing system reliability measures. A range or a distribution of the agent output variable or the agent output model parameters are used to quantify uncertainties within the agent autonomy framework.

The two-step uncertainty quantification could be performed. The first step quantifies agent output model related uncertainties associated with the model structure and model parameters (items 1 and 2 from the above list of uncertainty sources) while holding the input variables constant. The second step uses a simulation to update the uncertainty distribution of the agent output variable according to the uncertainty distributions of the input agent variables to quantify uncertainties introduced by the agent model inputs (item 3 from the above list of uncertainty sources). In the case of a single deterministic model representing the agent output variable, the first step of uncertainty quantification comes to evaluation the (uncertain) error term (which could be additive, multiplicative or both, as discussed in Section 5.1).

The above process may assist the separation of epistemic and aleatory uncertainties in the final results, however such separation is generally not a straightforward process. In most applications, the modeler could make a case for both types of uncertainties where, for instance, epistemic uncertainty is represented by a probability distribution of the value of the random (aleatory) characteristic. In order to make a clear distinction between epistemic and aleatory uncertainties, the modeler must have good understanding of the nature of the physical model of the phenomenon associated with the given variable. Within the agent autonomy modeling, which relies on engineering knowledge of the degradation and failure of a complex dynamic system and often uses limited or partially relevant data, making a distinction between the two types of uncertainties becomes quite challenging, if possible at all. It is much more critical for the modeler, however, to evaluate uncertainty importance of the variables representing the various agents in order to distinguish the top contributors to the uncertainty in the system reliability measures and identify uncertainty reduction opportunities.

Uncertainty importance assessment is performed within the context of agent autonomy execution. Global methods of uncertainty importance analysis by variance decomposition appear to be the most appropriate for the agent autonomy representing dynamic engineering system. The global methods, however, impose significant computational effort in evaluating the importance indexes that limits their practicality in many cases. Local methods of uncertainty importance analysis are relatively simple and computationally inexpensive, but are invalid or at least inaccurate in the identification of key contributors to the uncertainty when complex PoF modeling is involved. Despite their complexity, global methods of uncertainty importance analysis are yet to be used for agent autonomy containing bidirectional communication between agents and feedback loops. Such interactions between agents result in an “amplification” of uncertainty through the agent-based reliability model, making accurate identification of the uncertainty “drivers” of paramount importance.

The modeler's knowledge of physical failure mechanisms and availability of the associated PoF models is critical for the quantitative assessment of model-related uncertainties, especially in the agent autonomy approach which has an ultimate goal of bringing physics-of-failure into the system reliability assessment. In case of limited PoF knowledge of the system degradation processes, several PoF and empirical models $Y = f(X)$ should be tested as part of agent learning allowing each agent to "choose" the most appropriate model of its output variable according to the amount of uncertainty. In addition, weighting and combining several plausible models, and switching between the models according to the specified conditions could also be exercised by the agent during model selection process [116]. The modeler's judgment plays an important role when choosing which treatment options to be programmed into the agent. For complex models within the agent autonomy, special computer techniques may be developed to compare model uncertainties [117].

While an agent "matures" upon learning from new information, updating the functional form and parameters of the agent output model results in the reduction of epistemic uncertainties throughout the system reliability model. Aleatory uncertainty, often associated with the inherent variability of the output variables of the independent agents (i.e. Type I and Type II agents with no inputs from other agents), will remain in the model along with other sources of irreducible uncertainty (such as data uncertainties, calculation uncertainties, and data discretization uncertainties).

Data availability is a common source of epistemic uncertainty when limited and partially relevant data need to be used for reliability model development.

Bayesian inference is, therefore, defined as a preferred framework of agent learning to maximize use of the available information.

The agent autonomy approach also offers several improvements with respect to uncertainty characterization. Since the agent autonomy approach maximizes data usage and makes the agents mobile (usable in other applications), it is advantageous for the decision makers performing high-consequence risk analysis of a complex engineering system with limited knowledge about the system behavior. In addition, the agent autonomy approach to system reliability modeling allows for some reduction of subjectivity and arbitrariness in the definition of system failure scenarios and their consequences compared to fault tree and event tree methodologies because the intelligent agents evolve autonomously reflecting on all relevant failure scenarios (not only those believed to be the “worst case” sequences). This is particularly important for high-hazard industries, such as nuclear, aerospace, defense and several others, where both underestimation and overestimation of the criticality of accident scenarios lead to potentially significant consequences.

Another aspect of uncertainty characterization is related to the capability of the agent autonomy to provide realistic representation of the system dynamics as it evolves over time. For example, fault trees and event trees typically use the classical binary success/failure logic of system reliability representation. In addition, fault trees and event trees are static techniques which cannot take into account time-dependent evolutions of dynamic systems. In contrary, the agent autonomy approach effectively models the dynamics of system evolution in time and differentiates between the different levels of system performance depending on the degraded states of the constitutive parts and components. As a result, epistemic uncertainties, introduced by simplification and approximation of the reality, will be reduced for agent autonomy models.

Chapter 8: Case Study: Agent-Based Reliability Assessment of Gas Turbine Aircraft Engine Structures

8.1. Introduction

This chapter provides an application example of a gas turbine system reliability analysis by means of the agent-based PoF modeling in order to demonstrate concepts and methods of agent autonomy that are introduced in this work.

A gas turbine is a type of internal combustion engine used to power aircrafts, trains, ships, generators, or tanks. In the aerospace industry, current aircraft maintenance practices rely on highly conservative life estimates for critical gas turbine engine components to ensure that they are replaced prior to failure. While these practices have resulted in extremely low failure rates, they also reduce the aircraft's availability and incur significant labor costs to replace the components with significant remaining useful lives. In some cases, even highly conservative life estimates, however, cannot account for the extreme or unpredictable circumstances that contribute to many of the documented engine failures. It is therefore critical, to develop a reliability model that will enable accurate evaluation of the life of critical engine components for each known mission profile and specific use conditions. The accuracy of the model for determining the component life and monitoring its consumption significantly impacts the engine safety and life cycle cost. The physics-based agent autonomy approach, presented in this dissertation, will be used to combine the test and field data with the life models to estimate life to failure for gas turbine components.

8.2. Gas Turbine Overview

A turboprop engine is a turbine engine where an aircraft propeller is driven using a reduction gear [118]. A turboprop engine consists of air intake section, compressor, combustor, turbine (also called high pressure turbine), gearbox, and propelling nozzle (exhaust section), as shown on Figure 8-1. Air is drawn into the intake section, compressed by the compressor, and forced into the combustor. The compressed air is mixed with fuel in the combustion chamber, where the fuel-air mixture is then ignited by a spark. The fuel burns producing hot gases, which expand and drive the fan blades of the high pressure turbine. Most of the power generated by the turbine is transmitted to the propeller through the reduction gear, while some power is used to drive the compressor. The combustion gases expand further in the propelling nozzle where they are discharged into the atmosphere. The propelling nozzle creates only a small proportion of the total thrust generated by a turboprop engine (the turboprop engine's exhaust gases contain very low energy compared to a jet engine and make only a small contribution into the aircraft propulsion).

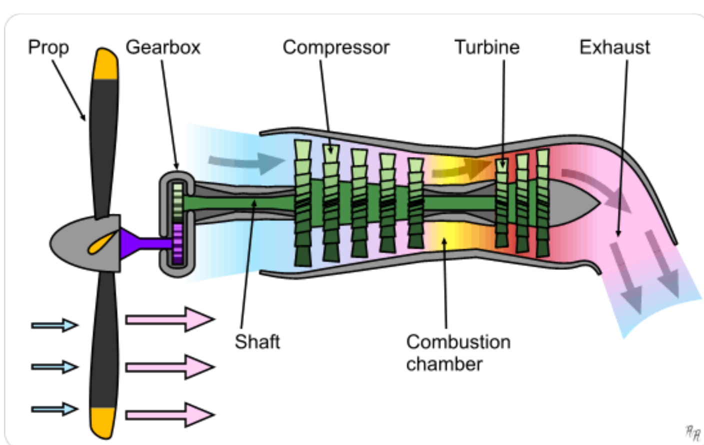


Figure 8-1: Schematic Diagram of a Turboprop Engine
(from http://commons.wikimedia.org/wiki/File:Turboprop_operation.png)

This case study involves high pressure turbine of a turboprop engine. A single turbine section (stage) contains a rotor wheel. The turbine wheel includes a turbine disk that holds turbine blades. The turbine stages are splined together and secured by the bolts. A roller bearing at the forward end and another at the aft end of the turbine shaft support the entire assembly. The number of turbine sections (stages) varies in different types of engines. The high pressure turbine sections are connected to the compressor sections with a shaft. The air pressure and temperature are rising while the air is compressed in the compressor stages of the engine. The air pressure and temperature are significantly increased further due to fuel combustion inside the combustor chamber, which is located between the compressor and the turbine. The high temperature and high pressure gases pass through the high pressure turbine stages, where the energy is extracted from the air flow, lowering the temperature and pressure of the air. The high pressure turbine is, therefore, exposed to the hottest and highest pressure air.

8.3. Gas Turbine Components Life Consumption

The design of high performance gas turbine engines have made the overall turbine structural reliability limited by the fatigue life of major rotating components of the high pressure turbine. Gas turbine discs are usually the most critical components which must endure substantial mechanical and thermal loading. The degraded performance of other components of the high pressure section adjacent to the disks, such as blades, shaft and bearings, also contribute to disk reliability.

If a problem arises in the turbine section, it will significantly affect the engine functionality and safety of the aircraft. Blade loss can be contained within the engine casing, while the catastrophic failure of a turbine wheel (disk and blade assembly) could cause a puncture of the engine casing by the larger fragments of the disc. Turbine engine bearing failures are another leading cause of engine loss.

In this work, the agent-oriented PoF model is developed for interacting failure mechanisms of a high pressure turbine sub-assembly of three components: turbine disk (the first stage disk), a high pressure shaft and two roller bearings. For simplicity, this research only deals with the above three types of components which are among the most important contributors to the reliability and safety of a high pressure turbine of an aircraft engine, based on evidence from accelerated life test data and field maintenance records. The first stage disk is chosen because the first stage turbine rotor components are the most severely loaded out of the four stages in the studied gas turbine engine. Cyclic fatigue is the leading failure mechanism for the studied components, as explained in Appendix A.

8.4. Physics-of-Failure Fatigue Life Model of Gas Turbine Components

Probabilistic-mechanistic life models of the fatigue failure mechanism in high pressure turbine bearings, shaft and disk are is shown on Figure 8-2, Figure 8-3 and Figure 8-4, respectively. These PoF models are developed from the physical principles of the component operation, considering the critical variables which contribute to the failure process, as described in Appendix A of this dissertation. A detailed derivation of the PoF-based life-stress equations for the high pressure turbine components can be found in Appendix A.

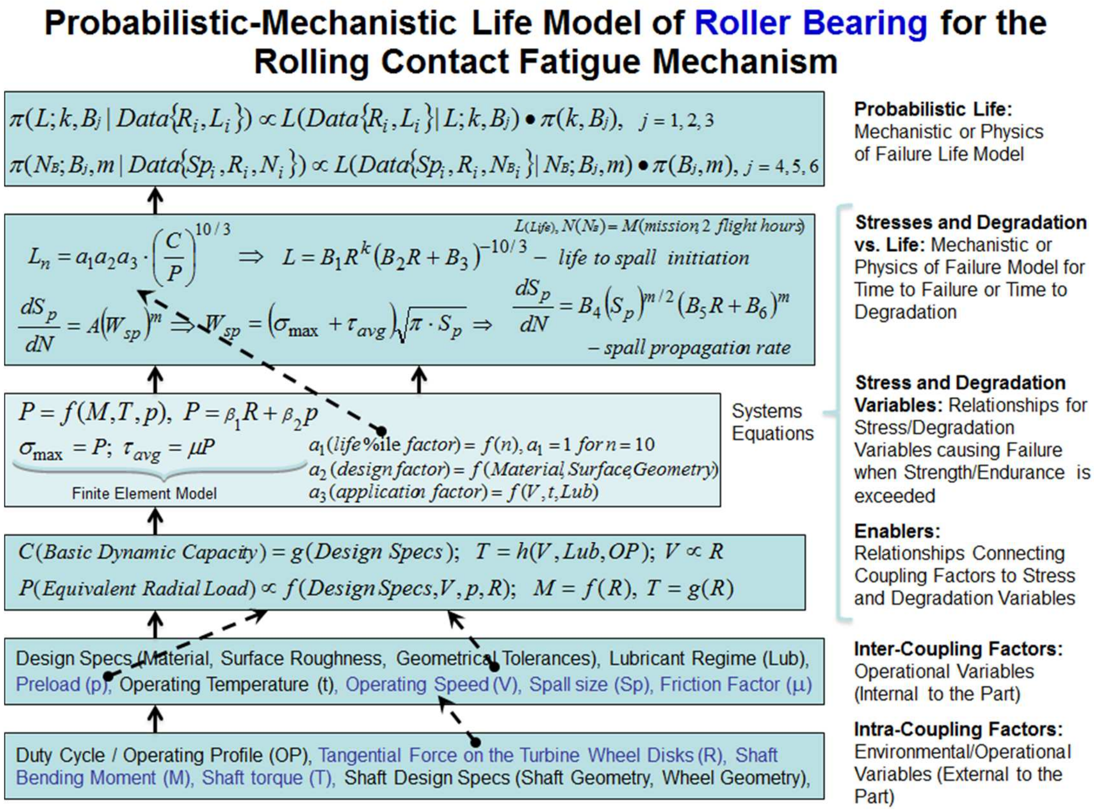


Figure 8-2: Reliability-Based Fatigue Life Model of High Pressure Turbine Bearing

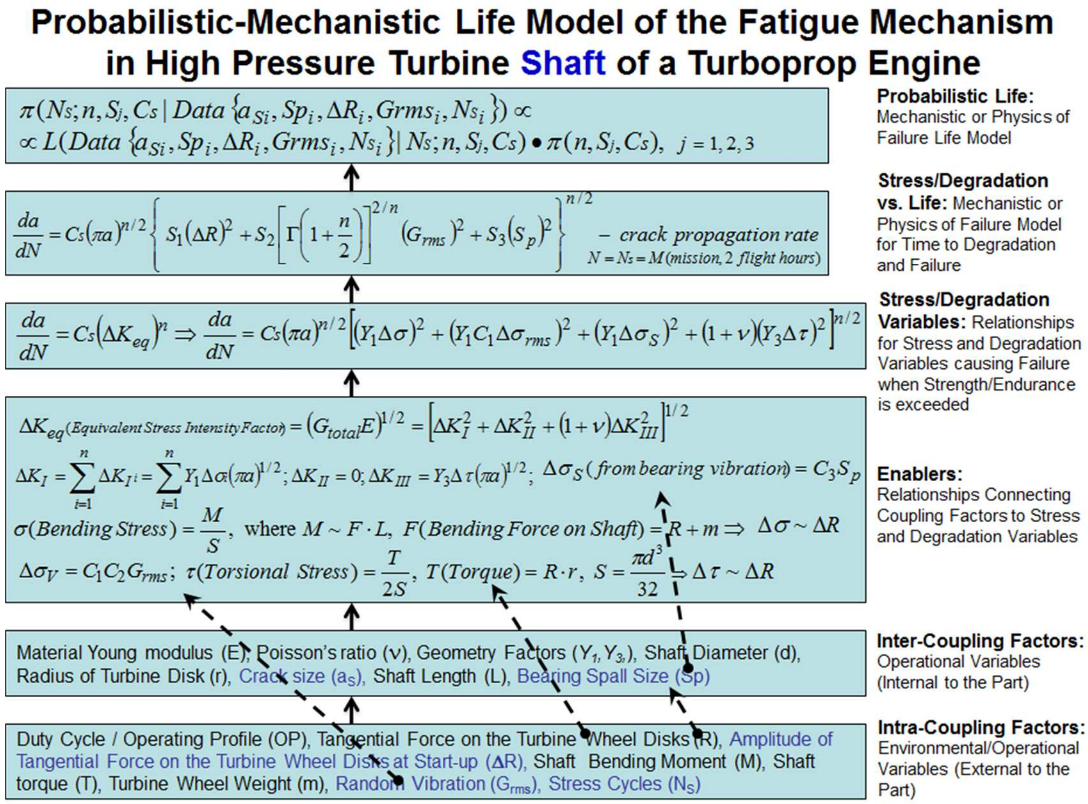


Figure 8-3: Reliability-Based Fatigue Life Model of High Pressure Turbine Shaft

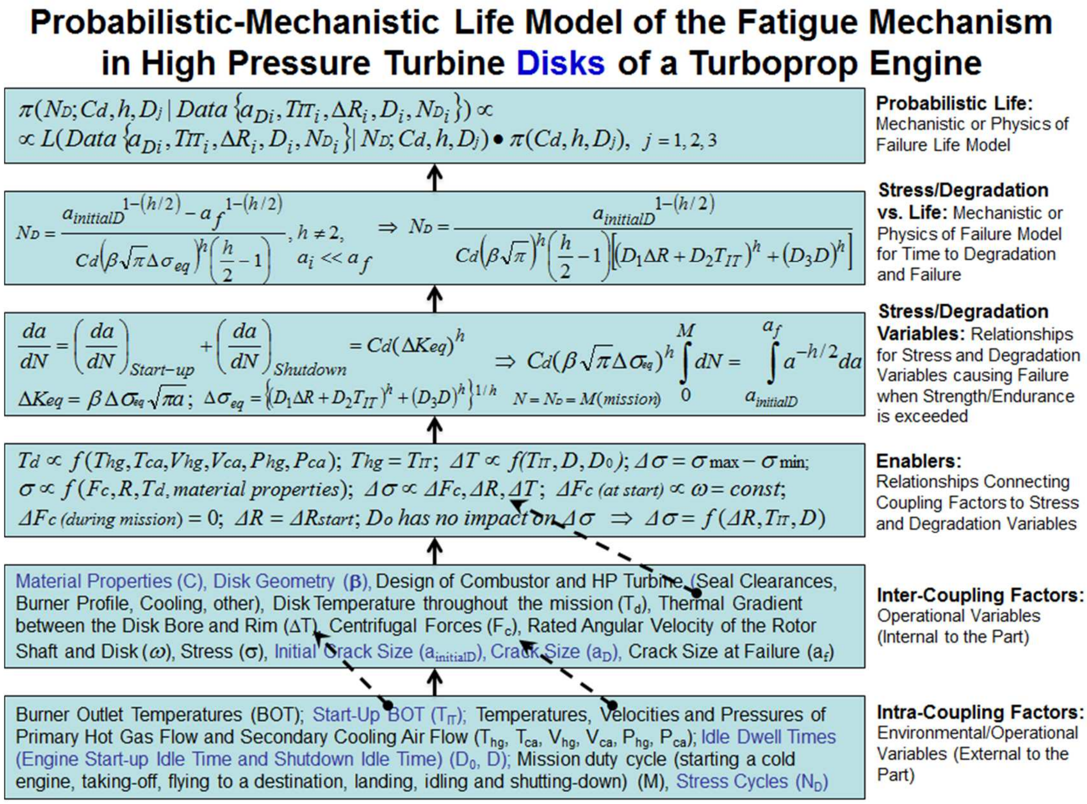


Figure 8-4: Reliability-Based Fatigue Life Model of High Pressure Turbine Disk

8.5. Definition of Agents

The next step is to develop an agent-oriented representation of the physical models of the gas turbine structures presented in the last sections, considering agent classification defined in Chapter 3. An intelligent agent is assigned to each element of the physical model of failure according to the nature of the element and its role in the PoF hierarchy (shown on Figure 8-2, Figure 8-3 and Figure 8-4). Type I Micro-Agents are listed in Table 8-1. Type II Macro-Agents and Type III Monitoring Agents are defined in Table 8-3 and Table 8-4, respectively.

Table 8-1: Agent Autonomy for Reliability Modeling of Gas Turbine Structures - Type I Micro-Agents

Type I Micro-Agents				
ID #	Name of Agent Output Variable	Letter ID	Representation of Agent Output Variable	Agent Learning Method
1	**Tangential Force on the Turbine Wheel Disks**	R	Monitored during operation to obtain **Data** $\{R_i\}$ and develop probability distribution $\pi(R)$	Classical methods of parametric distribution fitting to the **Data** $\{R_i\}$ to obtain probability distribution $\pi(R)$
2	**Amplitude of Tangential Force on the Turbine Wheel Disks at Start**	ΔR	Monitored during operation to obtain **Data** $\{\Delta R_i\}$ and develop probability distribution $\pi(\Delta R)$	Classical methods of distribution fitting to the **Data** $\{\Delta R_i\}$ to obtain probability distribution $\pi(\Delta R)$
3	**Random Vibration**	G_{rms}	Monitored during operation to obtain **Data** $\{G_{rmsi}\}$ and develop probability distribution $\pi(G_{rms})$	Classical methods of distribution fitting to the **Data** $\{G_{rmsi}\}$ to obtain probability distribution $\pi(G_{rms})$
4	**Burner Outlet Temperature (BOT) at Start**	T_{IT}	Monitored during operation to obtain **Data** $\{T_{ITi}\}$ and develop probability distribution $\pi(T_{IT})$	Classical methods of distribution fitting to the **Data** $\{T_{ITi}\}$ to obtain probability distribution $\pi(T_{IT})$
5	**Dwell Time before Shutdown**	D	Monitored during operation to obtain **Data** $\{D_i\}$ and develop probability distribution $\pi(D)$	Classical methods of distribution fitting to the **Data** $\{D_i\}$ to obtain probability distribution $\pi(D)$
6	**Initial Crack Size, Shaft**	$a_{initialS}$	Obtained from past experience and analysis, measured before operation start to obtain **Data** $\{a_{initialSi}\}$ and develop probability distribution $\pi(a_{initialS})$	Classical methods of distribution fitting to the **Data** $\{a_{initialSi}\}$ to obtain probability distribution $\pi(a_{initialS})$
7	**Initial Crack Size, Disk**	$a_{initialD}$	Obtained from past experience and analysis, measured before operation start to obtain **Data** $\{a_{initialDi}\}$ and develop probability distribution $\pi(a_{initialD})$	Classical methods of distribution fitting to the **Data** $\{a_{initialDi}\}$ to obtain probability distribution $\pi(a_{initialD})$
8	**Accumulated Missions**	M	Mission count from operation start to data collection time point. Deterministic quantity changing over time, taking known values	Following known change pattern over time (known values of test missions count at data collection time point)

According to the agent definitions in Table 8-1, the first seven Type I Micro-Agents are probabilistic independent agents that have no inputs from other agents. They do not have a model of the output variable and learn by parametric distribution analysis (classical) using operational data. The Type I Micro-Agent ID #8, accumulated missions, is a deterministic time dependent agent, evolving over time in a known manner.

All eleven Type II Macro-Agents, listed in Table 8-3, are dependent agents with several inputs from other agents. They have the PoF model of the output variable and learn by Bayesian inference using bench test data.

Type III Monitoring Agents, shown in Table 8-4, are developed to evaluate the remaining useful life for the components of each of the three tested engines (disk, shaft and two bearings) and for each engine as a system.

Table 8-2 provides a list of constants used in the case study which were not defined as agents due to their deterministic nature (however, having a sequential ID number assigned along with the agents). Some of these constants (such as disk geometry constant or bearing spall size threshold) can be assigned with an agent in future studies if uncertainties arise.

Table 8-2: Agent Autonomy for Reliability Modeling of Gas Turbine Structures - Constants

Constants				
ID #	Name of the Constant	Letter ID	Definition / Source	Value
9	**Disk Geometry Constant**	$\boldsymbol{\beta}$	Obtained from past experience and stress analysis	β = 1.12
10	**Bearing Spall Size Threshold**	$\boldsymbol{S_{pLimit}}$	Obtained from past experience and analysis	S_{pLimit} = 0.2 in^2
11	**Number of Tested Engines**	$\boldsymbol{E}$	Sample size, defined according to test set-up	E = 3

Table 8-3: Agent Autonomy for Reliability Modeling of Gas Turbine Structures - Type II Macro-Agents

Type II Macro-Agents							
			Public Attributes				
ID #	Name of Agent Output Variable	Letter ID	Representation of Agent Output Variable	Agent Learning Method	Input Agents and Constants	Data for Agent Learning	Agent Output Model Parameters
12	**Bearing Life to Spall Initiation (Bearing #1)**	L_1	• Weibull probability distribution $\pi(L_1) = W(\beta_1, L_1)$ • PoF model for L_1: see Equation 8-1	Bayesian Inference	R	Time to spall initiation (complete observations L_{1i} right censored observations t_{1i}), conditional on input variable R: $\{L_{1i}, t_{1i}, R_i\}$, $i = 1, \ldots, E$, where E is sample size	β_1 k $B_{1(1)}$ $B_{2(1)}$ $B_{3(1)}$
13	**Bearing Life to Spall Initiation (Bearing #2)**	L_2	• Weibull probability distribution $\pi(L_2) = W(\beta_2, L_2)$ • PoF model for L_2: see Equation 8-2	Bayesian Inference	R	Time to spall initiation (complete observations L_{2i} right censored observations t_{2i}), conditional on input variable R: $\{L_{2i}, t_{2i}, R_i\}$, $i = 1, \ldots, E$, where E is sample size	β_2 k $B_{1(2)}$ $B_{2(2)}$ $B_{3(2)}$

$$L_1 = B_{1(1)} R^k \left(B_{2(1)} R + B_{3(1)}\right)^{-10/3}$$

Equation 8-1

$$L_2 = B_{1(2)} R^k \left(B_{2(2)} R + B_{3(2)}\right)^{-10/3}$$

Equation 8-2

14	**Bearing Spall Size (Bearing #1)**	S_{p1}	• Lognormal probability distribution $\pi(S_{p1}) = LN(S_{p1}, \sigma_{B1})$ • PoF model for S_{p1}: see Equation 8-3	Bayesian Inference	R L_1 M	Missions M_i to spall propagation from zero to size S_{p1i}, conditional on input variable R: $\{M_i, R_i, S_{p1i}\}$, $i = 1, \ldots, E$, where E is sample size	σ_{B1} m $B_{4(1)}$ $B_{5(1)}$ $B_{6(1)}$

<table>
<tr><th colspan="8">Type II Macro-Agents</th></tr>
<tr><th rowspan="2">ID #</th><th rowspan="2">Name of Agent Output Variable</th><th rowspan="2">Letter ID</th><th colspan="5">Public Attributes</th></tr>
<tr><th>Representation of Agent Output Variable</th><th>Agent Learning Method</th><th>Input Agents and Constants</th><th>Data for Agent Learning</th><th>Agent Output Model Parameters</th></tr>
<tr><td>15</td><td>Bearing Spall Size (Bearing #2)</td><td>$\boldsymbol{S_{p2}}$</td><td>• Lognormal probability distribution $\boldsymbol{\pi(S_{p2}) = LN(S_{p2}, \sigma_{B2})}$
• PoF model for $\boldsymbol{S_{p2}}$: see Equation 8-4</td><td>Bayesian Inference</td><td>R
L_2
M</td><td>Missions M_i to spall propagation from zero to size S_{p2i}, conditional on input variable R: $\{M_i, R_i, S_{p2i}\}, i = 1, \ldots, E$, where E is sample size</td><td>σ_{B2}
m
$B_{4(2)}$
$B_{5(2)}$
$B_{6(2)}$</td></tr>
<tr><td colspan="8">$$S_{p1} = \left[B_{4(1)}\left(B_{5(1)}R + B_{6(1)}\right)^m \left(1 - \frac{m}{2}\right)(M - L_1)\right]^{\frac{1}{1-\frac{m}{2}}}, \qquad m \neq 2$$
Equation 8-3
$$S_{p2} = \left[B_{4(2)}\left(B_{5(2)}R + B_{6(2)}\right)^m \left(1 - \frac{m}{2}\right)(M - L_2)\right]^{\frac{1}{1-\frac{m}{2}}}, \qquad m \neq 2$$
Equation 8-4</td></tr>
<tr><td>16</td><td>Maximum Spall Size</td><td>$\boldsymbol{S_p}$</td><td>• Probability distribution $\boldsymbol{\pi(S_p)}$ is obtained by simulation via PoF model for S_p
• PoF model for $\boldsymbol{S_p}$: see Equation 8-5</td><td>Simulation</td><td>S_{p1}
S_{p2}</td><td>No data required</td><td>Parameters of probability distribution $\boldsymbol{\pi(S_p)}$, obtained via simulation from $\boldsymbol{\pi(S_{p1})}$ and $\boldsymbol{\pi(S_{p2})}$, are defined by selecting best fit parametric distribution function for the simulated distribution $\boldsymbol{\pi(S_p)}$</td></tr>
<tr><td colspan="8">$$S_p = Max\{S_{p1}, S_{p2}\}_M$$
Equation 8-5</td></tr>
</table>

Type II Macro-Agents							
			Public Attributes				
ID#	Name of Agent Output Variable	Letter ID	Representation of Agent Output Variable	Agent Learning Method	Input Agents and Constants	Data for Agent Learning	Agent Output Model Parameters
17	**Bearing Life to Spall Propagation to Critical Size at Failure (Bearing #1)**	$\boldsymbol{N_{B1}}$	• Weibull probability distribution $\boldsymbol{\pi(N_{B1}) = W(\beta_{B1}, N_{B1})}$ • PoF model for $\boldsymbol{N_{B1}}$: see Equation 8-6 (derived from PoF model for agent $\boldsymbol{S_{p1}}$, ID #14, note the same model parameters $B_{4(1)}$, $B_{5(1)}$, $B_{6(1)}$)	Bayesian Inference	R S_{pLimit}	Missions M_i to failure or to spall propagation to size S_{p1i}, conditional on input variable R: *{M_i, i-th Failure / No Failure Condition of Bearing #1, R_i, S_{p1i}}, i = 1, …, E,* where E is sample size	β_{B1} m $B_{4(1)}$ $B_{5(1)}$ $B_{6(1)}$
18	**Bearing Life to Spall Propagation to Critical Size at Failure (Bearing #2)**	$\boldsymbol{N_{B2}}$	• Weibull probability distribution $\boldsymbol{\pi(N_{B2}) = W(\beta_{B2}, N_{B2})}$ • PoF model for $\boldsymbol{N_{B2}}$: see Equation 8-7 (derived from PoF model for agent $\boldsymbol{S_{p2}}$, ID #15, note the same model parameters $B_{4(2)}$, $B_{5(2)}$, $B_{6(2)}$)	Bayesian Inference	R S_{pLimit}	Missions M_i to failure or to spall propagation to size S_{p2i}, conditional on input variable R: *{M_i, i-th Failure / No Failure Condition of Bearing #2, R_i, S_{p2i}}, i = 1, …, E,* where E is sample size	β_{B2} m $B_{4(2)}$ $B_{5(2)}$ $B_{6(2)}$

$$N_{B1} = \frac{\left(S_{pLimit}\right)^{1-\frac{m}{2}}}{B_{4(1)}\left(B_{5(1)}R + B_{6(1)}\right)^{m}\left(1-\frac{m}{2}\right)}, \qquad m \neq 2$$

Equation 8-6

$$N_{B2} = \frac{\left(S_{pLimit}\right)^{1-\frac{m}{2}}}{B_{4(2)}\left(B_{5(2)}R + B_{6(2)}\right)^{m}\left(1-\frac{m}{2}\right)}, \qquad m \neq 2$$

Equation 8-7

Type II Macro-Agents							
			Public Attributes				
ID#	Name of Agent Output Variable	Letter ID	Representation of Agent Output Variable	Agent Learning Method	Input Agents and Constants	Data for Agent Learning	Agent Output Model Parameters
19	**Shaft Crack Size**	a_S	• Lognormal probability distribution $\pi(a_S) = LN(\mu_S, \sigma_S)$ • PoF model for a_S: see Equation 8-8	Bayesian Inference	ΔR G_{rms} $a_{initialS}$ Sp M	Missions M_i to crack propagation to size a_{Si}, conditional on input variables ΔR, G_{rms}, S_p: $\{M_i, a_{Si}, \Delta R_i, G_{rmsi}, S_{pi}\}$, $i = 1, ..., E$, where E is sample size	σ_S n C_s S_1 S_2 S_3
$$a_S = \left[a_{initialS}{}^{1-n/2} - C_s \left[\pi \left\{ S_1(\Delta R)^2 + S_2 \left[\Gamma\left(1+\frac{n}{2}\right)\right]^{\frac{2}{n}} (G_{rms})^2 + S_3\left(S_p\right)^2 \right\} \right]^{\frac{n}{2}} \times \left(\frac{n}{2}-1\right) M \right]^{\frac{1}{1-n/2}}, \quad n > 2$$ Equation 8-8							
20	**Shaft Life to Crack Size at Failure**	N_S	• Weibull probability distribution $\pi(N_S) = W(\beta_S, N_S)$ • PoF model for N_S: see Equation 8-9 (derived from PoF model for agent a_S, ID #19, note the same model parameters S_1, S_2, S_3)	Bayesian Inference	ΔR G_{rms} $a_{initialS}$ Sp	Missions M_i to failure or to crack propagation to size a_{Si}, conditional on input variables ΔR, G_{rms}, S_p: $\{M_i$, *i-th Failure / No Failure Condition*, $a_{Si}, \Delta R_i, G_{rmsi}, S_{pi}\}$, $i = 1, ..., E$, where E is sample size	β_S n C_s S_1 S_2 S_3

<table>
<tr><th colspan="9">Type II Macro-Agents</th></tr>
<tr><th rowspan="2">ID#</th><th rowspan="2">Name of Agent Output Variable</th><th rowspan="2">Letter ID</th><th colspan="6">Public Attributes</th></tr>
<tr><th>Representation of Agent Output Variable</th><th>Agent Learning Method</th><th>Input Agents and Constants</th><th>Data for Agent Learning</th><th>Agent Output Model Parameters</th></tr>
<tr><td colspan="9">$$N_S = \frac{{a_{initialS}}^{1-n/2}}{C_s\left[\pi\left\{S_1(\Delta R)^2 + S_2\left[\Gamma\left(1+\frac{n}{2}\right)\right]^{2/n}(G_{rms})^2 + S_3\left(S_p\right)^2\right\}\right]^{n/2}\left(\frac{n}{2}-1\right)}, n > 2$$
Equation 8-9</td></tr>
<tr><td>21</td><td>Disk Crack Size</td><td>$\boldsymbol{a_D}$</td><td>• Lognormal probability distribution $\pi(a_D) = LN(\mu_D, \sigma_D)$
• PoF model for a_D: see Equation 8-10</td><td>Bayesian Inference</td><td>ΔR
T_{IT}
D
$a_{initialD}$
M
β</td><td>Missions M_i to crack propagation to size a_{Di}, conditional on input variables ΔR, T_{IT}, D: $\{M_i, a_{Di}, \Delta R_i, T_{ITi}, D_i\}$, $i = 1, \ldots, E$, where E is sample size</td><td>σ_D
h
C_d
D_1
D_2
D_3</td></tr>
<tr><td colspan="9">$$a_D = \left[{a_{initialD}}^{1-(h/2)} - C_d\left(\beta\sqrt{\pi}\right)^h\left(\frac{h}{2}-1\right)\{(D_1\Delta R + D_2 T_{IT})^h + (D_3 D^{-1})^h\}M\right]^{\frac{1}{1-(h/2)}}, h > 2$$
Equation 8-10</td></tr>
</table>

Type II Macro-Agents							
			Public Attributes				
ID#	Name of Agent Output Variable	Letter ID	Representation of Agent Output Variable	Agent Learning Method	Input Agents and Constants	Data for Agent Learning	Agent Output Model Parameters
22	**Disk Life to Crack Size at Failure**	$\boldsymbol{N_D}$	• Weibull probability distribution $\boldsymbol{\pi(N_D) = W(\beta_D, N_D)}$ • PoF model for $\boldsymbol{N_D}$: see Equation 8-11 (derived from PoF model for agent $\boldsymbol{a_D}$, ID #21, note the same parameters of the model, D_1, D_2, D_3)	Bayesian Inference	ΔR T_{IT} D $a_{initialD}$ β	Missions M_i to failure or to crack propagation to size a_{Di}, conditional on input variables ΔR, T_{IT}, D: $\{M_i$, *i-th Failure / No Failure Condition*, a_{Di}, ΔR_i, T_{ITi}, $D_i\}$, $i = 1, \ldots, E$, where E is sample size	β_D h C_d D_1 D_2 D_3

$$N_D = \frac{{a_{initialD}}^{1-(h/2)}}{C_d\left(\beta\sqrt{\pi}\right)^h\left(\frac{h}{2}-1\right)\{(D_1\Delta R + D_2 T_{IT})^h + (D_3 D^{-1})^h\}}, h > 2$$

Equation 8-11

Table 8-4: Agent Autonomy for Reliability Modeling of Gas Turbine Structures - Type III Monitoring Agents

Type III Monitoring Agents: Component-Monitoring Agents					
ID#	Name of Agent Output Variable	Letter ID	Public Attributes		
			Item Monitored	Representation of Agent Output Variable	Method of Agent Learning
23	**Remaining Useful Life of Bearing #1**	**RUL_{B1}**	Bearing #1	Probabilistic, by distribution $\pi(RUL_{B1})$ $RUL_{B1} = L_1 + N_{B1} - M$	Simulation from input agents L_1 (ID# 15), N_{B1} (ID# 17) and M (ID# 8)
24	**Remaining Useful Life of Bearing #2**	**RUL_{B2}**	Bearing #2	Probabilistic, by distribution $\pi(RUL_{B2})$ $RUL_{B2} = L_2 + N_{B2} - M$	Simulation from input agents L_2 (ID# 16), N_{B2} (ID# 18) and M (ID# 8)
25	**Remaining Useful Life of the Shaft**	**RUL_S**	Shaft	Probabilistic, by distribution $\pi(RUL_S)$ $RUL_S = N_S - M$	Simulation from input agents N_S (ID# 20) and M (ID# 8)
26	**Remaining Useful Life of the Disk**	**RUL_D**	Disk	Probabilistic, by distribution $\pi(RUL_D)$ $RUL_D = N_D - M$	Simulation from input agents N_D (ID# 21) and M (ID# 8)
Type III Monitoring Agents: System-Monitoring Agent					
ID#	Name of Agent Output Variable	Letter ID	Public Attributes		
			Components Included	Representation of Agent Output Variable	Method of Agent Learning
27	**Remaining Useful Life of the System**	**RUL**	Bearing #1 Bearing #2 Shaft Disk	Probabilistic, $RUL = Min\{RUL_{B1}, RUL_{B2}, RUL_S, RUL_D\}$	Simulation from Type III Agents ID# 23, ID# 24, ID# 25 and ID# 26

As shown in Table 8-4, Type III Monitoring Agents are assigned to evaluate the remaining useful life (RUL) of each component and the system by simulation over the respective RUL equation, given the probability distributions of the input agents. RUL, also called remaining service life or residual life, refers to the time left before observing a failure given the current age and condition of an item, and the past operation profile. Since the time to fatigue failure of each component is a random variable, the system RUL is also a random variable, and the distribution of RUL would be of interest for a full understanding of the system reliability. Probabilistic prognosis of the RUL of each component and the system with respect to a critical crack size (for the shaft and the disk) and critical spall size (for the bearings) can be obtained from the distributions of bearing spall size, shaft crack size, and disk crack size, assuming a known future usage profile.

An estimation of the component and system RUL has been extensively researched within the framework of PHM (prognostics and health management). RUL estimates are commonly obtained by data-driven approaches (machine learning and pattern recognition methods, described in Section 5.5) applied to failure precursor measurements, or according to the PoF-based cumulative damage model of the item of interest. “Fusion” approaches as a combination of the two are still being researched [119]. The difference between the PoF model-based PHM and the intelligent agent autonomy approach is that, compared to the PoF based life-stress models used in PHM, the agent autonomy offers more comprehensive and flexible framework for addressing dynamic interactions between several failure processes evolving over time within the system. The agent Autonomy approach to PPoF system reliability, when fully developed and validated, may become the most suitable solution for the “fusion” of data-driven and PoF-based models in PHM.

8.6. Agents Hierarchy

The agent hierarchy shown in Figure 8-5 was developed according to the agent classification given by Table 8-1, Table 8-3, Table 8-4 and the physical model of failure for each component (Figure 8-2, Figure 8-3, Figure 8-4) and the system (as a serial system of four components). This agent hierarchy represents the concept of Figure 3-2, as applicable to the agent structure of the studied system.

Comparing the agent hierarchy on Figure 8-5 with PoF models shown in Figure 8-2, Figure 8-3 and Figure 8-4, it can be seen that Type II Macro-Agents represent stress-strength and degradation-endurance variables, life to failure or time to degradation. Type I Micro-Agents represent only probabilistic coupling factors (i.e. those which are included in stress-life PoF relationships as random variables), while several deterministic coupling factors are included into the PoF model parameters, and are not assigned with an agent as they are not expected to change over the life of the turbine structure (the exceptions are disk geometry factor and spall size threshold, which are currently defined as constants in Table 8-2, but may need to be assigned with an agent, if it becomes necessary in future due to design changes to the respective components). For simplicity and to keep a reasonable size of the diagram, only probabilistic agents are presented on Figure 8-5. Enablers, stress vs. life relationships and probabilistic life relationships are not shown because they represent agents' internal knowledge, specifically the PoF models of the output variables of the respective Type II Macro-Agents and Type III Monitoring Agents. The deterministic time-dependent Type I Micro-Agent accumulated missions is also not shown on the agent hierarchy because it simply defines total accumulated time since time point zero.

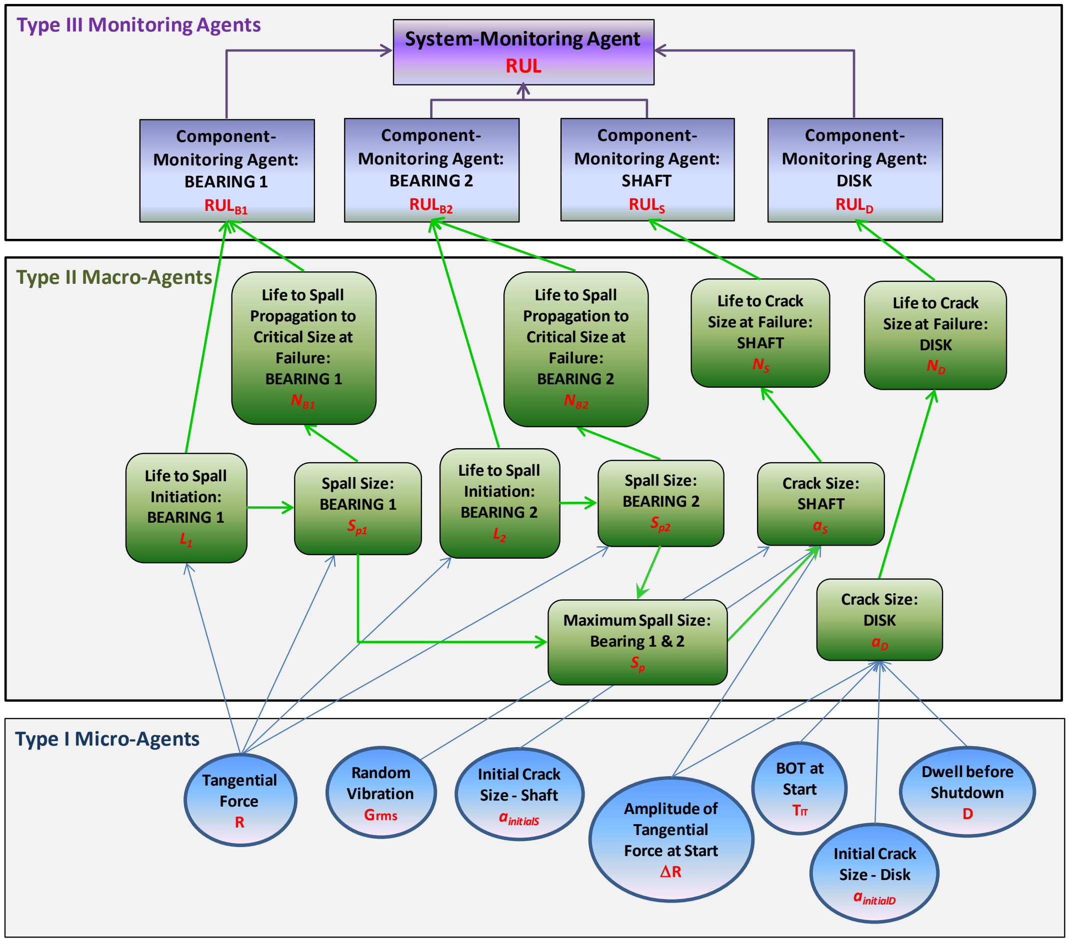


Figure 8-5: Hierarchy of Agents and their Communication Scheme

Agent definitions proposed in Section 8.5 and reflected in the current hierarchy of agents are also intended to demonstrate one critical objective of this work: the ability to capture the interactions between the failure mechanisms of system elements making the agent autonomy superior to other system reliability modeling methods. This capability is built specifically into the PoF model of the shaft as a dependency of the shaft crack size on the bearing spall size through the maximum value of the spall size among the two bearing for each given point in time. Equation 8-8 and, therefore, Equation 8-9 contain the term S_p which represents the Type II

Macro-Agent, maximum spall size, which is dependent on the spall size of each of two bearings. This example demonstrates the key capability of agent autonomy to explicitly model the interdependent degradation measures or other interdependent variables within system hierarchy.

It can be seen that the proposed agent hierarchy combines the agents of all types and shows the bidirectional communication between agents to model complex interdependencies between system elements. The arrows in Figure 8-5 depict physical and causal relationships between all types of agents. Type I Micro-Agents are the simplest independent agents which do not affect each other but only provide inputs to the higher level agents, particularly to Type II Macro-Agents. Type II Macro-Agents and Type III Monitoring Agents have complex relationships with each other and with other agents from one level up and one level down.

8.7. Agent Learning and Autonomy Properties

8.7.1. Mechanisms of Agent Learning

A mechanism of agent reasoning and learning requires special attention because learning is a key property of an intelligent autonomous agent along with the ability to activate/deactivate itself (autonomy property). The autonomy property is discussed in the next section 8.7.2. In this case study, agent reasoning and learning capabilities are established in accordance with the rules and procedures defined in Chapter 5 and Chapter 6. Specifically, independent Type I Micro-Agents are learning through the classical distribution analysis using the data collected during the flight test program and field operation, as described in Table 8-1. The learning property of Type II Macro-Agents is accomplished by Bayesian inference, as stated in Table 8-3. Type III Monitoring Agents evaluate the remaining useful life of the items they represent by simulation algorithm that is based on “series” system failure logic (system is considered failed if at least one failure mode occurs out of all possible modes).

The Bayesian framework presented below in Table 8-5 is an example of a learning capability of a Type II Macro-Agent, disk crack size. The same concept applies to the other Type II Macro-Agents.

Table 8-5: Learning Process of Type II Macro-Agent Disk Crack Size

Elements of Agent Learning Process	**Type II Macro-Agent** **Disk Crack Size**
Input Variables from Input Agents	Type I Micro-Agents: 1. Amplitude of Tangential Force on the Turbine Wheel Disks at Start, ΔR 2. BOT at start, T_{IT} 3. Dwell Time before Shutdown, D 4. Accumulated Missions, M 5. Disk Initial Crack Size, $a_{initialD}$ Constants: Disk Geometry Constant, β Note: *Disk Crack Size agent is time dependent via Accumulated Missions agent, M.*
Output Variable	Disk Crack Size, a_D

Elements of Agent Learning Process	Type II Macro-Agent
	Disk Crack Size
Model of Agent Output Variable - Past Beliefs	1. The PoF model connecting Disk Crack Size with operational stress factors: $$a_D = \left[{a_{initialD}}^{1-(h/2)} - C_d\left(\beta\sqrt{\pi}\right)^h \left(\frac{h}{2} - 1\right)\{(D_1 \Delta R + D_2 T_{IT})^h + (D_3 D)^h\} M \right]^{\frac{1}{1-(h/2)}}, \quad h > 2$$ 2. It is assumed that lognormal distribution represents the variability of Disk Crack Size, where μ_D and σ_D are the log-mean and log-standard deviation of the crack size distribution: $$\pi(a_D) = LN(\mu_D, \sigma_D)$$ 3. Using the PoF model of the disk crack size as a representation of the mean value of the crack size, the log-mean of the disk crack size distribution is expressed as follows: $$\mu_D = Ln\left(\left[{a_{initialD}}^{1-(h/2)} - C_d\left(\beta\sqrt{\pi}\right)^h \left(\frac{h}{2} - 1\right)\{(D_1 \Delta R + D_2 T_{IT})^h + (D_3 D)^h\} M \right]^{\frac{1}{1-(h/2)}} \right), \quad h > 2$$ 4. Substituting the above equation for μ_D into lognormal distribution model of crack size a_D yields conditional lognormal distribution functions of the crack size a_D given operational stress factors and accumulated missions:

Elements of Agent Learning Process	Type II Macro-Agent
	Disk Crack Size
	$$f(a_{Di}(h, C_d, D_1, D_2, D_3, \sigma_D)\mid M_i, \Delta R_i, T_{ITi}, D_i) = \frac{1}{\sigma_D \cdot a_{Di} \cdot \sqrt{2}} exp\left\{-\frac{1}{2\sigma_D{}^2}\left[Ln(a_{Di}) - Ln\left(\left[a_{initialD}{}^{1-(h/2)} - C_d(\beta\sqrt{\pi})^h\left(\frac{h}{2}-1\right)\{(D_1\Delta R_i + D_2 T_{ITi})^h + (D_3 D)^h\} M_i\right]^{\frac{1}{1-(h/2)}}\right)\right]\right\}, \quad h > 2$$ where $\{M_i, a_{Di}, \Delta R_i, T_{ITi}, D_i\}$, $i = 1, \ldots, E$, are the data points, E is sample size.
Parameters of Agent Output Model - Past Beliefs	Vector of model parameters $\theta\,(h, C_d, D_1, D_2, D_3, \sigma_D)^{-1}$ represents parameters of the PoF model of disk crack size. Parameters h, C_d, D_1, D_2, D_3, σ_D were initially estimated using the generic data produced from a benchmark model and used as prior estimates of the PoF model parameters within Bayesian updating framework (shown in "Model of Agent Output Variable - Updated Beliefs" section of this table).
New Data	The data $\{M_i, a_{Di}, \Delta R_i, T_{ITi}, D_i\}$, $i = 1, \ldots, E$, are obtained from engine bench test of three engines. The data include number of cycles (simulated missions) M_i, applied stresses (amplitude of tangential force on the turbine wheel disks at start ΔR_i, BOT at start T_{ITi}, dwell time before shutdown, D_i) and crack size measurements a_{Di}.
Parameters of Agent Output Model - Updated Beliefs	The updated beliefs about parameters of the agent output model are developed, as follows: 1. The likelihood function is: $$L(a_{Di}, M_i, \Delta R_i, T_{ITi}, D_i\mid\theta) = \prod_{i=1}^{E} f(a_{Di}(h, C_d, D_1, D_2, D_3, \sigma_D)\mid M_i, \Delta R_i, T_{ITi}, D_i)$$ where E is the number of systems studied (i.e. the number of tested engines). It must be noted that this likelihood function applies to the

Elements of Agent Learning Process	Type II Macro-Agent
	Disk Crack Size
	situations when crack size measurements are obtained under unchanging operational stresses ΔR, TIT and D, otherwise cumulative degradation (cumulative exposure) model must be introduced. This cases study is dealing with three engines (E = 3), each operating under the constant stresses ΔR, TIT and D. As such, cumulative degradation modeling will not be applied to disk crack growth. 2. The data $\{M_i, a_{Di}, \Delta R_i, T_{ITi}, D_i\}, i = 1, \ldots, E$ are used as an evidence for Bayesian updating. 3. Bayesian formulation for posteriori probability of the crack size model parameters θ (Internal Parameters Set of the agent): $$\pi(\theta\|Data) = \frac{L(Data\|\theta)\pi_0(\theta)}{\iiint L(Data\|\theta)\pi_0(\theta)d\theta}$$ Upon substitution of the expression of data likelihood function into the above equation will obtain updated form of posterior probability of crack size model parameters θ: $$\pi(\theta\|Data) = \pi(\theta\|a_{Di}, M_i, \Delta R_i, T_{ITi}, D_i) = \\ = \frac{L(a_{Di}, M_i, \Delta R_i, T_{ITi}, D_i\|\theta)\pi_0(\theta)}{\int L(a_{Di}, M_i, \Delta R_i, T_{ITi}, D_i\|\theta)\pi_0(\theta)d\theta}$$ This function provides the updated model parameters of the agent output, crack size a_D, as an outcome of agent learning.

<table>
<tr><th rowspan="2">Elements of Agent Learning Process</th><th>Type II Macro-Agent</th></tr>
<tr><th>Disk Crack Size</th></tr>
<tr><td>Model of Agent Output Variable
- Updated Beliefs</td><td>Same as “Model of Agent Output Variable - Past Beliefs”.
The expected distribution of disk crack size a_D is obtained by integration over all possible values of model parameters θ:
$$\overline{f(a_D|M,\Delta R,T_{IT},D)} = = \int f(a_D(\theta)|M,\Delta R,T_{IT},D)\pi(\theta|Data)d\theta = = \int \left[\frac{1}{\sigma_D \cdot a_{Di} \cdot \sqrt{2}} \times exp\left\{-\frac{1}{2{\sigma_D}^2}\left[Ln(a_{Di}) - Ln\left(\left[{a_{initialD}}^{1-(h/2)} - C_d\left(\beta\sqrt{\pi}\right)^h\left(\frac{h}{2}-1\right)\{(D_1\Delta R_i + D_2 T_{ITi})^h + (D_3 D)^h\}M_i\right]^{\frac{1}{1-(h/2)}}\right)\right]\right\} \times \pi(h,C_d,D_1,D_2,D_3,\sigma_D|Data)\right] dhdC_d dD_1 dD_2 dD_3 d\sigma_D\,,$$
$h > 2$
WinBUGS/OpenBUGS program [120], [121] is used for Bayesian updating according to the above formulations.</td></tr>
</table>

8.7.2. Autonomy Property of Agents

Due to the nature of agents in this case study, the activation/deactivation property is only assigned to the independent Type I Micro-Agents, specifically agents ID# 1 to 7 in Table 8-1. All Type II Macro-Agents, summarized in Table 8-3, represent damage accumulation related characteristics of four critical components of a gas turbine engine, and will remain in the active status in order to serve the purpose of the probabilistic reliability modeling. The Importance Index method of local sensitivity analysis is chosen for the modeling of agent activation and deactivation capability of the specified Type I Micro-Agents because engineering knowledge of the studied components suggests that there are no interactions between the output variables of these agents. Nonlinearity of the underlying PoF models, however, suggests that Sobol's method of global sensitivity analysis should be used if a higher accuracy of estimation is required. Since this case study is only intended to demonstrate the autonomy property, the Importance Index method is considered sufficient and preferred due to its simplicity.

8.7.3. Computational Resources

Several software programs (WinBUGS/OpenBUGS [120], [121], Oracle Crystal Ball [122], Minitab [73]) were used in combination to perform Bayesian inference, simulation, classical distribution fitting and uncertainty importance (sensitivity) analysis for the agent leaning and autonomy execution. Each step was programed in a different software program and the analysis results were manually transferred into another program to perform the next step of system modeling. As such, the case study was very time-consuming and computationally intense despite of the small size of the system consisting of four hardware components. A computer program would have to be developed to allow the definition of agent autonomy and execution of agent-oriented system reliability model within a single interface.

8.8. Agent-Oriented Probabilistic PoF Model of System Reliability

This section provides a summary of the results of the case study. It must be noted that a major part of the available design characteristics, material properties, operating conditions, and degradation parameters of the gas turbine components could not be published due to the confidentiality agreement with the sponsor companies. As such, the numerical values shown below are given for presentation only and may be different from their actual values.

The results of the learning process of Type I Micro-Agents based on operational data are shown in Table 8-6. Operational (flight test) data were collected three times to develop and further update the uncertainty distributions of the seven Type I Micro-Agents. A total of three updates occurred.

Table 8-6: Type I Micro-Agents - Operational Data and Learning Process

Type I Micro-Agents									
ID#	Name of Agent Output Variable	Letter ID	Public Attributes						
			Past Believes	New Data	Updated Believes (1st Update)	New Data	Updated Believes (2nd Update)	New Data	Updated Believes (3rd Update)
1	**Tangential Force on the Turbine Wheel Disks**	$\boldsymbol{R}$	Probability distribution **π(R) = Normal (Mean = 90 N, St.Dev. = 5 N)**	Operational Data: **{R_i}, i = 1, ..., 6**	Probability distribution **π(R) = Normal (Mean = 95 N, St.Dev. = 5 N)**	Operational Data: **{R_i}, i = 1, ..., 6**	Probability distribution **π(R) = Normal (Mean = 90 N, St.Dev. = 3 N)**	Operational Data: **{R_i}, i = 1, ..., 6**	Probability distribution **π(R) = Normal (Mean = 85 N, St.Dev. = 6 N)**
2	**Amplitude of Tangential Force on the Turbine Wheel Disks at Start**	ΔR	Probability distribution **π(ΔR) = Normal (Mean = 220 N, St.Dev. = 15 N)**	Operational Data: **{ΔR_i}, i = 1, ..., 6**	Probability distribution **π(ΔR) = Normal (Mean = 210 N, St.Dev. = 20 N)**	Operational Data: **{ΔR_i}, i = 1, ..., 6**	Probability distribution **π(ΔR) = Normal (Mean = 215 N, St.Dev. = 15 N)**	Operational Data: **{ΔR_i}, i = 1, ..., 6**	Probability distribution **π(ΔR) = Normal (Mean = 230 N, St.Dev. = 10 N)**
3	**Random Vibration**	G_{rms}	Probability distribution **π(G_{rms}) = Normal (Mean = 6 G_{rms}, St.Dev. = 1 G_{rms})**	Operational Data: **{G_{rmsi}}, i = 1, ..., 6**	Probability distribution **π(G_{rms}) = Normal (Mean = 5.8 G_{rms}, St.Dev. = 1.1 G_{rms})**	Operational Data: **{G_{rmsi}}, i = 1, ..., 6**	Probability distribution **π(G_{rms}) = Normal (Mean = 6.1 G_{rms}, St.Dev. = 1.3 G_{rms})**	Operational Data: **{G_{rmsi}}, i = 1, ..., 6**	Probability distribution **π(G_{rms}) = Normal (Mean = 6 G_{rms}, St.Dev. = 0.8 G_{rms})**
4	**BOT at start**	T_{IT}	Probability distribution **π(T_{IT}) = Normal (Mean = 1025°C, St.Dev. = 110°C)**	Operational Data: **{T_{ITi}}, i = 1, ..., 3**	Probability distribution **π(T_{IT}) = Normal (Mean = 990°C, St.Dev. = 75°C)**	Operational Data: **{T_{ITi}}, i = 1, ..., 3**	Probability distribution **π(T_{IT}) = Normal (Mean = 980°C, St.Dev. = 50°C)**	Operational Data: **{T_{ITi}}, i = 1, ..., 3**	Probability distribution **π(T_{IT}) = Normal (Mean = 970°C, St.Dev. = 55°C)**
5	**Dwell Time before Shutdown**	D	Probability distribution **π(D) = Lognormal (Mean = 150 sec., St.Dev. = 50 sec.)**	Operational Data: **{D_i}, i = 1, ..., 3**	Probability distribution **π(D) = Lognormal (Mean = 145 sec., St.Dev. = 60 sec.)**	Operational Data: **{D_i}, i = 1, ..., 3**	Probability distribution **π(D) = Lognormal (Mean = 160 sec., St.Dev. = 45 sec.)**	Operational Data: **{D_i}, i = 1, ..., 3**	Probability distribution **π(D) = Lognormal (Mean = 155 sec., St.Dev. = 40 sec.)**
6	**Initial Crack Size (depth), Shaft**	$a_{initialS}$	Probability distribution **π($a_{initialS}$) = Weibull (Mean = 0.014 in, St.Dev. = 0.0025 in)**	Not available	Probability distribution **π($a_{initialS}$) = Weibull (Mean = 0.014 in, St.Dev. = 0.0025 in)**	Not available	Probability distribution **π($a_{initialS}$) = Weibull (Mean = 0.014 in, St.Dev. = 0.0025 in)**	Not available	Probability distribution **π($a_{initialS}$) = Weibull (Mean = 0.014 in, St.Dev. = 0.0025 in)**
7	**Initial Crack Size (depth), Disk**	$a_{initialD}$	Probability distribution **π($a_{initialD}$) = Weibull (Mean = 0.01 in, St.Dev. = 0.003 in)**	Not available	Probability distribution **π($a_{initialD}$) = Weibull (Mean = 0.01 in, St.Dev. = 0.003 in)**	Not available	Probability distribution **π($a_{initialD}$) = Weibull (Mean = 0.01 in, St.Dev. = 0.003 in)**	Not available	Probability distribution **π($a_{initialD}$) = Weibull (Mean = 0.01 in, St.Dev. = 0.003 in)**

The data for the learning process of Type II Macro-Agents have been obtained from the engine bench test. The mission (engine cycle) represents starting a cold engine, taking-off, flying to a destination, landing, and shutting down. The following tables (Table 8-7 to Table 8-10) contain extracts from the calculation file, demonstrating the learning process of Type II Macro-Agents based on the data collected during the bench test of three prototype engines. The test duration was 2000 engine cycles, each simulating a mission cycle under the defined conditions. Seven data sets were obtained from the test, each containing the records of input variables (from input agents) and the respective measurements of the agent output variable, collected at 500 engine cycles, 1000 engine cycles, 1200 engine cycles, 1400 engine cycles, 1600 engine cycles, 1800 engine cycles and 2000 engine cycles. Bayesian inference was used to aggregate the available data into the updated beliefs about the PoF model parameters of the agent output variable. A total of seven updates occurred, the first one at 500 missions and the seventh at 2000 missions. For example, new data set 5 (at 1600 engine cycles) for Type II Macro-Agent ID #19, shaft crack size, contains the following data (Table 8-9):

Engine #1: ΔR = 230 N, G_{rms} = 8 Grms, a_S = 0.043 in, S_p = 0.015 in^2

Engine #2: ΔR = 250 N, G_{rms} = 10 Grms, a_S = 0.035 in, S_p = 0.024 in^2

Engine #3: ΔR = 270 N, G_{rms} = 14 Grms, a_S = 0.030 in, S_p = 0.031 in^2

Here ΔR, G_{rms} and S_p represent the input agents, a_S is agent output variable (shaft crack size). Bayesian inference formalism, similar to the model shown in Table 8-5, was used to obtain the updated beliefs about the parameters, S_1, S_2 and S_3, of the PoF model of the shaft crack size given by Equation 8-8. It must be noted that Type II

Macro-Agent ID #20, shaft life to crack size at failure, and Type II Macro-Agent ID #19, shaft crack size, share the same parameters, S_1, S_2 and S_3, because the PoF model of the agent ID #20 is another form of the PoF model of the agent ID #19. As such, the agent ID #20, shaft life to crack size at failure, learns simultaneously with the agent ID #19, shaft crack size, however, the resulting probability distributions of the two agents are different.

Table 8-11 contains an extract from the calculation file, demonstrating the learning process of Type III Monitoring Agents from the updated Type II Macro-Agents. The functional form of the agent output model remained unchanged for the all agents in this study. No special rules of behavior have been set for any agents.

Table 8-7: Type II Macro-Agents ID #12 and ID #13 - Test Data and the Learning Process

Type II Macro-Agents						
ID#	Name of Agent Output Variable	Letter ID	Public Attributes			
			Past Believes	New Data Set 1 ($j = 1$)	Updated Believes (1st Update)	New Data Set 2 ($j = 2$)
12	**Bearing Life to Spall Initiation (Bearing #1)**	$\boldsymbol{L_1}$	Probability distribution for each tested engine: **$\pi(L_1)$ = Weibull (β_1, L_1)** $L_{1i} = B_{1(1)} R_i^{\,k} \left(B_{2(1)} R_i + B_{3(1)} \right)^{-10/3}$ where i = 1, ..., E, E is sample size, E = 3. $B_{1(1)}$ = Uniform (1E+10, 1E+20) $B_{2(1)}$ = Uniform (0, 1) $B_{3(1)}$ = Uniform (0, 1E+04) **k** = -1 β_1 = Uniform (1, 8)	Test Data: Accumulated missions M_j, missions to spall initiation (left censored observations L_{1i}, right censored observations t_{1i}), conditional on Input Set variable R_i, $\{M_j, L_{1i}, t_{1i}, R_i\}$, $i = 1, ..., E$, where E is sample size, E = 3. R_1 = 110 N, R_2 = 130 N, R_3 = 150 N, $t_{11} = t_{12} = t_{13}$ = 500, $M_j = M_1$ = **500**	Probability distribution for each tested engine: **$\pi(L_1)$ = Weibull (β_1, L_1)** $L_{1i} = B_{1(1)} R_i^{\,k} \left(B_{2(1)} R_i + B_{3(1)} \right)^{-10/3}$ where i = 1, ..., E, E is sample size, E = 3. $B_{1(1)}$ = {Mean = 3.20E+16 , St.Dev. = 3.52E+15} $B_{2(1)}$ = {Mean = 0.13 , St.Dev. = 0.02} $B_{3(1)}$ = {Mean = 2.45E+03 , St.Dev. = 8.62E+01} **k** = -1 β_1 = {Mean = 5.3, St.Dev. = 1.0}	Test Data: Accumulated missions M_j, missions to spall initiation (interval censored observations L_{1i}, right censored observations t_{1i}), conditional on Input Set variable R_i, $\{M_j, L_{1i}, t_{1i}, R_i\}$, $i = 1, ..., E$, where E is sample size, E = 3. R_1 = 110 N, R_2 = 130 N, R_3 = 150 N, $t_{11} = t_{12} = t_{13}$ = 1000, $M_j = M_2$ = **1000**
13	**Bearing Life to Spall Initiation (Bearing #2)**	$\boldsymbol{L_2}$	Probability distribution for each tested engine: **$\pi(L_2)$ = Weibull (β_2, L_2)** $L_{2i} = B_{1(2)} R_i^{\,k} \left(B_{2(2)} R_i + B_{3(2)} \right)^{-10/3}$ where i = 1, ..., E, E is sample size, E = 3. $B_{1(2)}$ = Uniform (1E+10, 1E+20) $B_{2(2)}$ = Uniform (0, 1) $B_{3(2)}$ = Uniform (0, 1E+04) **k** = -1 β_2 = Uniform (1, 8)	Test Data: Accumulated missions M_j, missions to spall initiation (left censored observations L_{2i}, right censored observations t_{2i}), conditional on Input Set variable R_i, $\{M_j, L_{2i}, t_{2i}, R_i\}$, $i = 1, ..., E$, where E is sample size, E = 3. R_1 = 110 N, R_2 = 130 N, R_3 = 150 N, $t_{21} = t_{22} = t_{23}$ = 500, $M_j = M_1$ = **500**	Probability distribution for each tested engine: **$\pi(L_2)$ = Weibull (β_2, L_2)** $L_{2i} = B_{1(2)} R_i^{\,k} \left(B_{2(2)} R_i + B_{3(2)} \right)^{-10/3}$ where i = 1, ..., E, E is sample size, E = 3. $B_{1(2)}$ = {Mean = 1.59E+16 , St.Dev. = 2.82E+15} $B_{2(2)}$ = {Mean = 0.21 , St.Dev. = 0.03} $B_{3(2)}$ = {Mean = 2.04E+03 , St.Dev. = 6.46E+02} **k** = -1 β_2 = {Mean = 5.5, St.Dev. = 1.0}	Test Data: Accumulated missions M_j, missions to spall initiation (interval censored observations L_{2i}, right censored observations t_{2i}), conditional on Input Set variable R_i, $\{M_j, L_{2i}, t_{2i}, R_i\}$, $i = 1, ..., E$, where E is sample size, E = 3. R_1 = 110 N, R_2 = 130 N, R_3 = 150 N, $t_{21} = t_{22} = t_{23}$ = 1000, $M_j = M_2$ = **1000**

Table 8-8: Type II Macro-Agents ID #14 and ID #15 - Test Data and the Learning Process

Type II Macro-Agents						
ID#	Name of Agent Output Variable	Letter ID	Public Attributes			
			Past Believes	New Data Set 4 ($j = 4$)	Updated Believes (1st Update)	New Data Set 5 ($j = 5$)
14	**Bearing Spall Size (Bearing #1)**	$\boldsymbol{S_{p1}}$	Probability distribution $\boldsymbol{\pi(S_{p1}) = LN(S_{p1}, \sigma_{B1})}$ $S_{p1ij} = \left[B_{4(1)}\left(B_{5(1)}R_i + B_{6(1)}\right)^m \left(1-\frac{m}{2}\right) \times \left(M_j - M_{j-1}\right) + \left(S_{p1i(j-1)}\right)^{1-\frac{m}{2}} \right]^{\frac{1}{1-\frac{m}{2}}},$ $m \neq 2$ where i = 1, ..., E, E is sample size, E = 3. $B_{4(1)}$ = Uniform (1E+02, 1E+10) $B_{5(1)}$ = Uniform (0, 1E+03) $B_{6(1)}$ = Uniform (0, 1E+03) **m** = 0.7 $\boldsymbol{\sigma_{B1}}$ = Uniform (0, 1)	Test Data: Missions M_j to spall propagation to size S_{p1ij} from size $S_{p1i(j-1)}$ at previous check at M_{j-1} missions, conditional on Input Set variable R, $\{M_j, R_i, S_{p1ij}, S_{p1i(j-1)}\}$, $i = 1, ..., E$, where E is sample size, E = 3. R_1 = 110 N, R_2 = 130 N, R_3 = 150 N, $M_j = M_4 =$ **1400**, $M_{j-1} = M_3 = 1200$ $S_{p11j} = S_{p114}$ = 0.006 in^2 $S_{p11(j-1)} = S_{p113}$ is not applicable, $S_{p12j} = S_{p124}$ = 0.008 in^2 $S_{p12(j-1)} = S_{p123}$ is not applicable, $S_{p13j} = S_{p134}$ = 0.016 in^2 $S_{p13(j-1)} = S_{p133}$ = 0.004 in^2	Probability distribution $\boldsymbol{\pi(S_{p1}) = LN(S_{p1}, \sigma_{B1})}$ $S_{p1ij} = \left[B_{4(1)}\left(B_{5(1)}R_i + B_{6(1)}\right)^m \left(1-\frac{m}{2}\right) \times \left(M_j - M_{j-1}\right) + \left(S_{p1i(j-1)}\right)^{1-\frac{m}{2}} \right]^{\frac{1}{1-\frac{m}{2}}},$ $m \neq 2$ where i = 1, ..., E, E is sample size, E = 3. $B_{4(1)}$ = {Mean = 4.20E-07 , St.Dev. = 3.60E-08} $B_{5(1)}$ = {Mean = 69.50, St.Dev. = 5.5} $B_{6(1)}$ = {Mean = 0.01, St.Dev. = 0.02} **m** = 0.7 $\boldsymbol{\sigma_{B1}}$ = {Mean = 0.002, St.Dev. = 0.0006}	Test Data: Missions M_j to spall propagation to size S_{p1ij} from size $S_{p1i(j-1)}$ at previous check at M_{j-1} missions, conditional on Input Set variable R, $\{M_j, R_i, S_{p1ij}, S_{p1i(j-1)}\}$, $i = 1, ..., E$, where E is sample size, E = 3. R_1 = 110 N, R_2 = 130 N, R_3 = 150 N, $M_j = M_5 =$ **1600**, $M_{j-1} = M_4 = 1400$ $S_{p11j} = S_{p115}$ = 0.015 in^2 $S_{p11(j-1)} = S_{p114}$ = 0.006 in^2 $S_{p12j} = S_{p125}$ = 0.018 in^2 $S_{p12(j-1)} = S_{p124}$ = 0.008 in^2 $S_{p13j} = S_{p135}$ = 0.031 in^2 $S_{p13(j-1)} = S_{p134}$ = 0.016 in^2
15	**Bearing Spall Size (Bearing #2)**	$\boldsymbol{S_{p2}}$	Probability distribution $\boldsymbol{\pi(S_{p2}) = LN(S_{p2}, \sigma_{B2})}$ $S_{p2ij} = \left[B_{4(2)}\left(B_{5(2)}R_i + B_{6(2)}\right)^m \left(1-\frac{m}{2}\right) \times \left(M_j - M_{j-1}\right) + \left(S_{p2i(j-1)}\right)^{1-\frac{m}{2}} \right]^{\frac{1}{1-\frac{m}{2}}},$ $m \neq 2$ where i = 1, ..., E, E is sample size, E = 3. $B_{4(2)}$ = Uniform (1E+02, 1E+10) $B_{5(2)}$ = Uniform (0, 1E+03) $B_{6(2)}$ = Uniform (0, 1E+03) **m** = 0.7 $\boldsymbol{\sigma_{B2}}$ = Uniform (0, 1)	Test Data: Missions M_j to spall propagation to size S_{p2ij} from size $S_{p2i(j-1)}$ at previous check at M_{j-1} missions, conditional on Input Set variable R, $\{M_j, R_i, S_{p2ij}, S_{p2i(j-1)}\}$, $i = 1, ..., E$, where E is sample size, E = 3. R_1 = 110 N, R_2 = 130 N, R_3 = 150 N, $M_j = M_4 =$ **1400**, $M_{j-1} = M_3 = 1200$ $S_{p21j} = S_{p214}$ = 0.007 in^2 $S_{p21(j-1)} = S_{p213}$ is not applicable, $S_{p22j} = S_{p224}$ = 0.011 in^2 $S_{p22(j-1)} = S_{p223}$ = 0.003 in^2 $S_{p23j} = S_{p234}$ = 0.015 in^2 $S_{p23(j-1)} = S_{p233}$ = 0.005 in^2	Probability distribution $\boldsymbol{\pi(S_{p2}) = LN(S_{p2}, \sigma_{B2})}$ $S_{p2ij} = \left[B_{4(2)}\left(B_{5(2)}R_i + B_{6(2)}\right)^m \left(1-\frac{m}{2}\right) \times \left(M_j - M_{j-1}\right) + \left(S_{p2i(j-1)}\right)^{1-\frac{m}{2}} \right]^{\frac{1}{1-\frac{m}{2}}},$ $m \neq 2$ where i = 1, ..., E, E is sample size, E = 3. $B_{4(2)}$ = {Mean = 1.70E-06 , St.Dev. = 2.50E-07} $B_{5(2)}$ = {Mean = 5.50, St.Dev. = 1.5} $B_{6(2)}$ = {Mean = 250.70, St.Dev. = 50.60} **m** = 0.7 $\boldsymbol{\sigma_{B2}}$ = {Mean = 0.016, St.Dev. = 0.0009}	Test Data: Missions M_j to spall propagation to size S_{p2ij} from size $S_{p2i(j-1)}$ at previous check at M_{j-1} missions, conditional on Input Set variable R, $\{M_j, R_i, S_{p2ij}, S_{p2i(j-1)}\}$, $i = 1, ..., E$, where E is sample size, E = 3. R_1 = 110 N, R_2 = 130 N, R_3 = 150 N, $M_j = M_5 =$ **1600**, $M_{j-1} = M_4 = 1400$ $S_{p21j} = S_{p215}$ = 0.014 in^2 $S_{p21(j-1)} = S_{p214}$ = 0.007 in^2 $S_{p22j} = S_{p225}$ = 0.024 in^2 $S_{p22(j-1)} = S_{p224}$ = 0.011 in^2 $S_{p23j} = S_{p235}$ = 0.030 in^2 $S_{p23(j-1)} = S_{p234}$ = 0.015 in^2

Table 8-9: Type II Macro-Agent ID #19 - Test Data and Learning Process

Type II Macro-Agents						
ID#	Name of Agent Output Variable	Letter ID	Public Attributes			
			Updated Believes (4th Update)	New Data Set 5 ($j = 5$)	Updated Believes (5th Update)	New Data Set 6 ($j = 6$)
19	**Shaft Crack Size (cont.)**	a_S	Probability distribution $\boldsymbol{\pi(a_S) = LN(a_S,\ \sigma_S)}$	Accumulated missions: $M_j = M_5 =$ **1600**	Probability distribution $\boldsymbol{\pi(a_S) = LN(a_S,\ \sigma_S)}$	Accumulated missions: $M_j = M_6 =$ **1800**
			$a_{Sij} = \left[{a_{Si(j-1)}}^{1-n/2} - C_s \left[\pi \left\{ S_1 (\Delta R_i)^2 + S_2 \left[\Gamma\left(1 + \frac{n}{2}\right)\right]^{\frac{2}{n}} ({G_{rms}}_i)^2 \right. \right. \right.$ where $i = 1, ..., E$, E is sample size, $E = 3$.		$a_{Sij} = \left[{a_{Si(j-1)}}^{1-n/2} - C_s \left[\pi \left\{ S_1 (\Delta R_i)^2 + S_2 \left[\Gamma\left(1 + \frac{n}{2}\right)\right]^{\frac{2}{n}} ({G_{rms}}_i)^2 \right. \right. \right.$ where $i = 1, ..., E$, E is sample size, $E = 3$.	
			S_1 = {Mean = 0.014, St.Dev. = 0.03} S_2 = {Mean = 18.2, St.Dev. = 3.7} S_3 = {Mean = 4.3E+05, St.Dev. = 1.9E+5} **n** = 3.2 $\mathbf{C_S}$ = 1.65E-09 (in/cycle)/(Mpa(in)$^{1/2}$)^n $\boldsymbol{\sigma_S}$ = {Mean = 0.0005, St.Dev. = 0.0003}	Test Data: Missions M_j to crack propagation to size a_{Sij} from size $a_{Si(j-1)}$ at previous check at M_{j-1} missions, conditional on Input Set variables ΔR, G_{rms}, S_p: {M_j, ΔR_i, G_{rmsi}, a_{Sij}, $a_{Si(j-1)}$, S_{pij} = Max (S_{p1ij}, S_{p2ij}), $S_{pi(j-1)}$ = Max ($S_{p1i(j-1)}$, $S_{p2i(j-1)}$)}, $i = 1, ..., E$, where E is sample size, $E = 3$. $M_j = M_5 =$ **1600**, $M_{j-1} = M_4 = 1400$ ΔR_1 = 230 N, ΔR_2 = 250 N, ΔR_3 = 270 N G_{rms1} = 8, G_{rms2} = 10, G_{rms3} = 14 $a_{S1j} = a_{S15}$ = 0.043 in $a_{S2j} = a_{S25}$ = 0.035 in $a_{S3j} = a_{S35}$ = 0.030 in $a_{S1(j-1)} = a_{S14}$ = 0.040 in $a_{S2(j-1)} = a_{S24}$ = 0.031 in $a_{S3(j-1)} = a_{S34}$ = 0.022 in $S_{p1j} = S_{p15}$ = 0.015 in^2 $S_{p2j} = S_{p25}$ = 0.024 in^2 $S_{p3j} = S_{p35}$ = 0.031 in^2 $S_{p1(j-1)} = S_{p14}$ = 0.007 in^2 $S_{p2(j-1)} = S_{p24}$ = 0.011 in^2 $S_{p3(j-1)} = S_{p34}$ = 0.016 in^2	S_1 = {Mean = 0.012, St.Dev. = 0.02} S_2 = {Mean = 16.6, St.Dev. = 3.3} S_3 = {Mean = 2.9E+05, St.Dev. = 1.0E+5} **n** = 3.2 $\mathbf{C_S}$ = 1.65E-09 (in/cycle)/(Mpa(in)$^{1/2}$)^n $\boldsymbol{\sigma_S}$ = {Mean = 0.0009, St.Dev. = 0.0001}	Test Data: Missions M_j to crack propagation to size a_{Sij} from size $a_{Si(j-1)}$ at previous check at M_{j-1} missions, conditional on Input Set variables ΔR, G_{rms}, S_p: {M_j, ΔR_i, G_{rmsi}, a_{Sij}, $a_{Si(j-1)}$, S_{pij} = Max (S_{p1ij}, S_{p2ij}), $S_{pi(j-1)}$ = Max ($S_{p1i(j-1)}$, $S_{p2i(j-1)}$)}, $i = 1, ..., E$, where E is sample size, $E = 3$. $M_j = M_6 =$ **1800**, $M_{j-1} = M_5 = 1600$ ΔR_1 = 230 N, ΔR_2 = 250 N, ΔR_3 = 270 N G_{rms1} = 8, G_{rms2} = 10, G_{rms3} = 14 $a_{S1j} = a_{S16}$ = 0.046 in $a_{S2j} = a_{S26}$ = 0.038 in $a_{S3j} = a_{S36}$ = 0.035 in $a_{S1(j-1)} = a_{S15}$ = 0.043 in $a_{S2(j-1)} = a_{S25}$ = 0.035 in $a_{S3(j-1)} = a_{S35}$ = 0.030 in $S_{p1j} = S_{p16}$ = 0.031 $S_{p2j} = S_{p26}$ = 0.045 $S_{p3j} = S_{p36}$ = 0.056 $S_{p1(j-1)} = S_{p15}$ = 0.015 in^2 $S_{p2(j-1)} = S_{p25}$ = 0.024 in^2 $S_{p3(j-1)} = S_{p35}$ = 0.031 in^2

Table 8-10: Type II Macro-Agent ID #21 - Test Data and Learning Process

Type II Macro-Agents						
ID#	Name of Agent Output Variable	Letter ID	Public Attributes			
			Updated Believes (4th Update)	New Data Set 5 ($j = 5$)	Updated Believes (5th Update)	New Data Set 6 ($j = 6$)
21	**Disk Crack Size (cont.)**	$\boldsymbol{a_D}$	Probability distribution $\boldsymbol{\pi(a_D)} =$ $\boldsymbol{LN(\mu_D, \sigma_D)}$	Accumulated missions: $M_j = M_5 =$ **1600**	Probability distribution $\boldsymbol{\pi(a_D)} =$ $\boldsymbol{LN(\mu_D, \sigma_D)}$	Accumulated missions: $M_j = M_6 =$ **1800**
			$a_{Dij} = \left[a_{Di(j-1)}{}^{1-h/2} - C_d(\beta\sqrt{\pi})^h\left(\frac{h}{2}-1\right)\left\{(D_1\Delta R_i + D_2T_{ITi})^h + \left(D_3D_i{}^{-1}\right)^h\right\}(M_j - M_{j-1})\right]^{\frac{1}{1-h/2}},$ $h > 2$ where $i = 1, \ldots, E$, E is sample size, $E = 3$.		$a_{Dij} = \left[a_{Di(j-1)}{}^{1-h/2} - C_d(\beta\sqrt{\pi})^h\left(\frac{h}{2}-1\right)\left\{(D_1\Delta R_i + D_2T_{ITi})^h + \left(D_3D_i{}^{-1}\right)^h\right\}(M_j - M_{j-1})\right]^{\frac{1}{1-h/2}},$ $h > 2$ where $i = 1, \ldots, E$, E is sample size, $E = 3$.	
			D_1 = {Mean = 0.48, St.Dev. = 0.006} D_2 = {Mean = 0.09, St.Dev. = 0.04} D_3 = {Mean = 1.3E+04, St. Dev. = 6.4E+03} **h** = 2.5 $\mathbf{C_d}$ = 2.4E-10 (in/cycle)/(Mpa(in)$^{1/2}$)^n **β** = 1.12 $\boldsymbol{\sigma_S}$ = {Mean = 0.0025, St.Dev. = 0.001}	Test Data: Missions M_j to crack propagation to size a_{Dij} from size $a_{Di(j-1)}$ at previous check at M_{j-1} missions, conditional on Input Set variables ΔR, T_{IT}, D: $\{M_j, a_{Dij}, a_{Di(j-1)}, \Delta R_i, T_{ITi}, D_i\}$, $i = 1, \ldots, E$, where E is sample size, $E = 3$. ΔR_1 = 230 N, ΔR_2 = 250 N ΔR_3 = 270 N T_{IT1} = 1200 C, T_{IT2} = 1250C T_{IT3} = 1300 C D_1 = 180 s, D_2 = 120 s, D_3 = 60 s $a_{D1j} = a_{D15}$ = 0.091 in $a_{D2j} = a_{D25}$ = 0.068 in $a_{D3j} = a_{D35}$ = 0.043 in $a_{D1(j-1)} = a_{D14}$ = 0.078 in $a_{D2(j-1)} = a_{D24}$ = 0.057 in $a_{D3(j-1)} = a_{D34}$ = 0.035 in	D_1 = {Mean = 0.65, St.Dev. = 0.006} D_2 = {Mean = 0.12, St.Dev. = 0.01} D_3 = {Mean = 1.4E+04, St. Dev. = 3.2E+03} **h** = 2.5 $\mathbf{C_d}$ = 2.4E-10 (in/cycle)/(Mpa(in)$^{1/2}$)^n **β** = 1.12 $\boldsymbol{\sigma_S}$ = {Mean = 0.0018, St.Dev. = 0.0005}	Test Data: Missions M_j to crack propagation to size a_{Dij} from size $a_{Di(j-1)}$ at previous check at M_{j-1} missions, conditional on Input Set variables ΔR, T_{IT}, D: $\{M_j, a_{Dij}, a_{Di(j-1)}, \Delta R_i, T_{ITi}, D_i\}$, $i = 1, \ldots, E$, where E is sample size, $E = 3$. ΔR_1 = 230 N, ΔR_2 = 250 N ΔR_3 = 270 N T_{IT1} = 1200 C, T_{IT2} = 1250C T_{IT3} = 1300 C D_1 = 180 s, D_2 = 120 s, D_3 = 60 s $a_{D1j} = a_{D16}$ = 0.107 in $a_{D2j} = a_{D26}$ = 0.082 in $a_{D3j} = a_{D36}$ = 0.054 in $a_{D1(j-1)} = a_{D15}$ = 0.091 in $a_{D2(j-1)} = a_{D25}$ = 0.068 in $a_{D3(j-1)} = a_{D35}$ = 0.043 in

Table 8-11: Type III Monitoring Agents ID #23 to #25 - Learning Process

ID#	Name of Agent Output Variable	Letter ID	Past Believes	Updated Believes (1st Update)	Updated Believes (2nd Update)	Updated Believes (3rd Update)
23	**Remaining Useful Life of Bearing #1**	**RUL_{B1}**	Probability distribution $\pi(RUL_{B1})$ is obtained for each tested engine by simulation via PoF model for RUL_{B1} according to the "Past Believes" for agents L_1 (ID# 12), N_{B1} (ID# 17) and accumulated missions M_j: $RUL_{B1i} = L_{1i} + N_{B1i} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_0 = \mathbf{0}$	Probability distribution $\pi(RUL_{B1})$ is obtained for each tested engine by simulation via PoF model for RUL_{B1} according to the "Updated Believes (1st Update)" for agents L_1 (ID# 12), N_{B1} (ID# 17) and accumulated missions M_j: $RUL_{B1i} = L_{1i} + N_{B1i} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_1 = \mathbf{500}$	Probability distribution $\pi(RUL_{B1})$ is obtained for each tested engine by simulation via PoF model for RUL_{B1} according to the "Updated Believes (2nd Update)" for agents L_1 (ID# 12), N_{B1} (ID# 17) and accumulated missions M_j: $RUL_{B1i} = L_{1i} + N_{B1i} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_2 = \mathbf{1000}$	Probability distribution $\pi(RUL_{B1})$ is obtained for each tested engine by simulation via PoF model for RUL_{B1} according to the "Updated Believes (3rd Update)" for agents L_1 (ID# 12), N_{B1} (ID# 17) and accumulated missions M_j: $RUL_{B1i} = L_{1i} + N_{B1i} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_3 = \mathbf{1200}$
24	**Remaining Useful Life of Bearing #2**	**RUL_{B2}**	Probability distribution $\pi(RUL_{B2})$ is obtained for each tested engine by simulation via PoF model for RUL_{B2} according to the "Past Believes" for agents L_2 (ID# 13), N_{B2} (ID# 18) and accumulated missions M_j: $RUL_{B2i} = L_{2i} + N_{B2i} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_0 = \mathbf{0}$	Probability distribution $\pi(RUL_{B2})$ is obtained for each tested engine by simulation via PoF model for RUL_{B2} according to the "Updated Believes (1st Update)" for agents L_2 (ID# 13), N_{B2} (ID# 18) and accumulated missions M_j: $RUL_{B2i} = L_{2i} + N_{B2i} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_1 = \mathbf{500}$	Probability distribution $\pi(RUL_{B2})$ is obtained for each tested engine by simulation via PoF model for RUL_{B2} according to the "Updated Believes (2nd Update)" for agents L_2 (ID# 13), N_{B2} (ID# 18) and accumulated missions M_j: $RUL_{B2i} = L_{2i} + N_{B2i} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_2 = \mathbf{1000}$	Probability distribution $\pi(RUL_{B2})$ is obtained for each tested engine by simulation via PoF model for RUL_{B2} according to the "Updated Believes (3rd Update)" for agents L_2 (ID# 13), N_{B2} (ID# 18) and accumulated missions M_j: $RUL_{B2i} = L_{2i} + N_{B2i} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_3 = \mathbf{1200}$
25	**Remaining Useful Life of the Shaft**	**RUL_S**	Probability distribution $\pi(RUL_S)$ is obtained for each tested engine by simulation via PoF model for RUL_S according to the "Past Believes" for agent N_S (ID# 20) and accumulated missions M_j: $RUL_{Si} = N_{Si} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_0 = \mathbf{0}$	Probability distribution $\pi(RUL_S)$ is obtained for each tested engine by simulation via PoF model for RUL_S according to the "Updated Believes (1st Update)" for agent N_S (ID# 20) and accumulated missions M_j: $RUL_{Si} = N_{Si} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_1 = \mathbf{500}$	Probability distribution $\pi(RUL_S)$ is obtained for each tested engine by simulation via PoF model for RUL_S according to the "Updated Believes (2nd Update)" for agent N_S (ID# 20) and accumulated missions M_j: $RUL_{Si} = N_{Si} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_2 = \mathbf{1000}$	Probability distribution $\pi(RUL_S)$ is obtained for each tested engine by simulation via PoF model for RUL_S according to the "Updated Believes (3rd Update)" for agent N_S (ID# 20) and accumulated missions M_j: $RUL_{Si} = N_{Si} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_3 = \mathbf{1200}$
26	**Remaining Useful Life of the Disk**	**RUL_D**	Probability distribution $\pi(RUL_D)$ is obtained for each tested engine by simulation via PoF model for RUL_D according to the "Past Believes" for agent N_D (ID# 22) and accumulated missions M_j: $RUL_{Di} = N_{Di} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_0 = \mathbf{0}$	Probability distribution $\pi(RUL_D)$ is obtained for each tested engine by simulation via PoF model for RUL_D according to the "Updated Believes (1st Update)" for agent N_D (ID# 20) and accumulated missions M_j: $RUL_{Di} = N_{Di} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_1 = \mathbf{500}$	Probability distribution $\pi(RUL_D)$ is obtained for each tested engine by simulation via PoF model for RUL_D according to the "Updated Believes (2nd Update)" for agent N_D (ID# 20) and accumulated missions M_j: $RUL_{Di} = N_{Di} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_2 = \mathbf{1000}$	Probability distribution $\pi(RUL_D)$ is obtained for each tested engine by simulation via PoF model for RUL_D according to the "Updated Believes (3rd Update)" for agent N_D (ID# 20) and accumulated missions M_j: $RUL_{Di} = N_{Di} - M_j$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_3 = \mathbf{1200}$

ID#	Name of Agent Output Variable	Letter ID	Past Believes	Updated Believes (1st Update)	Updated Believes (2nd Update)	Updated Believes (3rd Update)
27	**Remaining Useful Life of the System**	**RUL**	Probability distribution $\pi(RUL)$ is obtained for each tested engine by simulation via PoF model for RUL according to the "Past Believes" for agents RUL_{B1} (ID# 23), RUL_{B2} (ID# 24), RUL_S (ID# 25), RUL_D (ID# 26) and accumulated missions M_j: $RUL = Minimum\ \{RUL_{B1i} + RUL_{B2i} + RUL_{Si} + RUL_{Di} - M_j\}$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_0 = \mathbf{0}$	Probability distribution $\pi(RUL)$ is obtained for each tested engine by simulation via PoF model for RUL according to the "Updated Believes (1st Update)" for agents RUL_{B1} (ID# 23), RUL_{B2} (ID# 24), RUL_S (ID# 25), RUL_D (ID# 26) and accumulated missions M_j: $RUL = Minimum\ \{RUL_{B1i} + RUL_{B2i} + RUL_{Si} + RUL_{Di} - M_j\}$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_1 = \mathbf{500}$	Probability distribution $\pi(RUL)$ is obtained for each tested engine by simulation via PoF model for RUL according to the "Updated Believes (2nd Update)" for agents RUL_{B1} (ID# 23), RUL_{B2} (ID# 24), RUL_S (ID# 25), RUL_D (ID# 26) and accumulated missions M_j: $RUL = Minimum\ \{RUL_{B1i} + RUL_{B2i} + RUL_{Si} + RUL_{Di} - M_j\}$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_2 = \mathbf{1000}$	Probability distribution $\pi(RUL)$ is obtained for each tested engine by simulation via PoF model for RUL according to the "Updated Believes (3rd Update)" for agents RUL_{B1} (ID# 23), RUL_{B2} (ID# 24), RUL_S (ID# 25), RUL_D (ID# 26) and accumulated missions M_j: $RUL = Minimum\ \{RUL_{B1i} + RUL_{B2i} + RUL_{Si} + RUL_{Di} - M_j\}$, where $i = 1, \ldots, E$, E is sample size, $E = 3$, $M_j = M_3 = \mathbf{1200}$

The prediction of system RUL occurred after 2000 engine test cycles upon the 7th round of agents learning. The engine test continued after 2000 missions until failure of at least one of four components (shaft, disk and two bearings) in each of three tested engines. The RUL predicted upon 2000 engine test cycles was compared to the true value of RUL obtained upon system failure, and the results are summarized in Table 8-12. The true value of RUL at 2000 engine cycles (named "RUL True Value" in Table 8-12) appeared to be within the 90% confidence interval of the predicted RUL (named "RUL Prediction – 90% Confidence Bounds" in Table 8-12).

Table 8-12: Summary of RUL Results
(all values are given in engine cycles)

Tested Engine	Component				**System**	**Weakest Component**
	Bearing 1	**Bearing 2**	**Shaft**	**Disk**		
RUL Prediction						
90% Confidence Bounds					Mean	According to Mean RUL
Engine #1	803 - 1185	721 - 1024	694 - 997	**606 - 828**	**700**	**Disk**
Engine #2	716 - 954	610 - 799	**534 - 796**	768 - 956	**634**	**Shaft**
Engine #3	485 - 721	455 - 653	**327 - 573**	740 - 1115	**432**	**Shaft**
RUL True Value						
Engine #1	1005	900	820	**720**	**720**	**Disk**
Engine #2	815	675	**640**	840	**640**	**Shaft**
Engine #3	575	540	**420**	890	**420**	**Shaft**
RUL Prediction (*Shaft-Bearings Interaction excluded*)						
90% Confidence Bounds					Mean	According to Mean RUL
Engine #1	803 - 1185	721 - 1024	2238 - 2551	**606 - 828**	**705**	**Disk**
Engine #2	716 - 954	**610 - 799**	1319 - 1523	768 - 956	**698**	**Bearing 2**
Engine #3	485 - 721	**455 - 653**	910 - 1254	740 - 1115	**531**	**Bearing 2**

Interactions between failure mechanisms of several components were treated within the agent autonomy. Specifically, the PoF model of the shaft (built into Type II Macro-Agent, shaft crack size, ID #19 and Type II Macro-Agent, shaft life to crack size at failure, ID#20) includes a term denoting the maximum spall size of the two bearings (given by Type II Macro-Agent, maximum spall size, ID #16). This implies that a spall progression in the two bearings accelerates the degradation of the shaft (shaft crack growth). Evolution of the respective agents over time resulted in accurate predictions of system reliability, as mentioned above. Removing shaft-bearings interactions caused a poor performance of the model (refer to section "RUL Prediction (Shaft-Bearings Interaction excluded) – 90% Confidence Bounds" in Table 8-12), proving a significance of this interaction for system evolution.

Table 8-13 demonstrates the agent autonomy property (i.e. self-activation and deactivation of Type I Micro-agents during system evolution). Some agents (ID#2 and ID#5) deactivated themselves when more data became available, while others remained active due to their significant contribution to the uncertainty of the associated Type II Macro-Agents' output variables (listed in column "Input to Agents ID# in Table 8-13).

All other properties of agents are applicable as defined in Chapter 4: dependent agents react to the updates of their respective input agents, all agents have a goal of self-evolving following the system degradation processes, and cooperate by communicating their believes to other agents. All agents are mobile because they could be reused in unrelated but relevant applications.

Table 8-13: Activation/Deactivation Property of Type I Micro-Agents

Type I Micro-Agents										
ID#	Name of Agent Output Variable	Letter ID	Input to Agents ID#	Agent Status						
				Updated Beliefs (1st Update)	Updated Beliefs (2nd Update)	Updated Beliefs (3rd Update)	Updated Beliefs (4th Update)	Updated Beliefs (5th Update)	Updated Beliefs (6th Update)	Updated Beliefs (7th Update)
1	**Tangential Force on the Turbine Wheel Disks**	$\boldsymbol{R}$	12, 13	Active	Active	Active	Active	Active	Active	Active
			14, 17	Active	Active	Active	Active	Active	Active	Active
			15, 18	Active	Active	Active	Active	Active	Active	Active
2	**Amplitude of Tangential Force on the Turbine Wheel Disks at Start**	$\boldsymbol{\Delta R}$	19, 20	Active	Active	Active	Active	Active	Inactive R = 230 N	Inactive R = 230 N
			21, 22	Active	Active	Active	Active	Active	Active	Active
3	**Random Vibration**	$\boldsymbol{G_{rms}}$	19, 20	Active	Active	Active	Active	Active	Active	Active
4	**BOT at start**	$\boldsymbol{T_{IT}}$	21, 22	Inactive T_{IT} = 1025°C	Active	Active	Active	Active	Active	Active
5	**Dwell Time before Shutdown**	$\boldsymbol{D}$	21, 22	Active	Inactive D = 160 s	Inactive D = 155 s	Inactive D = 155 s	Inactive D = 155 s	Inactive D = 155 s	Inactive D = 155 s
6	**Initial Crack Size (depth), Shaft**	$\boldsymbol{a_{initialS}}$	19, 20	Active	Active	Active	Active	Active	Active	Active
7	**Initial Crack Size (depth), Disk**	$\boldsymbol{a_{initialD}}$	21, 22	Active	Active	Active	Active	Active	Active	Active

8.9. Case Study Summary and Conclusions

A comprehensive case study was presented to demonstrate the agent-oriented approach to PoF reliability modeling of the dynamic system with interacting failure mechanisms. Agent autonomy was developed to represent the degradation processes of gas turbine engine components during engine operation. The predicted RUL for a system of several components was in agreement with the actual time to failure. The interactions between the failure mechanisms of several components (the shaft and two bearings) were incorporated into the agent-based system model and were shown to be critical for the model accuracy. Several finding were made in relation to the learning, autonomy and mobility properties of the agents:

1. Type II Macro-Agents appeared to be better “learners” than Type I Micro-Agents and Type III Monitoring Agents. The learning process of Type II Macro-Agents involved acquisition of our knowledge about the progression of degradation processes within the interdependent and interacting system components in dynamic environment, which is the “core” of system reliability modeling. In contrary, the Type I Micro-Agents learned by updating their knowledge about the time trend or probability distribution of the associated random variables, while the Type III Monitoring Agents used a simulation algorithm to combine probability distributions according the applicable logic. The agent learning process, therefore, was found to be the most intense and mathematically complex for the Type II Macro-Agents.
2. Another finding of the case study concerned the agent autonomy property. While all Type I Micro-Agents activated/deactivated themselves during the

seven steps of the system model update, the Type II Macro-Agents remained active at all times during the system evolution because they represented degradation characteristics of the system components that are critical for the system reliability. The Type II Macro-Agents were the key “players” in the probabilistic representation of system degradation and failure processes and the major contributors to the uncertainty in the system RUL due to complexity of their PoF models, lack of prior information and limited test and field data for model development. The Type III Monitoring Agents also remained active in order to deliver probabilistic representation of the remaining useful life of each component and the system as an ultimate objective of the agent-oriented reliability modeling. The autonomy property, therefore, was found to be most effective in deactivating “unimportant” Type I Micro-Agents. The Type II Macro-Agents and Type III Monitoring Agents remained active throughout the study since the engineering knowledge suggested their continuous “importance” for the uncertainty within the system reliability model.

3. The mobility property of the agents in the case study was found to be the most important for Type II Macro-Agents. As these agents learned and executed their autonomy during system evolution, they acquired significant knowledge about the elements they represent, specifically degradation characteristics of the engine components. These agents turned into "experts" that could be reused in future studies of the same or similar engines.

The case study also highlighted the substantial computational effort required to complete the agent-oriented reliability assessment. A combination of several software programs (WinBUGS/OpenBUGS [120], [121], Oracle Crystal Ball [122], Minitab [73]) was used for agent learning and autonomy execution. The system reliability model became very large even though the most complex methods of agent learning, such as Kalman filters or artificial neural networks, were not utilized. The application of machine learning and pattern recognition methods for agent learning, or global sensitivity analysis methods for agent autonomy execution, would have required the use of MATLAB, SAS or custom developed code in addition to the software programs used in this case study. Software developments, therefore, are needed in order to create a single program for system reliability modeling by PoF-based agent autonomy. Such software program should offer multiple data analysis and simulation methods (such as Monte Carlo and Latin Hypercube sampling, Markov chain Monte Carlo (MCMC) methods, classical distribution fitting, model-based distribution analysis, Kalman filters, and other machine learning and pattern recognition algorithms, and variance-based uncertainty importance methods). The applicable methods would be chosen by the modeler and sequenced within a single interface.

Chapter 9: Summary, Conclusions and Recommendations for Future Research

9.1. Summary and Conclusions

The following summary outlines the contributions, findings and conclusions of this work:

1. Agent autonomy was proposed as a solution method for the physics-of-failure based reliability modeling of complex engineering system. The concept of agent autonomy originated from Artificial Intelligence (AI) as a computational inference for modeling intelligent Multi-Agents Systems. In system reliability modeling, however, agent hierarchy, classification, and the properties of agents are somewhat different from those in computer science and artificial intelligence. This distinction implies that current research advances our ability to perform dynamic degradation and reliability analysis by defining and modeling agents in the context of reliability engineering.
2. In the agent-oriented system reliability model, each element of the system (piece part, component, environmental and operational parameter, software characteristic, or human element) is replaced by an agent as an intelligent piece of software that represents the entire knowledge about the item. Each agent is represented by a single output variable, Y, which could be either a function of one or more input variables, X_i ($i = 1, \ldots, n$), given by the input agents, $Y = f(X_i)$, or an independent random variable that has no inputs from

other agents. A system is a network of agents - autonomous intelligent entities which communicate with each other during system evolution.

3. Three classes of agents are introduced according to the different types of entities within the physical processes of degradation and failure at all levels of system hierarchy, from materials and piece parts to components and the entire system, considering software and human elements. The proposed classification and hierarchy of agents is flexible enough to model engineering system of any type and complexity.
4. The properties of each class of agents were described based on the classification of agent properties introduced by M. Azarkhail [1] and considering the physical characteristics of agents' counterparts in the real system and their role in system evolution. Seven properties of intelligent agents are: internal knowledge, learning/reasoning, proactivity/goal orientation, communication/cooperation, autonomy and mobility. While the same definitions of agent properties as in [1] were used in the current research, some properties, specifically learning and autonomy, were taken to a new level and further developed in this research.
5. The learning property of agents was defined as an agent's ability to use new data and previous experiences to update an agent's internal knowledge, particularly an agent's beliefs about the functional form (type) and parameters of the PoF or empirical model of agent output variable. In addition, a dependent agent (with one or more inputs from other agents) obtains the latest (updated) beliefs of the respective input agents and performs a simulation over

the updated model of the agent output variable using the latest beliefs about the input variables in order to update the probability distribution of the agent output variable. This is how agents fulfill their goal of self-evolving in the dynamic system environment. The updated probability distributions of the agent output variables are further used for prognosis (e.g. evaluation of the remaining useful life as system reliability measure).

6. Several algorithms of agent learning, specifically aimed to update parameters of the PoF or empirical model of the agent output variable, were proposed along with the guidelines for the selection of a learning algorithm, depending on agent class, availability of the PoF or empirical model of the system element represented by the agent, and the types of data used for agent learning. Bayesian inference and Bayesian Fusion were identified as the preferred methods of agent learning because they allow the recursive (sequential) updating of agent beliefs using all available data, such as to maximize system reliability knowledge and minimize the computation time.
7. The intelligent agent is not only capable to evolve over time without supervision (by means of learning), but also has some degree of control over its participation in system evolution, particularly the ability to activate and deactivate itself. The activation/deactivation capability of agents represents the autonomy property and is developed in this research by means of sensitivity analysis/uncertainty importance methods, particularly Sobol's method of global sensitivity analysis or the Importance Index method of local sensitivity analysis (where applicable).

8. Agent autonomy proposed in this research successfully is capable to model the interdependency and interactions between failure mechanisms of different elements of a system, specifically where the degradation process in one element (part, material or component) activates or accelerates the failure mechanisms of other elements. The dependent elements are explicitly introduced into the PoF or empirical models of the respective system elements and output variables of the associated agents, allowing for bidirectional communication between agents where interacting failure mechanisms exist.
9. The agent-oriented approach to PoF reliability modeling was demonstrated by a comprehensive case study involving the reliability modeling of a gas turbine and aircraft engine structures. Agent autonomy was developed to represent the interacting failure mechanisms of the engine components during engine operation. The predicted RUL for a system of several components was in agreement with the actual time to failure. When agent output models were revised to remove the interactions between failure mechanisms, the system model performed poorly, proving a significant advantage of the proposed methodology in modeling complex engineering systems with interacting and interdependent failure mechanisms.
10. Two major findings of the case study are related to the learning and autonomy properties of intelligent agents. Type II Macro-Agents appeared to be better "learners" than Type I Micro-Agents or Type III Monitoring Agents. While the learning process of most Type I Micro-Agents constituted time trending or improved knowledge about the probability distribution of a random variable,

Type II Macro-Agents' learning reflected on the progression of the degradation processes of the interdependent and interacting system elements in a dynamic environment. Type III Monitoring Agents use simulation to combine several probability distributions according to certain logic as the associated agents evolve over time. The learning process of the Type II Macro-Agents is the most intense and forms the "core" of the system reliability modeling by agent autonomy.

11. Another finding of the case study concerns the autonomy property of agents. It was found that, while agent activation/deactivation is accomplished by all Type I Micro-Agents, most of Type II Macro-Agents remain active during the system evolution because they carry the knowledge about the key "players" in the system degradation processes. Almost all Type III Monitoring Agents also remain active in order to probabilistically represent the system parts and components. Specifically, Type III System Monitoring Agent remains in an active state at all times during system evolution because deactivation of this agent defeats the purpose of probabilistic modeling of system reliability. The autonomy property, however plays an important role in agent-oriented system reliability modeling because the number of Type I Micro-Agents in the agent hierarchy is considerably large compared to the number of Type II Macro-Agents and Type III Monitoring Agents.
12. The fundamental difference between the agent autonomy and all existing methods of probabilistic PoF-based reliability modeling and simulation is in the autonomy property of intelligent agents and the capability of the agent

autonomy to model the interacting failure mechanisms of system elements at all levels of system hierarchy. These two features bring more reality into the reliability models, making the agent autonomy superior to the existing methods of system reliability modeling.

13. As agents learn and execute their autonomy during system evolution, they become richer in their knowledge about the elements that they represent and their "importance" to the other elements of the system hierarchy, and take an active role in system evolution by activating or deactivating themselves. As a result, the agents become mature "experts" that could be reused in other unrelated but relevant applications to reduce computational effort, minimize data requirements, and provide high quality prior information for further learning.
14. Despite of the advantages described above the agent autonomy approach has several limitations, such as some ambiguity in selection between Type I Micro-Agents and Type II Macro-Agents that may exist for complex PoF models, limited mobility of agents according to their relevance to another application.
15. Some challenges pertaining to other reliability assessment methods are also present for the agent autonomy, specifically uncertainty characterization:
    a) Both epistemic and aleatory uncertainties are present within the agent autonomy model as they emerge from various sources. Separation of epistemic and aleatory uncertainties is generally not a straightforward

process for any modeling methodology and requires good engineering knowledge about the modeled phenomenon.

b) Uncertainty importance assessment is performed within the context of agent autonomy execution and requires use of global methods of variance decomposition because of the complexity of agent autonomy modeling structure (PoF-based modeling framework containing bidirectional communication between agents and feedback loops). The global methods of uncertainty importance analysis, however, impose significant computational effort in evaluating the importance indexes.

c) Data availability is another source of uncertainty and a common challenge for all probabilistic modeling methods when limited and partially relevant data are used for reliability model development. Bayesian inference is, therefore, identified as a preferred framework of agent learning to maximize use of the available information and minimize uncertainty.

d) Modeler's knowledge of physical failure mechanisms and availability of PoF models is critical for quantitative assessment of the model related uncertainties within the agent autonomy. In case of limited PoF knowledge of the system degradation processes or high complexity of the plausible PoF models, special computer techniques may need to be developed to evaluate and compare model uncertainties.

16. The agent autonomy approach also offers several improvements with respect to uncertainty characterization:

a) The agent autonomy approach to system reliability modeling allows some reduction of subjectivity and arbitrariness in the definition of system failure scenarios and their consequences compared to fault tree and event tree methodologies, for example, because the intelligent agents evolve autonomously reflecting on all relevant failure scenarios (not only those believed to be the "worst case" sequences).

b) Another aspect of uncertainty characterization is related to the capability of the agent autonomy to provide realistic representation of the system evolution over time. For example, fault trees and event trees typically use the classical binary success/failure logic of system reliability representation. In addition, fault trees and event trees are static techniques which cannot take into account time-dependent evolutions of dynamic systems. In contrary, the agent autonomy effectively models dynamics of the system evolution in time and can differentiate between different levels of system performance depending on the degraded states of the constitutive parts and components. As a result, epistemic uncertainties, introduced by simplification and approximation of the reality, will be reduced in agent autonomy models.

17. A disadvantage of the agent-oriented approach to system reliability modeling is the high computational effort. While Bayesian Fusion techniques and global methods of uncertainty importance/sensitivity analysis are computationally intense due to their complexity, the reliability model of a system with very few components becomes large and difficult to handle even if Bayesian

inference and simple local methods of sensitivity analysis are used, as in the presented case study of gas turbine structures. Software developments are necessary to further support the use and expansion of PoF-based agent autonomy as system reliability modeling approach. Despite of the significant computational effort, the complexity of agent autonomy framework is reasonable and manageable, and agent-based system reliability modeling is practically attainable in contrast to the traditional simulation.

9.2. Recommendations for Future Research

Since the scope of the agent autonomy approach to PPoF modeling of dynamic systems is very broad, further research is required to develop some of the specific aspects of the proposed methodology, as follows:

1. *Performability Modeling*

   Many real-world systems are composed of multi-state components, which have different performance levels where one cannot formulate an "all or nothing" type of failure criteria. Even though this work introduced representation of the item degradation process within the agent oriented model, the qualitative definition of the consequent change in capacity and/or performance efficiency of the system will need to be included in future research. A partial loss of function and degraded performance models will need to be introduced into the agent structure.

2. *Phased-Mission Modeling*

The agent-oriented approach to phased-mission modeling has been discussed in [123] and [124]. It was noted that the hierarchy of agents will, in general, change from phase to phase during the mission. Based on the requirements of each phase, assemblies and their related components are called for duty. In addition to the changes in agent hierarchy, the agent-oriented approach proposed in this research also implies that the functional form of the PoF or empirical model of agent output variables may change between mission phases. This is because the physical failure mechanisms and failure logic of the system elements and the system itself may differ from phase to phase. As a result, the system reorients itself and the structure of agent autonomy becomes dynamic.

While this research makes provisions for introduction of phased-mission rules within an agent's internal knowledge (particularly, as part of special rules of agent behavior), further work is required to elaborate the dynamics of agent hierarchy, change in agent type, and rules of agent's beliefs update at all levels due to the effects of mission phases.

3. *Representation of Human Interactions within Agent Hierarchy*

The agent definition and classification proposed in this research includes human factors as part of the agent-oriented system model. Further research may be necessary to elaborate on the detailed methods of introducing a human element into the agent autonomy. The agent learning methods may be somewhat different from those introduced in this research for hardware parts and components due to the subjectivity involved in human reliability analysis.

4. *Software Faults and their Impact on Agent-Oriented System Model*

    The agent definition and classification developed in this research contains agents representing software programs in addition to hardware parts and components as part of an agent-oriented system model. Further research may be necessary to elaborate on the detailed methods of software reliability modeling within agent autonomy. Software agents may need to have specific definitions of agent properties that are somewhat different than those introduced in this research.

5. *Representation of Scheduled and Unscheduled Inspection and Maintenance Actions, Renewal Process Modeling*

    Agent definition and classification established in this research makes provisions for defining agents that represent inspection and maintenance tasks. This work, however, does not provide detailed examples of introducing these tasks into the agent autonomy framework, which could be a subject for further research. Renewal process and availability assessment would have to be represented within the agent autonomy by means of stochastic models of reliability of repairable system to address inspection, maintenance, testing and repair activities required for mechanical components.

6. *Mobility Property of Agents in Case of Partial Relevance*

    In order to fully develop the mobility property of agents, the ability to reuse partially relevant agents in other system applications may need to be elaborated (e.g. by adjusting agent output model in order to reflect on the degree of relevance).

Appendices

Appendix A: Fatigue Failures of Gas Turbine Aircraft Engine Structures

A.1. Physics-of-Failure Life Model of Gas Turbine Bearings

While many types of mechanical components fail due to fatigue, bearings are unique since fatigue cracks in bearings form under high compressive stress, and spall formation is a result of the continuous initiation and coalescence of thousands of small cracks, rather that the propagation of a single dominant crack. Thus, traditional approaches to fatigue modeling (such as Paris law) perform poorly when applied to bearings.

Other equations have been developed to quantify the life of a bearing under a given set of operating conditions [125]. Current methods for predicting the life of rolling element bearings are based on the initiation of the first spall. Spalling is defined as subsurface chipping and breaking [126], [127]. Spalls are generated when a micro piece of metal flakes off the rolling surface. In aircraft propulsion applications, bearings are typically designed to have a fatigue life greater than the design life of the subsystem they are in. However, debris contamination (due to inadequate sealing or contaminants in the lubricant) or mishandling can damage the bearing surfaces and lead to spall initiation. Misalignment is also a prolific source of premature spalling.

Once an incipient spall has formed, it causes a stress concentration and becomes self-propagating. When initiated, a spall grows relatively quickly producing high vibration levels, and debris in the oil. Due to the relatively short remaining life following spall initiation, the appearance of a spall typically serves as a criterion for failure of bearings in critical applications. While spall progression typically occurs more quickly than spall initiation, studies showed that 3 to 20 % of a particular bearings useful life remains after spall initiation [128].

Typically an equation similar in form to Equation A-1 is used to predict the life-to-spall initiation of a roller bearing [126], [128] - [130], as follows:

$$L_n = a_1 a_2 a_3 \cdot \left(\frac{C}{P} \right)^{10/3}$$

Equation A-1

where C is basic dynamic capacity (dynamic load rating) of the bearing, and P is equivalent radial load (pressure) on the bearing, a_1 is life adjustment factor (a_1 = 1.0 for $L_n = L_{10}$), a_2 is life adjustment factor for bearing materials, a_3 is life adjustment factor for application conditions (temperature, rotation speed, lubrication regime). Random vibration would cause bearing degradation (particularly false brinelling) only in a stationary bearing, therefore the life adjustment factor, a_3, does not include random vibration stress.

For a particular bearing material and specific operational conditions of the engine, the mean life is a function of two variables: equivalent radial load, P, and operating speed, V. In turn, bearing speed, V, is proportional to the tangential force on the turbine wheel disks, R, which generates torque on the shaft, so that:

$$a_3 \propto \frac{1}{V} \propto \frac{1}{R}$$

Equation A-2

Equivalent radial load, P, is derived from shaft bending moment, M_b, shaft torque, T, and bearing preload, p. As shown in Section A.2. of this appendix, both the shaft bending moment and the shaft torque are proportional only to one variable, tangential force on the turbine wheel disks, R. Equivalent radial load, P, can be written as:

$$P = \beta_1 R + \beta_2 p$$

Equation A-3

where β_1 and β_2 are proportionality constants related to the geometry of the shaft and the bearing. Under these considerations, Equation A-1 for life to spall initiation can be reduced to:

$$L = B_1 R^k (B_2 R + B_3)^{-10/3}$$

Equation A-4

where B_1, B_2 and B_3 are proportionality constants combining all material and geometrical constants, bearing preload, p, and basic dynamic capacity, C.

Life to spall initiation can be defined in accumulated flight hours or accumulated missions, if the mission duration is constant. In this work, the aircraft mission has an average duration of two flight hours and does not substantially change from flight to flight. Therefore, the life to spall initiation will be counted in aircraft missions, M, each being two hours long. The mission, M, represents starting a cold engine, taking-off, flying to a destination, landing, idling and shutting-down. This stress history from the time of start-up to landing is defined as a disk cycle.

The actual beginning of spalling is invisible because the origin is usually below the surface. The first visible sign is a small crack and this too is usually indiscernible. The crack cannot be seen or indirectly detected otherwise. For practical purposes, therefore, life to spall initiation given by Equation A-4 is defined as the time to the development of a detectable spall. The time between incipient and advanced spalling varies with speed and load, but in any event it is not a sudden condition that will cause destructive failure within a matter of hours. With condition monitoring by regular inspection a fatigue spalling can be detected. Generally, the failed component is replaced and operation of the mechanism recommences.

When a fatigue spall is formed in a bearing, the contact stress, heat generation rate and vibration are increased, accelerating the formation of fatigue cracks within the unfailed subsurface material of the contact area. The contact surface continuously deteriorates upon the creation of new subsurface cracks and propagation of the existing ones. The fatigue spall grows during the bearing operation until the entire contact area has been roughened, leading to bearing failure [131].

In most applications, the replacement of the bearing after the generation of the initial fatigue spall is an acceptable practice. However, in some applications the initiation of the first spall does not indicate the end of the useful life of the bearing. Depending on the amount of acceptable vibratory loading within the system and the means for heat dissipation, a bearing with a fatigue spall can be used for many cycles of operation beyond the initial failure.

Stable spall progression is characterized by gradual spall growth and exhibits low broadband vibration amplitudes. The onset of unstable spall progression coincides with increasing broadband vibration amplitudes [128]. A study in [131] used a modified semi-empirical method for predicting spall progression rates for roller bearings as:

$$\frac{dD}{dN} = K_1 \left(D_P\right)^{K_2}$$

Equation A-5

where K_1 and K_2 are empirical constants to be determined for the bearings under the operating conditions for which Equation A-5 is used, N is a cycle count, D_P is a degree of bearing damage due to spall progression.

Damage mechanics based fatigue model, therefore, would explicitly take into account the gradual material degradation that occurs during rolling contact cycling under applied stresses. As discussed in [125], damage evolution is assumed to occur according to an equation of the following form:

$$\frac{dD}{dN} = \left(\frac{\Delta\tau}{K_3(1 - D_P)}\right)^m$$

Equation A-6

where K_3 and m are material constants, $\Delta\tau$ is shear stress range acting along the inter-element joint (bearing surface). The damage variable D can be further defined as follows [132]:

$$D_P = \frac{S_D}{S}$$

Equation A-7

where S_D is the damaged surface area as projected area of all spalls, and S is the total surface area of the bearing.

The study in [128] uses a similar model of spall propagation which relates spall progression rate, dS_p/dN, to the spall similitude, W_{sp}, using two constants (A and m):

$$\frac{dS_p}{dN} = A\left(W_{sp}\right)^m$$

Equation A-8

The spall similitude is defined in [128] in terms of the maximum radial (normal) stress, σ_{max}, average shear stress, τ_{avg}, and the spall length, S_p:

$$W_{sp} = \left(\sigma_{\max} + \tau_{avg}\right)\sqrt{\pi \cdot S_p}$$

Equation A-9

Note that $\sigma_{max} = \sigma_n = P$ is the normal stress on the surface resulting from pressure, P, τ_{avg} is friction generated shear stress, $\tau_{avg} = \tau_f = \mu P$, where μ is a friction factor. The friction factor is determined based on the lubrication regime of the bearing.

Combining Equation A-3, Equation A-8 and Equation A-9 results in the following semi-empirical model of spall propagation rate, represents bearing damage accumulation as a function of material loss:

$$\frac{dS_p}{dN} = B_4\left(S_p\right)^{m/2}(B_5 R + B_6)^m$$

Equation A-10

where dS_p/dN is damage accumulation rate due to spall propagation, and B_4, B_5 and B_6 are proportionality constants capturing all constants included in Equation A-3, Equation A-8 and Equation A-9. Stress cycle count, N, for the turbine bearing (will be defined as N_B from now on) represents one aircraft mission, M, two flight hours long.

The spall length, S_p, can be substituted by spall size as a percentage of bearing surface, S_D.

Integration of Equation A-10 gives time to spall progression to the size, S_{pi}, as the number of stress cycles passed since spall initiation (at the time of bearing inspection), N_{Bi}:

$$N_{Bi} = M_i - L = \frac{\left(S_{pi}\right)^{1-\frac{m}{2}}}{B_4(B_5R + B_6)^m \left(1 - \frac{m}{2}\right)}, \qquad m \neq 2$$

Equation A-11

The number of missions accumulated up to a certain time, M_i, includes life to spall initiation, L, and stress cycles (missions), N_{Bi}, accumulated since spall initiation until propagation to the size, S_{pi}:

$$M_i = L + N_{Bi}$$

Equation A-12

Equation A-11 can be solved for S_{pi} to obtain the relationship between the spall size and stress cycles from spall initiation, N_{Bi} (or $M_i - L$):

$$S_{pi} = \left[B_4(B_5R + B_6)^m \left(1 - \frac{m}{2}\right)(M_i - L)\right]^{\frac{1}{1-\frac{m}{2}}}, \qquad m \neq 2$$

Equation A-13

The degree of spall progression, denoted by S_{pi} in Equation A-11 and Equation A-13, can be detected during periodic inspections by measuring the spall length, or spall size as a percentage of track surface, or oil particle quantity. The latter approach, however, requires preliminary lab testing to obtain scaling of the oil particle quantity to the approximate spall size [128].

The spall length is zero before the initiation of a detectable spall. Spall size threshold defined as S_{pLimit} is allowable material removal. When exceeded, a bearing is considered failed. Spall progression life, N_p, can be obtained from Equation A-10 by integrating over the range of spall progression from zero to S_{pLimit} (or from Equation A-11 by replacing S_{pi} with S_{pLimit}):

$$N_B = \frac{\left(S_{pLimit}\right)^{1-\frac{m}{2}}}{B_4(B_5R + B_6)^m\left(1 - \frac{m}{2}\right)}, \qquad m \neq 2$$

Equation A-14

While tangential force on the turbine wheel disks, R, is explicitly defined in Equation A-4 and Equation A-10 as a random variable, other parameters, such as preloading, misalignment, lubrication, material parameters, geometrical and manufacturing tolerances of the particular bearing, are included in the constants k, m, B_1 to B_6, which will be probabilistically defined within the framework of PoF-based agent autonomy, as described in Chapter 8. This approach was used in order to simplify the life model of the bearings. It is always an option, however, to explicitly introduce any variable into this PoF-based life model, if engineering knowledge and data availability allows.

A.2. Physics-of-Failure Life Model of Gas Turbine Shaft

Most catastrophic mechanical failures in power rotor shafts occur under cyclic bending that is combined with steady torsion, where cyclic bending stress is due to the self-weight bending during the rotation or possible misalignment between roller bearings. The multi-axial mixed-mode fatigue crack growth is an occurrence common to many engineering structures and components. The fundamental postulate of the linear elastic fracture mechanics (LEFM) is that the behavior of a crack (i.e. whether it grows or not, and how fast it grows) is determined by one parameter, the stress intensity factor (SIF). This factor is a function of the applied loading and the geometry of the cracked component. For mixed-mode loading, the fatigue crack growth rate may be expressed by the Paris law, where the SIF range is replaced by an equivalent SIF range, ΔK_{eq}:

$$\frac{da}{dN} = C_s\left(\Delta K_{eq}\right)^n$$

Equation A-15

where C_s and n are constants depending on the several parameters such as the material type, microstructure, environment, stress ratio R, K_{max} ($\Delta K = K_{max} - K_{min}$), fracture toughness K_{Ic}. There are many approaches proposed to define the equivalent SIF range ΔK_{eq} for mixed-mode loadings [133]. One of them is based on the equivalence of energy release rate, G, and the stress intensity factor, K, for nominal elastic loading [134]. Adding the individual energy release rates for a planar crack under plane stress conditions for the three loading modes [135] results in the equation:

$$G_{total} = G_I + G_{II} + G_{III} = \frac{K_I^2}{E} + \frac{K_{II}^2}{E} + (1+\nu)\frac{K_{III}^2}{E}$$

Equation A-16

This leads to the following equivalent stress intensity factor range:

$$\Delta K_{eq} = (G_{total}E)^{1/2} = [\Delta K_I^2 + \Delta K_{II}^2 + (1+\nu)\Delta K_{III}^2]^{1/2}$$

Equation A-17

where G_{total} is the total strain energy release, and E is the Young modulus,

$$\Delta K_I = Y_1 \Delta\sigma(\pi a)^{1/2}$$

Equation A-18

$$\Delta K_{II} = Y_2 \Delta\sigma(\pi a)^{1/2}$$

Equation A-19

$$\Delta K_{III} = Y_3 \Delta\sigma(\pi a)^{1/2}$$

Equation A-20

and Y_1, Y_2, Y_3 are geometry factors (for three crack loading modes, Mode I, Mode II and Mode III, respectively), depending on the crack geometry, material Poisson's ratio, ν, and the loading conditions, $\Delta\sigma$ is an amplitude of the remote applied bending stress, $\Delta\tau$ is an amplitude of the remote applied shear stress (Mode II) or torsion stress (Mode III), and a is a crack size [134].

The term ΔK_{eq} can be used for the mixed-mode loading condition in a Paris law type equation (Equation A-15) to obtain the crack growth rate, *da/dN*, or cycles to failure through integration. Equation A-17 is used to obtain Mode I and Mode III SIFs along the front of the semi-elliptical surface cracks in shafts subjected to simultaneous bending and torsion (where the surface crack is on a normal plane to the axis of the shaft):

$$\Delta K_{eq} = [(Y_1 \Delta\sigma)^2(\pi a) + (1+\nu)(Y_3 \Delta\tau)^2(\pi a)]^{1/2}$$
$$= (\pi a)^{1/2}[(Y_1 \Delta\sigma)^2 + (1+\nu)(Y_3 \Delta\tau)^2]^{1/2}$$

Equation A-21

Equation A-15 for Paris law of crack growth rate then becomes:

$$\frac{da}{dN} = C_s(\pi a)^{n/2}[(Y_1\Delta\sigma)^2 + (1+\nu)(Y_3\Delta\tau)^2]^{n/2}$$

Equation A-22

For a shaft supported in bearings at its ends and carrying the turbine wheels, the maximum normal stress due to bending loads, σ, and the torsion shear stress, τ, are found on the surface of the shaft. Bending and torsion shear stresses are the functions of shaft geometry and loading conditions, as follows:

1. Bending stress, σ, is proportional to the bending moment, M_b, in the location of the initial crack and section modulus, *S*, for the solid circular shaft with diameter *d* and length *L*:

$$\sigma = \frac{M_b}{S}$$

Equation A-23

where

$$M_b = c \cdot F \cdot L \sim F \cdot L$$

Equation A-24

$$S = \frac{\pi d^3}{32}$$

Equation A-25

The bending force on the shaft, *F*, is a combination of turbine wheels weight (mass), *m*, and tangential force on the turbine wheel disks, *R* (due to tangential force generated on the turbine blades as a response to the surface (aerodynamic) loads), and *c* is a constant related to the location of the initial crack. The surface loads on the turbine blades are associated with aerodynamic forces resulting from the impingement of hot gases on the surfaces of the blades. The tangential force is a push force applied perpendicular to the radial line of the shaft causing the driven shaft to rotate at a certain speed. As such, the rotational speed of the shaft is proportional to the tangential force generated on the turbine blades. The tangential force on the turbine wheel disks, *R*, is steady during normal steady-state operation (constant rotational speed) and variable otherwise at operation start and upon operation completion, and is proportional to the change in speed from zero to operating speed at start and from operating speed to zero prior to the shutdown.

2. Torsional stress, τ, for the solid circular shaft with a diameter *d* is proportional to the torque, *T*, section modulus, *S*, and radius of turbine wheel disk, *r*:

$$\tau = \frac{T}{2S}$$

Equation A-26

where

$$T = R \cdot r$$

Equation A-27

$$S = \frac{\pi d^3}{32}$$

Equation A-28

The bending stress is fully reversed as the shaft rotates (with zero mean stress), while the torsion stress is a steady stress during normal steady-state operation and variable otherwise with a non-zero mean and zero minimum stress at operation start and upon operation completion. Notations $\Delta\sigma$ and $\Delta\tau$ in Equation A-21 and Equation A-22 are the stress ranges that would have existed at the crack location, but for the initial uncracked shaft. Under constant amplitude loading conditions, these would stay constant. Both amplitudes are, however, variable, as they are affected by the operational profile and the utilization of the engine.

Further, random vibration changes the loading conditions of the entire structure (including the shaft) and the response spectrum of the cracked components (as demonstrated in [136]), which affects the fatigue crack progression through the material. In addition, damage accumulation in the adjacent structural components over time (particularly bearings considered in this work and described in Section A.1. of this appendix) impacts the applied loads of the turbine shaft and alters stress state. These two factors further changing total bending stress by adding two components:

1. A stress component due to random vibration, $\Delta\sigma_V$, which can be written as $\Delta\sigma_V = C_1\Delta\sigma_{rms}$ (according to [136]), where $\Delta\sigma_{rms}$ is equivalent tensile stress range due to random vibration and C_1 is a correction factor for the statistical distribution of the random vibration response (called equivalent damage constant). If G_{rms} is the root mean square value of the acceleration obtained from the random vibration input and the response curve (or power spectral density (PSD) vs. frequency curve), then the equivalent relative displacement, Z_{rms}, is proportional to the G_{rms} value, the 1st natural frequency and a damping coefficient of the shaft [136], [137]. Stress component, $\Delta\sigma_V$, is proportional to the equivalent dynamic displacement, Z_{rms}, and the response natural frequency [136], and can be written as:

$$\Delta\sigma_V = C_1 C_2 G_{rms}$$

Equation A-29

where the constant C_2 encompasses all material and geometrical constants relating Z_{rms} and G_{rms}, $\Delta\sigma_V$ and Z_{rms} (including the 1st natural frequency and damping coefficient). As shown in [136], the equivalent damage constant C_1 is only a function of the Paris law exponent, n:

$$C_1 = \left[2^{n/2}\Gamma\left(1+\frac{n}{2}\right)\right]^{1/n}$$

Equation A-30

The standard value of random vibration, G_{rms}, for engine structures of a fixed wing aircraft is obtained from DO-160F (curve D on Figure 8-1) [138] as 8.92 G_{rms} for the standard vibration and 12.61 G_{rms} for high vibration conditions. The random vibration signal is assumed to have a stationary Gaussian distribution, and the instantaneous vibration acceleration peaks of the control signal are limited to three times the G_{rms} acceleration level. Also, the random vibration input profile is considered to follow a "smooth" and uniform PSD curve given by DO-160F (Figure 8-1) [138] without notches (peaks and valleys in the frequency domain), so that the shift in natural frequency does not affect the stress state (and as such, life to failure) [136]. The frequency range used is 20 Hz to 2000 Hz.

2. The stress component due to sinusoidal vibration caused by the damaged bearing due to spalling, $\Delta\sigma_S$, is proportional to the degree of the bearing's surface damage due to spall progression, S_p, by the maximum radial displacement, $Z_S \sim S_p$. The stress component $\Delta\sigma_S$ is proportional to the maximum radial displacement Z_S [136], and can be written as:

$$\Delta\sigma_S = C_3 S_p$$

Equation A-31

where the proportionality constant C_3 includes all material and geometrical constants relating Z_S and S_p, $\Delta\sigma_S$ and Z_S.

Considering all sources of bending stress described above, Equation A-21 becomes:

$$\Delta K_{eq} = (\pi a)^{1/2}[(Y_1\Delta\sigma)^2 + (Y_1 C_1 \Delta\sigma_{rms})^2 + (Y_1\Delta\sigma_S)^2 + (1+\nu)(Y_3\Delta\tau)^2]^{1/2}$$

Equation A-32

Using Equation A-32 for the equivalent SIF range, Equation A-15 for Paris law of crack growth rate can be written as:

$$\frac{da}{dN} = C_s(\pi a)^{n/2}[(Y_1\Delta\sigma)^2 + (Y_1 C_1 \Delta\sigma_{rms})^2 + (Y_1\Delta\sigma_S)^2 + (1+\nu)(Y_3\Delta\tau)^2]^{n/2}$$

Equation A-33

After substituting Equations A-23 to A-31 into Equation A-33 and combing all proportionality constants into model coefficients will obtain Paris equation of crack growth rate for the turbine shaft in the following form:

$$\frac{da}{dN} = C_s(\pi a)^{n/2}\left[S_1(\Delta R)^2 + S_2\left[\Gamma\left(1+\frac{n}{2}\right)\right]^{2/n}(G_{rms})^2 + S_3\left(S_p\right)^2\right]^{n/2}$$

Equation A-34

where S_1, S_2 and S_3 are model coefficients. While ΔR and G_{rms} may depend on several operational factors, S_p is a function of time as the degradation of the bearing proceeds. Although this semi-empirical model will utilize an experimental data in lieu of the finite element analysis (FEA), the physics of the underlying fatigue failure will be retained. This approach is preferred over a fully empirical model which is not rooted in the underlying physical process.

Stress cycle count N for the shaft (will be defined as N_S from now on) represents one aircraft mission M, two flight hours long. The relation between aircraft missions accumulated up to a certain time, M_i, and stress cycle count for roller bearings, N_{Bi}, (Equation A-12) and for the shaft, N_{Si}:

$$M_i = L + N_{Bi} = N_{Si}$$

Equation A-35

The number of missions (stress cycles) to failure can be further defined for the turbine shaft by integrating both sides of Equation A-34 and considering that $a_{initialS}^{1-(n/2)} >> a_{Sf}^{1-(n/2)}$ (since $n > 2$ for metals and $a_{initialS} << a_{Sf}$ according to the past experience), where $a_{initialS}$ is the initial crack size and a_{Sf} is the crack size at failure:

$$N_S = \frac{a_{initialS}^{1-n/2}}{C_s\left[\pi\left\{S_1(\Delta R)^2 + S_2\left[\Gamma\left(1+\frac{n}{2}\right)\right]^{2/n}(G_{rms})^2 + S_3\left(S_p\right)^2\right\}\right]^{n/2}\left(\frac{n}{2}-1\right)}, n > 2$$

Equation A-36

The number of missions (stress cycles) to the development of crack size a_{Si} becomes:

$$N_{Si} = M_i = \frac{a_{initialS}^{1-n/2} - a_{Si}^{1-n/2}}{C_s\left[\pi\left\{S_1(\Delta R)^2 + S_2\left[\Gamma\left(1+\frac{n}{2}\right)\right]^{2/n}(G_{rms})^2 + S_3\left(S_p\right)^2\right\}\right]^{n/2}\left(\frac{n}{2}-1\right)},$$

$$n > 2$$

Equation A-37

Equation E-37 can be solved for a_{Si} to obtain a relationship between crack size and the number of accumulated stress cycles (at the time of shaft inspection), N_{Si} (equal to accumulated aircraft missions M_i):

$$a_{Si} = \left[a_{initialS}{}^{1-n/2} - C_s \left[\pi \left\{ S_1(\Delta R)^2 + S_2 \left[\Gamma\left(1+\frac{n}{2}\right) \right]^{\frac{2}{n}} (G_{rms})^2 + S_3\left(S_p\right)^2 \right\} \right]^{\frac{n}{2}} \times \right.$$

$$\left. \left(\frac{n}{2}-1\right) M_i \right]^{\frac{1}{1-n/2}}, \quad n > 2$$

Equation A-38

It must be noted that, while the amplitude of tangential force on the turbine wheel disks, ΔR, and a random vibration level, G_{rms}, are explicitly defined in Equation A-34 as random variables, other variables, such as material parameters, geometry and manufacturing tolerances of the shaft, are included in the constants n, C_s, S_1 to S_3, which will be probabilistically defined within the framework of PoF-based agent autonomy, as described in Chapter 8. This approach was used to reduce the complexity of the life model of HP turbine shaft, although it is always an option to explicitly define any variable within the life model where engineering knowledge allows and data are available.

A separate note should be made about the degree of bearing surface damage due to spall progression (spall size), S_p, which appears in Equation A-34, Equation A-36 and Equation A-37. Since two bearings are included in the system under the study, one of the two bearings with the largest spall size at a given time to be considered in the above equations, as it results in maximum radial displacement. As such, spall size, S_p, can be written as:

$$S_p = Max\left\{S_{p1}, S_{p2}\right\}$$

Equation A-39

A relationship between spall size at any given time S_{pi} and stress cycles of the shaft, N_{Si} (or M_i, see Equation A-35), becomes:

$$S_{pi} = Max\{S_{p1i}, S_{p2i}\}_{N_{Si}}$$

Equation A-40

In order to develop a definition of the amplitude of tangential force on the turbine wheel disks, ΔR, it needs to be noted that the gas turbine components generally experience cyclic thermo-mechanical loading during engine startup/shutdown, while cruise (steady holds) may cause thermo-mechanical creep fatigue damage to the material. Since creep deformation is considered to be negligible due to the advancements in material technology, cyclic loading during startup/ shutdown represents the major loading cycle of gas turbine engine components, which can have a severer impact on the life of the material in comparison to the isothermal conditions during steady state cruise. ΔR in Equation A-34 and further is defined as the amplitude of tangential force on the turbine wheel disks at start-up. As turboprop engines are designed to produce torque, which is driving a propeller, a power output of turboprop engines is measured by a torquemeter. A torquemeter provides the means of accurately measuring the torque input from the shaft into the reduction gear assembly. As such, the torquemeter measures the power applied to the shaft. Torque measurements were used to monitor tangential force on the turbine wheel disks R, particularly to measure the force amplitude ΔR at start-up.

A.3. Physics-of-Failure Life Model of Gas Turbine Disk

This PoF model is developed from the physical principles considering critical variables which contribute to the failure process, as described in [139], [140]. The publications [139], [140] present a case study in which structural reliability-based methodologies are used to assess the cyclic fatigue life of a high pressure turbine of a turboprop engine. The disk material is a cast nickel-base superalloy. The component experiences fatigue damage at the disk rim from mechanically and thermally induced loading. The mechanical loads are caused by the disk rotational speed, and the thermal loads are caused by the temperature gradients throughout the disk. The loadings change with the mission cycle, causing complex stress cycles in the disk. The variation in the primary hot gas and secondary cooling air, the uncertainty of the complex mission loading, and the scatter in mission duration are considered to be the major sources of uncertainty.

The fatigue life of the HP turbine disk is measured by the number of flight missions and can be generally separated into crack initiation and crack propagation phases. To define the initial crack size ($a_{initialD}$), the procedure in [139], [140] assumes a crack initiation life of zero cycles, and chooses a range of initial equivalent flaw sizes based on the comparison of the analytical crack growth results with crack size data from disks inspected in the field. The crack size does not represent the physically realized crack initiation size, but is a crack in the LEFM regime that represents the culmination of crack nucleation and the small crack effect. In other words, although the disk may not have a crack size of $a_{initialD}$ at zero cycles, the crack growth analysis and field experience indicates that the disk "acts" as though it has a crack of size $a_{initialD}$ at zero cycles.

The crack propagation life is determined using a LEFM model (the Paris law). The fracture mechanics analysis for predicting crack growth in disks under the flight cycle loading assumes the cyclic growth of a surface crack that is initiated by the loading of unflawed material (growth of a subsurface crack from an inherent material defect is not considered because the processing of a vacuum-melted superalloy used in aircraft gas turbines now includes stringent controls which minimize the occurrence and size of these defects). Consider the Paris law representation of an edge crack in an infinite plate subjected to a constant stress cycle:

$$\frac{da}{dN} = C_d (\Delta K)^h$$

Equation A-41

where a is the crack length, N is the number of cycles, da/dN is the crack growth rate, ΔK is the stress intensity factor, C_d and h are the constants related to material properties. The factor ΔK is determined using the stress cycle and crack geometry:

$$\Delta K = \beta(\Delta\sigma)\sqrt{\pi a}$$

Equation A-42

where $\Delta\sigma$ is the stress range, β is the geometry constant, β = 1.12 [139], [140].

Gas turbine disks are subjected to complex thermo-mechanical loading with many uncertainties. Potentially significant random variables that influence disk life include BOT (burner outlet temperature) during startup, burner profile, dwell time at idle during startup, and dwell time at idle before shutdown (all categorized as coupling factors within agent autonomy developed in this work). For modeling purposes, quantitative estimation of the above variables, as well as the characterization of the relationships connecting these variables to stress and degradation characteristics of the turbine disk, are derived from the first principles and/or obtained from the FEA [139]. The disk stresses and strains throughout the mission were shown to be determined by the following variables:

1. *Material properties*

   In the present study, the Paris law coefficient C_d and the Paris law exponent h are treated as random variables to represent the scatter in the material data. According to studies [141] and [142], the value of the Paris law exponent, h, for the turbine disk material may vary in the range of $h = 2$ to $h = 6$. It must be noted that both constants are temperature dependent, because the disk operating temperature, T_d, impacts crack growth rate. Three main groups of mechanisms are cited for the temperature dependence of superalloys in the temperature range of 600–800°C [142]: (i) environmental effects including dynamic embrittlement involving atomic oxygen and/or oxidation of grain boundaries at and ahead of the crack tip, and grain boundary carbides (the latter two processes are stress assisted), (ii) creep (especially at grain boundaries), and (iii) changes in the monotonic mechanical properties of the material, most notably the yield strength and the threshold of stress intensity factor range, ΔK_{th}, due to thermally activated modes of dislocation movement and other effects. The relative importance of these processes can change with temperature and crack tip stress cycle, and thus crack growth rate will change depending on these parameters. For these reasons, material constants C_d and h

are treated as a random variables reflecting on the temperature dependency of crack propagation rate.

For the material of the studied turbine disk, thermally activated processes are not dominant compared to the relative importance of pure mechanical fatigue. This is not to say that oxidation or creep does not occur, but that the oxidation or creep that occurs is not significantly enhancing the fatigue crack growth. In this case, pure cyclic fatigue processes (due to mechanical stresses and temperature gradient between bore and rim) are considered to predominate.

2. *Mechanical Loads*

A rotating component in the hot section of a turbine engine is subject to a combination of surface (aerodynamic) loads, centrifugal loads and thermal loads. The surface loads are associated with aerodynamic forces, resulting mainly from the impingement of hot gases on the surfaces of blades. The centrifugal loads that arise from the mass of the rotated disc and blades are usually the most critical loads that act on a turbine disc. Thermal loads are driven by a high temperature in the turbine section and a thermal gradient between the disk bore and rim.

a) *Centrifugal forces (F_c)*

Centrifugal forces are proportional to the square of the rotational speed of the disk, where the latter is expressed as angular velocity of the disk ω or linear velocity v:

$$F_c = mv^2/r$$

Equation A-43

$$v = r \cdot \omega$$

Equation A-44

$$F_c = mr\omega^2$$

Equation A-45

$$\omega = \frac{RPM \times 2\pi}{60}$$

Equation A-46

$$F_c = mr\left(RPM\,\frac{2\pi}{60}\right)^2 = 0.01097mr(RPM)^2$$

Equation A-47

where *r* is the radius of turbine wheel disk, *m* is a mass of the disk, and *RPM* is rotational speed in revolutions per minute. Centrifugal stress, therefore, is proportional to the square of the rotational speed of the turbine rotor.

Since the optimum power turbine speed is maintained during engine operation that is driving the mechanical loads at a constant speed, the amplitude of the centrifugal stress is close to zero during normal steady-state operation and is variable otherwise at operation start and upon operation completion. This is proportional to the change in turbine rotor speed from zero to the maximum engine rotational speed at start and down to zero when the engine is returned to standstill. The timing of these two major cycles of acceleration from standstill and coating to rest is well defined by the operational settings and does not vary from flight to flight. Rotational speed of the turbine shaft (RPM) is, however, monitored during the mission by the speed indicators for conformance. Centrifugal stresses, therefore, are consistent from flight to flight and are not included in the list of significant random variables. They will be indirectly accounted for in the constants of the PoF model.

b) *Thermal Loads*

The thermal loads are defined by the disk temperature throughout the mission, T_p, which is a function of temperature, velocity and pressure of the primary hot gas flow and the secondary cooling air flow. For a given engine installed on a specific aircraft model, perturbation analysis of the random variables in the heat transfer model showed two random variables to have a significant effect on the disk temperatures: BOT at startup and dwell time at idle before shutdown.

Minor cycles of disk temperature change, experienced during flight due to increased fuel flow at times of load increase, are considered negligible during steady state operation of the given aircraft (according to turbine inlet temperature measurements), and thus are not included in the list of significant random variables in this work.

- *Burner outlet temperature (BOT) during start*

  The BOT is also known as combustion section outlet temperature. The maximum combustion section outlet temperature (turbine inlet temperature) in this engine is above 1000°C (up to

1300°C). The highest temperature in any turbine engine is measured at the turbine inlet. Turbine inlet temperature, therefore, is considered to be one of the limiting factors in the operation a gas turbine engine. The temperature of a turbine section is monitored with the turbine inlet temperature gauge, turbine outlet temperature gauge, inter-stage turbine temperature gauge, and turbine gas temperature gauge.

The BOT, or the turbine inlet temperature (TIT), is a function of several operational variables such as the compressor discharge temperature, gas pressure and total fuel metered to the engine. Start BOT is a random variable assumed to be time independent for model simplification. This assumption is supported by the fact that the turbine inlet temperature is monitored and maintained within the allowable limits by adjusting several engine parameters. BOT for each start was obtained from TIT gauge readings.

- *Dwell time at idle before shutdown (D)*

Longer idle dwell times affect the thermal loading by allowing the temperatures to stabilize, thus decreasing the thermal gradient between the disk bore and rim during the startup and shutdown. Stress analysis showed that the dwell time at idle during start did not affect the major or minor stress cycles [139], [140]. The dwell time before shutdown was estimated by interviewing the aircrafts operators.

c) *Aerodynamic Forces (R)*

The aerodynamic forces are imposed on the surface of turbine blades by compressed airflow, and result in tangential force on the turbine wheel disks, R, which defines the rotor torque, T (a twisting force applied to the turbine shaft, see Equation A-26). The magnitude of the aerodynamic forces (and torque output) depends on the working temperature and the working pressure in a combustion chamber and the resulting air flow.

A turboprop engine turbine is required to produce torque for driving a propeller. The turbine torque output is measured by the torquemeter. Since constant a power turbine speed must be maintained during engine operation, the torquemeter measures the power applied to the turbine shaft, where power, W, is a product of torque, T, and rotational speed, ω (angular velocity):

$$W = \omega \cdot T = \omega \cdot r \cdot R$$

Equation A-48

Torquemeter readings are used to monitor tangential force on the turbine wheel disks, R.

While a constant turbine speed is maintained during engine operation, the amplitude of the tangential force on the turbine wheel disks, ΔR, is non-zero at times when load increase is required. The amplitude, ΔR, is the highest, however, at operation start and upon operation completion, resulting in a change in the turbine torque from zero to the required value at startup, and down to zero when the engine is returned to standstill. Tangential force amplitude, ΔR, during these two major cycles of acceleration (from standstill and coming to rest) will be considered in the fatigue model of the disk as one of the key sources of cyclic stress. Minor torque cycles of varying size that are experienced due to the load increase that is required to maintain a constant power turbine speed are considered to be negligible during the steady state operation of a given aircraft (according to the available torque measurements), and thus are not included in the list of significant random variables in this work. These cycles will be indirectly accounted for in the constants of the PoF model described below.

Based on the above considerations, the stress amplitude $\Delta\sigma$ is a function of BOT at start, the amplitude of tangential force on the turbine wheel disks at start-up, and the dwell time before shutdown:

$$\Delta\sigma = f\left(\Delta R, T_{IT}, D\right)$$

Equation A-49

According to the Paris law Equation A-41, the initial crack size ($a_{initialD}$) is also a significant random variable. To summarize the above, the fatigue life of a HP turbine disk as a function of the significant random variables can be written as:

$$N_D = f\left(\Delta R, T_{IT}, D, a_{initialD}, C_d, h\right)$$

Equation A-50

where N_D is the number of missions until failure, T_{IT} is the start BOT, D is the dwell time before shutdown, $a_{initialD}$ is the initial crack size, C_d and h are material constants.

Considering all sources of cyclic stresses during the mission, Equation A-41 takes a form:

$$\frac{da}{dN} = \left(\frac{da}{dN}\right)_{Start-up} + \left(\frac{da}{dN}\right)_{Shutdown}$$

Equation A-51

where start-up and shutdown components can be expanded further as:

$$\left(\frac{da}{dN}\right)_{Start-up} = C_d\left(\beta\sqrt{\pi a}\Delta\sigma_{Start-up}\right)^h = C_d\left[\beta\sqrt{\pi a}\left(D_1\Delta R + D_2 T_{IT}\right)\right]^h$$

Equation A-52

$$\left(\frac{da}{dN}\right)_{Shutdown} = C_d\left(\beta\sqrt{\pi a}\Delta\sigma_{Shutdown}\right)^h = C_d\left[\beta\sqrt{\pi a}\left(D_3 D^{-1}\right)\right]^h$$

Equation A-53

Substituting Equation A-52 and Equation A-53 into Equation E-51 will obtain:

$$\frac{da}{dN} = C_d\left(\beta\sqrt{\pi a}\right)^h\{(D_1\Delta R + D_2 T_{IT})^h + (D_3 D^{-1})^h\}$$

Equation A-54

Integrating both sides of the last equation, and considering one flight mission as a stress cycle for the disk, will obtain:

$$C_d\left(\beta\sqrt{\pi}\Delta\sigma_{eq}\right)^h \int_0^{N_D} dN = \int_{a_{initialD}}^{a_{Df}} a^{-h/2} da$$

Equation A-55

$$\Delta\sigma_{eq} = \left\{\left(D_1\Delta R + D_2 T_{IT}\right)^h + \left(D_3 D^{-1}\right)^h\right\}^{1/h}$$

Equation A-56

$$N_D = \frac{a_{initialD}^{\;1-(h/2)} - a_{Df}^{\;1-(h/2)}}{C_d\left(\beta\sqrt{\pi}\Delta\sigma_{eq}\right)^h\left(\frac{h}{2}-1\right)}, h \neq 2$$

Equation A-57

where N_D is the number of missions to failure, $a_{initialD}$ is the initial crack size in the disk, and a_{Df} is the crack size at failure. Since $h > 2$ for metals and $a_{initialD} << a_{Df}$ (according to the past experience), then $a_{initialD}^{\;1-(h/2)} >> a_{Df}^{\;1-(h/2)}$ and, considering Equation A-56, Equation A-57 can be written as following:

$$N_D = \frac{a_{initialD}^{\;1-(h/2)}}{C_d\left(\beta\sqrt{\pi}\right)^h\left(\frac{h}{2}-1\right)\{(D_1\Delta R + D_2T_{IT})^h + (D_3D^{-1})^h\}}, h > 2$$

Equation A-58

The number of missions to crack size a_{Di} is given by:

$$N_{Di} = \frac{a_{initialD}^{\;1-(h/2)} - a_{Di}^{\;1-(h/2)}}{C_d\left(\beta\sqrt{\pi}\right)^h\left(\frac{h}{2}-1\right)\{(D_1\Delta R + D_2T_{IT})^h + (D_3D^{-1})^h\}}, h > 2$$

Equation A-59

Equation A-59 can be solved for a_{Di} to obtain a relationship between the crack size and the number of accumulated stress cycles (at the time of disk inspection), N_{Di} (equal to accumulated aircraft missions M_i):

$$a_{Di} =$$

$$= \left[a_{initialD}^{\;1-(h/2)} - C_d\left(\beta\sqrt{\pi}\right)^h\left(\frac{h}{2}-1\right)\{(D_1\Delta R + D_2T_{IT})^h + (D_3D^{-1})^h\}M_i\right]^{\frac{1}{1-(h/2)}},$$

$$h > 2$$

Equation A-60

Stress cycle count, N, for the disk, defined as N_D, represents one aircraft mission, M, two flight hours long. The relation between the aircraft missions accumulated up to a certain time, M_i, and the stress cycle count for roller bearings, N_{Bi}, (Equation A-12), the shaft, N_{Si} (Equation A-35) and the disk, N_{Di}:

$$M_i = L + N_{Bi} = N_{Si} = N_{Di}$$

Equation A-61

While the amplitude of the tangential force on the turbine wheel disks, ΔR, BOT at start-up, T_{IT}, and dwell time at shutdown, D, are explicitly defined in Equation A-58 as random variables, material parameters C and h are also assigned to be random variables to indirectly account for insignificant variation of operating parameters during steady state phase of the mission, although all geometry and manufacturing tolerances of the particular disk are included in the constants D_1 to D_3. These are probabilistically defined within the framework of PoF-based agent autonomy, as described in Chapter 8 of this dissertation.

Bibliography

1. Azarkhail, M., “Agent Autonomy Approach to Physics-Based Reliability Modeling of Structures and Mechanical Systems”, PhD thesis, University of Maryland, College Park, 2007.

2. Azarkhail, M., Modarres, M., "The Evolution and History of Reliability Engineering: Rise of Mechanistic Reliability Modeling", International Journal of Performability Engineering, Volume 8, Issue 1, January 2012, Pages 35-47.

3. Stamatelatos, M., Dezfuli, H., “Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners”, NASA/SP-2011-3421, Second Edition, December 2011, NASA Headquarters, Washington DC.

4. Pearl, J., "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference", Morgan Kaufmann, 1988, revised second printing 2001.

5. Kurowicka, D., Cooke, R. M., "Uncertainty Analysis with High Dimensional Dependence Modelling", John Wiley & Sons, 2006.

6. Aldemir, T. et al., "Current State of Reliability Modelling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments", NUREG/CR-6901, U.S. Nuclear Regulatory Commission, Washington DC, 2006.

7. Aldemir, T. et al., "Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments", NUREG/CR-6942, U.S. Nuclear Regulatory Commission, Washington DC, 2007.

8. Rausand, M., Hsyland, A., "System reliability theory”, John Wiley & Sons, 2004.
9. Vesely, W. E., et al., 2002, “NASA Fault Tree Handbook with Aerospace Applications”, National Aeronautics and Space Administration (NASA), Washington DC.
10. Epstein, B., Weissman, I., "Mathematical models for system reliability", Taylor & Francis, 2008.
11. MIL-HDBK-338B, "Electronic Reliability Design Handbook", 1 October 1998.
12. Lisnianski, A., "The Markov Reward Model for a Multi-State System Reliability Assessment with Variable Demand", Quality Technology & Quantitative Management, Vol. 4, No. 2, 2007, Pages 265-278.
13. Trivedi, K.S., Malhotra, M., Fricks, R.M., "Markov reward approach to performability and reliability analysis", Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 1994, MASCOTS '94, Proceedings of the Second International Workshop on, Publication Year: 1994 , Pages 7-11.
14. Labeau, P.E. et al., "Dynamic reliability: towards an integrated platform for probabilistic risk assessment", Reliability Engineering & System Safety, Volume 68, Issue 3, June 2000, Pages 219-254.
15. Matsuoka, T., Kobayashi, M., "The GO-FLOW reliability analysis methodology - analysis of common cause failures with uncertainty", Nuclear Engineering and Design, Volume 175, Issue 3, November 1997, Pages 205-214.

16. Hu, Y., Modarres, M., "Evaluating System Behavior Through Dynamic Master Logic Diagram (DMLD) Modelling", Reliability Engineering and System Safety, Vol. 64, 1999, Pages 241-269.

17. Kae-Sheng, H., Mosleh, A., "The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants", Volume 52, Issue 3, June 1996, Pages 297-314.

18. Aldemir, T. et al., "Probabilistic risk assessment modelling of digital instrumentation and control systems using two dynamic methodologies", Reliability Engineering & System Safety, Volume 95, Issue 10, October 2010, Pages 1011-1039.

19. Hakobyan, A. et al., "Dynamic generation of accident progression event trees", Nuclear Engineering and Design, Volume 238, Issue 12, December 2008, Pages 3457-3467.

20. Hofer, E. et al., "Dynamic Event Trees for Probabilistic Safety Analysis", GRS, Garsching, Germany, 2004.

21. Devooght, J., Smidts, C., "Probabilistic Reactor Dynamics l: The theory of continuous event trees", Nuclear Science and Engineering, Volume 111, 1992, Pages 229-240.

22. Izquierdo J.M. et al., "Relationship between probabilistic dynamics and event trees", Reliability Engineering & System Safety, Volume 52, Issue 3, June 1996, Pages 197-209.

23. Devooght, J., Smidts, C., "Probabilistic dynamics as a tool for dynamic PSA", Reliability Engineering and System Safety, Volume 52, Issue 3, June 1996, Pages 185-196.

24. Cojazzi, G., "The DYLAM approach for the dynamic reliability analysis of systems", Reliability Engineering & System Safety, Volume 52, Issue 3, June 1996, Pages 279-296.

25. Acosta, C., Siu, N., "Dynamic event trees in accident sequence analysis: application to steam generator tube rupture", Reliability Engineering & System Safety, Volume 41, Issue 2, 1993, Pages 135-154.

26. Marseguerra, M., Zio E., "Monte Carlo approach to PSA for dynamic process systems", Reliability Engineering & System Safety, Volume 52, Issue 3, June 1996, Pages 227-241.

27. Catalyurek, U. et al., "Development of a code-agnostic computational infrastructure for the dynamic generation of accident progression event trees", Reliability Engineering & System Safety, Volume 95, Issue 3, 2010, Pages 278-294.

28. Rutt, B. et al., "Distributed Dynamic Event Tree Generation for Reliability and Risk Assessment", Challenges of Large Applications in Distributed Environments, 2006 IEEE, Pages: 61-70.

29. Tombuyses, B., Aldemir, T., "Computational efficiency of the continuous cell-to-cell mapping technique as a function of integration schemes", Reliability Engineering & System Safety, Volume 58, Issue 3, December 1997, Pages 215-223.

30. Aldemir, T., "Computer-Assisted Markov Failure Modeling of Process Control Systems", IEEE Transactions on Reliability, Volume R-36, Issue 1, April 1987, Pages 133-144.

31. Belhadj, M., Aldemir, T., "Some computational improvements in process system reliability and safety analysis using dynamic methodologies", Reliability Engineering & System Safety, Volume 52, Issue 3, June 1996, Pages 339-347.

32. Cacciabue, P.C., et al., “Dynamic logical analytical methodology versus fault tree: The case of auxiliary feedwater system of a nuclear power plant”, Nuclear Technology, Volume 74, Issue 2, August 1986, Pages 195-208.

33. Marchand, S. et al., "DDET and Monte Carlo simulation to solve some dynamic reliability problems", Proc. of PSAM-4, Volume 3, 1998, Pages 2055-2060.

34. Merrill, B. et al., "A recent version of MELCOR for fusion safety applications", Fusion Engineering and Design, Volume 85, Issues 7-9, December 2010, Pages 1479-1483.

35. Hakobyan, A. et al., "A Methodology for Generating Dynamic Accident Progression Event Trees for Level-2 PRA", PHYSOR-2006, ANS Topical Meeting on Reactor Physics, Vancouver, BC, Canada, 2006 September 10-14.

36. Chang, Y.H.J., Mosleh, A., "Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents", Part 1 - Part 5, Reliability Engineering & System Safety, Volume 92, Issue 8, 2007, Pages 997-1101.

37. Zio, E., “Reliability engineering: Old problems and new challenges”, Reliability Engineering and System Safety, 94 (2009), 125-141.

38. Liu Y., Zuo M.J., Huang H.Z., "Dynamic reliability assessment for multi-state deteriorating systems", Chemical Engineering Transactions, 33, 2013, Pages 535-540.

39. Letot C., Dehombreux P., "Dynamic reliability degradation based models and maintenance optimization", Proceedings of the 9th National Congress on Theoretical and Applied Mechanics, Brussels (Belgium), May 2012.

40. Nourelfatha, M., Châteletb, E., Nahasa, N., "Joint redundancy and imperfect preventive maintenance optimization for series–parallel multi-state degraded systems", Reliability Engineering & System Safety, Volume 103, July 2012, Pages 51-60.

41. Soro, I.W., Nourelfath, M., Aït-Kadi D., "Performance evaluation of multi-state degraded systems with minimal repairs and imperfect preventive maintenance", Reliability Engineering & System Safety, Volume 95, Issue 2, February 2010, Pages 65-69.

42. Li, W., Pham, H., "Reliability modeling of multi-state degraded systems with multi-competing failures and random shocks", IEEE Transactions on Reliability, Vol. 54, No. 2, June 2005, Pages 297-303.

43. Mohaghegh, Z., Modarres, M., "A Probabilistic Physics-Of-Failure Approach to Common Cause Failures in Reliability Assessment of Structures and Components", American Nuclear Society, 2011 Winter Meeting, October 30-November 3, 2011.

44. Mohaghegh, Z., Modarres, M., Christou, A., "Physics-based Common Cause Failure Modeling in Probabilistic Risk Analysis: A Mechanistic Perspective",

Proceedings of the ASME 2011 Power Conference, POWER2011, July 12-14, 2011, Denver, Colorado, USA.

45. Iamsumang, C., “Computational Algorithm for Dynamic Hybrid Bayesian Network in On-line System Health Management Applications”, PhD thesis, University of Maryland, College Park, 2014.

46. Luck, M., Ashri, R., d'Inverno, M., 2004, “Agent-Based Software Development”, Artech House Publishers, London, UK.

47. Odell, J., 2002, “Objects and Agents Compared”, Journal of Object technology, Vol. 1, No. 1, Pages 41-53.

48. Sage, A., Rouse, W., “Handbook of Systems Engineering and Management”, John Wiley & Sons, 2009.

49. Urbano, P. de A. et al., “Introducing Reliability and Real-Time Features in Flexible Agent-Oriented Automation Systems”, Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on, June 2004, Pages 89-94.

50. Labarthe, O. et al., “Toward a methodological framework for agent-based modelling and simulation of supply chains in a mass customization context”, Simulation Modelling Practice and Theory, Vol. 15, 2007, Pages 113-136.

51. Kaegi, M., Mock, R., Kroger, W., ”Analyzing maintenance strategies by agent-based simulations: A feasibility study”, Reliability Engineering & System Safety, Volume 94, Issue 9, September 2009, Pages 1416-1421.

52. Wang, M., Nowostawski, M., Purvis, M., “Declarative agent programming support for a FIPA-Compliant agent platform”, ProMAS'05 Proceedings of the

Third international conference on Programming Multi-Agent Systems, Springer-Verlag Berlin, Heidelberg, 2006, Pages 252-266.

53. Šmídl, V., "Bayesian Approach to Multi-Agent Systems", Institute of Information Theory and Automation, Prague, Czech Republic, 2005.

54. Rabbat, R. R., "Bayesian Expert Systems and Multi-Agent Modeling for Learner-Centric Web-Based Education", PhD Dissertation, Massachusetts Institute of Technology, MA, 2005.

55. Gowadia, V., Farkas, C., Valtorta, M., "PAID: A Probabilistic Agent-Based Intrusion Detection system", Computers & Security, 2005, Vol. 24, Issue 7, Pages 529-545.

56. Santos, E.R., Fagundes, M.S., Vicari, R.M., "An Ontology-Based Approach to Interoperability for Bayesian Agents", the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 07), Pages 1304-1306.

57. "Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence", edited by Gerhard Weiss, The MIT Press, Cambridge, Massachusetts, USA, 2000.

58. Xiang, Y., "Probabilistic reasoning in multiagent systems: a graphical models approach", Cambridge: Cambridge University Press, 2002.

59. Castelfranchi, C., Falcone, R., "Towards a theory of delegation for agent-based systems", Robotics and Autonomous Systems, Vol. 24, 1998, Pages 141-157.

60. Maomi Ueno (2010), "Bayesian Agent in E-learning", in "E-learning Experiences and Future", Edited by Safeeullah Soomro, ISBN: 978-953-307-

092-6, InTech. Available from: http://www.intechopen.com/books/e-learningexperiences-and-future/bayesian-agent-in-e-learning

61. Ueno, M., Okamoto, T., “Bayesian Agent in e-Learning”, Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on, Publication Year: 2007, Pages 282-284.
62. Lin, F., Norrie, D., “Schema-based conversation modeling for agent-oriented manufacturing systems”, Computers in Industry, Vol. 46, 2001, Pages 259-274.
63. Tian, Y. et al., “Modeling errors in daily precipitation measurements: Additive or multiplicative?”, Geophysics Research Letters, Vol. 40, Issue 10, 2013, Pages 2060-2065.
64. Marseguerra, M. et al., "A concept paper on dynamic reliability via Monte Carlo simulation", Mathematics and Computers in Simulation, Volume 47, Issues 2–5, August 1998, Pages 371-382.
65. Rubinstein, R. Y., Kroese, D. P., "Simulation and the Monte Carlo Method", 2nd Edition, John Wiley & Sons, 2008.
66. Rinne, H., “The Weibull distribution: a handbook”, Taylor & Francis, 2009.
67. Nelson, W., “Applied life data analysis”, John Wiley & Sons, 1982.
68. Ushakov, I., “Handbook of reliability engineering”, John Wiley & Sons, 1994.
69. Modarres, M., Kaminskiy, M. & Krivtsov, V., "Reliability engineering and risk analysis: a practical guide", Taylor & Francis, 2009.
70. Meeker, W.Q., Escobar, L.A., "Statistical methods for reliability data", 1st Edition, John Wiley and Sons, 1998.

71. O'Connor, A.N., “Probability Distributions Used in Reliability Engineering”, RiAC, 2011.

72. Nelson, W., "Accelerated testing: statistical models, test plans and data analysis", Wiley Series in Probability and Statistics, John Wiley & Sons, 1990.

73. MINITAB 16 Statistical Software, Tutorials.

74. Box, G.E.P., Jenkins, G.M., “Time Series Analysis: Forecasting and Control”, 1994, 3rd Edition, Prentice Hall.

75. Hamilton, J.D., “Time Series Analysis”, Princeton University Press, 1994.

76. Zhang, H., Kang, R., Pecht, M., “A hybrid prognostics and health management approach for condition-based maintenance”, Industrial Engineering and Engineering Management, 2009, IEEM 2009, IEEE International Conference on, Dec. 2009, Pages 1165-1169.

77. Pecht, M., “Prognostics and Health Management of Electronics”, John Wiley & Sons, 2008.

78. Rasmussen, C. E., Williams, C. K. I., “Gaussian Processes for Machine Learning”, the MIT Press, 2006.

79. Cheng S., Pecht, M., “Multivariate State Estimation Technique for Remaining Useful Life Prediction of Electronic Products”, AAAI Fall Symposium on Artificial Intelligence for Prognostics, Arlington, VA, Nov, 2007, Pages 26-32.

80. Chin, S.C., Ray, A., Rajagopalan, V., “Symbolic time series analysis for anomaly detection: A comparative evaluation”, Signal Processing 85 (2005), Pages 1859-1868.

81. Khatkhate, A. et al., “Symbolic time-series analysis for anomaly detection in mechanical systems”, Mechatronics, IEEE/ASME Transactions on, Volume 11, Issue 4, 2006 , Pages 439-447.

82. Sotiris V., Pecht M., “Support Vector Prognostics Analysis of Electronic Products and Systems”, AAAI Fall Symposium on Artificial Intelligence for Prognostics, Arlington, VA, Nov, 2007, Pages 120-127.

83. Tobon-Mejia, D.A. et al., “Hidden Markov Models for failure diagnostic and prognostic”, Prognostics and System Health Management Conference (PHM-Shenzhen), 2011, Pages 1-8.

84. Gustafsson F. et al., “Particle Filters for Positioning, Navigation and Tracking”, IEEE Transactions on Signal Processing, Feb 2001.

85. Kumar, S., Sotiris, V., Pecht, M., “Health Assessment of Electronic Products using Mahalanobis Distance and Projection Pursuit Analysis”, International Journal of Computer, Information, Systems Science, and Engineering, Vol. 2, Issue 4, 2008, Pages 242-250.

86. Saha, B. et al., “Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework”, Instrumentation and Measurement, IEEE Transactions on, Volume 58, Issue 2, 2009, Pages 291-296.

87. Zio, E., Di Maio, F., “Fatigue crack growth estimation by relevance vector machine”, Expert Systems with Applications, Volume 39, Issue 12, 2012, Pages 10681-10692.

88. Niu, G. et al., “Health monitoring of electronic products based on Mahalanobis distance and Weibull decision metrics”, Microelectronics Reliability, Volume 51, Issue 2, February 2011, Pages 279-284.
89. Vichare, N., Pecht, M., “Prognostics and Health Management of Electronics”, IEEE Transactions on Components and Packaging Technologies, Vol. 29, No. 1, March 2006.
90. Tian, Z., Zuo, M.J., “Health condition prognostics of gears using a recurrent neural network approach”, Annual Reliability and Maintainability Symposium (RAMS) Proceedings, 2009, Pages 460-465.
91. Kolodziejczyk, T., "A Methodological Approach Ball Bearing Damage Prediction under Fretting Wear Conditions," 2008 4th International IEEE Conference "Intelligent Systems".
92. Wayne, M., Modarres, M. “Modeling Rate of Occurrence of Failures with Log-Gaussian Process Models: A Case Study for Prognosis and Health Management of a Fleet of Vehicles”, International Journal of Performability Engineering, Vol. 9, No. 6 (2013), pages 701-714.
93. Gelman, A. et al., "Bayesian data analysis", Chapman & Hall, 2004.
94. Hamada, M. et al., "Bayesian reliability", Springer, 2008.
95. Groen, G. J., Droguett, E. L., “Bayesian estimation of the variability of reliability measures”, Annual Reliability and Maintainability Symposium (RAMS) Proceedings, 2003, Pages 182-187.
96. Chookah, M. et al., "A probabilistic physics-of-failure model for prognostic health management of structures subject to pitting and corrosion-fatigue",

Reliability Engineering & System Safety, Volume 96, Issue 12, December 2011, Pages 1601-1610.

97. Chookah, M., “Structuring a probabilistic model for reliability evaluation of piping subject to corrosion-fatigue degradation”, PhD Dissertation, University of Maryland, College Park, MD, 2009.

98. Vachtsevanos, G.J. et al., “Intelligent fault diagnosis and prognosis for engineering systems”, John Wiley & Sons, 2006.

99. Rabiei, M., “A Bayesian framework for structural health management using acoustic emission monitoring and periodic inspections”, PhD thesis, University of Maryland, College Park, MD, 2011.

100. Welch, G., Bishop, G., “An introduction to the Kalman filter”, University of North Carolina, Chapel Hill, NC, 1995.

101. Simon, D., “Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches”, Wiley-Interscience, 2006.

102. Modarres, M., “Risk Analysis in Engineering: Techniques, Tools, and Trends”, Taylor & Francis, 2006.

103. Satelli, A., “Sensitivity analysis for importance assessment”, Risk Analysis, Vol. 22, No. 3, 2002, Pages 579-590.

104. Borgonovo, E., “A new uncertainty importance measure”, Reliability Engineering & System Safety, Vol. 92, Issue 6, 2007, Pages 771-784.

105. Borgonovo, E., “Measuring uncertainty importance: investigation and comparison of alternative approaches”, Risk Analysis, Vol. 26, No. 5, 2006, Pages 1349-1361.

106. Tang, Y. et al., “Comparing sensitivity analysis methods to advance lumped watershed model identification and evaluation”, Hydrol. Earth Syst. Sci., 11, 793-817, doi:10.5194/hess-11-793-2007, 2007.

107. Oakley, J., O’Hagan, A., “Probabilistic sensitivity analysis of complex models: a Bayesian approach”, Journal of Royal Statistical Society Series B - Statistical Methodology, 66, Part 3, 2004, pages 751-769.

108. Helton, J.C. et al., "Survey of Sampling-Based Methods for Uncertainty and Sensitivity Analysis", Sandia Report SAND2006-2901, Sandia National Laboratories, 2006.

109. Helton, J.C., Davis, F.J., "Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems", Sandia Report SAND2001-0417, Sandia National Laboratories, 2001.

110. Montgomery, D.C., "Design and Analysis of Experiments", John Wiley & Sons, 1997.

111. Downing, D.J. et al., "An Examination of Response-Surface Methodologies for Uncertainty Analysis in Assessment Models", Technometrics, 1985, Vol. 27, No. 2, pages 151-163.

112. Saltelli, A. et al., "Sensitivity Analysis in Practice" John Wiley & Sons, 2004.

113. Satelli, A., “Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index.”, Computer Physics Communications, Volume 181, 2010, Pages 259-270.

114. Glen, G., Isaacs, K., “Estimating Sobol sensitivity indices using correlations”, Environmental Modelling & Software, Volume 37, November 2012, Pages 157-166.

115. Ikonen, T., Tulkki, V., “The importance of input interactions in the uncertainty and sensitivity analysis of nuclear fuel behavior”, Nuclear Engineering and Design, Volume 275, August 2014, Pages 229-241.

116. Pourgol-Mohamad, M., Mosleh, A., Modarres, M., “Structured treatment of model uncertainty in complex thermal-hydraulics codes: Technical challenges, prospective and characterization”, Nuclear Engineering and Design, Volume 241, Issue 1, January 2011, Pages 285-295.

117. Pourgol-Mohamad, M., “Thermal–hydraulics system codes uncertainty assessment: A review of the methodologies”, Annals of Nuclear Energy, Volume 36, Issues 11-12, November-December 2009, Pages 1774-1786.

118. “The Gas Turbine Handbook”, U.S. Department of Energy, Office of Fossil energy, national Energy Technology Laboratory, 2006.

119. Sony, M., Pecht, M., “Prognostics of Systems: Approaches and Applications”, 24th International Conference on Condition Monitoring and Diagnostics Engineering Management, 30th May to 1st June 2011, Stavanger, Norway.

120. Lunn, D. et al., “The BUGS Book: A Practical Introduction to Bayesian Analysis”, Chapman and Hall, 2012.

121. Congdon, P., “Bayesian Statistical Modelling”, John Wiley, 2006.

122. “Oracle Crystal Ball and Minitab”, an Oracle White Paper, June 2011.

123. Azarkhail, M., Modarres, M., “Agent Autonomy Approach to Risk Assessment of Complex Dynamic Systems", The Journal of the Reliability Information Analysis Center, 2007 Q3, Pages 13-20.

124. Azarkhail, M., Modarres, M., “An Intelligent-Agent-Oriented Approach to Risk Analysis of Complex Dynamic Systems with Applications in Planetary Missions”, Proc. of the Eighth International Conference on Probabilistic Safety Assessment and Management (PSAM), ASME, New Orleans, USA, 2006.

125. Sadeghi, F. et al., “A Review of Rolling Contact fatigue”, Transactions of the ASME - Journal of Tribology, Vol. 131, October 2009.

126. NSWC-11, "Handbook of Reliability Prediction Procedures for Mechanical Equipment", Naval Surface Warfare Center, Carderock Division, May 2011.

127. Marble, S., Morton, B.P., “Predicting the remaining life of propulsion system bearings”, Proceedings of the 2006 IEEE Aerospace Conference, Big Sky, MT, USA.

128. Roemer, M.J., Byington, C.S., Sheldon, J., “Advanced Vibration Analysis to Support Prognosis of Rotating Machinery Components”, International Journal of COMADEM, Volume 11, Issue 2, 2008, Pages 2-11.

129. ANSI/ABMA 11–1990, “Load Ratings and Fatigue Life for Roller Bearings,” The Anti-Friction Bearing Manufacturers Association, Washington, DC.

130. Shimizu, S. et al., “New data analysis of probabilistic stress-life (P–S–N) curve and its application for structural materials”, International Journal of Fatigue, Volume 32, Issue 3, 2010, Pages 565-575.

131. Kotzalas, M., Harris, T. A., "Fatigue Failure Progression in Ball Bearings", Transactions of the ASME - Journal of Tribology, Vol. 123, 2001, Pages 238-242.

132. Xu, G., Sadeghi, F., “Spall Initiation and Propagation Due to Debris Denting”, Wear, Volume 201, 1996, Pages 106-116.

133. Qian, J., Fatemi, A., "Mixed Mode Fatigue Crack Growth: A Literature Survey," Engineering Fracture Mechanics, Vol. 55, No.6, 1996, Pages 969-990.

134. Fonte, M. et al., “The effect of steady torsion on fatigue crack growth in shafts”, International Journal of Fatigue, Vol. 28, Issues 5–6, 2006, Pages 609-617.

135. Stephens, R.I. et al., “Metal Fatigue in Engineering”, John Wiley & Sons, 2000.

136. Paulus, M.E., “Fatigue Damage Accumulation Due to Complex Random Vibration Environments: Application to Single-Axis and Multi-Axis Vibration”, PhD Dissertation, University of Maryland, College Park, MD, 2011.

137. Steinberg, D.S., “Vibration analysis for electronic equipment”, John Wiley & Sons, 2000.

138. RTCA/DO-160F “Environmental Conditions and Test Procedures for Airborne Equipment”, prepared by RTCA (Radio Technical Commission for Aeronautics) Special Committee 135, issued on December 6, 2007.

139. Tryon, R., Cruse, T., Mahadevan, S., “Development of a reliability-based fatigue life model for gas turbine engine structures”, Engineering Fracture Mechanics, Volume 53, Issue 5, March 1996, Pages 807-828.

140. NASA/CR-97-206215, “Fatigue Reliability of Gas Turbine Engine Structures”, Lewis Research Center, October 1997.

141. Tong, J. et al., "Creep, fatigue and oxidation in crack growth in advanced nickel base superalloys", International Journal of Fatigue, Vol. 23, Issue 10, 2001, Pages 897-902.

142. Starink, M.J., Reed, P.A.S., "Thermal activation of fatigue crack growth: Analysing the mechanisms of fatigue crack propagation in superalloys", Materials Science and Engineering, Vol. 491, Issues 1-2, 2008, Pages 279-289.