Adaptive, State-Based Modeling Applied to Prognostics and Health Management of Industrial Rotating Equipment

by

Maxwell Toothman

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2023

Doctoral Committee:

Professor Kira Barton, Co-Chair
Professor Dawn M. Tilbury, Co-Chair
Research Associate Professor Theodor Freiheit
Professor Judy Jin
Associate Research Scientist James Moyne

Maxwell Toothman
toothman@umich.edu
ORCID iD: 0000-0001-7563-4831

ACKNOWLEDGEMENTS

I would like to thank my advisors, Professor Kira Barton and Professor Dawn M. Tilbury, for their guidance and support throughout my time as a Ph.D. student. They were instrumental in my development as a student, researcher, and leader over the last several years. I am extremely grateful for their kindness and empathy in addition to their technical advice. James Moyne was similarly influential for my research, and I am appreciative of the numerous insights that he provided. I would also like to thank the other members of my committee, Professor Theodor Freiheit and Professor Judy Jin, for providing feedback that helped me to improve my research and dissertation.

Additionally, I would like to thank the members of the Barton Research Group and the Tilbury Research Group for fostering a collaborative and uplifting community. In particular, Ilya Kovalenko and Efe Balta served as excellent mentors during my first years as a graduate student, and I would not have achieved this milestone without their support and guidance. I am also grateful for the help from Marcus Badgett and the student members of our lab's Multidisciplinary Design Program team, which helped me to develop many of the case studies in this dissertation.

A University Partnership Initiative with The Dow Chemical Company made the work presented here possible, and I would like to thank the leaders of this industry collaboration: Birgit Braun, Scott J. Bury, and Yixin Ye. They provided feedback that maximized the impact of this research in addition to supporting the project financially. They also sponsored and supervised my internship, which was a great experience that allowed me to implement my research in industry.

Finally, I would like to thank my friends and family for supporting me during my time in graduate school. Their backing made Ann Arbor and the University of Michigan feel like a new home and gave me the confidence to pursue my goals. I am especially grateful for my girlfriend, Bridget Shayka, and my parents for their continual support and encouragement throughout this experience.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

LIST OF ABBREVIATIONS

AAS asset administration shell

CDF cumulative distribution function

DT digital twin

ERS equipment reliability strategy

EWMA exponentially-weighted moving average

FMEA failure mode and effects analysis

FMSA failure mode symptoms analysis

GMM Gaussian mixture model

GPM general path model

HDR high density region

HMM hidden Markov model

HVC hierarchical variable clustering

IHS instantaneous health state

IMS intelligent maintenance systems

IoT internet of things

ISO International Organization for Standardization

LDA linear discriminant analysis

LDPE low-density polyethylene

LSTM long short-term memory network

MAP maximum a posteriori

MAPE mean absolute percent error

MTTF mean time to failure

NIST National Institute of Standards and Technology

PDF probability density function

PHM prognostics and health management

PHS predictive health state

RMS root mean square

RMSE root-mean-square error

RNN recurrent neural network

RPM rotations per minute

RUL remaining useful life

SDLC system development life cycle

SME subject matter expert

TTF time to failure

ABSTRACT

Unplanned downtime due to equipment health problems imposes significant costs on many manufacturing operations in the form of maintenance expenditures and lost production. To reduce these costs, companies have become interested in leveraging the latest smart manufacturing technologies to implement predictive maintenance strategies. The field of prognostics and health management research seeks to support this transition by developing modeling and analysis tools that provide manufacturers with real-time feedback on the health of their equipment throughout its life cycle. Based on this information, companies can minimize production interruptions by administering maintenance procedures only when necessary and at opportune times.

Predictive maintenance strategies rely on machine health models to derive actionable insights on the current and future health of machines from sensor measurements. The ability to quickly develop and deploy accurate health models is then critical for the advancement of predictive maintenance across the manufacturing industry. However, creating these models in manufacturing applications can be challenging due to numerous disturbances that commonly impact machines, lack of sufficient data to characterize machine behavior leading up to health events, and uncertainty surrounding machine degradation processes.

This dissertation proposes health modeling frameworks that address these challenges to facilitate more widespread adoption of predictive maintenance strategies. Novel, state-based modeling methods are developed to allow machine health models to maintain a memory of recent sensor measurements and prior modeling results when analyzing system health. This approach is well-suited for dynamic manufacturing environments and supports representations of multi-stage degradation processes. An extensible digital twin framework provides the tools to model the health of complex systems with hierarchies of standardized, reusable digital twins. An adaptive modeling framework is also proposed to probabilistically detect and diagnose ongoing degradation processes based on machine signal trends while simultaneously monitoring for unforeseen degradation modes. Multiple case studies with different types of rotating equipment demonstrate how the contributions of this dissertation allow manufacturers to develop standardized machine health models for a broad range of applications to fully realize the cost and production benefits of predictive maintenance.

CHAPTER 1

Introduction

1.1 Motivation

Manufacturers across industries are experiencing growing production demand due to a variety of global economic forces [1, 2]. The inevitable lag in the deployment of new capital has led companies to strive for increased efficiencies from existing resources in response to this growing demand. Emerging smart manufacturing and Industry 4.0 innovations have provided companies with a suite of new tools to reconfigure existing operations to be more efficient, robust, and flexible. 3D printing systems, virtual commissioning platforms, and cloud computing services are just a few of the technologies currently revolutionizing the manufacturing industry. Equipment health problems though, still represent a serious threat to production and efficiency goals. Unexpected machine failure and subsequent repair procedures can significantly reduce factory production rates along with imposing substantial material and labor costs [3, 4]. A recent survey from the National Institute of Standards and Technology (NIST) estimates that maintenance-related problems cost the manufacturing industry $220 billion in 2016, with $105 billion of that total attributed to lost sales [5].

Maintenance strategies, defined as policies for reducing costs and downtime that result from equipment health problems, vary significantly between manufacturing operations but are generally described as either reactive, preventative, or predictive [6]. Reactive maintenance, also referred to as corrective maintenance or run-to-failure, is the oldest type of maintenance strategy and is characterized by repairing equipment only when it is rendered non-operational by a health problem [7, 8]. The risk of catastrophic machine failure is relatively high under a reactive maintenance strategy, which can lead to costly repairs and long periods of downtime [9]. It is also difficult to anticipate the length of downtime periods and create appropriate maintenance budgets under a reactive maintenance strategy. These issues have led manufacturers to adopt preventative maintenance strategies, which remain popular today. Preventative maintenance strategies involve carrying out maintenance procedures and overhauling equipment according to time or production-based schedules [10]. This approach is effective for preventing catastrophic failure, but manufacturers tend to implement overly-conservative schedules that prescribe frequent, often unnecessary, maintenance procedures. As a result, preventative maintenance strategies can still impose significant costs and downtime on operations [7].

Recent advances in Industry 4.0 and smart manufacturing technologies have paved the way for industrial manufacturers to reduce downtime even further by implementing predictive maintenance strategies [11, 12]. Certain works in the literature use the terms condition-based maintenance, proactive maintenance, and prescriptive maintenance to refer to similar approaches. Under a predictive maintenance strategy, maintenance and repair procedures are carried out based on estimates of current and future equipment health states [13, 14]. The desired outcome is the ability to recognize health problems as they emerge and either resolve them immediately, via operational changes or low-cost maintenance procedures, or to schedule maintenance at an opportune time in the near future. Ideally, these decisions are informed by diagnoses about the root cause of a health problem, predictions of when problems will worsen or culminate in equipment fault, and prescriptions for how to resolve problems. The

NIST study cited above compared the maintenance costs of manufacturers that utilize predominantly reactive, preventative, and predictive maintenance strategies [5]. While the direct maintenance costs of preventative and predictive manufacturers were 81.7% higher than those of reactive manufacturers, these manufacturers experienced 52.7% less unplanned downtime and 78.5% fewer defects compared to reactive manufacturers. Further investigation showed that predictive manufacturers experienced 18.5% less downtime and 87.3% fewer defects than preventative manufacturers.

The field of prognostics and health management (PHM) research seeks to develop methods that facilitate predictive maintenance strategies [15, 16, 17]. PHM research includes a variety of approaches for detecting anomalies, diagnosing degradation, and predicting equipment faults based on real-time data streams from machine sensors, referred to here as machine signals [18, 19]. A promising application for PHM research is industrial rotating equipment (such as pumps, turbines, and compressors), a class of machines that is commonly used to support processing of liquid and gaseous material in manufacturing operations [20, 21, 22]. These types of machines often operate continuously for long periods of time, which means that machines are unaffected by frequent startups and shutdowns and machine sensors can be sampled regularly to monitor equipment health. The work presented here has been implemented and validated with this class of machines.

Despite the potential cost and production benefits, manufacturers have been slow to deploy PHM solutions, defined as implementations of concepts and methods from PHM research, and adapt their maintenance strategies to make use of the information that these solutions provide. One study reports that only 24% of oil and natural gas companies describe their maintenance approaches as predictive [23]. This lag is not caused by a poor understanding of industrial equipment or insufficient modeling technology. Indeed, a wealth of research has documented successful degradation detection and fault prediction results from case studies [12, 16, 22]. Rather, some common limitations of existing PHM solutions tend to hinder their feasibility and success in industrial applications [24, 25]. Two of these limitations will be addressed in this work.

First, existing PHM solutions often rely on "snapshot" analysis, diagnosing and predicting machine health based only on the magnitudes of the most recent machine sensor measurements. This type of approach can be successful for academic case studies, where PHM solutions are trained and tested during controlled, uninterrupted experiments. In contrast, manufacturing environments are highly uncontrolled, with common disturbances, such as changes in operating parameters, that can shift machine signal baselines. Snapshot-based PHM solutions can quickly become outdated in industrial applications if frequent model retraining is not carried out. These solutions also have difficulty tracking multi-stage degradation processes, which can arise as machine dynamics and component interactions change over time.

The second limitation stems from the fact that degradation and failure are relatively rare in individual machines, resulting in limited availability of failure events to model degradation and predict failure. Thus, manufacturers need to implement predictive maintenance strategies across many of the machines in a factory to fully realize their benefits. However, existing PHM solutions are primarily developed for individual machines, components, or health problems, with architectures and outputs that are specific to a single application. As a result, it is not clear how aspects of PHM solutions can be reused across equipment fleets, which feature many similar pieces of equipment operating alongside one another. It is also difficult to combine PHM solutions to model complex machines that have multiple components, each of which is susceptible to different health problems. These capabilities are critical for manufacturers interested in implementing predictive maintenance strategies at a factory-wide scale.

This dissertation develops modeling strategies that facilitate the deployment of

PHM solutions. The application environment in which these strategies are developed and demonstrated is industrial rotating equipment. Future work can extend the underlying concepts and approaches to other types of equipment as the basic aspects of the developed strategies are not specific to rotating equipment. The novel modeling methods and frameworks proposed here employ state-based analysis of machine signals to generate health state estimates and predictions that are informed by recent observations and a machine's current degradation stage (when degradation is known to be a multi-stage process). Additionally, a digital twin framework for fault monitoring introduces standardized processes for combining and reusing individual health models across machines.

1.2 Contributions

The contributions of this dissertation are described below and illustrated in Fig. 1.1.

1. *A general operating model for state-based PHM*

State-based PHM represents an alternative to the snapshot-based approach taken by most existing PHM solutions. With a state-based approach, information from historical machine sensor measurements and maintenance events is retained and influences the analysis of the most recent machine sensor measurements. PHM solutions can also be structured so that different health models are used to predict faults based on estimates of a system's current degradation stage and mode, as depicted in Fig. 1.1. A few recent works have utilized state-based analysis in individual PHM solutions [26, 27, 28, 29, 30, 31, 32], but a general framework for implementing this approach has not been proposed. Therefore, the first contribution of this dissertation is a discrete-state operating model for industrial equipment, designed to facilitate the development of state-based PHM solutions. The model includes a hierarchical set of online states that describe machine health with respect to quantitative health specifi-

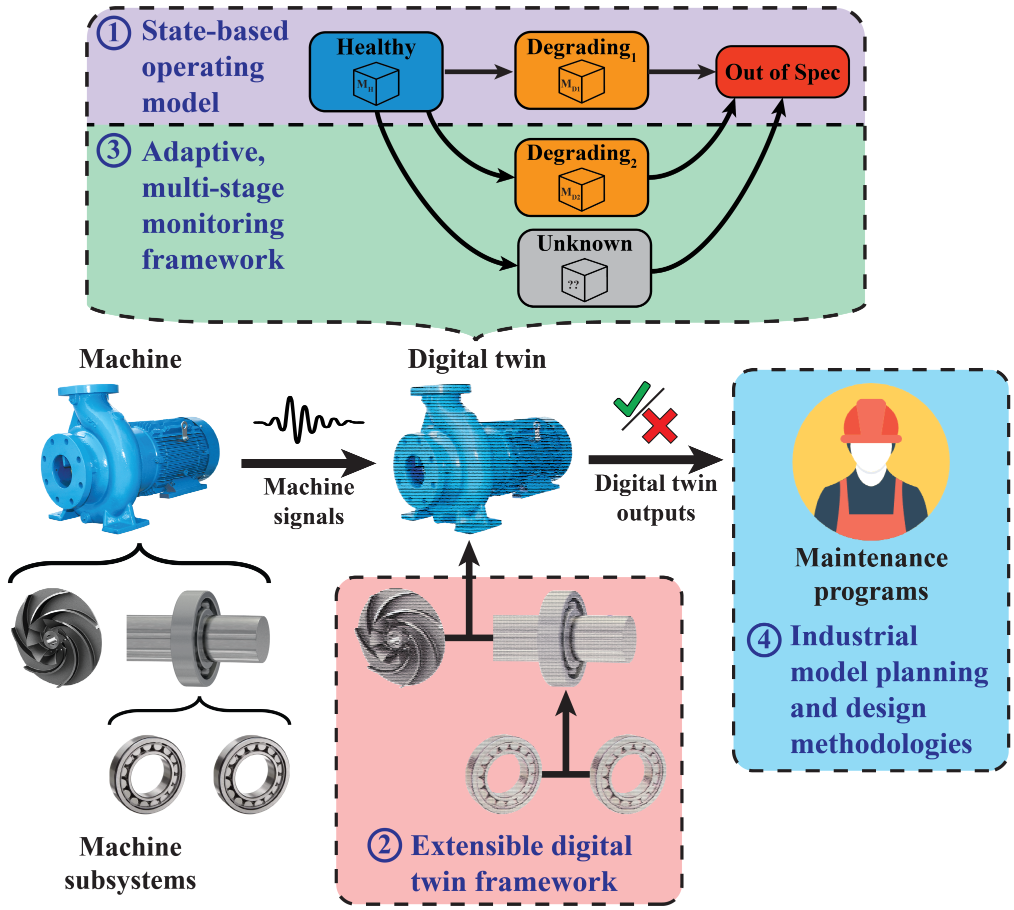

Figure 1.1: Depictions of the four contributions of this dissertation

cations and a set of offline states that contextualize gaps in machine operation. Based on this model, two sub-contributions are made and are listed below.

(a) A method for trend-based repair quality assessment in industrial equipment

(b) A method for multi-stage fault prediction in rolling element bearings

This material is presented in Chapter 3 and published in [33, 34].

2. *An extensible digital twin framework for PHM*

For predictive maintenance strategies to become widespread in the manufacturing industry, modeling experts must be able to reuse PHM solutions across similar machines and components, and combine modeling resources to fully describe the health of complex equipment. Recent work has recognized the need to standardize the internal processes and architecture of individual PHM solutions [19, 35, 36, 37, 38], but standardized methods for combining and transmitting information between PHM solutions have not been specified in full detail. The second contribution of this dissertation addresses this limitation by introducing a purpose-driven digital twin framework for PHM. The framework defines a set of digital twin classes that fulfill different PHM capabilities and specifications for aggregation relationships that can be used to combine subsystem and component digital twins to represent complex machines, as shown in Fig. 1.1. This material is presented in Chapter 4 and published in [39, 40].

3. *An adaptive and extensible framework for multi-stage degradation monitoring and anomaly detection*

The two contributions discussed above provide a foundation for state-based PHM of industrial equipment, but the dynamic nature of manufacturing environments still presents a variety of challenges for PHM solutions. The length and rate of machine degradation can vary significantly between fault instances, and model training data from periods of unhealthy operation are often limited, which makes predicting faults difficult even with a state-based modeling approach. Several methods for adapting low-order trend-based degradation models based on recent sensor measurements have been proposed in existing literature [41, 42, 43, 30, 31]. However, these works only monitor a single, known degradation mode and assume that degradation follows a fixed stage sequence. The third contribution of this dissertation is a modeling framework for detecting degradation and predicting faults across multiple modes while also monitoring for unforeseen degradation modes, as shown in Fig. 1.1. This work includes a methodology that uses real-time sensor measurements to adapt signal trajectory parameters, estimate a machine's degradation stage history, and predict faults. This material is presented in Chapter 5 and published in [44].

4. *Methodology for planning and designing industrial PHM solutions*

Accurate and robust health models are critical for the success of predictive maintenance strategies in the manufacturing industry, but modeling is just one aspect of PHM. A number of other planning and design decisions limit the effectiveness and lifetime of industrial PHM solutions more often than equipment modeling capabilities. Challenges related to defining the scope of a solution, specifying clear and actionable output quantities, and pre-processing historical data for model training are extremely common in manufacturing applications. Methods for overcoming them have received relatively little attention compared to new health models. The fourth contribution of this dissertation is a methodology for the planning and design of industrial PHM solutions that outlines decisions and challenges that arise during these development stages. To fully illustrate how this methodology can be put into practice, a detailed description of the development of a PHM solution for an industrial hyper compressor is included. This material is presented in Chapter 6 and published in [45].

1.3 Research Impact

Manufacturers can utilize the work in this dissertation to deploy PHM solutions that enable factory-wide predictive maintenance strategies. The state-based modeling methods presented here make it possible to derive standardized estimates of equipment health and are better suited to dynamic manufacturing environments than existing snapshot-based methods. These capabilities facilitate modeling the health of more machines in a manufacturing plant and lengthen the lifetime of machine health models. Manufacturers can then achieve the predictive maintenance benefits of reduced downtime, maintenance costs, and lost production at a larger scale than previously possible. The general modeling frameworks specified here also make it possible to incorporate the latest methods from current and future PHM research into predictive maintenance strategies to continually expand and improve them.

1.4 Dissertation Overview

This dissertation presents novel, state-based modeling approaches and extensible modeling frameworks to support the deployment of PHM solutions in manufacturing environments. The contributions documented here have been developed to act as a bridge between the advanced modeling and analysis research being conducted in academia and the practical realities of deploying and maintaining PHM solutions in industrial manufacturing environments. Chapter 2 provides background information and a discussion of recent literature. Chapter 3 details a general operating model for state-based PHM and two novel health monitoring strategies that utilize this model. Chapter 4 builds upon the concept of state-based PHM by specifying a digital twin framework for fault monitoring that facilitates the reuse and aggregation of modeling resources. Chapter 5 proposes an adaptive fault prediction methodology that uses input from machine experts to monitor multi-stage degradation processes in place of extensive historical datasets. Chapter 6 analyzes several challenges unique to developing PHM solutions for industrial applications and discusses how methods from this dissertation and related literature can be used to overcome them. Finally, Chapter 7 contains concluding remarks and recommendations for future work in this research area.

CHAPTER 2

Background

This chapter first introduces several approaches for modeling the health of manufacturing equipment that are widely used in prognostics and health management (PHM) literature. This culminates in a discussion about the advantages of state-based health modeling and an overview of notable state-based model structures. Recently-proposed frameworks to develop and deploy PHM solutions that utilize these models to monitor system health in real-time are then surveyed. The chapter concludes by reviewing several types of adaptive PHM solutions, which seek to respond to real-time sensor measurements and changes in machine operating parameters either by tuning or switching between system health models.

2.1 Health Modeling of Manufacturing Equipment

The PHM research area has benefited from the interest and efforts of a number of sectors, including aerospace, automotive, energy, and manufacturing industries [15, 46, 47, 48]. Corporations and other organizations in all of these industries stand to benefit from PHM solutions that provide feedback on the health of their assets.

In manufacturing, much of the PHM research is focused on modeling machine health. The term "mechanical system" (or simply "system") will be used here to refer to machines, machine components, and machine subsystems, which can all be modeled as self-contained entities. Historically, health models have been used for both offline and online analysis of mechanical systems. Research on offline system health modeling includes a variety of methods to analyze failure time datasets and describe system reliability using statistical models such as the Weibull distribution [49, 50]. These models are used to estimate metrics such as a system's failure rate and mean time to failure (MTTF) and to design the time or production-based schedules that preventative maintenance strategies rely upon [9].

More recently, PHM researchers in industry and academia have begun to develop online system health models to analyze machine sensor measurements in real-time and make estimates about current and/or future system health states. Online health models will be the focus of the work discussed in this dissertation because they are critical for implementing predictive maintenance strategies. These models have a range of structures and complexities that reflect the diversity of health problems that can arise in industrial equipment. Physics-based models are often used in applications where a single degradation process is of interest and the physics of the process are well-understood [51]. For instance, the Paris-Erdogan model for crack growth [52] has been applied to predict failure times for gears [53]. The physics of bearing degradation have also been explored to detect the presence and severity of bearing surface defects [54, 55].

The spread of smart sensing technology, including industrial acoustic and vibration sensors, has prompted an increase in the popularity of data-driven health models, many of which are based on concepts from machine learning and artificial intelligence. Common model structures include neural networks, support vector machines, and decision trees [19, 56]. The popularity of these methods is due, in part, to their versatility in modeling different types of machines and failure types [14, 57]. Data-driven models have been proposed to monitor health problems in motors [58, 59], pumps [60, 61], and compressors [62, 63], among many other types of machines. These models often require large quantities of data for training, but are useful alternatives to physics-based models when historical datasets are available.

The majority of health modeling approaches currently implemented in industry generate results from snapshots of machine signals collected at the current point in time. These models are designed to learn underlying relationships between the magnitudes of machine signals and machine health. However, common external disturbances such as maintenance procedures and changes in operating parameters may change these relationships. PHM solutions that use snapshot-based models then must be tuned or fully retrained frequently to preserve their accuracy.

In response to this limitation, interest in health models that account for time-series behavior in machine signal measurements has grown. General path modeling is a notable example that explicitly considers time-series behavior in machine signals. Under this approach, subject matter experts (SMEs) specify a machine signal feature that is known to indicate degradation and an underlying degradation trend. Real-time data are then used to detect and parameterize this trend, which can be extrapolated to predict system faults [43, 64]. Existing PHM literature commonly uses general path models (GPMs) with linear [42] and exponential [41] trajectories. Some PHM solutions use GPMs with stochastic parameters that can be re-estimated according to a Bayesian update strategy when new signal measurements are available. The resulting fault predictions are then probabilistic windows that correspond to a user-specified level of certainty. These GPMs are useful for tracking degradation processes that are characterized by a single continuous trend in one machine signal, but they are not able to monitor multi-stage or multivariate degradation trends.

Compared to snapshot-based models, trend-based models are more robust to shifts in the baseline values of machine signals. These models are also better-suited to detect early-stage degradation in industrial equipment that is often characterized by gradual trends in machine signals. Recognizing early signs of health problems is often crucial for triggering low-cost maintenance procedures and avoiding downtime.

2.2 State-Based PHM

The trend-based methods discussed above demonstrate the value of considering historical system measurements when modeling the health of industrial equipment. State-based PHM extends this concept further by synthesizing historical measurements into estimates about the health state of a system that then inform subsequent analysis. These state estimates can be discrete and act as the output of a PHM solution. “Healthy”, “degrading, and “faulty” classifications are common examples. State estimates may also be continuous, such as when a value of a health index informs system fault predictions [19]. When system health classifications are desired, state-based PHM solutions can incorporate prior state estimates into the current analysis, which is useful because system health states tend to be closely correlated in time. State-based PHM is well-suited to predict faults during multi-stage degradation processes as well [32]. In these types of applications, a state-based PHM solution can maintain an estimate of the system’s current degradation stage and adapt fault predictions accordingly.

Recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) are notable types of data-driven, state-based modeling tools that have been utilized in PHM research. These models are able to learn how to represent continuous system health states from large historical training datasets, and then maintain and update these states based on real-time data [65]. Existing work has demonstrated promising results when using these types of models to predict faults in manufacturing equipment [66, 67].

Hidden Markov models (HMMs) are another useful modeling tool when PHM solutions developers wish to capture multiple discrete system health states. With this approach, the Viterbi algorithm can be used to compute the most likely state history of a system based on features extracted from machine signals [68]. The Baum-Welch algorithm is commonly used to estimate the parameters of HMMs based on historical data [69]. Recent PHM research has used HMMs to identify the presence and extent of system health problems in industrial equipment [29, 70]. Hidden semi-Markov models can be used when fault prediction capabilities are desired [71]. These models allow state transition probabilities to change over time to reflect system deterioration. The time remaining until a system enters a failure state, commonly referred to as remaining useful life (RUL), can then be estimated using methods described in [72] and [73].

State-based PHM can be implemented using physics-based models of system dynamics or degradation processes as well. These models are commonly structured as state-space models or differential equations. A survey of model-based PHM approaches for detecting and diagnosing degradation can be found in [32]. Observer-based methods have been proposed to estimate health-related model parameters based on system outputs, as in [26, 27]. Related work has detected degradation in industrial equipment by compute residuals between expected and actual system outputs [28, 74].

The literature cited above clearly demonstrates the value of state-based PHM for manufacturing applications, but the vast majority of existing research in this area has been developed specifically for individual systems or health problems. As a result, the meanings of the system health states that these solutions utilize and the implications of their outputs vary widely. This may not be a problem for operations that wish to apply methods from PHM research in isolated systems, in which case it may be reasonable to evaluate and implement PHM solutions on a case-by-case basis. However, this approach presents significant challenges for manufacturers interested in deploying state-based PHM solutions at a factory-wide scale to support predictive maintenance strategies. For this to be feasible, the PHM solutions that an operation implements must be based on a standardized scheme for representing system health states. Such a scheme, though, has not been put forth in existing literature. Chapter 3 of this dissertation addresses this research gap by proposing a general operating model to support the development of state-based PHM solutions for manufacturing equipment.

2.3 PHM Modeling Frameworks

While a standardized representation of system health is useful for planning the scope of an industrial PHM solution, there are a number of other decisions that must be made when developing a program that is able to analyze a mechanical system's health in real-time. Various computations related to pre-processing machine signals, health modeling, and synthesizing modeling results must all take place within a PHM solution whenever new data is collected. For manufacturers to develop, deploy, and maintain many solutions across their various machines, it is necessary for them to have a consistent structure. Thus, a great deal of recent PHM research has been devoted to creating formal methodologies for developing and structuring PHM solutions.

Many of these works specify, at a high level, the general capabilities and processes that PHM solution must implement to determine health-related outputs from real-time data [16, 19, 35, 36]. A well-known example of a PHM capability framework is described in International Organization for Standardization (ISO) Standard 13374, which specifies six general capabilities that are integral to PHM: data acquisition, data manipulation, state detection, health assessment, prognostic assessment, and advisory generation [37]. The Open System Architecture for Condition Based Maintenance, developed by the MIMOSA organization, also proposes a general architecture for individual PHM solutions, with standardized communication interfaces for sending information between functional blocks [38].

A related category of research has explored the role of internet of things (IoT) and cloud technologies in PHM strategies. Several computational frameworks provide security-minded solutions for collecting data from multiple, distributed sensing nodes, storing information in cloud-based databases, and interfacing with third-party services and applications [75, 76]. In [77], Gao et al. discusses the need for predictive maintenance strategies to be compatible with cloud computing and reviews several computing platforms that can help facilitate this position. Cloud computing for PHM is put into practice in [78], which describes a novel distributed computing platform with multiple agents that communicate across a network to derive system health state estimates.

These general PHM capabilities frameworks and low-level computational frameworks have facilitated the initial deployment of isolated PHM solutions in industry, but manufacturers have begun to face additional challenges related to organizing and maintaining modeling resources. Among these are the ability to update equipment models as new information becomes available, to combine insights from multiple components and health models to make conclusions about system health, and to reuse models across manufacturing equipment fleets. Future PHM research will be focused on realizing these capabilities, and the newly-emerging field of digital twins has the potential to provide many enabling technologies.

The concept of a digital twin (DT) was first proposed in 2002 by Michael Grieves in a presentation on product life cycle management [79] describing data flow between a physical system and a virtual representation of that system. Interest in DTs has exploded since then, though subsequent research efforts are not in perfect agreement on the definition of a digital twin [25, 80, 81, 82]. NASA is often credited with developing the first definition of a DT in 2012: "an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin" [83]. Recent reviews have derived more general definitions based on the shared attributes of DTs developed for manufacturing, healthcare, and city management applications, among others. The authors of [81] and [84], for instance, conclude that a DT is a "live digital coupling of the state of a physical asset or process to a virtual representation with a functional output." ISO Standard 23247 provides a more focused definition of DTs for manufacturing: "a fit for purpose digital representation of an observable manufacturing element with synchronization between the element and its digital representation" [85].

One point of concurrence in recent DT literature is that the synchronization and flow of information between a physical system and its DT represents a distinction between DTs and traditional models or simulations [25, 80, 86, 87]. While DTs rely upon models and may run simulations, this synchronization allows them to run alongside a physical system, with the ability to adapt to changes in their counterpart and provide modeling results in real-time. This characteristic makes DTs well-suited for PHM applications and PHM researchers have already started utilizing them [82, 88]. The authors of [89] use a machine DT to combine multiple models with different structures to make more informed conclusions about system health and [90] provide an example of how DTs can update the parameters of machine models to adapt to changes in its physical counterpart.

Similar to the existing work on state-based PHM, PHM-related DTs in the literature have primarily been proposed for individual systems and use cases. For widespread deployment of PHM solutions and predictive maintenance strategies to be feasible for manufacturers, general architectures for PHM DTs must be developed. This foundation will allow researchers to further explore how the model management and adaptation capabilities of digital twin can be used to improve upon the static PHM solutions specified by the frameworks discussed previously [16, 19, 35, 36]. Additionally, recent reviews and standards have identified the need to combine PHM modeling resources to represent complex systems with interrelated subsystems and components [91, 92, 93, 94]. DT aggregation relationships, as proposed in [25], represent a promising mechanism to realize this capability, but existing work has not explored how this can be implemented with PHM-related DTs. Chapter 4 presents a DT framework for PHM that has been developed to overcome these limitations.

2.4 Adaptive Modeling for PHM

Despite the wealth of health modeling research discussed above, for many industrial manufacturing applications there is still a great deal of uncertainty surrounding the impact of external disturbances on system degradation behavior. In addition to shifting machine signal baselines, maintenance procedures, operating parameter changes, and other factors can influence the rate at which degradation progresses and the relative length of stages in multi-stage degradation processes. Developing adaptive PHM solutions and intuitive frameworks for deploying these solutions and communicating their results to system operators will then be critical for the success of predictive maintenance in the manufacturing industry

Though the bulk of existing literature presents static PHM solutions, with health models that are trained and deployed once, some advances in adaptive PHM technology have been made of late. In general, these works adapt the parameters of health state estimation and fault prediction models based on recent machine sensor measurements. The previously-introduced general path modeling approaches use Bayesian updates to adapt the parameters of linear and exponential signal trajectories based on real-time data [41, 42, 43]. Subsequent work has proposed methods to explicitly model the effects of operating parameters such as load and speed on the degradation behavior of machine signals [95, 96, 97]. Changes in these parameters during operation can then be incorporated into fault predictions. The ability for DTs to detect changes in a manufacturing environment or a machine's operating parameters and adapt system health models accordingly has also been explored in recent work [98, 99].

Another area of research focuses on adaptive modeling of multi-stage degradation processes, also referred to as multi-phase degradation. The authors of [56] describe how two-stage degradation processes can be modeled as a Weiner process with an abrupt jump, and use real-time sensor measurements to estimate the parameters of this process and detect the stage transition point. This concept has been extended in [31] and [100] to describe degradation processes with multiple stage transitions. Piecewise linear functions are used in [30] in place of Wiener processes to achieve a similar capability.

In individual applications, researchers have had success detecting and diagnosing health problems by modeling machines as hybrid systems. With this approach, Kalman and particle filters can be used to detect the onset of degradation and diagnose a system's current degradation mode based on how closely machine signals adhere to predefined linear and nonlinear state-space models [74, 101]. Researchers have applied this concept to detect faults in aircrafts [102, 103], underwater vehicles [104], multi-tank flow systems [28], and batteries [105]. The use of state-space models allows these PHM solutions to adapt to varying machine signal baselines, extending their lifetimes and the scope of systems that they can monitor, as discussed previously. When health problems are detected, transitions between discrete degradation stages can also be detected and predicted based on real-time sensor measurements rather than prior assumptions about stage duration.

Existing adaptive PHM solutions have proven successful for isolated applications, but they are subject to a number of assumptions that are often not valid in the manufacturing industry. In manufacturing applications, system experts and operators may not have a perfect understanding of the health problems that a system is susceptible to, or the order in which degradation stages will emerge prior to system fault. Existing work has not presented a general adaptive modeling framework that is compatible with these practical realities. Proposed methods from general path modeling literature assume that a system is subject to a single degradation mode that consists of a single continuous degradation process. Methods that analyze multiple degradation modes and support modeling multi-stage degradation processes still assume that a system will follow a fixed stage sequence and do not monitor for unknown or anomalous degradation modes. Chapter 5 presents a modeling framework and an adaptive fault prediction methodology that addresses these limitations.

2.5 Developing Industrial PHM Solutions

Selecting an appropriate health modeling approach is crucial when developing an industrial PHM solution, but a number of other considerations must be made during this process. PHM solution developers are also tasked with soliciting input from multiple types of project stakeholders and subject matter experts, selecting actionable output quantities, and defining pathways to integrate a solution into existing maintenance routines. These challenges and several others arise when researchers seek to transfer techniques from academic labs into the real world, but have received little attention in literature. Chapter 6 presents a methodology for developing industrial PHM solutions that is derived from the system development life cycle [106, 107]. Methods for overcoming common challenges that arise during the planning and design stages are discussed in detail.

CHAPTER 3

A General Operating Model for State-Based PHM

This chapter presents work published in [33, 40]. As discussed in Section 2.2, state-based prognostics and health management (PHM) solutions have primarily been developed for individual use cases, which limits the scale at which PHM concepts and predictive maintenance strategies can be deployed. In addition, most existing solutions have been designed for and tested on systems operating in controlled environments with minimal disruptions. As a result, it can be difficult to train and test the health models that these solutions rely upon with data from manufacturing machines that experience regular gaps in operation due to maintenance procedures and other external events.

The primary contribution of this chapter is: a discrete-state operating model that facilitates state-based PHM solutions for manufacturing equipment. The states in this model can be used to describe the health of mechanical systems during online operation. Events that describe transitions between online and offline operation are also included to describe gaps in historical and real-time data. To demonstrate how this model can be used to develop state-based PHM solutions across different types of systems, two novel sub-contributions are also included: (a) a method for trend-based repair quality assessment in industrial equipment and (b) a method for multi-stage fault prediction in rolling element bearings.

The rest of the chapter is structured as follows. Section 3.1 provides general classifications for anomalous behavior in manufacturing equipment and specifies the scope of the proposed model. Section 3.2 presents the general operating model and provides definitions for its states and transitions. Section 3.3 describes a novel method for repair quality assessment that uses the operating model to represent and contextualize transitions between online and offline periods. Section 3.4 describes a novel method for fault prediction in rolling element bearings that uses the model to represent multi-stage degradation. Conclusions of this chapter are given in Section 3.5.

3.1 Problem Framing

Fig. 3.1 shows various causes for anomalies in industrial manufacturing systems, defined as any occurrence that is different from what is standard, normal, or expected [108]. Anomaly causes include forces external to the machine being analyzed (such as network delay or material input that is out of specification). While the contributions of this dissertation can be applied to monitor process and product anomalies like these, this discussion is focused on a sub-class of anomalies known as equipment faults, defined as anomalies that are related to an unwanted situation originating within an industrial manufacturing machine [108]. These behaviors may be associated with the violation of a cost or safety standard and require maintenance to resolve if current conditions persist.

The immediate need for maintenance distinguishes equipment faults from other types of anomalies that simply warrant increased supervision by system operators. For example, intermittent sensor spikes into warning regions might fall into this latter category of anomalies. The occurrence and frequency of these spikes can be provided as inputs to digital twins (DTs) that enable predictive health monitoring, but they can generally be viewed as precursors to a machine fault that may occur in the near future. For example, a temporary drop in the output pressure of a pump would not

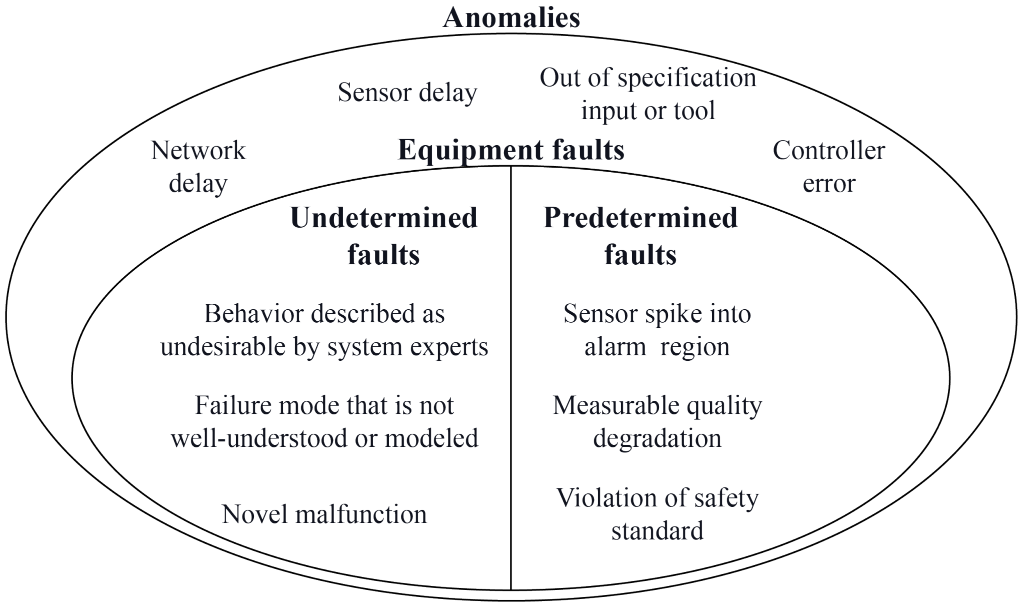

Figure 3.1: Sub-spaces of anomalous machine behavior and examples of anomaly types

require maintenance, but a predictive model may track the frequency of these drops to determine when these anomalies will cause the pump's average hourly flow rate to drop below a critical threshold.

Within equipment faults, a further distinction is made between predetermined faults and undetermined faults. Predetermined faults have a quantifiable impact on the operation of a machine that system experts are aware of and can be monitored while a machine is operating. These faults are often formally assessed in equipment reliability strategy (ERS) reports and equipment-specific sections of failure mode and effects analysis (FMEA) reports. Undetermined faults do not have these characteristics. This includes behavior that is deemed undesirable by system experts but does not have a known, quantifiable impact on machine operation, or novel malfunctions that were not anticipated or monitored prior to their occurrence. The PHM ontology in [109] presents an detailed analysis of system faults from different perspectives, including their underlying causes and relationship to system failures, which is beyond the scope of this work.

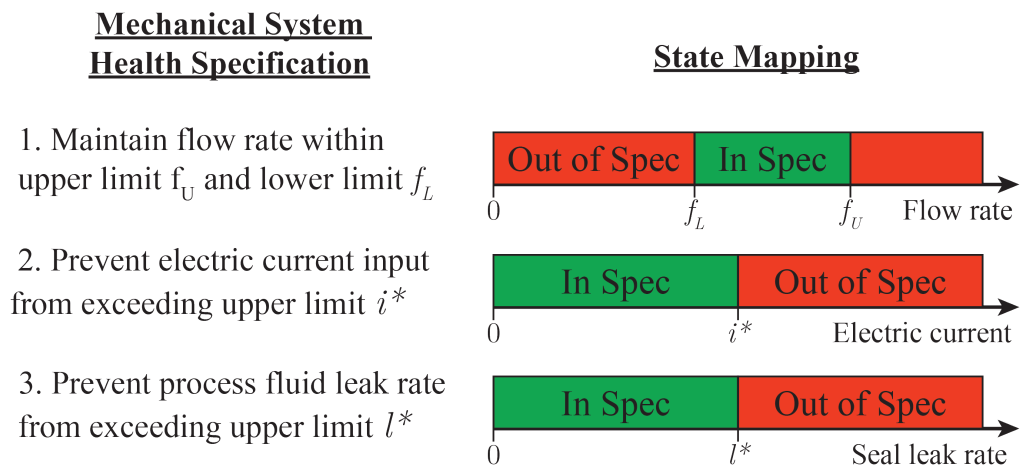

Figure 3.2: Examples of mechanical system health specifications

Because predetermined faults have a known, quantifiable impact on machine operation, they can be described using a mechanical system health specification, defined as a mapping between the information available from a mechanical system and the states in the set $\{In\ Spec,\ Out\ of\ Spec\}$. In general, a system that is *In Spec* can be considered operational, while a system that is *Out of Spec* has encountered a problem that requires an intervention if the current conditions persist. These states are often implicitly defined in the form of alarms set by system experts to trigger maintenance actions. Mechanical system health specifications may refer to a wide range of system attributes, including performance metrics (such as throughput or product quality), operating costs (such as energy or material consumption), and safety measures (such as temperature or noise limits). [109] provides details on a process known as failure mode symptoms analysis (FMSA) that can be used to identify the effects of system faults and define corresponding health specifications. A collection of example specifications for an industrial pump can be seen in Fig. 3.2.

Based on this understanding of machine behavior, the contribution presented in this chapter can be understood as a framework for system health monitoring with respect to predetermined fault types, defined as the process of estimating a mechanical system's current and future compliance with a set of mechanical system health speci-

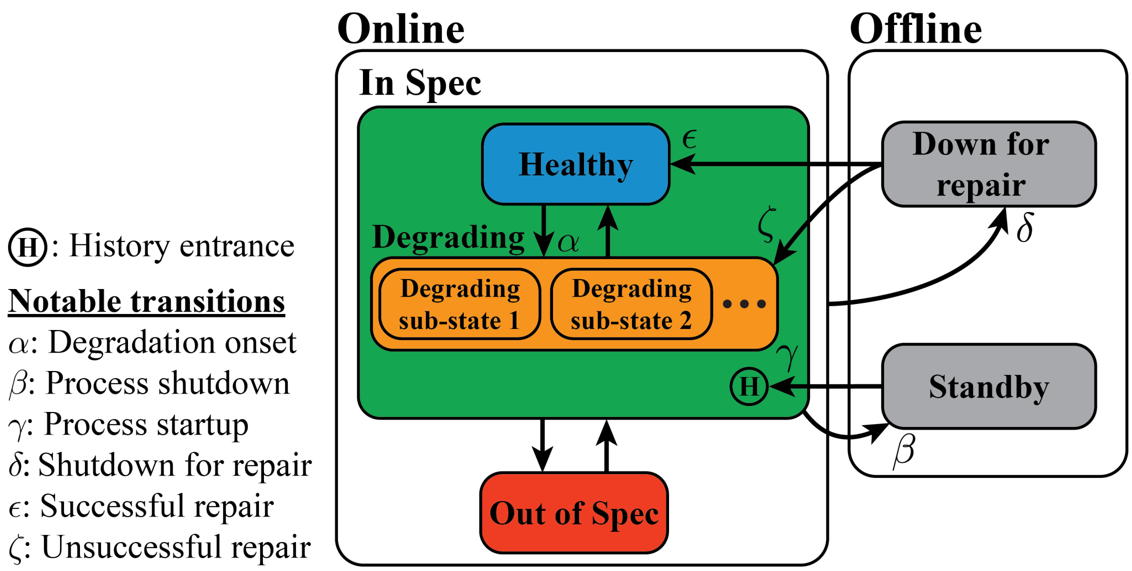

Figure 3.3: General operating model for state-based PHM showing hierarchy of online and offline system states and transitions between them

fications. It is worth noting that health monitoring is a critical aspect of PHM, but is just one piece of a complete PHM strategy. Further PHM capabilities, such as undetermined fault detection and maintenance quality assessment, should be implemented using additional modeling resources.

3.2 General Operating Model

A general model operating that can be used to develop state-based PHM solutions is presented in this section. The statechart in Fig. 3.3 depicts the set of states (S) and set of events (E) included in this model using the visual formalism introduced in [110].

The model consists of two super-states: *Online* and *Offline*. A machine can be considered *Online* when it is operating as part of a continuous manufacturing line. While online, a system may be *In Spec* or *Out of Spec*, with respect to a predefined system health specification, as described in Section 3.1 The *In Spec* state contains both *Healthy* and *Degrading* states to represent the general progression of mechanical systems.

This operating model utilizes a common assumption in PHM literature: that equipment health can be represented as a two-stage process. The first stage (*Healthy*) is characterized by steady-state behavior. During the second stage (*Degrading*), a machine begins to show signs of health problems that may worsen in the future. This representation of equipment health has been used to predict impending faults in compressors [111] and rolling element bearings [112], and mirrors the vibration evaluation zones defined in engineering standard ISO 20816 [113]. Here, this concept is extended to allow the *Degrading* state to have multiple sub-states, if useful for a particular application.

During healthy operation, a machine is performing as expected within a manufacturing line, with no signs of wear or impending failure. A degradation onset event (α) represents the time at which signs of degradation become apparent. After a degradation onset event, a machine transitions to a *Degrading* state. A process shutdown (β) or a shutdown for repair event (δ) can trigger a transition to an *Offline* super-state, which describes periods when a machine is bypassed in a manufacturing line. Bringing a machine offline to conduct a repair procedure is considered a shutdown for repair event and results in a transition to a *Down for repair* state. Alternatively, bringing a machine offline for reasons unrelated to repair is considered a process shutdown event, which results in a transition to a *Standby* state.

The key distinction between process shutdown and shutdown for repair events is that the latter can only occur while a machine is within a *Degrading* state, while a process shutdown event can occur whenever a machine is online. The *Down for repair* state captures all activities associated with machine repair, up to the point when a machine is brought back online. A transition to an *Online* super-state can be triggered by either successful repair (ϵ) or unsuccessful repair events (ζ), at which point a machine resumes operation in a *Healthy* or *Degrading* state, respectively.

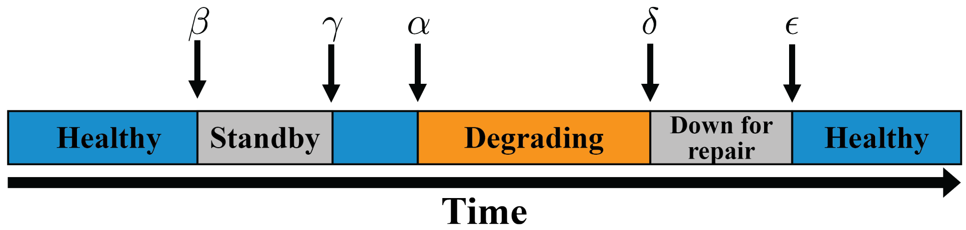

Figure 3.4: Example timeline of system states and events before and after a repair procedures

A process startup event (γ) triggers a transition from a *Standby* state to an *Online* super-state through a history entrance. The history entrance, denoted by (H) in the statechart, means that the system resumes operation in the *Online* state that was visited most recently. The *Standby* state captures periods in which a machine is offline for reasons unrelated to machine health, and the history entrance is appropriate when machines spend relatively short periods of time in a *Standby* state. In applications where machines experience long periods of inactivity, it is possible for new machine health problems to emerge while offline. It would then be necessary to employ a state estimation method, like the one described in Section 3.3, to monitor machine health after transitioning out of a *Standby* state. Future work may expand this model to describe long offline periods by creating *Standby* sub-states with separate transitions to *Healthy* and *Degrading* states.

It is helpful to consider this operating model in the context of an example equipment life timeline, shown in Fig. 3.4. Healthy operation of the machine portrayed here is interrupted by a transition to a *Standby* state before a degradation onset event triggers a transition to a *Degrading* state. Signs of equipment degradation prompt a shutdown for repair event, which is followed by a successful repair event that brings the machine back to a *Healthy* state.

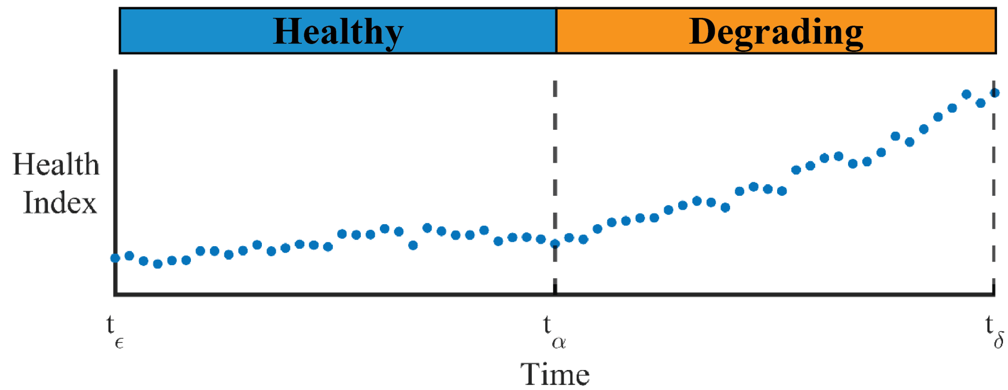

Figure 3.5: Illustration of expected stationary index behavior during healthy periods and linear index trend during degrading periods

3.3 Trend-Based Repair Quality Assessment

This section presents a method for building and training trend-based health models that can be used to assess the quality of system repairs. First, an offline model building process is described, followed by an online repair quality assessment process process.

Model building consists of three sub-processes: index selection, state labeling, and state characterization. The index selection sub-process assumes that manufacturers have the ability to repeatedly measure features from machine sensors. System knowledge is needed to filter these features, which can often be numerous and diverse, into a single, scalar health index that describes equipment health. The state-labeling sub-process looks backwards in time from a shutdown for repair event to identify the point of degradation onset. This sub-process relies on the assumption that health index values are roughly stationary during periods of healthy operation and follow a monotonic trajectory during periods of degrading operation. Fig. 3.5 shows an example of stationary index behavior followed by a linear degradation trajectory. The state characterization sub-process uses labeled historical datasets to describe time-series behavior of the health index in *Healthy* and *Degrading* states.

The remainder of the section describes an online process for repair quality assessment based on index values collected after repairs. The assessment process can be repeated, whenever new measurements are available, to classify repairs as successful or unsuccessful.

3.3.1 Model Building: Index Selection

The first sub-process of model building involves selecting a scalar health index (i) from a machine's features ($\boldsymbol{f}$). Individual features may be the direct output of a sensor, such as instantaneous temperature or pressure readings, or an aggregate quantity that describes a machine signal snapshot. Index selection is represented mathematically by the function $g(\boldsymbol{f}) \rightarrow i$. The choice of a health index is highly dependent on the equipment being monitored as well as a plant's sensing capabilities, so knowledge from subject matter experts should be used to define an appropriate selector function. The function may output a univariate health index, using a standard basis vector of the feature space, or may involve a multivariate transformation, such as a projection onto a principle component of a dataset. Examples of both are shown in Eq. 3.1, where w^T represents a principle component weight vector.

$$\begin{array}{cc} \underline{\text{Univariate Selector}} & \underline{\text{Multivariate Selector}} \\ i = \hat{e}_1 \times \boldsymbol{f} & i = w^T \times \boldsymbol{f} \\ \text{where } \hat{e}_1 = \begin{bmatrix} 1 & 0 & 0 & \dots \end{bmatrix} & \end{array} \tag{3.1}$$

The repair assessment method relies on the assumption that the index follows a linear trajectory with non-zero slope during periods of equipment degradation. So, the behavior of potential health indices should be analyzed during known periods of degradation to ensure this assumption is satisfied. In cases where degradation is non-linear, it may be possible to apply a transformation that linearizes the index trajectory, as in [41].

3.3.2 Model Building: State Labeling

Historical data is defined here as feature values measured between start time (t_1) and end time (t_N). If feature values are measured at constant rate, Δt, historical data can be specified by the quantities $(F, t_1, t_N, \Delta t)$, where feature measurements are stored in the matrix $F = [\boldsymbol{f}_{t_1} \quad \boldsymbol{f}_{t_2} \quad \dots \quad \boldsymbol{f}_{t_N}]$. A feature matrix F can be mapped to a time-series history of health index values by applying the selector function $g(\boldsymbol{f})$ to each column of the matrix.

This method assumes that all the events presented in the general operating model, excluding degradation onset, can be retroactively identified and used to segment historical index values. The state labeling sub-process involves identifying the occurrence of a degradation onset event within a set of health index values measured after a successful repair event and before a shutdown for repair event. A dataset fitting those requirements can be expressed as the vector $I = [i_{t_\epsilon} \quad \dots \quad i_{t_\delta}]$.

It is possible for a vector I to be interrupted by transitions into and out of a *Standby* state. This historical period can still be used for model building if system experts are confident that the degradation process was not affected by the interruptions. In this case, index values measured while a machine occupies a *Standby* state, and their corresponding timestamps, should be omitted from the datasets analyzed here. More work is necessary to assess the impact of extended *Standby* periods on the degradation of rotating equipment.

If a machine transitions to a *Healthy* state at t_ϵ and transitions from a *Degrading* state at t_δ, the general operating model requires that a degradation onset event occur at some intermediate time t_α, where $t_\epsilon < t_\alpha < t_\delta$. The state labeling sub-process relies on the assumption that health index values follow a linear degradation trajectory. A dataset's fit to a linear trajectory can be measured using the coefficient of determination (R^2) statistic. Degradation onset is identified by determining the set of consecutive index values prior to t_δ with a maximum R^2 statistic. Very small datasets

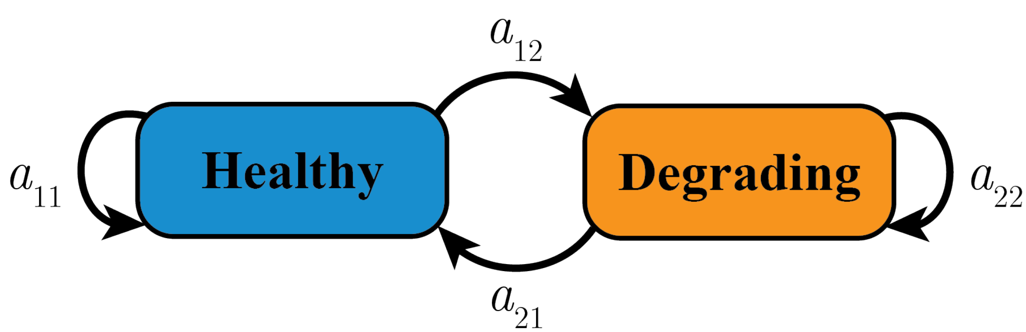

Figure 3.6: Depiction of the hidden Markov health model used to assess the quality of repair procedures

are likely to skew the R^2 statistic, so it is necessary to restrict the search space for t_α to $T_\alpha = [t_\epsilon \quad \ldots \quad t_\alpha^{max}]$ where t_α^{max} is slightly offset from t_δ. This process is framed as an optimization problem in Eq. 3.2 and can usually be solved with a brute force search because there are a finite number of values that t_α can take.

$$t_\alpha = \underset{t_n \in T_\alpha}{\arg\max}\, R^2(i_{t_n} \quad \ldots \quad i_{t_\delta}) \tag{3.2}$$

Once t_α is identified, the index vector I is divided into subsets $I_H = [i_{t_\epsilon} \quad \ldots \quad i_{t_{\alpha-1}}]$ and $I_D = [i_{t_\alpha} \quad \ldots \quad i_{t_\delta}]$ which are labeled as “Healthy” and “Degrading”, respectively.

3.3.3 Model Building: State Characterization

Labeled I_H and I_D datasets are analyzed to characterize machine behavior in *Healthy* and *Degrading* states. The proposed health model has a hidden Markov model structure with a Markov chain that consists of *Healthy* and *Degrading* states, as shown in Fig. 3.6. The states in this Markov chain are unobservable, but datasets I_H and I_D are used to characterize state behavior using an index slope parameter.

When applied over a window of sequential index values $[i_{t_1} \quad \ldots \quad i_{t_N}]$, and corresponding timestamps $[t_1 \quad \ldots \quad t_N]$, linear least-squares regression analysis outputs a pair of coefficients (b, c) that minimize the sum of squared residuals as defined in Eq. 3.3.

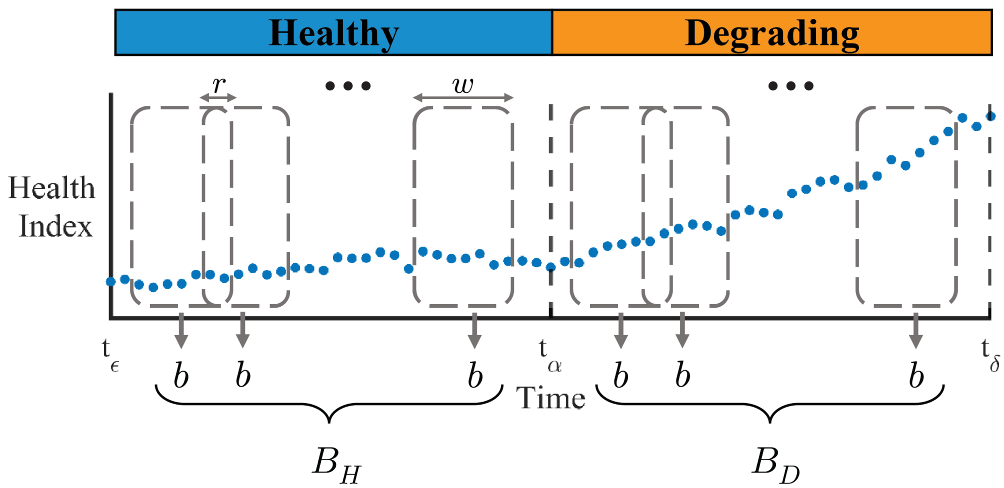

Figure 3.7: Illustration of index windows and index slope datasets during healthy and degrading periods

$$r = \sum_{i=1}^{N} i_{t_i} - (b \times t_i + c) \tag{3.3}$$

The coefficient b is a measure of how index values vary with time over the windowed dataset, and the value of this coefficient is the observable state in the hidden Markov health model. Historical datasets I_H and I_D are first windowed (with window size w and window overlap r), then a linear least-squares regression analysis is applied over each window to calculate two sets of coefficients (B_H and B_D) that describe the time-series behavior of health index values. An example diagram showing windowed index values and the b coefficients that make up B_H and B_D is presented in Fig. 3.7.

The repair assessment process assumes that b coefficients in *Healthy* and *Degrading* states have a Gaussian distribution. If historical b coefficients deviate significantly from Gaussian behavior, tools such as the Box-Cox transformation can make these observations resemble Gaussian data, as in [114]. Observation probability distributions in the hidden Markov health model are then defined by the mean and standard deviation of datasets B_H and B_D.

The hidden Markov health model is designed to model machine behavior during uninterrupted operation, so the transition probabilities $a_{21} = 0$ and $a_{22} = 1$ are set to impose the requirement that once a machine has begun degrading, it will continue to do so until brought offline for repair. Historical data is also used to estimate the value of a_{12}, the probability that a healthy machine will transition to a *Degrading* state between two consecutive observation windows. An I_H dataset can be considered a set of Bernoulli trials that are repeated until the first occurrence of success (transition to a *Degrading* state).

The maximum likelihood estimator for the probability of success from a set of Bernoulli trials is $\hat{p} = {}^{\sum x}/_{n}$ where $\sum x$ is the total number of successes out of n trials [115]. Given window parameters w and r, an I_H dataset of length l contains $O = {}^{l-r}/_{w-r}$ observation windows, which corresponds to $O - 1$ transitions to a *Healthy* state before a transition to a *Degrading* state. An estimate for the probability of degradation onset is then $\hat{a}_{12} = {}^{1}/_{O}$. This can easily be extended if multiple I_H datasets are available. For N different I_H datasets with observation windows $[O_1 \quad \ldots \quad O_N]$, the maximum likelihood estimate for a_{12} is defined by Eq. 3.4.

$$\hat{a}_{12} = \frac{N}{\sum_{i=1}^{N} O_i} \tag{3.4}$$

Initial state probabilities π_H and π_D should be defined to reflect prior expectations of repair success. Probability values can be defined as $\pi_H = \pi_D = 0.5$ to make an unbiased estimation, or set according to historical repair success rates.

3.3.4 Online Repair Assessment

The repair assessment process analyzes health index values measured after a machine has been brought online to assess the quality of repair procedures and should be repeated when new measurements become available. Post-repair index values are aggregated in an I_{PR} dataset and grouped into observation windows in an identical manner as the I_H and I_D datasets. Linear regression is then applied over each window

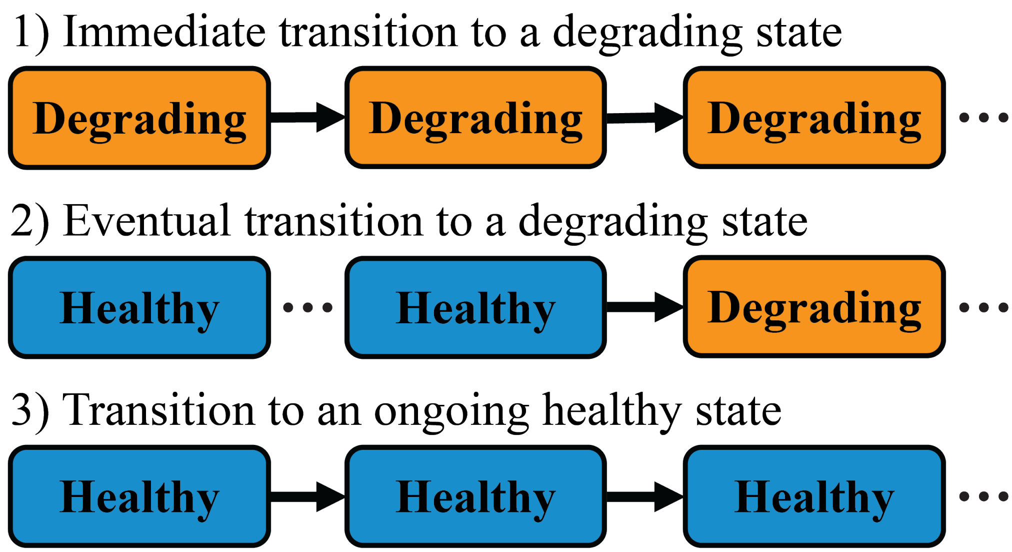

Figure 3.8: Categories of possible health state sequences after repair procedures

to calculate a time-series sequence of slope coefficients, B_{PR}. The Viterbi algorithm [116] is carried out to determine the most likely post-repair state sequence given the hidden Markov health model and observation sequence B_{PR}.

The most likely state sequence will have one of the three structures listed in Fig. 3.8. Sequence structures 1 and 3 describe uniform sequences, with entirely *Degrading* and *Healthy* states, respectively. Structure 2 includes an initial *Healthy* state followed by a transition to a *Degrading* state. The initial state in the most likely state sequence is used to classify repair procedures as successful or unsuccessful. Sequences with structure 1 exhibit an initial *Degrading* state, indicating an unsuccessful repair procedure. Sequences with structure 2 or 3 exhibit an initial *Healthy* state, indicating a successful repair procedure.

It may be necessary to make a repair quality assessment that is more detailed than binary classification. In this case, repairs can be assessed based on how quickly a machine re-enters the *Degrading* state. Estimated state sequences with structure 1 would classify a repair procedure as immediately unsuccessful. The time of degradation onset in sequences with structure 2 could be recorded to flag repairs that resulted in a quick re-occurrence of degrading behavior. A sequence with structure 3 would suggest that not enough information is available to assess the quality of a repair procedure.

3.3.5 Case Study

The following subsection details a case study that uses historical data from a centrifugal pump operating in a light hydrocarbon manufacturing plant to assess the quality of multiple repair procedures. One of the available features for this piece of equipment is peak proximity magnitude, which describes the maximum displacement of the pump over a short sampling interval and is related to machine vibration. System experts have indicated that elevated proximity measurements are associated with machine degradation. So, this feature is selected as the case study health index (i).

Health index measurements over a 6-month period are available for analysis. Repair logs indicate that the beginning of the dataset coincides with a successful repair event and that two transitions to the *Standby* state occurred. Index measurements from these *Standby* periods are omitted from the dataset because degradation was assumed to be unaffected by these interruptions. Three repair procedures were carried out, with logs indicating that two unsuccessful repair procedures, at times $t_{\delta,1}$ and $t_{\delta,2}$, were followed by a third, successful repair at time $t_{\delta,3}$. This case study aims to retroactively classify the repair procedures using index measurements taken after each one, with the goal of arriving at classifications that match those given in the repair logs.

Index values measured between the start of this dataset (at time t_ϵ) and the first shutdown for repair event (at time $t_{\delta,1}$) act as historical data. A linear trend during equipment degradation is shown to be approximately satisfied here based on the general increase in displacement values preceding the first repair event. By identifying

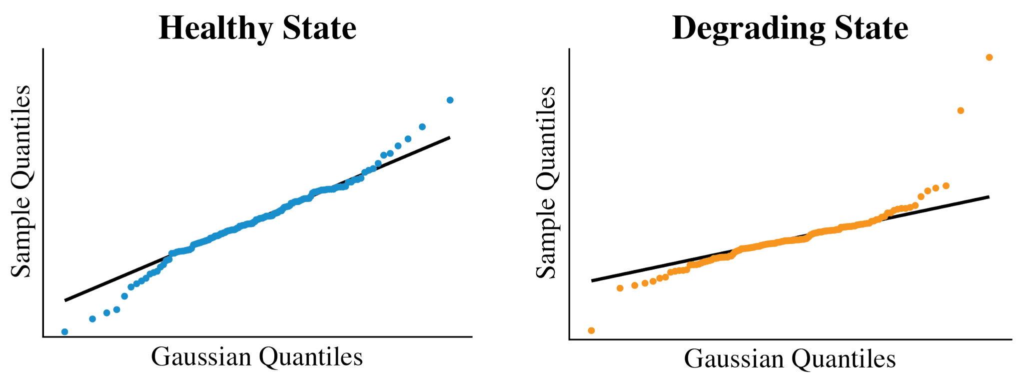

Figure 3.9: Gaussian QQ plots for historical case study data from healthy and degrading periods

the subset of index values with a maximum R^2 statistic, the time of degradation onset (t_α) is estimated.

A window size of 12 values with 6-value window overlap is selected to compile B_H and B_D datasets from the historical data. Unbiased initial state probabilities $\pi_H = \pi_D = 0.5$ are used here because no historical repair success rate is known. The mean and standard deviation of B_H and B_D characterize Gaussian observation probability distributions for a hidden Markov health model.

The assumption that b coefficients adhere to a Gaussian distribution in both health states is supported by the quantile-quantile plots in Fig. 3.9. Except for a small number of outliers, most observations are clustered around the reference line that denotes perfect agreement between the sample and Gaussian distributions. When these outliers, three observations in the *Degrading* state and two observations in the *Healthy* state, are excluded, a Shapiro-Wilk test at the 95% confidence level fails to reject the null hypothesis that the samples are drawn from a normally distributed population.

The toolbox published by [117] is used to repeatedly carry out the Viterbi algorithm, beginning with the first coefficient observed after repair then incorporating coefficients sequentially. Repair procedures are classified as successful or unsuccessful based on the initial state of the estimated health state sequence. Results are shown in

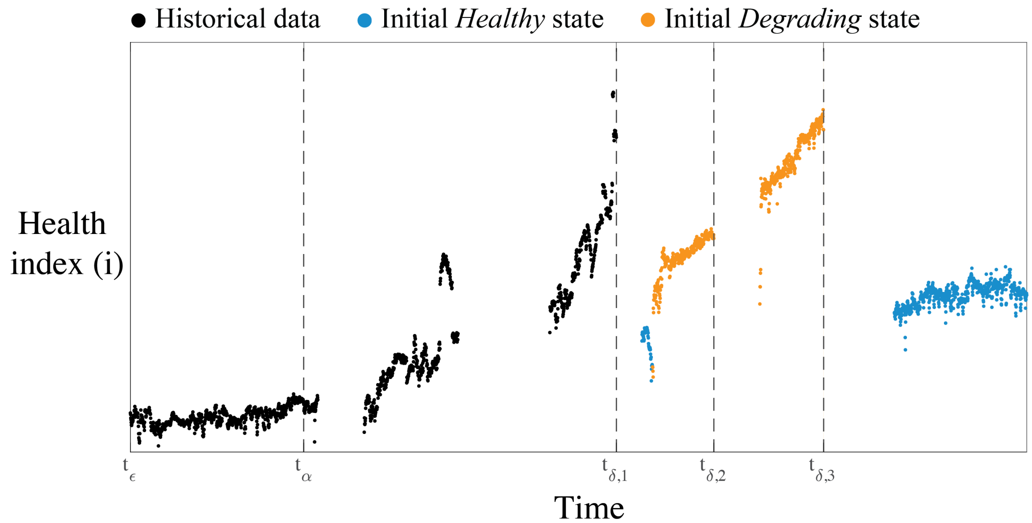

Figure 3.10: Case study health index dataset with post-repair classifications

Fig. 3.10, where the full health index dataset is segmented by events from the general operating model. The color of each data point indicates the initial state in the most likely health state sequence given values up to that point. The assessment process converges to an accurate classification after all three repair procedures. The second and third repairs procedures are immediately classified correctly, while the first repair is classified correctly after 53 hours.

A traditional approach to health monitoring uses feature-specific healthy, warning, and alarm regions to classify machine health. Applying this approach to the case study data results in all repairs being classified as successful, because peak proximity measurements never exceeded the feature's warning threshold. It is also common for existing methods to make classifications based on feature magnitudes alone. When peak proximity magnitudes replace b coefficients as observations in the hidden Markov health model, the first two repairs are correctly classified, but the third repair is misclassified. The limitations of these alternative methods reinforce the need for repair assessment to consider trends in feature values which may shift in magnitude between periods of online operation.

To test the the proposed approach further, the process is repeated with two other features. With input electrical power as a health index, assessment of the first repair converges to a correct classification after 48 hours, while the second and third repairs are immediately classified correctly. With vibration measurements from an accelerometer as a health index, the first and second repairs are immediately classified correctly, while assessment of the third repair converges to a correct classification after 24 hours.

3.4 Multi-Stage Fault Prediction in Rolling Element Bearings

In addition to contextualizing transitions between online and offline periods, the general operating model proposed in this chapter supports segmenting the *Degrading* state into multiple sub-states, as shown in Fig. 3.3. Representing system degradation in this way is useful if machine signal behavior changes as health problems worsen. This section presents an example of this phenomenon in rolling element bearings and demonstrates how state-based PHM is advantageous for monitoring system health. First, a multi-stage model of rolling element bearing degradation is introduced. Then, an online method for estimating the health state of rolling element bearings and predicting future faults based on real-time vibration measurements is presented.

3.4.1 Health Model Structure

In the literature, bearing lifetime is commonly represented as a two-state process [114, 118]. The first state is the *Healthy* state, characterized by steady-state behavior of bearing features, and the second state is the *Degrading* state, characterized by time-series trends in machine features that can be used to detect and track bearing deterioration. However, the degradation of bearings can be complex, such that a single model is unable to track the entire process. So, this study augments the fundamental

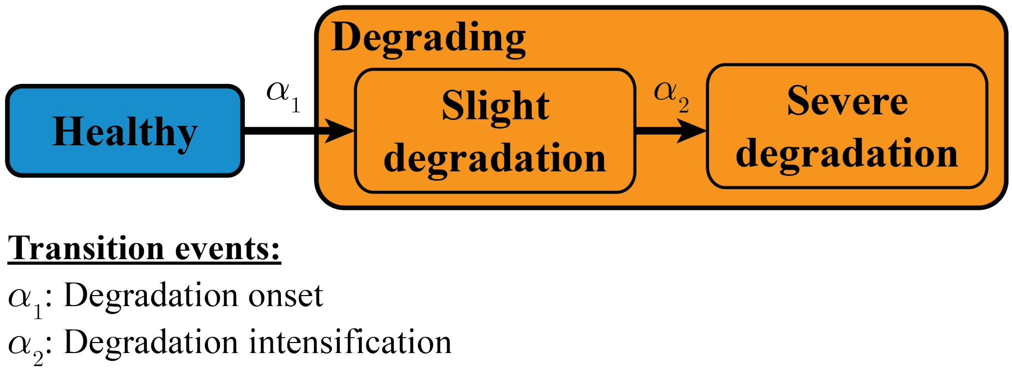

Figure 3.11: Multi-stage model of rolling element bearing degradation

two-state bearing model by splitting the *Degrading* state into two sub-states, *Slight Degradation* and *Severe Degradation*, as shown in Fig. 3.11. As described in later sections, the addition of a *Slight Degradation* state allows the health model to detect the presence of degradation, a valuable warning flag for system operators, even before the degradation process has progressed far enough to make reliable time to failure (TTF) predictions.

The steady-state descriptor introduced in Section 3.3 is carried over here to describe bearing behavior in the *Healthy* state. We make the assumption that the distributions of machine features in the *Healthy* state are approximately normal within a bounded range of a normal mean. When a degradation onset event (α_1) occurs, this means the bearing is about to degrade. The detection of a degradation onset event triggers the bearing to enter the *Slight Degradation* state. Generally, the *Slight Degradation* state is characterized by a departure from steady-state behavior and the emergence of time-series trends in one or more features. Within this study, a linear model is used to track slight degradation.

Following this state, a degradation intensification event (α_2) can be detected, at which point the degradation process is accelerated such that the model used in the *Slight Degradation* state can no longer describe the bearing's degradation. At this point, the bearing enters the *Severe Degradation* state. In this state, the degrada-

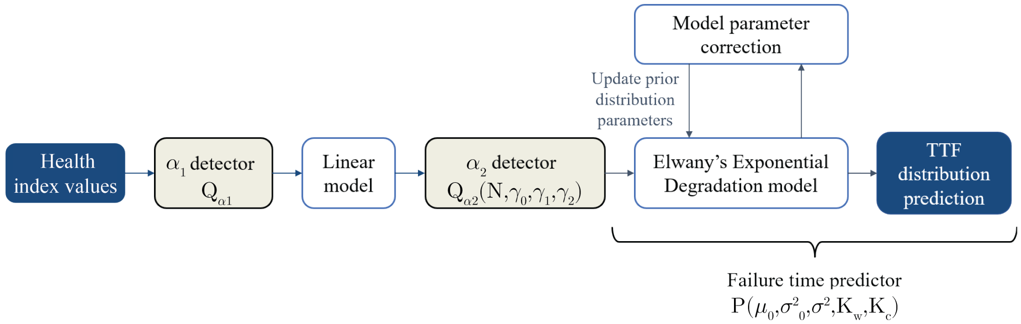

Figure 3.12: Computations and flow of information involved in the method used to track multi-stage degradation in rolling element bearings

tion process has progressed enough that it becomes possible to make reliable TTF predictions using a new degradation model. An exponential model is used here to track severe degradation, a choice that is supported by previous studies on bearing degradation [41, 42, 119].

The failure prediction process proposed in this section is illustrated in Fig. 3.12. All components necessary to predict the TTF distribution of a bearing are included in an entity called a multi-state model (M). A multi-state health-model consists of a model for degradation onset detection ($Q_{\alpha 1}$), a model for degradation intensification ($Q_{\alpha 2}$), and a failure time predictor model (P). Each of these components are described later sections, and can be written as functions of their model parameters, as in Fig. 3.12.

The proposed method uses a combination of machine features as the multivariate detection health index $I_D = (i_1, i_2, \cdots i_m)$ to detect the occurrence of the degradation onset event (α_1) for each bearing. The features chosen here should show dynamic behaviors when in the early stage of bearing degradation, deviating from the uniform behavior in *Healthy* state. The method for α_1 detection consists of the detection based on each individual feature i_d ($d \in \mathbb{Z}, 1 \leq d \leq m$), which is normalized independently. The α_1 detection begins with a window of n feature values, where the mean of these n values in the window (X) is computed. Meanwhile, the mean (a) and the standard deviation (σ) of all of the measured values are computed. As the window moves, the mean value over the window ($\overline{X}$) is continuously compared to the historical mean (a). Based on the assumption that all the features included in I_D follow normal distributions in the *Healthy* state, approximately 99.7% of the data points should lie within three standard deviations of the mean. So we set the threshold to $a \pm 3\sigma$. Thus, if

$$|\overline{X} - a| > 3\sigma, \tag{3.5}$$

then a candidate degradation onset is detected. If a candidate degradation onset is detected in all of the features $i_d \in I_D$, then α_1 is detected and the bearing should transition to *Slight Degradation* state. Alternatively, other α_1 detection criteria, such as a majority of features crossing the threshold, can also be used in different system contexts.

Once a bearing enters a *Slight Degradation* state, a univariate prognostic health index I_M is used to track degradation. The physical quantity represented by I_M can vary based on the bearing and the monitoring system's sensing capabilities. For example, this index may track the internal bearing temperature or the power present in a particular vibration frequency band. So, knowledge from system experts should be relied upon to select an index that reliably indicates system degradation. Automated processes for health index selection proposed in recent literature can also be utilized [120, 121].

The approach for degradation intensification (α_2) detection begins with a window of N prognostic health index values. The moving window is updated once a new data point is observed. In each iteration, a least squares linear regression and an exponential regression are performed over the window. Additionally, a linear regression model is fitted to all of the observed data. An example of the expected behavior of three regressions during the detection of α_2 is shown in Fig. 3.13. The R^2 values are computed for each regression model. The R^2 thresholds γ_0, γ_1, and γ_2 are predefined for

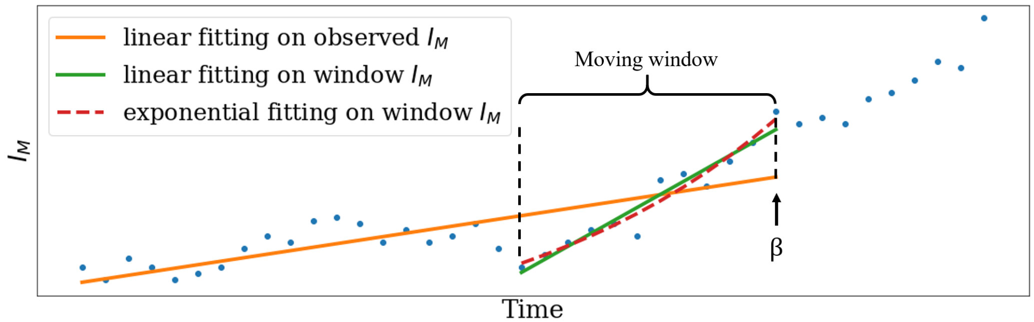

Figure 3.13: Illustration of regression models used to detect degradation intensification (α_2)

the linear regression on the window data, exponential regression on window data and the linear regression on all of the observed data, respectively. Usually, these thresholds are initially determined from historical instances. The condition that represents the detection of α_2 is given by

$$\begin{cases} R_0^2 < \gamma_0 \\ R_1^2 > \gamma_1 \\ R_2^2 < \gamma_2 \end{cases} \tag{3.6}$$

where R_0^2 , R_1^2 , and R_2^2 are the R^2 values for the linear regression on window data, exponential regression on window data, and the linear regression on all observed data. The first instance when all these regression values violate their respective thresholds simultaneously indicates the time of degradation intensification (α_2).

3.4.2 Prediction of TTF Distribution with Bayesian Updating Method

Once degradation intensification (α_2) is detected, the bearing is assumed to be in the *Severe Degradation* state, for which an exponential random coefficient model is applied to predict the distribution of TTF. The method described here treats slight degradation and severe degradation as separate processes with distinct trend models. So, only prognostic health index values measured after degradation intensification are used to fit the exponential severe degradation model. The similarity between a linear trend and an early exponential trend suggests that it may be reasonable to the view slight degradation stage as the beginning of a single exponential degradation process. However, our analysis of several bearing failure instances has shown that segmenting the degradation process results in more accurate TTF predictions.

Random coefficient models are often used to describe stochastic degradation processes by adding random coefficients into degradation models [19]. When online measurements are available to track the process being modeled, the distributions of the random coefficients can be adapted using a Bayesian update rule, as in [41]. In [42], linear and exponential random coefficient models with error terms that follow Brownian motion were proposed and the mean remaining life was used for TTF estimation. We extend the random coefficient model from these works and incorporate it into the proposed multi-state health model.

For simplicity of computation, we assume the exponential curve passes through the last data point of the linear curve in the *Slight Degradation* state to fix the offset (b) of the exponential model. The exponential random coefficient model with one parameter can be written in the form

$$S(t_k) = \phi \cdot e^{\theta t_k + \varepsilon(t_k)} + b, \tag{3.7}$$

where $S(t_k)$ is the value of the prognostic health index I_M at time t_k, ϕ is a constant deterministic parameter, θ is a random variable coefficient assumed to follow a normal distribution ($\pi(\theta)$) with mean μ_0 and variance σ_0^2, b is the fixed offset determined by the linear *Slight Degradation* model, and $\varepsilon(t_k)$ is an error term following a Brownian motion with mean zero and variance parameter σ^2. For mathematical convenience, we take the logarithm of $S(t_k)$. Then the model can be written as follows:

$$L(t_k) = \ln(S(t_k) - b) = \phi' + \theta t_k + \varepsilon(t_k), \tag{3.8}$$

where $L(t_k)$ is the logarithm of $S(t_k)$ and $\phi' = \ln(\phi)$.

A Bayesian update method is used to adapt the exponential model parameter θ when new measurements of the prognostic health index are made. With this approach, the prior distribution of the model parameter is updated based on a new measurement, resulting in a posterior distribution that becomes the prior for the next measurement.

Equations to compute the posterior distribution parameters are derived in [42] and defined by Eq. 3.9 and Eq. 3.10.

$$\mu_{\theta,t_k} = \frac{\mu_0\sigma^2 + (L(t_k) - \phi')\,\sigma_0^2}{t_k\sigma_0^2 + \sigma^2} \tag{3.9}$$

$$\sigma^2_{\theta,t_k} = \frac{\sigma^2\sigma_0^2}{t_k\sigma_0^2 + \sigma^2} \tag{3.10}$$

One limitation of this update method, however, is that the closed-form solutions in Eq. 3.9 and Eq. 3.10 force the exponential model to weigh each historical data point equally, thus making the exponential curve emphasize early data and fail to capture the rapid increase when the TTF becomes small. In order to solve this problem, our proposed method introduces a correction term to adjust μ_{θ,t_k} before using it to update the prior distribution for the next prediction. The error correction process can make the model more adaptive to the rising trends of actual data, and improve the accuracy of the TTF estimation.

To correct the posterior mean for θ at time t_k, an error term is defined, as shown in Eq. 3.11, as the sum of the weighted difference between actual and predicted I_M from the beginning of the *Severe Degradation* state to time t_k, when the most recent data is observed.

$$e(t_k) = \sum_{i=1}^{k} w(K_w, t_i) \cdot (s(t_i) - \tilde{S}(t_i)) \tag{3.11}$$

In Eq. 3.11, $e(t_k)$ is an error term for the correction of μ_{θ,t_k} at time t_k, $s(t_i)$ is the actual I_M at t_i, $\tilde{S}(t_i)$ is the predicted I_M calculated as $\tilde{S}(t_i) = \phi \cdot \exp(\mu_{\theta,t_k} \cdot t_k) + b$, and

$w(K_w, t_i)$ is an exponential weighting term, $w(K_w, t_i) = (1 - K_w)K_w^{k-i}$. A correction term is given as

$$\Delta\theta(t_k) = K_c \cdot e(t_k), \tag{3.12}$$

where K_c is a constant correction gain. Now, the value we use to update the mean of the model parameter θ in the prior distribution for the next prediction is

$$\mu'_{\theta,t_k} = \mu_{\theta,t_k} + \Delta\theta(t_k). \tag{3.13}$$

Having computed the posterior distribution of θ at time t_k, we would now like to determine the distribution of the TTF of the monitored bearing. For this purpose, we need to find a predefined failure threshold δ for the prognostic health index I_M. In general, such failure thresholds are not always clearly defined and are mostly unavailable in on-line analysis, so estimating one for a given application requires knowledge of industrial standards and machine expertise [41]. The following method is an extension of the approach in [42].

Give the failure threshold δ and a time t_k, we define the random variable $L(t_k + t)$ as the logged value of I_M observed after t time units, $t > 0$. Given $L(t_1), \cdots, L(t_k)$ observed at times $t_1, \cdots, t_k$, the mean and variance of $L(t_k + t)$ are given as

$$\tilde{\mu}\left(t_k + t\right) = \mu_{\theta,t_k} t + L\left(t_k\right) \tag{3.14}$$

$$\tilde{\sigma}^2\left(t_k + t\right) = \sigma^2_{\theta,t_k} t^2 + \sigma^2 t. \tag{3.15}$$

Next, we define a random variable T to be the remaining life of a partially degraded bearing such that $L(t_k + T) = \delta$. Then, the conditional cumulative distribution function (CDF) of T given $L(t_1), \cdots, L(t_k)$ can be computed as follows:

$$P(T \leq t | L(t_1), \cdots, L(t_k)) = \Phi \left(\frac{\tilde{\mu}\,(t + t_k) - \delta}{\tilde{\sigma}\,(t + t_k)} \right), \tag{3.16}$$

where $\Phi(\cdot)$ is the CDF of a standard normal random variable. Now, we have shown how to find the conditional CDF of the remaining life of a bearing at time t_k. We can easily find the CDF of the TTF by adding current time t_k to the remaining life. This procedure for Bayesian updating of prior distribution parameters and estimation of the TTF distribution is performed every time a new prognostic health index I_M is observed by the model.

3.4.3 Case Study

In this section, we describe how the proposed multi-state health model and training process is implemented to analyze the bearing run-to-failure dataset provided by the Intelligent Maintenance Systems (IMS) Center [122] and discuss the results of the model training experiments.

The proposed method is tested on the IMS bearing data. Four bearings are installed on the shaft and high sensitivity quartz accelerometers are installed on the bearing housing. An alternating current motor is coupled to the shaft to keep the rotation speed constant at 2000 rotations per minute (RPM) and a radial load of 6000 lbs is applied onto the shaft and bearings by a spring mechanism. The data packet includes the vibration data of four bearings in the test system through three test-to-failure experiments. Each data set that corresponds to each test consists of individual files that are 1-second vibration signal snapshots recorded every 10 minutes. Each file consists of 20,480 points with the sampling rate set at 20 kHz. These data are recorded every 10 minutes (except the first 43 files for test 1 are taken every 5 minutes).

Table 3.1: Statistic feature expressions.

Feature	**Expression**
Kurtosis	$\frac{1}{N}\sum_{i=1}^{N}\frac{(x_i-\overline{x})^4}{\sigma^4}$
Skewness	$\frac{1}{N}\sum_{i=1}^{N}\frac{(x_i-\overline{x})^3}{\sigma^3}$
Peak Frequency	$\max(f)$
Peak-to-peak	$x_{\max} - x_{\min}$
Root Mean Square (RMS)	$(\frac{1}{N}\sum_{i=1}^{N} x_i^2)^{\frac{1}{2}}$

To detect degradation onset in this dataset, a combination of kurtosis, skewness, peak frequency, and peak-to-peak features, defined in Table 3.1, was used as the multivariate detection health index I_D. The root mean square (RMS) feature, as defined in Table 3.1, was selected as the univariate prognostic health index I_M for the detection of degradation intensification (α_2) and the prediction of TTF distribution. In order to obtain a fixed failure threshold value δ, we perform an ordinary linear regression on the last 10 data points before failure, and take the last value from the linear model as δ.

We first applied the proposed model to a single bearing, bearing 3 in test 3 of the IMS dataset, to observe how the TTF distribution changes as more data are given to the model. The model parameters are fine-tuned after several off-line tests. The bottom plot in Fig. 3.14 shows the two-stage degradation regression model fit to this bearing's prognostic health index I_M. The predicted trajectory for I_M in the *Slight Degradation* state is generated by a linear model, while that for I_M in the *Severe Degradation* state is generated by an exponential model. We find that the bearing's prognostic health index I_M closely follows the degradation trajectories predicted by the proposed model and that the exponential model reaches the failure threshold at approximately the same time as the I_M measurements. This behavior supports the decision to model bearing degradation as a two-stage process with linear and exponential trajectories. For comparison, the top plot in Fig. 3.14 shows the exponential regression model that would be generated if bearing degradation was treated as a

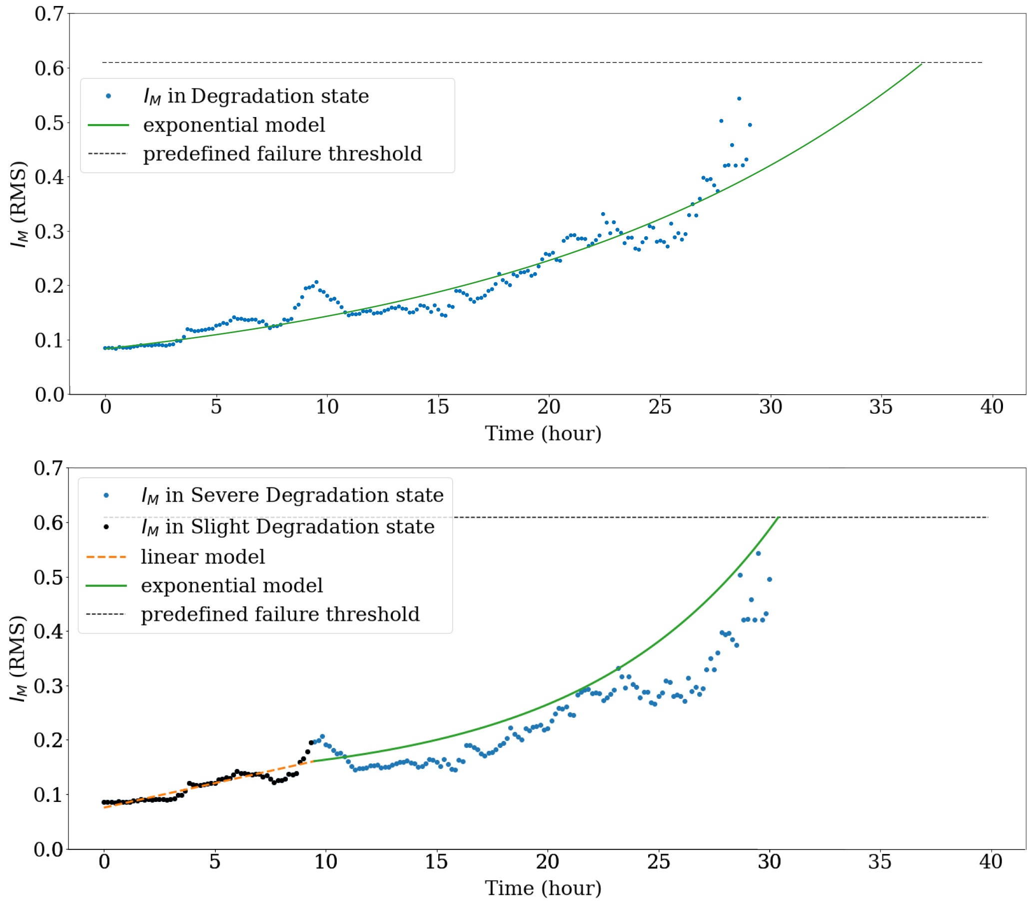

Figure 3.14: I_M degradation trajectories and regression models for single-stage degradation process (top) and 2-stage degradation process (bottom)

single-stage process. This plot shows close agreement between the exponential model and early I_M measurements, but the exponential model does not adapt to the increased rate of bearing degradation later in the test. As a result, the exponential model reaches the failure threshold significantly after the I_M measurements, leading to inaccurate TTF predictions.

Fig. 3.15 shows the probability density function (PDF) of TTF at different degrees of degradation. The bearing is at 0% severe degradation the instant that degradation intensification (α_2) is detected, and at 100% severe degradation the instant that it fails. As time passes, the peak of the PDF approaches zero hours, indicating that TTF predictions become smaller as the degradation progresses. Also, the peak value

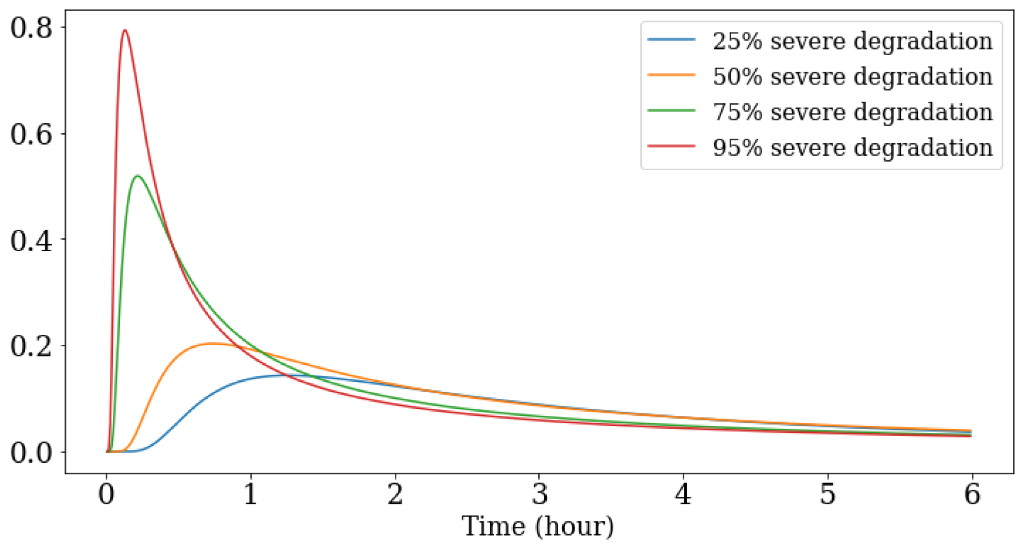

Figure 3.15: PDF of remaining useful life at different percentage of bearing's life

Table 3.2: TTF prediction results on bearing 3 in test 3 in the IMS dataset

% severe degradation time	Actual failure time (hour)	Predicted failure time (hour)	5th percentile	95th percentile	% Error D
25%	30.5	33.02	25.60	>100	8.25%
50%	30.5	31.31	25.25	>100	2.64%
75%	30.5	30.96	24.86	76.06	1.50%
95%	30.5	31.02	24.79	63.69	1.69%

of the PDF becomes larger as time passes, indicating that TTF can be predicted with higher confidence.

Table 3.2 illustrates the prediction results on bearing 3 in test 3 at 25%, 50%, 75%, and 95% percentage of the severe degradation time. At each listed prediction time, the predicted TTF, an approximate 90th percentile interval, and a percentage prediction error D are computed and shown in the table. As more degradation data is seen by the model, the predicted TTF approaches the true TTF. The range of the 90th percentile interval also becomes smaller, illustrating that model prediction becomes more confident.

3.5 Conclusions

In this chapter, a general operating model for industrial equipment is presented. For widespread deployment of predictive maintenance strategies to be feasible in manufacturing operations, representations of equipment health and health modeling outputs must be standardized and consistent across the equipment in a factory. The online states in the proposed model provide general representations of equipment health with respect to quantitative system health specifications. The bearing fault prediction method presented here shows how this model supports the development of state-based PHM solutions to monitor multi-stage degradation processes. The transitions to offline states can be used to capture gaps in machine operation that are common in manufacturing environments. The repair quality assessment method presented here shows how these transitions can be used to pre-process historical data and provides an approach to differentiate between successful and unsuccessful repair events.

CHAPTER 4

An Extensible Digital Twin Framework for PHM

This chapter presents the work published in [40]. As discussed in Section 2.3, existing work to specify general architectures for prognostics and health management (PHM) solutions has focused on individual applications and has not fully investigated the ability to re-use and combine digital twins (DTs) to represent manufacturing equipment. As manufacturers continue to deploy PHM solutions across a wider range of systems these capabilities will be critical. The contribution of this chapter is a framework for PHM that utilizes DTs to monitor the health of complex manufacturing systems. This framework defines a set of general, PHM-related DT classes and specifies their internal architectures and input/output behavior in detail. Each of these DTs classes is designed to fulfill a state estimation or prediction purpose that is derived from the operating model introduced in Chapter 3. In this way, the DT framework provides a structured approach to deploying this operating model to monitor the health of mechanical systems in real-time. Formal communication interfaces and aggregation relationships are defined here as well to allow these DTs to be re-used and combined.

The rest of the chapter is organized as follows. Section 4.1 discusses the concept of purpose-driven digital twins in general, then defines several classes of DTs that are used for health monitoring (collectively referred to as PHM DTs) and the aggregation relationships that can be used to communicate between them. Section 4.2 provides a general architecture for these PHM DTs and discusses the internal communications within each DT class. Section 4.3 presents a case study that uses the proposed framework to model the health of an industrial pump by aggregating the outputs of several subsystem and component DTs. Conclusions of this chapter are given in Section 4.4.

4.1 PHM Digital Twins

4.1.1 Digital Twin Definition

The proposed framework adopts the digital twin definition that is put forth in [25]:

> *Digital twin:* A purpose-driven virtual representation of a physical manufacturing system or process that uses data to remain synchronized with its physical counterpart. Its capability and scope are limited to a pre-defined purpose and application environment.

A crucial aspect of this definition is that DTs should be developed for a specific purpose and provide outputs that speak to that purpose. Therefore, multiple DTs may be used to represent a single system if it is advantageous to analyze multiple aspects of a system (throughput and machine health, for example). This is a distinction from previous research and research in other industries that seeks to develop unified DTs that describe every aspect of a system. A purpose-driven framework can be more appropriate for manufacturing operations that require actionable insights about many systems that may be modeled from different perspectives and at different resolutions.

Moyne et al.'s [25] depiction of a DT, as seen in Fig. 4.1, shows that a DT combines a computational engine with one or more models . Each model emulates some aspect of the physical system or process being represented. Models may be physics-based,

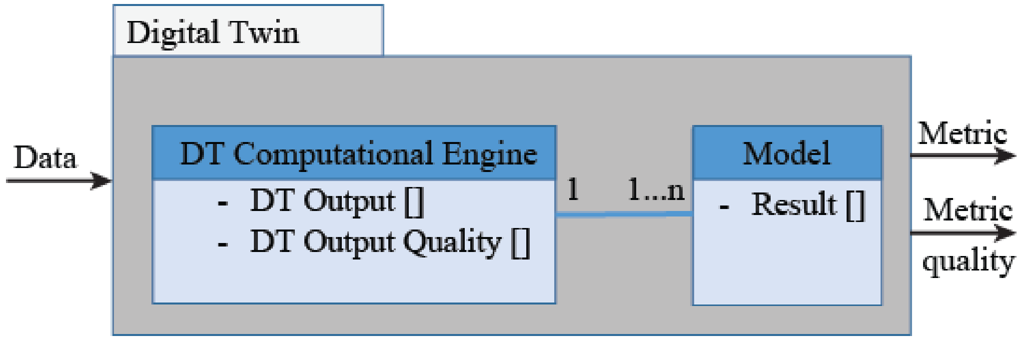

Figure 4.1: General depiction of the input-output behavior and internal components of a digital twin

data-driven, or hybrid, depending on the system being represented and the DT's purpose. A DT's computational engine implements any desired model support functions, such as computing model input features from raw machine signals, tracking a model's alignment with a physical system, or updating models based on new information, and synthesizes model results to generate one or more DT output metrics, which fulfill the DT's purpose. Examples of output metrics include indications of events (e.g., equipment fault), real-time estimates of key performance indicators (e.g., throughput), and recommendations for actions (e.g., modified assembly line configuration). DTs must also generate a quantity that describes the quality of its output metric(s).

As presented in [25], the DTs discussed here are considered to be instantiations of pre-existing DT classes. A DT class describes the purpose and output metric(s) of DTs belonging to the class. Specific implementation details, including the models that inform a DT and the processes carried out by a computational engine, may vary between DT instances. This allows system experts to define general DT classes that can be applied across systems while preserving the flexibility to model individual machines and components as they see fit. Subsection 4.1.2 and Subsection 4.1.3 define a set of DT classes that are used for instantaneous health monitoring and predictive health monitoring, respectively.

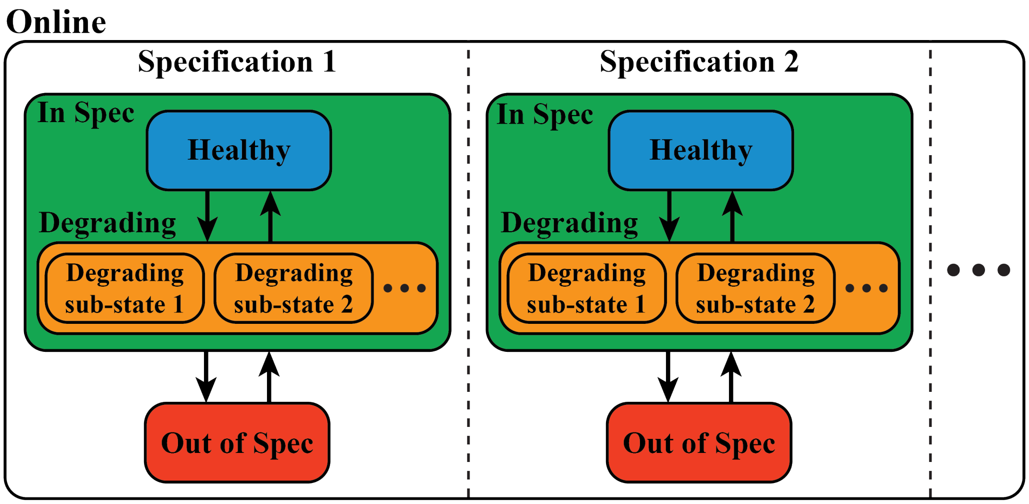

Figure 4.2: Statechart used to represent system health with respect to multiple health specifications during online periods

4.1.2 Instantaneous Health Monitoring

The proposed DT framework for PHM draws upon a discrete-state model, shown in Fig. 4.2, to represent mechanical system health based on compliance with system specifications. The entire model exists within an *Online* super-state that is active whenever a system is operating as part of a manufacturing process.

For each specification, the model includes two instantaneous health states, *In Spec* and *Out of Spec*, that describe a system's compliance with that specification at the current point in time. As discussed above, a system is *In Spec* when it is in compliance with the specification being modeled and *Out of Spec* when that specification is violated. During online monitoring, DTs within the framework provide instantaneous health state (IHS) estimates to communicate information about a system's compliance with a specification at a single point in time. An IHS estimate is defined as a 2-tuple consisting of a state (s_I) in the set {*In Spec*, *Out of Spec*} and a state probability (p_I) in the interval $[0.5, 1]$, as expressed below. The state s_I represents the most likely instantaneous health state of the system with respect to the specification being modeled, while p_I represents the probability that the system is in state s_I.

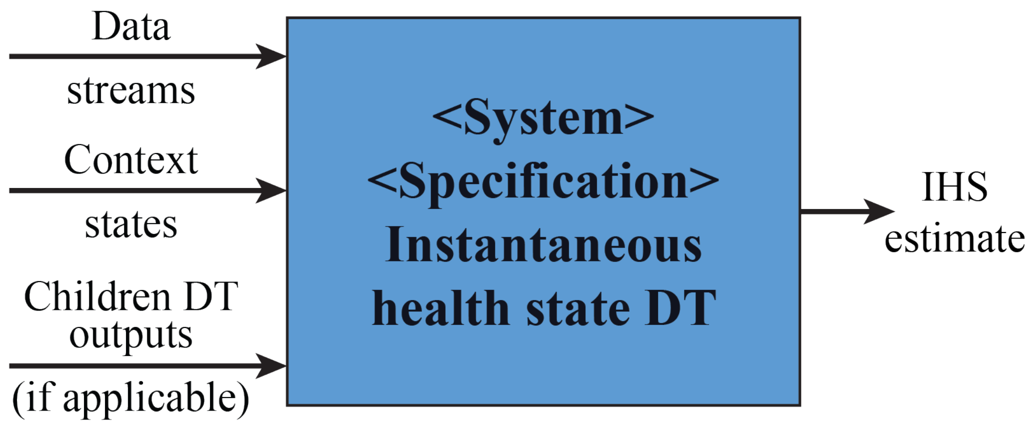

Figure 4.3: I/O behavior and naming convention of the instantaneous health state digital twin

$$\text{IHS estimate: } \{s_I, p_I\} \tag{4.1}$$

Within the DT framework, instantaneous health state DTs are used to compute IHS estimates for mechanical systems. The general I/O behavior of the IHS DT is shown in Fig. 4.3. Inputs to the DT include data streams that contain information about the health of the system being monitored (vibration signals for example) as well as context states that contain information about the system's current operating parameters (rotating speed or product type for example). If the IHS DT aggregates any other DTs, the outputs of those DTs will also be included in the DT inputs. More information on DT aggregation is provided in Subsection 4.1.4. Section 4.2 describes the computational and modeling resources that the IHS DT uses to compute IHS estimates based on this information.

4.1.3 Predictive Health Monitoring

In some applications, an instantaneous monitoring approach by itself can reduce machine downtime and maintenance costs. In other applications, there may be sufficient knowledge about system degradation to support a predictive monitoring approach. When this is the case, the DT classes described below can be used to generate predictive health state (PHS) estimates that provide information about system health in the future. Under a predictive monitoring approach, the *In Spec* state in Fig. 4.2 is further sub-divided into *Healthy* and *Degrading* states. When a system is *Healthy*, there are no indications of impending faults. When a system is *Degrading*, a known degradation mode is underway and prognosis can be carried out to estimate the time remaining before a fault occurs. Multiple *Degrading* sub-states may also be defined to describe systems that are susceptible to multiple degradation modes that impact the same specification (as discussed later in this sub-section) or complex degradation modes with multiple stages. Prognosis may be associated with one or more of these sub-states, e.g., degradation might be modeled as a sequence of degradation stages where quantifiable prognosis is only associated with the later degradation stages.

A PHS estimate is defined as a 3-tuple consisting of a degradation state (s_P) from the set $\{Healthy, Degrading\}$ (the set of *Degrading* sub-states can be substituted for the *Degrading* state when these exist), a state probability p_P in the interval $[0, 1]$ that describes the likelihood that the system is in state s_P, and a prediction of the time remaining until system fault, referred to as a remaining useful life (RUL) description. All RUL descriptions assume that the system is in state s_P, so the state probability p_P is not considered when computing RUL descriptions. Additionally, RUL descriptions assume that a system's context states, which capture all operating parameters such as system load, remain constant. RUL descriptions are commonly structured as an interval $[RUL_L, RUL_U]$, where $RUL_L, RUL_U \in \mathbb{R}^+$ represent the lower and upper bounds of an interval that estimates the time remaining until a system fault occurs. This interval should be associated with a predetermined confidence level. System RUL may also be described by a probability density function (PDF) for the system's RUL or a cumulative distribution function (CDF) that describes the probability that RUL will be less than time t. PHS estimates with these RUL descriptions are expressed below.

$$\text{PHS estimate with RUL interval: } \{s_P, p_P, [RUL_L, RUL_U]\} \tag{4.2}$$

$$\text{PHS estimate with RUL PDF: } \{s_P, p_P, P(RUL)\} \tag{4.3}$$

$$\text{PHS estimatewith RUL CDF: } \{s_P, p_P, P(RUL \leq t)\} \tag{4.4}$$

The PHS DT aggregates the outputs of degradation state and RUL DTs to produce a PHS estimate. The general I/O behavior of these DTs is shown in Fig. 4.4. These DTs have access to a set of data streams and context states, as well as DT outputs from any children DTs. A degradation state DT first determines a degradation state (s_P) and a state probability (p_P) for the system that are sent to the PHS DT. If the probability that the system is in the *Degrading* state crosses a lower threshold set by system experts, PHS estimation is halted, and the output of the PHS DT is simply the state $Healthy$. Otherwise s_P is sent to an RUL DT that contains the resources to compute the system's RUL description. The PHS DT aggregates this description with the current degradation state and state probability to obtain a PHS estimate.

Some systems may be susceptible to multiple degradation modes, and the DT framework offers multiple options for handling this. When degradation modes are treated as mutually exclusive, multiple degradation state DTs and RUL DTs may be linked to a single PHS DT that outputs multiple PHS estimates. For systems where degradation modes may occur simultaneously, a parent PHS DT can synthesize the results of multiple process-specific PHS DTs. The DT aggregation relationships that make this possible are detailed in Subsection 4.1.4.

4.1.4 Digital Twin Aggregation

Recent research has identified DT aggregation, the ability to combine DTs into multi-level hierarchies, as a promising tool for representing complex mechanical sys-

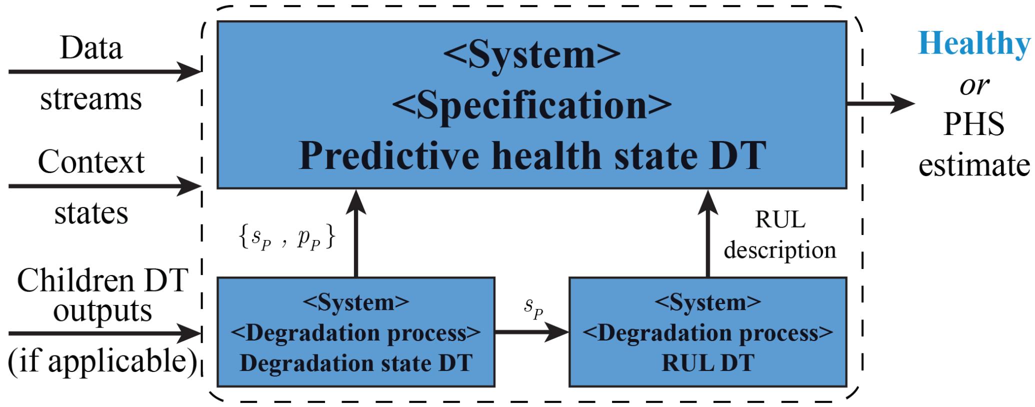

Figure 4.4: I/O behavior and naming conventions of digital twins used for predictive health state estimation

tems. In [91], Schroeder et al. discusses the value of using DT aggregation to model a factory by aggregating machine DTs, for example. The specifications developed by Plattform Industrie 4.0 for Asset Administration Shells (AASs), which have many similarities with DTs, also stress the importance of standardizing the structure and interfaces of AASs to facilitate communication and interoperability [93, 94]. [92] demonstrates how aggregation is useful in a predictive maintenance applications by defining an AAS aggregator that can synthesize the outputs of multiple AAS devices to provide system-level predictions.

In this framework, aggregation is used to model a "whole/part" relationship, where DTs representing a "whole" system (referred to as parent DTs) communicate with DTs that each represent a system part (referred to as children DTs). A detailed discussion of DT aggregation with purpose-driven DTs can be found in [25]. The DTs introduced in Subsection 4.1.3 utilize aggregation to accomplish predictive health monitoring using three DT classes. The remainder of this section discusses two other use cases for aggregation in the DT framework: modeling multiple system health specifications and modeling multiple system components.

4.1.4.1 Aggregation of System Health Specifications

A core principle of the DT framework is the multidimensional nature of mechanical system health. Modern manufacturing machines are typically subject to multiple system specifications, each of which may be monitored with different modeling resources. The DT framework represents multidimensional system health by allowing instantaneous health state DTs and predictive health state DTs to act as children to a parent DT, known as a health DT. In practice, it is logical to assume that the names of health DTs will begin with the name of the machine being monitored followed by the term "health DT". The remainder of this chapter will refer to health DTs as machine health DTs to avoid confusion with the DTs introduced previously. A machine health DT, designed to be the root of the aggregation hierarchy for a manufacturing machine, is responsible for compiling the DT outputs from its children DTs into multi-dimensional quantities that comprise its own DT output. The general I/O behavior of the machine health DT is shown in Fig. 4.5. DT aggregation hierarchies should be structured so that all modeling and computational work necessary to monitor a system is done by children DTs aggregated by a machine health DT. The IHS and PHS estimates reported to a machine health DT are then complete and independent of one another.

While a machine health DT will include all IHS estimates from its children in its multidimensional IHS estimate output, certain PHS estimates may be excluded from its multidimensional PHS estimate output. As shown in Fig. 4.2, the *Healthy* and *Degrading* states that predictive health monitoring seek to identify are sub-states of the *In Spec* state. PHS estimation can then be understood as a supplemental process to be carried out when there is a reasonable chance that a system is *In Spec*. For specifications where predictive monitoring is possible, a machine health DT uses a specification's IHS estimate to determine whether to report that specification's PHS estimate. If the state s_I is *Out of Spec*, and the state probability p_I exceeds a

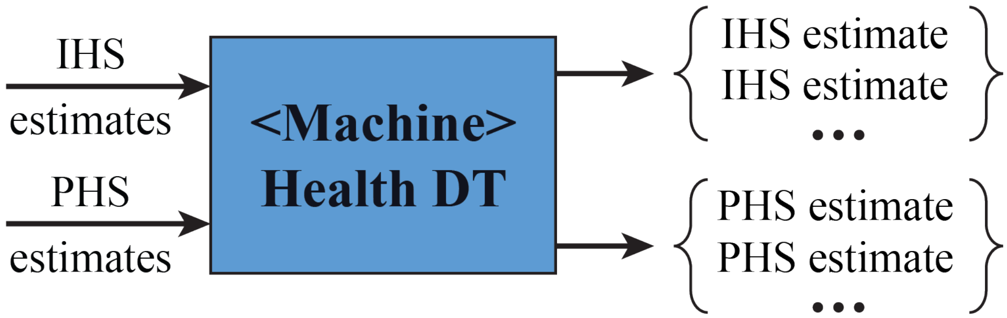

Figure 4.5: I/O behavior and naming convention of the health digital twin

predefined upper threshold, the PHS estimate is left empty. The selective reporting of PHS estimates does not halt PHS estimation in the background, however. In some applications, it may be necessary for degradation state and RUL DTs to continue to run even when a system is almost certainly *Out of Spec*. For example, when a system fault is triggered by a single component, ongoing degradation in other components should continue to be monitored so that predictions can be reported when the system fault is resolved.

4.1.4.2 Aggregation of System Components

Often, manufacturing machines are made up of multiple components that work together to help the machine achieve its function. For example, an industrial pump may rely upon a motor, a rotating shaft, and an impeller to maintain a minimum discharge flow rate. Degradation in any of these components may lead to a system fault, so all of them must be considered when monitoring this specification. DT aggregation can be used to model complex equipment by defining a system DT that acts as a parent to multiple component DTs. The system DT's inputs will then include the outputs of its component DTs.

Component DTs may be structured as instantaneous or predictive health state DTs based on the type of monitoring being conducted, and a new, component-specific specification that is associated with the DT's output should be defined for each component. Because system DTs receive data from their component DTs in the form of IHS or PHS estimates, this type of aggregation relationship is best suited for systems where component degradation can be considered independent or when there are well known conditional relationships between component degradation modes. With this approach, system DTs may implement recently-proposed methods that use hierarchical Bayesian networks to represent parent-child relationships between mechanical system components when estimating current and future machine health states [123, 124]. Cases in which component faults and degradation modes are closely coupled and system state estimates cannot be derived from component DT outputs should instead be modeled with a single PHS DT, potentially with multiple *Degrading* sub-states that capture the effects of overlapping degradation modes. This PHS DT may implement the RUL prediction method presented in [123] for systems that are comprised of multiple components with coupled degradation processes.

4.2 Digital Twin Architectures

This section details the architecture and flow of information within each of the classes of DTs specified by this framework, collectively referred to as PHM DTs. First, a general architecture inherited by all PHM DTs is presented. Then, details specific to each class of DT introduced in Section 4.1 are provided. Background information on purpose-driven DTs that provides a foundation for this framework can be found in [25] and [125].

4.2.1 General Architecture of PHM Digital Twins

Fig. 4.6 shows the architecture that is common across all PHM DTs. This architecture is an extension of the DT representation introduced in [25] and depicted in Fig. 4.1, with more details regarding the computational engine and the flow of information within PHM DTs. The capabilities of the computational engine are grouped

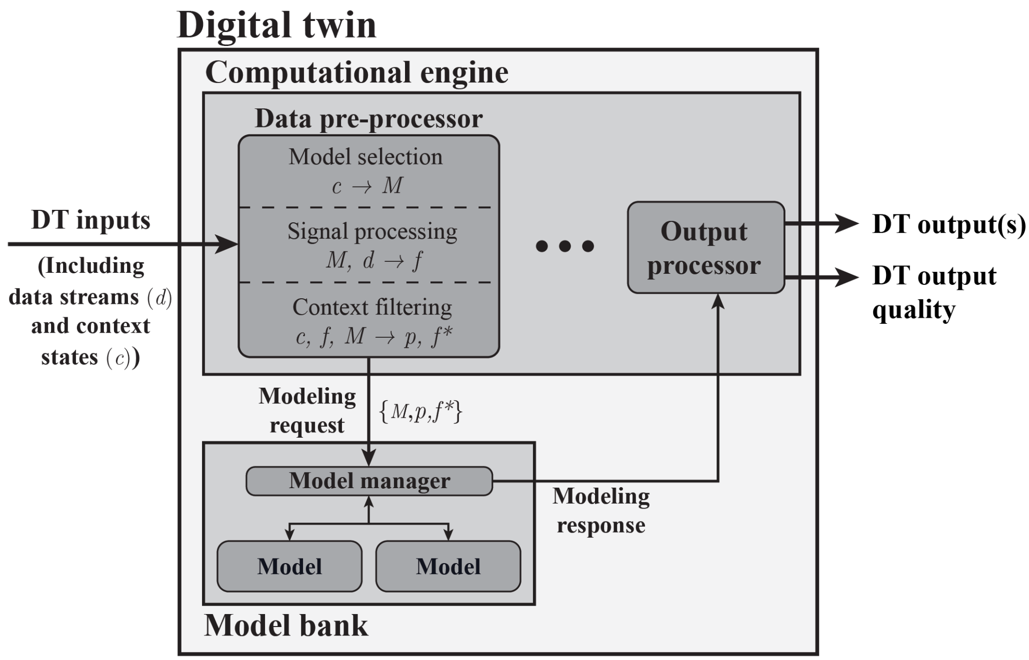

Figure 4.6: General architecture and flow of information within all DTs defined in this framework

into modules for the purpose of visualization. The data pre-processor module is responsible for all computations necessary to extract information from the DT's input signals and prepare a modeling request that is sent to the model bank. For this discussion, any DT inputs from children DTs will be considered part of the DT's data stream inputs. Data pre-processing functions are split into three categories: model selection, signal processing, and context filtering.

For systems with variable operating parameters, a context state (c) may be used to represent a system's current parameters. A DT's model bank can then support multiple models that can be queried based on the value of (c). This capability can be useful for systems with discrete operating modes, such as a machine that has different models trained for each recipe of product that it processes. Model selection computations determine the model, specified with a model identifier (M), that should be queried to determine the current IHS estimate. These computations are represented by the expression $c \rightarrow M$. While one or more models may not selected based on the

the current value of c, other computational engine modules can update models in the background to track machine drift or other system dynamics.

Signal processing computations extract features (f) from the DT input data streams (d) based on the input specifications for the selected model. Common signal processing operations include deriving frequency-domain features from time-series accelerometer measurements and applying moving window averages to raw sensor signals. These computations are represented by the expression $M, d \rightarrow f$.

Context filtering computations specify a set of modeling parameters (p) for the selected model based on the system's context state (c). This step allows a DT to make model transformations in response to the system's current operating conditions. Model transformations, such as shifting the value of a signal threshold or translating the expected distribution of signals from a $Healthy$ system, may be useful for systems with continuous operating parameters. If a continuous relationship between system context and modeling parameters can be identified, a DT can adapt to novel context states without training a new model. Additionally, feature transformation may also be carried out to adapt to different system contexts. This step maps a set of features (f) from one space to a transformed set of features (f^*) in another space. These computations are represented by the expression $c, f, M \rightarrow p, f^*$. The final data preprocessing step involves packaging the model specifier (M), transformed features (f^*), and model parameters (p) into a modeling request that is sent to the model bank.

DTs in the framework use one or more models, housed in a model bank, to make conclusions about system health. A model manager directs the flow of information within the model bank. During online estimation, this involves unpacking the information within modeling requests and querying the selected models with the specified modeling parameters. The model manager is also responsible for packaging modeling results, which may have a variety of forms depending on the types of models being used, into a modeling response that is sent back to the computational engine. Examples of modeling responses will be provided in the forthcoming sections on DT classes.

The output processor is the other computational engine module specified here, though additional modules can be developed and implemented in future work. This module is responsible for preparing the DT outputs. Output computations may not be necessary in some DTs, such as when instantaneous health monitoring is implemented with a limits-based approach. In cases like this, the output of a health model that contains the limit-checking logic can be passed through as the output of an IHS DT. Other applications, though, may use output computations to format more complex modeling results or to filter modeling results based on time-series dynamics.

4.2.2 Architecture of IHS Digital Twins

The IHS DT is used for online estimation of a system's compliance with a single specification at the current time. The structure of the data pre-processor module within the DT's computational engine is inherited from the general DT architecture described in Subsection 4.2.1. The output processor and the modeling responses that are used to determine DT outputs can have a variety of structures and formats depending on the application. An implementation approach that uses a hidden Markov model (HMM) within the output processor to account for time-series dynamics in machine behavior is described below.

Under this approach, models within an IHS DT, referred to as fault detection models, may be probabilistic or non-probabilistic. Non-probabilistic fault detection models, such as binary classifiers, are structured to provide an instantaneous health state classification ($\hat{s}_I$) as a modeling result. Probabilistic fault detection models are structured to provide the likelihood of observing the model inputs given each possible IHS of the system ($P(f^*|s_I)$). When a non-probabilistic model is utilized within an IHS DT, the model manager must maintain statistics describing the reliability of the model's results. A model's historical true positive rate is used to describe the reliability of $Out\ of\ Spec$ IHS classifications ($P(\hat{s}_I = Out\ of\ Spec|s_I = Out\ of\ Spec)$) while a model's historical true negative rate is used to describe the reliability of $In\ Spec$ IHS classifications ($P(\hat{s}_I = In\ Spec|s_I = In\ Spec)$). When the model manager receives results from the queried models, it then prepares a modeling response that is sent back to the computational engine. For non-probabilistic models, $P(\hat{s}_I|s_I)$ values for each IHS s_I are determined based on the model's IHS classification and these values are sent to the computational engine as the modeling response. For probabilistic models, the input feature probabilities ($P(f^*|s_I)$) are sent to the computational engine as the modeling response.

The output processor module within the computational engine is responsible for determining the quantities that make up the DT's IHS estimate output. Within the output processor is an HMM shown in the top image of Fig. 4.7, which represents the possible instantaneous health states of a system. A HMM-based approach allows for probabilistic modeling of physical states and state transitions in systems [29, 45, 68]. State transition probabilities should be specified based on knowledge of the system being modeled, the fault being monitored, and the sampling rate of data collected from the system. The automated method for estimating HMM transition probabilities from historical machine data presented in [33] is used here.

When new modeling responses are provided to the the output processor, the module uses the Viterbi algorithm to compute the most likely current state of the system and the likelihood of this state given a history of modeling responses up to the current point in time [68]. The values of $P(\hat{s}_I|s_I)$ or ($P(f^*|s_I)$) represent the observation probabilities for state s_I that are the inputs to the Viterbi algorithm. The state with the highest likelihood at the current time (s_I) is grouped with its corresponding state probability (p_I) into an IHS estimate that is the IHS DT output. It is worth noting that this output computation method requires a time-series history of modeling

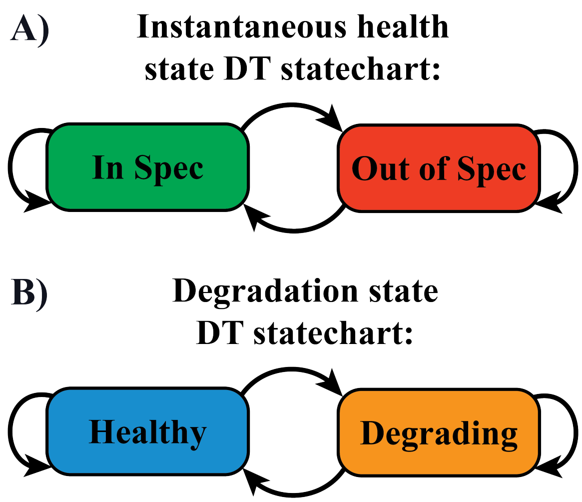

Figure 4.7: A) Hidden Markov model statechart used by the instantaneous health state DT, B) Hidden Markov model statechart used by the degradation state DT

responses, so recent modeling responses should be stored within the computational engine or in a database that the output processor can access. Subject matter expertise should be utilized to specify the length of modeling response history that is used to make state estimates. It may be advisable to truncate this history after a fixed period of time, or to reset the history whenever a context state change occurs.

4.2.3 Architecture of PHS Digital Twins

As depicted in Fig. 4.4, the DTs used for PHS monitoring are the PHS DT, the degradation state DT, and the RUL DT. The PHS DT's primary role is to aggregate the outputs of degradation state and RUL DTs. These simple operations can be implemented in a manner that is appropriate for current application, so details of the PHS DT architecture are not specified here. The bulk of the modeling and analysis work is carried out by the degradation state and RUL DTs, which are discussed below.

4.2.3.1 Degradation State Digital Twin

The degradation state DT is used to detect the presence of a degradation process that makes it possible to predict future system faults. This discussion assumes that no

Degrading sub-states are defined for the system being monitored. When *Degrading* sub-state are defined, the DT communications can easily be adapted by substituting the set of *Degrading* sub-states wherever the *Degrading* state is mentioned below. Like the IHS DT, the degradation state DT inherits from the general structure of the data pre-processor module. An HMM-based approach to estimating a system's degradation state and state probabilities, like the one presented for IHS DTs, can be also implemented within a degradation state DT's output processor.

With this approach, models within a degradation state DT, referred to as degradation detection models, may be non-probabilistic, providing degradation state classifications ($\hat{s}_P$) from the set $\{Healthy, Degrading\}$ or probabilistic, providing likelihood values for the model inputs given each state in the same set. When a non-probabilistic model is utilized, the degradation state DT's model manager must maintain the same model reliability statistics described in Subsection 4.2.2 for IHS DTs and prepares a modeling response in the same manner.

Within the degradation state DT's output processor is an HMM shown in the bottom image of Fig. 4.7, which represents the possible degradation states of a system. The transition from the *Degrading* state to the *Healthy* state may be removed in some cases where degradation is irreversible. In the same manner as the IHS DT, the degradation state DT's output processor uses the Viterbi algorithm to compute the most likely degradation state of the system and the likelihood of this state given a history of modeling responses. In contrast to the IHS DT, when the probability of the *Degrading* state is greater than 0, the state (s_p) and state probability (p_P) will be the *Degrading* state and its associated state probability. These quantities are the outputs of the degradation state DT. The same data storage and response history considerations discussed in Subsection 4.2.2 also apply here.

4.2.3.2 RUL Digital Twin

The RUL DT is used to make predictions about when a system fault may occur. RUL estimates from this class of DT are based on an assumption that the system is *Degrading*, so these predictions should be considered alongside the degradation state probability outputs from degradation state DTs when making maintenance decisions. Inputs to the RUL DT include degradation states (s_p) computed by a degradation state DT in addition to the general DT inputs described previously. The data preprocessor module may include a history of these states ($[s_p]$) and a history of features ($[f^*]$) dating back to the onset of degradation in modeling requests. Feature histories are necessary for any prediction model that considers time-series dynamics, and degradation state histories may be relevant when degradation is modeled as a multi-stage process.

Because the purpose-driven DTs discussed here are compatible with a wide range of data-driven, physics-based, and hybrid models, the exact structure of the model(s) that an RUL DT may use to compute fault predictions can be selected based on knowledge of the system and the degradation process being represented. For example, recent research has analyzed the dynamics of gear and bearing degradation [51], and it may be suitable to use a physics-based model to estimate the RUL of a bearing based on vibration data. However, data-driven models may be more appropriate for applications where the physics of degradation are not well-understood. The case study presented in Section 4.3 uses a data-driven, exponential regression model to predict when system fault will occur. Regardless of the modeling approach, the RUL DT provides the ability to estimate and update fault predictions based on real-time data from a system, differentiating it from offline methods such as the Weibull analysis [50].

Modeling responses within RUL DTs may have different structures depending on the form of the RUL prediction model and the form of the desired RUL DT output. For example, an RUL PDF may provided by the model bank in the form of a closed-form expression or a data table. In this case, the output processor may further contextualize the RUL results by computing a 95% confidence interval from this PDF. Decisions about these quantities should be made after considering the available modeling resources and the RUL estimates that are most actionable for a plant's maintenance scheduling system.

4.3 Case Study

This section presents a case study that implements the DT framework for an industrial gas pump that is subject to multiple mechanical system health specifications. The structure of the pump's DT aggregation hierarchy is detailed first, then the architecture of each DT is described.

4.3.1 DT Aggregation Hierarchy

Two dimensions of pump health that are commonly monitored in industrial applications will be analyzed in this case study and are described by the health specifications below.

Seal system specification: Prevent seal leak rate from exceeding a negligible upper limit: l^*

Bearing system specification: Maintain vibration root mean square (RMS) of all bearings below upper limit: 3 g

The DT aggregation hierarchy used to monitor these specifications is shown in Fig. 4.8. The DT connections depicted by the colored lines show multiple different use cases for DT aggregation. Red lines show the connections that combine individual bearing DTs into bearing system DTs and purple lines show the connections that combine the bearing and seal system DTs into the pump DT. These connections

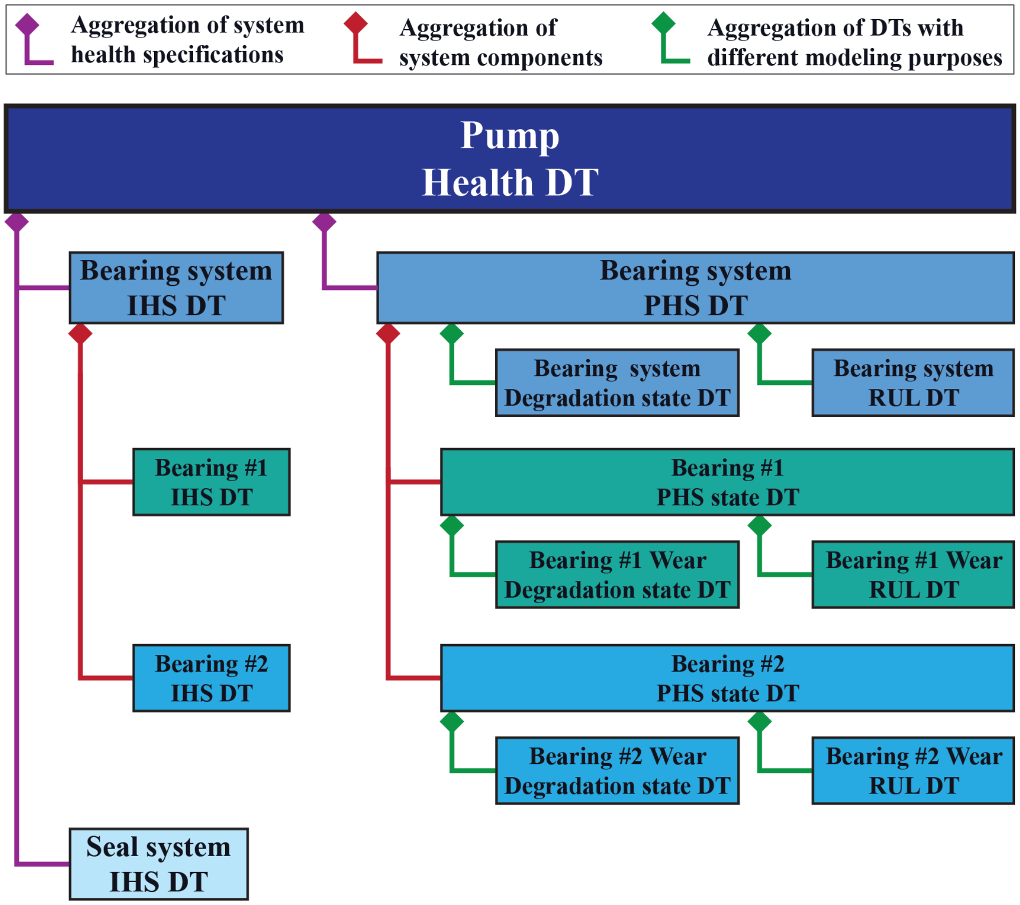

Figure 4.8: Digital twin hierarchy used to represent an industrial pump and its subsystems and components in this case study

mirror the aggregation relationships discussed in [91, 92]. This hierarchy also depicts aggregation of DTs with different modeling purposes, shown with green lines, which is a unique feature of the purpose-driven DTs used here.

The pump's seal system specification (prevent seal leak rate from exceeding a negligible upper limit: l^*) pertains to the pump's ability to contain process gas. A seal wraps around the shaft of the pump to prevent process gas from leaking outside the machine. A seal flush system maintains suitable seal temperature and pressure conditions by circulating seal flush fluid around the seal. It is not possible to measure the seal leak rate of the pump directly, but measurements from the seal flush system can be used to detect the presence of non-negligible seal leaks. Only instantaneous monitoring of seal system specification is implemented here using a seal system IHS DT. This demonstrates the framework's compatibility with applications where predictive monitoring is not possible or advisable due to inadequate sensing capabilities, modeling knowledge, or business incentives. The output of this DT gives feedback about the presence of seal leaks.

The pump's bearing system specification (maintain vibration RMS of all bearings below upper limit: 3 g) pertains to the vibration levels in the two rolling element bearings that support its rotating shaft. Bearing vibration RMS, which describes magnitude of a series of accelerometer measurements, is used to quantify this aspect of system health. While the seal system specification directly relates to a problem with system health, vibration RMS can act as an indirect indicator for a number of problems, including an imbalanced rotor, contaminated or insufficient bearing lubrication, or wear on a bearing surface, that are difficult to monitor directly. As a result, it is common for operators to monitor vibration RMS and trigger maintenance when the feature increases [34].

In this implementation, predictive monitoring of the bearing system specification is possible, so a bearing system PHS DT is used alongside a bearing system IHS DT to detect degradation and make fault predictions. The system-level DTs act as parents to two DTs that monitor a single bearing. The individual bearing DTs monitor the specification provided below for their respective bearings.

Individual bearing specification: Maintain bearing vibration RMS below upper limit: 3 g

Results from both of the bearing DTs are sent to the bearing system DT, which uses that information to make state estimates with respect to the bearing system specification.

4.3.2 Instantaneous Seal System Monitoring

4.3.2.1 Seal System IHS DT

Instantaneous monitoring of the seal system specification takes place within the seal system IHS DT. Data stream inputs to the DT consist of flow rate and pressure measurements of the pump's seal flush fluid. During seal leak faults, these measurements will deviate from set point values due to a disruption of the seal flush fluid flow by leaking process gas. DT inputs also include the flow rate and pressure set points that are considered context states for the system. These values are formatted as model parameters (p) and sent to the model bank along with the current flow rate and pressure measurements (f^*). No model selection or signal processing mappings are implemented in the data pre-processor module of the computational engine. Historical data from known $In\ Spec$ and $Out\ of\ Spec$ periods is used to train a single model that consists of two bivariate Gaussian distributions. The sample mean and covariance statistics from the labeled datasets fully define these distributions, with the exception of the mean vector of the $In\ Spec$ distribution, which acts as a modeling parameter that assumes the value of the flow rate and pressure set points. When modeling requests are received by the model bank, the PDF values of the current flow rate and pressure values are computed for each distribution ($P(f^*|s_I)$) and packaged into a modeling response. The Viterbi algorithm, as previously described, is carried out within the output processor to derive the most likely IHS of the system (s_I) and the probability of this state (p_I), which make up the IHS estimate that is the output of the DT.

4.3.2.2 Seal System Monitoring Results

Seal flush data from a pump operating in a light hydrocarbon manufacturing plant operated by The Dow Chemical Company [1] is used to demonstrate fault monitoring

[1]The Dow Chemical Company, Midland, Michigan

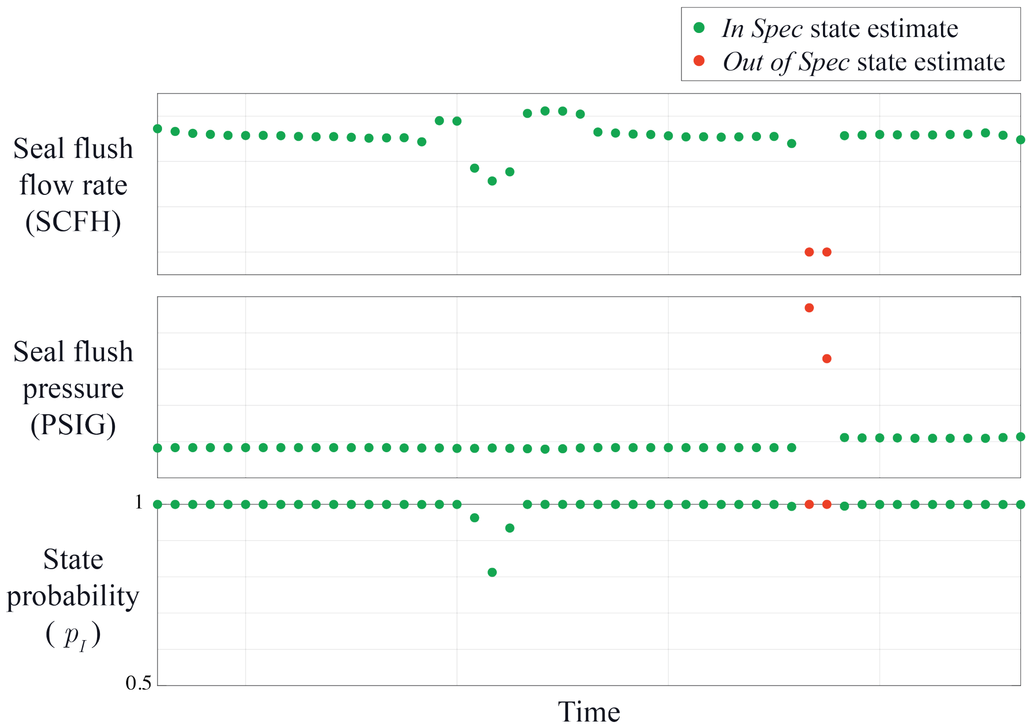

Figure 4.9: Snapshot of machine sensor measurements and seal system IHS DT outputs

with respect to this specification. Fig. 4.9 shows a snapshot of the results when these data are input to the seal system IHS DT. The top two plots show the values of the input data streams provided to the DT, color-coded by the IHS estimate determined by the DT. The bottom plot shows the state probabilities at each point in time. This timeline shows the DT's ability to detect seal leak faults, as evidenced by the *Out of Spec* state estimates that occur in the latter half of the snapshot, which align with maintenance reports citing a seal leak. The state probabilities (p_I) also provide added context to the *In Spec* state estimates earlier in the timeline. A collection of three data points are given relatively low state probabilities due to a minor deviation in seal flush flow rate measurements. This information would be useful for system operators as an early warning for seal leak faults that may occur in the future.

4.3.3 Instantaneous Bearing System Monitoring

4.3.3.1 Bearing IHS DTs

Because bearing vibration RMS can be calculated directly from accelerometer data streams, the bearing IHS DTs implement a simple limits-based approach to monitoring this specification. The bearing #1 and bearing #2 IHS DTs each receive batches of high-frequency accelerometer measurements as data stream inputs from their respective bearings and compute the RMS of these measurements within the data pre-processor module. This feature is then sent to the model bank where a univariate threshold model compares the current feature value to the threshold value (3 g), to determine the most likely instantaneous health state ($\hat{s}_P$). Sensor noise is assumed to be negligible in this implementation, so this model is regarded is perfectly accurate by the model manager. Therefore, modeling responses will have the form

$$\begin{aligned} P(\hat{s}_P|s_P) = 1 \quad \text{if} \quad \hat{s}_P = s_P \\ P(\hat{s}_P|s_P) = 0 \quad \text{if} \quad \hat{s}_P \neq s_P \end{aligned}$$

4.3.3.2 Bearing System IHS DT

The individual bearing IHS estimates make up the inputs to the bearing system IHS DT and no further pre-processing takes place in this DT. A model within the model bank treats the individual bearing states as logical values, with the *In Spec* state as "False" and the *Out of Spec* state as "True," and carries out a logical "OR" operation to compute a system-level state. That model is regarded as perfectly accurate by the model manager so the system state and a state probability of $s_P = 1$ is the output of the bearing system IHS DT.

4.3.4 Predictive Bearing System Monitoring

The DT hierarchy enabling predictive monitoring has a similar structure to the instantaneous DT hierarchy, with a bearing system PHS DT acting as a parent to individual bearing PHS DTs. Each bearing PHS DT uses information from degradation state and RUL DTs to prepare bearing-specific PHS estimates. During bearing degradation, vibration RMS measurements from individual bearings are modeled as an exponential function of time, as shown in Eq. 4.5,

$$v(t) = \theta e^{\beta t + \epsilon(t_k) - (\sigma^2 t/2)} \tag{4.5}$$

where $v(t)$ represents vibration RMS measurements at time t, θ and β are stochastic model parameters, and $\epsilon(t_k)$ is an error term following Brownian motion. This model and the forthcoming RUL calculations are derived in [42].

The models described below are trained using data from a series of historical run-to-failure tests. These datasets consist of vibration RMS measurements collected as an originally *Healthy* system is operated until fault occurs. The historical datasets end when the bearing being tested violates the individual bearings specification given above. Once trained with these data, the models are implemented within the case study DTs to monitor bearing system health during subsequent tests.

The bearing system specification listed above may be violated if vibration levels in either bearing exceed the upper threshold. The bearing system predictive monitoring DTs then manage the modeling resources necessary to derive system-level PHS estimates from the outputs of its components. In this case study, degradation in the two bearings are assumed to be independent processes. This is reasonable for systems with sufficient damping to prevent the vibration of one bearing from impacting others. If this is not the case, or if bearing degradation shifts the system's loading forces in a way that could accelerate degradation, the models in this case study should be modified to account for this.

4.3.4.1 Bearing Degradation State DTs

Degradation detection DTs for each bearing receive batches of vibration measurements from accelerometers as inputs and the data pre-processor module computes the RMS of each batch. Still in the data pre-processor, an exponential model is fit to the current RMS value and a short history of previous RMS values. The model's β parameter describes the rate of increase in RMS values. The degradation detection method used here assumes that this parameter will remain close to 0 while a bearing is *Healthy* and increase once degradation begins. Therefore, the value of the β parameter is extracted as the feature f^* that is sent to the model bank. The θ parameter in the exponential model describes the baseline magnitude of *Healthy* RMS measurements, which is known to vary between run-to-failure tests and is not used for degradation detection

The degradation detection model used here has a similar structure to the seal system's fault detection model, with normal probability distributions describing the values of f^* in *Healthy* and *Degrading* states. The parameters of the univariate Gaussian distributions are estimated from historical data. The modeling result is a set of two observation likelihood values ($P(f^*|s_I)$) that is sent to the output processor. The Viterbi algorithm is then used to determine the most likely state probability of the bearing at the current time. This case study does not use a lower threshold to halt PHS estimation, so whenever the state probability associated with the *Degrading* state is greater than zero, the state $s_P = Degrading$ and that state probability are the outputs of the degradation state DT. The state $s_P = Healthy$ is the DT output only when the *Degrading* state probability is 0.

4.3.4.2 Bearing RUL DTs

The bearing RUL DTs receive states s_p and the accelerometer data streams as inputs. When bearing degradation is detected ($s_p = Degrading$), vibration RMS values are computed from accelerometer measurements in the data pre-processor module and all RMS values from the first *Degrading* classification up to the current time are sent to the model bank as a set of features ($[f^*]$). The data-driven exponential degradation model developed by [42] and shown in Eq. 4.5 is implemented in the model bank. With this approach, historical data is used to define the prior distributions for the stochastic model parameters. Bayesian updates of these distributions are then made in real time based on the DT input features $[f^*]$ observed during the current degradation instance. After every new observation, an updated RUL CDF $P(F \leq t)$ (where F is a random variable denoting the time until bearing fault occurs) is computed. This CDF, formatted as a data table, is sent to the output processor module, which passes the information along as the output of the bearing RUL DT. Finally, the PHS DT for each bearing aggregates the outputs of the degradation state and RUL DTs into a PHS estimate that is sent to the bearing system PHS DT.

4.3.4.3 Bearing System Degradation State DT

The bearing system degradation state DT receives the PHS estimates from the bearing PHS DTs and sends the *Degrading* state probabilities for each bearing (p_P^{B1} and p_P^{B2}) to the model bank. A degradation detection model implements a logical "OR" operation, as shown below, to determine the probability that any of the bearings in the system are degrading (p_P^{BS}).

$$p_P^{BS} = p_P^{B1} + p_P^{B2} - p_P^{B1} * p_P^{B2} \tag{4.6}$$

This result is sent to the output processor, which outputs the state ($s_P^{BS} = Degrading$ and p_P^{BS} when $p_P^{BS} > 0$, or $s_P^{BS} = Healthy$ otherwise.

4.3.4.4 Bearing System RUL DT

The bearing system RUL DT uses a similar principle to derive bearing system RUL predictions. This DT receives the individual bearing fault CDFs as inputs and sends this information to the model bank. The bearing CDFs represent $P(F_i < t)$, where F_i is the time remaining before fault in bearing i. A model in the model bank uses Eq. 4.7 to compute the bearing system CDF ($P(F_{BS} < t)$) that is sent to the output processor.

$$P(F_{BS} \leq t) = P(F_1 \leq t) + P(F_2 \leq t) - P(F_1 \leq t) * P(F_2 \leq t) \tag{4.7}$$

The bearing system RUL DT seeks to identify an interval $[RUL_L, RUL_U]$ that contains the true fault time of the bearing system with 95% confidence. The output processor implements a non-linear equation solver to determine the values of RUL_L and RUL_U that solve Eq. 4.8.

$$P(RUL_L \leq F_{BS} \leq RUL_U) = P(F_{BS} \leq RUL_U) - P(F_{BS} \leq RUL_L) = 0.95 \tag{4.8}$$

This interval is the output of the bearing system RUL DT. The bearing system PHS DT completes the predictive monitoring process by aggregating the outputs of the bearing system degradation state and RUL DTs into a complete PHS estimate that is sent to the pump health DT.

4.3.4.5 Bearing System Monitoring Results

Bearing vibration data from the PRONOSTIA experimental rotating testbed, documented in [126], was used to demonstrate predictive monitoring with respect to

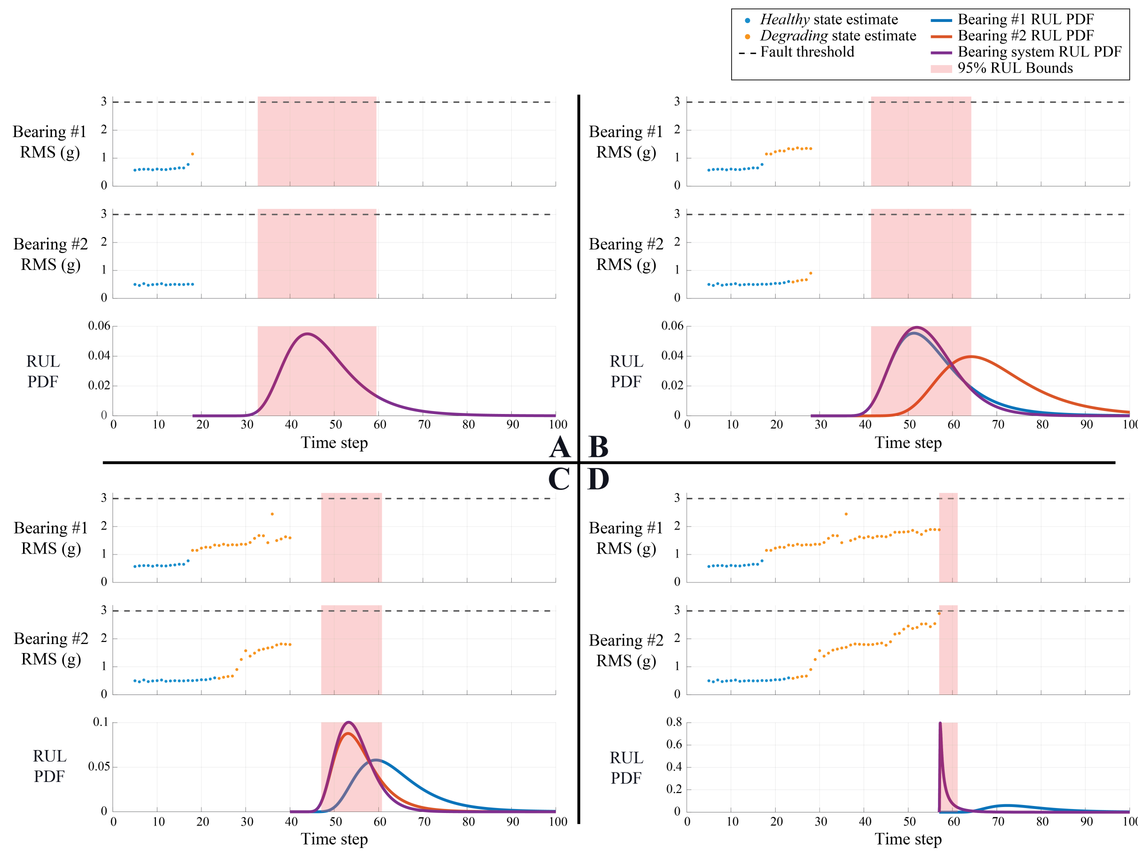

Figure 4.10: Snapshots of bearing vibration RMS measurements and DT outputs during case study: A) After degradation onset in one bearing, B) After degradation onset in both bearings, C) Mid-way through bearing degradation, D) Immediately before fault in bearing #2

this specification. Fig. 4.10 shows a snapshots of the outputs from the bearing and bearing system DTs as the bearing system nears a fault. The top two plots in each snapshot show the history of bearing RMS measurements up to the current point in time, color coded by the degradation state output of the bearing's PHS DT. The bottom plot shows the estimated RUL PDF for the individual bearings, computed by taking the derivative of the bearing CDFs. These are plotted alongside the bearing system's 95% RUL interval and the bearing system RUL PDF. Snapshot A shows the degradation states and RUL predictions when degradation has begun in both bearings Snapshots C and D show the system RUL interval narrow as degradation progresses, up until the point where bearing system fault is imminent.

4.3.5 Discussion

While the component models described above have been deployed individually in prior work, this case study demonstrates how the DT classes and aggregation relationships introduced in Section 4.1 and Section 4.2 can synthesize modeling results into standardized DT outputs that describe system-level machine health with respect to quantitative specifications. The models used here are merely examples of methods that could be used to derive DT outputs. In industrial applications, system experts may implement many forms of physics-based, data-driven, or hybrid models within each DT's model bank. The case study also shows how the proposed DT framework enables online health monitoring informed by several types of machine signals. This approach often provides more accurate health state estimates and results in less machine downtime compared to offline analyses of historical data that are used to generate fixed maintenance schedules.

The aggregation hierarchy in Fig. 4.8 demonstrates how the PHM DT classes can be re-used across components and subsystems. In the case study, instances of the IHS DT class represent both the pump's bearing system and seal system, with different models used to derive the same DT output metrics. The individual bearing PHS DTs also provide an example of how DT class instances can re-use models to represent similar components. With this approach, component and subsystem DTs can be considered modular entities and aggregation hierarchies can easily be extended to represent complex mechanical systems.

4.4 Conclusions

This chapter presents a DT framework for PHM that enables instantaneous and predictive health monitoring of complex manufacturing equipment. For manufacturing plants that feature many different machines, a standardized, scalable framework like the one presented here is critical for deploying and maintaining PHM solutions. The DTs in this framework provide templates for organizing health modeling resources and may aggregated into modular DT hierarchies to represent complex systems. Additionally, because IHS and PHS estimates are made with respect to quantitative health specifications, the implications of these quantities are easily interpretable for operators and maintenance personnel. A case study with an industrial pump made up of multiple subsystems and components demonstrates these capabilities.

CHAPTER 5

Adaptive, Multi-stage Fault Prediction Framework

This chapter presents the work published in [44]. The contributions of Chapter 3 and Chapter 4 provide methods to develop and deploy state-based prognostics and health management (PHM) solutions, but, for many applications, PHM solution developers may not have a clear understanding of how degradation will progress or how it will be reflected in machine signals. As discussed in Section 2.4, existing state-based prediction methods are limited to tracking a fixed, pre-defined sequence of degradation stages. These methods also rely on large quantities of historical training data to profile degrading system behavior. High quality, labeled datasets are often not available from manufacturing equipment, but manufacturing operations typically have operators that are extremely knowledgeable about a plant's machines and can describe unhealthy machine behavior qualitatively.

The contribution of this chapter is a state-based framework for modeling multi-stage degradation processes in industrial equipment with an accompanying, adaptive methodology for state estimation and fault prediction. The framework uses concepts from object-oriented programming to represent known system health stages with feature trajectory classes and defines an instantiation mechanism to describe the subset of stages and feature trajectories observed in individual degradation episodes. These feature trajectory classes can be defined by system experts without extensive historical training datasets. The adaptive methodology uses a particle filter to track multiple possible health stage sequences simultaneously and uses Markov chain Monte Carlo sampling to compare recent sensor measurements with known feature trajectories to detect anomalous degradation behavior. Additionally, the methodology provides a mechanism to extend a system's multi-stage health model to accommodate new degradation stages, which may describe unforeseen fault modes.

The rest of the chapter is organized as follows. Section 5.1 presents a modeling framework that allows multi-stage degradation processes to be defined by subject matter experts (SMEs) based on general feature trajectories. Section 5.2 presents an adaptive methodology for stage estimation and fault prediction based on real-time sensor measurements collected from a system. Section 5.3 discusses a case study with the proposed framework. Conclusions of this chapter are given in Section 5.4.

5.1 Modeling Architecture

This section defines the modeling resources that the proposed framework uses to monitor degradation and predict system faults. A summary of the nomenclature that will be used to describe these resources is provided in Table 5.1.

5.1.1 Feature Trajectories

The expected behavior of one or more *signal features* (x) over time is modeled using state-based feature trajectories. A *feature trajectory* is defined as a discrete-time state equation that describes feature values at the next time step (x_{k+1}) as a function of feature values at the current time step (x_k) and a set of *trajectory parameters* (P), as expressed in Eq. 5.1.

$$x_{k+1} = f(x_k, P) \tag{5.1}$$

Table 5.1: Nomenclature for the proposed adaptive modeling framework

Term	**Description**
Signal feature	Property extracted from one or more raw machine signals
Feature trajectory	Discrete-time state equation that describes the expected evolution of one or more feature values over time
Feature trajectory class	Set of specifications for defining a feature trajectory
Degradation episode	Period of system operation that ends in a fault
Health stage	State of system health that is characterized by a feature trajectory
Trending feature trajectory class	Feature trajectory class that predicts drift in feature values over time
Non-trending feature trajectory class	Feature trajectory class that does not predict drift in feature values over time
Global system health automaton	Automaton with all possible system health stages
Local system health automaton	Automaton with system health stages that describe an ongoing degradation episode

As in object-oriented programming, individual feature trajectories are considered to be instantiations of *feature trajectory classes* that have been specified by machine experts prior to online monitoring. These classes define the signal features that the trajectory describes, the structure of the state equation (f), and designates each parameter in P as either fixed or variable. Fixed parameters have a constant value that remains the same throughout a degradation episode, while the values of variable parameters are continually updated during degradation based on recent feature observations. All feature trajectory classes should include a method to instantiate trajectories by estimating the value of all parameters based on a set of feature observations. Classes should also include a method to update the variable parameters of existing trajectories based on new feature observations. Finally, class definitions may include parameter constraints, which define the range of values that variable parameters can take.

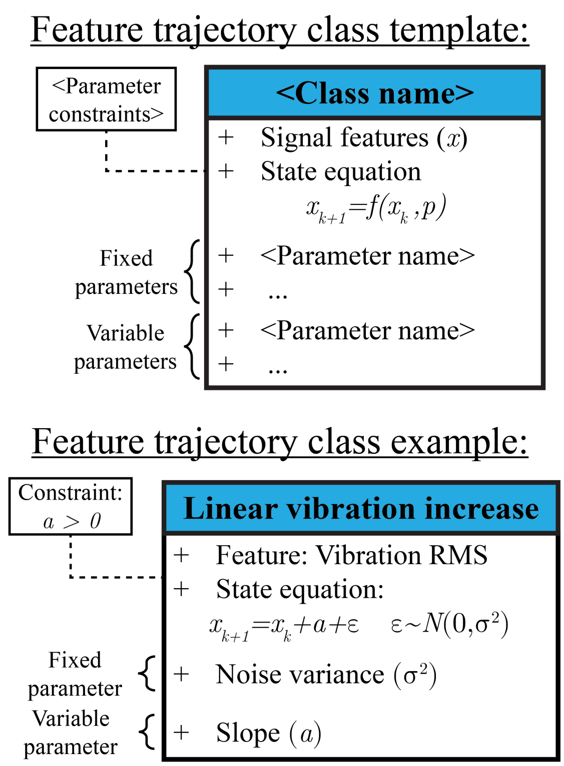

Figure 5.1: Feature trajectory class template and example

Fig. 5.1 depicts the elements of a feature trajectory class as well as an example class describing a linear increase in a common vibration feature: the root mean square (RMS) of raw accelerometer measurements. The state equation for this example ($x_{k+1} = x_k + a + \varepsilon$) describes a linear increase in x, that is impacted by a Gaussian random noise variable ($\varepsilon \sim \mathcal{N}(0, \sigma^2)$). The set of parameters for this class can then be expressed as $P = [a, \sigma^2]$. The slope parameter (a) is designated as variable, so that it can be updated as new feature observations are made. The noise variance parameter (σ^2) is designated as fixed because deviations from a linear trend are known to be relatively consistent between degradation episodes. This mirrors the assumption of fixed error distributions made by many existing trend-based modeling methods [43, 95]. A constraint is placed on a to ensure that this parameter does not fall below 0.

Feature trajectory classes that predict feature drift, such as the linear and exponential trajectories analyzed in [34, 41, 42], will be referred to as *trending feature trajectory classes*. Feature trajectory classes that do not predict feature drift in any direction will be referred to as *non-trending feature trajectory classes*. These classes may predict that feature values remain near a fixed set point value or be used to simulate a Wiener process, for example. Non-trending behavior is often useful to describe healthy system behavior, or offsets in feature values that precede trending behavior during degradation.

5.1.2 System Health Automata

The proposed framework uses global and local system health automata to represent system health stages. Both types of automata include a set of stages (S), a set of events (E), and a transition function ($g : S \times E \to S$). Prior to online monitoring, SMEs must define a *global system health automaton* that includes all possible system health stages that a system may experience during a degradation episode. SMEs are free to define stages based on knowledge of physical degradation processes or knowledge they possess from observing or analyzing historical faults in a system. This framework requires that all global automata have a $Healthy$ stage, which acts as a system's initial stage during online monitoring, and a $Out\ of\ Spec$ stage, which may not have any outgoing transitions. Each stage in a global automata, except the $Out\ of\ Spec$ stage, should be associated with a feature trajectory class that describes the general behavior of one or more signal features in that stage. Additionally, all stages with a transition to the $Out\ of\ Spec$ stage should be associated with a trending feature trajectory class, as they will be used to make fault predictions.

The stages of system health automata are designed to be mutually exclusive with respect to one another. In applications where multiple, simultaneous degradation processes can impact a machine, experts may include a health stage in a system's global health automaton to capture this behavior. Feature trajectory classes with multidimensional state equations are useful here for describing degradation trends in multiple machine signals. When simultaneous degradation processes can be assumed to be independent of one another, potentially because they impact different machine components, it may be appropriate to create multiple health automata to monitor these processes separately. This avoids the stage explosion problem that can result from creating new health stages to capture every possible case of simultaneous degradation.

A *local system health automaton* includes the system health stages that describe an ongoing degradation episode based on recent feature observations. The structures of local automata are adapted in real-time according to the methodology described in Section 5.2, but the stages and transitions will always be a subset of those included in a system's global automaton. Each stage of a local automaton is associated with a feature trajectory, which is an instance of that stage's associated trajectory class in the global system health automaton. These trajectories will have values assigned to all parameters, derived from feature observations made during the ongoing degradation episode. At the beginning of a degradation episode, a system's local automaton includes only the $Healthy$ stage and additional health stages are added as degradation trends emerge, as described in Section 5.2.

Fig. 5.2 shows an example of a global automaton for a system that may experience linear and/or exponential feature increases prior to system fault. A local automaton that may be used to monitor an ongoing degradation episode is also shown. This local automaton omits the $Exponential\ increase$ stage (which can be considered inactive), potentially because the recent feature observations are clearly linear. If this trend starts to change, the local automaton can be expanded to include the $Exponential\ increase$ stage.

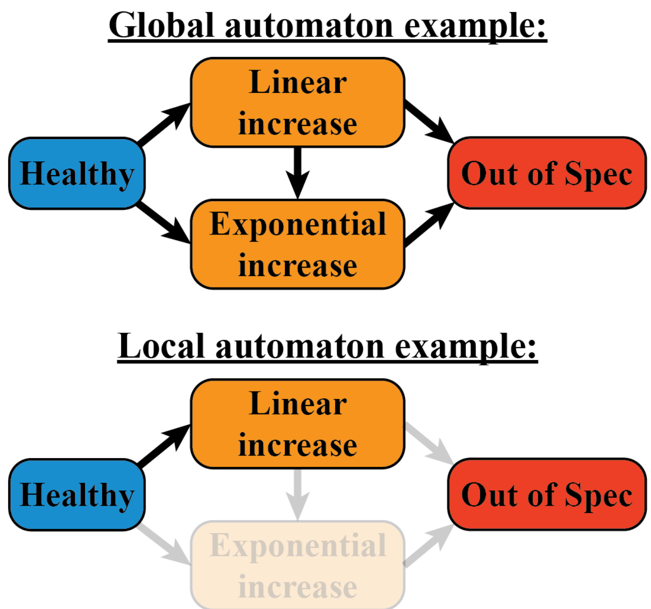

Figure 5.2: Examples of a global system health automaton and a possible local automaton

5.2 Fault Prediction Methodology

This section proposes an adaptive methodology for estimating a system's recent health stage history and predicting system faults during online operation. A case study on rolling element bearing faults is used to demonstrate aspects of the methodology. The methodology, depicted in Fig. 5.3, includes a set of four processes that are repeated whenever new feature observations are made: stage estimation, trajectory updating, fault prediction, and anomaly detection. Together these processes will be referred to as routine monitoring. Based on the results of anomaly detection, local automata expansion and global automata expansion may be triggered.

5.2.1 Stage Estimation

In applications where a system can be described with a state space model, filtering involves estimating a system's hidden state history based on recent observations. Analytical filtering solutions have been developed for certain special cases including the Kalman filter for linear Gaussian state space models [127]. Filtering for non-

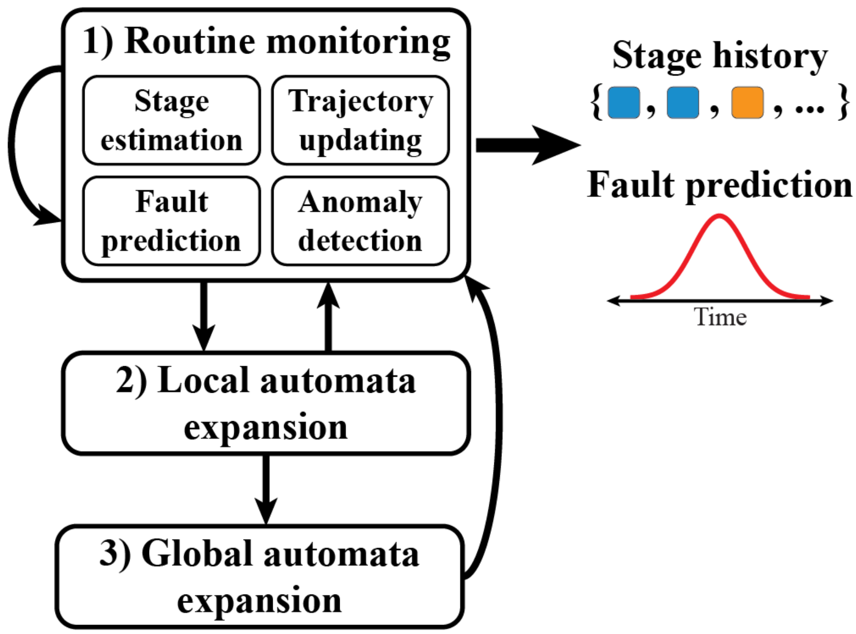

Figure 5.3: Processes included in the adaptive fault prediction methodology

linear systems, however, requires alternative approaches. The extended Kalman filter, for example, linearizes a non-linear model so that the traditional Kalman filtering equations can be applied to compute Gaussian state estimate approximations [128].

As computing power has become cheaper and more widespread, particle filtering has emerged as a popular approach for nonlinear systems that is not restricted to Gaussian approximations. Particle filters track a finite number of particles that can be analyzed as an empirical distribution to describe a system's state history [129]. The method is recursive, so state estimates can be updated based on new observations, and repeatedly assesses a system's entire state history, revising state estimates at previous time when appropriate. These attributes make particle filtering well-suited to analyze a wide variety of non-linear systems.

Stage estimation uses a particle filter to evaluate the health stage histories that may describe an ongoing degradation episode. Here, each particle represents a weighted sample of a possible health stage history ($[s_1, \ldots, s_t]$) and feature history ($[x_1, \ldots, x_t]$) at all time steps up to the current time (t), as shown in Eq. 5.2. The prediction and update steps described below are then carried out whenever new feature observations are made.

$$\text{Particle:} \left\{ \begin{matrix} [s_1, \ldots, s_t] \\ \\ [x_1, \ldots, x_t] \end{matrix} \quad , \quad \bar{w} \right\} \tag{5.2}$$

The well-known dynamics-based prediction step [129] for particle filters is adapted to enable monitoring of multi-stage degradation. First, the next health stage of a particle (s_{t+1}) is predicted based on a stage transition probability matrix (T) that corresponds to the system's local automaton. The next state of each particle (x_{t+1}) is then predicted based on the state equation of stage s_{t+1}.

Eq. 5.3 shows an example of a valid transition probability matrix for the local automaton in Fig. 5.2, where *Healthy* is the first stage and *Linear increase* is the second stage.

$$T = \begin{bmatrix} p_{11} & p_{12} \\ \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} 0.9 & 0.1 \\ \\ 0 & 1 \end{bmatrix} \tag{5.3}$$

The structure of a system's local automaton will change throughout a degradation episode, and the size of this transition matrix must change accordingly. A simple method for maintaining a valid transition probability matrix is to define a same-stage transition probability for each stage. The probabilities of all outbound transitions in the local automaton can then be assigned equal values such that all rows of the matrix sum to 1. SMEs can specify a transition probability matrix that corresponds with the global automaton, and the rows of this matrix can be normalized whenever stages are added or removed from the local automaton.

The standard updating method for particle filters [129] is implemented here to adjust the weights of each particle based on a new feature observation (y_{t+1}). A dynamics-based prediction step is used, so a particle's pre-normalized, updated weight (w_{t+1}) can be determined using Eq. 5.4, where $\bar{w}_t$ is the normalized particle weight from the previous time step.

$$w_{t+1} = P(y_{t+1}|x_{t+1})\bar{w}_t \tag{5.4}$$

A Gaussian observation model is assumed, so $P(y_{t+1}|x_{t+1}) = \mathcal{N}(y_{t+1}|x_{t+1}, \Gamma)$. The observation noise parameter (Γ) can be tuned to change the responsiveness of the filter to new observations. When this process has been completed for each particle, normalized particle weights ($\bar{w}_{t+1}$) are then computed by dividing each w_{t+1} by the sum of all pre-normalized particle weights.

In particle filter implementations, there is a tendency for weights to become concentrated in a small number of particles, a problem known as particle degeneracy. To combat this, particles are periodically re-sampled based on their weights. After every update, an effective particle size (N_{eff}) is computed according to

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}(\bar{w}^i)^2} \tag{5.5}$$

where N is the number of particles being tracked. Particles are resampled when this drops below a critical threshold (N_{crit}) [129].

As shown in Fig. 5.3, one of the outputs of this methodology is a distribution ($P(s^*, t)$) describing the probability that the system was in stage s^* at time t. This distribution can be evaluated by determining the proportion of particles with a stage history that satisfies $s_t = s^*$.

5.2.2 Trajectory Updating

The trajectory updating process adapts the variable parameters of all active feature trajectories after every new feature observation. An exponentially-weighted moving average (EWMA) filter [130] is implemented here to make these updates. When new feature values are observed, the parameter estimation method for each active feature trajectory is used to compute a new parameter estimate based on the current feature observation (y_t) and the most recent previous feature observation (y_{t-1}). For example, the linear trajectory class shown in Fig. 5.1 has a single variable parameter (a) and may compute parameter estimates by finding the difference between sequential feature observations:

$$a^* = y_t - y_{t-1} \tag{5.6}$$

The EWMA equation, shown in Eq. 5.7, is then used to update the value of this variable parameter using this new estimate and the previous parameter value (a_{t-1}).

$$a_t = \beta a_{t-1} + (1 - \beta) a^* \tag{5.7}$$

The rate at which trajectory parameters are updated is dictated by a decay rate variable (β), which may be set to any value in $[0, 1]$. In cases where a trajectory's updated parameters violate a class constraint, the particles currently in this stage are transitioned to other health stages according to the local automaton's transition probability matrix.

5.2.3 Fault Prediction

Depending on the extent of degradation, the updated stage estimates and feature trajectories may be used to predict system fault. This decision is dictated by the proportion of particles that are in a fault-adjacent stage (a stage with a transition to the *Out of Spec* stage in a system's global automaton). SMEs may set a lower proportion threshold if it is useful to generate fault predictions while a system's true health stage is still uncertain, or a higher threshold if fault predictions are only necessary later in a degradation episode. When this threshold is met, fault prediction is implemented by projecting fault-adjacent particles forward in time. As discussed in Section 3.1, this framework assumes that system faults are defined by quantitative limits on one or more features. Future particle states ($[x_{t+1}, x_{t+2}, \ldots]$) are repeatedly calculated using each particle's state update equation until the fault threshold is met. The time at which this occurs represents a sample of the system fault time. Together, the fault times of all fault-adjacent particles can be aggregated into an empirical distribution that probabilistically estimates the system fault time. System fault predictions can be derived from this empirical distribution using statistics like an expected value or a confidence interval.

5.2.4 Anomaly Detection

The final routine monitoring process, anomaly detection, detects significant disparities between the most recent feature observation (y_t) and particles being used to monitor degradation. When this occurs, stages from a system's global automaton may be added to the current local automaton to better model the ongoing degradation episode, as described in Subsection 5.2.5. Anomaly detection is implemented here by considering the probability of observing y_t given the current particle states: $X_t = [x_t^1 \ldots x_t^N]$, where N is the number of particles being tracked.

A Gaussian observation model is again assumed for each particle, such that $P(y_t|x_t) = \mathcal{N}(y_t|x_t, \Gamma)$. A Gaussian mixture model (GMM) [131] can then be used to describe the observation probability distribution with multiple, weighted particles, as shown in Eq. 5.8.

$$P(y_t|X_t) = \sum_{i=1}^{N} \bar{w}_i \, \mathcal{N}(y_t|x_t^i, \Gamma) \tag{5.8}$$

The value of $P(y_t|X_t)$ can easily be computed for new observations, but it cannot be used for anomaly detection without some context about the shape of the underlying distribution. This GMM will be multi-modal and potentially multivariate, so basic statistical anomaly detection methods like the Z-test cannot be applied. Instead, a high density region (HDR) is used to describe the observation probability distribution

A $(100 - \alpha)\%$ high density region for random variable X with density function $f(x)$ is the subset $R(f_\alpha)$ of the sample space of X such that

$$R(f_\alpha) = \{x : f(x) \geq f_\alpha\} \tag{5.9}$$

where f_α is the largest constant such that $Pr(X \in R(f_\alpha)) \geq 1 - \alpha$. The value of f_α that defines a $(100 - \alpha)\%$ HDR is estimated by generating independent samples from $f(x)$. Because f_α is defined such that $Pr(X \in R(f_\alpha)) \geq 1 - \alpha$, f_α will be the α quantile of the samples [132].

When new observations are made, a Markov chain Monte Carlo sampling method is used to generate sample from $P(y_t|X_t)$ and estimate f_α for the observation probability distribution. If $P(y_t|X_t) \geq f_\alpha$, the observation lies within the $(100-\alpha)\%$ HDR and is classified as non-anomalous. If $P(y_t|X_t) < f_\alpha$, the observation lies outside the $(100 - \alpha)\%$ HDR and is classified as anomalous. The value of α can be tuned by SMEs to control the sensitivity of anomaly detection. When α is close to 0, only significant deviations between particles and observations will be considered anomalous. So, larger values may be appropriate when detecting early degradation is critical to a predictive maintenance strategy, even at the risk of false positives. A continuous anomaly score can also be generated for each observation (y_t) by computing the percentile of y_t with respect to the samples from $P(y_t|X_t)$. This score will be in $[0, 100]$, with larger values associated with more anomalous observations.

5.2.5 Local Automata Expansion

Local automata expansion provides a mechanism to add inactive stages from a system's global automaton to a local automaton that describes the ongoing degradation episode. In general, this process is triggered by the results of anomaly detection, but SMEs can define the exact criteria for expansion based on the application. For example, automata expansion can be triggered by three consecutive anomalies, four anomalies out of five consecutive observations, or any conditions adapted from statistical process control (such as the Western Electric rules [133]). In any case, these criteria should classify a set of recent feature observations as anomalous.

The expansion process involves adding a set of stages and transitions from the global system health automaton to the local system health automaton. Expansion candidates are first identified based on the particle stages prior to the first anomalous feature observation ($S_{current}$). Using the global automaton's transition function ($g : S \times E \to S$), a set of inactive transitions ($\hat{E}_{new}$) originating from the stages in $S_{current}$ are compiled. The stages that these transitions lead to ($\hat{S}_{new}$) are then identified, and instances of their associated feature trajectory classes are trained using the recent anomalous observations. The set of stages (S_{new}) with valid feature trajectories (trajectories with variable parameters that do not conflict with class constraints) are then added to the current local automaton along with the transitions that lead to those stages (E_{new}). Finally, routine monitoring is repeated for the recent anomalous data with the newly expanded local automaton.

5.2.6 Global Automata Expansion

Persistent anomalies after local automata expansion are an indication that a system's global automaton is not able to describe the current machine behavior. This may be the result of an unexpected disturbance or a novel degradation mode. In this case, global automata expansion should be carried out under the guidance of system experts to add additional stages and feature trajectory classes to a system's global automaton.

Global automata expansion may be implemented in a variety of ways based on the system being monitored. For systems with a small number of sensors or when system experts are very knowledgeable about a system, expansion can be done manually by identifying the feature behavior that is causing the anomaly and incorporating this behavior into the global automaton either by modifying an existing feature trajectory class or adding a new health stage to the automaton. For complex systems with many machine signals, automated expansion processes supervised by system experts may be appropriate. For example, principle component analysis can be used to identify the machine signal features that vary significantly during a recent anomalous period. Candidate health stages could be created by defining new trajectory classes for these features with simple, pre-defined state equations. System experts would then select between these candidate stages based on their knowledge of possible degradation modes and goodness of fit metrics such as the coefficient of determination (R^2).

5.3 Case Study

This section presents a case study that implements the proposed modeling framework and fault prediction methodology to monitor rolling element bearings.

5.3.1 Modeling Setup

The dataset analyzed here was collected and published by the FEMTO-ST Institute [126] and includes vibration data from several tests that are each considered to be a degradation episode. Seven tests carried out with identical bearings and operating parameters (1800 rpm and 4000 N radial load) are analyzed here. Though these parameters are held constant, the degradation trends seen in vibration measurements show significant variations between tests, as discussed in Subsection 5.3.2.

The case study seeks to predict faults in rolling element bearings using a vibration feature known as root mean square, which describes the magnitude of vibration measurements. This feature is known to increase as a bearing degrades and is commonly used to detect bearing degradation [34]. Noise is filtered from the raw RMS values using an EWMA filter with a 0.75 decay rate. The documentation for the

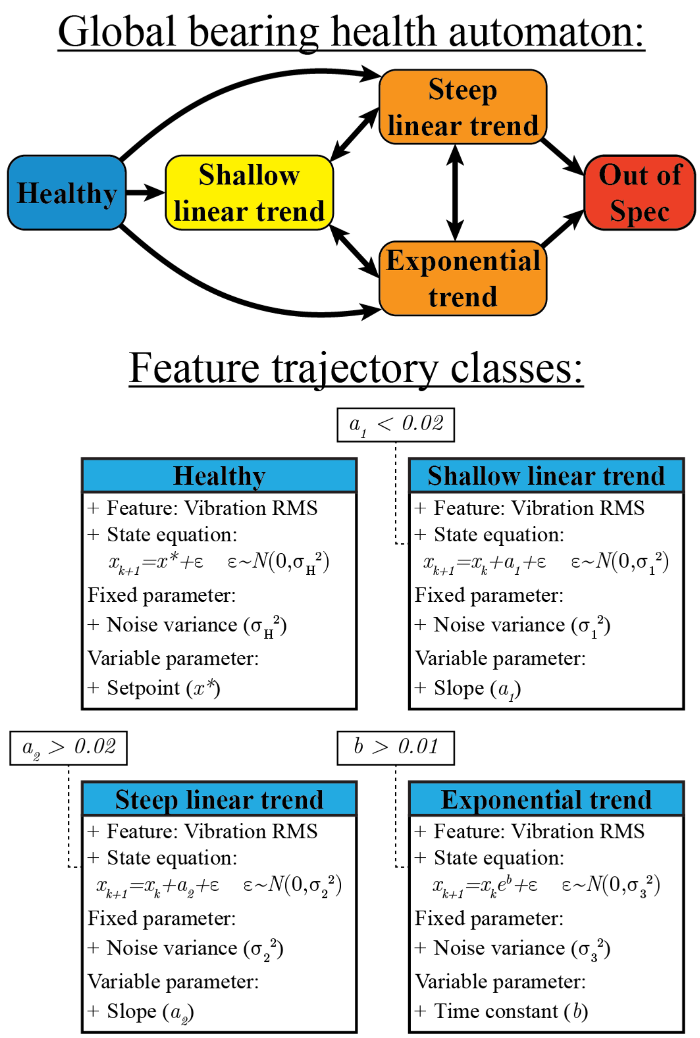

Figure 5.4: Global automaton and feature trajectory classes used to model rolling element bearing degradation in this case study

FEMTO-ST Institute dataset does not specify the criterion that was used to stop each test, so this analysis retroactively calculates a fault threshold for each test by averaging the final five vibration RMS measurements. While this method of defining a fault threshold is not practical for real-time monitoring, it was necessitated by the structure of the case study dataset. In practice, fault thresholds can be set based on pre-existing alarm thresholds on machine signals, or an analysis of historical feature observations before emergency shutdowns.

The proposed framework and methodology are used to predict bearing faults in each of the seven degradation episodes. The global automaton shown in Fig. 5.4 represents the set of possible health stages. *Steep linear trend* and *Exponential trend* stages are included to describe late-stage degradation. The trajectory classes for these stages both have lower limits on their variable parameters to ensure that they are only activated when significant RMS increases are observed. A *Shallow linear trend* stage is also included to describe slow increases in RMS that often occur early in the degradation process. The trajectory class for this stage has an upper limit on its slope parameter to force a transition to the later stages when RMS starts to increase more rapidly. Notably, there is no transition between the *Shallow linear trend* stage and the $Out\ of\ Spec$ stage, so fault predictions will not be computed when stage estimates indicate that the system is in this stage.

During most of each degradation episode, RMS observations show very little variation, so this analysis focuses on the final 30% of each episode. The mean of the RMS observations from the first 70% is used to define the $Healthy$ stage's setpoint parameter (x^*). The variance of the Gaussian observation model (Γ) is set to be the variance of the $Healthy$ RMS observations, and the noise variance parameters ($\sigma_H, \sigma_1, \sigma_2, \sigma_3$) are all defined to be twice this value. During routine monitoring, a critical particle size threshold (N_{crit}) of 200 is used to trigger resampling, and fault predictions are computed when 50% of particles are in a fault-adjacent stage. Local automata expansion is triggered when 5 consecutive RMS observations are classified as anomalous, where anomaly detection is implemented with $\alpha = 25$. During all tests, local automata are defined such that all stages have a 0.9 same-stage transition probability, with the remaining 0.1 probability split equally among all other transitions.

The $\alpha - \lambda$ accuracy metric described in [134] is adapted for this case study, as shown in Eq. 5.10. This new metric (Acc) measures the similarity of predicted fault distributions with the actual system fault time.

$$Acc = \frac{\sum_{i=1}^{n} \int_{T_U}^{T_L} \pi_i(ToF_{pred})\, d\, ToF_{pred}}{n} \tag{5.10}$$

Here, $\pi_i(ToF_{pred})$ is an empirical distribution generated at time t_i describing the predicted fault time (ToF_{pred}), and n is the total number of fault predictions made throughout a degradation episode. For each fault prediction, this metric computes the mass of $\pi_i(ToF_{pred})$ that lies within a range bounded by $T_L = ToF_{act} - T^*$ and $T_U = ToF_{act} + T^*$, where ToF_{act} is the actual system fault time and T^* is an allowable margin of error. A margin of error of 20 time steps is used in this case study. The average mass that lies within this range over all fault predictions made during a degradation episode is then calculated.

The metric *Acc* rewards distributions that are narrow and centered near the actual fault time, which give machine operators precise feedback about when a fault is expected to occur. However, one limitation of the metric is that it places identical weights on all predictions made throughout a degradation episode, which may not be appropriate in all applications. Several alternative prognostic performance metrics are described in [134], including some that place increasing weights on predictions as degradation progresses.

5.3.2 Results

Fig. 5.5 shows a snapshot of the methodology's stage estimation and fault prediction results during one degradation episode. Here, RMS observations begin to increase slightly from their *Healthy* setpoint, triggering a transition to the *Shallow linear trend* stage. After some time, the RMS feature begins to increase more rapidly, which is best described by the *Steep linear trend* stage. When this occurs, the methodology begins to make fault predictions, such as the one shown in the figure, which can be

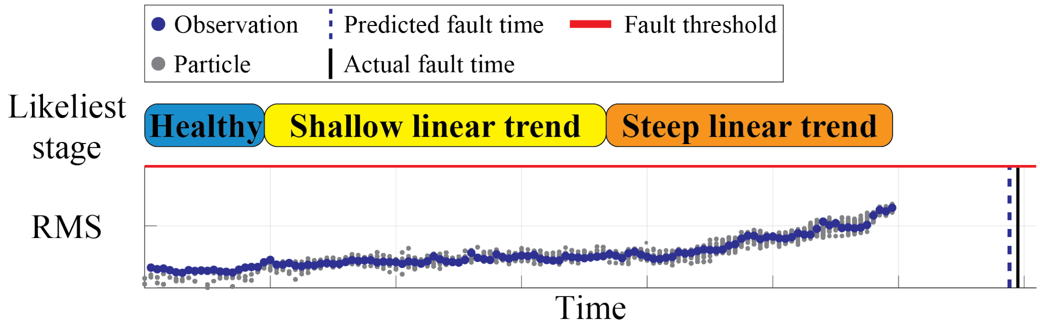

Figure 5.5: Snapshot of bearing vibration RMS measurements, stage estimation, and fault prediction results during an experimental test

compared with the bearing's actual fault time.

Table 5.2 shows the accuracy rates of the proposed multi-stage model's fault predictions for all seven degradation episodes. For comparison, prediction accuracy results for the linear and exponential degradation models proposed by [42] are also shown. These models assume the degradation follows a fixed trajectory with random parameters (a slope parameter for the linear model and multiplier and time constant parameters for the exponential model). The prior distributions of these parameters must be defined based on historical data and a Bayesian update method generates posterior distributions when new observations are collected. For each test, prior distributions are defined using the other six tests, an approach known as k-fold cross validation. Because these single-stage models do not account for the static and shallow linear trend behavior that takes place at the beginning of the degradation episodes, only the data collected after the first fault prediction made by the multi-stage model are used here. This allows us to use the same ToF_{act} value across all fault prediction methods.

5.3.3 Discussion

The comparison between prediction performance given in Table 5.2 demonstrates that the proposed multi-stage model generates the most accurate predictions in five of the seven tests. Further examination shows that the accuracies of all three predictions

Table 5.2: Comparison of accuracy rates across different fault prediction methods, Best accuracy rate for each test shown in green

Test No.	Multi-stage model *Acc*	Bayesian linear model *Acc*	Bayesian exponential model *Acc*
1	**54%**	3%	4%
2	67%	**73%**	40%
3	**23%**	1%	3%
4	**35%**	22%	5%
5	**98%**	85%	79%
6	**96%**	84%	69%
7	60%	**74%**	36%

methods are low for tests that exhibit slow, gradual degradation. This includes Tests 1, 3 and 4, which have degradation periods that are 15%, 26%, and 23% of the total test length, respectively. In contrast, the other tests have degradation periods that last an average of 2% of the total test length. Fault prediction is more difficult during gradual degradation because early predictions are far removed from the actual system fault time and because feature trajectories may change significantly over the course of degradation. However, the proposed multi-stage model's prediction accuracy is significantly higher for Tests 1, 3, and 4 than the single-stage models' accuracies because it has the ability to adapt to shifting feature trajectories during longer degradation periods.

The proposed modeling framework also provides implementation advantages that can make it more desirable for industrial applications. The Bayesian degradation models depend on multiple historical degradation episodes (six are used here) to define the prior distributions of their parameters, while the multi-stage model uses only *Healthy* data from the current degradation episode to define its setpoint and noise variance parameters. Additionally, the Bayesian models rely on the assumption that the time at which a system starts degrading is definitively known. In contrast, the multi-stage model explicitly considers static and early degradation behavior with

Table 5.3: Impact of particle filter size on average anomaly score, fault prediction accuracy, and computation time

Particle size threshold (N_{crit})	Average anomaly score	Prediction accuracy (Acc)	Computation time per time step
5	32.66	56.95%	0.57 sec
10	24.25	59.96%	0.61 sec
20	23.05	63.84%	0.62 sec
50	20.15	64.76%	0.69 sec
100	19.07	66.65%	0.80 sec
200	18.98	66.85%	1.06 sec
500	17.99	66.58%	1.73 sec
1000	18.06	66.87%	2.68 sec

the *Healthy* and *Shallow linear trend* stages and detects transitions between these stages in real-time. These characteristics, coupled with the promising prediction accuracy rates observed in this case study, make the proposed framework well-suited for modeling the health of industrial rotating equipment.

Another aspect of the methodology that this case study investigates is the impact of particle filter size on prediction accuracy. The fault prediction process relies on a particle filter to maintain a set of particles that represent samples of possible health stage histories. As discussed in Subsection 5.2.1, the particle size threshold parameter (N_{crit}) dictates the size of this filter because the stage estimation process triggers a resampling of particles when the effective particle size drops below N_{crit}.

Table 5.3 shows how varying the particle filter size affects different characteristics of the fault prediction methodology for a single degradation episode. Several particle filters with different N_{crit} values were used to analyze the data from Test 2, and the average anomaly score, the prediction accuracy rate (Acc), and the average computation time per time step were recorded. This analysis shows that increasing the size of the particle filter results in decreased average anomaly scores and increased prediction accuracy rates up to $N_{crit} = 100$. Both metrics plateau after this point, but computation time continues to increase, indicating that additional particles do not provide prediction accuracy benefits that might justify the added computation time. Results from the other tests reiterate the findings presented here and informed the particle size threshold used in this case study ($N_{crit} = 200$).

In practice, machine operators and maintenance engineers may employ a similar analysis with historical system faults to determine an appropriate particle filter size for a particular application. If historical system faults are not available, multiple particle filters with different sizes could be deployed to monitor a degradation episode simultaneously. System operators could then identify an appropriate particle filter size by identifying the size at which anomaly scores plateau. Larger particle filters should be terminated as they are not likely to generate more accurate fault predictions.

5.4 Conclusions

This chapter presents a state-based modeling framework for monitoring multi-stage degradation processes and predicting system faults. In manufacturing applications, degradation often occurs over long periods of time, with shifts in machine signal trends that reflect changes in machine dynamics. Additionally, precise characterizations of degradation behavior are often not known and extensive historical data may not be available to train a health model to learn these characterization. Pre-existing knowledge from machine experts is a valuable resource for predictive models that can act as a substitute to large training datasets. The framework presented here allows SMEs to incorporate their knowledge of system health problems into multi-stage degradation models that can be used for online degradation detection and fault prediction. The framework is applicable to a wide range of mechanical systems, as feature trajectory classes can be defined and combined to describe many kinds of systems and fault modes. A case study with time-series data from multiple rolling element bearing faults demonstrates the model development and fault prediction advantages of this approach.

CHAPTER 6

Overcoming Challenges Associated with Developing Industrial PHM Solutions

This chapter presents the work published in [45]. Existing prognostics and health management (PHM) research and the contributions discussed in previous chapters provide useful frameworks and methods for modeling the health of industrial manufacturing equipment, but many challenges related to developing industrial PHM solutions remain. Many of these are related to planning the scope of a proposed solution and designing the internal architecture and flow of information within a solution. Methods to overcome these planning and design challenges have received relatively little attention in academic literature compared to the development of general frameworks and new modeling techniques [24, 25].

The contribution of this chapter is a framework for developing industrial PHM solutions that is based on the system development life cycle (SDLC). Many of the life cycle stages have been explored in general PHM literature, but several challenges associated with the planning and design of industrial PHM solutions have not been studied in detail. The framework includes a novel methodology for planning the scope of industrial PHM solutions, which specifies several considerations that are unique to manufacturing applications and identifies the types of experts that should be consulted during planning. A methodology for designing industrial PHM solutions builds upon existing, general PHM architectures [19, 135] by proposing additional computational processes (context adaptation and output processing) that are critical in industrial applications. Additionally, two challenges that regularly afflict industrial PHM solutions: data quality and modeling degradation trends, are identified and recommendations for overcoming them are provided.

The rest of the chapter is organized as follows. Section 6.1 discusses the proposed framework for developing PHM solutions and associated challenges for industrial applications. Section 6.2 and Section 6.3 present methodologies for the planning and design stages of the development process, respectively. Section 6.4 describes how the framework was implemented to develop a PHM solution for an industrial hyper compressor, and Section 6.5 provides general insights from that case study. Conclusions of this chapter are given in Section 6.6.

6.1 Development of Industrial PHM Solutions

The SDLC describes a set of stages that a system goes through from its inception to its deployment and maintenance in the field [136]. The SDLC has been used extensively in the development of software-based applications [106, 107], including digital twins for manufacturing systems [25, 125]. Detailed descriptions of the SDLC stages and methods for progressing through them can be found in [136, 137]. While the SDLC provides a useful framework for developing a broad range of systems and applications, the decisions and considerations that should be made at each stage of the life cycle are not always clear. This section investigates two of the SDLC stages: planning and design, identifying several challenges that commonly arise when developing industrial PHM solutions.

Fig. 6.1 depicts the first several stages of the SDLC, which are usually completed offline based on subject matter expertise and input from end users. The transitions and output quantities in this diagram assume that the SDLC stages are completed

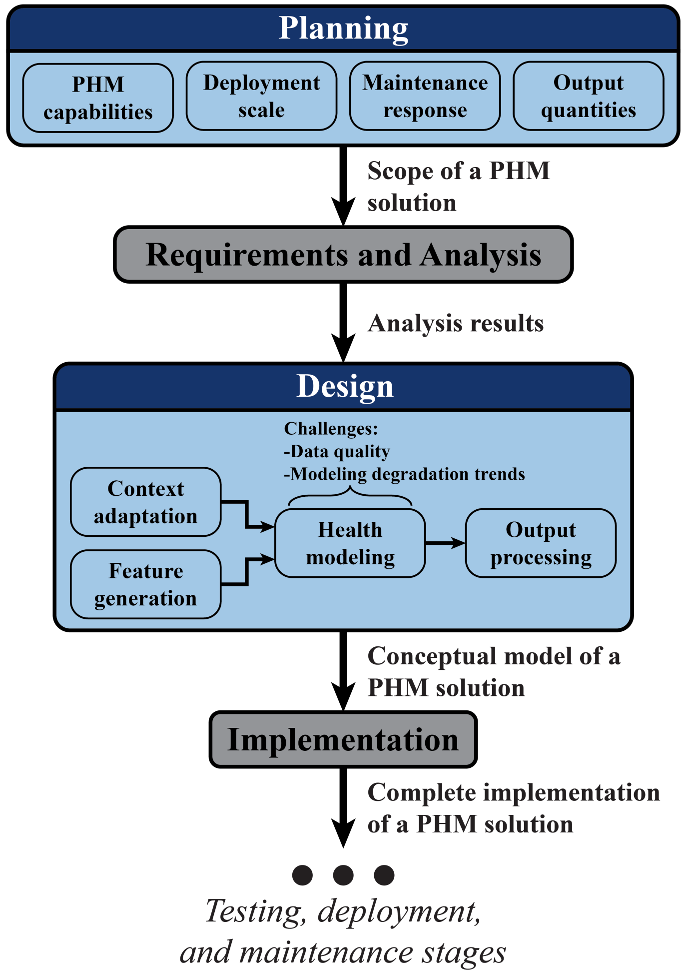

Figure 6.1: Early stages of the software design life cycle for industrial PHM solutions with depictions of the proposed planning and design methodologies

sequentially. In reality, stages will be revisited and decisions will be revised based on insights from subsequent testing, deployment, and maintenance stages. A number of different methodologies for transitioning through and revisiting the SDLC stages exist, including waterfall, V-shaped, and iterative methods [107, 138]. This chapter will focus on the first iteration of the early SDLC stages.

For industrial PHM solutions, the planning stage involves defining a solution's scope, or the role that it plays within a manufacturing operation's broader reliability strategy. At the beginning of this stage, solution developers should identify one or more PHM capabilities that would bring value to a manufacturing operation but are currently unavailable. This decision will guide subsequent specifications concerning the solution's deployment scale, maintenance response, and output quantities. Feedback from all system stakeholders (including reliability experts and equipment operators, among others) should be incorporated into a solution's scope. If a PHM solution's scope is not explicitly and thoroughly defined, its impact will suffer and the solution will not persist in the work process. Section 6.2 provides a methodology for defining the scope of an industrial PHM solution as well as classifications for the kinds of expertise that should be solicited during the planning stage.

The second stage of the SDLC, requirements and analysis, uses both qualitative and quantitative analysis to determine whether an operation's data and modeling resources are sufficient to develop a solution that satisfies the scope defined in the planning stage. The decisions and considerations that should be made during this stage are highly specific to the PHM solution being developed, so they will not be investigated here. [25] and [125] provide a general overview of how this stage may be implemented for purpose-driven digital twin systems, which can be applied to individual industrial PHM solutions.

Design is the third stage of the SDLC and the second SDLC stage that will be discussed in detail in this chapter. Industrial PHM solutions can combine several different models and computational processes to convert machine signals into results that are useful for minimizing system downtime. During the design stage, the structures of these processes are selected, and decisions are made about how health models will be trained. Recent literature has proposed approaches for designing PHM solutions for non-industrial applications, but industrial equipment and manufacturing environments often present additional challenges. Section 6.3 provides a general architecture for industrial PHM solutions that calls out processes not always considered for non-industrial applications. Two additional challenges that influence the design of industrial PHM solutions, data quality and modeling time-series degradation trends, are then discussed.

The material presented in prior chapters can be used to facilitate the remaining stages of the SDLC, including implementing, testing, deploying, and maintaining industrial PHM solutions, which are outside the scope of this chapter. Some stages, such as those related to implementation and deployment, will be highly specific to individual PHM solutions and manufacturing operations, while others have been examined by general PHM literature and can be directly applied to industrial PHM solutions [125, 139, 140].

6.2 Planning Stage

This section first introduces four types of subject matter expertise that should inform the planning stage of the SDLC for industrial PHM solutions. A methodology for completing this stage is then presented.

6.2.1 Subject Matter Expertise Classifications

Four types of expertise that are necessary during the planning stage (and subsequent SDLC stages) are defined below. In practice, individuals may possess multiple types of expertise, and there may be individuals possessing broad expertise, which overlaps with all of the classifications presented here. Those individuals should also be consulted during the planning stage, to supplement the knowledge of more specialized experts.

Business expertise: Knowledge of the financial aspects of a manufacturing operation. Business experts have an understanding of the production demands that apply to a manufacturing operation and the financial impacts of disruptions. An understanding of the costs (absolute or relative) of false negatives and false positives from

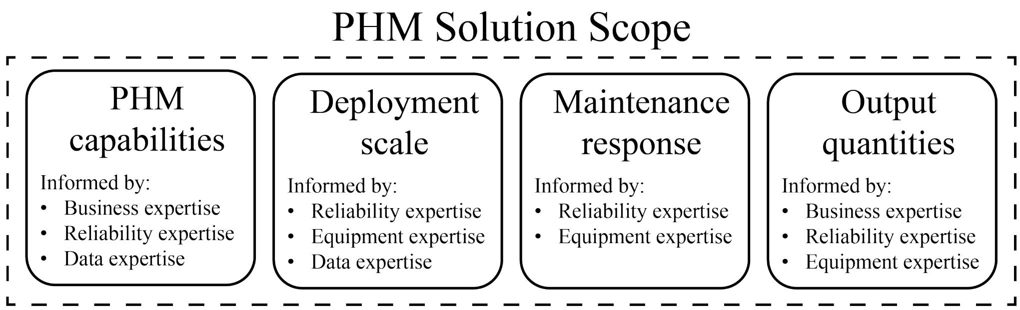

Figure 6.2: Specifications that constitute the scope of an industrial PHM solution and the subject matter expertise that should inform them

a prospective PHM solution is also necessary.

Reliability expertise: Knowledge of a manufacturing operation's current reliability strategy. Reliability experts are aware of the initiatives that are currently in place to ensure that a manufacturing operation remains online, including scheduled preventative maintenance procedures and protocols for administering reactive maintenance.

Equipment expertise: Knowledge of the machine(s) that a prospective PHM solution will monitor. Equipment experts have an understanding of the mechanical and electrical components that make up the machine and an awareness of its potential faults and failure modes.

Data expertise: Knowledge of a manufacturing operation's data collection capabilities. Data experts are aware of the signals that are available to be used in a PHM solution and understand how system health problems can manifest themselves in these signals. They are also familiar with the data reporting and storage infrastructure that will be used to implement the PHM solution and can provide insight into how data collection and processing can be improved to better support PHM solutions.

6.2.2 Planning Methodology

For industrial PHM solutions, the planning stage of the SDLC consists of defining four specifications that describe the solution's role in a manufacturing operation, referred to as the scope of a PHM solution. These specifications and the expertise that should inform them are depicted in Fig. 6.2. Each specification is described below in a suggested definition order, but the appropriate order for individual solutions may vary.

The definition of a PHM solution's scope should begin by specifying one or more high-value problems that exist within an operation's current reliability strategy. Based on this decision, one or more modeling and analysis tasks can be selected to address these problems. The academic research community has identified many PHM-related tasks [12, 141], which will be referred to as PHM capabilities. Several PHM capabilities are defined in [40], including detecting ongoing equipment degradation, diagnosing the root cause of degradation, and predicting when maintenance will be necessary. General anomaly detection and repair quality assessment are other examples of PHM capabilities that may be enabled. If multiple capabilities are identified as potentially useful, they may all be included in the scope of a prospective PHM solution. Validation results obtained later in the development process will dictate which capabilities can actually be delivered. Reliability expertise is necessary here to estimate the benefits of each PHM capability in terms of system uptime. Business expertise is also necessary to weigh the financial impacts of these benefits against the cost of developing a PHM solution that enables the desired capabilities. Finally, data expertise is needed to determine if a PHM capability can be enabled and maintained with the current data system or if the data system can be improved to better support a capability.

A PHM solution's scope should also specify the scale at which the solution will be deployed. This includes basic decisions about which piece of equipment or equipment subsystem to monitor, which should be motivated by pre-existing reliability shortcomings. Deployment scale specifications may also identify events that will trigger the activation or deactivation of a solution. For example, it may be appropriate to deactivate a PHM solution around startup and shutdown events because machine signals can be erratic during these periods If several similar machines are part of an equipment fleet, then a decision should be made about how many of these machines will be monitored by the PHM solution. Reliability and equipment experts should use their understanding of the likelihood of health problems throughout a machine's lifetime to inform these decisions. Data experts should also weigh in on the sensing capabilities across equipment fleets, which may influence the scale at which a solution can be deployed.

A set of maintenance actions that may be taken based on the outputs of a PHM solution should be specified next. Potential actions range from intensifying the manual supervision of a system to immediately shutting down a system for repair. Maintenance actions can be triggered automatically based on the outputs of a PHM solution, or can be the result of a negotiation between a PHM solution and an equipment operator. If the latter is preferred, the negotiation questions and a set of potential actions should be specified. Reliability expertise is necessary here to define how a prospective PHM solution can fit into an operation's existing reliability strategy. A key aspect of this specification is the amount of time that is necessary to take these actions. Equipment expertise should also be solicited to ensure that the specified actions are feasible and compatible with the allotted time frame.

Finally, a set of desired output quantities from the prospective PHM solution and the minimum levels of accuracy that would deliver a net benefit to an operation should be specified. PHM solutions designed for fault or degradation detection may output discrete state classifications while solutions designed for fault prediction may output probability density functions (PDFs) . Any desired measures of uncertainty, such as state probability estimates or confidence intervals, should also be specified here. Input from equipment experts is necessary to select output quantities that will allow the actions specified above to be taken. Business and reliability experts should also use their understanding of an operation's downtime and maintenance costs to specify minimum accuracy rates for a prospective solution.

6.3 Design Stage

This section is focused on the design of industrial PHM solutions. A general architecture for industrial PHM solutions is first proposed. The section then introduces two challenges that commonly arise in manufacturing environments: data quality and modeling time-series degradation trends. Several methods to adapt the design of equipment health models to overcome these challenges are discussed.

6.3.1 PHM Solution Architecture

Fig. 6.3 depicts a general architecture for industrial PHM solutions that includes a set of computational processes and a flow of information between them. A foundational aspect of the architecture is a computing platform that supports these processes and aggregates input data streams. The choice of an appropriate computing platform will be strongly influenced by a company's information technology (IT) and data security policies. For any application, establishing a reliable connection from data acquisition devices and historical databases to the chosen computing platform is essential.

The computational processes described below represent implementation choices that developers must make when designing a PHM solution. When developers identify multiple options for implementing a computational process, different candidate solutions can be designed in parallel and represented based on this architecture. Each process is described below in their runtime order, or the order in which they are executed when a PHM solution is actively monitoring the real-time health of a system. It is not necessary to design a solution in this order, though, and processes may be revised multiple times based on testing results.

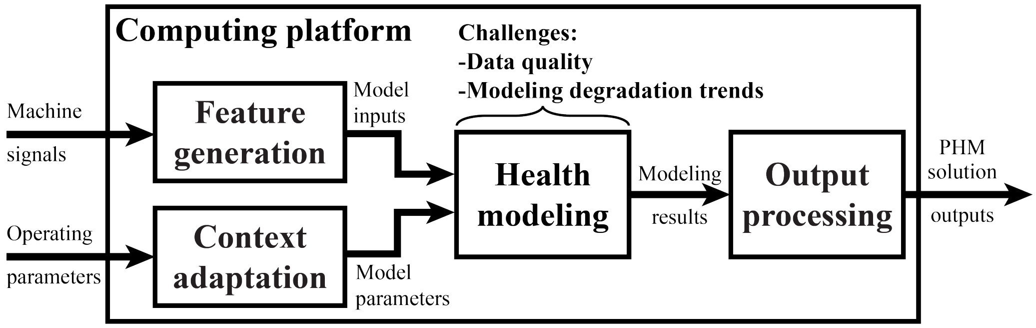

Figure 6.3: Architecture and flow of information within an industrial PHM solution

A feature generation process receives signal measurements from a machine and computes a set of model input features from those measurements. Common feature generation processes include extracting averages from windows of signal measurements or performing Fourier transforms on vibration data to extract frequency-domain features. When many different signals are available from a system, equipment experts can greatly simplify the feature generation process by identifying a subset of machine signals that contain useful information about the health of a system. Data experts should also be relied upon here to translate qualitative descriptions of unhealthy behavior provided by equipment experts into quantitative signal features that can be used as model inputs.

Context adaptation can be executed in parallel with feature generation during online monitoring. This architecture defines system context as the operating parameters and environmental factors that change over a system's lifetime (either intentionally or as a side-effect of degradation) and impact the machine signals monitored by a PHM solution (and thus the behavior being detected or predicted by the PHM solution). Rotating speed and product type are two examples of parameters that commonly impact system behavior. A context adaptation process tracks a set of context states and maps these state values to modeling parameter values that dictate how a system is analyzed in its current context [39]. While a PHM solution may forgo context adaptation by keeping modeling parameters constant throughout a system's lifetime, it is often beneficial to use context adaptation to translate system health models or switch between multiple models based on system context. Equipment experts often have an understanding of what parameters vary throughout a system's lifetime and how these changes impact system behavior. This expertise should be used to identify a set of context state candidates that are considered for inclusion in a solution's context adaptation process.

The health modeling process uses the outputs from feature generation and context adaptation to produce modeling results that describe system health in some way. Recent literature has proposed a diverse set of health modeling approaches, but only a small number of these may be appropriate based on the scope of the prospective PHM solution. Solution developers should carefully consider two challenges that commonly afflict industrial PHM solutions: data quality and modeling time-series degradation trends. While academic research commonly uses data collected from defective equipment or run-to-failure tests, it is usually not feasible to collect these data from manufacturing equipment that must be kept online. Historical measurements collected during online operation are often the only available datasets and require careful, SME-informed pre-processing to be used for model training. Additionally, equipment failures may be rare events, which makes these datasets unbalanced and presents difficulties training many of the machine learning-based models proposed by recent literature. Another challenge for PHM developers is equipment degradation that is characterized by time-series trends in machine signals (potentially beginning from varying baseline values). When this is a possibility, equipment health models should be designed to consider histories of recent observations when estimating current and future health states. Methods that neglect system history when making health state estimates are either limited in their ability to detect early-stage degradation, such as univariate signal limit monitoring, or hyper-specific to a single application, which limits their robustness to disturbances. Subsection 6.3.2 and Subsection 6.3.3 discuss methods that can be used to overcome both of these challenges when designing industrial PHM solutions.

Finally, output processing consists of any computations to format or synthesize modeling results into solution outputs that are delivered to end users. This may involve aggregating or merging the outputs of multiple health models, or appending estimates of uncertainty to modeling results. Output processing may not be necessary for all applications, but can be used in some cases to convert modeling results into the output quantities specified by a solution's scope. Both equipment and reliability experts should be consulted when selecting an output processing approach to ensure that the implications of solution outputs are clear and actionable for the solution's end-users.

6.3.2 Data Quality

6.3.2.1 Pre-Processing Historical Data

Early in the development of a PHM solution, decisions must be made about the source of data that will be used to train the health models within a PHM solution. In many industrial applications, databases of machine signal measurements from periods of online operation, referred to as historical datasets, are the only available source. In their raw form, though, these datasets are not well suited for model training. A method for filtering and labeling these historical datasets is presented here.

The method begins by identifying the following events in a historical dataset. Equipment maintenance logs and historical equipment state tag values can help accomplish this.

> $Fault$: An occurrence associated with an unwanted situation within an manufacturing system that must be resolved through maintenance
>
> $Shutdown\ for\ repair$: Transition of a system from an online state to an

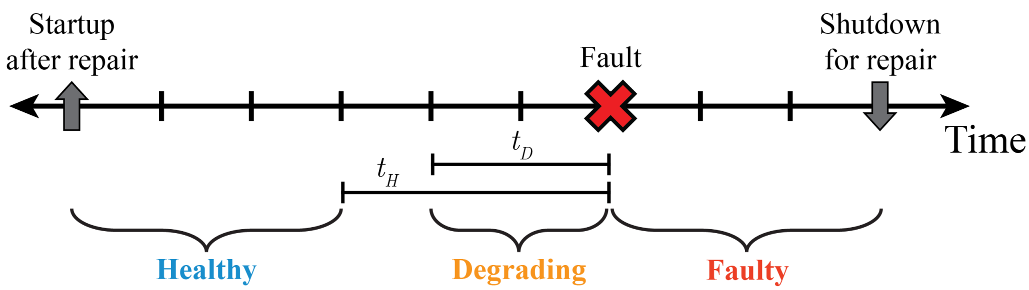

Figure 6.4: Historical data labeling convention over an example timeline

offline state for the purpose of conducting a repair procedure

Startup after repair: Transition of a system from an offline state to online state after a repair procedure has been completed

Fig. 6.4 shows an example timeline for a historical dataset with these events labeled. Input from equipment and reliability experts should be used to specify an assumed degradation length (t_D) and a healthy buffer length (t_H) for the system and fault being monitored. The value of t_D defines the maximum time prior to a system fault that a system can reasonably be assumed to be *Degrading*. The value of t_H defines the minimum time prior to a system fault that a system can reasonably be assumed to be *Healthy*. Periods of historical data can then be labeled as *Healthy*, *Degrading*, or *Faulty* based on these parameters, as shown in Fig. 6.4. The labeled datasets can be used to train models for different PHM capabilities, including degradation detection, fault detection, and fault prediction.

6.3.2.2 Combined Training Datasets

PHM solution developers may also be limited by the amount of data that is available for training for model training. Industrial equipment often undergoes shifts in operating conditions, such as maintenance events involving mechanical design changes, that change the distribution of machine signals and render older historical data unsuitable for model training. Additionally, historical datasets usually suffer from class imbalance, characterized by a small amount of unhealthy measurements relative to healthy measurements. This can be the result of rare $Fault$ events or the practice of carrying out preventative maintenance procedures before system degradation begins. Limited training data limits both the types of health models that can be implemented in a PHM solution and the accuracy of the solution.

To overcome data sparsity in individual machines or contexts, a context adaptation approach can combine datasets from related machines or contexts. In practice, this can mean combining data from identical machines operating in the same plant, data from similar machines operating in different environments, or even data from the same machine from different periods of time (before and after machine overhauls, for example). Previous research has proposed several methods for applying knowledge learned in one domain to other domains, a transfer learning approach known as domain adaptation [142, 143]. The developers of an industrial PHM solution may choose to implement any of these methods, which span a wide range of theoretical and computational complexities. Recent literature has also proposed metrics for quantifying the similarity of datasets from different machines or contexts. Maximum mean discrepancy is often used to estimate the difference between datasets and both transfer component analysis and linear discriminant analysis are designed to identify low-order transformations that align datasets from different domains [144, 145]. These analysis methods can help avoid decreases in model accuracy that result from combining datasets with drastically different distributions.

An initial foray into domain adaptation can be realized by simply normalizing datasets from separate contexts before combining them into a single model training dataset. This entails defining a set of k context states that are each associated with a system or operating condition that is considered distinct from other context states. Mean and standard deviation vectors μ_i, σ_i are calculated for each context state, based on the data collected under those conditions, where $1 \leq i \leq k$. The data associated with each context state are then normalized based on these mean and standard deviation vectors before combining all data into a combined dataset that is used for model training. If a PHM solution is developed with this approach, the context-specific mean and standard deviation vectors must be retained to normalize new measurements before providing them as inputs to the trained models.

6.3.3 Modeling Time-Series Degradation Trends

Another challenge that should be accounted for when designing an industrial PHM solution is the possibility of time-series degradation trends in manufacturing equipment. Recognizing gradual degrading behavior in industrial equipment is often crucial for triggering low-cost maintenance procedures and avoiding downtime. Many complex equipment degradation processes also experience multiple discrete degradation stages that are characterized by different machine signal behaviors. To account for these phenomena, health models can maintain estimates about the current state of equipment health that are informed by historical measurements, an approach that will be referred to as state-based health modeling.

General path modeling is an example of a state-based modeling approach that identifies gradual degradation trends in machine signals and extrapolates this behavior to make failure predictions [34, 42]. Certain general path models have a small number of stochastic parameters that can be re-estimated according to a Bayesian update strategy when new signal measurements are made. The resulting failure predictions are then probabilistic windows that correspond to a user-specified level of certainty. When equipment faults are rare and training data is limited, these low-complexity models can be more practical to train and deploy than deep learning models.

A survey of other model-based methods for detecting and diagnosing degradation can be found in [32]. These include strategies that represent equipment dynamics and degradation trends using state-space models or differential equations. Observer-based methods can then be used to estimate health-related model parameters based on system outputs [26, 27] or compute residuals between expected and actual system outputs that indicate degradation [28, 74].

In cases where developers would like to capture multiple discrete degradation stages in a health model, hidden Markov models (HMMs) are a useful modeling tool [29, 68]. The unobservable states in an HMM can be used to represent discrete system health stages, including a $Healthy$ state characterized by steady-state behavior in machine signals, and one or more unhealthy states, such as $Degrading$ or $Faulty$ states, that are characterized by dynamic signal behavior or undesirable signal levels. A $Degrading$ state may also be divided into multiple sub-states to represent different physical degradation mechanisms that are known to precede system faults. In an HMM, each of the unobservable states are associated with an observation probability distribution that describes the likelihood of certain observable features. For PHM solutions, these observable features may be the values of machine signals, or trend-based features extracted from windows of machine signals [33]. The Viterbi algorithm can then be used to compute the most likely state history of a system based on historical observations.

HMM state estimation methods can also be used to incorporate state-based modeling into preexisting health state classifiers (when a model's historical accuracy rates are known). To achieve this, a model's classification results are treated as observable features. In an HMM, each state's observation probability distribution defines the probability of making observation o at time t when the system is in state S_i. This can be expressed as

$$b_i(o) = P(o \text{ at } t | q_t = S_i) \tag{6.1}$$

where $B = b_i(o)$ is the observation probability distribution of state S_i and q_t is the true state of the system at time t. A classifier's accuracy rates, often formatted as a confusion matrix, express the same likelihoods. Observations at each time t

are predicted state classification ($\hat{q}_t$), and $b_i(\hat{q}_t)$ is the likelihood of receiving a $\hat{q}_t$ classification when the true system state is S_i. A classifier's true positive rate and false positive rate define the observation probability distributions in each state. The Viterbi algorithm can then be used to make health state estimates based on a history of classification results.

In some applications, Markov models of equipment health can enable predictive capabilities. Hidden semi-Markov models are used most often for fault prediction [71]. These models allow state transition probabilities to change over time to reflect system deterioration. The time remaining until a system enters a failure state, commonly referred to as remaining useful life (RUL), can then be estimated using methods described in [72] and [73].

6.4 Case Study PHM Solution Development

This section describes the process of developing a PHM solution during a graduate student internship at The Dow Chemical Company. The details presented here have been reviewed by the company and approved for external release. Each of the SDLC stages shown in Fig. 6.1 were used to develop four candidate PHM solutions that are currently being tested before deployment. These solutions and the process of developing them are presented here.

6.4.1 Planning

The planning stage of the development process for this PHM solution involved defining the solution's scope based on a pre-existing reliability problem at a low-density polyethylene (LDPE) manufacturing operation. In recent years, the operation has been plagued by frequent faults in an industrial hyper compressor that result in emergency shutdowns and lost production. Business and reliability experts first determined that the manufacturing operation would benefit from a PHM solution that provides degradation detection capabilities. Such a solution would provide warnings of impending faults in the machine.

Reliability, equipment, and data experts agreed that the system should be deployed to monitor the health of the hyper compressor during all periods of online operation. Additionally, because the hyper compressor consists of eight cylinders that operate in parallel and are all susceptible to faults, all cylinder subsystems are included in the PHM solution's scope. Equipment experts were then consulted to define how a PHM solution could be integrated into the plant's existing reliability strategy. They concluded that, given an indication of an impending fault in the hyper compressor, equipment and reliability experts would first attempt to resolve the fault without a shutdown by manipulating the operating parameters of the machine. If this is not possible or unsuccessful, equipment maintenance would be scheduled at a convenient time in the near future.

Finally, reliability and equipment experts specified a set of desired PHM solution outputs. An important decision was made here to monitor the health of each hyper compressor cylinder individually, rather than monitoring the overall health of the machine. The most desirable output of the prospective PHM solution is indications of degradation in each cylinder in the form of a discrete health state from the set $\{Healthy, Degrading\}$. Another desirable output quantity from a prospective PHM solution is a set of signal identifiers that accompany *Degrading* health state classifications to indicate the machine signals that are deviating from expected behavior. This information may help equipment experts to identify the root cause of degradation and determine if it can be resolved without a shutdown. Experts also concluded that the PHM solution would need to provide warnings of ongoing degradation approximately 2 weeks before system fault to have sufficient time to investigate the problem and schedule system maintenance. Because the PHM solution is meant to serve a purely advisory role in the operation's reliability strategy, no minimum accuracy rates were imposed on the solution at this stage.

6.4.2 Requirements and Analysis

At this stage in the development process, experts conducted an analysis to gauge the feasibility of the proposed PHM solution. Several years ago, the hyper compressor was equipped with several vibration sensors on each cylinder to facilitate the detection of faults and degradation. Other standard operating sensors, such as inlet and outlet gas properties, are also available to be used as inputs to a PHM solution. Additionally, the manufacturing operation has a historical dataset of these measurements dating back several years along with equipment failure dates that can be used for model training. Experts agreed that these sensing capabilities and training datasets would be adequate to develop a PHM solution to enable degradation detection.

6.4.3 Design

The design stage of the PHM development process resulted in four PHM solution candidates, each with a different set of implementations for the four computational processes discussed in Subsection 6.3.1. The architectures of the solution candidates are depicted in Fig. 6.5 and each implementation approach is described below.

6.4.3.1 Context Adaptation

Different context adaptation approaches were developed to satisfy requests for cylinder-specific health state state estimates. The first approach, used in Solutions 1-3, involves building cylinder-specific models based on data from each cylinder individually. A cylinder identifier, which can be considered a context state, accompanies new measurements and dictates which model is used to make health state estimates. An alternate approach, used in Solution 4, normalizes historical data from each cylinder and combines them to form a combined training dataset, as described in Sub-

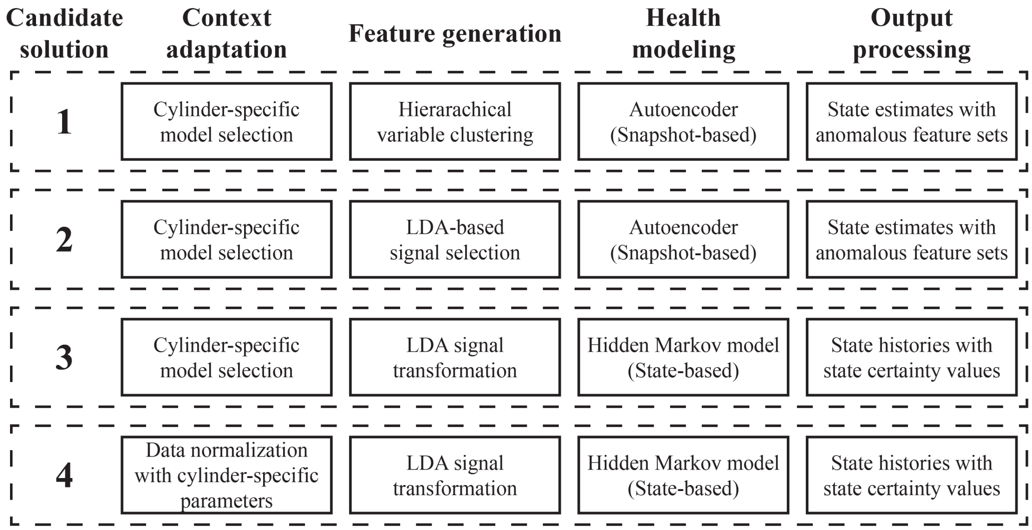

Figure 6.5: Architectures and computation implementations of the four candidate PHM solutions considered in this case study

section 6.3.2. With this approach, cylinder identifiers dictate the parameters used to normalize new measurements before providing these values as inputs to a single health model that makes health estimates for all cylinders.

6.4.3.2 Feature Generation

The LDPE manufacturing operation collects a large amount of data from various pieces of equipment and subsystems, so sensor selection and feature generation is a critical aspect of the PHM solution. With the help of equipment and data experts, a subset of the operation's signals were identified as potentially relevant for monitoring the health of the hyper compressor. The signals in this subset were used as inputs to a set of feature generation approaches that were proposed and tested during the development process.

The feature generation approach in Solution 1, known as hierarchical variable clustering (HVC), uses principle component analysis to form clusters of highly-correlated signals and select a signal representative from each cluster [146]. Historical data from the hyper compressor was labeled according to the process described in Subsection 6.3.2, and data from periods of *Healthy* and *Degrading* operation were combined to form a training dataset for the algorithm. While data from *Degrading* periods was used here, the HVC algorithm can be used to generate features from *Healthy* data alone if data sparsity is a concern for an application. The outcome of this approach is a reduced number of signals that capture the variability of the training dataset and can be used as health model input features.

The feature generation approaches in Solutions 2-4 are based on the linear discriminant analysis (LDA) method described in [145]. LDA seeks to derive a linear transformation that minimizes the distance between data points with identical classifications and maximizes the distance between data points with different classifications. Conventional LDA is used in Solutions 3 and 4 to determine a set of components that map the original machine signals to smaller, transformed input features. Historical data with *Healthy* and *Degrading* classifications were used to determine these components. However, it can be difficult to interpret the meanings and implications of these transformed features. Solution 2 uses a modified feature generation approach that seeks to identify model input features that exhibit the greatest difference between *Healthy* and *Degrading* periods. The approach first ranks the original machine signals based on the magnitude of their coefficients in the first LDA component. The d highest-ranking machine signals are then selected as health model input features. The parameter d is set equal to the number of variable clusters computed for Solution 1, so that the same autoencoder model can be used in both candidate solutions.

6.4.3.3 Health Modeling

Candidate solutions 1 and 2 use autoencoder health models to make health state classifications for each cylinder. Autoencoders are a form of neural network that seek to compress and then reconstruct their input features [147]. Autoencoders are well-suited for applications where equipment faults are rare because they can be trained with exclusively *Healthy* historical data, which is readily available from most manufacturing equipment. Significant deviations between the original features and the reconstructed features indicate anomalous behavior in the system being monitored. Health state classifications are obtained here by checking whether the root-mean-square error (RMSE) between original and reconstructed features exceeds a pre-defined threshold. If so, the system is classified as *Degrading*. Otherwise, the system is classified as *Healthy*. The structure of the autoencoder networks used in these solutions was optimized to provide robust representations of the input space during normal operating conditions. Because the autoencoder implemented here uses only the most recent measurements from the hyper compressor to make health state estimates, this modeling approach is denoted as "snapshot-based" in Fig. 6.5

Solutions 3 and 4 use HMMs to make health state classifications. The statechart shown in Fig. 6.6 depicts the HMM's unobservable states and possible transitions. The observation probability distributions for each unobservable state are estimated by fitting a Gaussian distribution to historical data from *Healthy* and *Degrading* periods. In Solution 3, different observation probability distributions are estimated for each cylinder using data from that cylinder only. In Solution 4, normalized data from all cylinders are combined into larger *Healthy* and *Degrading* datasets that are used to estimate the HMM's observation probability distribution. Identical copies of the HMM are then deployed to monitor each cylinder. As described in Subsection 6.3.3, the Viterbi algorithm is used to estimate the health state history of each cylinder whenever new signal measurements are collected. Because the health state estimates provided by an HMM are informed by a recent history of hyper compressor measurements, this modeling is denoted as "state-based" in Fig. 6.5.

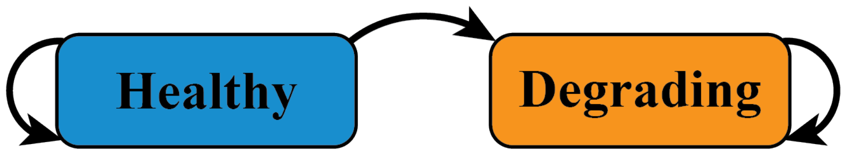

Figure 6.6: Hidden Markov model statechart used to represent hyper compressor health in this case study

6.4.3.4 Output Processing

All solutions developed for this application provide the most important output quantity: state estimates from the set $\{Healthy, Degrading\}$ describing the current health state of each hyper compressor cylinder. However, each solution carries out computations to provide additional outputs that add context to the health state estimates. It is critical to consider the full set of outputs that each solution provides, alongside measures of testing performance, when deciding which candidate solution to deploy.

For Solutions 1 and 2, which use an autoencoder modeling structure, health state estimates are determined based on the combined RMSE of all input features. However, the solution's input features can also be ranked by comparing feature-specific error values with one another. This information supplements *Degrading* state estimates with an ordered set of input features that are deviating most significantly from expected behavior. The highest-ranking features can inform efforts to investigate and resolve system degradation.

Because of their feature generation implementations, Solutions 3 and 4 are not able to provide sets of anomalous input features, but can provide probabilistic state certainty values and retroactive estimates of previous health states. These quantities are computed as part of the Viterbi state estimation algorithm, so no further additional computations are necessary to derive them. State certainty values allow system and reliability experts to calibrate their maintenance response based on the confidence that a solution has in its health state estimates. For example, one may choose to merely monitor a system more closely when certainty in a *Degrading* state estimate is low, or to immediately schedule a planned shutdown when confidence in a *Degrading* state estimate is high. The solutions' ability to revise previous health state estimates may be useful when appropriate maintenance actions are dependent on the time since system degradation began. Measurements that were once considered *Healthy* may be revised to *Degrading* based on subsequent measurements, for instance, allowing experts to trigger a more aggressive maintenance response.

6.5 Case Study Insights

This section discusses general takeaways from the early stages of developing the case study PHM solution using the methods presented in Section 6.2 and Section 6.3, as well as guidelines for applying these methods in other applications.

6.5.1 Planning

The methodology detailed in Section 6.2 guided the planning stage and organized the input received from the PHM solution's various stakeholders. An important piece of input solicited during this process was requests from reliability and maintenance experts for modeling outputs that could be used to investigate the root cause of system degradation. This information influenced the health modelling and output processing approaches in Solutions 1 and 2. Another impactful decision made during the scope definition process was the 2-week detection requirement. The assumed degradation length (t_D) was set to 2 weeks because of this requirement, which had significant ramifications for the size of model training datasets and model performance. It is critical to specify this value early in the model development process to ensure a PHM solution's long-term effectiveness. When developers impose a similar detection requirement on their solution, a feasibility study should be conducted during the requirements and analysis stage to determine whether the available machine signals exhibit anomalous behavior prior to the minimum detection time.

6.5.2 Design

The PHM solution architecture and computational processes introduced in Section 6.3 helped to standardize the development and comparison of candidate PHM solutions. This made it easier to identify when approaches can be combined, like Solution 1 which uses an autoencoder with input features generated from hierarchical variable clustering, or substituted, like solutions 3 and 4 which use different context adaptation approaches. The proposed architecture also calls attention to context adaptation and output processing decisions, which are commonly overlooked in industrial PHM solutions. Decisions about which context states are incorporated into a solution and what output quantities a solution delivers have a considerable impact on its accuracy and value for end users.

6.5.3 Data Quality

A primary takeaway from implementing the strategies proposed in Subsection 6.3.2 for filtering and labelling historical training data is the need for high-quality documentation on machine faults and shutdowns. This information is necessary to identify the historical *Fault* events and period of system degradation. If the machine conditions that motivate a machine shutdown and the duration of the problem are not recorded in sufficient detail, it is impossible to pre-process historical data for model training.

Like most industrial applications, there are a limited number of historical degrading periods for each of the hyper compressor cylinders. PHM Solution 4 was designed to address this limitation by training a single health model with historical data from all cylinders. Early testing shows that the approach improved degradation detection capabilities in all cylinders, most significantly in cylinders with relatively few periods of degradation in their histories. This result suggests that data normalization can be an effective approach for identifying degradation trends that are shared between machines or machine subsystems and boosting the accuracy of system models with limited training data.

6.5.4 Modeling Time-Series Degradation Trends

HMMs were utilized in Solutions 3 and 4 to implement a state-based analysis of hyper compressor system health. Compared to the autoencoder solutions, the HMM solutions exhibited fewer false positives in preliminary tests, which increased overall accuracy rates. Their ability to consider a time series history of recent measurements when making state estimates proved valuable in this application, especially when brief process disruptions introduce variability into signal measurements.

The performance gains, however, come with added model training work and reductions in model transparency. HMM parameters, including observation probability distributions and state transition probabilities, must be estimated prior to model deployment. Strategies for estimating HMM parameters can be found in [33, 68], but they are not as well-defined or as automated as parameter tuning methods for other data-driven model types. Additionally, Solutions 3 and 4 use a small number of transformed features as model inputs to reduce computational complexity. Because the features are linear combinations of a large number of machine signals, an operator's ability to identify the root cause of degradation based on feature trends may be diminished. In some applications, this limitation can outweigh the benefits to model accuracy that an HMM achieves.

6.6 Conclusions

This chapter presents a framework for developing industrial PHM solutions derived from the first steps of the SDLC. Recent academic literature includes a wide variety of PHM-related modeling approaches, but the practical challenges associated with developing them for industrial environments are not always considered. The SDLC stages examined in detail here: planning and design, are critical for the long-term success of individual PHM solutions and the expansion of predictive maintenance strategies in manufacturing. The methodologies developed for completing these stages and the methods for overcoming common challenges are valuable for manufacturers as they integrate PHM solutions into their reliability strategies. The case study presented here provides a concrete example of how this framework can be applied to develop PHM solutions for manufacturing equipment and gives key takeaways for PHM developers interested applying these concepts in their applications.

CHAPTER 7

Conclusions and Future Directions

The emergence of Industry 4.0 and smart manufacturing technologies in recent years has given manufacturers access to more machine and process data than ever before. Researchers in academia and industry have identified equipment reliability as an aspect of manufacturing operations that could benefit from the greater visibility that new sensing and data collection programs offer. While it may seem as though access to large quantities of machine data would be immediately useful for detecting and diagnosing health problems, the practical reality is that much of these data currently go unused. Accurate, scalable, and trustworthy modeling and analysis programs are difficult to design and deploy in any environment, but especially in manufacturing operations, where disturbances are frequent and pre-existing production quotas must be met. While many advanced and innovative health modeling methods have been proposed in academia, the industry is just beginning to implement concepts from the literature in manufacturing operations beyond individual, isolated use cases. For predictive maintenance to be feasible at a larger scale, health modeling frameworks and methods that address the unique challenges of manufacturing applications must be developed.

This dissertation proposes adaptive, state-based health modeling frameworks and methods designed to facilitate the deployment of industrial prognostics and health management (PHM) solutions. State-based PHM solutions represent an opportunity to detect and predict unhealthy machine behavior using information from time-series machine signal trends and multi-stage models of degradation. Health models that take this approach are often more robust to changes in operating parameters and can be deployed across a wider range of machines than existing snapshot-based models. The operating model and digital twin (DT) framework presented in Chapter 3 and Chapter 4 provide a foundation to develop standardized, state-based PHM solutions for a wide range of mechanical systems. Case studies with multiple types of rotating equipment demonstrate the general nature of these contributions and the value that it provides. An adaptive framework presented in Chapter 5 builds upon this foundation to provide state estimation and fault prediction capabilities for applications where exact models of equipment degradation or large model training datasets are not available. The results of an analysis of multiple run-to-failure experiments demonstrate how this method provides early warnings of system health problems and accurate predictions of future faults.

7.1 Contributions

Fig. 7.1 depicts the contributions of this dissertation, which are listed and summarized below, in the context of the equipment operating model introduced in Chapter 3 and the PHM solution development timeline discussed in Chapter 6.

1. *A general operating model for state-based PHM*

A discrete-state model of mechanical manufacturing systems during online and offline periods is presented to support the design of state-based PHM solutions, as shown in Fig. 7.1. This model features a hierarchy of online states that describe machine health with respect to quantitative system health specifications. The inclusion of degrading sub-states allows PHM solutions to track multi-stage degradation processes.

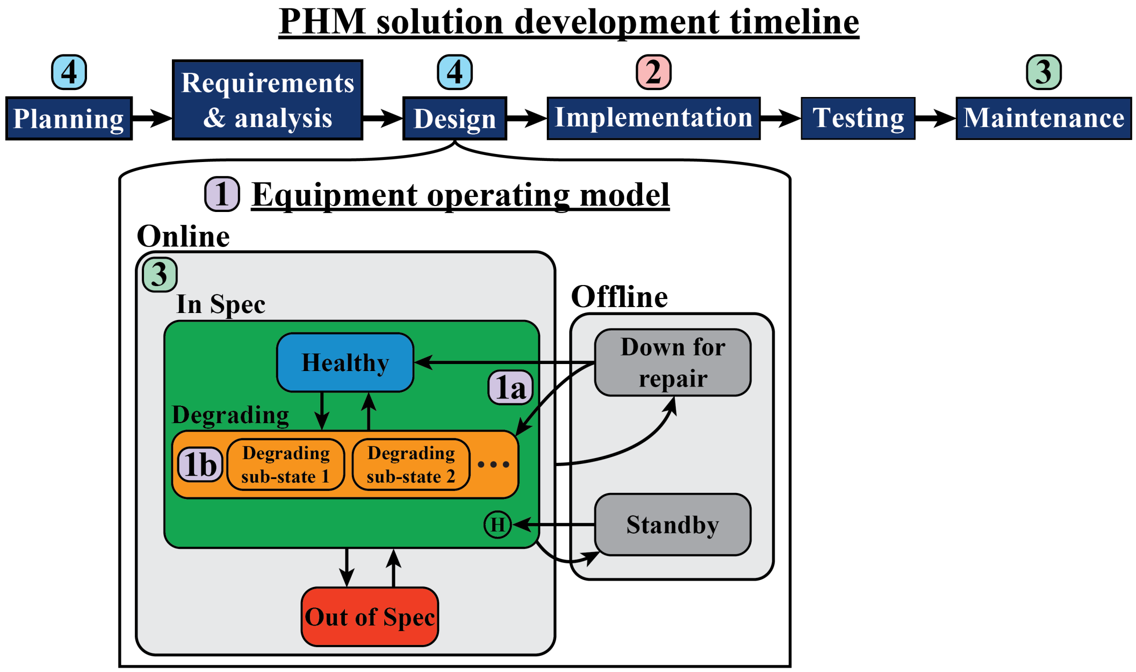

Figure 7.1: Contributions of this dissertation in the context of the PHM solution development timeline and the industrial equipment operating model

Transitions between offline and online operation are also incorporated into the model to contextualize gaps in machine operation, which can be attributed to equipment maintenance or other events in a manufacturing process.

Two novel, state-based modeling methods are developed based on this operating model and represent sub-contributions of this dissertation:

(a) A method for trend-based repair quality assessment in industrial equipment

(b) A method for multi-stage fault prediction in rolling element bearings

The first method provides the ability to estimate a system's health state after a transition out of a *Down for repair* state and the second method provides an example of how the *Degrading* state can be divided into multiple sub-states to enable fault prediction in rolling element bearings. Case studies that implement these methods to model the health of a boiler feedwater pump and rolling element bearings demonstrate how the underlying model captures general characteristics of mechanical system health as well as the advantages of state-based modeling approaches.

2. *An extensible digital twin framework for PHM*

A framework that facilitates the implementation of a wide range of PHM capabilities using purpose-driven digital twins is presented, as shown in Fig. 7.1. The operating model proposed by the first contribution is the basis for this framework, which focuses on the modeling and computation processes that are required to deploy PHM solutions to monitor system health in real-time. The concept of purpose-driven DTs is applied here to define a set of general DT classes that perform different state estimation and prediction functions and generate standardized outputs. This approach allows a general DT architecture and class-specific I/O protocols to be reused across related machines and components and makes it possible to establish aggregation relationships between DTs to represent complex equipment. A case study uses the framework to model the health of a pump using a hierarchy of subsystem and component DTs.

3. *An adaptive and extensible* framework for multi-stage degradation monitoring and anomaly detection

A framework that extends the state-based modeling methods proposed in Chapter 3 to detect multiple different degradation modes and recognize unexpected degradation behavior is presented, as shown in Fig. 7.1. This approach is compatible with the high degrees of uncertainty surrounding industrial equipment health and limitations on the amount of model training data that is available from unhealthy systems. The proposed modeling framework uses trend-based feature trajectories specified by system experts to represent possible system degradation stages. Based on this representation of system health, an adaptive methodology probabilistically estimates a system's current and historical degradation stages based on recent sensor measurements and extrapolates the current trajectories to predict system faults. This contribution also provides mechanisms to maintain PHM solutions by detecting novel degradation modes and expanding a machine's multi-stage health model accordingly. A case study analyzing multiple rolling element bearing faults shows that this approach achieves comparable prediction accuracy rates compared to existing methods and detects signs of early degradation that those methods miss.

4. *Methodology for planning and designing industrial PHM solutions*
Structured methodologies for planning and designing industrial PHM solutions are presented here, as shown in Fig. 7.1. While the design of health models is the focus of most academic literature, and much of the work in this dissertation, the planning stage of industrial PHM solution development can ultimately determine its usefulness for predictive maintenance. This contribution outlines key decisions that should be made when planning the scope of an industrial PHM solution and the stakeholders that should be consulted during this process. Additionally, several challenges that are common in industrial applications but not always considered when designing PHM solutions are identified. Methods to overcome these challenges are proposed, along with a general computational architecture for industrial PHM solutions that makes it straightforward to compare different solution candidates. The details and takeaways of developing a PHM solution for an industrial hyper compressor are presented to advise researchers and engineers interested in implementing predictive maintenance strategies in their operations.

7.2 Opportunities for Future Work

The contributions of this dissertation provide a foundation for state-based PHM solutions for manufacturing equipment that can be extended in several directions by future work. While the frameworks and modeling methods put forth here are general to a wide range of systems, the case studies used to validate them were limited to rotating equipment. This is a common limitation of PHM research in general, due in large part to the volume of public experimental data from rotating equipment relative to other types of machines. Further testing and validation should be done with other systems, robots or batch production equipment for example, to identify whether additional considerations are necessary to apply these PHM concepts throughout the manufacturing industry. Additional limitations of this work represent promising areas for future research. Three of these are described below.

7.2.1 Unsupervised Learning of Multi-Stage Degradation Models

The contributions of Chapter 3 and Chapter 4 provide a general framework to develop state-based PHM solutions, but the case studies in these chapters demonstrate that significant input from subject matter experts (SMEs) is required to specify the stages of system degradation and machine signal trends that characterize degradation. Future research could extend this work by proposing automated methods to identify degradation stages and associated feature trajectories from unlabeled datasets. Existing work has put forth unsupervised learning methods that cluster machine data based on the magnitudes of machine signals [29, 70, 71], which could be extended to analyze time-series signal trends in the same manner. Breakthroughs in this area would significantly reduce the time needed to develop state-based PHM solutions and potentially uncover insights on equipment health that SMEs were not previously aware of.

7.2.2 SME-Informed Model Updates Based on System Anomalies

The modeling framework and methodology proposed in Chapter 5 support the detection of feature trajectories that deviate from a system's global health automaton and provide mechanisms to expand global automata in response to new degradation modes. However, further investigation of how to characterize anomalous behavior and propose new health stages would be extremely valuable for manufacturers. In industrial applications, model maintenance actions will need to be approved by system experts, but many analysis steps could be automated to facilitate rapid responses to unexpected events. This research extension would also be a testing ground for two-way communication methods between humans and DTs, which have not yet been explored in detail.

7.2.3 Transfer Learning Across Related Systems, Environments, and Contexts

Finally, differences in domains are a major obstacle to using much of the machine data currently collected by manufacturers. Often, it is not clear whether sensor measurements collected from different machines, different environments, or under different operating contexts should be analyzed separately or may be combined to train a single, universal system health model. The answer will surely vary based on the application and the type of system being modeled, but there are currently no widely-accepted methods for segmenting and grouping datasets from different domains for health modeling purposes.

Transfer learning is an emerging area of research that strives to tackle this type of problem [142]. PHM researchers have begun to utilize concepts from transfer learning in recent years, proposing methods to measure similarity between machine datasets and learn domain-invariant feature sub-spaces [98, 99, 143]. Future work can build on this research by developing methods to identify multivariate health indicators that show consistent degradation trends across machines in different environments and contexts. Further investigation of how low-order feature or model transformations can be used to adapt to operating context changes is also worthwhile.

7.3 Impact

The contributions of this dissertation will facilitate more widespread adoption of predictive maintenance strategies in the manufacturing industry. The proposed operating model for mechanical systems gives companies a structured basis for creating state-based PHM solutions. This will reduce the time needed to develop PHM solutions and allow manufacturers to take advantage of the transparency and robustness benefits that state-based modeling offers.

The extensible DT framework described here will allow manufacturers to connect and reuse PHM modeling resources more easily. The DT classes used by the framework can be deployed across different types of machines, while allowing modifications to individual models and pre-processing functions to be made where necessary. DT aggregation relationships also support communication between DTs that can be used to represent complex machines. When manufacturing equipment is modeled with DT hierarchies, subsystem and component DTs can be considered modular entities that may be duplicated, swapped out, or re-used across different systems.

The adaptive modeling framework can significantly reduce the resources needed to develop and train predictive health models in applications where SMEs have knowledge about system degradation. With this approach, multi-stage degradation models are created without the need for extensive historical training datasets. Anomaly detection capabilities that are not included in existing methods are also enabled. The ability to differentiate between expected and unexpected degradation modes gives maintenance engineers and system operators more information to resolve ongoing health problems and helps to instill confidence in PHM solution outputs.

APPENDIX A

Data Sources

Chapter 3

The case study presented in Section 3.3 uses historical data from a boiler feedwater pump operating in a manufacturing plant owned by The Dow Chemical Company. This dataset is confidential, so it is not publicly available. The case study presented in Section 3.4 uses data from multiple run-to-failure experiments with rolling element bearings. This dataset is documented in [122].

Chapter 4

The case study presented in Section 4.3 uses historical data from an ethylene pump operating in a manufacturing plant owned by The Dow Chemical Company. This dataset is confidential, so it is not publicly available. The case study also uses data from multiple run-to-failure experiments with rolling element bearings. This dataset is documented in [126].

Chapter 5

The case study presented in Section 5.3 uses data from multiple run-to-failure experiments with rolling element bearings. This dataset is documented in [126].

Chapter 6

The case study presented in Section 6.4 uses historical data from a hyper compressor operating in a manufacturing plant owned by The Dow Chemical Company. This dataset is confidential, so it is not publicly available.

BIBLIOGRAPHY

[1] "2022 manufacturing industry outlook," Deloitte, 2021. [Online]. Available: https://www2.deloitte.com/us/en/pages/energy-and-resources/articles/manufacturing-industry-outlook.html

[2] "November 2021 manufacturing ISM report on business," Institute for Supply Management, 2021. [Online]. Available: https://www.ismworld.org/supply-management-news-and-reports/reports/ism-report-on-business/pmi/november/

[3] R. K. Mobley, *An Introduction To Predictive Maintenance.* Betterworth-Heinemann, 2002.

[4] M. Haarman, M. Mulders, and C. Vassiliadis, "Predictive maintenance 4.0: Predict the unpredictable," PWC, 2017. [Online]. Available: https://www.pwc.be/en/documents/20171016-predictive-maintenance-4-0.pdf

[5] D. S. Thomas and B. Weiss, "Maintenance costs and advanced maintenance techniques: Survey and analysis," *International Journal of Prognostics and Health Management*, vol. 12, no. 1, 2021.

[6] K. F. Martin, "A review by discussion of condition monitoring and fault diagnosis in machine tools," *International Journal of Machine Tools and Manufacture*, vol. 34, no. 4, pp. 527–551, 1994.

[7] S.-H. Ding and S. Kamaruddin, "Maintenance policy optimization – literature review and directions," *The International Journal of Advanced Manufacturing Technology*, vol. 76, pp. 1263–1283, 2015.

[8] C. K. Mechefske and Z. Wang, "Using fuzzy linguistics to select optimum maintenance and condition monitoring strategies," *Mechanical Systems and Signal Processing*, vol. 15, no. 6, pp. 1129–1140, 2001.

[9] R. Ahmad and S. Kamaruddin, "An overview of time-based and condition-based maintenance in industrial applications," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 135–149, 2012.

[10] H. Wang, "A survey of maintenance policies of deteriorating systems," *European Journal of Operational Research*, vol. 139, no. 3, pp. 469–489, 2002.

[11] S. Alaswad and Y. Xiang, “A review on condition-based maintenance optimization models for stochastically deteriorating system,” *Reliability Engineering and System Safety*, vol. 157, pp. 54–63, 2017.

[12] A. K. S. Jardine, D. Lin, and D. Banjevic, “A review on machinery diagnostics and prognostics implementing condition-based maintenance,” *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[13] J. Iskandar, J. Moyne, S. Kommisetti, P. Hawkins, and M. Armacost, “Predictive maintenance in semiconductor manufacturing,” in *26th Annual SEMI Advanced Semiconductor Manufacturing Conference*, 2015, pp. 384–389.

[14] F. A. A. M. N. Carvalho, Thyago P.and Soares, V. Roberto, R. d. P. Francisco, J. P. Basto, and S. G. S. Alcalá, “A systematic literature review of machine learning methods applied to predictive maintenance,” *Computers & Industrial Engineering*, vol. 137, 2019.

[15] Y. Hu, X. Miao, Y. Si, E. Pan, and E. Zio, “Prognostics and health management: A review from the perspectives of design, development and decision,” *Reliability Engineering & System Safety*, vol. 217, p. 108063, 2022.

[16] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, “Prognostics and health management design for rotary machinery systems reviews, methodology, and applications,” *Mechanical Systems and Signal Processing*, vol. 42, no. 1-2, pp. 314–334, 2014.

[17] D. Wang, K.-L. Tsui, and Q. Miao, “Prognostics and health management: A review of vibration based bearing and gear health indicators,” *IEEE Access*, vol. 6, pp. 665–676, 2017.

[18] E. Zio, “Prognostics and health management (PHM): Where are we and where do we (need to) go in theory and practice,” *Reliability Engineering & System Safety*, vol. 218, p. 108119, 2022.

[19] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, “Machinery health prognostics: A systematic review from data acquisition to RUL prediction,” *Mechanical Systems and Signal Processing*, vol. 104, pp. 799 – 834, 2018.

[20] I. El-Thalji and E. Jantunen, “A summary of fault modelling and predictive health monitoring of rolling element bearings,” *Mechanical Systems and Signal Processing*, vol. 60, pp. 252–272, 2015.

[21] M. S. Kan, A. C. Tan, and J. Mathew, “A review on prognostic techniques for non-stationary and non-linear rotating systems,” *Mechanical Systems and Signal Processing*, vol. 62, pp. 1–20, 2015.

[22] R. Liu, B. Yang, E. Zio, and X. Chen, “Artificial intelligence for fault diagnosis of rotating machinery: A review,” *Mechanical Systems and Signal Processing*, vol. 108, pp. 33–47, 2018.

[23] "The impact of digital on unplanned downtime," GE Oil & Gas, 2016. [Online]. Available: https://www.ge.com/digital/sites/default/files/download_assets/ge-the-impact-of-digital-on-unplanned-downtime.pdf

[24] T. Zonta, C. André da Costa, R. d. R. Righi, M. José de Lima, E. Silveira da Trindade, and G. Pyng Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Computers & Industrial Engineering*, vol. 150, 2020.

[25] J. Moyne, Y. Qamsane, E. C. Balta, I. Kovalenko, J. Faris, K. Barton, and D. M. Tilbury, "A requirements driven digital twin framework: specification and opportunities," *IEEE Access*, vol. 8, pp. 107 781–107 801, 2020.

[26] S. Bachir, S. Tnani, J.-C. Trigeassou, and G. Champenois, "Diagnosis by parameter estimation of stator and rotor faults occurring in induction machines," *IEEE Transactions on industrial electronics*, vol. 53, no. 3, pp. 963–973, 2006.

[27] R. Doraiswami, C. P. Diduch, and J. Tang, "A new diagnostic model for identifying parametric faults," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 533–544, 2009.

[28] S. Yin and X. Zhu, "Intelligent particle filter and its application to fault detection of nonlinear system," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3852–3861, 2015.

[29] D. Zhang, A. D. Bailey, and D. Djurdjanovic, "Bayesian identification of hidden Markov models and their use for condition-based monitoring," *IEEE Transactions on Reliability*, vol. 65, no. 3, pp. 1471–1482, 2016.

[30] Y. Wen, J. Wu, and Y. Yuan, "Multiple-phase modeling of degradation signal for condition monitoring and remaining useful life prediction," *IEEE Transactions on Reliability*, vol. 66, no. 3, pp. 924–938, 2017.

[31] G. Liao, H. Yin, M. Chen, and Z. Lin, "Remaining useful life prediction for multi-phase deteriorating process based on wiener process," *Reliability Engineering & System Safety*, vol. 207, p. 107361, 2021.

[32] Z. Zhao, P. X. Liu, and J. Gao, "Model-based fault diagnosis methods for systems with stochastic process–a survey," *Neurocomputing*, 2022.

[33] M. Toothman, B. Braun, S. J. Bury, M. Dessauer, K. Henderson, R. Wright, D. M. Tilbury, J. Moyne, and K. Barton, "Trend-based repair quality assessment for industrial rotating equipment," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1675–1680, 2020.

[34] Y. Zhao, M. Toothman, J. Moyne, and K. Barton, "An adaptive modeling framework for bearing failure prediction," *Electronics*, vol. 11, no. 2, p. 257, 2022.

[35] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," *arXiv preprint arXiv:1912.07383*, 2019.

[36] A. Cachada, J. Barbosa, P. Leitão, C. A. Geraldes, L. Deusdado, J. Costa, C. Teixeira, J. Teixeira, A. H. Moreira, P. M. Moreira, and L. Romero, "Maintenance 4.0: Intelligent and predictive maintenance system architecture," in *2018 IEEE 23rd international conference on emerging technologies and factory automation (ETFA)*, vol. 1. IEEE, 2018, pp. 139–146.

[37] ISO, "Condition monitoring and diagnostics of machines - data processing, communication and presentation part 1: General guidelines," International Organization for Standardization, ISO 13374-1:2003, 2003.

[38] M. Lebold, K. Reichard, C. S. Byington, and R. Orsagh, "OSA-CBM architecture development with emphasis on XML implementations," in *Maintenance and Reliability Conference (MARCON)*, 2002, pp. 6–8.

[39] M. Toothman, B. Braun, S. J. Bury, M. Dessauer, K. Henderson, S. Phillips, Y. Ye, D. M. Tilbury, J. Moyne, and K. Barton, "A digital twin framework for mechanical system health state estimation," in *Modeling, Estimation, and Control Conference 2021*, 2021, pp. 1–7.

[40] M. Toothman, B. Braun, S. J. Bury, J. Moyne, D. M. Tilbury, Y. Ye, and K. Barton, "A digital twin framework for prognostics and health management," *Computers in Industry*, vol. 150, p. 103948, 2023.

[41] N. Z. Gebraeel, M. A. Lawley, R. Li, and J. K. Ryan, "Residual-life distributions from component degradation signals: A Bayesian approach," *IIE Transactions*, vol. 37, no. 6, pp. 543–557, 2005.

[42] A. Elwany and N. Gebraeel, "Real-time estimation of mean remaining life using sensor-based degradation models," *Journal of Manufacturing Science and Engineering*, vol. 131, no. 5, 09 2009.

[43] J. Coble and J. W. Hines, "Incorporating prior belief in the general path model: A comparison of information sources," *Nuclear Engineering and Technology*, vol. 46, no. 6, pp. 773–782, 2014.

[44] M. Toothman, B. Braun, S. J. Bury, J. Moyne, D. M. Tilbury, and K. Barton, "An adaptive state-based framework for fault prediction in rotating equipment," in *19th International Conference on Automation Science and Engineering*, 2023.

[45] M. Toothman, B. Braun, S. J. Bury, J. Moyne, D. M. Tilbury, Y. Ye, and K. Barton, "Overcoming challenges associated with developing industrial prognostics and health management solutions," *Sensors*, vol. 23, no. 8, p. 4009, 2023.

[46] J. Coble, P. Ramuhalli, L. Bond, J. Hines, and B. Ipadhyaya, "A review of prognostics and health management applications in nuclear power plants," *International Journal of Prognostics and Health Management*, 2015.

[47] D. Nguyen, M. Kefalas, K. Yang, A. Apostolidis, M. Olhofer, S. Limmer, and T. Bäck, "A review: Prognostics and health management in automotive and aerospace," *International Journal of Prognostics and Health Management*, vol. 10, no. 2, p. 35, 2019.

[48] M. Ahsan, S. Stoyanov, and C. Bailey, "Prognostics of automotive electronics with data driven approach: A review," in *2016 39th International Spring Seminar on Electronics Technology (ISSE)*. IEEE, 2016, pp. 279–284.

[49] I. J. Jóźwiak, "An introduction to the studies of reliability of systems using the weibull proportional hazards model," *Microelectronics Reliability*, vol. 37, no. 6, pp. 915–918, 1997.

[50] W. B. Nelson, *Applied life data analysis*. John Wiley & Sons, 2005.

[51] A. Cubillo, S. Perinpanayagam, and M. Esperon-Miguez, "A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery," *Advances in Mechanical Engineering*, vol. 8, pp. 1–21, 2016.

[52] P. Paris and F. Erdogan, "A critical analysis of crack propagation laws," *Journal of Basic Engineering*, vol. 85, pp. 528–533, 1963.

[53] F. Zhao, Z. Tian, and Y. Zeng, "Uncertainty quantification in gear remaining useful life prediction through an integrated prognostics method," *IEEE Transactions on Reliability*, vol. 62, pp. 146–159, 2013.

[54] Y. Li, S. Billington, C. Zhang, T. Kurfess, S. Danyluk, and S. Liang, "Adaptive prognostics for rolling element bearing condition," *Mechanical Systems and Signal Processing*, vol. 13, pp. 103–113, 1999.

[55] J. Qiu, C. Zhang, B. B. Seth, and S. Y. Liang, "Damage mechanics approach for bearing lifetime prognostics," *Mechanical Systems and Signal Processing*, vol. 16, pp. 817–829, 2002.

[56] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE Systems Journal*, vol. 13, pp. 2213–2227, 2019.

[57] C. Ferreira and G. Gonçalves, "Remaining useful life prediction and challenges: A literature review on the use of machine learning methods," *Journal of Manufacturing Systems*, vol. 63, pp. 550–562, 2022.

[58] M. J. Gómez, C. Castejón, and J. C. García-Prada, "Automatic condition monitoring system for crack detection in rotating machinery," *Reliability Engineering & System Safety*, vol. 152, pp. 239–247, 2016.

[59] A. Sadeghian, Z. Ye, and B. Wu, "Online detection of broken rotor bars in induction motors by wavelet packet decomposition and artificial neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, pp. 2253–2263, 2009.

[60] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, pp. 5990–5998, 2018.

[61] V. Muralidharan and V. Sugumaran, "A comparative study of naïve bayes classifier and bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis," *Applied Soft Computing*, vol. 12, no. 8, pp. 2023–2029, 2012.

[62] X. Liang, F. Duan, D. Mba, and B. Ian, "Centrifugal compressor diagnosis using kernel pca and fuzzy clustering," in *Asset Intelligence through Integration and Interoperability and Contemporary Vibration Engineering Technologies: Proceedings of the 12th World Congress on Engineering Asset Management and the 13th International Conference on Vibration Engineering and Technology of Machinery.* Springer, 2019, pp. 373–381.

[63] A. Hafaifa, F. Laaouad, and K. Laroussi, "Fuzzy logic approach applied to the surge detection and isolation in centrifugal compressor," *Automatic Control and Computer Sciences*, vol. 44, no. 1, pp. 53–59, 2010.

[64] C. J. Lu and W. O. Meeker, "Using degradation measures to estimate a time-to-failure distribution," *Technometrics*, vol. 35, no. 2, pp. 161–174, 1993.

[65] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *arXiv preprint arXiv:1801.01078*, 2017.

[66] D. Bruneo and F. De Vita, "On the use of lstm networks for predictive maintenance in smart industries," in *2019 IEEE international conference on smart computing (smartcomp).* IEEE, 2019, pp. 241–248.

[67] Y. Cheng, K. Hu, J. Wu, H. Zhu, and X. Shao, "A convolutional neural network based degradation indicator construction and health prognosis using bidirectional long short-term memory network for rolling bearings," *Advanced Engineering Informatics*, vol. 48, p. 101247, 2021.

[68] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.

[69] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[70] M. E. Cholette and D. Djurdjanovic, "Degradation modeling and monitoring of machines using operation-specific hidden Markov models," *IIE Transactions*, vol. 46, no. 10, pp. 1107–1123, 2014.

[71] S.-Z. Yu, "Hidden semi-Markov models," *Artificial intelligence*, vol. 174, no. 2, pp. 215–243, 2010.

[72] M. Dong and D. He, "A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology," *Mechanical systems and signal processing*, vol. 21, no. 5, pp. 2248–2266, 2007.

[73] Y. Peng and M. Dong, "A prognosis method using age-dependent hidden semi-Markov model for equipment health prediction," *Mechanical Systems and Signal Processing*, vol. 25, no. 1, pp. 237–252, 2011.

[74] S. Tafazoli and X. Sun, "Hybrid system state tracking and fault detection using particle filters," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 6, pp. 1078–1087, 2006.

[75] S. Langarica, C. Rüffelmacher, and F. Núñez, "An industrial internet application for real-time fault diagnosis in industrial motors," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 284–295, 2019.

[76] W. Yu, T. Dillon, F. Mostafa, W. Rahayu, and Y. Liu, "A global manufacturing big data ecosystem for fault detection in predictive maintenance," *IEEE Transactions on Industrial Electronics*, vol. 16, no. 1, pp. 183–192, 2020.

[77] R. Gao, L. Wang, R. Teti, D. Dornfeld, S. Kumara, M. Mori, and M. Helu, "Cloud-enabled prognosis for manufacturing," *CIRP Annals*, vol. 64, pp. 749–772, 2015.

[78] J. Wang, L. Zhang, L. Duan, and R. X. Gao, "A new paradigm of cloud-based predictive maintenance for intelligent manufacturing," *Journal of Intelligent Manufacturing*, vol. 28, pp. 1125–1137, 2017.

[79] M. Grieves and J. Vickers, *Transdisciplinary Perspectives on Complex Systems*. Springer, 2017.

[80] C. Cimino, E. Negri, and L. Fumagalli, "Review of digital twin applications in manufacturing," *Computers in Industry*, vol. 113, p. 103130, 2019.

[81] H. Boyes and T. Watson, "Digital twins: An analysis framework and open issues," *Computers in Industry*, vol. 143, p. 103763, 2022.

[82] I. Errandonea, S. Beltran, and S. Arrizabalaga, "Digital twin for maintenance: A literature review," *Computers in Industry*, vol. 123, 2020.

[83] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang, "Modeling, simulation, information technology & processing roadmap," *National Aeronautics and Space Administration*, vol. 32, no. 2012, pp. 1–38, 2012.

[84] J. Eyre, S. Hyde, D. Walker, S. Ojo, O. Hayes, R. Hartley, R. Scott, and J. Bray, "Untangling the requirements of a digital twin," *Advanced Manufacturing Research Centre. Technical Report. Available online: https://www. amrc. co.uk/files/document/406/1605271035_1604658922_AMRC_Digital_Twin_AW.pdf*, 2020.

[85] ISO, "Automation systems and integration - digital twin framework for manufacturing part 1: Overview and general principles," International Organization for Standardization, ISO 23247-1:2021, 2021.

[86] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in CPS-based production systems," *Procedia manufacturing*, vol. 11, pp. 939–948, 2017.

[87] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108 952–108 971, 2020.

[88] S. Lu and Y. Wen, "Semi-supervised condition monitoring and visualization of fused magnesium furnace," *IEEE Transactions on Automation Science and Engineering*, 2021.

[89] W. Luo, T. Hu, Y. Yee, C. Zhanga, and Y. Wei, "A hybrid predictive maintenance approach for CNC machine tool driven by digital twin," *Robotics and Computer Integrated Manufacturing*, vol. 65, 2020.

[90] J. Wang, L. Ye, R. X. Gao, C. Li, and L. Zhang, "Digital twin for rotating machinery fault diagnosis in smart manufacturing," *International Journal of Production Research*, vol. 57, no. 12, pp. 3920–3934, 2019.

[91] G. N. Schroeder, C. Steinmetz, R. N. Rodrigues, R. V. B. Henriques, A. Rettberg, and C. E. Pereira, "A methodology for digital twin modeling and deployment for industry 4.0," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 556–567, 2020.

[92] S. Cavalieri and M. G. Salafia, "A model for predictive maintenance based on asset administration shell," *Sensors*, vol. 20, no. 21, p. 6028, 2020.

[93] DIN, "Reference architecture model industrie 4.0 (rami4.0)," German Institute for Standardisation, Tech. Rep. 91345:2016-04, April 2016.

[94] P. I. 4.0, "Relationships between i4.0 components - composite components and smart production," Plattform Industrie 4.0, Working Paper, June 2017.

[95] N. Gebraeel and J. Pan, "Prognostic degradation models for computing and updating residual life distributions in a time-varying environment," *IEEE Transactions on Reliability*, vol. 57, no. 4, pp. 539–550, 2008.

[96] L. Bian and N. Gebraeel, "Stochastic methodology for prognostics under continuously varying environmental profiles," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 6, no. 3, pp. 260–270, 2013.

[97] R. Zhou, N. Serban, and N. Gebraeel, "Degradation-based residual life prediction under different environments," *The Annals of Applied Statistics*, vol. 8, no. 3, pp. 1671 – 1689, 2014.

[98] Q. Yang, X. Li, H. Cai, Y.-M. Hsu, J. Lee, C. H. Yang, Z. L. Li, and M. Y. Lin, "Fault prognosis of industrial robots in dynamic working regimes: Find degradation in variations," *Measurement*, vol. 173, p. 108545, 2021.

[99] H. Cai, J. Feng, W. Li, Y.-M. Hsu, and J. Lee, "Similarity-based particle filter for remaining useful life prediction with enhanced performance," *Applied Soft Computing*, vol. 94, p. 106474, 2020.

[100] Y. Wen, J. Wu, D. Das, and T.-L. B. Tseng, "Degradation modeling and rul prediction using wiener process subject to multiple change points and unit heterogeneity," *Reliability Engineering & System Safety*, vol. 176, pp. 113–124, 2018.

[101] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni, "Particle filter-based prognostics: Review, discussion and perspectives," *Mechanical Systems and Signal Processing*, vol. 72, pp. 2–31, 2016.

[102] E. Iglésis, N. Horri, J. Brusey, K. Dahia, and H. Piet-Lahanier, "Simultaneous actuator and sensor faults estimation for aircraft using a jump-Markov regularized particle filter," in *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2021, pp. 1–10.

[103] M. Mazzoleni, G. Maroni, Y. Maccarana, S. Formentin, and F. Previdi, "Fault detection in airliner electro-mechanical actuators via hybrid particle filtering," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2860–2865, 2017.

[104] Y.-s. Sun, X.-r. Ran, Y.-m. Li, G.-c. Zhang, and Y.-h. Zhang, "Thruster fault diagnosis method based on Gaussian particle filter for autonomous underwater vehicles," *International Journal of Naval Architecture and Ocean Engineering*, vol. 8, no. 3, pp. 243–251, 2016.

[105] H. Zheng, R. Wang, Y. Yang, J. Yin, Y. Li, Y. Li, and M. Xu, "Cross-domain fault diagnosis using knowledge transfer strategy: A review," *IEEE Access*, vol. 7, pp. 129 260–129 290, 2019.

[106] W. W. Royce, "Managing the development of large software systems: concepts and techniques," in *Proceedings of the 9th international conference on Software Engineering*, 1987, pp. 328–338.

[107] T. Bhuvaneswari and S. Prabaharan, "A survey on software development life cycle models," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 5, pp. 262–267, 2013.

[108] F. Lopez, M. Saez, Y. Shao, E. C. Balta, J. Moyne, Z. M. Mao, K. Barton, and D. M. Tilbury, "Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1885–1892, 2017.

[109] D. L. Nunez and M. Borsato, "An ontology-based model for prognostics and health management of machines," *Journal of Industrial Information Integration*, vol. 6, pp. 33–46, 2017.

[110] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, 1987.

[111] V. T. Tran and B.-S. Yang, "An intelligent condition-based maintenance platform for rotating machinery," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2977–2988, 2012.

[112] N. Li, Y. Lei, J. Lin, and S. X. Ding, "An improved exponential model for predicting remaining useful life of rolling element bearings," *IEEE Transactions on Industrial Electronics*, vol. 62, pp. 7762–7773, 2015.

[113] ISO, "Mechanical vibration — measurement and evaluation of machine vibration — part 1: General guidelines," International Organization for Standardization, ISO 20816-1:2016, 2016.

[114] X. Jin, Y. Sun, Z. Que, Y. Wang, and T. W. S. Chow, "Anomaly detection and fault prognosis for bearings," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 9, pp. 2046–2054, 2016.

[115] G. Casella and R. L. Berger, *Statistical Inference*. Duxbury, 2002.

[116] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[117] K. Murphy, "Hidden Markov model (HMM) toolbox for MATLAB." [Online]. Available: www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

[118] A. Ginart, I. Barlas, J. Goldin, and J. L. Dorrity, "Automated feature selection for embeddable prognostic and health monitoring (PHM) architectures," in *2006 IEEE Autotestcon*, 2006, pp. 195–201.

[119] Y. Shao and K. Nezu, "Prognosis of remaining bearing life using neural networks," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 214, no. 3, pp. 217–230, 2000.

[120] F. Yang, M. S. Habibullah, T. Zhang, Z. Xu, P. Lim, and S. Nadarajan, "Health index-based prognostics for remaining useful life predictions in electrical machines," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2633–2644, 2016.

[121] K. Liu, A. Chehade, and C. Song, "Optimize the signal quality of the composite health index via data fusion for degradation modeling and prognostic analysis," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 3, pp. 1504–1514, 2017.

[122] J. Lee, H. Qiu, G. Yu, J. Lin, and R. T. S. (2007), "Bearing data set," Center for Intelligent Maintenance Systems (IMS), University of Cincinnati., 2007. [Online]. Available: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository

[123] M. Li, J. Liu, J. Li, and B. U. Kim, "Bayesian modeling of multi-state hierarchical systems with multi-level information aggregation," *Reliability Engineering & System Safety*, vol. 124, pp. 158–164, 2014.

[124] D. Lee and R. Pan, "Predictive maintenance of complex system with multi-level reliability structure," *International Journal of Production Research*, vol. 55, no. 16, pp. 4785–4801, 2017.

[125] Y. Qamsane, J. Moyne, M. Toothman, I. Kovalenko, E. C. Balta, J. Faris, D. M. Tilbury, and K. Barton, "A methodology to develop and implement digital twin solutions for manufacturing systems," *IEEE Access*, 2021.

[126] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, and C. Varnier, "Pronostia: An experimental platform for bearings accelerated degradation tests," in *Proceedings of the IEEE International Conference on Prognostics and Health Management IEEE, Beijing, China*, 2012, pp. 23–25.

[127] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960.

[128] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[129] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, pp. 197–208, 2000.

[130] S. Roberts, "Control chart tests based on geometric moving averages," *Technometrics*, vol. 42, no. 1, pp. 97–101, 2000.

[131] D. A. Reynolds *et al.*, "Gaussian mixture models." *Encyclopedia of biometrics*, vol. 741, no. 659-663, 2009.

[132] R. J. Hyndman, "Computing and graphing highest density regions," *The American Statistician*, vol. 50, no. 2, pp. 120–126, May 1996.

[133] D. C. Montgomery, *Introduction to statistical quality control.* John Wiley & Sons, 2020.

[134] K. Goebel, A. Saxena, S. Saha, B. Saha, and J. Celaya, "Prognostic performance metrics," *Machine learning and knowledge discovery for engineering systems health management*, vol. 147, p. 20, 2011.

[135] R. Li, W. J. Verhagen, and R. Curran, "A systematic methodology for prognostic and health management system architecture definition," *Reliability Engineering & System Safety*, vol. 193, p. 106598, 2020.

[136] N. B. Ruparelia, "Software development lifecycle models," *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 3, pp. 8–13, 2010.

[137] ISO, "Systems and software engineering - vocabulary," International Organization for Standardization, ISO 24765:2017, 2017.

[138] K. Forsberg and H. Mooz, "The relationship of systems engineering to the project cycle," *Engineering Management Journal*, vol. 4, no. 3, pp. 36–43, 1992.

[139] N. D. Birrell and M. A. Ould, *A practical handbook for software development.* Cambridge University Press, 1988.

[140] B. W. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, no. 5, pp. 61–72, 1988.

[141] A. Abid, M. T. Khan, and J. Iqbal, "A review on fault detection and diagnosis techniques: basics and beyond," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3639–3664, 2021.

[142] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[143] R. Yan, F. Shen, C. Sun, and X. Chen, "Knowledge transfer for rotary machine fault diagnosis," *IEEE Sensors Journal*, vol. 20, no. 15, pp. 8374–8393, 2019.

[144] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE transactions on neural networks*, vol. 22, no. 2, pp. 199–210, 2010.

[145] A. J. Izenman, "Linear discriminant analysis," in *Modern multivariate statistical techniques.* Springer, 2013, pp. 237–280.

[146] "SAS/STAT 15.2 user's guide: The VARCLUS procedure," SAS Institute Inc., 2020. [Online]. Available: https://documentation.sas.com/doc/en/statug/15.2/statug_varclus_toc.htm

[147] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.