UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

A FAST-RUNNING FAILURE PROGNOSTIC ALGORITHM BASED ON A NON-HOMOGENEOUS MARKOV CHAIN

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

MAURICIO ESTEBAN GONZÁLEZ GUTIÉRREZ

PROFESOR GUÍA:
JORGE SILVA SANCHEZ

PROFESOR CO-GUÍA:
MARCOS ORCHARD CONCHA

MIEMBROS DE LA COMISIÓN:
KAMAL MEDJAHER
FELIPE TOBAR HENRÍQUEZ

Este trabajo ha sido parcialmente financiado por:
ANID Chile – ANID-PFCHA/MagísterNacional/2019-22191445

SANTIAGO DE CHILE
2021

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA
RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: **MAURICIO ESTEBAN GONZÁLEZ GUTIÉRREZ**
FECHA: JUNIO 2021
PROFESOR GUÍA: JORGE SILVA SÁNCHEZ

A FAST-RUNNING FAILURE PROGNOSTIC ALGORITHM BASED ON A NON-HOMOGENEOUS MARKOV CHAIN

Typically, model-based prognostic algorithms estimate the remaining-useful-life distribution by characterizing the probable system trajectories described through state-space models. Unfortunately, as the state dimension increases or the prognostic horizon enlarges, the computational time of such algorithms augments considerably, complicating their real-time execution.

To overcome this difficulty, this work proposes a paradigm change in model-based prognostic algorithms; instead of tracking the state-space trajectories, a Fast-Running Markov Chain-based Prognostic Algorithm (FRMC-PA) is proposed, capable of estimating the time-of-failure probability mass function directly. FRMC-PA is based on a two-state non-homogeneous discrete-time Markov chain, where state "0" describes the operative situation and state "1" represents a catastrophic failure event. FRMC-PA is composed of two stages: i) offline stage, which leverages historical data to train a regression model that maps the system variables to the transition probabilities of the binary-stochastic process; ii) online stage, which combines the regression model built previously and real-time observations to estimate the system's remaining useful life.

This method is validated using the case study of battery discharge. Results show that FRMC-PA can transfer most of the computational cost to the offline stage, achieving an online computational-time reduction of 99 % compared with a Monte-Carlo-based prognostic, without significantly sacrificing the prognostic accuracy.

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA

RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE INGENIERO CIVIL ELÉCTRICO

POR: **MAURICIO ESTEBAN GONZÁLEZ GUTIÉRREZ**

FECHA: JUNIO 2021
PROFESOR GUÍA: JORGE SILVA SÁNCHEZ

A FAST-RUNNING FAILURE PROGNOSTIC ALGORITHM BASED ON A NON-HOMOGENEOUS MARKOV CHAIN

Típicamente, los algoritmos de pronóstico basados en modelo estiman la vida útil del sistema utilizando una caracterización estocástica de las trayectorias del estado. Desafortunadamente, al aumentar la dimensión del estado o el horizonte de predicción, el tiempo computacional de dichos algoritmos crece considerablemente, complejizando su ejecución en tiempo real.

En lugar de seguir las trayectorias del estado, esta tesis propone el Fast-Running Markov Chain-based Prognostic Algorithm (FRMC-PA) para estimar directamente la función de probabilidad del tiempo de falla. FRMC-PA se basa en una cadena de Markov no-homogénea con dos estados: "0" (sistema operativo) y "1" (falla catastrófica). FRMC-PA comprende dos etapas: i) fuera de línea, que utiliza data histórica para entrenar un modelo de regresión que, evaluando variables del sistema, estima directamente las probabilidades de transición de la cadena; ii) en línea, la cual combina el modelo de regresión obtenido con observaciones en tiempo real para estimar la vida útil del sistema.

Este método es validado en un caso de estudio de descarga de baterías. Los resultados muestran que FRMC-PA puede transferir gran parte del costo computacional a la etapa fuera de línea, reduciéndolo en un 99 % durante la etapa en línea, comparado con una técnica de pronóstico de Monte-Carlo.

Para mis padres

Contents

1. Introduction **1**

1.1. Motivation . . . 1

1.2. Objectives . . . 2

1.2.1. Main objective . . . 2

1.2.2. Specific objectives . . . 2

1.3. Hypotheses . . . 3

1.4. Structure . . . 3

2. Background and Literature Review **5**

2.1. Failure Prognostics . . . 5

2.1.1. Particle-filtering-based failure prognostics . . . 7

2.1.2. Markov chain-based failure prognostics . . . 10

3. Markov Chain Representation for Failure Prognostics **11**

3.1. Characterization of $\tau^L(\omega)$ in Eq. (3.7) . . . 15

3.2. Characterizing the transition probabilities of $\{\Theta_k^L(\omega)\}_{k\geq 0}$ . . . 16

4. Fast-Running Markov Chain-based Prognostic Algorithm **18**

4.1. Offline learning stage . . . 19

4.2. Online evaluation stage . . . 21

5. Synthetic Example **24**

5.1. Obtaining the transition probabilities of $\{\Theta_k(\omega)\}$ [FRMC-PA steps A1-A5] . . . 25

5.2. Training φ [FRMC-PA steps A6-A7] . . . 26

6. Experimental Validation **31**

6.1. Offline stage . . . 33

6.1.1. Learning the degradation dynamics . . . 33

6.1.2. Obtaining the transition probabilities of $\{\Theta_k(\omega)\}$ [FRMC-PA steps A1-A5] . . . 35

6.1.3. Training φ [FRMC-PA steps A6-A7] 37
6.1.4. Analysis of the offline stage computational burden 39
6.2. Online stage 39
6.2.1. Analysis of the online stage computational burden 40

7. Conclusions **45**
7.1. Future Work 45

Bibliography **47**

A. Discrete-time Markov Chains **53**

B. Proof of Propositions **56**
B.1. Proof of Proposition 1 56
B.2. Proof of Proposition 2 58
B.3. Proof of Proposition 4 59

C. Algorithmic Solutions to Compute Probability Distributions **60**

List of Tables

6.1. Performance of the GRU model $\hat{g}$ in terms of the MAPE index. This index is evaluated in the train set, the test set and the prognostic set. 34
6.3. Comparison of the time of failure distributions obtained using Monte Carlo simulation (MC) and our proposal (OP), considering different values of λ . . . 43

List of Figures

2.1. Illustration of the failure prognostic problem. 6
2.2. Illustration of the particle filter algorithm. 9
3.1. Illustration of the statistical dependencies between the system state stochastic process, the observation variable, the failure variable, and the time of failure random variable. 13
3.2. Diagram that describes the transition probabilities of the Markov chain $\{\Theta_k^L(\omega)\}_{k\geq 0}$ between time instants $k-1$ and k. 14
3.3. General scheme of the reduced representation and its use. 17
4.1. Graphical abstract of the proposed FRMC prognostic algorithm. 22
5.1. Evolution of the system state in the synthetic example, considering the mean and double the standard deviation. 25
5.2. Prior and posterior sequences in the synthetic example. The posterior sequence considered a measurement at $k = 1$ given by $y_1 = 0.4$. 26
5.3. Prior and posterior time of failure distribution in the synthetic example. The posterior distribution considered a measurement at $k = 1$ given by $y_1 = 0.4$. . . 27
5.4. Evolution of the loss in the synthetic example during the training process. . . 29
5.5. Performance of the GRU φ in the synthetic example, considering the prior distribution of the state from Figure 5.1 at $k_0 = 15$, and a measurement $y_{15} = 0.7$. Prior, posterior and predicted p_k^L sequence. 29
5.6. Performance of the GRU φ in the synthetic example, considering the prior distribution of the state from Figure 5.1 at $k_0 = 15$, and a measurement $y_{15} = 0.7$. Prior, posterior and predicted ToF distribution. φ predicts the transition probabilities in Figure 5.5, which are used to compute tho predicted ToF distribution using Eq. (3.12). 30
6.1. Measurements of current and voltage collected throughout the five discharge cycles in the electric bike. Each of the zones #1-5 corresponds to a discharge cycle. At the beginning of each, the battery was fully recharged. 32
6.2. Operational data collected during usage of the electric bike, used to train and test the GRU model $\hat{g}$. The predicted voltage values are shown, as well as the absolute error between the real and predicted voltages. 34

6.3. Illustration of the particle propagation for the FRMC-PA step A3 in the battery discharge case study. 36
6.4. Transition probabilities $p^L_{k|t_p}$ in the battery discharge case study, considering an initial condition $E_{t_p} = 1031666.5[J]$ and a measurement of $V_{t_p} = 34.51[V]$. . . . 37
6.5. Alpha-Lambda performance index for the prognostic algorithm. $\alpha = 15\,\%$. $\lambda = (t - T_p)/(T_{failure} - T_p)$. The expected RUL and the intervals of confidence of $30\,\%$ and $80\,\%$ were added. 40
6.6. Empirical ToF distribution obtained with the Monte-Carlo-based prognostic module for different numbers of simulations (N_r). 41
6.7. Analysis of the number of simulations for the Monte-Carlo-based prognostic module using the KL divergence. While, in general, the quality of the approximation increases with N_r, executing more than 1000 simulations does not show significant improvements. 42
6.8. Comparison of the ToF probability distributions obtained with our proposal (FRMC) and a Monte-Carlo-based prognostics (MC), considering different values of λ. The ground truth value of the time of failure (T_f), the mean value obtained with the Monte-Carlo prognostic module ($Mean_{MC}$), and the mean value obtained with our proposal ($Mean_{FRMC}$) are also illustrated. 43

Chapter 1

Introduction

1.1. Motivation

In the engineering discipline of Prognostics Health Management (PHM), there is a clear distinction between a fault (abnormal conditions in which a system is still operative) and a catastrophic failure (which implies the inoperability of a system). In this regard, Fault Diagnostic and Failure Prognostic (FDP) algorithms can enhance the safety of a system by detecting and isolating incipient faults (fault diagnostics) and subsequently forecasting the progression of these faults to the point of catastrophic failure (failure prognostics) [1]. Failure prognostic algorithms evaluate the reliability of a system in its current life cycle conditions, predicting the time at which a system or a component will no longer perform its intended function. This time is typically known as Remaining Useful Life (RUL), End-of-Life (EoL), Time-of-Failure (ToF), and Time-to-Failure (TtF) [2, 3].

One key challenge in the design and implementation of prognostic algorithms is to reduce the associated computational burden. For instance, real-time execution is desirable, even mandatory, for mission re-planning in Unmanned Aerial Mobility (UAM) [4, 5, 6]. Unfortunately, these systems often have limited computational resources on-board. Therefore, for such applications, the main goal of prognostic designers is to develop algorithms capable of computing effective predictions within a reasonable computational time period. Research in the field of real-time prognostic algorithms has explored a great variety of approaches. One of the most recognized efforts in this regard is presented in [7], where authors proposed a particle filter (PF) based algorithm for failure prognostics, providing the means to estimate the ToF Probability Density Function (PDF).

PF-based prognostic approaches are considered as the state-of-the-art for model-based prognostics by many PHM researchers [8]. Most of these works show promising results in

terms of prediction capability, but the computational burden is not treated as a priority [9]. Indeed, as particle-filtering-based failure prognostic methods intend to characterize the evolution of the state vector over extended time periods, the computational complexity of the algorithm increases significantly with the dimension of the state vector and the length of the prognostic horizon.

To address this issue, this work proposes a novel prognostic strategy based on a fundamental attribute of the problem. Given that catastrophic failures are binary stochastic processes in nature (where the system can only fail catastrophically once), it is reasonable to ask if it is necessary to track the full state of the system. In this regard, it is conjectured that there is a sufficient representation of the problem that provides a minimal description of the failure state. Although this simplified representation can lead to some mismatches in the prediction, it could achieve a significant reduction in the dimension of the relevant variable, and thus in the computational cost of calculating the ToF distribution.

1.2. Objectives

1.2.1. Main objective

This research effort aims to design, implement and validate a novel prognostic algorithm – called Fast-Running Markov Chain-based Prognostic Algorithm [FRMC-PA]– that alleviates the computational burden associated with its real-time execution. The proposed algorithm is comprised of two stages, and it is based on a reduced representation of a state-space degradation model. This reduced representation corresponds to a non-homogeneous discrete-time Markov chain model with two states: "0", to indicate that the system has not failed (it is faulty and degrading), and "1", to indicate that the system has already failed catastrophically.

1.2.2. Specific objectives

The specific objectives of this work are the following:

1. To represent a high-dimensional state-space degradation model (which characterizes a degradation process at a system level) with a non-homogeneous two-state discrete-time Hidden Markov Model (HMM), where the dimension of the state vector is significantly reduced.
2. To formulate analytic expressions that link both the state-space model and the Markov

chain representation. In particular, to derive a simple expression to obtain the ToF distribution based on the transition probabilities of the non-homogeneous Markov chain.

3. To design the FRMC-PA offline stage, whose purpose is to train the Markov chain model, obtaining a regression model from system variables to transition probabilities of the binary process. For this purpose, Machine Learning algorithms are used and tested to obtain a sequence of posterior estimates for transition probabilities, conditional on a given usage profile for the asset, measurements, and a failure threshold.

4. To design the FRMC-PA online stage, which aims to use the Markov chain and the obtained regression model to estimate the probability distribution of the RUL of the system, conditional on new measurements.

5. To test the proposed algorithm in a battery discharge case study, comparing its performance with a Monte-Carlo-based prognostic technique.

1.3. Hypotheses

This work aims at testing the following hypotheses:

1. Given a state-space model of a time-invariant non-regenerative degradation process, a two-state non-homogeneous discrete-time Markov chain is a sufficient representation of the process if the aim is to characterize the Time-of-Failure.

2. This reduced representation provides a way to estimate the Time-of-Failure probability distribution.

3. It is possible to design an online-efficient prognostic algorithm based on the sufficient representation of the degradation process.

1.4. Structure

This thesis is organized as follows. Chapter II presents the literature review regarding failure prognostics based on particle filters and Markov chains. Chapter III develops the proposed Markov model and explains the computation of the Time-of-Failure distribution based on this model. In Chapter IV, the FRMC-PA is presented, including the offline and the online stages. Chapter V includes a synthetic example, where the offline stage is illustrated. Chapter VI presents the experimental validation of the method in a case study of battery

discharge, explaining both the offline and online stages. Finally, Chapter VII summarizes the work and discusses some future extensions.

Chapter 2

Background and Literature Review

2.1. Failure Prognostics

Failure prognostics aims to predict a future time when a dynamical system can present a catastrophic failure, meaning that it will no longer perform its intended function [1]. The importance of this problem arises from some areas of applications, including to prevent the excessive expenditure on maintenance of equipment or components, to avoid the loss of production in industry due to the detention of machinery, and to reduce the risk of loss of human life (failure in transport systems, medical equipment, among others) [10].

More precisely, failure prognostics consist on the extrapolation of system health indicators into the future, providing information about the risk of catastrophic failures and helping to take preventive measures and maximize the system performance within a given prediction horizon. Figure 2.1 illustrates graphically this problem, where X_k corresponds to the state-space variable and Y_k to the observation variable. At time t_p, after a fault is detected and diagnosed, the posterior PDF for X_{t_p} –given measurements until t_p– is estimated and propagated over time using the system model. A failure is detected when the system state reaches a failure set (or hazard zone) L, typically characterized by a threshold representing a maximum or minimum admissible value of the state X_k. This defines the ToF variable, τ^L, which is associated to the hazard zone L. Regardless of the application, the objective of any failure prognostic algorithm is to estimate the probability distribution of the ToF τ^L [10].

ToF estimation is relevant as it provides information about the operational continuity of a system, i.e., it allows knowing for how much time the system can be operated before incurring a failure that could be catastrophic. Given the complexity of industrial systems, in general it is not possible to calculate the ToF with arbitrary precision, and thus, it is relevant to properly characterize the uncertainty associated with ToF estimates [11]. Accor-

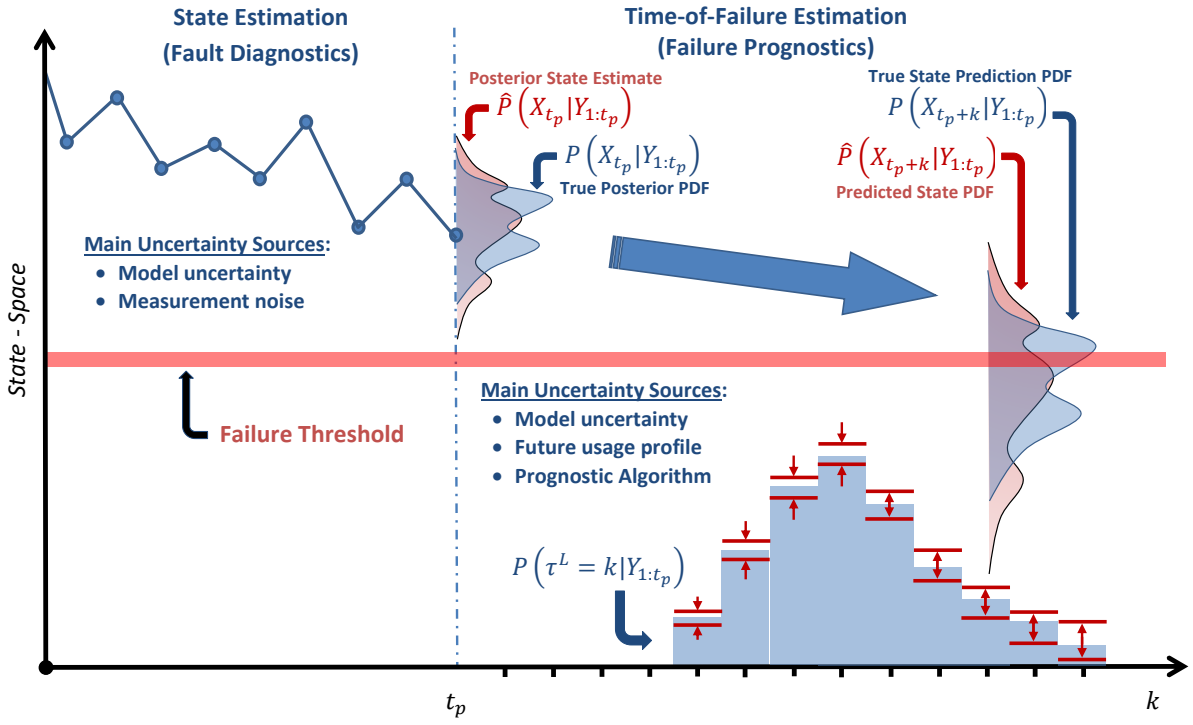

Figure 2.1: The Failure Prognostics Problem. Prognostics are executed at time t_p, after a fault is diagnosed. X_k corresponds to the system state variable. Y_k is the observation variable. The posterior PDF for X_{t_p} –given measurements until t_p– is estimated and propagated over time using the system model. A failure is detected when the system state reaches a failure set L, typically characterized by a threshold representing a maximum or minimum admissible value of the state X_k. This defines the ToF variable, τ^L, which is associated to the hazard zone L. The objective of any failure prognostic algorithm is to estimate the probability distribution of τ^L. Figure extracted and adapted from [10].

ding to [12], ToF prediction methods can be broadly classified in two groups: offline and online schemes. Offline methods typically utilize computationally expensive algorithms that offer highly accurate (or precise) results. In contrast, online methods should offer a delicate balance between efficiency and efficacy. In [12] some examples of offline methodologies are mentioned for the characterization of uncertainty related to crack growth dynamics [13, 14], structural damage [15, 16], electronics [17], and mechanical bearings [18]. Examples for the implementation of online prognostic and RUL estimation schemes can be found in [7, 19]. Also, various efforts aimed at quantifying the uncertainty associated with ToF estimates are found for the problem of energetic autonomy and capacity degradation of lithium-ion batteries [20, 21, 22, 23], and degradation of pneumatic valves [24]. Methodologies inspired on efficient sampling techniques [25] and analytical methods [26] have also been explored.

2.1.1. Particle-filtering-based failure prognostics

Failure prognostic algorithms typically use information provided by Bayesian filtering modules to determine a reasonable initial condition for the long-term predictions needed to characterize the ToF distribution adequately. Bayesian filters are able to fuse prior information of the system (for example, the degradation model structure) with real-time observations acquired once the fault has been diagnosed [27]. However, it is important to note that Bayesian filters are not used for prognostics but for state estimation [8] since prognostic algorithms need to characterize future sequences of state vector PDFs [9], assuming no new measurements have been acquired. Thus, their implementation implies challenges that are beyond the typical Bayesian filtering schemes.

In this regard, particle filters are a class of Bayesian filtering algorithms that has been widely accepted within the PHM community to tackle the failure prognostic problem. Moreover, PF-based prognostic approaches are considered as the state-of-the-art for model-based failure prognostics by many PHM researchers [8]. As a result, they have motivated an extensive number of publications in different applications [10] such as State-of-Charge and State-of-Health prognostics of batteries [22, 28, 21, 29, 30], prediction of crack faults [31], analysis of haul trucks failures based on the oil total base number [32], prediction of failures on analog electronic circuits [33], analysis of the power degradation phenomena in proton exchange membrane fuel cells [34, 35, 36, 37], design of fault-tolerant components [38], electrical machines failure prediction [39], micro-electro-mechanical systems failure prognostics [40], wind turbines shaft bearing degradation tracking [41], diesel motors health prediction [42], RUL prediction for automated machines [43], among others [44, 45, 46].

A particle-filtering-based prognostic algorithm is a method for future uncertainty characterization that uses sequential Monte-Carlo techniques to obtain a state posterior PDF during the filtering stage. These algorithms use a particle population to represent the state PDF and provide a way to propagate the particles to the future to characterize the evolution of the state distribution within a prediction horizon [9]. Formally, particle filter algorithms are designed to obtain samples sequentially from a target state probability distribution $P(X_{0:P_h})$ [47]. This algorithms generate a set of $N_p \gg 1$ weighted particles $\{w_k^i, x_k^i\}_{k=0\ldots P_h}^{i=1\ldots N_p}$, with $w_k^i \geq 0$ and $\sum_{i=1}^{N_p} w_k^i = 1$, $\forall k \geq 0$, satisfying that [22]:

$$\sum_{i=1}^{N_p} w_k^i \psi_k\left(x_k^i\right) \xrightarrow[N_p \to \infty]{} \int \psi_k\left(x_k\right) P\left(x_k\right) dx_k \quad , \quad \forall k \in \{0, \ldots, P_h\}, \tag{2.1}$$

in probability, where ψ_k is any P-integrable function. In the most basic PF implementation

–the sequential importance sampling [47]–, these particles are propagated through time using the prior state transition PDF $P(x_k^i|x_{0:k-1}^i)$, given by the degradation model dynamics.

On the other hand, the weights update and propagation depend on the availability of measurements at a particular time. Suppose that there are observations until time $t_p \leq P_h$, namely $\{y_k\}_{k=0,\ldots,t_p}$. For $0 \leq k \leq t_p$, it is possible to evaluate the weight w_k^i based on the measurement likelihood $P(y_k|x_{0:k}^i)$ and the previous weight w_{k-1}^i in the form $w_k^i \propto w_{k-1}^i \cdot P(y_k|x_{0:k}^i)$, $\forall i = 1, \ldots, N_p$. In contrast, for $t_p < k \leq P_h$ there are no measurements available, but it is still important to propagate weights for the prognostic task. In this case, the future weights are inherited from the filtering stage in the form $w_k^i = w_{k-1}^i$, $\forall i = 1, \ldots, N_p$.

Based on this, it is possible to obtain an empirical representation of the state distribution [27]:

$$\hat{P}_{N_p}(X_k = x) \approx \sum_{i=1}^{N_p} w_k^i \cdot \delta\left(x - x_k^i\right) \quad , \quad \forall k \in \{0, \ldots, P_h\}, \tag{2.2}$$

where $\delta(\cdot)$ is the Dirac delta function. The weight updating and particle propagation processes are schematized in Figure 2.2

Focusing on the prognostic task, the objective is to estimate the ToF probability distribution based on these empirical state distributions and the system hazard zone L. Considering that the ToF is the first time when the system state reaches the hazard zone, it is possible to approximate the ToF distribution using the expression [7]:

$$P\left(\tau^L = k\right) \approx \sum_{i=1}^{N_p} w_k^i \cdot P(\tau^L = k|X = x_{0:k}^i) \quad , \quad \forall k \in \{0, \ldots, P_h\}, \tag{2.3}$$

where $P(\tau^L = k|X = x_{0:k}^i) = 1$ if and only if the trajectory $x_{0:k}^i$ reaches L at time k (not before). Otherwise, $P(\tau^L = k|X = x_{0:k}^i) = 0$.

As stated before, the filtering stage provides the initial condition for the prediction stage. For each new measurement acquired in real-time, a new filtering step is applied and, hence, it is possible to update the prognostic stage. This procedure may improve the estimation of the ToF distribution because of the better characterization of the posterior state PDF. On the other hand, updating the prognostics every time a new measurement is available requires high computational resources to be executed [9].

To address this computational burden issue, some authors have explored variations of PF-based prognostic algorithms. Sbarufatti et al. [48] propose an algorithm that combines PF and radial basis function neural networks. Chang et al. [49] explore a hybrid prognostic

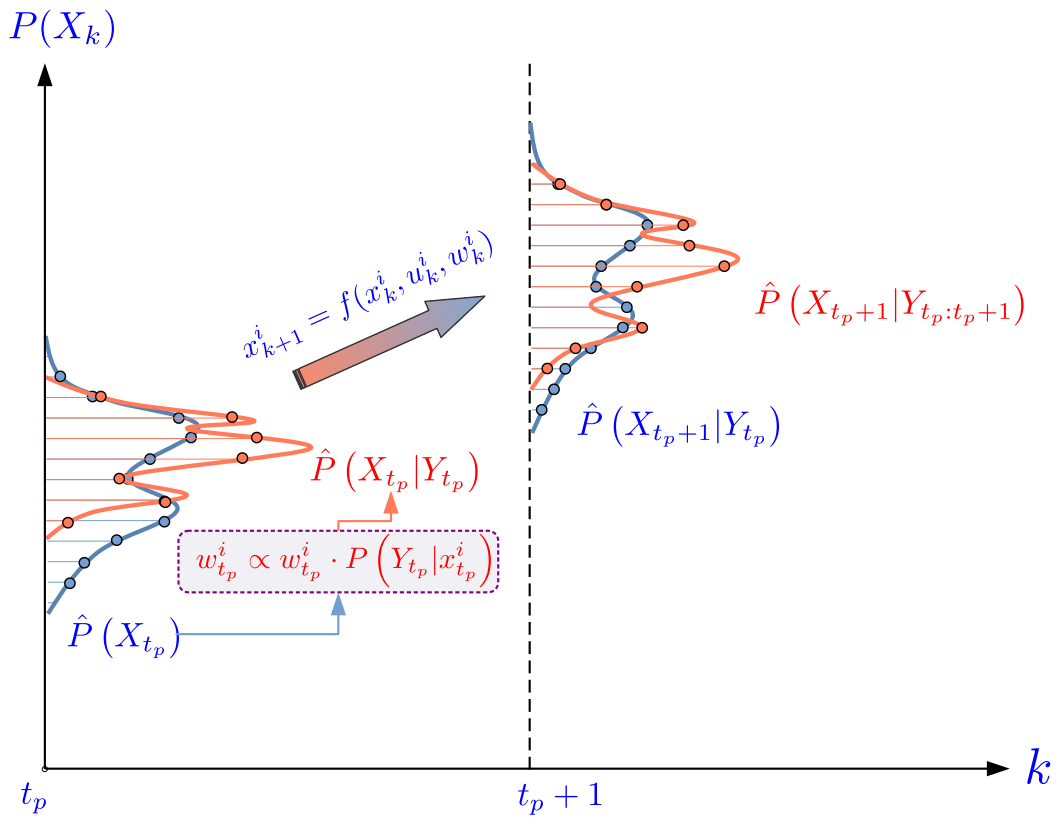

Figure 2.2: Illustration of the particle filter algorithm. A particle population represents the prior state distribution. Given measurements, weights are updated based on the likelihood of the particles. Then, the particles are propagated through time utilizing the system dynamics. If new new measurements are available, the weights are updated again, and so on.

scheme with the capability of uncertainty assessment that combines PF and Relevance Vector Machines (RVMs). Another interesting approach to deal with real-time prognostics was introduced in [50] and then extended in [51, 52]: it computes prognostic through Lebesgue sampling-based procedure, wherein prediction steps are discretized in the state space instead of the time axis. Although significant computation burden can be saved [51, 52], this approach requires a dynamic model in Lebesgue space that is extremely difficult to identify in the case of a complex engineering system. On the other hand, Rozas *et al.* [9] proposed a method based on a time-variant prognostic update, in which the sample time is decided according to a novel metric to evaluate the performance of prognostic algorithms in real-time.

While the computational savings obtained by the mentioned methods are valuable, it is important to note that they still require a powerful computational platform to execute their predictive algorithms in real-time.

2.1.2. Markov chain-based failure prognostics

Considering the objectives of this work, it is important to note that other research efforts have incorporated Markov chain models in failure prognostics. Please refer to Appendix A to review some definitions and properties regarding Markov chains.

In [22], a two-state Markov chain characterizes the discharge profiles of a battery, where one state represents a high energy consumption profile and the other one a low energy consumption profile. Maximum likelihood estimators were used to estimate the transition matrix. Geramifard *et al.* [53] used a stationary hidden Markov model (HMM) to explore a representation for the failure, establishing an explicit relationship between the hidden states values and the actual health states of the system. This relationship is then exploited for formulation and parameter estimation in the proposed approach. A Mixture of Gaussians Hidden Markov Model (MoG-HMM) is utilized in [54] to model the degradation phenomenon of bearings. This model is trained with features extracted from a Wavelet packet decomposition (WPD) applied to raw data. On the other hand, Tang *et al.* [55] proposed an approach to estimate the RUL of a degrading system under dynamic operational conditions, which are represented with a discrete-time Markov chain. In addition, the RUL prediction problem is formulated as a semi-Markov decision process framework. Chiachío *et al.* [56] use a multi-state Markov chain to represent the damage sates, with an invariant approximation of the transition probabilities. They tested the proposed methodology in a case study about stochastic fatigue damage evolution in metallic materials. Finally, in [57] a stochastic working mode is modeled through a flexible two-state semi-Markov model with phase-type distributed interval times. This is utilized to predict the RUL in batteries, where the system performance degrades with usage and recovers in storage.

In summary, while other works have used Markov chain models to represent the health states of a system, most of them use a multi-state homogeneous process (or a static approximation) for this representation. In contrast, the proposed approach utilizes a non-homogeneous two-state Markov chain to capture the failure dynamics, enhancing the model, and then providing an online-efficient prognostic algorithm to update the ToF distribution given new online measurements. This algorithm allocates most of the computational burden to the offline stage, leaving few computations to the online stage. Therefore, this approach has the potential to alleviate the online computational cost without substantially sacrificing the prognostic quality.

Chapter 3

Markov Chain Representation for Failure Prognostics

One of the main objectives of this work is to represent the degradation process of a dynamic system with a simpler model that reduces the dimension of the relevant variable. Until now, researchers from the PHM community have approached the failure prognostic problem by focusing their attention on a proper characterization of the future system state sequence. Although this approach is theoretically correct, it becomes impractical, even intractable, in the context where the dimension of the state vector or the prognostic horizon is large. Under the observation that the objective is to estimate the ToF, this work aims to obtain a representation independent of the system's complexity.

To develop the proposed methodology, the following random vectors are defined. Let $(\Omega, \mathcal{F}, P)$ be a probability space. Then:

$$\begin{aligned} X_k &: (\Omega, \mathcal{F}, P) \to (\mathbb{R}^{n_x}, \mathcal{B}(\mathbb{R}^{n_x})), \\ U_k &: (\Omega, \mathcal{F}, P) \to (\mathbb{R}^{n_u}, \mathcal{B}(\mathbb{R}^{n_u})), \\ Y_k &: (\Omega, \mathcal{F}, P) \to (\mathbb{R}^{n_y}, \mathcal{B}(\mathbb{R}^{n_y})), \end{aligned}$$

where $\mathcal{B}(\mathbb{R}^n)$ is the Borel σ-algebra on $\mathbb{R}^n$. These random vectors represent the system state variable, the exogenous input, and the observation variable, respectively, at some discrete instant $k \in \mathbb{N}$. Also, consider that the high-dimensional state-space model that characterizes the degradation process at the system level is given by:

$$X_{k+1}(\omega) = f(X_k(\omega), U_k(\omega), W_k(\omega)), \tag{3.1}$$

$$Y_k(\omega) = g(X_k(\omega), V_k(\omega)) \quad , \forall k \in \mathbb{N}. \tag{3.2}$$

In Eqs. (3.1) and (3.2), $f(\cdot)$ and $g(\cdot)$ are time-invariant functions, and $W_k(\omega) \in \mathbb{R}^{n_x}$, $V_k(\omega) \in \mathbb{R}^{n_y}$ are random vectors representing the process model uncertainty and the observation noise at time k, respectively. Consider that $W_k(\omega)$ and $V_k(\omega)$ are independent of each other and independent of $(X_k(\omega), Y_k(\omega), U_k(\omega))$. Also, it is assumed that $W_k(\omega)$ and $W_j(\omega)$ are independent if $k \neq j$, and the same goes to $V_k(\omega)$. Eqs. (3.1) and (3.2) generate the discrete-time stochastic processes $\{X_k(\omega)\}_{k\geq 0} \subset \mathbb{R}^{n_x}$ and $\{Y_k(\omega)\}_{k\geq 0} \subset \mathbb{R}^{n_y}$, which depend on the input random sequence $\{U_k(\omega)\}_{k\geq 0} \subset \mathbb{R}^{n_u}$. The sequence $\{U_k(\omega)\}_{k\geq 0}$ is known as the *usage profile* or the *exogenous input*, and it represents the manner the operator uses the system.

Next, the *failure event* binary random variable at each discrete instant k is defined by:

$$\Theta_k^L(\omega) : (\Omega, \mathcal{F}, P) \rightarrow (\{0\ ,\ 1\}, \mathcal{P}(\{0\ ,\ 1\})). \tag{3.3}$$

$\Theta_k^L(\omega)$ represents a failure event in the form that the system state $X_k(\omega) \in L \subseteq \mathbb{R}^{n_x}$. In other words, L corresponds to the system failure set that parameterizes the *hazard zone* of the problem. If the system state $X_k(\omega)$ has reached L, the state-space model indicates that the system has failed catastrophically up to time k, and then $\Theta_k^L(\omega) = 1$. On the other hand, $\Theta_k^L(\omega) = 0$ if the system model is degrading but has not failed catastrophically at time k. Note that this generates the discrete-time random process $\{\Theta_k^L(\omega)\}_{k\geq 0} \subset \{0,\, 1\}$. If it is assumed that the system is not catastrophically failed at $k-1$, the following conditional probabilities are known:

$$\begin{aligned} P(\Theta_k^L(\omega) = 1 | X_k(\omega) \in L, \Theta_{k-1}^L(\omega) = 0) = 1, \\ P(\Theta_k^L(\omega) = 0 | X_k(\omega) \in L^c, \Theta_{k-1}^L(\omega) = 0) = 1. \end{aligned} \tag{3.4}$$

Modeling the catastrophic nature of the failure set, 1 is considered an absorbent state of the model. This means that if the system reaches the failure state, the system cannot recover back from the failure state. Therefore:

$$P(\Theta_{k+1}^L(\omega) = 0 | \Theta_k^L(\omega) = 1) = 0 \quad , \forall k \geq 0. \tag{3.5}$$

Importantly, $\forall k \geq 1$, the *prior transition probability* at time k (given that no failure has happened before) is defined by:

$$p_k^L \triangleq P(\Theta_k^L(\omega) = 1 | \Theta_0^L(\omega) = 0, \ldots, \Theta_{k-1}^L(\omega) = 0). \tag{3.6}$$

It is assumed that the system has not failed at time $k = 0$, i.e., $p_0^L \triangleq 0$.

Finally, the Time-of-Failure is defined as a random variable $\tau^L(\omega) : (\Omega, \mathcal{F}, P) \rightarrow (\mathbb{N})$. This object measures the first discrete instant at which a failure occurs in the process $\{X_k(\omega)\}_{k\geq 0}$,

which is typically called the *first passage time* [11]. As the failure state is absorbent, then $\Theta_k^L(\omega) = 1\,, \forall k \geq \tau^L(\omega)$. Therefore:

$$\tau^L(\omega) \triangleq \min\{k \geq 1 : \Theta_k^L(\omega) = 1\}. \tag{3.7}$$

From Eq. (3.7), $\tau^L(\omega)$ is a function of the stochastic process $\{\Theta_k^L(\omega)\}_{k\geq 0}$ and behaves as a stopping time variable [58] (see Appendix A).

In summary, there are 4 layers that are illustrated in Figure 3.1. This diagram shows the statistical relationships between the stochastic processes that have been defined so far.

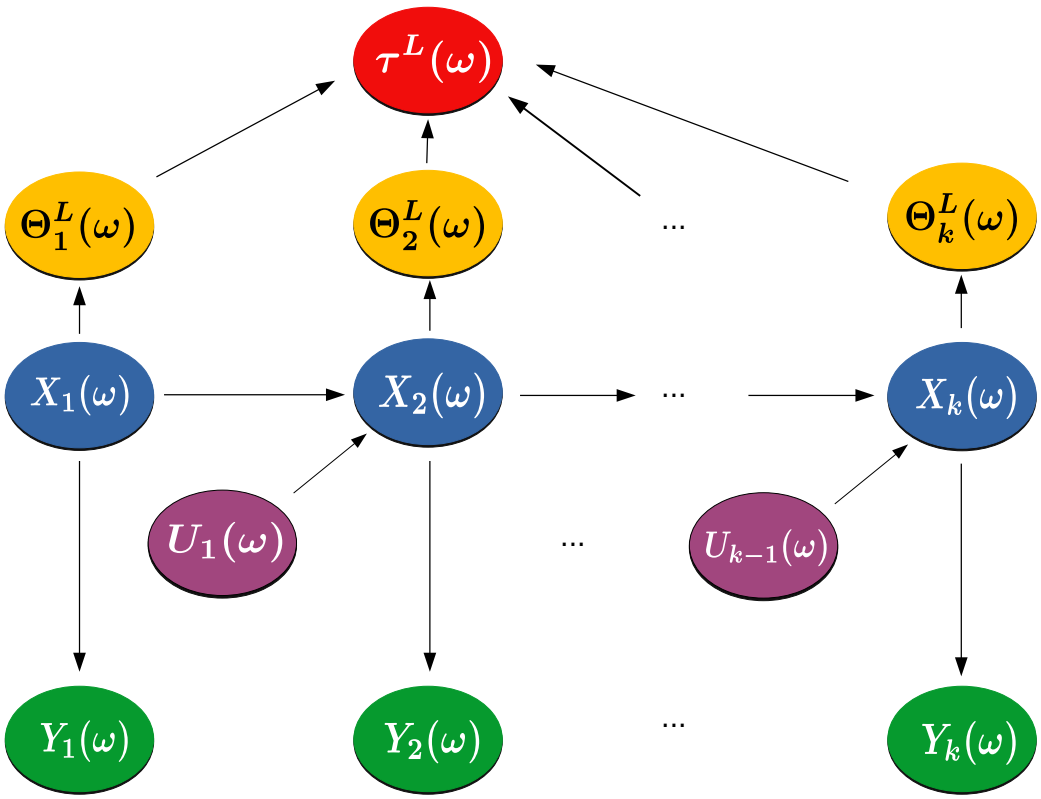

Figure 3.1: Illustration of statistical dependencies between the system state stochastic process (see Eq.(3.1)), the observation variable (Eq. (3.2)), the failure variable (Eq. (3.4)), and the time of failure random variable (Eq.(3.7)).

The following probabilistic relationships between layers are established:

- $X_k(\omega) \leftrightarrow Y_k(\omega)$: $\forall k \geq 0$ and $\forall x_k \in \mathbb{R}^{n_x}$, $y_k \in \mathbb{R}^{n_y}$. From Eq. (3.2):

$$P(Y_k(\omega) = y_k | X_k(\omega) = x_k) = P(V_k(\omega) = y_k - g(x_k)). \tag{3.8}$$

- $X_k(\omega) \leftrightarrow \Theta_k(\omega)$: This relationship is given by Eq. (3.4).

- $X_{k+1}(\omega) \leftrightarrow X_k(\omega)$, $U_k(\omega)$: $\forall k \geq 0$, $\forall x_k, x_{k+1} \in \mathbb{R}^{n_x}$, and $\forall u_k \in \mathbb{R}^{n_u}$. From Eq. (3.1):

$$P(X_{k+1}(\omega) = x_{k+1} | X_k(\omega) = x_k, U_k(\omega) = u_k) = P(W_k(\omega) = x_{k+1} - f(x_k, u_k)). \tag{3.9}$$

Below, a relevant property of the stochastic process $\{\Theta_k^L(\omega)\}_{k\geq 0}$ is presented.

PROPOSITION 1 The stochastic process $\{\Theta_k^L(\omega)\}_{k\geq 0}$ is a non-homogeneous Markov chain of order 1.

Proposition 1 derives from the absorbent condition of state "1". The proof of this result is presented in Appendix B.1. Using Proposition 1, $\forall k \geq 1$:

$$p_k^L = P(\Theta_k^L(\omega) = 1 | \Theta_0^L(\omega) = 0, \ldots, \Theta_{k-1}^L(\omega) = 0) = P(\Theta_k^L(\omega) = 1 | \Theta_{k-1}^L(\omega) = 0). \tag{3.10}$$

Therefore, the sequence $\{p_k^L\}_{k\geq 0}$ represents the time-variant transition probabilities of the Markov chain that characterizes the failure condition and, thus, the ToF prior probability distribution. Fig. 3.2 illustrates the transition between time instants $k-1$ and k.

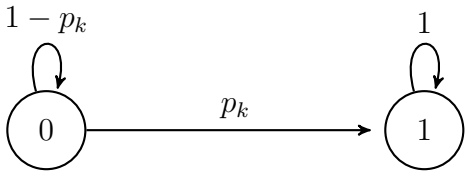

Figure 3.2: Diagram that describes the transition probabilities of the Markov chain $\{\Theta_k^L(\omega)\}_{k\geq 0}$ between time instants $k-1$ and k.

Importantly, the transition matrix of this non-homogeneous Markov chain is time-variant, and it is given by:

$$P_k^L := \begin{bmatrix} P(\Theta_k^L(\omega) = 0 | \Theta_{k-1}^L(\omega) = 0) & P(\Theta_k^L(\omega) = 1 | \Theta_{k-1}^L(\omega) = 0) \\ P(\Theta_k^L(\omega) = 0 | \Theta_{k-1}^L(\omega) = 1) & P(\Theta_k^L(\omega) = 1 | \Theta_{k-1}^L(\omega) = 1) \end{bmatrix} = \begin{bmatrix} 1 - p_k^L & p_k^L \\ 0 & 1 \end{bmatrix} \quad , \forall k \geq 1. \tag{3.11}$$

Consequently, $\{\Theta_k^L(\omega)\}_{k\geq 0}$ is a non-homogeneous Markov chain, in general.

3.1. Characterization of $\tau^L(\omega)$ in Eq. (3.7)

In this section, the previously defined stochastic processes are related with the time of failure random variable $\tau^L(\omega)$. This variable is the most important one when executing the prognostic task. From (3.7), the following property holds.

PROPOSITION 2 Suppose that $P(\Theta_0^L(\omega) = 0) = 1$ and $P(\exists k \in \mathbb{N}, X_k(\omega) \in L) = 1$. Then, $\forall k \geq 1$:

$$P(\tau^L(\omega) = k) = p_k^L \cdot \prod_{j=1}^{k-1}(1 - p_j^L). \tag{3.12}$$

The proof of Proposition 2 is presented in Appendix B.2. The important point of this result is that this representation just needs to know the transition probabilities $\{p_k^L\}_{k\geq 0}$ of the Markov chain $\{\Theta_k^L(\omega)\}_{k\geq 0}$ to characterize the probability mass function of $\tau^L(\omega)$. Note that, if $\{p_k^L\}_{k\geq 0}$ is known, Eq. (3.12) offers an implementable closed-form expression.

The hypothesis behind Proposition 2 are not too strong in a degrading system. It can easily be assumed that the system has not failed catastrophically at the beginning of the analysis, hence $P(\Theta_0^L(\omega) = 0) = 1$. On the other hand, $P(\exists k \in \mathbb{N}, X_k(\omega) \in L) = 1$ follows from the fact that a non-regenerative degrading system reaches the failure condition at some point (as the time tends to infinity), almost-surely.

In many contexts, the cumulative distribution of $\tau^L(\omega)$ is also useful for decision-making purposes. By definition:

$$P(\tau^L(\omega) \leq k) = \sum_{j=1}^{k} P(\tau^L(\omega) = j) \quad , \forall k \geq 1. \tag{3.13}$$

Then, the following result holds:

PROPOSITION 3 The cumulative probability distribution of τ^L is characterized by the stochastic process $\{\Theta_k^L\}_{k\geq 0}$ through the following equation:

$$P(\tau^L(\omega) \leq k) = P(\Theta_k^L(\omega) = 1) \quad , \forall k \geq 1. \tag{3.14}$$

Proof. The proof follows directly from:

$$P(\tau^L(\omega) \leq k) = 1 - P(\tau^L(\omega) > k) = 1 - P(\Theta_k^L(\omega) = 0) = P(\Theta_k^L(\omega) = 1).$$

Note that the events $\{\tau^L(\omega) > k\} = \{\Theta_k^L(\omega) = 0\}$ from the definition of $\tau^L(\omega)$ in Eq. (3.7). If the failure occurs after time k, then $\Theta_k^L(\omega) = 0$; the opposite is also true due to the absorbent condition of state 1 ■

Regarding Proposition 3, using the Markovian property (see Proposition 1) and denoting the initial distribution as $\mu = [1\ ,\ 0]$, it is easy to compute $P(\Theta_k^L(\omega) = 1)$ from the linear relation:

$$[P(\Theta_k^L(\omega) = 0) \quad , \quad P(\Theta_k^L(\omega) = 1)] = \mu \cdot P_1^L \cdot ... \cdot P_k^L \quad , \forall k \geq 1. \tag{3.15}$$

3.2. Characterizing the transition probabilities of $\{\Theta_k^L(\omega)\}_{k\geq 0}$

In the previous analysis, the sequence $\{p_k^L\}_{k\geq 0}$ was needed, which describes the transition probabilities of $\{\Theta_k^L(\omega)\}_{k\geq 0}$. It is clear that the dynamics of this Markov chain depend on trajectory of the system state. Indeed, the following result holds:

PROPOSITION 4 The transition probabilities of $\{\Theta_k^L(\omega)\}_{k\geq 0}$ can be obtained from the space-state probability distribution. Indeed, $\forall k \geq 1$:

$$p_k^L = \frac{P\left(X_k(\omega) \in L, (X_1(\omega), \dots X_{k-1}(\omega)) \in (L^c)^{k-1}\right)}{P\left((X_1(\omega), \dots X_{k-1}(\omega)) \in (L^c)^{k-1}\right)}. \tag{3.16}$$

The proof of this result is presented in Appendix B.3. Then, p_k^L measures the proportion of system state trajectories that have not failed until time $k-1$, but fail catastrophically at k. Note that the system dynamics given by the degradation model (3.1)-(3.2) allow to compute (3.16), given an initial state $X_0(\omega)$ and an usage profile $\{U_k(\omega)\}_{k\geq 0}$ through the total probability law. In other words, the transition probabilities $\{p_k^L\}_{k\geq 0}$ of the Markov chain process $\{\Theta_k^L(\omega)\}_{k\geq 0}$ are completely determined by the system dynamics, the initial state, and the usage profile.

Proposition 4 is the most important result of this work since it links both the state-space model given by Eqs. (3.1)-(3.2), and the proposed Markov chain model $\{\Theta_k^L(\omega)\}_{k\geq 0}$. While conventional prognostic algorithms focus on the state-space model to estimate the ToF, Proposition 2 indicates that it is enough to characterize the sequence $\{p_k^L\}_{k\geq 0}$, and Proposition 4 express how to accomplish this. Therefore, if the objective is to determine the ToF distribution, then describing the trajectory of the system state is equivalent to studying $\{p_k^L\}_{k\geq 0}$. These ideas are illustrated in the scheme of Figure 3.3.

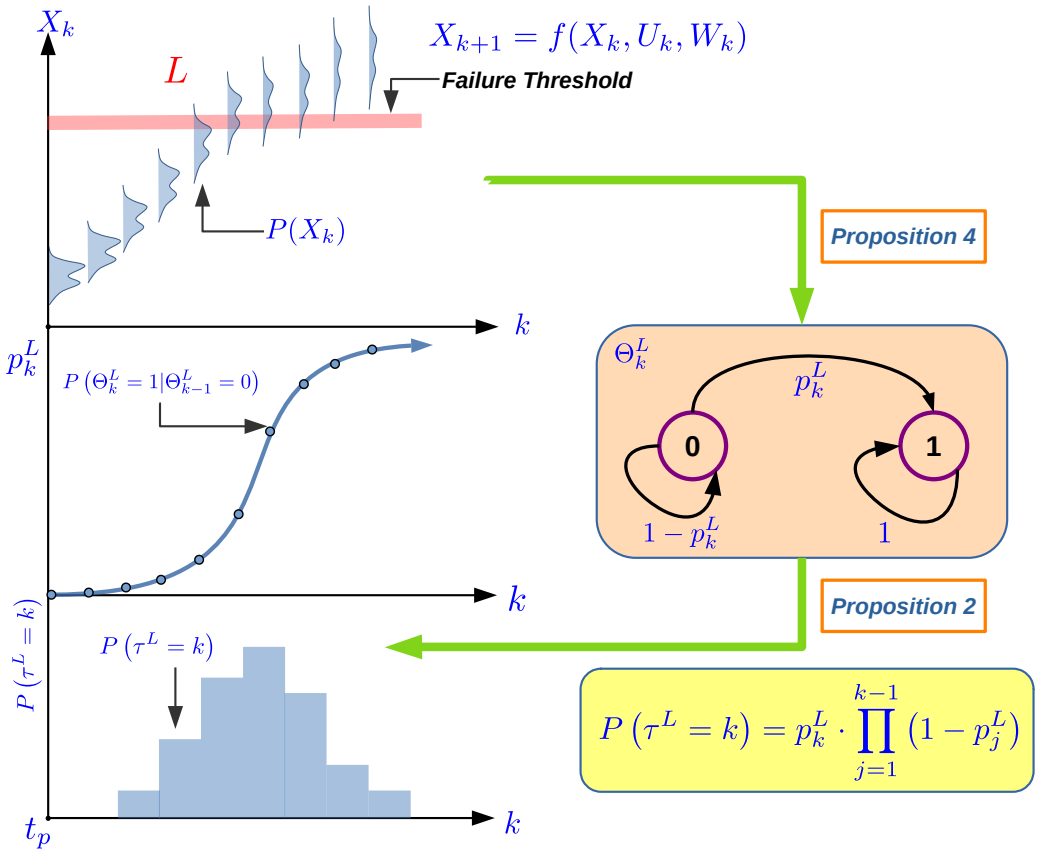

Figure 3.3: General scheme of the reduced representation and its use. Typically, space-state trajectories are used to compute the ToF distribution. In the proposed approach, the state trajectories are projected onto a two-state non-homogeneous Markov chain based on Proposition 4. Then, this reduced model is used to estimate the ToF distribution applying Proposition 2. The idea is to characterize the Markov chain model without predicting the state distributions, reducing the computational cost of the prognostic task.

Motivated by this observation, this formalization is used in the next chapter to propose a prognostic algorithm for degrading systems.

Chapter 4

Fast-Running Markov Chain-based Prognostic Algorithm

A new model representing the degradation process using a non-homogeneous Markov chain has been developed. Moreover, an explicit, closed, and simple equation to calculate the time of failure (ToF) mass probability distribution based on that model has been derived. This thesis proposal is to utilize this new representation to predict the ToF distribution and update this prediction as new measurements of the observation variable are received during the real-time operation of the system. With this aim, this chapter proposes a prognostic algorithm that comprises two stages: an offline stage (or learning stage) and an online stage (or evaluation stage). The algorithm is called *Fast-Running Markov Chain-based Prognostic Algorithm* (FRMC-PA).

For all practical purposes, consider a time instant t_p. Then, $\forall k \geq t_p$, the *posterior transition probability* at time k, given measurements up to time t_p, is defined by:

$$p^L_{k|t_p} \triangleq P(\Theta^L_k(\omega) = 1 | \Theta^L_{k-1}(\omega) = 0, Y_{1:t_p}(\omega) = y_{1:t_p}). \tag{4.1}$$

In other words, the posterior transition probabilities in Eq. (4.1) correspond to the transition probabilities in Eq. (3.10), conditional on measurements of the observation variable $Y_{1:t_p}(\omega)$. In this case, we are also interested in characterizing them utilizing the space-state trajectories.

PROPOSITION 5 The posterior transition probabilities of $\{\Theta^L_k(\omega)\}_{k\geq 0}$ can be computed from the conditional space-state distributions through the following relation. Let $t_p \in \mathbb{N}$. $\forall k \geq t_p$:

$$p^L_{k|t_p} = \frac{P\left(X_k(\omega) \in L, (X_1(\omega), \dots X_{k-1}(\omega)) \in (L^c)^{k-1} | Y_{1:t_p}(\omega) = y_{1:t_p}\right)}{P\left((X_1(\omega), \dots X_{k-1}(\omega)) \in (L^c)^{k-1} | Y_{1:t_p}(\omega) = y_{1:t_p}\right)}. \tag{4.2}$$

The derivation of this result follows the same arguments used in the proof of Proposition 4. Now, the stages of the proposed FRMC prognostic algorithm are presented.

4.1. Offline learning stage

The offline stage is executed before the system begins its operation. Its purpose is to build a function φ that takes a measurement of the observation variable y_{t_p} at time t_p and returns a posterior estimate sequence of transition probabilities $\{\hat{p}^L_{k|t_p}\}_{k=t_p \ldots t_p+P_h}$, where P_h is the prediction horizon. φ is then utilized in the online stage to execute real-time prognostics of the time of failure distribution.

More generally, φ can also use a prior sequence $\{p^L_k\}_{k=t_p \ldots t_p+N_{prior}}$ as an argument, where $N_{prior} < P_h$ is an hyper-parameter that determines the length of the prior sequence that is considered to compute the posterior sequence $\{\hat{p}^L_{k|t_p}\}_{k=t_p \ldots t_p+P_h}$. If that is the case, φ acts as a Bayesian processor because it combines prior knowledge $\left(\{p^L_k\}_{t_p \ldots t_p+N_{prior}}\right)$ and measurements $\left(y_{t_p}\right)$ to obtain an estimate of the posterior sequence $\{\hat{p}^L_{k|t_p}\}_{k=t_p \ldots t_p+P_h}$.

Supposing that the state-space model that describes the degradation process dynamics (3.1)-(3.2) is known, the offline stage includes the following steps:

A1) Use the model (3.1)-(3.2), an initial distribution $P(X_{t_p}(\omega))$ and a probabilistic characterization of the usage profile $\{\hat{U}_k(\omega)\}_{k=t_p \ldots t_p+P_h-1}$ (which may depend on $X_{t_p}(\omega)$) to compute a sequence of prior distributions denoted by $P\left(X_k(\omega)\right)_{k=t_p \ldots t_p+P_h}$. Formally, using the law of total probabilities, $\forall x_k \in \mathbb{R}^{n_x}$, $\forall k$ such that $t_p < k \leq t_p + P_h$:

$$
\begin{aligned}
&P(X_k(\omega) = x_k) \\
&\approx \int_{x_{t_p} \in \mathbb{R}^{n_x}} \int_{\substack{u_j \in \mathbb{R}^{n_u}, \\ t_p \leq j \leq t_p+P_h-1}} \Big[P\left(X_k(\omega) = x_k | X_{t_p}(\omega) = x_{t_p}, \hat{U}_{t_p:t_p+P_h-1}(\omega) = u_{t_p:t_p+P_h-1}\right) \\
&\qquad \cdot P\left(\hat{U}_{t_p:t_p+P_h-1}(\omega) = u_{t_p:t_p+P_h-1} | X_{t_p}(\omega) = x_{t_p}\right) \\
&\qquad \cdot P\left(X_{t_p}(\omega) = x_{t_p}\right) \Big] \, du_{(t_p:t_p+P_h-1)} \, dx_{t_p}. \qquad (4.3)
\end{aligned}
$$

Note that the first term of the integral can be calculated using Eq. (3.9), the second term from the probabilistic characterization of the usage profile, and the third term from the given initial distribution. Eq. (4.3) is an approximation since we are using a

probabilistic characterization of the exogenous input $\{\hat{U}_k(\omega)\}_{k=t_p \ldots t_p+P_h}$ instead of the real distribution, which may be unknown.

A2) Use the sequence of prior distributions for $\{X_k(\omega)\}_{k=t_p \ldots t_p+P_h}$ and the failure set L to compute a sequence of prior transition probabilities $\{p_k^L\}_{k=t_p \ldots t_p+N_{prior}}$ based on Eq. (3.16).

A3) Utilize the model (3.1)-(3.2), an initial distribution $P(X_{t_p}(\omega))$, a probabilistic characterization of the usage profile $\{\hat{U}_k(\omega)\}_{k=t_p \ldots t_p+P_h-1}$ and measurements $y_{1:t_p}$ to compute a sequence of model state conditional distributions $P\left(X_k(\omega)|Y_{1:t_p}(\omega)=y_{1:t_p}\right)_{k=t_p \ldots t_p+P_h}$. Formally, $\forall x_k \in \mathbb{R}^{n_x}$, $\forall k$ such that $t_p \leq k \leq t_p+P_h$:

$$
\begin{aligned}
& P\left(X_k(\omega)=x_k \ |Y_{1:t_p}(\omega)=y_{1:t_p}\right) \\
& \approx \int_{x_{t_p}\in\mathbb{R}^{n_x}} \int_{\substack{u_k\in\mathbb{R}^{n_u},\\ t_p\leq k\leq t_p+P_h-1}} \Big[P\left(X_k(\omega)=x_k|X_{t_p}(\omega)=x_{t_p}, \hat{U}_{t_p:t_p+P_h-1}(\omega)=u_{t_p:t_p+P_h-1},\right. \\
& \qquad\qquad \left. Y_{1:t_p}(\omega)=y_{1:t_p}\right)\cdot P\left(\hat{U}_{t_p:t_p+P_h-1}(\omega)=u_{t_p:t_p+P_h-1}|X_{t_p}(\omega)=x_{t_p}\right) \\
& \qquad\qquad \cdot P\left(X_{t_p}(\omega)=x_{t_p}|Y_{1:t_p}(\omega)=y_{1:t_p}\right)\Big]\, du_{(t_p:t_p+P_h-1)}\, dx_{t_p} \qquad (4.4) \\
& = \int_{x_{t_p}\in\mathbb{R}^{n_x}} \int_{\substack{u_k\in\mathbb{R}^{n_u},\\ t_p\leq k\leq t_p+P_h-1}} \Big[P\left(X_k(\omega)=x_k|X_{t_p}(\omega)=x_{t_p}, \hat{U}_{t_p:t_p+P_h-1}(\omega)=u_{t_p:t_p+P_h-1}\right) \\
& \qquad\qquad \cdot P\left(\hat{U}_{t_p:t_p+P_h-1}(\omega)=u_{t_p:t_p+P_h-1}|X_{t_p}(\omega)=x_{t_p}\right) \\
& \qquad\qquad \cdot P\left(X_{t_p}(\omega)=x_{t_p}|Y_{1:t_p}(\omega)=y_{1:t_p}\right)\Big]\, du_{(t_p:t_p+P_h-1)}\, dx_{t_p}. \qquad (4.5)
\end{aligned}
$$

In this case, the first term of the integral is simplified since $X_k(\omega)$ is independent of $Y_{1:t_p}$, conditional on X_{t_p}. This term can be calculated utilizing Eq. (3.9). The second term is the same as in Eq. (4.3). The third term can be calculated using Eq. (3.8) and the Bayes theorem.

A4) Utilize the sequence of conditional distributions $P\left(X_k(\omega)|Y_{1:t_p}(\omega)=y_{1:t_p}\right)_{k=t_p \ldots t_p+P_h}$ and the failure set L to compute a sequence of posterior transition probabilities $\{p_{k|t_p}^L\}_{k=t_p \ldots t_p+P_h}$ based on Eq. (4.2).

A5) Gather the triplet $\left(y_{t_p}, \{p_k^L\}_{k=t_p \ldots t_p+N_{prior}}, \{p_{k|t_p}^L\}_{k=t_p \ldots t_p+P_h}\right)$ as a data point.

A6) Using different prior initial distributions $P(X_{t_p}(\omega))$ and measurements $y_{1:t_p}$, iterate the previous steps to generate the data set:

$$
\mathcal{D}=\left\{\left(y_{t_p}^j, \{p_k^L\}_{k=t_p \ldots t_p+N_{prior}}^j, \{p_{k|t_p}^L\}_{k=t_p \ldots t_p+P_h}^j\right) : j\in\{1 \ldots N_{data}\}\right\}. \qquad (4.6)
$$

A7) Utilize the data set $\mathcal{D}$ as supervised data to train a multivariate regression model φ of

the form:

$$\varphi : \left(y_{t_p}, \{p_k^L\}_{k=t_p \ldots t_p+N_{prior}}\right) \mapsto \{\hat{p}_{k|t_p}^L\}_{k=t_p \ldots t_p+P_h}. \tag{4.7}$$

Consider the restriction $\hat{p}_{k|t_p} \in (0,1)$ during the training process.

4.2. Online evaluation stage

The objective of this stage is to utilize the model φ trained during the offline stage to execute the prognostic task in real-time. The idea is to avoid predictions on the space-state variable every time a new measurement is received (like in particle filter algorithms [7, 47]) and to use φ as a look up table to update the time of failure distribution, saving computational time in the process.

Let t_0 be the time when a fault is detected and diagnosed. The online stage comprises the following steps. Steps B1 and B2 are only necessary when a prior transition probabilities sequence is used as an input of the regression model φ.

B1) Use model (3.1)-(3.2), the initial condition x_{t_0} (for example, an estimation or a PDF obtained from the fault diagnostics) and a probabilistic characterization of the exogenous input $\{\hat{U}_k(\omega)\}_{k=t_0 \ldots t_0+P_h-1}$ to compute a sequence of prior transition probabilities $\{p_k^L\}_{k=t_0 \ldots t_0+P_h}$.

B2) Use $\{p_k^L\}_{k=t_p \ldots t_p+P_h}$ and Eq. (3.12) to compute the prior ToF distribution $\{P(\tau^L(\omega) = k)\}_{k=t_0 \ldots t_0+P_h}$.

B3) Utilize the measurement of the observation variable at time t_0+1 –namely y_{t_0+1}– and the regression model φ to compute posterior estimates of the transition probabilities in the form of:

$$\{\hat{p}_{k|t_0+1}^L\}_{k=t_0+1 \ldots t_0+1+P_h} = \varphi\left(y_{t_0+1}, \{p_k^L\}_{k=t_0+1 \ldots t_0+1+N_{prior}}\right). \tag{4.8}$$

B4) Update the ToF distribution using $\{\hat{p}_{k|t_0+1}^L\}_{k=t_0+1 \ldots t_0+1+P_h}$ and Eq. (3.12), obtaining estimates $\{\hat{P}(\tau^L(\omega) = k | Y_{1:t_0+1}(\omega) = y_{1:t_0+1})\}_{k=t_0+1 \ldots t_0+1+P_h}$.

B5) For $t \geq t_0+2$, utilize the measurement y_t and the model φ to compute posterior estimates in the form of:

$$\{\hat{p}_{k|t}^L\}_{k=t \ldots t+P_h} = \varphi\left(y_t, \{\hat{p}_{k|t-1}^L\}_{k=t \ldots t+N_{prior}}\right). \tag{4.9}$$

B6) Update the ToF distribution using $\{\hat{p}^L_{k|t}\}_{k=t\ldots t+P_h}$ and Eq. (3.12), obtaining estimates $\{P(\tau^L(\omega)=k|Y_{1:t}(\omega)=y_{1:t})\}_{k=t\ldots t+P_h}$.

To better understand the implementation of the proposed algorithm, Figure 4.1 summarizes the main steps of both the offline and the online stages, as well as the flux of information between them.

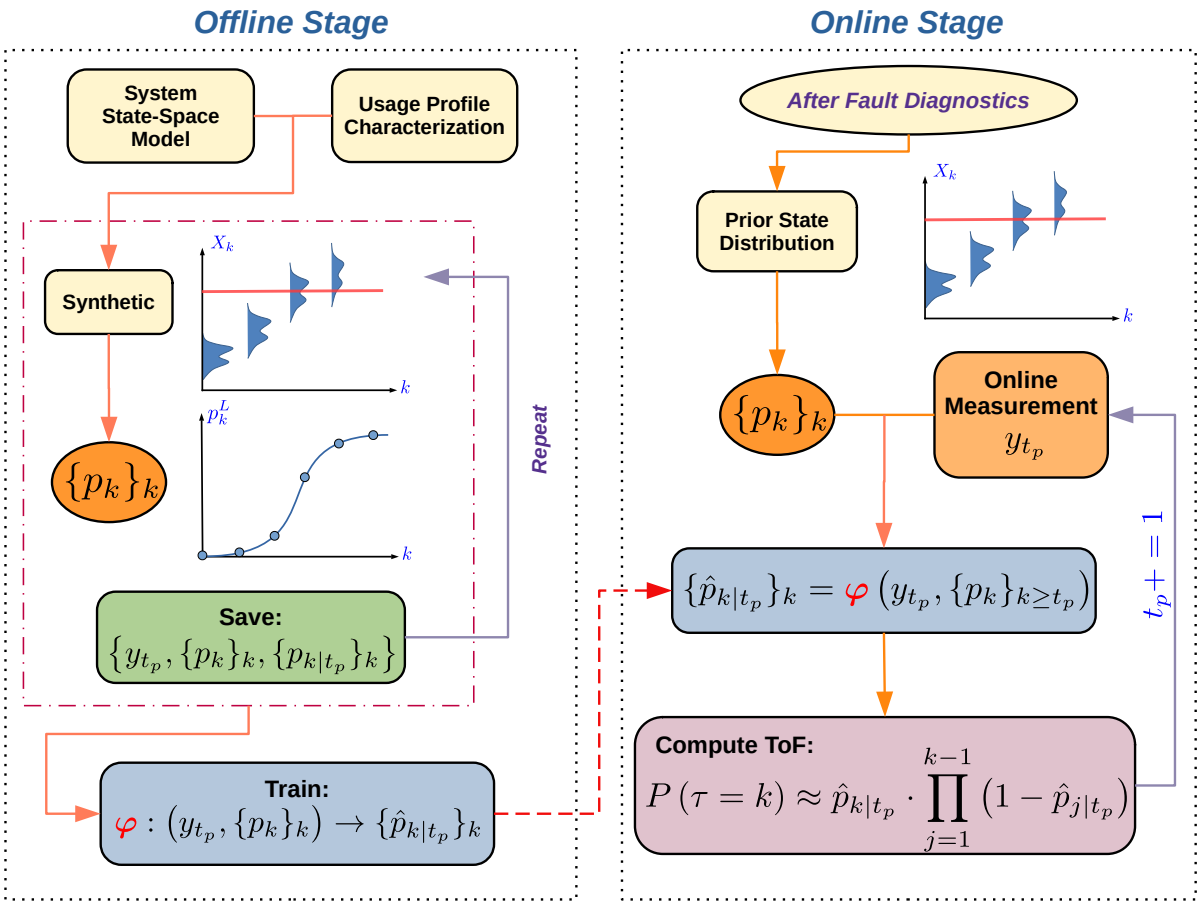

Figure 4.1: Graphical abstract of the proposed FRMC prognostic algorithm. The objective of the offline stage is to train the regression model φ, whereas the online stage utilizes φ to estimate the ToF distribution during the real-time operation of the system.

Remarks: Some of the steps of the proposed algorithm imply the calculation of probability distributions during the offline stage. For instance, Eqs. (3.16), (4.2), (4.3), (4.5). However, in practice, this is an expensive task because it implies the computation of multiple integrals. Appendix C details some numerical methods to approximate the desired distributions.

Note that failure prognostic algorithms that focus on the state-space model execute step

A3 online, every time a new measurement is received. This step is demanding in terms of computational efforts, becoming one of the main difficulties in conventional prognostic algorithms. On the other hand, the FRMC-PA executes step A3 during the offline stage, and captures the relevant information in the sequence $\{p^L_{k|t_p}\}_{k=t_p \dots t_p+P_h}$. In other words, the proposed algorithm transfers most of the computational burden to the offline stage, leaving few computations to the online stage and speeding up the update of the ToF distribution.

Regarding the regression model φ, it only uses the last measurement y_t (see Eq. (4.7)) instead of every previous measurement $y_{1:t}$. Nevertheless, as noted in Eq. (4.9), the information of the system up to time $t-1$ (including the previous measurements $y_{1:t-1}$) is incorporated in the sequence $\{\hat{p}^L_{k|t-1}\}_{k=t \dots t+N_{prior}}$. Combining this with the new measurement y_t, the transition probabilities are updated using φ, obtaining the posterior sequence $\{\hat{p}^L_{k|t}\}_{k=t \dots t+P_h}$.

Finally, it is important to mention some of the challenges that this proposal demands. To obtain good estimates of the posterior transition probabilities, it is essential to utilize an adequate regression model and generate a data set as diverse as possible. The quality of the obtained model φ is determinant in the performance of the prognostic algorithm during the online stage. Additionally, even that the computational burden is not as significant during the offline stage as it is in the online stage, a high-dimensional state could hinder the generation of the data set $\mathcal{D}$ in step A6.

In the same line, regarding the challenges, an important assumption is that the behavior of the degradation during the online stage should not be significantly different from the one learned in the training stage. Indeed, the usage profile stochastic characterization that is used during the offline training stage should remain valid during the online operation of the system. This way, it is possible to predict the behavior of the degradation using the built model φ. If there were different usage profiles (each of them characterized by a stochastic model), it would be necessary to train a regression model for each. Then, during the online stage, the algorithm should detect the most likely usage profile (from the last inputs, for example) and use the most adequate regression model to estimate the posterior transition probabilities.

In the next chapter, the FRMC-PA is applied in a synthetic scenario.

Chapter 5

Synthetic Example

This chapter illustrates the implementation of the proposed algorithm in a simple synthetic one-dimensional dynamic model that characterizes a degradation process. The focus is on the offline stage, explaining the training steps of the function φ in Eq. (4.7). The state-space equations of this model are:

$$X_{k+1}(\omega) = X_k(\omega) + U_k(\omega) + W_k(\omega) \tag{5.1}$$

$$Y_k(\omega) = X_k(\omega) + V_k(\omega) \quad , \forall k \in \mathbb{N}. \tag{5.2}$$

Consider the initial condition $X_0 = 0.1$ and a deterministic usage profile $U_k = 1/30\,,\ \forall k \geq 0$. Hence, the only sources of uncertainty are the process and observation noises, W_k and V_k. Both are considered Gaussians with mean 0 and standard deviations $\sigma_w = 0.03$ and $\sigma_v = 0.05$, respectively. All the previous characterizes the dynamics of the degradation process. On the other hand, the failure condition is set as $L = \{x \in \mathbb{R}^{n_x} : x \geq 0.9\}$, which means that the system fails when the state reaches the threshold $Th = 0.9$.

Given that the degradation process dynamics are known, the off-line stage includes two sub-stages. First, the transition probabilities of the Markov chain $\{\Theta_k(\omega)\}_{k\geq 0}$ are computed based on a probabilistic characterization of the state and a given measurement. Then, based on a generated data set, the regression model φ is trained to use it during the online stage.

5.1. Obtaining the transition probabilities of $\{\Theta_k(\omega)\}$ [FRMC-PA steps A1-A5]

Based on the synthetic model of Eqs. (5.1)-(5.2), the degradation process can be represented with the non-homogeneous Markov chain proposed in Chapter 3. For this purpose, a particle-filtering-based algorithm –detailed in Appendix C– was used to compute the sequence $\{p_k^L\}_{k=t_p \ldots t_p+P_h}$ in Eq. (3.16). It is important to mention that this step can also be performed using Monte Carlo simulation, since this is an offline stage in the process of implementing the proposed prognostic method.

In this example, the number of particles was $N_p = 5000$. Initializing every trajectory on the initial state $X_0 = 0.1$, the particles were propagated through the system dynamics given by (5.1), using different realizations of the process noise for each particle. Based on this, the prior distribution of the state was obtained (step A1). Figure 5.1 shows the mean value of the distribution at each time k, and an error bar corresponding to double the standard deviation of the particles. Note that it is enough to consider a prediction horizon of $P_h = 40$ time steps to characterize adequately the time of failure distribution because at that moment the failed condition is widely surpassed.

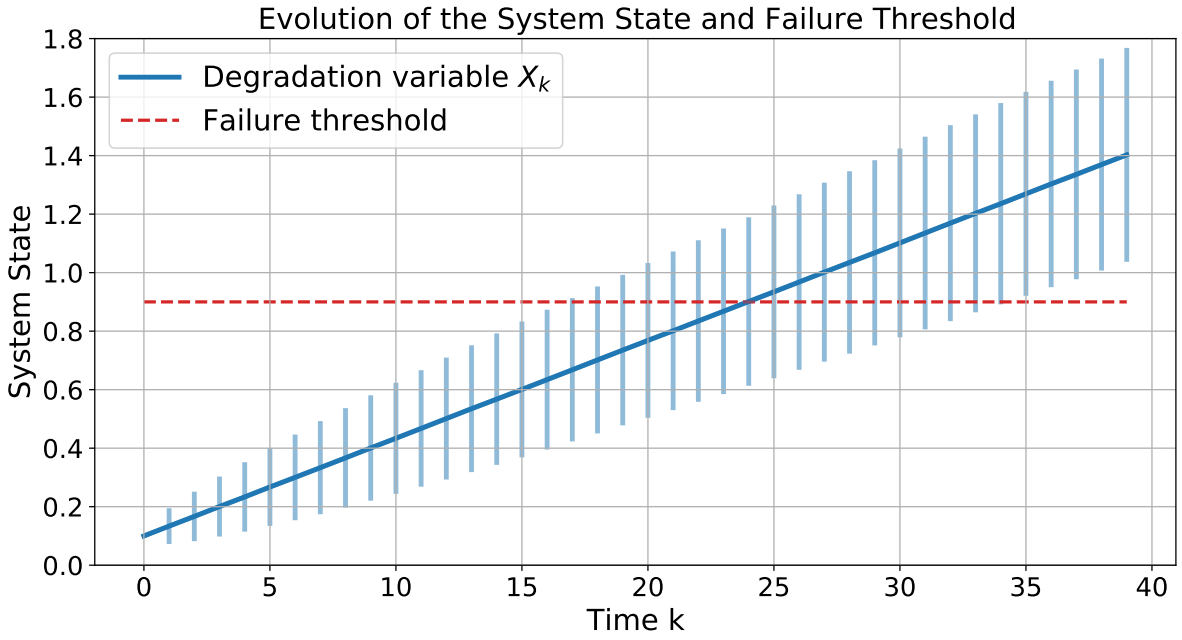

Figure 5.1: Evolution of the system state in the synthetic example, considering the mean and double the standard deviation.

Using the method explained in Appendix C, the transition probabilities $\{p_k^L\}_{k=1 \ldots 40}$ were computed based on the prior distribution of the system state (step A2). Then, from Eq. (3.12) the prior ToF distribution was obtained. The effects of an observation of the variable

Y_k on the transition probabilities $\{p_k^L\}_{k=1...40}$ were also explored. Specifically, the posterior sequence $\{\hat{p}_{k|1}^L\}_{k=1...40}$ after a measurement of Y_1 was obtained. This measurement modifies the estimation of the state distribution at $k = 1$, obtaining the posterior, and consequently modifies every future state distribution (step A3). To illustrate, let $Y_1 = y_1 = 0.4$. Considering (5.2), this measurement indicates that at $k = 1$ the system state is closer to the failure set than what is suggested by the prior particles distribution observed in Figure 5.1. Hence, it is expected that the posterior sequence $\{p_{k|1}^L\}_{k=1...40}$ (obtained in step A4) steeps up before the prior $\{p_k^L\}_{k=1...40}$, and that the posterior ToF distribution concentrates on smaller times than the prior ToF distribution. This is exactly what was obtained, as shown in Figures 5.2 and 5.3. Finally, the triplet $\left(y_1, \{p_k^L\}_{k=1...40}, \{p_{k|1}^L\}_{k=1...40}^j\right)$ was gathered as a data point (step A5) for the training process of φ.

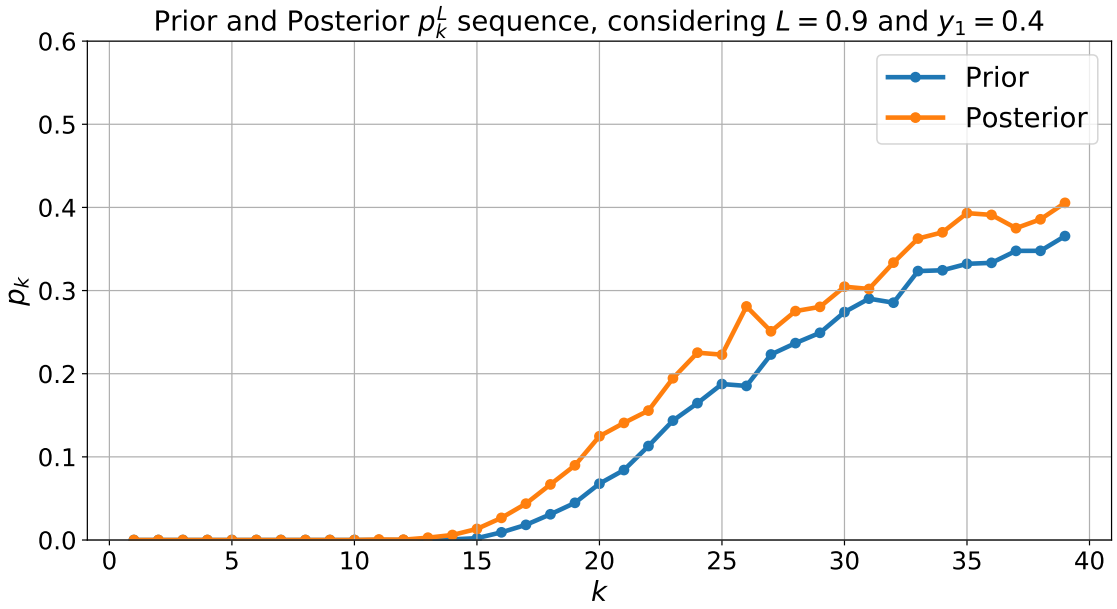

Figure 5.2: Prior and posterior sequences in the synthetic example. The posterior sequence considered a measurement at $k = 1$ given by $y_1 = 0.4$.

5.2. Training φ [FRMC-PA steps A6-A7]

The previous section explained how to compute the failure prior transition probabilities based on the system dynamics, an initial condition and a future usage profile. Then, using these transition probabilities, the ToF distribution was obtained. The transition probabilities and the ToF prior distribution were also updated given a measurement of the observation variable.

However, to generate the sequence $\{p_k^L\}$, a particle-filtering-based algorithm was used (details in Appendix C). As explained before, it is desirable to avoid using this type of

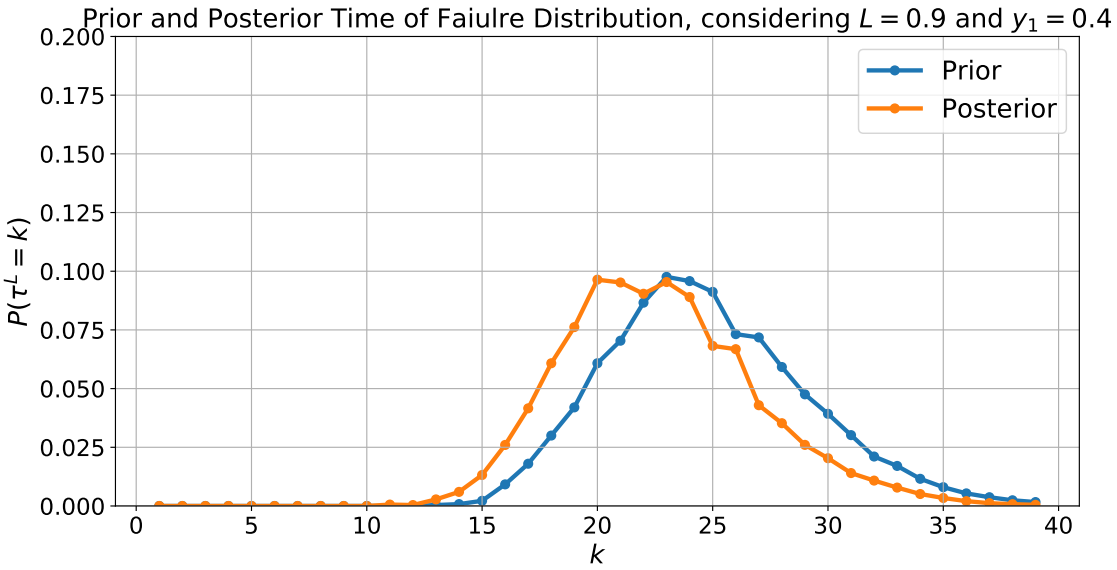

Figure 5.3: Prior and posterior time of failure distribution in the synthetic example. The posterior distribution considered a measurement at $k = 1$ given by $y_1 = 0.4$.

algorithms during the online stage, given their high computational cost and the need for update the prognostics every time a new measurement is observed. Hence, the objective now is to learn a regression model φ that takes a measurement of the observation variable Y_{t_p} and a portion of the prior transition probabilities $\{p_k^L\}_{k=t_p \ldots t_p+N_{prior}}$, and returns a posterior sequence of transition probabilities $\{\hat{p}_k^L\}_{k=t_p \ldots t_p+P_h}$ directly, without going through a particle filter algorithm. Then, during the online stage, φ is used to handle the measurements at any time k and update the transition probabilities and, more importantly, the ToF distribution. φ is in the form:

$$\varphi : \left(y_{t_p}, \{p_k^L\}_{k=t_p \ldots t_p+N_{prior}}\right) \mapsto \{\hat{p}_{k|t_p}^L\}_{k=t_p \ldots t_p+P_h}. \tag{5.3}$$

To learn the regression model φ, a Recurrent Neural Network (RNN) was used, since there are temporal features between the variables intended to learn. Indeed, the sequence $\{p_k^L\}_{k=t_p \ldots t_p+P_h}$ can be seen as a time series, since each p_k value is related to its neighbours and there are not large variations between them. In particular, a Gated Recurrent Unit (GRU) was implemented [59].

To train this model, it is important to generate data points $\Big(y_{t_p}, \{p_k^L\}_{k=t_p \ldots t_p+N_{prior}}, \{p_{k|t_p}^L\}_{k=t_p \ldots t_p+P_h}\Big)$ as diverse as possible (step A6). To do this in the example, different prior distributions of the system state X_k at $k = t_p$ were considered, represented which N_p particles. For each prior distribution of the state, different measurements of the observation variable Y_{t_p} were simulated based on Eq. (5.2). Then, the prior distribution $\Big(P(X_{t_p})\Big)^j$ was used to generate the prior sequence $\{p_k^L\}_{k=t_p \ldots t_p+N_{prior}}^j$ as explained in the previous section. On the other hand, the prior distribution $\Big(P(X_{t_p})\Big)^j$ and the measurement $y_{t_p}^j$ we-

re used to generate the posterior sequence $\{p^L_{k|t_p}\}^j_{k=t_p \dots P_h}$. By doing this, the data triplet $\left(y^j_{t_p}, \{p^L_k\}^j_{k=t_p \dots t_p+N_{prior}}, \{p^L_{k|t_p}\}^j_{k=t_p \dots t_p+P_h}\right)$ was obtained. This way, $N_{data} = 40000$ data points were generated, obtaining the data set:

$$\mathcal{D} = \left\{ \left(y^j_{t_p}, \{p^L_k\}^j_{k=t_p \dots t_p+N_{prior}}, \{p^L_{k|t_p}\}^j_{k=t_p \dots t_p+P_h}\right) : j \in \{1 \dots N_{data}\} \right\}. \tag{5.4}$$

This data set is then used to train the GRU φ (step A7). In the example, $N_{prior} = 20$, which means that the 20 first transition probabilities of the prior sequence $\{p^L_k\}$ were considered to estimate the posterior. Then, considering that $Y_k \in \mathbb{R}$, the GRU had 21 inputs. The output size corresponded to the complete prediction horizon $P_h = 40$. Hence, a many-to-many structure was used for the GRU. The GRU was trained in Python 3.7 by Keras at the CPU of a Laptop computer with an Intel i5 2.5$[GHz]$ processor and 16$[GB]$ of RAM. The number of units of the GRU was set to 50, using $ReLU$ as the activation function. After the GRU, a dense-connected layer was added to obtain an output of size 1. Iterating the cell $P_h = 40$ times, the desired dimension of the output is obtained. The data was split in 70 % train and 30 % test. Test data was used to monitor the learning process. Although 200 epochs were considered, an Early Stopping criterion was used to terminate the training process if the test loss (Mean Squared Error) showed no improvements with respect to its best value after 10 consecutive epochs. The batch size was fixed at 500 and the optimization algorithm was Adam.

The loss progression per epoch of the training process in the example is shown in Figure 5.4. Thanks to the early stopping criterion, the loss for the test set does not increase, hence there is no over-fitting. Figure 5.5 shows an example of the performance of the model given a prior distribution and a measurement at $k_0 = 15$. The prior and posterior sequences are obtained using a particle filter, whereas the predicted sequence is obtained using the GRU φ, which uses the prior sequence and the measurement to predict the posterior sequence. Figure 5.6 shows a comparison of the resulting time of failure distribution. As we can see, the regression model φ adequately predicts both the posterior transition probabilities and the time of failure distribution.

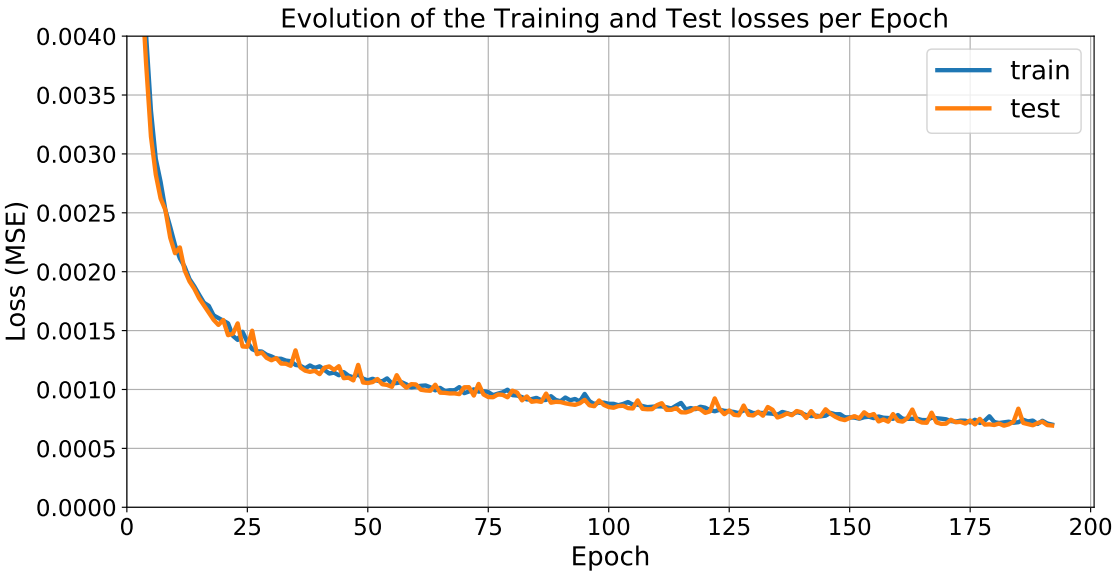

Figure 5.4: Evolution of the loss in the synthetic example during the training process.

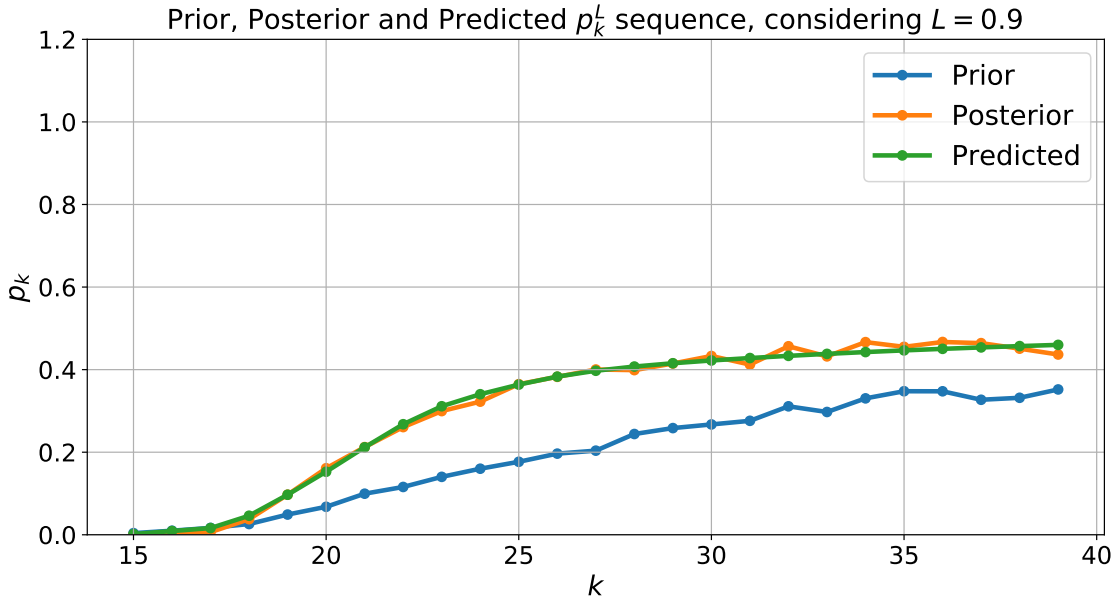

Figure 5.5: Performance of the GRU φ in the synthetic example, considering the prior distribution of the state from Figure 5.1 at $k_0 = 15$, and a measurement $y_{15} = 0.7$. Prior, posterior and predicted p_k^L sequence.

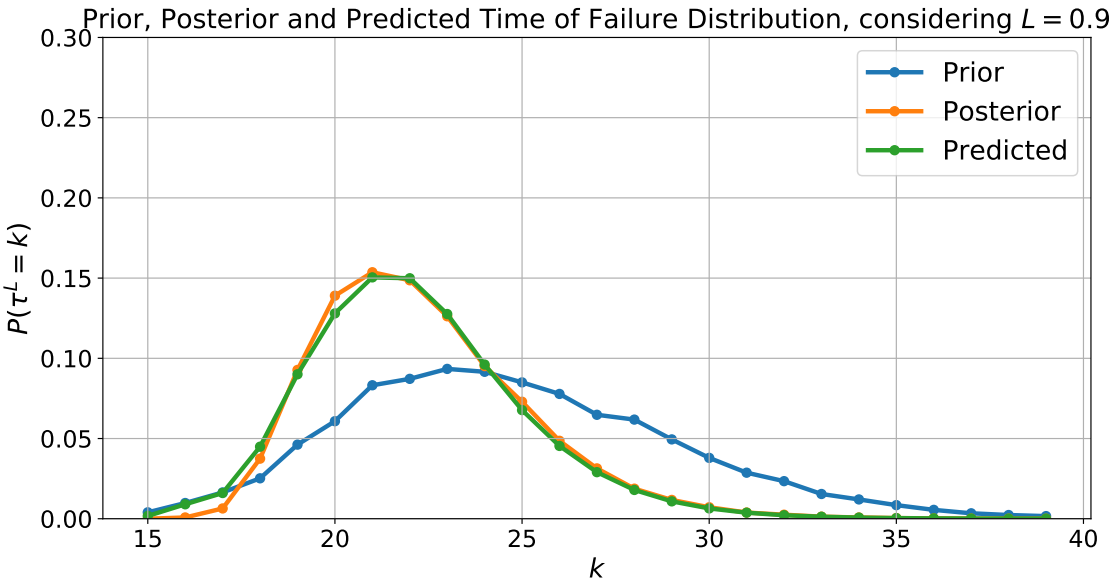

Figure 5.6: Performance of the GRU φ in the synthetic example, considering the prior distribution of the state from Figure 5.1 at $k_0 = 15$, and a measurement $y_{15} = 0.7$. Prior, posterior and predicted ToF distribution. φ predicts the transition probabilities in Figure 5.5, which are used to compute tho predicted ToF distribution using Eq. (3.12).

To evaluate the model, a mean sequence-by-sequence RMS test error is proposed. For every sequence in the test set, the norm-2 error between the real sequence and the predicted one was calculated. Then, the mean of the norm-2 error over all the test sequences was computed. Formally:

$$RMSE_{\varphi} \triangleq \frac{1}{N_{data}} \sum_{j=1}^{N_{data}} \left\| \left(p^L_{k|t_p}\right)^j_{k=t_p \dots t_p+P_h} - \varphi\left(y^j_{t_p}, \{p^L_k\}^j_{k=t_p \dots t_p+N_{prior}}\right) \right\|_2 . \tag{5.5}$$

Using this approach, a mean test sequence-by-sequence RMS error of 0.1362 was obtained for the transition probabilities sequences. On the other hand, for each sequence $\{p^L_{k|t_p}\}^j_{k=t_p \dots t_p+P_h}$ and its estimation using φ, the corresponding ToF distribution was calculated utilizing Eq. (3.12). In this case, the mean sequence-by-sequence RMS error was also calculated, obtaining a test RMS error of 0.0402 for the time of failure distribution.

Once the regression model φ has been trained, it can be used during the online stage. In this case, this stage was not implemented since this is a synthetic example and there is not a ToF ground truth value. In the next chapter, the proposed prognostic algorithm is applied to a case study of battery discharge of an electric bicycle, where real-world data is available and it is possible to implement and test the online stage.

Chapter 6

Experimental Validation

In this section, the implementation of the FRMC-PA in a battery discharge problem is illustrated [60]. In particular, the aim is to predict the end-of-discharge (EoD) time of the battery when it is powering an electric bicycle. The associated database includes voltage and current measurements acquired during 5 different routes of real operations of the bike. Before using the bike, the battery was always fully recharged. In this case, the degradation process corresponds to the discharge of the battery given the energy consumption of the bike. The failure condition is given by a cut-off voltage of $34[V]$. This means that a failure occurs when the voltage at the battery terminals is under $34[V]$. Note that this definition of the failure is used to apply the proposed framework to predict the battery EoD time and does not mean that the discharge condition corresponds literally to a catastrophic failure of the battery.

The complete data set is summarized in Fig. 6.1. The focus was on analyzing the data set #5, using data sets #1-4 as training data to learn the degradation model. For illustration purposes, the data until time $T_p = 18230[s]$ –in data set #5– was considered known, which implies a prognostic horizon of roughly $1200[s]$ (considering the failure condition of the system). In a real-world scenario, however, no observations of data set # 5 would be available considering that most of the learning process must be done offline and not during a particular route.

In this real-case scenario, a state-space model based on the law of energy conservation was utilized [61], and it is presented in Eqs. (6.1)-(6.2). In this model, the delivered energy corresponds to the state variable, the voltage is the observation variable, and the instantaneous power is the exogenous input. Formally, the state-space model corresponds to the following

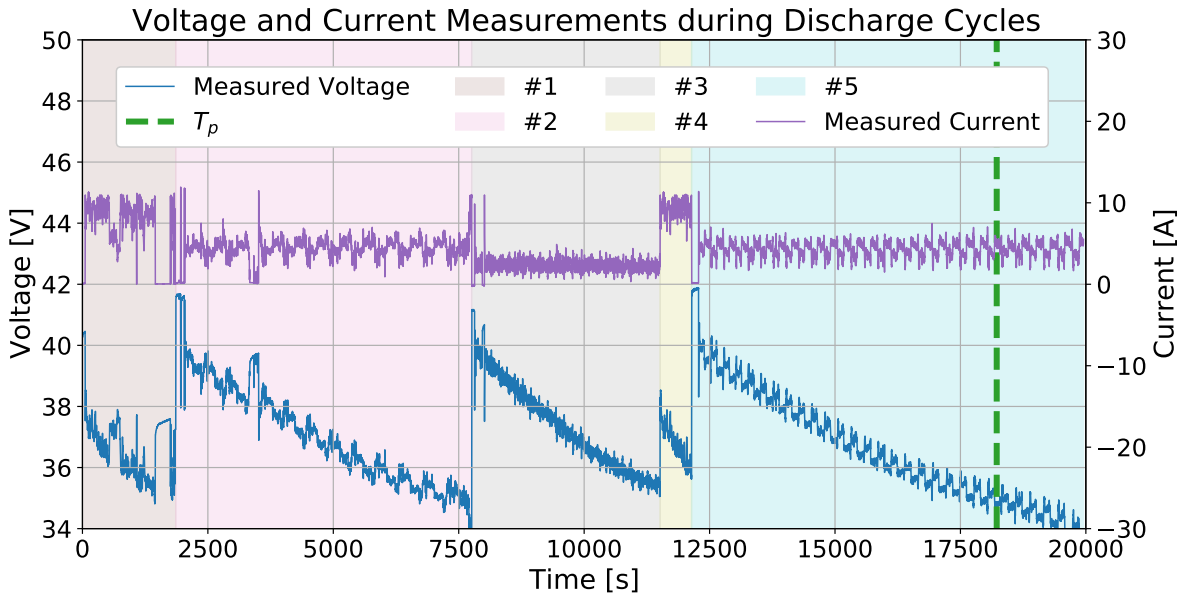

Figure 6.1: Measurements of current and voltage collected throughout the five discharge cycles in the electric bike. Each of the zones #1-5 corresponds to a discharge cycle. At the beginning of each, the battery was fully recharged.

equations:

$$\textbf{State transition equation:} \quad E_{k+1} = E_k + P_k \cdot \Delta t + W_k, \tag{6.1}$$

$$\textbf{Measurement equation:} \quad V_k = g(E_k, P_k) + N_k, \tag{6.2}$$

where E_k corresponds to the cumulative energy delivered by the battery from time 0 to k; P_k is the instantaneous power demand at time k; Δt is the sampling period ($1[s]$ in this particular case); V_k is the voltage measured at the battery terminals at time k. Note that $P_k = V_k \cdot I_k$, where I_k is the instantaneous current delivered by the battery. Hence, I_k can also be considered as the exogenous input. W_k and N_k are the process and the observation noises, respectively. Finally, $g(\cdot)$ is an unknown non linear function that describes the relationship between the measured voltage V_k and the system state E_k.

The next sections present the different steps of the proposed methodology to learn the regression model φ needed to execute the prognostic algorithm. During the offline stage, first the state-space model of the system is learned, in particular the function g in (6.2). Then, these dynamics are projected onto a two-state non-homogeneous Markov chain, obtaining the transition probabilities. Finally, the regression model φ is built to use it during the online stage, where the prognostic task is performed.

6.1. Offline stage

First, the offline stage is tackled. Its aim is to learn a regression model φ that takes a measurement of the observation variable V_{t_p} at time t_p and returns a posterior estimate sequence of transition probabilities $\{\hat{p}^L_{k|t_p}\}_{k=t_p \ldots t_p+P_h}$, where P_h is the prognostic horizon. In this case, the variable that defines the failure –V_k– depends on the other variables of the problem according to (6.1) and (6.2), where g is an unknown non-linear function. Hence, first the function g is represented using data-driven models.

6.1.1. Learning the degradation dynamics

To learn the observation relationship in (6.2), the available data was utilized to train an Artificial Neural Network (ANN). Recurrent Neural Networks were chosen because of the time dependencies observed in the voltage behaviour (see Figure 6.1). In addition, these models have been reported to provide positive results in terms of the prediction error [62, 63, 64]. Hence, they are appropriate to build the voltage predictive model $\hat{g}$. In particular, a Gated Recurrent Unit (GRU) was implemented, because of its simple structure and reasonable results [61, 59]. In this specific case study, the input of the GRU $\hat{g}$ at time k is structured as follows [61]:

$$\Gamma_k = \begin{bmatrix} I_{k-4} & E_{k-5} & P_{k-5} & \Delta I_{k-4} & \Delta V_{k-5} \\ I_{k-3} & E_{k-4} & P_{k-4} & \Delta I_{k-3} & \Delta V_{k-4} \\ I_{k-2} & E_{k-3} & P_{k-3} & \Delta I_{k-2} & \Delta V_{k-3} \\ I_{k-1} & E_{k-2} & P_{k-2} & \Delta I_{k-1} & \Delta V_{k-2} \\ I_k & E_{k-1} & P_{k-1} & \Delta I_k & \Delta V_{k-1} \end{bmatrix}, \tag{6.3}$$

where $\Delta I_k = I_k - I_{k-1}$, $\Delta V_k = V_k - V_{k-1}$. This input is passed through the GRU, obtaining as a result the predicted battery voltage $\hat{V}_k = \hat{g}(\Gamma_k)$.

For the training process, the available data set –shown in Figure 6.1– was split into 70 % train and 30 % test. The GRU was trained in Python 3.7 by Keras at the CPU of a Laptop computer with an Intel i5 2.5$[GHz]$ processor and 16$[GB]$ of RAM. In this case, the hidden size of the GRU unit was set at 64 and the activation function was ReLU. The batch size was fixed at 30. After the GRU, a linear dense-connected layer of 1 neuron was added.

Figure 6.2 shows the performance of the voltage prediction model when applied to the train and test data sets, including the absolute prediction error. In addition, Table 6.1 shows the performance of the model in terms of the Mean Absolute Percentage Error (MAPE) index. This index is evaluated in the train set, the test set, and the prognostic set. The last

corresponds to the data after time T_p, which is not considered for training purposes during the offline stage and will be used in the online stage.

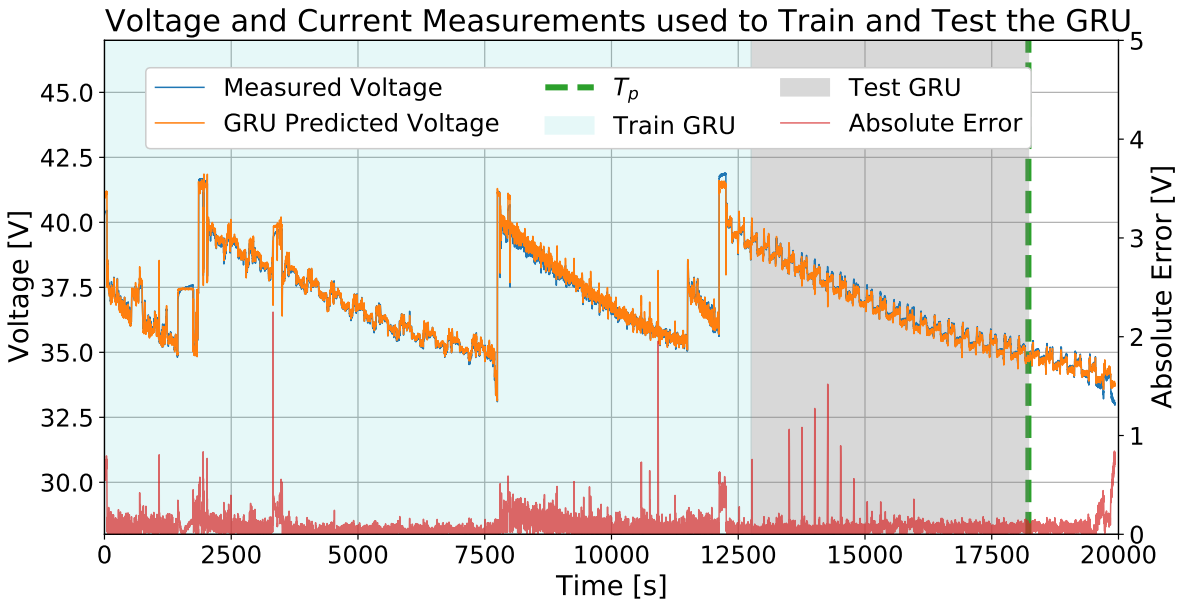

Figure 6.2: Operational data collected during usage of the electric bike, used to train and test the GRU model $\hat{g}$. The predicted voltage values are shown, as well as the absolute error between the real and predicted voltages.

	Train set	Test set	Prognostic set
MAPE [%]	0.2150	0.1750	0.3109

Table 6.1: Performance of the GRU model $\hat{g}$ in terms of the MAPE index. This index is evaluated in the train set, the test set and the prognostic set.

To obtain reliable predictions of the future voltages, the future usage profile needs to be characterized adequately. To do this, a probabilistic characterization based on an homogeneous Markov chain that considers different discharge current profiles was used [22]. In this case study, environmental temperature is relatively constant during the trip; hence, this variable was not considered during the analysis. To train the Markov chain that characterizes the usage profile, the latest 1000 acquired measurements of the current were utilized and 10 states were considered for the Markov chain. Additionally, the maximum likelihood criterion was used to estimate the transition matrix [22]. This stochastic characterization of the discharge current is denoted as $\{\hat{I}_k\}_{k\geq 0}$.

6.1.2. Obtaining the transition probabilities of $\{\Theta_k(\omega)\}$ [FRMC-PA steps A1-A5]

Once the model to predict the voltage in Equations (6.1)-(6.2) has been built and the future usage profile has been characterized, the degradation process can be represented with the non-homogeneous Markov chain proposed in Chapter 3. For this purpose, the particle-filter-based algorithm detailed in Appendix C was used to compute the sequence $\{p_k^L\}_{k=t_p...t_p+P_h}$ in Eq. (3.16).

In this particle-filtering-based implementation, the prognostic horizon was fixed at $P_h = 1200$ steps, corresponding to $1200[s]$. The number of particles, on the other hand, corresponds to $N_p = 3000$. Particles help to characterize the uncertainty associated with the future evolution in time of the battery terminal voltage V_k, since the failure event is defined by a condition where this voltage drops below a given threshold. Every particle x_k^i is associated to a weight w_k^i and an input of the GRU $\hat{g}$, Γ_k^i, defined in Eq. (6.3). This is important due to the need for the previous input of the $\hat{g}$ to define the next voltage value.

To propagate each of the particles, a usage profile must be simulated for each. As a probabilistic characterization of the usage profile is available ($\{\hat{I}_k\}_{k\geq 0}$), the following relationship given by the law of total probabilities was utilized. $\forall j \geq 0$:

$$P(V_{k+j}|V_k,\Gamma_k) = \int P(V_{k+j}|V_k,\Gamma_k,\hat{I}_{k+1...k+j}) \cdot P(\hat{I}_{k+1...k+j}|V_k,\Gamma_k) d\hat{I}_{k+1...k+j} \tag{6.4}$$

$$\approx \sum_{i=1}^{N_p} P(V_{k+j}|V_k,\Gamma_k,\hat{I}^i_{k+1...k+j}) \cdot P(\hat{I}^i_{k+1...k+j}|\Gamma_k), \tag{6.5}$$

$$\approx \frac{1}{N_p} \sum_{i=1}^{N_p} P(V_{k+j}|V_k,\Gamma_k,\hat{I}^i_{k+1...k+j}), \tag{6.6}$$

where $P(\hat{I}_{k+1...k+j}|V_k,\Gamma_k) = P(\hat{I}_{k+1...k+j}|\Gamma_k)$ since the future usage profile does not depend on the present voltage. The integral in (6.4) is defined over all feasible usage profiles given Γ_k, which provides the initial state of the Markov chain $\{\hat{I}_{k+j}\}_{j\geq 0}$ that characterizes the discharge current profile. In practice, calculating this integral is computationally expensive; hence, the approximation in (6.5) can be used, where a future current trajectory is sampled from the $\{\hat{I}_{k+j}\}_{j\geq 0}$ for every particle. Moreover, if N_p is sufficiently large, each sampled Markov chain trajectory has a similar probability of $P(I^i_{k+1...k+j}|\Gamma_k) \approx 1/N_p$ since every particle has the same weight. Therefore, the expression can be reduced to the sum in (6.6). The propagation process of the particles is illustrated in Figure 6.3.

In this case study, given that the sensors were sufficiently precise, it was supposed that

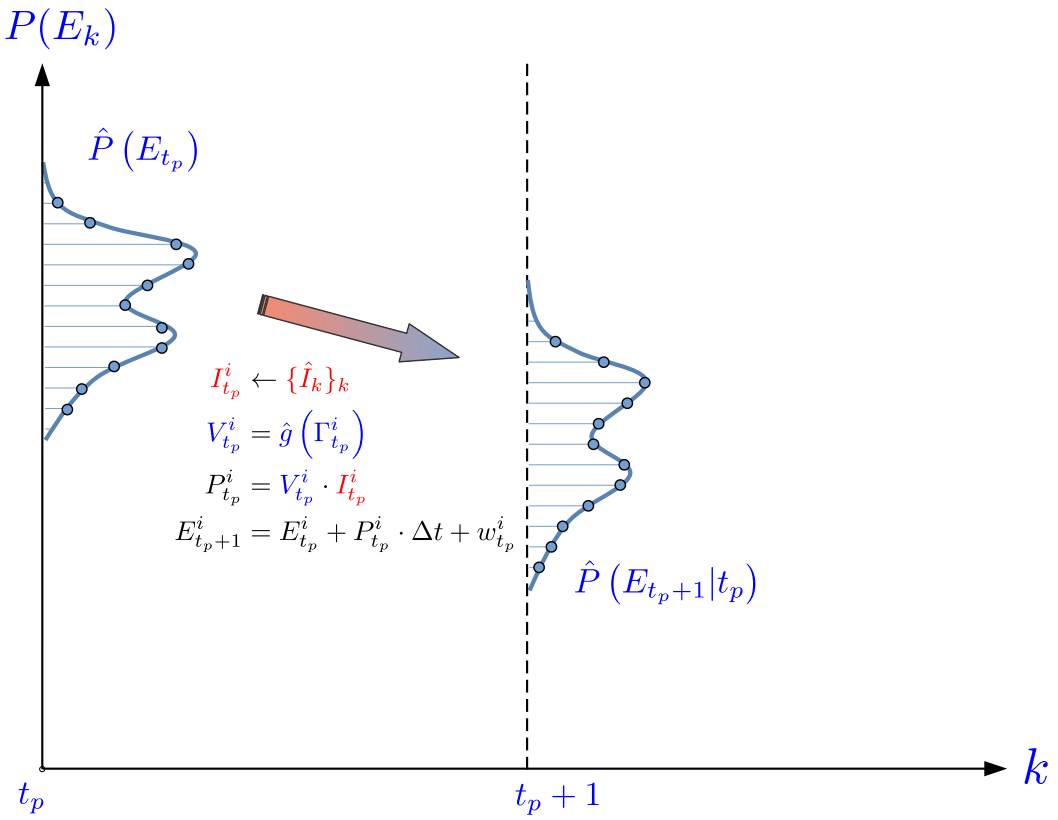

Figure 6.3: Illustration of the particles propagation for the FRMC-PA step A3 in the battery discharge case study. A current profile is sampled from the Markov chain $\{\hat{I}_k\}_k$. Then, the input Γ_{t_p} is updated to predict the voltage V_{t_p} using the GRU $\hat{g}$. Finally, the energy is propagated using the system dynamics in Eq. (6.1).

there is no uncertainty in the initial condition for the prognostic. This means that, if prognostics are executed at time k, the measurement V_k and the input (Γ_k) were considered known. This implies that there is no need to include a prior sequence $\{p^L_k\}_{k=t_p \ldots t_p+N_{prior}}$ as an input of the function φ, because there is no uncertainty in the initial condition and the sensors measurements are enough to estimate the posterior transition probabilities with reasonable precision (steps A1, A2 of the FRMC-PA are avoided).

Under the above considerations, particles were propagated from an initial condition (step A3) and then the transition probabilities $\{p^L_{k|t_p}\}_{k=t_p \ldots t_p+P_h}$ were computed using Eq. (3.16) (step A4). For illustration, let consider an initial condition of $E_{t_p} = 1031666.5[J]$ and a measurement of $V_{t_p} = 34.51[V]$. The obtained sequence estimate is shown in Figure 6.4. Finally, the triplet $(V_{t_p}, \Gamma_{t_p}, \{p^L_{k|t_p}\}_{k=t_p \ldots t_p+P_h})$ was gathered as a data point (step A5) for the training process of φ.

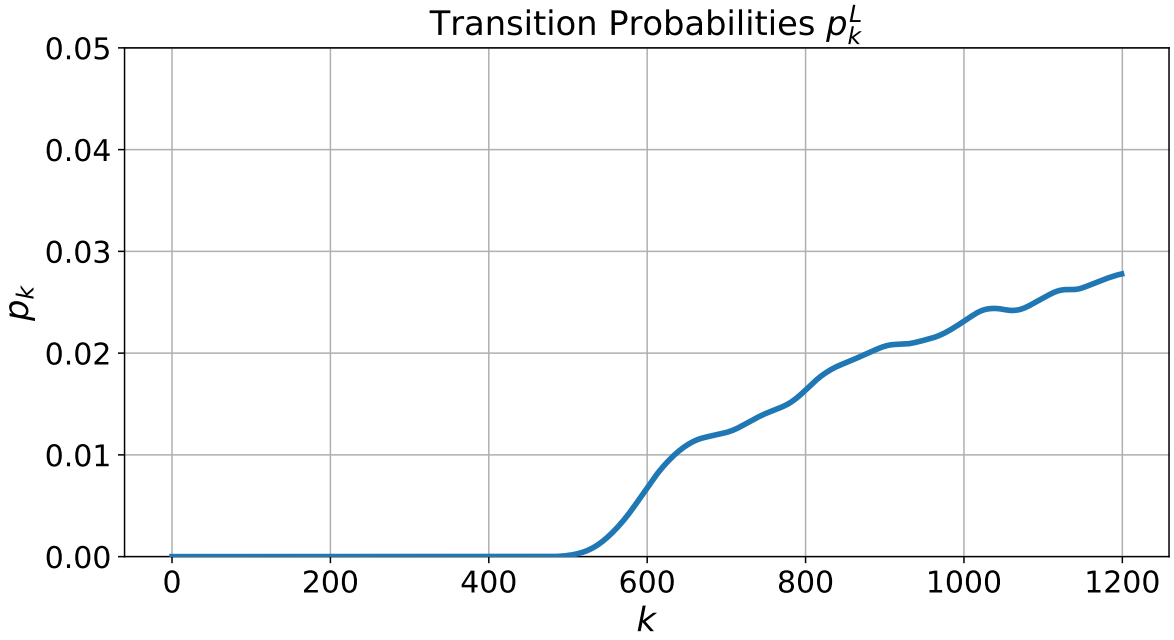

Figure 6.4: Transition probabilities $p^L_{k|t_p}$ in the battery discharge case study, considering an initial condition $E_{t_p} = 1031666.5[J]$ and a measurement of $V_{t_p} = 34.51[V]$.

6.1.3. Training φ [FRMC-PA steps A6-A7]

In the previous subsection, a posterior sequence estimate of the transition probabilities was generated based on an initial condition given by a voltage measurement V_{t_p} and an input Γ_{t_p}. To do this, particle-filtering-based approach was used. Hence, it was necessary to propagate the system dynamics given by Eqs. (6.1)-(6.2) for several simulated realizations of the discharge process. This implies a high offline computational burden, because the system state is propagated over a large prognostic horizon using the built model $\hat{g}$. Therefore, this subsection explains the steps to learn the function φ that maps (V_{t_p}, Γ_{t_p}) to $\{p^L_{k|t_p}\}_{k=t_p \ldots t_p+P_h}$ directly.

To learn the regression model φ, a Recurrent Neural Network (RNN) was utilized since there are temporal features between the variables we want to learn, as explained in Chapter 5. In particular, a Gated Recurrent Unit (GRU) was implemented [59].

The GRU input considered the voltage measurement V_{t_p} and the last input values from Eq. (6.3), given by $(I_{t_p}, E_{t_p-1}, P_{t_p-1}, \Delta I_{t_p}, \Delta V_{t_p-1})$. Thus:

$$\varphi : \left(I_{t_p}, E_{t_p-1}, P_{t_p-1}, \Delta I_{t_p}, \Delta V_{t_p-1}, V_{t_p}\right) \mapsto \{\hat{p}^L_{k|t_p}\}_{k=t_p \ldots t_p+P_h}. \tag{6.7}$$

Then, given measurements of the sensor, the GRU φ returns a sequence of length $P_h =$

1200. Hence, a many-to-many structure is considered. To train the GRU, $N_{data} = 2412$ data pairs were generated considering different simulated initial conditions, as explained in Section 4 (step A6). By doing this, the following data set was obtained:

$$\mathcal{D} = \left\{ \left(\left(I_{t_p}, E_{t_p-1}, P_{t_p-1}, \Delta I_{t_p}, \Delta V_{t_p-1}, V_{t_p} \right)^j, \{p^L_{k|t_p}\}^j_{k=t_p \dots t_p+P_h} \right) : j \in \{1 \dots N_{data}\} \right\}. \quad (6.8)$$

This data set was then used to train the GRU φ (step A7). Initially, the GRU cell has an input size of 6 (see Eq. (6.7)). However, after trying different configurations, it was determined that just considering the energy input is more adequate for the learning process. The other variables did not provide relevant information, and made the learning process more difficult. Additionally, a hidden size of 200 was considered, using $ReLU$ as the activation function. After the GRU, a dense-connected layer was added to obtain an output of size 1. The cell is iterated to obtain a total $P_h = 1200$ output size. The data set in Eq. (6.8) was split in 70 % train and 30 % test, using a batch size of 128. 20 epochs were considered using an Early Stopping criterion. The optimization algorithm was Adam.

To evaluate the model, the mean sequence-by-sequence RMS error presented in Chapter 5 was calculated. Formally, in this case:

$$RMSE_\varphi \triangleq \frac{1}{N_{data}} \sum_{j=1}^{N_{data}} \left\| \left(p^L_{k|t_p} \right)^j_{k=t_p \dots t_p+P_h} - \varphi \left(\left(I_{t_p}, E_{t_p-1}, P_{t_p-1}, \Delta I_{t_p}, \Delta V_{t_p-1}, V_{t_p} \right)^j \right) \right\|_2 . \quad (6.9)$$

Using this approach, a mean test sequence-by-sequence RMS error of 0.075 was obtained for the transition probabilities sequences. On the other hand, for each sequence $\{p^L_{k|t_p}\}^j_{k=t_p \dots t_p+P_h}$ and its estimation using φ, the corresponding ToF distribution was calculated from Eq. (3.12). In this case, the sequence-by-sequence RMS error was also calculated, obtaining a test RMS error of 0.01533 for the time of failure distribution.

Once the regression model has been trained, it can be utilized during the online stage. For the sake of clarity, Table 6.2 summarizes the different statistical models obtained through this section.

Type of space	State space		Reduced representation	
Model notation	$\hat{g}$	$\{\hat{I}_k\}_k$	φ	$\{\Theta_k\}_k$
Type of model	GRU	Homogeneous Markov chain	GRU	Non-homogeneous Markov chain
Model function	Voltage prediction	Usage profile characterization	To obtain transition probabilities	To represent the failure phenomenon

Table 6.2: Summary of the models obtained during the offline stage. State-space models are used to train the reduced-representation models, which are utilized during the online stage.

6.1.4. Analysis of the offline stage computational burden

The offline stage is super heavy in terms of computational burden. Indeed, as the offline stage intends to train a regression model, it needs to generate a big data set beforehand. For this, parallelization methods were used to make the synthetic generation of state trajectories more efficient. The synthetic trajectories were generated in groups of 120 data pairs $\left(\left(I_{t_p}, E_{t_p-1}, P_{t_p-1}, \Delta I_{t_p}, \Delta V_{t_p-1}, V_{t_p}\right), \{p^L_{k|t_p}\}_{k=t_p \ldots t_p+P_h}\right)$. The simulation of each group took approximately 23000 seconds. Since the total amount of data pairs was roughly 2400, 20 groups were generated. Therefore, the total computational time of the data set generation was approximately 128 hours.

On the other hand, the time used to train the regression model φ was 720 seconds, or 12 minutes. This is negligible compared with the time used to generate the data set. In summary, the offline stage took roughly 128 hours.

6.2. Online stage

As explained in Chapter 4, in the online stage the model φ is used to execute the prognostic task during the real-time operation of the system. To test the performance of the FRMC prognostic algorithm, the last portion of the bicycle circuit #5 was considered, starting from $T_p = 18230[s]$ (within data set 5). In this case, there is no need for a prior transition probabilities sequence since the model φ do not take prior sequences as arguments (steps B1, B2 are avoided). Then, at every time $t_p \geq T_p$, new measurements of V_{t_p} and I_{t_p} are received. The other variables are updated step-by-step using Eq. (6.1) to gather the inputs of φ in (6.7), evaluating them to obtain the posterior sequence estimate $\{\hat{p}^L_{k|t_p}\}_{k=t_p \ldots t_p+P_h}$ (step B5). Finally, the ToF distribution is updated using (3.12) (step B6). This process is executed every sample time, in this case, every second.

Figure 6.5 shows the performance of the FRMC-PA. For every time instant, the RUL distribution was estimated. Then, the mean value and the intervals of confidence of 30 % and 80 % were computed. This intervals are compared with the Alpha-Lambda performance index using $\alpha = 15\,\%$ [65]. λ is the proportion of time from $T_p = 18230[s]$ to reach the ground truth EoD time, corresponding to $T_{failure} = 19428[s]$. Specifically, $\lambda = (t_p - T_p)/(T_{failure} - T_p)$.

One important aspect to emphasize from the obtained results is the remarkable accuracy of the obtained RUL estimates that can be seen in Figure 6.5. It is worth pointing out that there is a small bias on the estimate of the true RUL. but most of the time the true RUL is underestimated. In other words, the algorithm is underestimating the EoD time. This bias

is not problematic because it makes the algorithm more robust to unexpected changes in the discharge current demand. It would be more problematic to overestimate the EoD time because, in that case, the failure condition could be reached without preventing it adequately. On the other hand, the uncertainty of the ToF distribution reduces slightly in time. However, this reduction is not enough to maintain the confidence interval within the cone given by α.

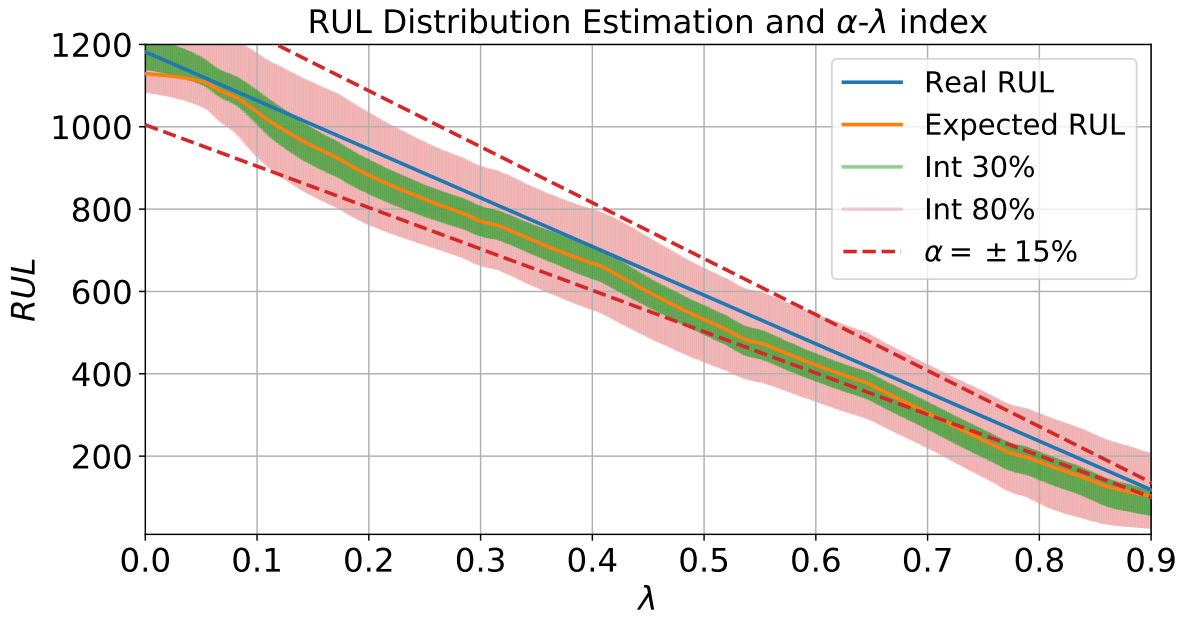

Figure 6.5: Alpha-Lambda performance index for the prognostic algorithm. $\alpha = 15\,\%$. $\lambda = (t - T_p)/(T_{failure} - T_p)$. The expected RUL and the intervals of confidence of 30 % and 80 % were added.

6.2.1. Analysis of the online stage computational burden

Once the quality of the prognostics has been checked with the Alpha-Lambda index, the computational burden of the FRMC-PA is evaluated. Every simulation was executed in the CPU of a Laptop computer with an Intel i5 2.5$[GHz]$ processor and 16$[GB]$ of RAM. The proposed method took 0.215 seconds in average every time the ToF distribution was updated, which is less than the sample period. Hence, it is possible to update the prognostics every time a new measurement arrives in a real-time operation.

Additionally, we compared the FRMC-PA with a prognostic module strategy based on Monte-Carlo simulation –called Monte-Carlo based Prognostic Algorithm (MC-PA). Each Monte-Carlo simulation is initiated with the same initial condition at time t, while the exogenous input of the battery (the battery discharge current) is computed as a realization of the model $\{\hat{I}_k\}_k$ that characterizes future discharge profiles. Then, the initial battery state is propagated over time based on Eqs. (6.1)-(6.2) until the predicted voltage drops below the

cut-off voltage. This simulation is iterated N_r times, obtaining N_r EoD times (one for each simulation), which are used to empirically approximate the EoD time distribution.

For this particular implementation, $N_r = 1000$ was chosen. This value was selected after an analysis of the obtained ToF probability distributions for $N_r \in \{10, 50, 100, 200, 500, 800, 1000, 1500, 2000, 2500, 3000\}$, some of which are shown in Fig. 6.6. Each distribution was compared with the Monte-Carlo simulation under $Nr = 10000$ (namely $MC(10000)$) using the Kullback-Leibler Divergence [66]. This comparison is shown in Fig. 6.7. While, in general, the divergence decreases with N_r (which means that the empirical distribution is more similar to $MC(10000)$), executing more than 1000 simulations does not show significant improvements.

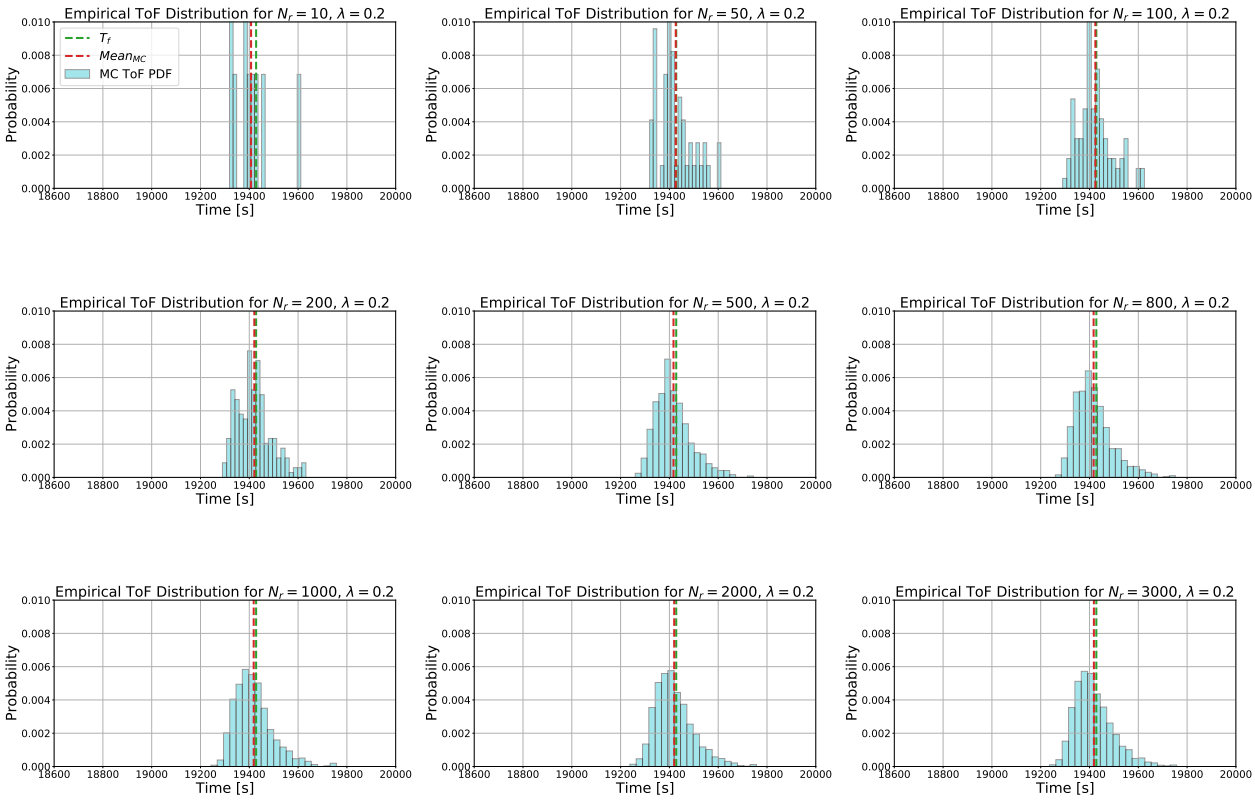

Figure 6.6: Empirical ToF distribution obtained with the Monte-Carlo-based prognostic module for different numbers of simulations (N_r).

To compare this thesis proposal with the Monte-Carlo prognostic module strategy, different values of $\lambda \in \{0.2, 0.4, 0.6, 0.8\}$ were considered, representing different time instants when the prognostic task is executed. Fig. 6.8 shows the corresponding distributions obtained with both methods. On this, the MC-PA approximates the ToF probability distribution using a histogram. On the other hand, the FRMC-PA computes the complete probability mass function of the ToF since it is based on an estimation of the transition probabilities (see

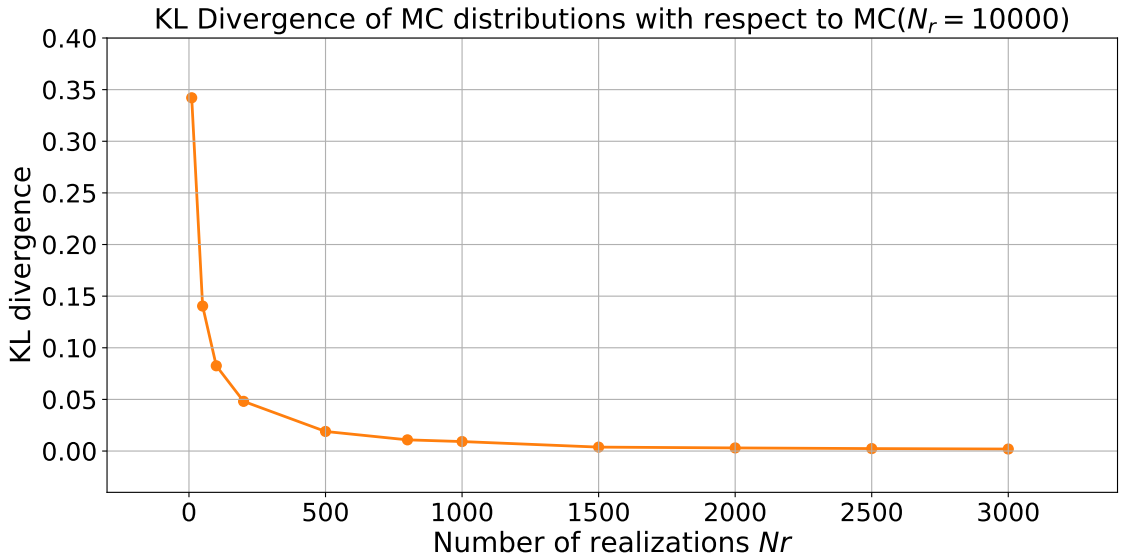

Figure 6.7: Analysis of the number of simulations for the Monte-Carlo-based prognostic module using the KL divergence. While, in general, the quality of the approximation increases with N_r, executing more than 1000 simulations does not show significant improvements.

Eq.(3.12)). Fig. 6.8 also shows the ground truth value of the time of failure (T_f), the mean value obtained with the Monte-Carlo prognostic module ($Mean_{MC}$), and the mean value obtained with the FRMC-PA ($Mean_{FRMC}$).

Complementing the results, Table 6.3 summarizes the comparison, where the expected values, the standard deviations, the sum of the obtained distributions, and the computation times of both methods were calculated .

In general, it is observed from Figure 6.8 that the proposed approach is very competitive in the sense that it offers an accurate representation of the ToF distribution for different regimes of λ when compared with the MC empirical estimation of that distribution. As seen in Figure 6.8 and Table 6.3, if the true value of the ToF is considered as a reference, the Monte-Carlo simulation is only marginally superior in terms of the expected value (bias) and the uncertainty (variance) in the prediction (see Figure 6.8). This indicates that the uncertainty of the prognostics observed in Figure 6.5 is due to the nature of the system itself –for instance, the uncertainty of the future usage profile–, and not an exclusive problem of the FRMC-PA.

However, this marginal gain of MC with respect to the proposed solution comes at a very high computationally cost. In fact, the gain of the FRMC-PA regarding the computational burden is remarkable, as we can be seen in Table 6.3. While the Monte Carlo simulation requires at least 150 seconds –2.5 minutes– to estimate the ToF distribution, the FRMC-PA completes this task in less than 1 second. This corresponds to a reduction of the computation

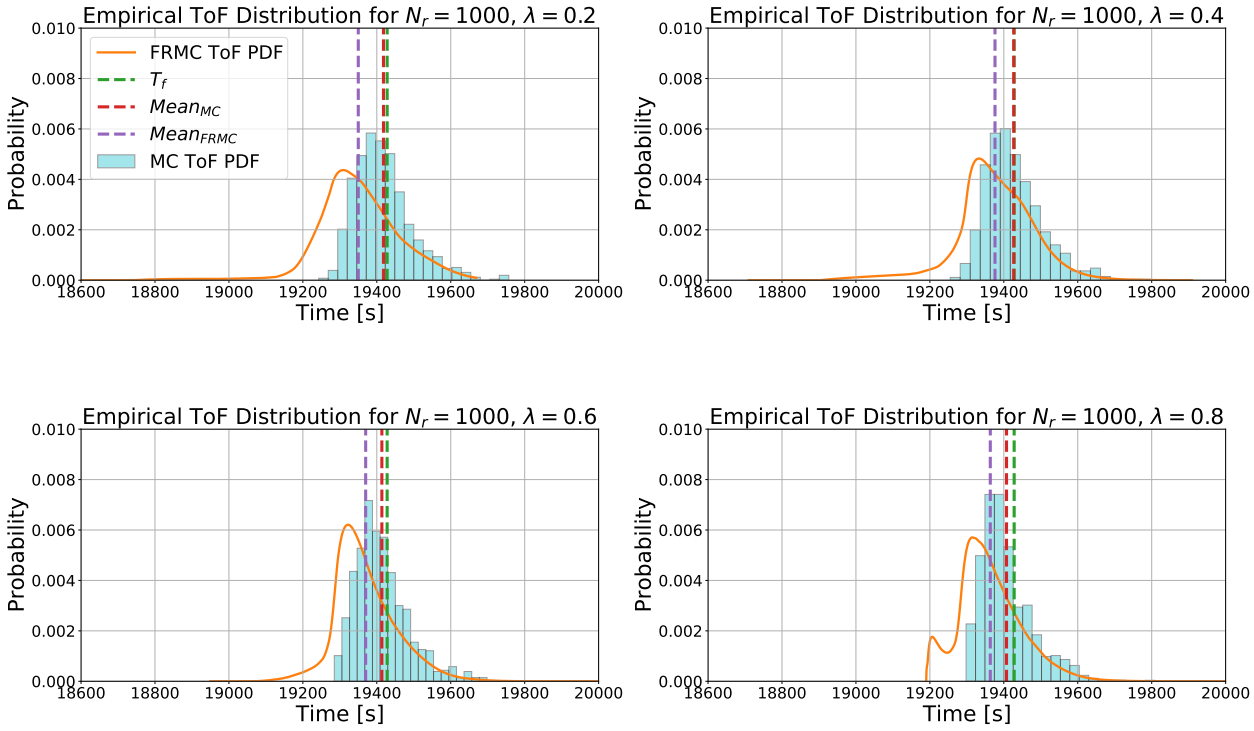

Figure 6.8: Comparison of the ToF probability distributions obtained with our proposal (FRMC) and a Monte-Carlo-based prognostics (MC), considering different values of λ. The ground truth value of the time of failure (T_f), the mean value obtained with the Monte-Carlo prognostic module ($Mean_{MC}$), and the mean value obtained with our proposal ($Mean_{FRMC}$) are also illustrated.

time of more than 99 % during the online stage. This impressive reduction is because the proposed method transferred most of the computational burden to the offline learning stage, leaving few computations to the online stage.

Ground truth	19428[s]							
λ	0.2		0.4		0.6		0.8	
Methodology	MC	FRMC	MC	FRMC	MC	FRMC	MC	FRMC
Expected value [s]	19418.1	19349.9	19426.8	19376.1	19413.7	19369.9	19407.3	19363.4
Standard Deviation [s]	77.808	111.730	76.350	106.743	71.943	84.539	69.901	86.044
Sum of the distribution	1.000	0.995	1.000	0.999	1.000	0.999	1.000	1.000
Computation time [s]	196.4366	0.173	177.844	0.125	167.939	0.126	154.907	0.127

Table 6.3: Comparison of the time of failure distributions obtained using the MC-PA and the FRMC-PA, considering different values of λ. The expected values, the standard deviations, the sum of the distributions and the computation times were compared. The FRMC-PA achieves a remarkable reduction of the computational burden of the prognostic task without significantly reducing the prognostic quality.

In summary, the proposed method achieves a massive reduction in the computational burden of the prognostics during the real-time operation of the system. The drawback is that

the quality of the prediction is slightly deteriorated compared with a Monte-Carlo simulation in terms of accuracy and precision. Based on this, a proposed strategy is to combine both approaches, using the FRMC-PA to execute the prognostic task during most of the real-time operation of the system. This way, computational efforts can be saved while obtaining an adequate prognostic quality. Then, when the system is near the failure event, and the prognostic accuracy becomes more critical, the prognostic methodology can switch to the Monte-Carlo prognostic module until the failure occurs or the operation of the system is interrupted. The determination of the optimal moment to switch between both approaches will be part of future research efforts.

Chapter 7

Conclusions

This thesis proposed a change of paradigm to address the problem of computational burden in failure prognostics. Instead of focusing our attention on the state-space trajectory (as conventional approaches), a novel degradation process representation using a two-states non-homogeneous Markov chain was proposed, which is entirely characterized by its transition probabilities. It was shown that this new approach is equivalent to studying the state-space trajectory if the focus is on characterizing the ToF distribution. Based on this model, the Fast-Running Markov Chain based Prognostic Algorithm (FRMC-PA) was presented, which comprises an offline training stage and an online inference stage. The algorithm aims at transferring most of the computational cost associated with the prediction to the offline stage, leaving few computations to the online stage.

FRMC-PA was implemented, tested and validated in a case study related to battery discharge prognostics. Obtained results show that the proposed strategy significantly reduces the computational cost during the real-time operation of the system, without substantially worsen the prognostic quality in terms of the estimated ToF distribution. Hence, this methodology is adequate for applications with limited on-board computational resources.

7.1. Future Work

Future research efforts will be oriented to explore the implementation of the FRMC-PA on more complex systems, with a higher dimensional state. This involves new challenges, including the offline computational cost of generating the training data set and the need for more sophisticated regression models to capture the transition probabilities.

If the previous implementation succeeds, the reduced representation of the degradation

process could enable the implementation of real-time failure prognostic schemes even in complex engineering systems, with dozens of components interacting. This would allow to scale up the quality and significance of the impact associated with the results already obtained by the PHM community in terms of the implementation of failure prognostic algorithms at a component level.

Other natural extensions of this work would be to obtain the transition probabilities from operational data directly, without training a state-space model for the system (as it was done in Section 6). This would accelerate the offline stage for degradation processes where monitoring data constitute the only available source of information of the system. A possible approach to achieve this is to use a deep learning method that fulfills the training of the degradation model, the estimation of the transition probabilities, and the prediction of the RUL directly.

On the other hand, developing the proposed strategy in systems with regeneration properties would also be challenging, since the Markovian property of the model holds under the absorbent condition. Additionally, it would be necessary to introduce *backward* transition probabilities, which would lead to a more complex model.

Finally, it would be interesting to explore the prognostic strategy proposed at the end of Chapter 6 and exploit the advantages of both the implemented methods: the efficiency of the FRMC-PA and the accuracy of the Monte Carlo-based technique. In normal conditions, it is preferable to use the FRMC-PA since it implies fewer computational resources. However, at some point –when the system is near the failure– it is relevant to obtain a more accurate estimation of the ToF distribution, which can be achieved using the Monte Carlo-based prognostic. Future research efforts will explore the optimal time to switch between both methods.

Bibliography

[1] M. Kordestani, M. Saif, M. Orchard, R. Razavi-Far, and K. Khorasani, "Failure prognosis and applications-A survey of recent literature," *IEEE Transactions on Reliability*, 2019.

[2] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley and Sons, 2007.

[3] M. Pecht, *Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things*. John Wiley and Sons, 2018.

[4] E. Balaban and J. Alonso, "An approach to prognostic decision making in the aerospace domain," *Annual Conference of the Prognostics and Health Management Society*, 2012.

[5] E. Balaban and J. Alonso, "A modeling framework for prognostic decision making and its application to uav mission planning," *Annual Conference of the Prognostics and Health Management Society*, 2013.

[6] G. Vachtsevanos, G. Georgoulas, and G. Nikolakopoulos, "Fault diagnosis, failure prognosis and fault tolerant control of aerospace/unmanned aerial systems," *24th Mediterranean Conference on Control and Automation (MED)*, June 2016.

[7] M. Orchard and G. Vachtsevanos, "A particle filtering-based framework for real-time fault diagnosis and failure prognosis in a turbine engine," *Proceedings of Mediterranean Conference on Control and Automation*, July 2007.

[8] M. Jouin, R. Gouriveau, D. Hissel, M. Péra, and N. Zerhouni, "Particle filter-based prognostics: review, discussion and perspectives," *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 2–31, May 2016.

[9] H. Rozas, F. Jaramillo, A. Perez, D. Kimenez, M. Orchard, and K. Medjaher, "A method for the reduction of the computational cost associated with the implementation of particle filter based failure prognostic algorithms," *Elsevier Mechanical Systems and Signal Processing*, 2020.

[10] D. Acuña, *Computation of time probability distributions for the occurrence of uncertain future events*. PhD thesis, Department of Electrical Engineering, Faculty of Mathematical and Physical Sciences, University of Chile, April 2020.

[11] D. Acuña, M. Orchard, and P. Wheeler, "Computation of time probability distributions for the occurrence of uncertain future events," *Elsevier Mechanical Systems and Signal Processing*, vol. 150, March 2021.

[12] S. Sankararaman and K. Goebel, "Why is the remaining useful life prediction uncertain?," *Annual Conference of the PHM Society 2013*, October 2013.

[13] S. Sankararaman, Y. Ling, and S. Mahadevan, "Uncertainty quantification and model validation of fatigue crack grow prediction," *Engineering Fracture Mechanics*, February 2011.

[14] S. Sankararaman, Y. Ling, C. Shantz, and S. Mahadevan, "Uncertainty quantification and model validation of fatigue crack grow prognosis," *International Journal of Prognostics and Health Management*, 2011.

[15] C. Farrar and N. Lieven, "Damage prognosis: the future of structural health monitoring," *Philosophical Transactions of the Royal Society A. Mathematical, Physical and Engineering Sciences*, 2007.

[16] A. Coppe, R. Haftka, N. Kim, and F. Yuan, "Uncertainty reduction of damage growth properties using structural health monitoring," *Journal of Aircraft*, 2010.

[17] J. Gu, D. Barker, and M. Pecht, "Uncertainty assessment of prognostics of electronics subject to random vibration," *IAAA Fall Symposyum - Technical Report*, 2007.

[18] H. Liao, W. Zhao, and H. Guo, "Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model," *Proceedings - Annual Reliability and Maintainability Symposium*, 2006.

[19] S. Engel, B. Gilmartin, K. Bongort, and A. Hess, "Prognostics, the real issues involved with predicting life remaining," *IEEE Aerospace Conference Proceedings*, 2000.

[20] J. Celaya, C. Kulkarni, G. Biswas, and K. Goebel, "Towards a model-based prognostics methodology for electrolytic capacitors: A case study based on electrical overstress accelerated aging," *International Journal of Prognostics and Health Management*, 2012.

[21] B. Olivares, M. C. Muñoz, M. Orchard, and J. Silva, "Particle-filtering-based prognosis framework for energy storage devices with a statistical characterization of state-of-health regeneration phenomena," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 2, pp. 364–376, 2013.

[22] D. Pola, H. Navarrete, M. Orchard, R. Rabié, M. Cerda, B. Olivares, J. Silva, P. Espinoza, and A. Pérez, "Particle-filtering-based discharge time prognosis for lithium-ion batteries with a statistical characterization of use profiles," *IEEE Transactions on Reliability*, vol. 64, June 2015.

[23] B. Saha and K. Goebel, "Uncertainty management for diagnostics and prognostics of

batteries using bayesian techniques," *IEEE Aerospace Conference Proceedings*, 2008.

[24] M. Daigle and K. Goebel, "Model-based prognostics under limited sensing," *IEEE Aerospace Conference Proceedings*, 2010.

[25] M. Daigle, A. Saxena, and K. Goebel, "An efficient deterministic approach to model-based prediction uncertainty estimation," *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, 2012.

[26] S. Sankararaman, M. Daigle, A. Saxena, and K. Goebel, "Analytical algorithms to quantify the uncertainty in remaining useful life prediction," *IEEE Aerospace Conference Proceedings*, 2013.

[27] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte-Carlo Methods in Practice*. Springer-Verlag New York, 2001.

[28] M. Orchard, M. Cerda, B. Olivares, and J. Silva, "Sequential monte carlo methods for discharge time prognosis in lithium-ion batteries," *International Journal of Prognostics and Health Management*, vol. 3, no. 2, 2012.

[29] Z. Chen, H. Sun, G. Dong, J. Wei, and J. Wu, "Particle filter-based state-of-charge estimation and remaining-dischargeable-time prediction method for lithium-ion batteries," *Journal of Power Sources*, vol. 414, pp. 158–166, Feb 2019.

[30] D. Acuña and M. Orchard, "Particle-filtering-based failure prognosis via sigma-points: application to lithium-ion battery state-of-charge monitoring," *Mechanical Systems and Signal Processing, Elsevier*, vol. 85, pp. 827–848, February 2017.

[31] E. Zio and G. Peloni, "Particle filtering prognostic estimation of the remaining useful life of nonlinear components," *Reliability Engineering and System Safety, Elsevier*, vol. 96, no. 3, pp. 403–409, 2011.

[32] C. Ley and M. Orchard, "Chi-squared smoothed adaptive particle-filtering based prognosis," *Mechanical Systems and Signal Processing*, vol. 82, pp. 148–165, Jan 2017.

[33] A. Sarathi, B. Long, and M. Pecht, "Prognostics methods for analog electronic circuits," *Annual Conference of the PHM Society*, vol. 4, no. 1, pp. 403–409, 2012.

[34] J. K. Kimotho, T. Meyer, and W. Sextro, "Pem fuel cell prognostics using particle filter with model parameter adaptation," *IEEE Conference on Prognostics and Health Management (PHM)*, pp. 1–6, 2014.

[35] M. Jouin, R. Gouriveau, D. Hissel, M. Péra, and N. Zerhouni, "Prognostics of pem fuel cell in a particle filtering framework," *International Journal of Hydrogen Energy*, vol. 39, pp. 481–494, 2014.

[36] D. Zhang, C. Cadet, C. Bérenguer, and N. Yousfi-Steiner, "Some improvements of particle filtering based prognosis for pem fuel cells," *IFAC-Papers On Line*, vol. 49, no. 28,

pp. 162–167, 2016.

[37] M. Jha, M. Bressel, B. O. Bouamama, G. Tanguy, M. Hilairet, and D. Hissel, "Particle filter based prognostics of pem fuel cell in bond graph framework," *Conférence Internationale des Energies Renouvelables*, 2020.

[38] Y. Hu, P. Baraldi, F. D. Maio, and E. Zio, "A particle filtering and kernel smoothing-based approach for new design components prognostics," *Reliability Engineering and System Safety, Elsevier*, vol. 134, pp. 19–31, February 2015.

[39] M. Yu, D. Wang, A. Ukil, V. Vaiyapuri, N. Sivakumar, N. Sivakumar, A. K. Gupta, and V. Nguyen, "Model-based failure prediction for electric machines using particle filter," *13th International Conference on Control Automation Robotics Vision (ICARCV)*, pp. 1811–1816, February 2014.

[40] H. Skima, K. Medjaher, C. Varnier, E. Dedu, J. Bourgeois, and N. Zerhouni, "Fault prognostics of micro-electro-mechanical systems using particle filtering," *IFAC-PapersOnLine*, vol. 49, pp. 226–231, 12 2016.

[41] L. Saidi, J. B. Ali, E. Bechhoefer, and M. Bendouzid, "Particle filter-based prognostic approach for high-speed shaft bearing wind turbine progressive degradations," *Proceedings of Annual Conference of the IEEE Industrial Electronics Society*, Dec 2017.

[42] H. C. Vu, P. Do, B. Iung, and F. Peysson, "A case study on health prediction of an industrial diesel motor using particle filtering," *Prognostics and System Health Management Conference, PHM-Harbin*, Oct 2017.

[43] M. Barbieri, K. Nguyen, R. Diversi, K. Medjaher, and A. Tilli, "Rul prediction for automatic machines: a mixed edge-cloud solution based on model-of-signals and particle filtering techniques," *Journal of Intelligent Manufacturing*, November 2020.

[44] C. Chen, G. Vachtsevanos, and M. Orchard, "Machine remaining useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach," *Mechanical Systems and Signal Processing, Elsevier*, vol. 28, pp. 597–607, April 2012.

[45] J. Fan, K. Yung, and M. Pecht, "Predicting long-term lumen maintenance life of led light sources using a particle filter-based prognostic approach," *Expert Systems with Applications*, vol. 42, pp. 2411–2420, Apr 2015.

[46] F. Tamssaouet, K. Nguyen, K. Medjaher, and M. Orchard, "A contribution to online system-level prognostics based on adaptive degradation models," *PHM Society European Conference*, vol. 5, July 2020.

[47] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, February 2002.

[48] C. Sbarufatti, M. Corbetta, M. Giglio, and F. Cadini, "Adaptive prognosis of lithium-ion batteries based on the combination of particle filters and radial basis function neural networks," *J. Power Sources*, vol. 344, pp. 128–140, 2017.

[49] Y. Chang and H. Fang, "A hybrid prognostic method for system degradation based on particle filter and relevance vector machine," *Realiab. Eng. Syst. Saf.*, vol. 186, pp. 51–63, Jun 2019.

[50] W. Yan, B. Zhang, X. Wang, W. Dou, and J. Wang, "Lebesgue sampling-based diagnosis and prognosis for lithium -ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 63, March 2016.

[51] W. Yan, B. Zhang, G. Zhao, J. Weddington, and G. Niu, "Uncertainty management in lebesgue-sampling-based diagnosis and prognosis for lithium-ion battery," *IEEE Transactions on Industrial Electronics*, vol. 64, pp. 8158–8166, Oct 2017.

[52] W. Yan, B. Zhang, W. Dou, D. Liu, and Y. Peng, "Low-cost adaptive lebesgue sampling particle filtering approach for real-time li-ion battery diagnosis and prognosis," *IEEE Transactions on Automation Science and Engineering*, vol. 14, pp. 1601–1611, Oct 2017.

[53] O. Geramifard, J.-X. Xu, J.-H. Zhou, and X. Li, "A physically segmented hidden Markov model approach for continuous tool condition monitoring: Diagnostics and prognostics," *IEEE Transactions on Industrial Informatics*, vol. 8, November 2012.

[54] D. Tobon-Mejia, K. Medjaher, N. Zerhouni, and G. Tripot, "A data-driven failure prognostics method based on mixture of Gaussians hidden Markov models," *IEEE Transactions on Reliability*, vol. 61, June 2012.

[55] D. Tang, J. Cao, and J. Yu, "Remaining useful life prediction for engineering systems under dynamic operational conditions: a semi-Markov decision process-based approach," *Chines Journal of Aeronautics*, vol. 32, September 2018.

[56] J. Chiachío, M. Jalón, M. Chiachío, and A. Kolios, "A markov chains prognostics framework for complex degradation processes," *Elsevier Reliability Engineering and System Safety*, vol. 195, March 2020.

[57] Z.-X. Zhang, X.-S. Si, C.-H. Hu, and M. G. Pecht, "A prognostic model for stochastic degrading systems with state recovery: application to li-ion batteries," *IEEE Transactions on Reliability*, vol. 66, December 2017.

[58] J. Norris, *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics, 1997.

[59] A. N. Shewalkar, "Comparision of RNN, LSTM and GRU on speech recognition data," *North Dakota State University of Agriculture and Applied Science*, October 2018.

[60] C. Díaz, V. Quintero, A. Pérez, F. Jaramillo, C. Burgos-Mellado, M. E. Orchard, D. Sáez,

and R. Cárdenas, "Particle-filtering-based prognostics for the state of maximum power available in lithium-ion batteries at electromobility applications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7187–7200, 2020.

[61] H. Rozas, D. Troncoso-Kurtovic, C. P. Ley, and M. E. Orchard, "Lithium-ion state-of-latent-energy (sole): A fresh new look to the problem of energy autonomy prognostics in storage systems," *Elsevier Journal of Energy Storage*, 2021.

[62] E. Chemali, P. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, "Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 65, August 2018.

[63] F. Yang, W. Li, C. Li, and Q. Miao, "State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network," *Energy*, vol. 175, pp. 66–75, 2019.

[64] H. Chaoui and C. C. Ibe-Ekeocha, "State of charge and state of health estimation for lithium batteries using recurrent neural networks," *IEEE Transactions on Vehicular Technology*, vol. 66, October 2017.

[65] L. Tang, M. Orchard, K. Goebel, and G. Vachtsevanos, "Novel metrics and methodologies for the verification and validation of prognostic algorithms," *IEEEAC*, January 2011.

[66] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, John Wiley and Sons, 2006.

[67] F. Jaramillo, M. Valderrama, V. Quintero, A. Pérez, and M. Orchard, "Time-of-failure probability mass function computation using the first-passage-time method applied to particle filter-based prognostics," *Annual Conference of the PHM Society*, vol. 12, November 2020.

Appendix A

Discrete-time Markov Chains

Definition 1 (Stochastic Process) Let I be a countable set. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A *stochastic process* indexed by $\Lambda \subseteq [0,\ \infty)$ is a family of random variables $X = (X_\lambda)_{\lambda \in \Lambda}$, where:

$$X_\lambda : \Omega \to I$$

is $\mathcal{F} - \mathcal{P}(I)$-measurable, $\forall \lambda \in \Lambda$. If Λ is a countable set, $(X_\lambda)_{\lambda \in \Lambda}$ is a *discrete-time stochastic process*.

Notation. P denotes the probability measure induced by $\mathbb{P}$ on I. This means that, for $A \subseteq I$:

$$P(X_\lambda \in A) = \mathbb{P}(X_\lambda^{-1}(A)) = \mathbb{P}\left(\{\omega \in \Omega : X_\lambda(\omega) \in A\}\right). \tag{A.1}$$

Definition 2 (Discrete-time Markov Chain) A *discrete-time Markov Chain* of order 1 with values on I is a discrete-time stochastic process $X = (X_n)_{n \in \mathbb{N}}$ that satisfies:

$$P(X_{n+1} = i_{n+1} | X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) = P(X_{n+1} = i_{n+1} | X_n = i_n), \tag{A.2}$$

$\forall i_0, i_1, \ldots, i_{n+1} \in I$ with $P(X_0 = i_0, X_1 = i_1, \ldots X_n = i_n) > 0$.

In other words, in a Markov chain stochastic process the future is independent of the past, conditional on the present. (A.2) is known as the *Markov property*. A Markov chain also satisfies that:

$$P(X_{n+1} \in A | X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) = P(X_{n+1} \in A | X_n = i_n),\ \forall A \subseteq I. \tag{A.3}$$

Definition 3 Let X be a discrete-time Markov chain with values on I. Let $\mu = (\mu_i)_{i \in I}$ such

that:

$$\mu_i = P(X_0 = i), \tag{A.4}$$

Then μ is called the *initial distribution* of X. Let P^k be a matrix, $\forall k \in \mathbb{N}\backslash\{0\}$, such that $P^k = (p^k_{i,j})_{i,j\in I}$, with:

$$p^k_{i,j} = P(X_k = j | X_{k-1} = i). \tag{A.5}$$

Then P^k is called the *transition matrix* of X at time k.

Definition 4 A Markov chain is said to be *homogeneous* if $\forall n, m > 0$ and $\forall i, j \in I$:

$$P(X_n = j | X_{n-1}) = P(X_m = j | X_{m-1} = i). \tag{A.6}$$

In this case, $P^k = P^q$, $\forall k, q \in \mathbb{N}$, and the transition matrix of the Markov chain is simply noted by P.

Notation. If $(X_n)_{n\in\mathbb{N}}$ is an homogeneous Markov chain with initial distribution μ and transition matrix P, it is said that $(X_n)_{n\in\mathbb{N}}$ is $Markov(\mu, P)$.

Notation. For $i \in I$, $\delta_i = (\delta_{ij} : j \in I)$ is the *unit mass* at i, where:

$$\delta ij = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{A.7}$$

THEOREM 1 (Characterization of Markov chains) $X = (X_n)_{n\in\mathbb{N}}$ is a discrete-time Markov chain with initial distribution μ and transition matrices $(P^k)_{k\in\mathbb{N}}$ if and only if:

$$\forall n \geq 0\ ,\ \forall i_0, i_1, \ldots, i_{n+1} \in I :$$
$$P(X_0 = i_0, X_1 = i_1, \ldots, X_{n+1} = i_{n+1}) = \mu_{i_0} \cdot p^1_{i_0,i_1} \cdot p^2_{i_1,i_2} \ldots p^{n+1}_{i_n,i_{n+1}}. \tag{A.8}$$

Note that if X is an homogeneous Markov chain, the previous expression reduces to:

$$P(X_0 = i_0, X_1 = i_1, \ldots, X_{n+1} = i_{n+1}) = \mu_{i_0} \cdot p_{i_0,i_1} \cdot p_{i_1,i_2} \ldots p_{i_n,i_{n+1}}. \tag{A.9}$$

Definition 5 (Stopping Time) A random variable $\tau : \Omega \to \{0, 1, 2, \ldots\} \cup \{\infty\}$ is called a *stopping time* if, for $n \geq 0$, the event $\{\omega : \tau(\omega) = n\}$ depends only on $X_0, X_1, \ldots, X_n$.

Intuitively, by watching the process, it is possible to determine the time when τ occurs. In other words, the event $\{\omega : \tau(\omega) = n\}$ depends only on the past and the present, and not on the future.

THEOREM 2 (Strong Markov Property) Let $(X_n)_{n\in\mathbb{N}}$ be $Markov(\mu, P)$. Let τ be a stopping time of $(X_n)_{n\in\mathbb{N}}$. Then, conditional on $\tau < \infty$ and $X_\tau = i$, $(X_{\tau+n})_{n\in\mathbb{N}}$ is $Markov(\delta_i, P)$, and independent of $X_0, X_1, \ldots X_\tau$.

To review the proofs of Theorems 1 and 2, and get deeper into Markov chains and its properties, please refer to [58].

Appendix B

Proof of Propositions

B.1. Proof of Proposition 1

Given $i_0, \ldots, i_{k+1} \in \{0, 1\}$ such that $P(\Theta_0 = i_0, \Theta_1 = i_1, \ldots, \Theta_k = i_k) > 0$, the objective is to prove that:

$$P(\Theta_{k+1} = i_{k+1} | \Theta_0 = i_0, \ldots, \Theta_k = i_k) = P(\Theta_{k+1} = i_{k+1} | \Theta_k = i_k). \tag{B.1}$$

Given the assumption that it is not possible to have a catastrophic failure and then recover from it, $P(\Theta_0 = i_0, \Theta_1 = i_1, \ldots, \Theta_k = i_k) = 0$ if $\exists n_0, m_0 \in \{0, \ldots, k\}$ such that $n_0 < m_0$ and $i_{n_0} = 1,\ i_{m_0} = 0$. Then, those cases cannot be considered as conditioning events, by definition.

Considering the previous observation, the proof demonstrates the Markov property in two different cases: the system has failed before time $k+1$, or not. Then, it concludes on the full stochastic process.

- Consider the case where the system fails at time $n_0 < k+1$, i.e., $i_{n_0} = 1$. Then, all future states are failed ones. Hence:

$$P(\Theta_{k+1} = i_{k+1} | \Theta_0 = i_0, \ldots, \Theta_{i_{n_0}} = 1, \ldots, \Theta_k = \underbrace{i_k}_{1}) = \begin{cases} 1 & \text{, if } i_{k+1} = 1 \\ 0 & \text{, if } i_{k+1} = 0. \end{cases} \tag{B.2}$$

Note that in this case it is sufficient to observe the last failure state variable Θ_k. Indeed,

$\Theta_k = 1$ indicates that the system has already failed, whenever it happened. Then:

$$P(\Theta_{k+1} = i_{k+1}|\Theta_k = 1) = \begin{cases} 1 & \text{, if } i_{k+1} = 1 \\ 0 & \text{, if } i_{k+1} = 0. \end{cases} \tag{B.3}$$

Therefore, from (B.2) and (B.3):

$$P(\Theta_{k+1} = i_{k+1}|\Theta_0 = i_0, \ldots, \Theta_{i_{n_0}} = 1, \ldots, \Theta_k = 1) = P(\Theta_{k+1} = i_{k+1}|\Theta_k = 1). \tag{B.4}$$

Therefore, the Markov property holds in this case.

- Now, suppose that the system has not failed up to time k. This means that every previous failure state $\Theta_j = 0$, $\forall j \leq k$, due to the absorbent property of state "1". In this case, the next failure state Θ_{k+1} is not a deterministic variable, because the system may fail or not at that instant. Then, by definition:

$$P(\Theta_{k+1} = i_{k+1}|\Theta_0 = 0, \ldots, \Theta_k = 0) = \frac{P(\Theta_0 = 0, \ldots, \Theta_k = 0, \Theta_{k+1} = i_{k+1})}{P(\Theta_0 = 0, \ldots, \Theta_k = 0)}. \tag{B.5}$$

Both terms in the right side of (B.5) are analysed. By definition of conditional probability:

$$P(\Theta_0 = 0, \ldots, \Theta_k = 0) = \underbrace{P(\Theta_0 = 0, \ldots, \Theta_{k-1} = 0|\Theta_k = 0)}_{1} \cdot P(\Theta_k = 0) = P(\Theta_k = 0). \tag{B.6}$$

The conditional probability is 1 because of the absorbent property of the failure state "1". In other words, if the system has not failed at time k, then with probability 1 it has not failed before.

Similarly:

$$\begin{aligned} &P(\Theta_0 = 0, \ldots, \Theta_k = 0, \Theta_{k+1} = i_{k+1}) \\ &\quad = \underbrace{P(\Theta_0 = 0, \ldots, \Theta_{k-1} = 0|\Theta_k = 0, \Theta_{k+1} = i_{k+1})}_{1} \cdot P(\Theta_k = 0, \Theta_{k+1} = i_{k+1}) \\ &\quad = P(\Theta_k = 0, \Theta_{k+1} = i_{k+1}). \end{aligned} \tag{B.7}$$

Hence, combining Eqs. (B.5), (B.6) and (B.7):

$$\begin{aligned} P(\Theta_{k+1} = i_{k+1} | \Theta_0 = 0, \ldots, \Theta_k = 0) &= \frac{P(\Theta_0 = 0, \ldots, \Theta_k = 0, \Theta_{k+1} = i_{k+1})}{P(\Theta_0 = 0, \ldots, \Theta_k = 0)} \\ &= \frac{P(\Theta_k = 0, \Theta_{k+1} = i_{k+1})}{P(\Theta_k = 0)} \\ &= P(\Theta_{k+1} = i_{k+1} | \Theta_k = 0). \end{aligned} \tag{B.8}$$

Therefore, the Markov property holds also when the system has not failed previously.

It has been proved that in both cases where the condition event probability $P(\Theta_0 = i_0, \ldots, \Theta_k = i_k) \neq 0$, the Markovian property is satisfied, concluding that the process is a Markov chain of order 1. ■

B.2. Proof of Proposition 2

Considering that $P(\exists k_0 \in \mathbb{N}, X_{k_0}(\omega) \in L) = 1$, then $P(\tau^L(\omega) < \infty) = 1$. Hence, from the definition of the random variable $\tau^L(\omega)$:

$$\begin{aligned} P(\tau^L(\omega) = k) &= P(\min\{j \geq 1 : \Theta_j^L(\omega) = 1\} = k) = P(\{0\,0\,\ldots\,0\underset{\uparrow_k}{1}\,1\ldots\}) \\ &= P(\{0\,0\,\ldots\,0\underset{\uparrow_k}{1}\}) = P(\Theta_0 = 0, \ldots, \Theta_k^L = 1) \\ &= P(\Theta_k^L = 1 | \Theta_0^L = 0, \ldots \Theta_{k-1}^L = 0) \cdot P(\Theta_0^L = 0, \ldots \Theta_{k-1}^L = 0). \end{aligned}$$

Conditioning the second term iteratively and considering that the process is a Markov chain, then:

$$\begin{aligned} P(\tau^L(\omega) = k) &= P(\Theta_k^L(\omega) = 1 | \Theta_0^L = 0, \ldots \Theta_{k-1}^L = 0) \\ &\quad \cdot \prod_{j=1}^{k-1} P(\Theta_j^L = 0 | \Theta_0^L = 0, \ldots \Theta_{j-1}^L = 0) \cdot P(\Theta_0^L = 0) \\ &= \underbrace{P(\Theta_k^L = 1 | \Theta_{k-1}^L = 0)}_{p_k^L} \cdot \prod_{j=1}^{k-1} \underbrace{P(\Theta_j^L = 0 | \Theta_{j-1}^L = 0)}_{1-p_j^L} \cdot \underbrace{P(\Theta_0^L = 0)}_{1} \end{aligned}$$

This implies that:

$$P(\tau^L(\omega) = k) = p_k^L \cdot \prod_{j=1}^{k-1} (1 - p_j^L). \quad \blacksquare \tag{B.9}$$

B.3. Proof of Proposition 4

By definition, $\forall k \geq 1$:

$$\begin{aligned} p_k^L =& P\left(\Theta_k^L(\omega) = 1 | \Theta_{k-1}^L(\omega) = 0\right) \\ =& \frac{P\left(\Theta_k^L(\omega) = 1, \Theta_{k-1}^L(\omega) = 0\right)}{P\left(\Theta_{k-1}^L(\omega) = 0\right)}. \end{aligned}$$

$$\text{This implies that:} \quad p_k^L = \frac{P\left(X_k(\omega) \in L, (X_1(\omega), \dots X_{k-1}(\omega)) \in (L^c)^{k-1}\right)}{P\left((X_1(\omega), \dots X_{k-1}) \in (L^c)^{k-1}\right)} \quad ,$$

since the events:

$$\{\omega \in \Omega : \Theta_k^L(\omega) = 1, \Theta_{k-1}^L(\omega) = 0\} = \{\omega \in \Omega : X_k(\omega) \in L, (X_1(\omega), \dots X_{k-1}(\omega)) \in (L^c)^{k-1}\},$$
$$\text{and } \{\omega \in \Omega : \Theta_{k-1}^L(\omega) = 0\} = \{\omega \in \Omega : (X_1(\omega), \dots X_{k-1}(\omega)) \in (L^c)^{k-1}\}. \quad \blacksquare$$

Appendix C

Algorithmic Solutions to Compute Probability Distributions

Theoretically, it is possible to calculate the sequence $\{p_k^L\}_{k=t_p \ldots t_p+P_h}$ (prior or posterior) using Equation (3.16) or (4.2). However, in practice this is an expensive task because it implies the computation of multiple integrals. Alternatively, this task can be approximated either using Monte Carlo simulation or executing particle filter algorithms [7] to estimate the distributions $P(X_k(\omega))$, $k \in \{t_p, \ldots, t_p + P_h\}$. As explained in Chapter 2, particle filters are a class of algorithms proposed to obtain samples sequentially from a target state probability distribution $P(X_{t_p:k})$ [47]. This algorithms generate a set of $N_p \gg 1$ weighted particles $\{w_k^i, x_k^i\}_{k=t_p \ldots t_p+P_h}^{i=1 \ldots N_p}$, with $w_k^i \geq 0$ and $\sum_{i=1}^{N_p} w_k^i = 1$, $\forall k \geq t_p$, satisfying that [22]:

$$\sum_{i=1}^{N_p} w_k^i \psi_k \left(x_k^i\right) \xrightarrow[N_p \to \infty]{} \int \psi_k \left(x_k\right) P \left(x_k\right) dx_k, \tag{C.1}$$

in probability, where ψ_k is any P-integrable function. These particles are propagated through time using the system dynamics given by Eq. (3.1). Based on this, it is possible to obtain an empirical representation of the state distribution:

$$\hat{P}_{N_p}(X_k = x) \approx \sum_{i=1}^{N_p} w_k^i \cdot \delta \left(x - x_k^i\right). \tag{C.2}$$

As explained in Chapter 2, particle filters can also be used to estimate the posterior distribution of the state given a measurement y_k at some time k, updating weights based on the measurement likelihood obtained from (3.2). This way, $w_k^i \propto w_{k-1}^i \cdot P(y_k | x_k^i)$. Additionally, a resampling of the particles can be executed to keep diversity in the population and to

not concentrate the mass of the distribution on few particles. The resampling consists of selecting new particles from the current population randomly. Each particle x_k^i is selected with probability w_k^i. After this process, all the weights are reset to $w_k^i = 1/N_p$, $\forall i \in \{1 \dots N_p\}$ and the particles can continue their propagation [47].

Hence, whether measurements are available or not, it is possible to estimate the transition probabilities in (3.16) or (4.2). Note that, given an initial condition x_0 and an usage profile $\{u_j\}_{j=0}^{k-1}$, it is possible to generate a sequence of particles that approximate the state distribution trough time using Eq. (C.2). Based on this empirical distribution, both conditional probabilities on (3.16) are approximated. Both terms consider particles that *have not failed* until $k-1$, which means that the particle has never reached the failure set L before k. Formally, it is said that a particle i has not failed before k if $x_j^i \in L^c$, $\forall j < k$. Denote by $H_{k-1} \subseteq \{1, \dots, N_p\}$ the set of particles that have not failed before k. On the other hand, $F_k \subseteq H_{k-1}$ denotes the set of particles that have failed at k, but not before. Formally:

$$\begin{aligned} H_{k-1} &= \left\{ i \in \{1, \dots, N_p\} : x_j^i \in L^c \,, \forall j \leq k-1 \right\} \\ F_k &= \left\{ i \in \{1, \dots, N_p\} : i \in H_{k-1} \ \wedge \ x_k^i \in L \right\}. \end{aligned} \tag{C.3}$$

Using this, p_k^L can be approximated by the following expression [67]:

$$\hat{p}_k^L = \frac{\sum\limits_{i \in F_k} w_k^i}{\sum\limits_{i \in H_{k-1}} w_k^i} \leq 1. \tag{C.4}$$

In other words, $\hat{p}_k^L$ is the weighted proportion of particles that have not failed before k and fail at k. Using these approximations, it is clear that the particles that have already failed before k are not useful to calculate any future $\hat{p}_j^L$, $j \geq k$. Each particle is useful until it fails because after that it will not be used to estimate $\hat{p}_j^L$, $j \geq k$. Moreover, at some point there will not be enough useful particles to obtain and adequate estimation of p_k^L because most of the particles will be in the failure set. To address this under-sampling issue, a *major resampling* process is proposed.

The major resampling consists of selecting new particles from not failed ones when necessary. At time k, a major resampling is executed after calculating $\hat{p}_k^L$ if few non-failed particles remain for the next time step. A minimum proportion of non-failed particles $\eta_{mr} = N_{mr}/N_p$ is established. Hence, if $|H_k| < N_{mr}$, a major resampling is executed. In this process, each particle x_k^i , $i \in H_k$, is selected with probability $\dfrac{w_k^i}{\sum_{i \in H_k} w_k^i}$. After this, every weight is reset to $w_k^i = 1/N_p$ and every particle is a non-failed one. By doing this, the particle filter loses the

correct characterization of the system state distribution over time because particles are being forced to remain around the hazard zone to measure how many of them surpass the failure threshold at every time instant. It is worth pointing out that this is not an issue for the problem itself because the objective is to characterize the transition probabilities $\{p_k^L\}_{k=t_p \ldots t_p+P_h}$, instead of the future state probability distribution $\{P(X_k(\omega))\}_{k=t_p \ldots t_p+P_h}$.

The implementation of this process is summarized in Algorithm 1. For explanatory purposes, an given that the system is time-invariant, it is assumed that $t_p = 1$.

Algorithm 1: Particle filter with major resampling

Input : system dynamics $f(\cdot)$, observation relationship $g(\cdot)$, process model uncertainty distribution $\{W_k\}_{k\geq 0}$, observation noise distribution $\{V_k\}_{k\geq 0}$, failure set L, prognostic horizon N_{pred}, number of particles N_p, initial distribution of x_0, measurement y_1 (optional), usage profile probabilistic characterization $\{\hat{U}_k\}_{k\geq 0}$, minimum proportion of non-failed particles η_{mr}.

Output: prior transition probabilities $\{p_k^L\}_{k\geq 1}$, if y_1 was not included. Otherwise, posterior transition probabilities $\{p_{k|1}^L\}_{k\geq 1}$.

Initialization;
for *i = 1* **to** N_p **do**
  $x_0^i \rightsquigarrow$ Initial condition of particle i sampled from the prior distribution of x_0;
  $w_0^i \leftarrow \frac{1}{N_p}$
end
Recursion;
for $k = 1$ **to** N_{pred} **do**
  Particles propagation;
  for $i = 1$ **to** N_p **do**
    $u_{k-1}^i \rightsquigarrow$ Exogenous input sampled from the usage profile characterization $\{\hat{U}_k\}_{k\geq 0}$;
    $\omega_{k-1}^i \rightsquigarrow$ Process uncertainty sampled from W_{k-1};
    $x_k^i \leftarrow f(x_{k-1}^i, u_{k-1}^i, \omega_{k-1}^i)$;
    if $k = 1$ *and* y_1 *was included* **then**
      Weights update ;
      $w_1^i \propto w_0^i \cdot P(y_1|x_1^i) \leftarrow$ likelihood obtained from $g(\cdot)$ and the observation noise distribution V_1;
      Resampling ;
      $x_1^i \rightsquigarrow$ Sampled from particles $\{x_1^i\}$ with mass probability distribution $\{w_1^i\}$;
      $w_1^i \leftarrow \frac{1}{N_p}$
    end
    else
      Weights update ;
      $w_k^i \leftarrow w_{k-1}^i$
    end
  end
  p_k^L computation ;
  $p_k^L \leftarrow$ from Eq. (C.4) ;
  Major resampling ;
  if $|H_k| < \eta_{mr} \cdot N_p$ **then**
    $x_k^i \rightsquigarrow$ Sampled from particles $\{x_k^i : i \in H_k\}$ with mass probability distribution given by $\left\{ \frac{w_k^i}{\sum_{i\in H_k} w_k^i} : i \in H_k \right\}$
    $w_k^i \leftarrow \frac{1}{N_p}$
  end
end
return $\{p_k^L\}_{k\geq 1}$ or $\{p_{k|1}^L\}_{k\geq 1}$, as appropriate.