

Upscaling Prognostics for Aerospace Structures

Thesis

Oscar Carpentier

Upscaling Prognostics for Aerospace Structures

Thesis

by

Oscar Carpentier

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Wednesday Octobre 23, 2024 at 09:30 AM.

Student number: 4773624

Project duration: September 15, 2023 – Octobre 23, 2024

Thesis committee: Associate Prof. dr. ir. D. Zarouchas, TU Delft, chair
Associate Prof. dr. ir. S. Giovanni Pereira Castro, TU Delft, examiner
Assistant Prof. dr. ir. N. Eleftheroglou, TU Delft, supervisor
PhD candidate M. Salinas-Camus, TU Delft, supervisor

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

Writing this thesis has been a long journey filled with countless Python errors, LaTeX issues, grammatical mistakes, mathematical slip-ups, and dense literature. Although I wrote this thesis on my own, I couldn't have done it without the support of my wonderful friends and family.

I want to thank the Echo Crew (Evert De Vroey, Bram Amant, Tine Couteur, and Thomas van Himbeek) for brightening up every day at Echo and sharing many well-deserved coffees. I am also deeply grateful to my friends at Moeder Vaarse (Mathieu, Arnaud, Kasper Masure, Aurelie Soors, and Niels Rubbrecht) for their genuine care and for creating so many unforgettable memories together.

Special thanks go to Kato Vanaken, who has always supported and stood by me. I am also thankful to my older brother and sister, who (as they often claim) helped shape me into the person I am today.

Not every thesis student can just come knocking on their supervisors' door whenever they need something. For the fact that I could and their invaluable guidance, I would like to express my gratitude to Nick Eleftheroglou and Mariana Salinas Camus.

Lastly, I would like to thank my parents for giving me the opportunity to study here and for standing by me every step of the way.

*Oscar Carpentier
Delft, October 2024*

Summary

In the aerospace industry, traditional maintenance practices rely on time-scheduled servicing, often leading to unnecessary costs and inefficiencies as maintenance is performed without considering the actual condition of structures. This has driven growing interest in predictive maintenance (PdM), which uses prognostic health management (PHM) to optimise maintenance schedules based on real-time data. While significant progress has been made in prognostics for individual components, there is a gap in applying these techniques to larger, more complex aerospace structures. This thesis addresses that gap by focusing on upscaling prognostic models for aerospace structures. Specifically, it adapts the interoperability input-output model (IIM) to predict the remaining useful life (RUL) of higher-level structures using the training data of the lower-level components of that structure.

Current research on prognostics primarily addresses low-level structures, such as individual coupons, with a limited focus on larger structures. To address this gap, the thesis investigates upscaling methods, specifically system-level prognostics (SLP), which considers a structure as a combination of interconnected components. The IIM is chosen due to its transparency, low complexity, scalability and ability to model interdependencies, making it suitable for aerospace applications.

The methodology adapts the IIM for aerospace structures by incorporating data-driven base predictors, specifically the hidden Markov model (HMM) and support vector regression (SVR). These predictors generate RUL estimates, which are then used to train the IIM through offline and online algorithms. A physical specimen, designed to resemble a higher-level aerospace structure comprising lower-level components, is subjected to fatigue testing to collect real-world data, which is used for SLP.

The case study examines the IIM's predictions using this specimens' data and compares them to results from the base predictors. It evaluates how the model responds to different training data and system behaviours, including sudden failure scenarios. The thesis concludes by validating the modified IIM's effectiveness for system-level prognostics in aerospace structures and offers insights for further refinement and application of this approach in real-world aerospace settings.

Contents

Preface	i
Summary	ii
Nomenclature	v
1 Introduction	1
2 Background Information	3
2.1 Prognostics for Aerospace Structures	3
2.2 System-Level Prognostics	4
2.3 Background for this Research	5
3 Methodology	6
3.1 Base Predictor	6
3.1.1 Hidden Markov Model	6
3.1.2 Support Vector Regression	7
3.2 IIM	8
3.2.1 Modification	9
3.2.2 Gradient Descent Method	9
3.3 Training	10
3.3.1 Offline Training	10
3.3.2 Online Training	10
3.3.3 Determination of Hyperparameters	11
3.4 Verification	12
3.4.1 Verification on Artificial Perfect Data	12
3.4.2 Artificial Data	13
3.4.3 Verification on Variable Data	16
4 Case Study	18
4.1 Manufacturing	18
4.1.1 Specimen Choice	18
4.1.2 Manufacturing Process	20
4.2 Test Set-up	21
4.2.1 Machine Setup	21
4.2.2 Sensor Setup	22
4.3 Data Processing	24
4.3.1 Data Selection	24
4.3.2 Data Pre-Processing	25
4.3.3 Data Clustering	28
4.4 Results of the Base Predictor	30
4.4.1 HMM Model Determination	30
4.4.2 SVR Model Determination	31
4.5 Results and Discussion	33
4.5.1 Results of the IIM Compared to the Base Predictors	34
4.5.2 Comparison of the IIM Results of the Normalised Data VS Averaged Strain Data	36
4.5.3 Comparison of the IIM Results of the Different Base Predictors	37
4.5.4 Error of the IIM with Different Amounts of Training Data	38
4.5.5 Concluding Remarks about the Results of the IIM	39
5 Conclusions and Recommendations	40

References	42
A Raw Strain-data of Specimens	44
B Results of the base predictors for the Coupons	48
C Predictions of the HMM Base Predictor of the Specimen	51
D Predictions of the SVR Base Predictor of the Specimen	53
E Prediction Results	55
F Error of the IIM VS Base Predictors	66
G Error of the IIM using both the SVR and HMM Base Predictor	70
H Error of the IIM Normalised VS Averaged Strain	73
I Error of the IIM with Different Amount of Training Data	75

Nomenclature

Abbreviations

Abbreviation	Definition
AE	Acoustic Emission
BIC	Bayesian Information Criterion
DIC	Digital Image Correlation
GCN	Graph Convolutional Network
GD	Gradient Descent
HI	Health Indicator
HMM	Hidden Markov Model
IIM	Input-Output Model
LSTM	Long Short-Term Memory
MK	Mann-Kendall
MSE	Mean Squared Error
PdM	Predictive Maintenance
PHM	Prognostic Health Management
RUL	Remaining Useful Life
SLP	System-Level Prognostics
SRUL	System Remaining Useful Life
SVR	Support Vector Regression
UQ	Uncertainty Quantification

Introduction

In the aerospace industry, traditional maintenance practices for structures involve servicing at fixed intervals, known as time-scheduled maintenance. While this approach aims to prevent failures, it often leads to unnecessary costs due to premature maintenance without actual damage [1]. Consequently, there has been a significant increase of interest towards predictive maintenance (PdM), which leverages prognostic health management (PHM) to schedule maintenance based on the actual condition of the structures rather than on a predefined timetable.

PHM entails detecting the degradation and diagnosing failures to manage them proactively [2]. By predicting the remaining useful life (RUL) of components, PHM facilitates the introduction of PdM, where maintenance is scheduled based on these predictions. This proactive approach reduces downtime by enabling pre-emptive planning for tools and spare parts, streamlining maintenance and improving cost efficiency.

A crucial aspect of PHM is the prognostic task, which involves predicting the RUL based on sensor data. However, current research has predominantly focused on low-level structures, such as individual components or coupons, limiting the application of these models to larger aerospace structures [3]. This gap highlights the need for further research to scale up these prognostic models to more complex systems.

The building block approach is frequently used in structural design and testing, where models are trained on lower-level elements and subsequently applied to higher-level components. An example is the SHM framework proposed in [3], where the model is initially trained on the coupon and element levels before being extended to the component level. Similarly, system-level prognostics (SLP) considers a system composed of multiple subsystems, aligning it closely with the building block methodology. While various SLP models have been reviewed in [4], none have yet been tested specifically for aerospace structures. However, [5] suggests that the inoperability input-output model (IIM) and the graph convolutional network (GCN) are promising for aerospace applications.

Here, it is chosen to investigate the possibilities of the IIM model further. Contrary to the GCN model, it is not a black-box model and provides a clear connection between the component's degradation. It is also a relatively simple method requiring only a small amount of data to be trained. The online training method of the IIM is also computationally more efficient, as it only requires gradient descent to update the A-matrix. In contrast, the GCN model involves training a graph convolutional encoder with multiple hidden layers during online operations, leading to higher computational demands.

This thesis aims to investigate the applicability and performance of the IIM for aerospace structures. To do this, the IIM will be adapted to aerospace structures, and a representational aerospace structure will be created to test the IIM thoroughly. To be more efficient, the aerospace structural specimen is based on open-hole coupon research conducted simultaneously within the research group. These considerations combined lead to the following research questions that will guide the rest of the thesis.

How can the SRUL, of three open-hole coupons that are connected at the top and bottom, be predicted by the use of SLP techniques, such as the IIM, based on prognostic models that are trained on single open-hole coupons?

- 1. How could the IIM be modified for the given aerospace structure?*
- 2. How does the IIM model respond to different types of input data and base predictors?*
- 3. What is the behaviour of the prediction error based on a varying amount of training data?*

To address this research question, the IIM must first be modified by incorporating inputs from base predictors and adapting its framework for application to aerospace structures. The performance of this modified model will be validated using real-world data to assess its performance. A specimen resembling a higher-level component, analogous to open-hole coupons, will be manufactured and subjected to fatigue testing until failure. Data will be collected through multiple sensors, which will then be used to test and evaluate the performance of the modified IIM model.

This thesis will be subdivided into four major chapters: background information, the methodology, a case study and the conclusions.

First, in the background information chapter, additional explanations will be provided to ensure a thorough understanding of the concepts necessary for the remainder of this research. The discussion will begin by addressing the need for upscaling prognostics, followed by a brief overview of how SLPs relate to this process. Subsequently, more detailed information specific to this research will be presented.

Second, in the methodology chapter, the models will be discussed in detail, beginning with an explanation of the base predictors. This will be followed by a discussion on how the IIM model has been adapted for aerospace structures. The training algorithms used for both offline and online operations will then be elaborated upon, and the chapter will conclude with a verification of the modified IIM model utilising simulated data.

Third, in the case study, an application of the adapted IIM will be provided. This will be done by first providing the specimen, explaining why it was chosen, and how it was manufactured. Then, the entire test set-up will be presented in detail. This is followed by an explanation of how the data has been processed. Next, the determination of all hyperparameters of the base predictor will be provided. Finally, the results will be presented and discussed.

Fourth, a conclusion to the research question and a few recommendations will be provided.

Background Information

This chapter provides the foundational background necessary to contextualize this research. It explores the necessity of advancing prognostic methods to manage large-scale aerospace structures, focusing on system-level prognostics (SLP) as a solution. Different SLP approaches will be reviewed, highlighting two suitable candidates for aerospace applications. Finally, some more practical information will be given to provide a better understanding of the research.

2.1. Prognostics for Aerospace Structures

Various maintenance strategies are available for use. Currently, aerospace systems use the time-scheduled maintenance strategy [1]. This means that maintenance is performed at predetermined times. This is quite costly, especially when the system is never allowed to fail. Corrective maintenance is a strategy where only maintenance is performed after a failure has been observed. In many aerospace structures, this is not allowed for safety reasons. Predictive maintenance (PdM) offers a third strategy. Here, maintenance is planned based on the information acquired in the diagnostic and prognostic process. This information should allow the user to schedule a maintenance date and duration together with the tools and parts necessary for that maintenance. PdM aims to enhance system availability and reliability, which leads to cost savings and guaranteed safety. These three strategies are visualised in Figure 2.1.

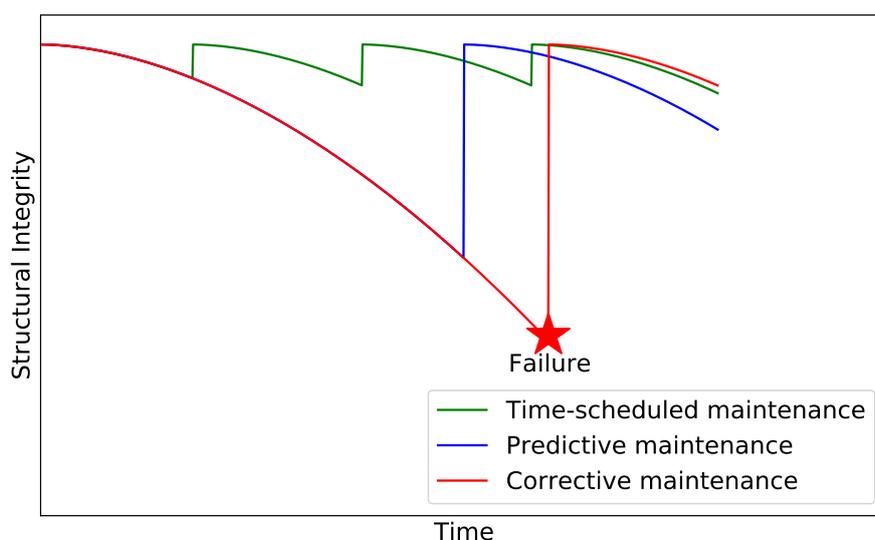


Figure 2.1: Example of maintenance strategies possible with PHM retrieved from [1].

The process of acquiring all the necessary information needed to perform PdM is called prognostic

health management (PHM). Here, PHM is the framework where the degradation is estimated, and the failures in the system are diagnosed to predict its remaining useful life (RUL), such that the systems' failures can be managed proactively [2]. A degradation model and some input data are used to predict the RUL for the prognostic step in PHM. This degradation model can be based on physical models (model-based), data-driven models or a combination of those two (hybrid models) [4]. For aerospace structures, data-driven approaches are the most promising. This can be attributed to the fact that it is hard to upscale model-based prognostics to larger structures, and they have limited capabilities for composites.

In literature, research on prognostics for aerospace structures is mainly limited to the coupon or subsystem level due to the additional prognostic difficulties in large and complex structures. The prognostics must be applied to the aircraft component level for real-life applications in aerospace structures. The building block approach as seen in Figure 2.2 is often used in structural design and testing. This approach consists of performing many tests on the lower-level coupons and decreasing the amount of tests for the higher-level elements. This is necessary as the testing on the lower levels is relatively cheap and can happen faster than on higher levels. This type of upscaling could be achieved by using SLP.

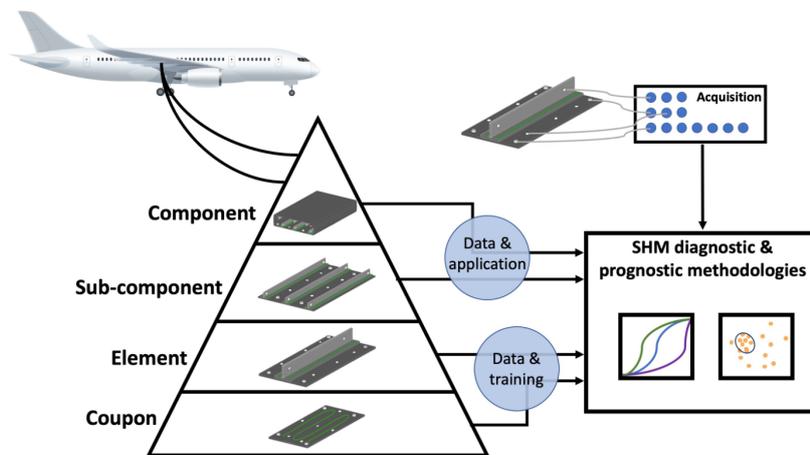


Figure 2.2: Schematic representation of a PHM approach based on the building block approach [3].

2.2. System-Level Prognostics

As discussed in the previous chapter, aerospace structure prognostics are mainly focused on providing models for specific coupons or elements. To achieve some sort of upscaling needed for real-life applications, the higher-level components could be seen as systems comprising different subsystems (e.g., stringers and plates) with certain interactions. This is called SLP, [4] proposes a definition: "System-level prognostics corresponds to the estimation of a system's RUL (SRUL) knowing its current health state and future conditions of use." Here, a system comprises multiple subsystems or components that interact with each other and the environment to fulfil certain functions. In [6] and [7], they show that the system could fail before the individual components would. This indicates that the SRUL is not just equal to the lowest RUL of the subsystems. A recent and extensive literature review on SLP is provided in [4].

The review shows that most existing SLP models are designed for general systems and have not yet been adapted for aerospace applications. For aerospace structures, the SLP must be able to model interdependencies. This need arises because the lower-level structures, which can be seen as the subsystems, are physically connected. This causes significant interactions in the load distribution and, hence, in the degradation of the structures. These interdependencies are generally not known beforehand. Therefore, it is important the SLP can model this based on data-driven methods. Finally, the models must be scalable to apply SLP to large aerospace structures. By looking at these criteria, only two possible candidates remain from the listed possibilities given in [4]; namely, the input-output inoperability method (IIM) and the graph-convolutional network (GCN).

Next to the SLP methods mentioned in [4], a first attempt has been made to upscale prognostics in [8]. The research in [8] is based on the literature review given in [3]. This research is the first paper, to the best of the author's knowledge, that tries to develop an upscaling method for PHM in aerospace structures. This paper achieves results using an ensemble method. Essentially, they try to create the same data for a larger structure as for the smaller one to identify and predict specific failure mechanisms. It is a direct extrapolation of the base predictors and would neglect the fact that there are different interactions due to the upscaling. Only the data of known failure mechanisms should be acquired for this method to work. This neglects the new interactions and will inherently create errors due to this simplification. This leaves the GCN model and the IIM as the most promising SLP-model candidates for aerospace structures.

2.3. Background for this Research

Open-hole coupons are often tested for fatigue. They are easily manufactured and tested, which makes them ideal to serve as subsystems in SLP. Next to this, multiple tests on open-hole aluminium coupons have been recently performed within the Aerospace Structures & Materials department. These tests will serve as the lower-level training data to save time and money, and a higher-level structure will be constructed from these coupons.

Very few large-scale tests are performed on large aerospace structures. Often, only one specimen is fatigue tested until failure due to the large cost and effort. Therefore, the SLP model used should be able to handle this scarcity of data. When comparing the IIM to the GCN model it becomes clear that the IIM is the lowest in complexity [4]. This reduced complexity means it requires less data for training, making it a strong candidate for further investigation, especially given the scarcity of large-scale testing data in aerospace structures.

The IIM proposed by [9] generally works as presented below in Figure 2.3. Here, data is collected from one or more sensors per component. A base predictor then uses this data to form a prediction about the inoperability of the component ($c_i(t)$). This base predictor has been trained on the component it needs to predict for, isolated from the entire system. The IIM then calculates an adapted inoperability ($q_i(t)$). Finally, the maximum or minimum, depending on whether the system has a parallel or serial configuration, is used to calculate the final inoperability. This system-level inoperability is then directly converted to an SRUL.

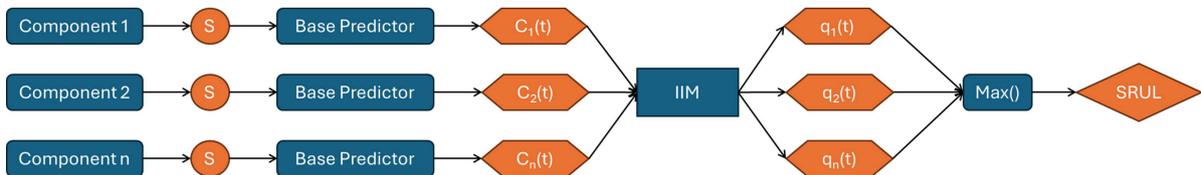


Figure 2.3: General schematic of the original IIM, proposed by [9].

The base predictors used in [9] were model-based. As explained before, this is less desired in aerospace structures. Next, common base predictors in aerospace structures provide an RUL instead of an operability. With a new general framework, visualised in Figure 2.4, some adaptations must be made to the IIM. In the following chapter, different base predictors will be chosen and explained. Then, a modification of the IIM will be proposed and presented.

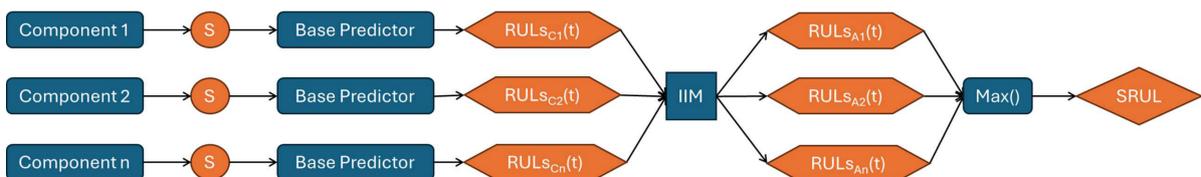


Figure 2.4: General schematic of the proposed adapted IIM.

3

Methodology

As discussed in the previous chapter, several adaptations are required to meet the specific needs of aerospace applications to perform SLP using the IIM for aerospace structures. To tailor the IIM for aerospace structures, the base predictors need to shift from model-based to data-driven approaches. Two distinct data-driven base predictors will be employed to address the second sub-question of the research. A choice is made to use an HMM and SVR, which will be explained in section 3.1. Then, the IIM itself will be adapted to handle RULs as input instead of using the inoperabilities. The modified IIM and its training algorithm will be presented in section 3.2. Finally, the IIM will be verified in section 3.4.

3.1. Base Predictor

The base predictor is the prognostic model that takes the SHM data from the components and converts it to a components' RUL prediction. Three types of prognostic models can be considered: data-driven, model-based and hybrid models [4]. Data-driven models have been chosen as they are more universally applicable to aerospace structures. These models can be readily applied to both composites and metals without significant modifications for larger structural components. Two base predictors will be used to see how the IIM responds to different inputs. Both a support vector regression (SVR) and a hidden Markov model (HMM) are chosen as base predictors.

The HMM is chosen as it is a simple base predictor and a probabilistic model. It takes SHM data and uses unsupervised learning to predict the hidden state corresponding to the health state. An RUL prediction is provided based on the current health state and the trained HMM degradation model. In [10, 11], it has been proven that the family of HMMs can provide accurate results. As a probabilistic model, the HMM is inherently stochastic, allowing it to capture aleatoric uncertainties [12], which are crucial for comprehensive uncertainty quantification (UQ).

The SVR model is also selected as a base predictor. This model is relatively simple, using continuous data and supervised learning to directly predict a component's RUL. Unlike the HMM, the SVR is not a probabilistic model. This means it cannot capture aleatoric uncertainties, resulting in a less comprehensive UQ, making it less ideal for prognostics. However, evaluating the impact of different base predictors is essential for addressing the research question. The distinct characteristics of the SVR and HMM are valuable for understanding the effects of various base predictors. This section will explain the implementation of both the HMM and SVR models and the underlying assumptions.

3.1.1. Hidden Markov Model

An HMM is an extension of the Markov chain where a layer of hidden states is added. These states should correspond to the physical degradation of the system, but they are not directly observable. The observable data that the HMM uses to make predictions correspond to the SHM data, which contains a stochastic relation to the hidden states. This additional layer increases the complexity of the model [13]. HMMs can use both continuous and discrete SHM data. Here, a choice is made to use discrete data as it allows for a higher computational efficiency. An HMM is defined by five elements: the number of

hidden states N , the number of different observations m , the initial state distribution π , the transition probability matrix \mathbf{A} and the emission probability matrix \mathbf{B} [13]. The determination of these elements will be discussed here.

In prognostics, it is often assumed that the system starts in perfect condition when it begins operations. In this research, the same assumption will be made to reduce the number of estimated parameters. This assumption is reflected in the initial state distribution to be defined as follows:

$$\pi = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{N \times 1}$$

The transition and emission matrix are then obtained by unsupervised model training. Specifically, the Baum-Welch algorithm is then used to estimate the parameters of these matrices [14]. For this training, an initial guess of both matrices must be given. Then, the algorithm will update these matrices based on the available SHM data.

The initial guess of the transition and emission matrix is important as it can define some major characteristics. The transition matrix must reflect a unidirectional degradation process when using the HMM for prognostics, particularly in aerospace structures. For example, an aerospace structure can only experience further degradation over time. Once a failure occurs, the structure reaches its final state and cannot revert to a previous state. Furthermore, since degradation typically progresses gradually, the HMM should be configured to transition only between consecutive states. Taking into account these important properties gives an initialisation of the transition matrix of size $N \times N$ -states, as follows:

$$A = \begin{bmatrix} \alpha & 1 - \alpha & 0 & \cdots & 0 \\ 0 & \alpha & 1 - \alpha & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (3.1)$$

The initialisation parameter can be freely chosen between 0-1. It will be initialised with $\alpha = 0.5$ to train this model. The emission matrix, however, does not have these strict requirements. Here, every hidden state can be linked to any discrete observation except for the final one, which represents failure. As a result, the emission matrix will be initialised using a uniform distribution across all entries, excluding the last row and column.

The number of hidden states and discrete observations in an HMM is not predefined. To determine these, the number of discrete observations must first be established by evaluating certain key qualities of the input data. In prognostics, three critical qualities are monotonicity, trendability, and prognosability [15]. However, only monotonicity will be considered for determining the number of clusters, as it is the most important feature for input data in HMMs [10]. The monotonicity of the clustered data can then be assessed and compared using the Mann-Kendall criterion [10].

After determining the number of observations, the number of states can be determined. This is done using the Bayesian information criterion (BIC). This criterion establishes an equilibrium between the best model fit for the given data and the low complexity of the model [16].

Now that the HMM is completely defined, it is possible to identify the health state of a system using SHM data. This does not yet equal the often-used prognostic prediction of RUL. So after the determination of the current health state, the forward algorithm is used to actually determine the remaining useful life [12].

3.1.2. Support Vector Regression

An SVR is chosen as another base predictor as it is a simple and straightforward model to implement. This model uses supervised learning and can use continuous input data. Contrary to the HMM, it is

not a probabilistic model but a deterministic one. The SVR uses an ϵ -insensitivity region and tries to optimise the kernel function to have as many as possible points within that region. This allows the model to consider only a few points while optimising, making it faster. It also helps to reduce noise and allows the model to have good generalization characteristics [17].

A kernel function and some hyperparameters must be chosen to perform this regression. These hyperparameters are the type of scaling of the data γ , the regularization parameter C and the width of the insensitivity region ϵ . The kernel function is the most difficult to determine and greatly impacts the other hyperparameters. Therefore, this is selected first. A few possible kernel functions are chosen by looking at the relation between the truncated ϵ_{yy} and the True RULs. This selection is then evaluated by performing a normal least-squares regression to find the right parameters of those functions and compare the mean squared error (MSE). After the kernel function is determined, a grid search is performed in two stages to determine the other hyperparameters of the SVR. The first stage searches for the best type of γ and determines the best order of magnitude for C and ϵ , which can later be refined. The second stage consists of a grid search where the kernel function and γ are set, and the C and ϵ are optimised. This method identifies an optimal SVR model while keeping the computational time within reasonable bounds.

3.2. IIM

The IIM originated from economics and was later adapted for the SLP of general systems in [18], where the prognostics of individual components are carried out using model-based methods. For this research, the base predictors had to be changed. This caused some changes to the IIM as well. However, the main elements are similar. From this point forward, all mentions of the IIM refer to the adapted version. The IIM consists of two main phases: an offline and an online part. This allows the model to adapt to the specific characteristics of the monitored system. A flowchart of the model's workflow is presented below in Figure 3.1. Initially, the offline phase is executed. In this phase, base predictors generate initial RUL estimates per component, denoted as $RUL_{c(t)}$. These estimates are combined with an initial guess of the A -matrix to create the IIM, defined by an offline A -matrix, A_{offline} .

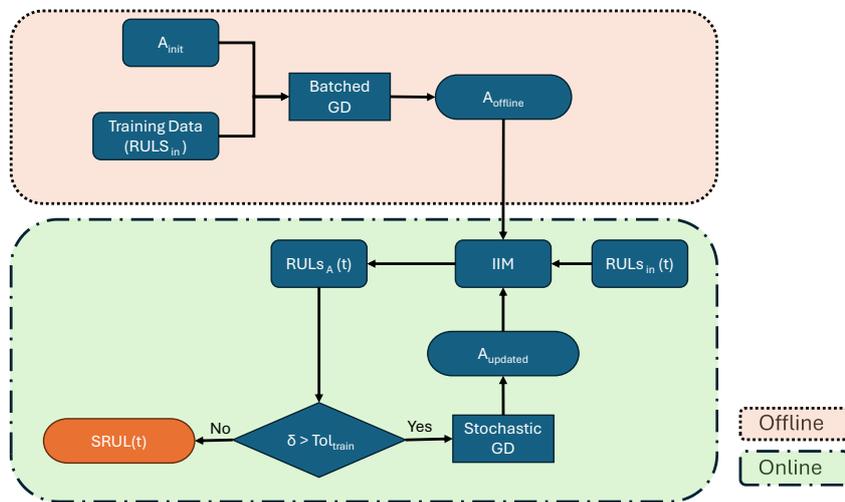


Figure 3.1: Flowchart of the adapted IIM's online and offline workings.

During operations, the model enters the online phase. The base predictors continuously generate updated $RUL_{c(t)}$, which are fed into the IIM. Using A_{offline} , the model produces new $RUL_A(t)$ predictions. If the difference between the new $RUL_A(t)$ and initial $RUL_A(0)$ predictions, denoted as $\delta = |RUL_A(t) - RUL_A(0)|$, exceeds a predefined threshold Tol_{train} , the A -matrix is updated through online training.

The updated $RUL_A(t)$ predictions are then used to determine the SRUL by taking the maximum value of the adjusted RUL estimates.

In this section, the adaptation of the IIM model itself will be discussed first in subsection 3.2.1. Then, the gradient descent (GD) method used to train and update the A -matrix will be discussed in subsection 3.2.2. An explanation of the specific application of GD will then be provided, including details on the training process during both offline and online operations in section 3.3.

3.2.1. Modification

The IIM model proposed by [18] can be described by the following equation:

$$q(t) = \kappa(t) \cdot [A \cdot q(t-1) + c(t)] \quad (3.2)$$

Here, $q(t)$ represents a vector of the overall inoperabilities at time t , $\kappa(t)$ a matrix of the environmental contributions at time t , $c(t)$ the components' internal inoperabilities at time t and A is the matrix representing the interdependencies of the components. This equation was designed with model-based base predictors to give input operabilities. This equation will be modified to match the input of data-driven models and the needs of aerospace structures in general.

First, as aircraft fly in different operational conditions that are constantly changing, the influences of the environment are difficult to determine. Consequently, the environmental factor, denoted as $\kappa(t)$, will be excluded from the analysis. Additionally, incorporating this factor would necessitate data from a range of environmental conditions, which falls outside the scope of this study. In the previous IIM model, the inoperabilities are given by a number between 0 and 1. These represent the degradation of the system and are later converted to the SRUL. The case study presented in [18] uses components with a relatively straightforward connection between their inoperability and RUL. For aerospace structures, this is not always as easily converted. Therefore, an adaptation is proposed where the components' RULs are used as inputs. The proposed adaptation is described in Equation 3.3:

$$RUL_{A(t)} = RUL_{c(t)} \circ (A \cdot z(t)) + RUL_{c(t)} \quad (3.3)$$

In this equation, $RUL_{A(t)}$ is a vector representing the adapted RULs of the components at time t . $RUL_{c(t)}$ is a vector representing the RULs predicted by the base predictor at time t of the components. A is a matrix representing the interactions between the components. The symbol \circ represents the Hadamard product or pointwise multiplication. Then $z(t)$ is the relative degradation of the components at time t and is calculated using the following equation:

$$z(t) = \frac{RUL_{c(t)}}{\text{mean}(RUL_{c(t)})} \quad (3.4)$$

This relative degradation can be seen as similar to the inoperabilities at time t . The elements within the vector $z(t)$ are typically close to 1. This vector indicates whether a component's degradation is above or below the average level, reflecting the relative performance of a given subsystem or part of a structure. To then determine the SRUL, the maximum of the vector $RUL_{A(t)}$ is used. This comes from the reasoning that $RUL_{A(t)}$ predicts the true RULs of the components given the interdependencies. This method for determining the SRUL assumes that the entire structure fails only when all its components have failed. Since this is the first study of SLP in the context of aerospace structures, this assumption should be considered speculative rather than definitive.

3.2.2. Gradient Descent Method

The GD to update the A -matrix can be performed in a few different ways: a batch GD, a stochastic GD or a mini-batch GD [19]. First, a batched gradient descent can be computationally expensive and provide slower convergence, but it is generally more stable. A stochastic GD provides faster convergence and is less computationally expensive. Next to this, it also can adapt itself to changing patterns. The mini-batch GD combines both and can be the compromise between both types. For offline training,

the stability of the batch GD is important, and the computation time is less important. Therefore, the batch GD is used for this. For online training, however, computation time and the ability to change to different patterns are important. Therefore, the stochastic GD is used for online training.

To calculate the gradient, a loss function must first be chosen. The standard mean squared error (MSE) will be used, as shown below.

$$\mathcal{L}(\text{RUL}_{A(t)}, \text{True RUL}) = \frac{1}{N} \cdot (\text{RUL}_{A(t)} - \text{True RUL})^2 \quad (3.5)$$

The simple gradient for each entry a_{ij} of the A -matrix must then be calculated to extract the gradient for the A -matrix. By solving this, the following solution can be obtained:

$$\frac{\partial \mathcal{L}}{\partial a_{ij}} = \frac{2}{N} \cdot \text{RUL}_c(i) \cdot (\text{RUL}_A(i) - \text{True RUL}_{(i)}) \cdot z_{(j)} \quad (3.6)$$

Here, the subscript i stands for the RUL- values of the i^{th} component, and N is the number of components. The A -matrix is then updated based on this gradient as presented in Equation 3.7. Here, the learning rate λ , which must be determined beforehand.

$$a_{ij\text{new}} = a_{ij\text{prev}} - \lambda \cdot \frac{\partial \mathcal{L}}{\partial a_{ij}} \quad (3.7)$$

This GD process is done iteratively until a stopping criterion is met. This is mainly done by defining the number of iterations beforehand. Other stopping criteria are available, such as assessing the absolute value of the gradient or checking whether the error of the predictions converges. A choice of the stopping criterion will be given in section 3.3

3.3. Training

For the IIM to make a prediction, the A -matrix should be estimated. This is the only parameter that must be updated in the training process. This section will explain how the training algorithm will be executed and what will be used as a label during offline and online operations.

3.3.1. Offline Training

The true RULs are known in offline training, enabling a straightforward labelling process. The A -matrix is updated by comparing the predicted $\text{RUL}_{A(t)}$ at each time point to the True RULs. This A -matrix is then updated using batched GD as discussed in subsection 3.2.2. The stopping criterion should also be determined for the offline training. Due to the batched GD, it is difficult to observe whether the gradient is converged. As the gradient is an average of a certain number of points with some noise, the gradient will probably never go to zero completely. The same problem could occur by checking the convergence of the prediction error. As a batched GD is used, the predictions might fluctuate at convergence, causing the error of predictions to stay higher than a given convergence criterion. Therefore, a fixed number of iterations is opted for. This amount of iterations can be determined using trial and error on the available training datasets. The offline training process is outlined in the flow chart in Figure 3.2.

3.3.2. Online Training

True RULs are not available during online training, so an alternative method for labelling is required. This approach compares previous time-step predictions to the current $\text{RUL}_{c(t)}$. A timestep difference k is introduced to enable the model to predict changes over larger intervals, allowing it to be updated according to its specific characteristics while preserving its long-term predictive capabilities. The value of this parameter will be determined through trial and error on the training datasets, which will be elaborated on in the following subsection. Since it is generally preferable to keep computational efforts low during operations, the system should avoid constant retraining. Continuous training is not ideal; therefore, training is initiated only when the prediction error observed at the last update exceeds a defined threshold, $\text{Tot}_{\text{train}}$.

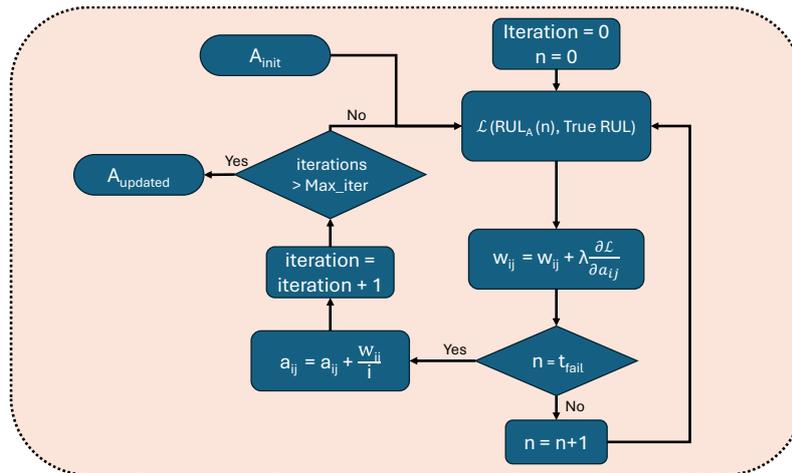


Figure 3.2: Flowchart of the offline training scheme for the IIM.

When the training is initialised, the GD starts updating the A -matrix. As discussed previously, a stopping criterion is necessary. Stopping criteria were initially set to terminate the GD optimisation when the gradient's magnitude fell below a certain threshold. However, this approach frequently led to divergence and gradient explosion, which persisted despite adjustments to the threshold value. Setting it too high eventually reduced the impact of updates to the IIM, diminishing the model's adaptability. An alternative stopping criterion was considered based on the convergence of the prediction error. This was not deemed suitable due to the lack of a true RUL for validation. Consequently, a fixed number of iterations was chosen as the stopping criterion for GD. The rationale for determining this fixed number of iterations is elaborated in the following subsection. A summarising diagram of the online training scheme is presented in Figure 3.3.

3.3.3. Determination of Hyperparameters

Usually, data is divided into three groups for general ML tasks. Namely, the training, validating and testing data. This allows the user to identify hyperparameters to optimise the training without overfitting the model. As for large aerospace structures, there is not enough data available, so a different approach is used to determine the hyperparameters using only one training specimen. These hyperparameters are the learning rate, the number of iterations during offline training, the online training tolerance Tol_{train} , the label time difference k and the number of iterations during online training.

The learning rate must be small enough for the A -matrix to converge and to prevent numerical instability or divergence during training. Yet, choosing the learning rate too low increases the number of iterations needed and has the possibility of converging to local minima of the loss function. As computational time is important during online operations, the learning rate will be determined by finding the largest value that does not cause divergence and dividing it by 5 to include a safety factor. This safety factor was chosen arbitrarily.

A simple trial and error can be used to choose the number of iterations during offline training. By looking at the convergence for the error and limiting the training time, a number can be chosen. This choice is relatively minor because each aerospace structure has unique characteristics. Therefore, the A -matrix will be updated online

Similarly, selecting the value for Tol_{train} requires a trial-and-error process using the training specimens. The goal is to enable the model to update quickly before the predictions deviate significantly. However, the model should not initiate training for every minor deviation in the predictions.

The training sample should be utilised to establish k and the number of iterations required for online

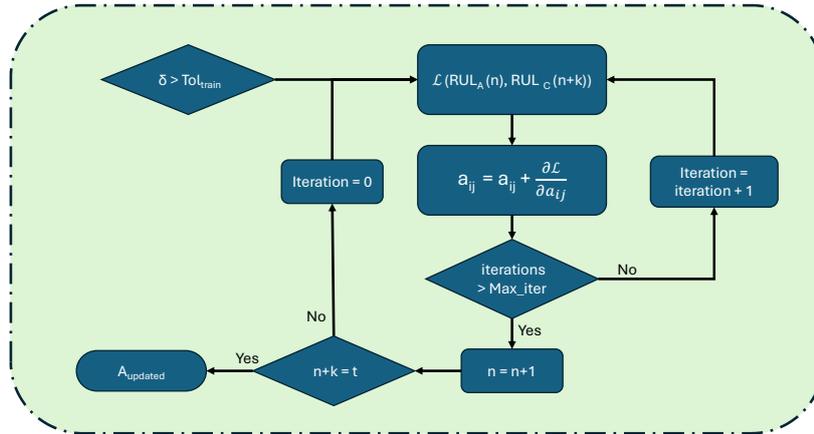


Figure 3.3: Schematic representation of the online training approach.

training. It would be unrepresentative to train the model offline with one specimen and subsequently use the same data to determine the number of iterations for online training. Therefore, the online training process begins with an arbitrarily A -matrix initialisation. The number of iterations and k are then adjusted, considering two key factors: the long-term predictive capability and the ability to adapt to sudden changes in the data's behaviour quickly. The number of iterations is chosen to allow the IIM to respond to significant trends in the data while avoiding overreacting to minor perturbations. While no specific metric is applied, these characteristics are evaluated based on the plots of the SRUL predictions.

3.4. Verification

This section presents a verification of the IIM, which requires using artificial data. An idealised dataset is generated to validate the fundamental aspects of the model. Following this, three distinct artificial datasets were created: the first two demonstrate different degradation patterns to assess the model's response to varying system characteristics, while the third simulates a sudden failure to test the IIM's ability to manage abrupt breakdowns. Initially, the model will be verified using idealised artificial data, as outlined in subsection 3.4.1, where both the online and offline phases will be tested to ensure the model's fundamental functionality. Following this, subsection 3.4.2 will explain the artificial data generation process used for the verification. Subsequently, the model will be tested on variable data in subsection 3.4.3, assessing its ability to adapt while maintaining correct operation.

3.4.1. Verification on Artificial Perfect Data

In an ideal scenario, the prognostic model would precisely predict the true RUL, resulting in a straight line of accurate predictions. For the input data, this implies that each base predictor accurately forecasts the true RUL of its respective component, with variations across components. As previously discussed, the system is assumed to fail when the last component fails, meaning the true SRUL corresponds to the maximum True RUL among the components. For the verification process in this chapter, a random initial true RUL value is selected within the arbitrary range of 180 to 400 cycles, from which the remaining data is generated. An example of the data used as input is provided below in Figure 3.4.

So first, to see if the online training works properly, the IIM is trained and tested on one single degradation history for different amounts of iterations. Here, the error of the predicted RUL should converge to zero for the offline model to work properly. As shown below in Figure 3.5 the error converges to zero. This shows that the offline training of the model works as expected. For this verification test, the learning rate was set to $1.4e - 6$. This is determined based upon the method discussed in subsection 3.3.3.

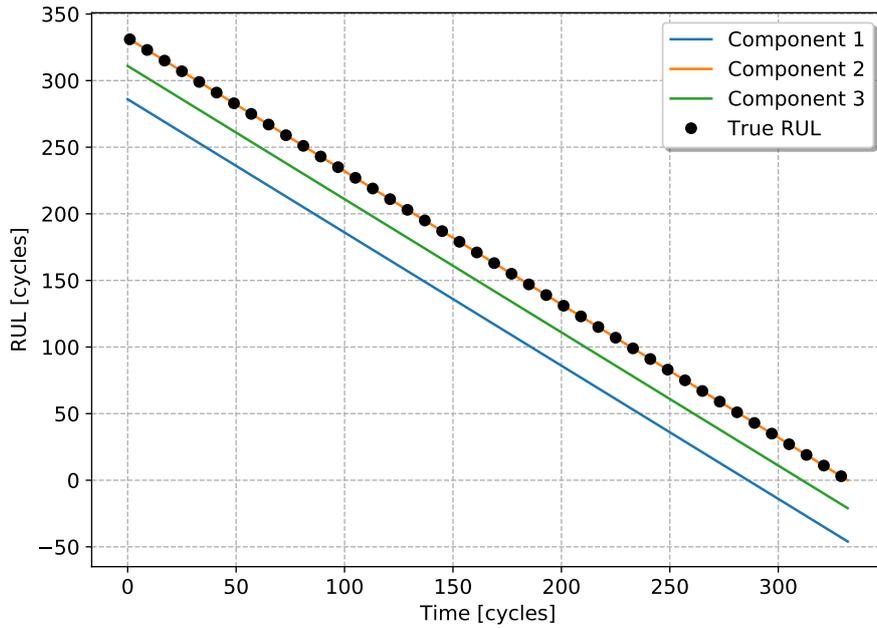


Figure 3.4: Example of the perfect data used for verification.

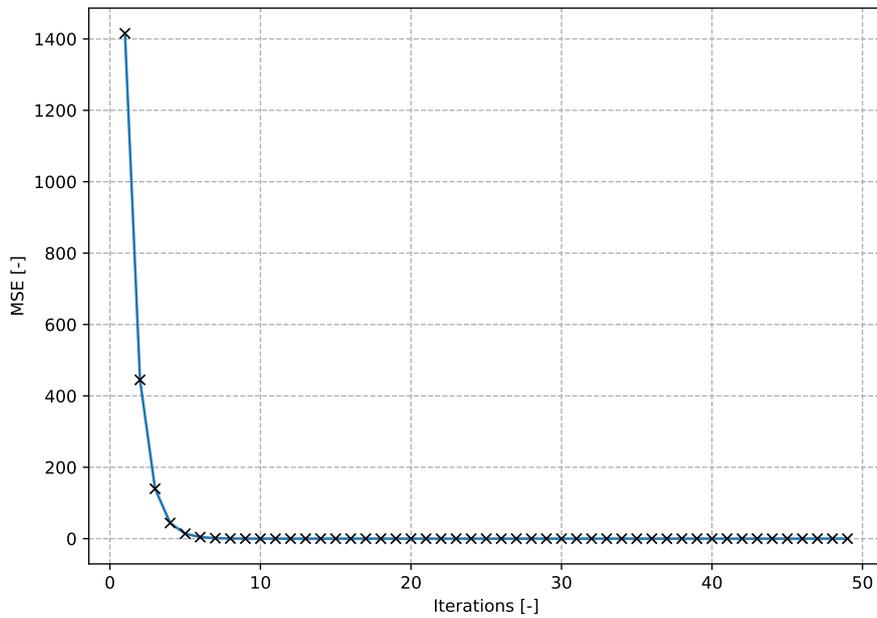


Figure 3.5: MSE of the offline predictions of the IIM in function of different numbers of iterations used for GD.

The IIM was first trained offline to verify the model’s online training using randomly generated perfect data. The hyperparameters were then chosen as outlined in subsection 3.3.3, with the learning rate set to 1.2×10^{-6} , Tol_{train} to 0.1, k to 10, and the number of online iterations set to 100. The model was applied online using a newly generated perfect dataset. The model’s error over time is shown in Figure 3.6, where it is evident that the error reduces to zero once the online training begins.

3.4.2. Artificial Data

Artificial RUL data was made to further verify and investigate the IIM and see the effects of different types of interdependencies and how the model interacts with them. As the interactions are unknown at this stage, three different types of artificial datasets were made. One where the individual RULs are

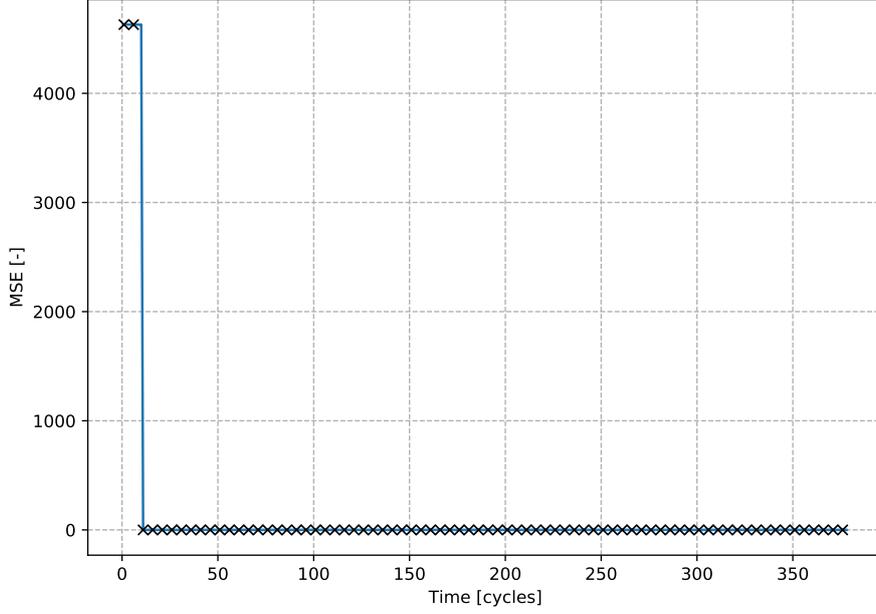


Figure 3.6: MSE during online operations using unseen perfect data and applying online training scheme.

independent of each other until one component's RUL goes to zero (fails). Only then will an increase in degradation be added. The second artificial dataset contains dependencies. If one component has a higher degradation than the others, it will start degrading slower and the other components will start degrading faster. The third artificial dataset represents a sudden failure in one of the subsystems, causing its RUL to drop to zero abruptly.

For both datasets, an initial RUL (RUL_0), the average degradation per timestep μ_0 and the standard deviation for the degradation (σ_0) are randomly chosen from a uniform distribution, as presented below. These are the base parameters that will form the artificial datasets.

$$\begin{aligned} RUL_0^n &= \text{Uniform}(18000; 25000) \\ \sigma_0^n &= \text{Uniform}(0.14; 0.16) \\ \mu_0^n &= \text{Uniform}(0.8; 1; 1.12) \end{aligned} \quad (3.8)$$

Then, starting from the RUL_0 , a random number is subtracted from the RUL from the previous timestep. The determination of this random number differs per dataset. For the independent dataset, a random number generator provides the number given μ_0^n and σ_0^n as given in Equation 3.9. After the failure of a component k , the μ_0^k is divided by the number of remaining components and added to the μ_0^n of the remaining components as presented in Equation 3.10.

$$RUL_i^n = RUL_{i-1}^n - \text{Gauss}(\mu_0^n; \sigma_0^n) \quad (3.9)$$

$$\mu_0^n = \mu_0^n + \frac{\mu_0^k}{\# \text{ working components}} \quad (\forall n \in 1, 2, \dots, N) \quad (3.10)$$

For the dependent dataset, the same random number generator is used. The μ_i^n , which defines the random number that should be subtracted every timestep, does alter. This is done to reflect a dependency between components. The exact calculation can be found in Equation 3.12 and is performed at every timestep. The RUL is then calculated similarly to the independent dataset, as seen below in Equation 3.11.

$$RUL_i^n = RUL_{i-1}^n - \text{Gauss}(\mu_i^n; \sigma^n) \quad (3.11)$$

$$\mu_i^n = \mu_0^n - \frac{(RUL_{i-1}^n - \min(RUL_{i-1}^n \forall n \in 1, 2, \dots, N))}{\text{Mean}(RUL_0^n \forall n \in 1, 2, \dots, N)} \quad (3.12)$$

An example of the artificial data can be seen in Figure 3.7. The same initialisations were used for both the dependent and the independent datasets. Even with equal input, it is clear that the behaviour of the two RUL predictions is different. This will be useful for verifying the IIM in the next section.

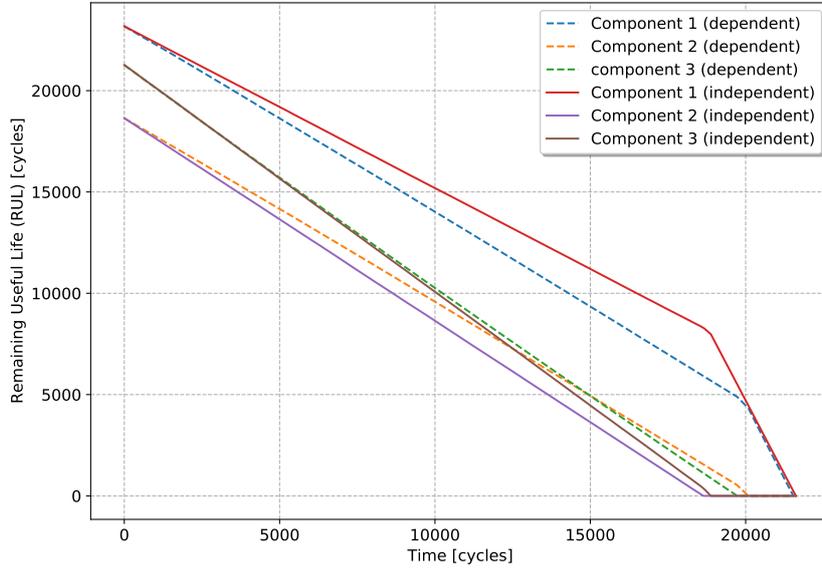


Figure 3.7: Example of the artificial data used for verifying and investigating the IIM.

Lastly, the third dataset, representing sudden failure, is generated using the same method as the independent dataset, except that one component's RUL is abruptly set to zero. An example of these input RULs is shown in Figure 3.8.

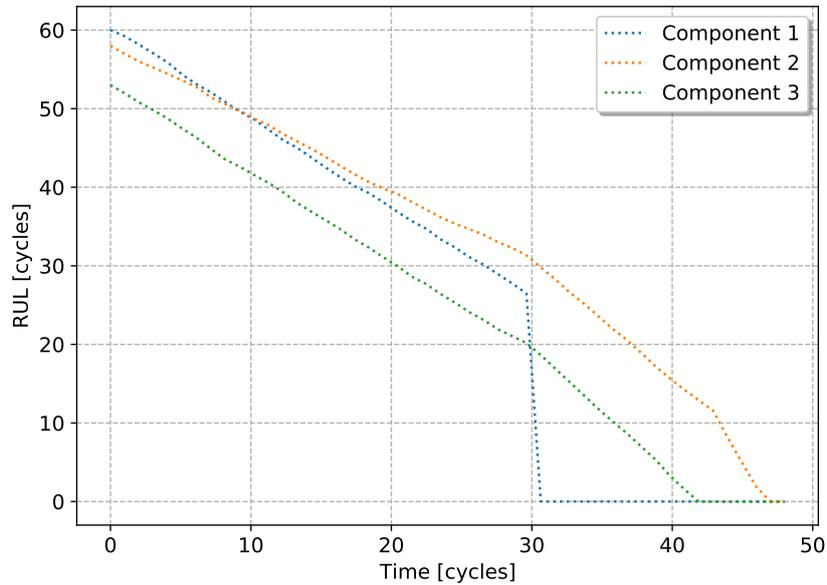


Figure 3.8: Example of the sudden failure artificial data used for verifying and investigating the IIM.

3.4.3. Verification on Variable Data

The model should be able to adapt to new systems, as each has its own characteristics due to certain manufacturing and operational variations. In reality, it is also highly likely that the model will have to be trained on one system and that it has to perform on others that may have some different characteristics. The three questions stated below must be examined to investigate and verify this model on variable data. This will be done by the artificial data created as presented in previous subsection 3.4.2.

1. Does the model increase its accuracy when it increases training data?
2. Does the model adapt itself to data with different characteristics?
3. Can the model adapt to sudden changes in the input data and system's behaviour, such as the sudden failure of a component?

First, some hyperparameters must be determined. This is done using the method provided in subsection 3.3.3. This led to the parameters provided in Table 3.1. These hyperparameters will be used for the rest of this subsection. The model is trained with additional degradation histories in each iteration to address the first question. For every iteration, the model's error is calculated across 10 different degradation histories to assess the change in error. This is done for both the dependent and the independent dataset. The MSE of the IIM as a function of the increasing amount of training data is presented in figures 3.9 and 3.10 for respectively the dependent and independent data. As shown, the error decreases as expected with the growing volume of training data, eventually converging after approximately 150 degradation histories for both types. It is important to note that the error does not converge to zero. This is due to imperfections and randomness in the data, which leads to a residual error in the model's performance. For the further verification of the model, 150 degradation histories will be used for offline training.

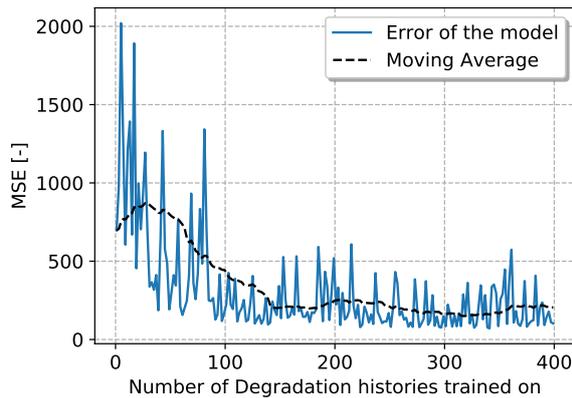


Figure 3.9: MSE of the IIM trained offline on increasing amounts of degradation histories using the dependent artificial data.

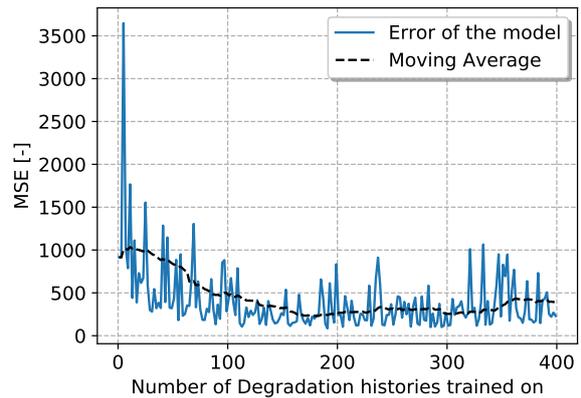


Figure 3.10: MSE of the IIM trained offline on increasing amounts of degradation histories using the independent artificial data.

Table 3.1: Hyperparameters used for the verification on artificial data.

-	Offline			Online				
Parameter	Learning rate	Iterations	Batch	Learning rate	k	Iterations	Tol _{train}	Batch
Value	6e-5	40	True	6e-5	20	100	0.1	False

Then, the second question will be investigated by assessing the model's ability to adapt to different system characteristics. The error in predictions for dependent and independent datasets will be compared. Two models will be trained sequentially, with the first being trained on 150 dependent degradation histories, followed by the second model trained on 150 independent degradation histories. The models will then each be applied online for both data types to observe the impact on the error, enabling a direct comparison. Initially, the MSE of the IIM might be relatively high for the dataset it was not originally trained on. However, after online training, the error should be reduced to a

similar order of magnitude. The MSE of the IIM trained offline on the dependent dataset is shown in Figure 3.11, while Figure 3.12 illustrates the MSE for the model trained on the independent dataset. In both cases, the error for the untrained dataset type is initially higher but converges, demonstrating the effectiveness of the IIM's online training.

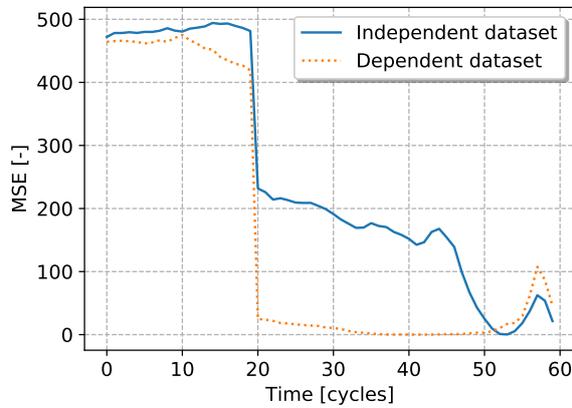


Figure 3.11: Comparison of the MSE of predictions generated by a model trained on the dependent dataset, applied online for the independent and dependent degradation histories.

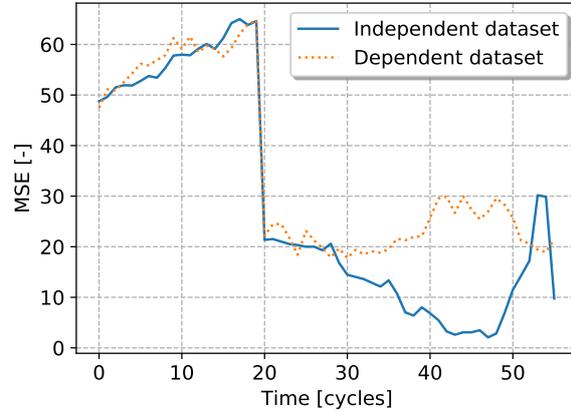


Figure 3.12: Comparison of the MSE of predictions generated by a model trained on the independent dataset, applied online for the independent and dependent degradation histories.

Finally, the third question could be investigated using the dataset where the input RUL suddenly jumps to zero at a given timestep. For this example, a sudden failure at $t = 30$ will be used. The results are shown below in Figure 3.13, where it is evident that the IIM's error initially spikes during online operations but quickly decreases after a few timesteps, falling below the error of the offline model. As anticipated, the online training enhances prediction accuracy, even during a sudden failure.

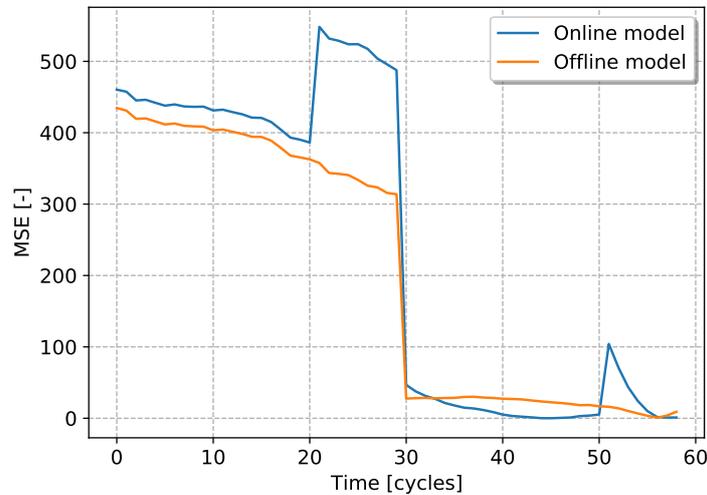


Figure 3.13: Comparison of the MSE of an off- and online IIM, using input data representing sudden failure.

From this section, it became clear that the model behaves as expected. For perfect data, the error of the model goes to zero using online training. For variable data, the model adapts itself to perform better using online learning than it did before. This shows that the model is built correctly. Now, a case study must be performed to see whether the model correctly handles actual RUL data from aerospace structures so that it can be validated. This will be done in the next chapter.

4

Case Study

To address the research questions, the adapted IIM must be tested on a representative aerospace component. Given the availability of fatigue data from aluminium open-hole coupons, this case study leverages that data to minimise manufacturing and testing time. The study builds upon these coupons using the existing data as training input. Subsequently, a specimen of three connected open-hole coupons was manufactured and tested. This allows for evaluating the IIM's performance in predicting the remaining useful life of a higher-level structure comprising lower-level components.

The manufacturing process of the specimens was conducted in two batches of four specimens, though only six of the eight specimens provided usable data due to issues encountered during testing, specifically specimens three to eight. These specimens are then subjected to fatigue tests that are similar to the single open-hole coupons and the same types of data are collected. Data will then be processed in two different ways to evaluate the IIM. One method will use the raw sensor data, while the other will apply a normalisation process that resembles a good health index (HI). This approach allows for evaluating how different input data types affect the IIM's performance.

The case study will first discuss the choice of the specimen and how they are manufactured exactly in section 4.1. Then, it will be explained how the fatigue testing machine and the sensors are set up in detail in section 4.2. Following, the two different approaches to data processing will be mentioned in section 4.3. Next, the results of the base predictors, trained on the single open-hole coupons, are mentioned in section 4.4. Finally, the results of the IIM, together with a discussion, will be provided in section 4.5.

4.1. Manufacturing

Tests must be performed to validate the model and answer the research questions. For this, a specimen must be made to test the proposed model. A specimen as shown in Figure 4.1 is used. Before manufacturing these specimens, it was determined that they should be made in two batches, each consisting of 4 specimens. This enabled a rectification of any errors in the manufacturing process that may have arisen during the testing. Following the initial testing phase, it was evident that the notches were frequently oversized, leading to a failure time that deviated substantially from the coupons used during training. Therefore, a different approach was used in the second batch to make the notches. The choice of the specimen will be explained in subsection 4.1.1. Then, the manufacturing process will be explained in subsection 4.1.2, including the difference in the notches of the two batches.

4.1.1. Specimen Choice

This thesis investigates whether the IIM model can be used for SLP for aerospace structures. To validate the model, fatigue testing is required on a specimen. Given the availability of data on open-hole aluminium coupons, the specimen will be constructed from the same aluminium alloy. The specimen's shape will be designed to closely match the individual coupons used for training. Therefore, a configuration consisting of three open-hole coupons connected at the top and bottom, as shown in

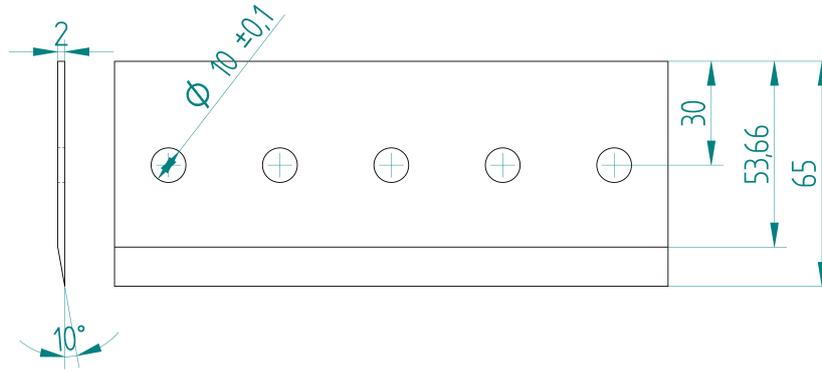


Figure 4.2: Tab's dimensions provided in mm.

4.1.2. Manufacturing Process

The specimen itself and the tabs are manufactured using a CNC machine. This is done based on three-dimensional digital drawings. The specimen was cut from a 2mm thick sheet of aluminium 7075-T6. The holes and slots were milled using a CNC machine. The tabs were cut from the same aluminium using the same methods. The process of glueing the tabs and making the notches will be explained in more detail in the following subsections.

Glueing the tabs

The tabs were bonded to the specimen using Scotch-Weld EC-9323 B/A adhesive. This two-component adhesive was selected due to its high strength and common use in aerospace structures. To ensure a strong bond, the following steps were implemented:

- The surfaces are roughened by sandblasting them.
- The surfaces are cleaned with acetone.
- The adhesive is mixed in a 27:100 ratio.
- A vacuum bag is prepared for the curing process.
- The tabs are placed on the specimen and fixated using tape along the tapered edge.
- The adhesive is applied to the surfaces of both the tabs as the specimen.
- Glass beads of $150\mu\text{m}$ are sprinkled on top of the glue to ensure a minimum thickness as specified by the manufacturer. At this point in manufacturing, the specimen looks as in Figure 4.3.
- The tabs are folded on top of the specimen.
- The specimen is placed in the vacuum bag and the bag is sealed.
- The vacuum pump is turned on (-250 hPa), and the adhesive is cured for at least 4 hours.
- To ensure the specimen can be bolted onto the machine, the holes are filed manually, as the holes of the tabs and specimen are often not perfectly aligned.

Making Notches

A notch was introduced to the specimen to ensure crack initiation at a specific location, allowing the cameras and sensors to be focused accordingly and reducing the number of cycles until failure. This approach primarily aimed to minimise the required testing time for the experiments. The notches were created using an electrical fretsaw; however, a notable issue arose. The intended notch size of 0.5 [mm] was not consistently achieved. As outlined earlier in this section, two batches were produced, each employing a different manufacturing method.

For the first batch, the notches were made on eyesight. As 0.5 [mm] is too little to draw on the specimens properly, they were sawn at sight. This caused quite a big variation, as seen in Figure 4.4 and in 4.5, where the notches are notably different. This resulted in low lifetimes significantly differing from the coupons' testing data. To resolve this problem in the second batch, a template made of wood was made.

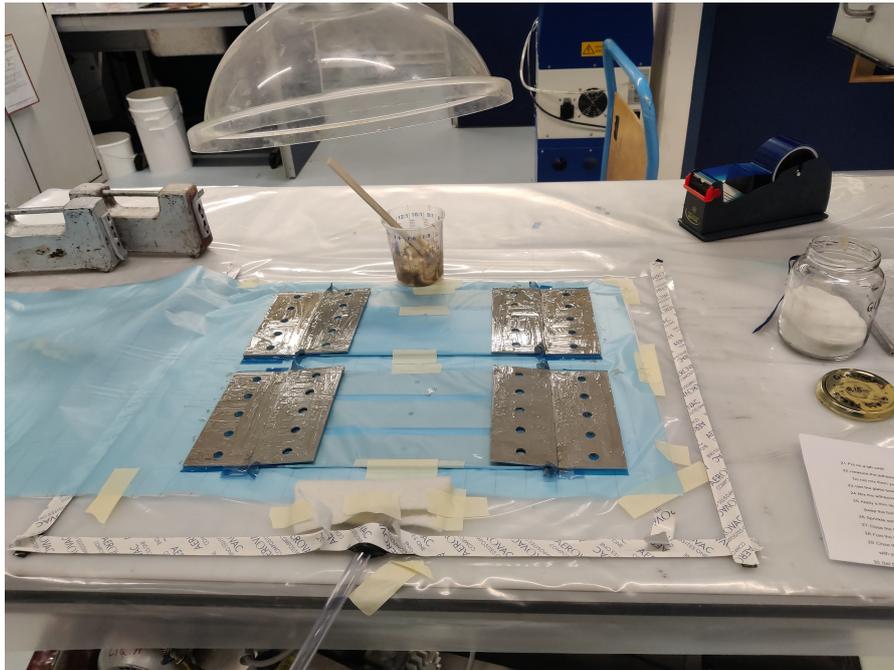


Figure 4.3: Figure of the glueing process after the glass beads are applied before the tabs are closed.

This template was made using a laser cutter, increasing the accuracy. This template was then attached to the three components of the specimen, and the notches were made using the electrical fretsaw again.

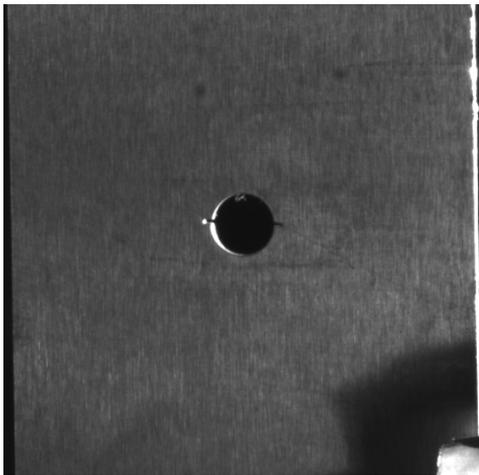


Figure 4.4: Close-up of two notches belonging to specimen 1.

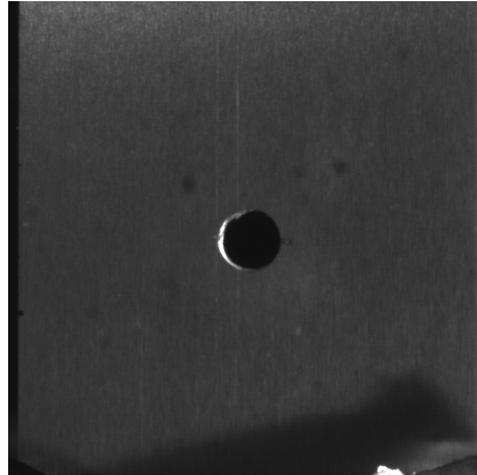


Figure 4.5: Close-up of two notches belonging to specimen 3.

4.2. Test Set-up

In this section, the test setup will be discussed. First, the specimen and machine setup will be provided in subsection 4.2.1. Then, the sensors used and how they are installed will be discussed in subsection 4.2.2.

4.2.1. Machine Setup

The machine used is an MTS fatigue-loading machine with a maximum loading of $60[kN]$. To connect any specimen, the fatigue loading machine has two clamps at the top and bottom, with a width of $45[mm]$. Connector pieces were used because these are too small to fit in the chosen specimen. The set-up of the specimen and the machine then looks as follows:



Figure 4.6: Machine set-up

The coupons used for training data were tested with a maximum load of $9[kN]$. As the specimen consists of three coupons, the maximum load is multiplied by three and set to $27[kN]$. The machine is then set to pull from a load ranging from $2[kN]$ until $27[kN]$ at a frequency of $5[Hz]$. This is done using load control as the amount of displacement for this given force changes throughout the testing. The loading program is configured to run 250 cycles at 5 Hz. After completing the cycles, it gradually increases to the maximum load, holding for 5 seconds to capture images of the specimen. The load is then gradually reduced, and the cycle continues. This load sequence is plotted below in Figure 4.7. Whenever the machine detects excessive displacement ($13[mm]$), it stops the testing cycle, as the specimen most likely failed.

4.2.2. Sensor Setup

The testing of the coupons was performed using three different types of sensors. Namely, digital image correlation (DIC), acoustic emission (AE) and pictures of the crack region were used. The same type of sensors will be used as the coupon training data must match the specimen's. How exactly they are applied and configured will be explained here.

First, the AE sensors must be applied to the specimen. The coupons' AE sensors were fixated using clamps. For the specimen, this was not possible due to the larger shape. A 3D-printed holder, made from PLA, was glued to the aluminium specimen to hold them in place. This was done $7.5[cm]$ above and below the middle hole, similar to the coupons. The AE sensors were placed inside these holders

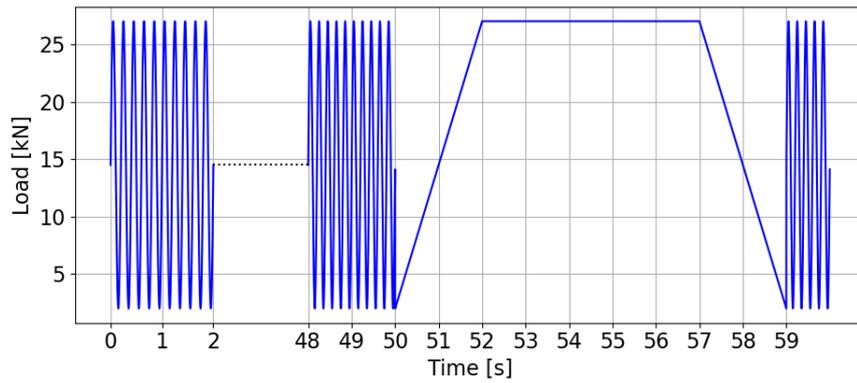


Figure 4.7: Loading diagram of the programmed cycles of the fatigue machine.

with a water-based coupling agent between the sensor and the specimen, as seen in Figure 4.8. The data acquisition system of the AE sensors was then connected, and a threshold of $50[dB]$ was set to exclude noise from the measurements. The pencil-lead break test was executed to verify the sensors' connection. Here, the sensors were correctly calibrated when no signals were received from the AE sensors when nothing touched the specimen and a sharp signal of at least $95[dB]$ was recorded when the pencil lead was broken near the sensor. This ensures that no noise is recorded and the sensors are correctly connected.



Figure 4.8: Example of the fixation of the AE sensors to one of the specimens tested.

Second, the specimen must be painted to use DIC. A matte white base coat was applied, followed by black speckles using spray paint. Since the speckle pattern was created manually, the speckle size was not uniform; however, it was visually assessed and deemed sufficient for the application.

Since only in-plane deformation is expected, a 2D DIC configuration was chosen, as 3D DIC would require extensive calibration and significantly increase time and effort. As previously mentioned, the manufacturing and testing were conducted in two batches. A single 5 MP camera captured the entire specimen for the first batch. In the second batch, two 5 MP cameras were employed: one to capture the entire specimen, as in the first batch, and the other to focus on the middle holes, providing higher resolution in the area where the crack developed. Additionally, extra lighting was used to minimise the impact of daylight on the DIC data.

Third, images of the crack were captured using a standard 4 MP greyscale camera. These images, taken from the unpainted side of the specimen, focused solely on the middle holes to document the crack and its progression throughout the testing sequence. As with the DIC setup, additional lighting was employed to enhance image clarity and minimise the effects of ambient light.

While these images and DIC measurements were captured at specific intervals, AE monitoring provided continuous data throughout the testing. Unlike DIC and crack length measurements, which are active techniques triggered only during the holding phase of the testing sequence, AE is a passive sensor. It records data whenever an event occurs within the material, offering a much higher sampling rate. Whereas images are taken every 251 cycles, AE data can be captured multiple times per cycle, depending on the activity in the specimen. This contrast highlights the complementary nature of the two methods, combining high-resolution temporal data from AE with detailed visual and spatial measurements from the DIC and crack imaging.

4.3. Data Processing

This section outlines the steps taken to process the data collected from the testing of the coupons and specimens 3-8, as only these produced usable results. The data processing involves selecting the most relevant datasets for analysis, pre-processing them to ensure consistency and suitability for use in base predictors, and quantising the data for use with the HMM model. Two types of processed data are generated: averaged strain data and normalised strain data, each serving a different purpose in the analysis. The following subsections describe the data selection, pre-processing, and clustering procedures, leading to the final datasets used in the remainder of this study.

4.3.1. Data Selection

Three types of data were collected while testing the eight specimens: AE data, DIC data, and crack propagation images. The DIC data and images were captured every 251 cycles to minimise the total data volume, resulting in a lower temporal resolution than the AE data. While AE data contains extensive information, extracting meaningful insights is challenging. Since the specimen consists of three different components, the AE sensors also record signals from cracks in the other two, making it difficult to directly compare AE data from individual coupons to that of the specimen. Although AE offers more detailed temporal data, it was excluded from further analysis due to the complexity and time required to extract relevant features. Similarly, while crack images provide a clear depiction of degradation, extracting crack length measurements would also be time-consuming. Therefore, DIC data was chosen for analysis, as it is straightforward to extract and provides a strong monotonic relationship with the degradation process.

DIC data requires a reference image to determine the displacement of individual speckles. This reference image was captured with the specimen under a tensile load of 1 [kN]. To process the data, an appropriate subset size must be selected. A larger subset reduces noise but also decreases resolution. For this analysis, a subset of 19 pixels was chosen to calculate the strain field. Given that the strain field for the entire specimen is excessive and unnecessary, only the axial strain in the vertical direction (ϵ_{yy}) around the holes, where cracks and degradation are expected, was extracted. The location of the extracted strain data is shown in Figure 4.9. This line has a width equal to the 19-pixel subset size and represents an average across that width.

This raw data from specimen five can be seen in Figure 4.10; the raw data of the other specimens can then be found in Appendix A. These are the axial strains at the lines presented in Figure 4.9. It can be clearly seen that the strains generally increase in function over time. Next, it can also be observed that the component that fails first has the highest strains. This is the type of data that the base predictor will

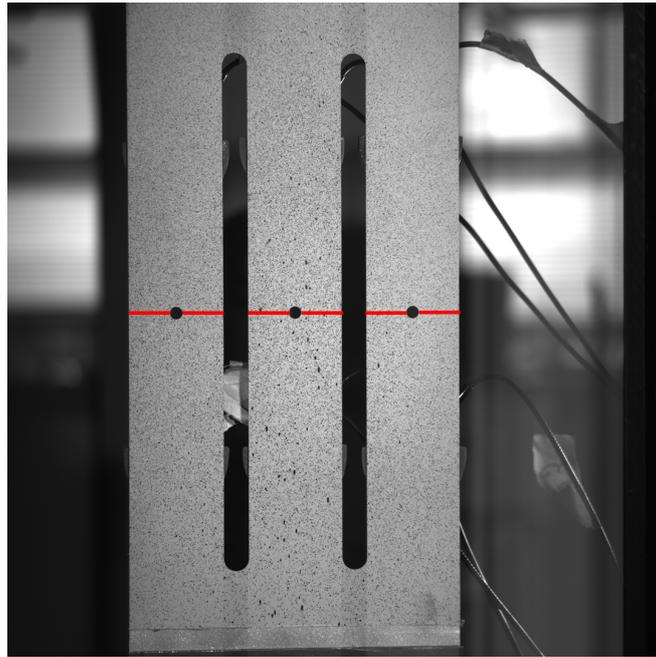


Figure 4.9: Location of the extracted strain data of the specimens.

need. This data will then be processed further to be used by the base predictor.

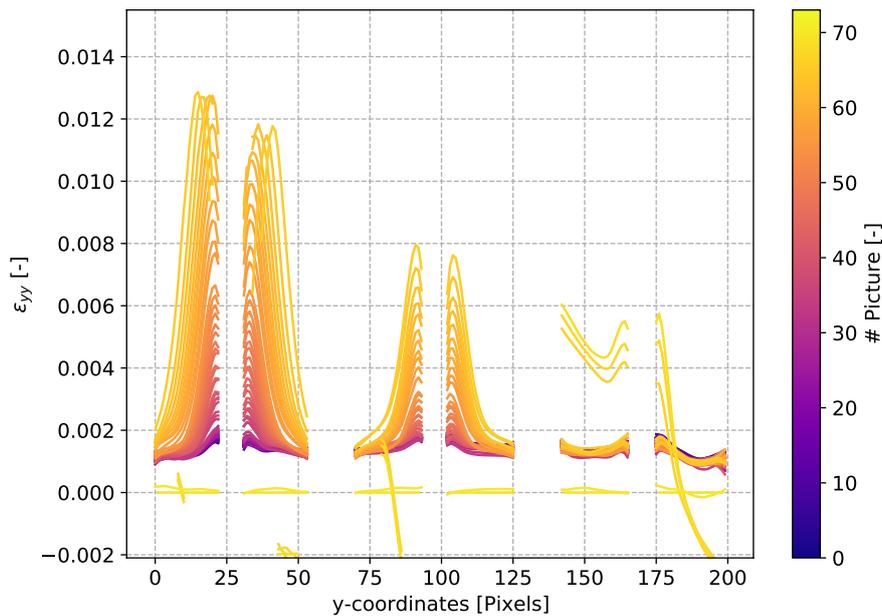


Figure 4.10: Extracted strain data from DIC measurement, specimen five.

4.3.2. Data Pre-Processing

Now, the data that will be used for the base predictor is selected. However, it must still be processed to be optimally utilised by the base predictor. So, to simplify the data, the average of every line of strains is calculated per component per DIC picture. These average strains can be seen in figures 4.11 to 4.13. A few things can be noted here. First, it is observed that the degradation histories differ significantly in the number of cycles per specimen. This is less ideal for the HMM base-predictor that will be used. Second, the range of maximum strain values per specimen differs greatly, ranging from 0.007 to 0.03. For certain

specimens, the strain values even become negative after the failure of components. This is caused by the extremely large displacement of the speckles after failure. These large deformations cause problems for the DIC software. Third, not every specimen's average strain values start from zero or the same value. This is also undesirable for the base predictor as it expects that the undegraded measurements have the same value.

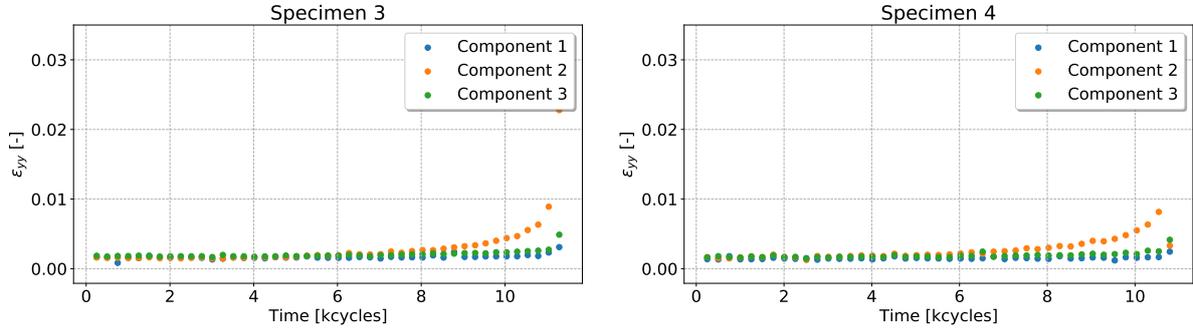


Figure 4.11: Average axial strains of the middle of specimens three and four.

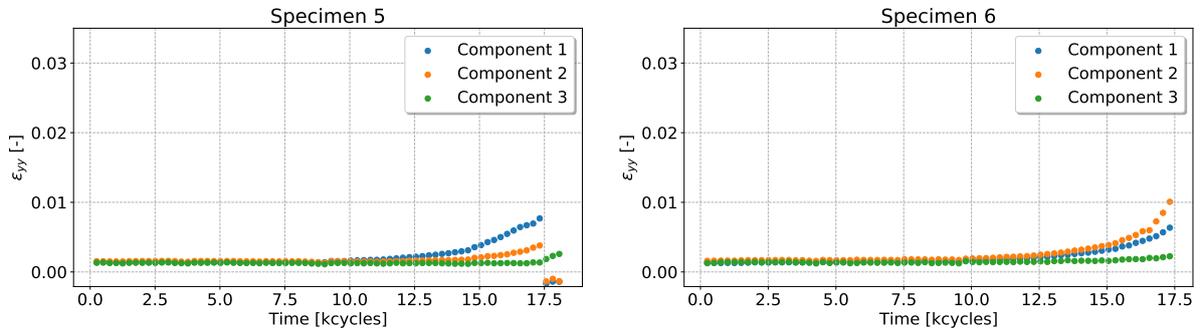


Figure 4.12: Average axial strains of the middle of specimens five and six.

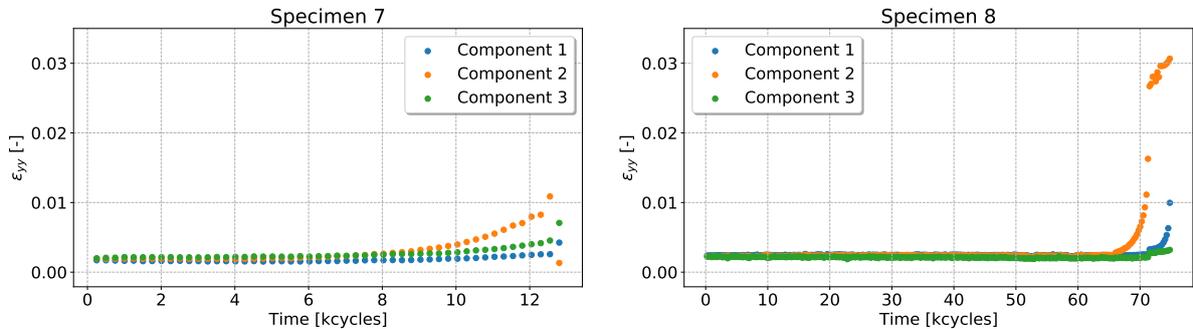


Figure 4.13: Average axial strains of the middle of specimens seven and eight.

The data processing is carried out in several steps to address the issues mentioned earlier. First, the data is truncated to include only the period from when the crack is first visible in the images to when the specimen fails, ensuring the dataset captures the actual degradation phase. Next, strain values are truncated at a threshold of 0.007, corresponding to the material's yield stress, calculated as $\epsilon = \frac{\sigma_y}{E} = 0.007$. Finally, all the values are subtracted by the first value observed. The outcome of these steps is illustrated in figures 4.14 to 4.16. This data will be used later as inputs for the base predictors and will be called the averaged strain data.

However, despite these steps, the data is not strictly monotonic, as the axial strain occasionally decreases over time, a characteristic undesirable in prognostics. To resolve this, a health indicator (HI) could help transform the data into a strictly monotonic time series. A normalised dataset was created to

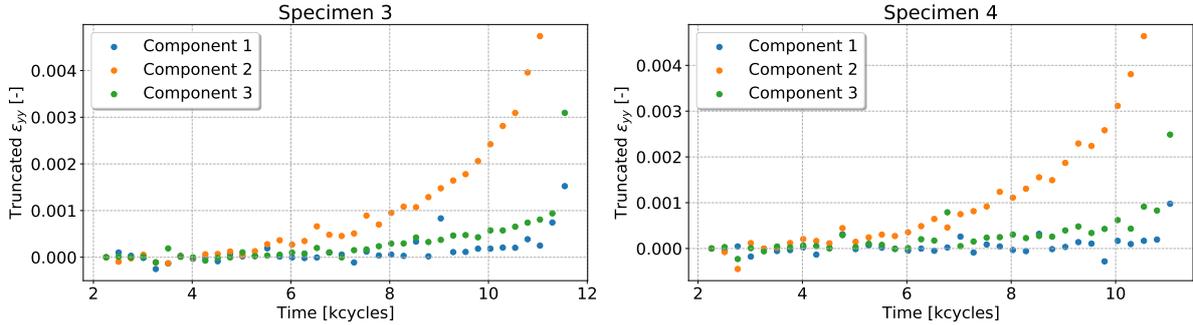


Figure 4.14: Truncated average strain data from specimens three and four.

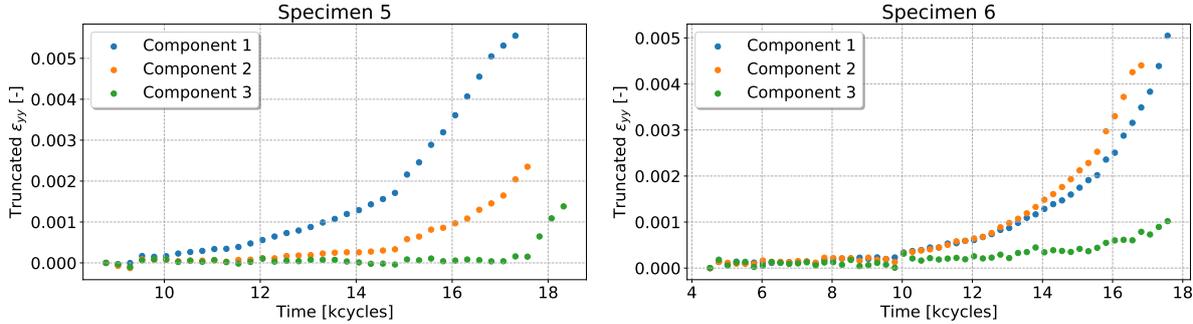


Figure 4.15: Truncated average strain data from specimens five and six.

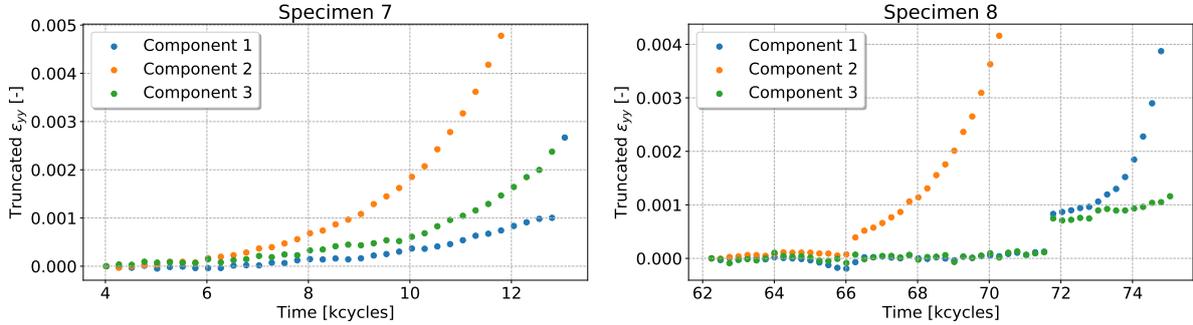


Figure 4.16: Truncated average strain data from specimens seven and eight.

explore the potential improvement in the IIM prediction. In this approach, axial strain values were cumulatively summed and divided by their maximum value, ensuring the data ranged from 0 to 1 and became strictly monotonic. Although this transformation cannot be applied in real-time operations, it mimics the behaviour of an effective HI, making it a useful representation for analysis. The results are displayed in figures 4.17 to 4.19, and this dataset will be referred to as the normalised data.

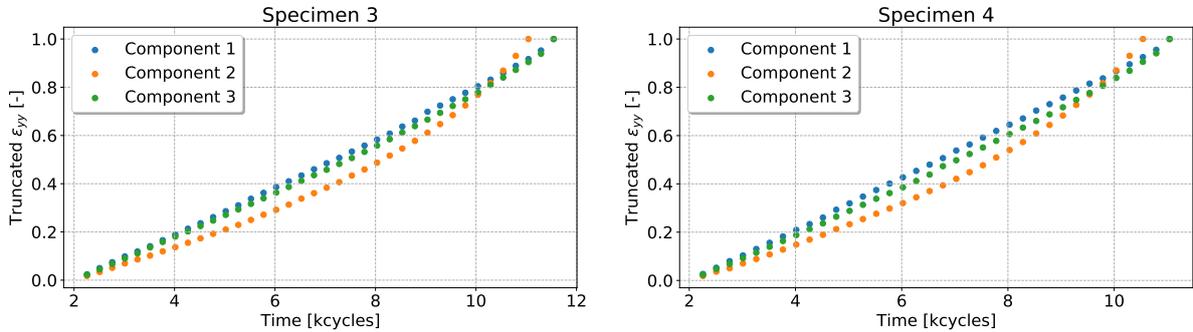


Figure 4.17: Normalised truncated strain data from specimens three and four.

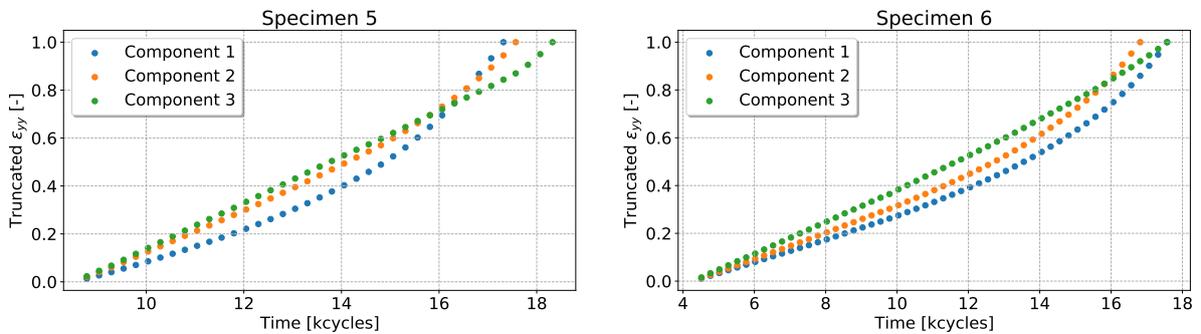


Figure 4.18: Normalised truncated strain data from specimens five and six.

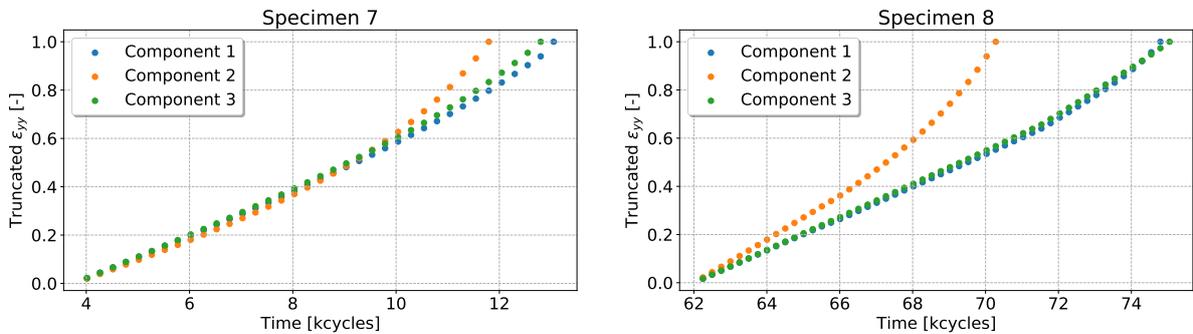


Figure 4.19: Normalised truncated strain data from specimens seven and eight.

The SVR base predictor will use the normalised and averaged strain data provided in this subsection. However, these two types of data must be converted to discrete data for the HMM base predictor. This will be discussed in the following subsection.

4.3.3. Data Clustering

For the HMM to function, discrete data is required. Thus, the continuous data from the previous subsection will be quantised using clustering techniques. Three clustering methods are considered: equal-range, quantile binning, and K-means clustering. The optimal clustering method and number of clusters are determined using the Mann-Kendall (MK) criterion, which evaluates monotonicity, a key feature for prognostic methods. Higher MK values indicate more monotonic clustering. Since there are

two different processed datasets, the MK criterion will be applied to both to compare their monotonicity. The final choice of clustering method and number of clusters will be based on the MK criterion applied to the coupon data.

In Figure 4.20, the three different clustering methods' MK values can be seen for the averaged strain datasets. Here, it is clear that the quantile method generally has the best monotonicity. Therefore, the quantile method is chosen. It is desirable to reduce the number of clusters as much as possible while keeping a high monotonicity to simplify the model. Therefore, the lowest number of clusters where the MK criterion is converged should be used. Figure 4.20 shows that the curves converge around 15 observations, which will be used for the coming RUL predictions for the averaged strain dataset.

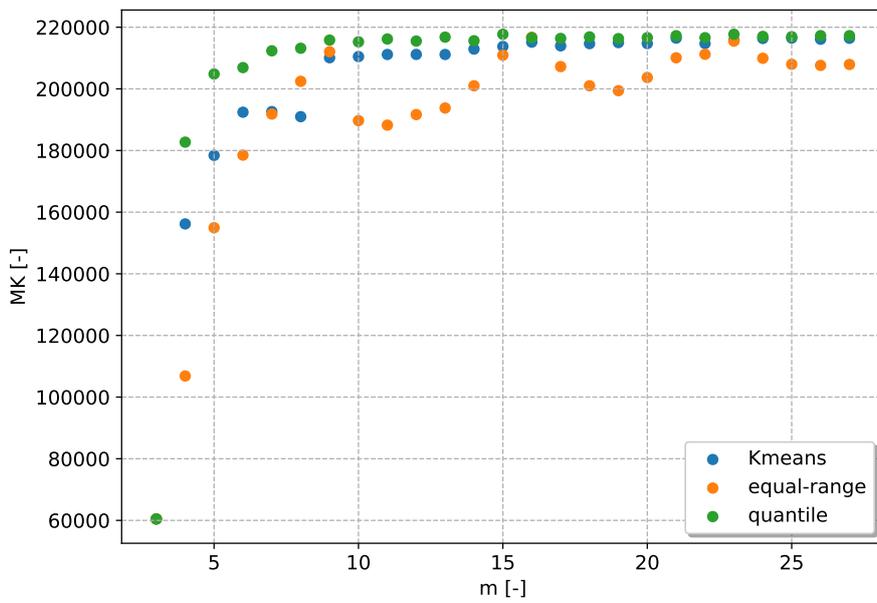


Figure 4.20: Mann-Kendall criterion of averaged strain data

The same should be done for the normalised data. The MK-criterion can be seen in Figure 4.21. There, it can be observed that the graph converges around 10. To determine the exact method a zoomed in figure is presented in Figure 4.22. From this, the Kmeans method performs slightly better and will be used for the normalised data.

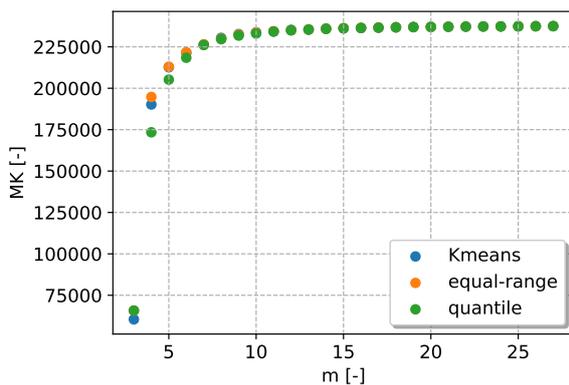


Figure 4.21: Mann-Kendall criterion of normalised data

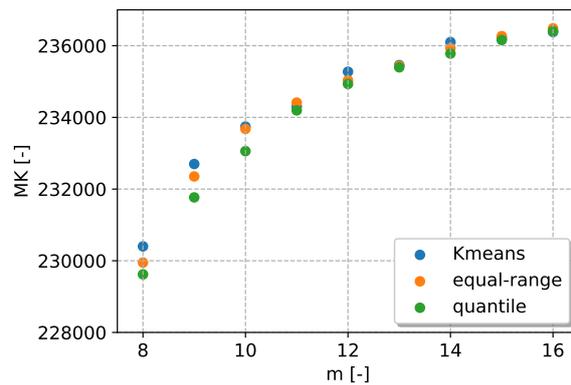


Figure 4.22: Zoomed in Mann-Kendall criterion of averaged strain data

Once the number of clusters and clustering method were determined, the data is represented in Figure 4.24 and Figure 4.23 for the averaged strain and normalised datasets, respectively. It is evident that the normalised data exhibits improved monotonicity compared to the averaged strain data, as there

are no decreases in the observations—values either remain constant or increase over time.

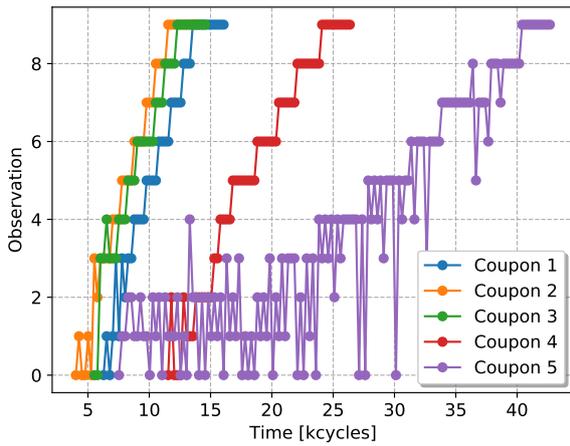


Figure 4.23: Clusters of the averaged strain dataset using 10 observations and the quantile clustering method.

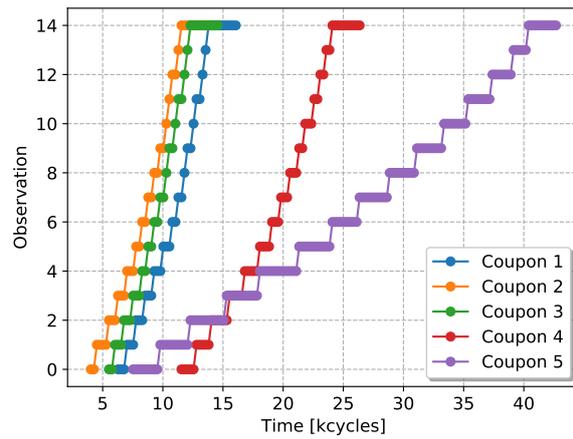


Figure 4.24: Clusters of the normalised dataset using 15 observations and the K-means clustering method.

4.4. Results of the Base Predictor

The IIM uses RUL predictions from the base predictors as input. These base predictors are trained and selected based on data from single open-hole coupons. The first subsection focuses on configuring the number of hidden states and determining the transition and observation matrices for the HMM. The second subsection addresses selecting parameters and kernel functions for the SVR.

4.4.1. HMM Model Determination

To achieve results from the HMM, the number of hidden states must be determined. This is done using the BIC as explained in section 3.1. The BIC and the model are different for the normalised and averaged strain data. Multiple HMMs are made and evaluated to calculate it. The models are also different because there are different amounts of clusters in the normalised and averaged strain data. Therefore, two different BIC scores are calculated. The BIC for the normalised HMM and averaged strain HMM can be respectively seen in Figure 4.26 and 4.25. From these figures, 4 hidden states are chosen for the model using the averaged strain data and five for the model using the normalised data.

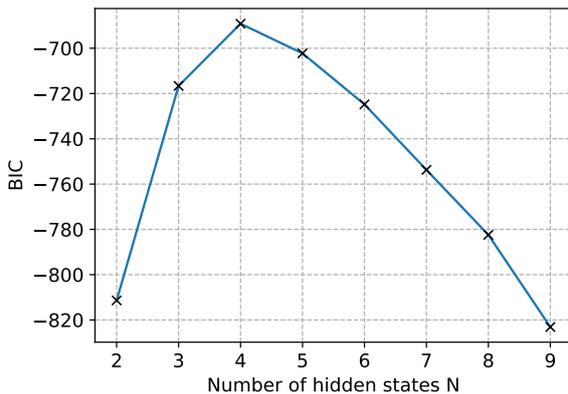


Figure 4.25: BIC criterion for the coupons' averaged strain data

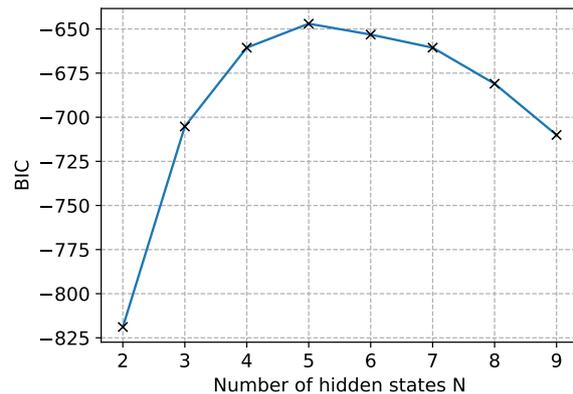


Figure 4.26: BIC criterion for the coupons' normalised data

With the number of observations and hidden states determined for the HMM on both datasets, the models can now be trained using the coupons' data. The averaged strain data's resulting transition and observation matrices are shown in Figure 4.27. It is important to verify that these matrices align with the expected behaviour of coupon degradation. As anticipated, the transition matrix indicates that the model expects the specimen to remain in its current state more frequently than transitioning to

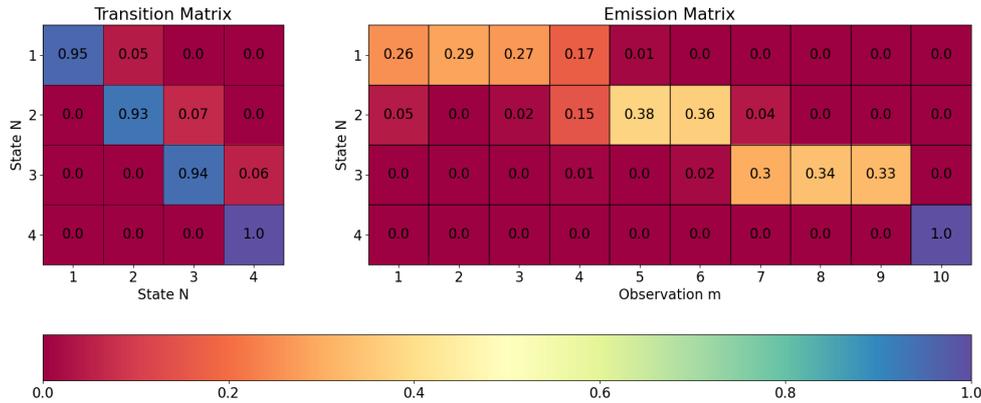


Figure 4.27: Representation of the transition and emission matrix of the HMM trained on the averaged strain dataset.

the next one. Additionally, the model can only transition between consecutive states, as expected. The emission matrix reveals a clear trend, with the first set of clusters corresponding to the initial state and progressively decreasing. This pattern confirms monotonic observations, improving the hidden state’s relationship and making predictions more accurate.

The results for the normalised dataset are quite similar and can be seen in Figure 4.28. These have the same general characteristics. However, the emission matrix of this HMM model has a more clearly defined relation between state and observation.

The trained HMM gives the following RUL predictions for the coupons presented in Figure B.1 and B.2 for the averaged strain data, and in Figure B.3 and B.4 for the normalised data. It can be noted that the RUL prediction of coupon 1 in Figure B.3 is not visible. This is caused by the similar data between coupon 1 and coupon 2, resulting in the exact same RUL predictions by the HMM.

The previous predictions were for the coupons’ data, which is also the training data. The results of the predictions of the triplets’ data cannot really be compared to the true RUL per component. However, the resulting predictions can be observed in Appendix C. Although the errors of the individual components’ predictions cannot be compared it is clear that the predictions of the normalised data have more trendable predictions. These phenomena will be further explored in the next subsection, together with the results of the IIM.

4.4.2. SVR Model Determination

The first step in building the SVR model is determining the appropriate kernel function. By examining the data that will be input into the SVR model, as shown in figures 4.29 and 4.30, several potential kernel functions can be considered as good candidates. The options to be explored include a power-law function, an exponential function, and a polynomial function, as described below. To choose the best function, a simple least-square regression is performed on the data first to determine the optimal parameters and then compare the average MSE of the different functions.

$$\left\{ \begin{array}{ll} \text{Power law:} & ax^b + c \\ \text{Exponential:} & ae^{bx} + c \\ \text{Polynomial:} & ax^2 + bx + c \end{array} \right.$$

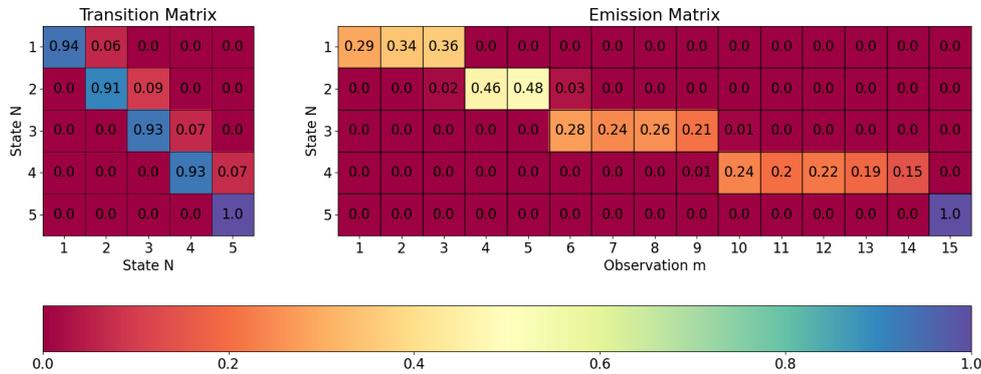


Figure 4.28: Representation of the transition and emission matrix of the HMM trained on the normalised dataset.

Table 4.1: Comparison of different function fits.

-	Averaged strain				Normalised			
Function	MSE	a	b	c	MSE	a	b	c
Exponential	34.47	29.98	-1286.04	2.944	27.84	49.22	-1.60	-10.21
Power law	-	-	-	-	36.83	1.98E5	-5.51E-5	-1.98E5
Polynomial	-	-	-	-	200.63	-10.90	1.57	9.39

The result of this comparison is provided in Table 4.1. For the averaged strain data, it can be observed that no results are provided for both the power function and the polynomial function. This is caused by the fact that the regression functions could not converge. For the normalised data, however, all functions converged. There, the exponential function clearly provides the best performance. The two different exponential functions will then be used for the SVR.

For the SVR, a search is then performed on the data of the coupons to determine all hyperparameters using the previously chosen function as explained in section 3.1. The coarse searching grid used for both types of data can be found in Equation 4.1, the refined searching grid can then be seen in Equation 4.2 and 4.3 for the averaged strain and normalised data respectively. The optimal parameters for the SVR can be found in Table 4.2.

By using these parameters the trained SVR eventually gives the following predictions seen in figures B.5 to B.8. The results of the predictions of the triplets' data can then be found in Appendix D. Here, it can be observed that while the results using the averaged strain data are quite erratic, the normalised data for the SVR gives predictions that seem to follow a straight line very well. This is in contrast to the predictions of the HMM, whose predictions are more stage-wise.

$$\begin{cases} C : \{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.5, 1, 10, 100, 500, 1000\} \\ \gamma : \{\text{scale}, \text{auto}\} \\ \epsilon : \{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.5\} \end{cases} \quad (4.1)$$

$$\begin{cases} C : \{10, 50, 100, 150, 200, 300, 400\} \\ \gamma : \{\text{scale}\} \\ \text{kernel} : \{\text{rbf}\} \\ \epsilon : \{0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09\} \end{cases} \quad (4.2)$$

$$\begin{cases} C : \{0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \\ \gamma : \{\text{scale}\} \\ \epsilon : \{0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.03, 0.04, 0.05\} \end{cases} \quad (4.3)$$

Table 4.2: Hyperparameter values for both SVR models

Parameter	Averaged strain	Normalised
C	100	0.9
Kernel	Exponential	Exponential
ϵ	0.07	0.02
γ	scaled	scaled

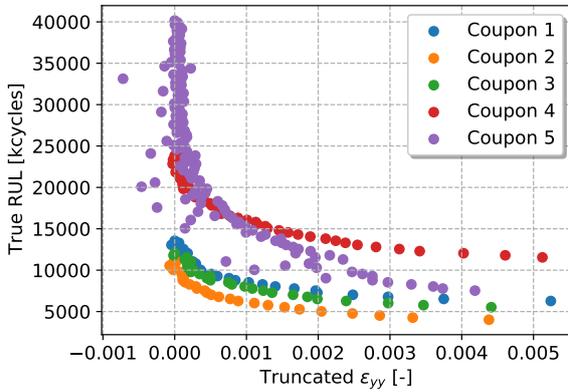


Figure 4.29: True RULs of the coupons in function of the truncated strains for the averaged strain data.

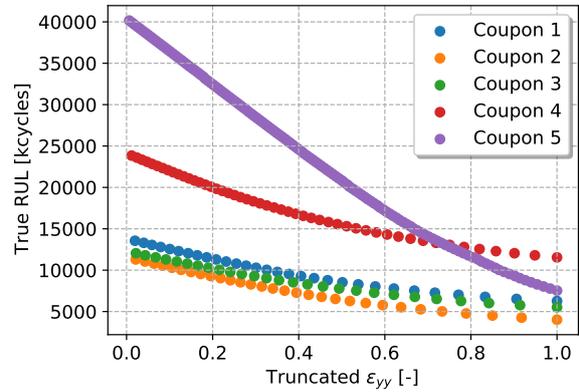


Figure 4.30: True RULs of the coupons in function of the truncated strains for the normalised data.

Using these parameters, the trained SVR produces the predictions for the coupons as shown in figures B.5 to B.8. The results for the predictions based on the triplets' data are provided in Appendix D. It can be observed that while the predictions using the averaged strain data appear quite erratic, those using the normalised data for the SVR tend to follow a straight line closely. This contrasts with the predictions from the HMM, which exhibit a more stage-wise behaviour.

4.5. Results and Discussion

The purpose of the IIM is to be used on larger aerospace structures, where typically only a single structure is tested until failure. This is due to the high cost, significant time investment, and the inherent complexity of such testing. So, for the initial results and to compare the different input data, the IIM will be trained offline on one specimen. This specimen will also be used to determine the hyperparameters as mentioned in subsection 3.3.3. The hyperparameters are determined separately for each type of input data, namely, normalised HMM, normalised SVR, averaged strain data HMM and averaged strain data SVR. Most parameters were similar as they are the same input data and have the same order of magnitude. The optimal k and number of iterations, however, did change. The hyperparameters for the four different IIMs can be found below in Table 4.3.

Table 4.3: Hyperparameters used for the different IIMs.

Parameter	Offline			Online			
	Learning rate	Iterations	Batch	Learning rate	k	Iterations	Batch
Normalised HMM	2e-5	2000	True	6e-5	5	100	False
Normalised SVR	2e-5	2000	True	6e-5	10	500	False
Averaged strain data HMM	2e-5	2000	True	4e-5	15	100	False
Averaged strain data SVR	2e-5	2000	True	4e-5	8	100	False

Now the offline model is determined, the results of the online predictions using the IIM can be found in Appendix E. This chapter will highlight a few examples, and the errors of the different results will be evaluated. First, the results of the IIM will be compared to the base predictors, which serve as a baseline. Then, the results of the IIM for the averaged strain data will be compared to the results of the normalised data. Next, the influence of the different base predictors will be investigated. Finally, a synthesis of these different results will be made.

4.5.1. Results of the IIM Compared to the Base Predictors

It is hard to determine whether the model, in general, provides good predictions or not. Yet, it can be compared to the most basic SLP available. Namely, the maximum of the input RULs. Here, three different MSEs are provided to compare the IIM. First is the error of the maximum of the base predictors. This should represent a baseline with which the IIM can be compared. Second, the IIM's prediction error is given when it is only trained offline. Third, the error of the IIM that has also been trained online is provided. The additional performance of the online training can also be observed by comparing the offline and online errors. These three errors of all specimens and with different input types can be found in Appendix F.

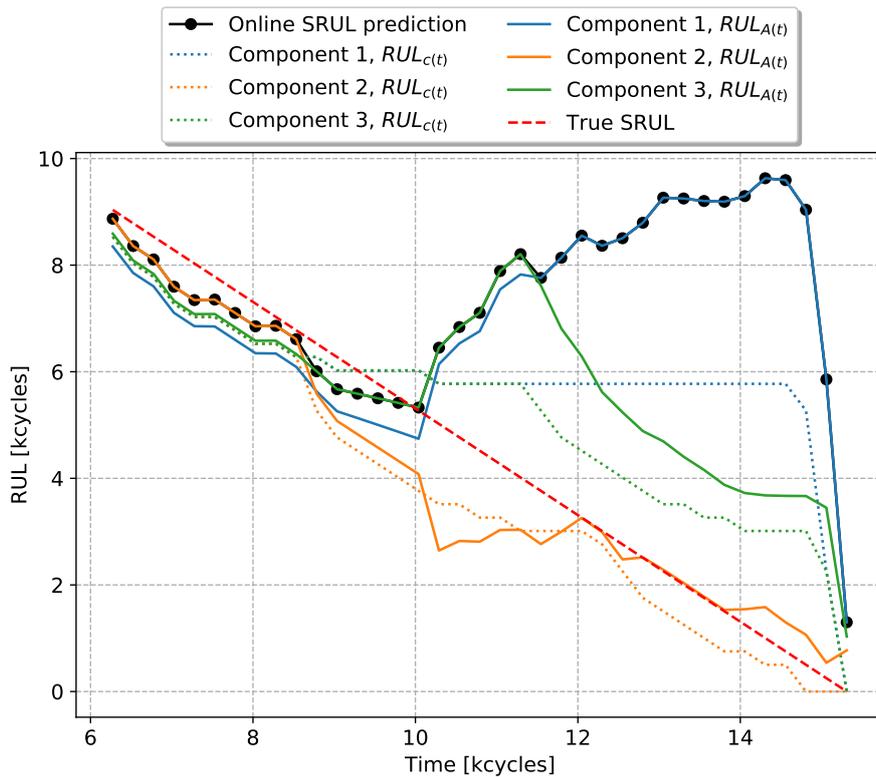


Figure 4.31: Results of the IIM trained online on specimen four. Using averaged strain data and RULs provided by the HMM base-predictor.

When examining these errors, it is evident that the online IIM errors are rarely consistently lower than those of the base predictor. In contrast, the offline IIM error often outperforms the online version and the base predictors. To better understand this behaviour, several trends can be noted. The IIM's predictions for the specimens, using averaged strain data and the HMM base predictors, show high error peaks during the second half of their fatigue life. As seen in figures 4.31 and 4.32, these high peaks occur when the base predictor provides very flat predictions. In such cases, the IIM assumes that this trend will continue; however, these components will experience rapid degradation when one component fails. This trend should be identified and remembered during the offline training. By looking at the offline IIM's error, it can be seen that the IIM does identify this trend in training offline, but when the online training starts, this is not remembered. These flat predictions can be seen in all predictions using the averaged strain data and these high errors. Further trends will be investigated by examining the difference between specific input data in the coming sections.

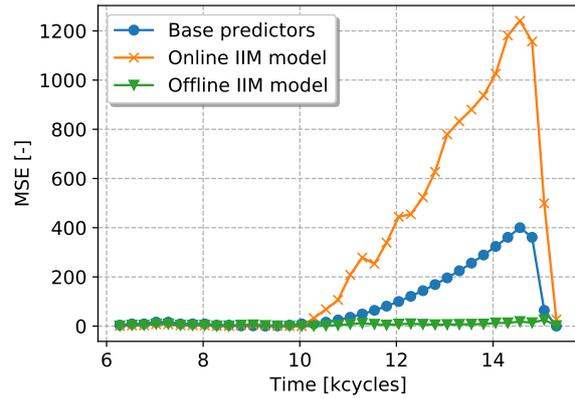


Figure 4.32: Error of the HMM base predictor and the IIM of specimen four using the averaged strain data.

In verifying the IIM (subsection 3.4.3), it was investigated whether the IIM could deal with a sudden failure or sudden big changes. There, the IIM showed that it could adapt its predictions reasonably fast to these sudden changes in input data. In Figure 4.33, shown below, a sudden big change does also occur. Here, the second component fails relatively early compared to the other two. Consequently, the other two RUL_s_c decrease dramatically in the following measurements. Although the IIM had way too high RUL_A predictions for the first and third components, it quickly adjusts to the new situation and provides decent predictions for the SRUL near the end. This is a positive indicator of the IIM's ability to deal with sudden changes in the input data and confirms the verification.

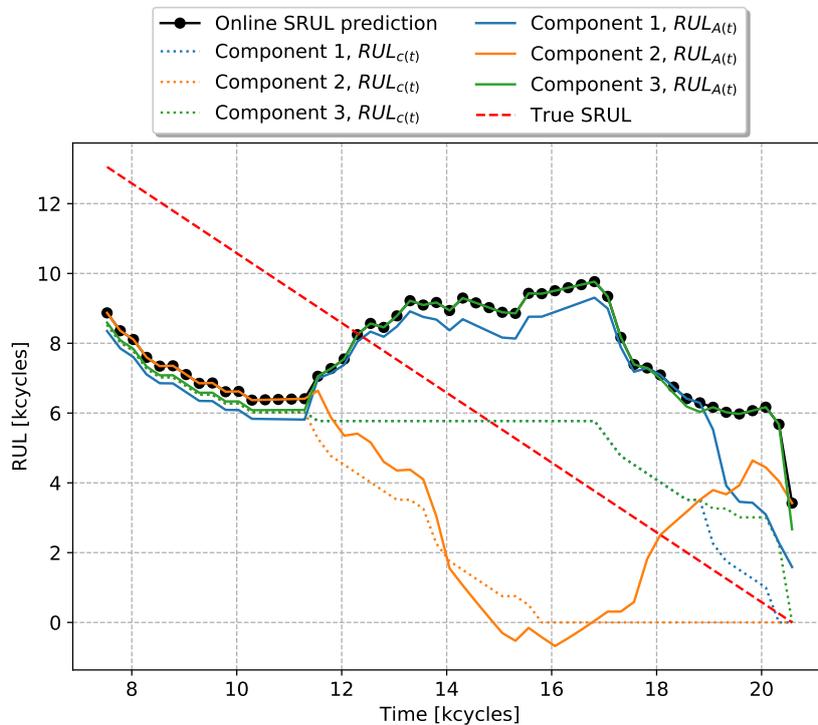


Figure 4.33: Prediction of the IIM trained online on specimen eight. Using averaged strain data and RULs provided by the HMM base predictor.

4.5.2. Comparison of the IIM Results of the Normalised Data VS Averaged Strain Data

When looking at figures H.1 to H.5, there is a clear difference in error between the normalised and averaged strain data. Around the middle of the measurements, the predictions of the IIM using averaged strain data increase and are always higher than the error of the IIM using the normalised data. This could be explained by the increase of linear smoothness in the RUL_{S_c} . The curves from the normalised data are much straighter and exhibit less erratic behaviour. The R^2 method often used in regression confirms this increase in linear smoothness as provided in Figure 4.34. The closer the R^2 value is to one, the higher the linear smoothness of the RUL_{S_c} is. A clear difference between the normalised and averaged strain data can be observed. This translates to prediction errors as well.

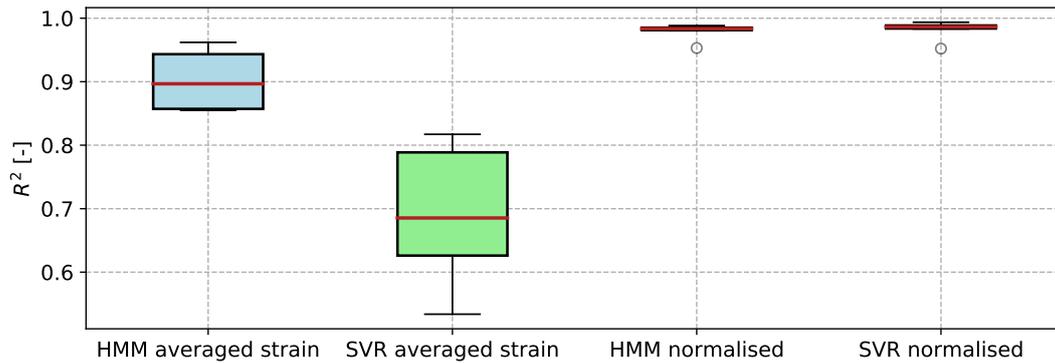


Figure 4.34: Boxplot representing the R^2 -values of the different kinds of RUL_{S_c} .

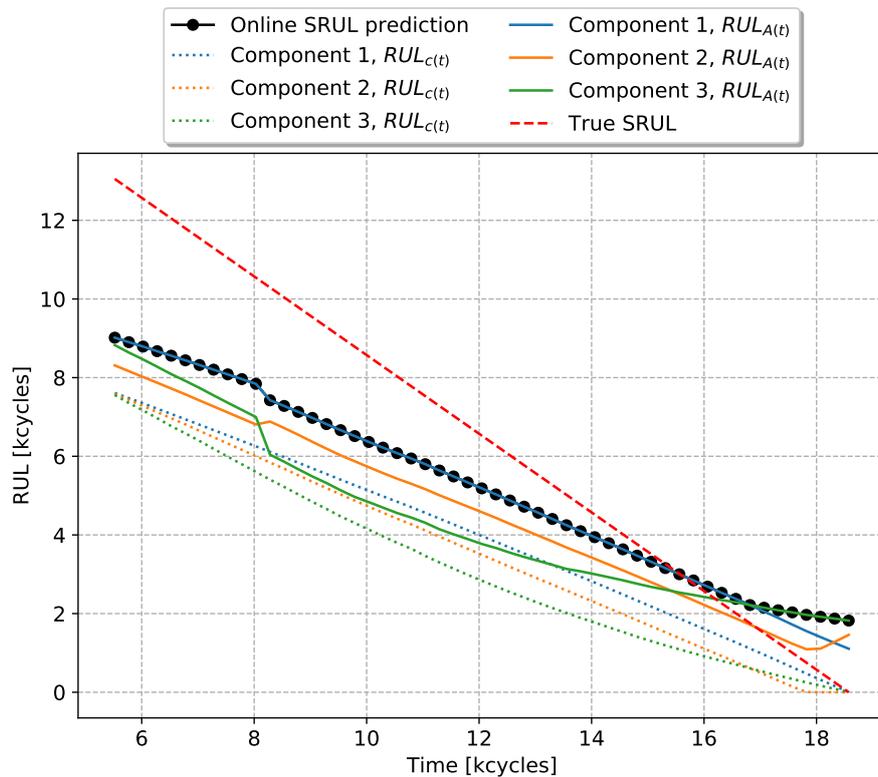


Figure 4.35: Results of the IIM trained online on specimen six. Normalised data and RULs provided by the SVR base predictor were used.

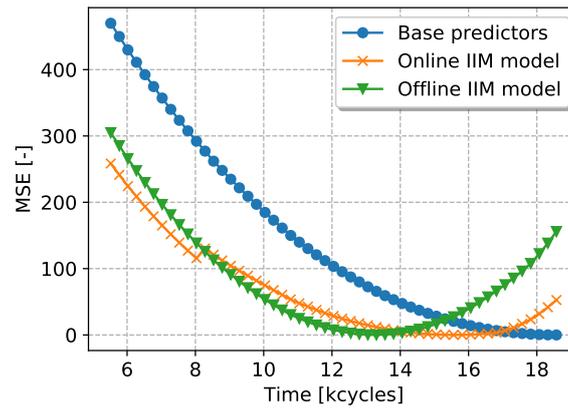


Figure 4.36: Error of the HMM base predictor and the IIM of specimen four using the averaged strain data.

The predictions generated by the IIM, with the SVR model as the base predictor and using the normalised data, exhibit a highly linear trend. An example can be found below in Figure 4.35 where it can be seen that there are no kinks in the RUL_{S_c} , but only a steady decrease. This is similar to all other specimens. This type of data caused the IIM to provide higher accuracy for most of the fatigue life for all specimens, except at the end-of-life stage. An example of this error can be found in Figure 4.36 and this is similar to all other specimens, which can be found in figures F.16 to F.20.

4.5.3. Comparison of the IIM Results of the Different Base Predictors

The differences between the SVR and HMM are present but less pronounced than those between the normalised and averaged strain data. For the averaged strain data in figures G.1 to G.3 and G.5 the error start relatively low for both base predictors and rises in the second half of the predictions where at the end it drops again. These peaks are generally much higher for the HMM than for the SVR (except for specimen 7).

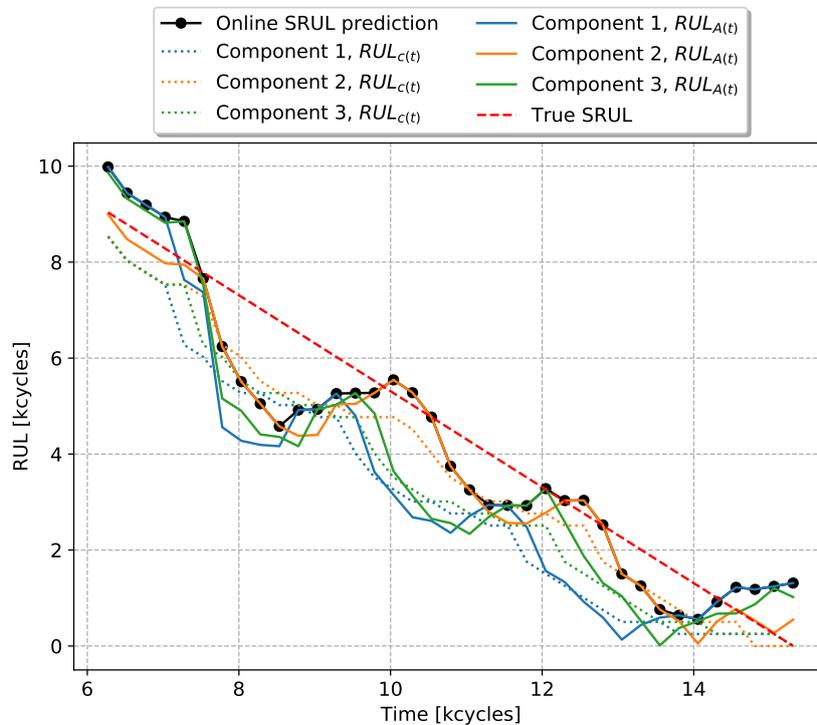


Figure 4.37: Results of the IIM trained online on specimen four. Using normalised data and RULs provided by the HMM base predictor.

There is also a specific phenomenon in the error of the HMM. In the plots of the HMM using normalised data, there are clear peaks and drops. This can be seen below in figures 4.37 and 4.38 where an example of the predictions and the corresponding errors can be found. In Figure 4.39, the error of the IIM using the HMM as the base predictor is illustrated. The vertical dotted lines indicate transitions between hidden states in the base predictions. The error peaks are closely aligned with these transitions between hidden states. After changing the hidden state, the error rises a few timesteps, after which it drops again. This is caused by the online learning scheme that extrapolates the latest trends. In this case, the RUL_{S_c} stay equal, so the online learning will extrapolate this to have a higher RUL_A while this is only momentary. This is partially caused by the GD method. As a normal GD and not a batched GD is used in online learning, the latest trends will contribute more to the new RUL_{S_A} than the initial RUL_{S_c} .

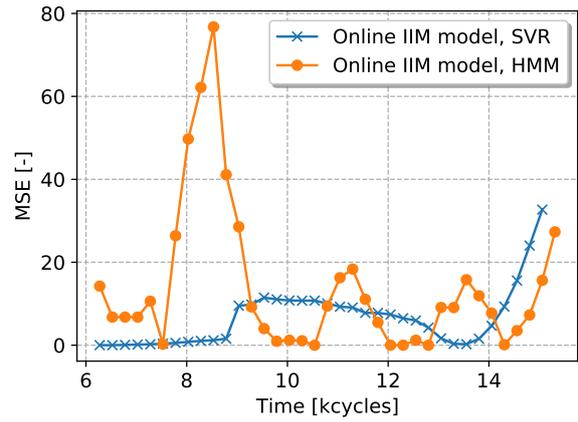


Figure 4.38: Error of the IIM for both the model based on the SVR and HMM base predictors of specimen four using the normalised data.

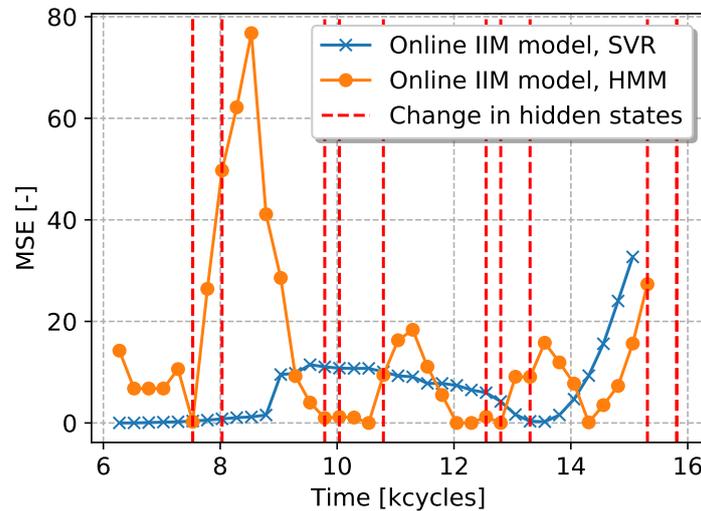


Figure 4.39: Errors of the IIM for both the model based on the SVR and HMM base predictors of specimen four using the normalised data with the state transition indicated.

4.5.4. Error of the IIM with Different Amounts of Training Data

For data-driven methods, more data generally provides better results. To see the effects of a different amount of data on the performance of the IIM, the IIM is trained offline with an increasing amount of training data. Specifically, the model is first trained offline only on specimen three and tested on specimens six to eight. Then, it is trained on both specimens three and four and finally on specimens three, four and five. All the results can be found in Appendix I, an example of the change in error for the IIM using SVR and normalised data an example is given below in Figure 4.40.

Based on the observed errors, the additional offline training data appears to have little to no effect on the errors of the online-trained IIM. The performance of the IIM without online training barely changes as well, where there is sometimes a small improvement and sometimes a small deterioration. Except in specimens figures I.7 and I.9 a clear and large improvement took place, yet no improvement was seen in the online predictions.

This brings up two key considerations: First, offline training minimally impacts the IIM's performance across all scenarios. Second, the model may lack sufficient complexity to improve its predictions by

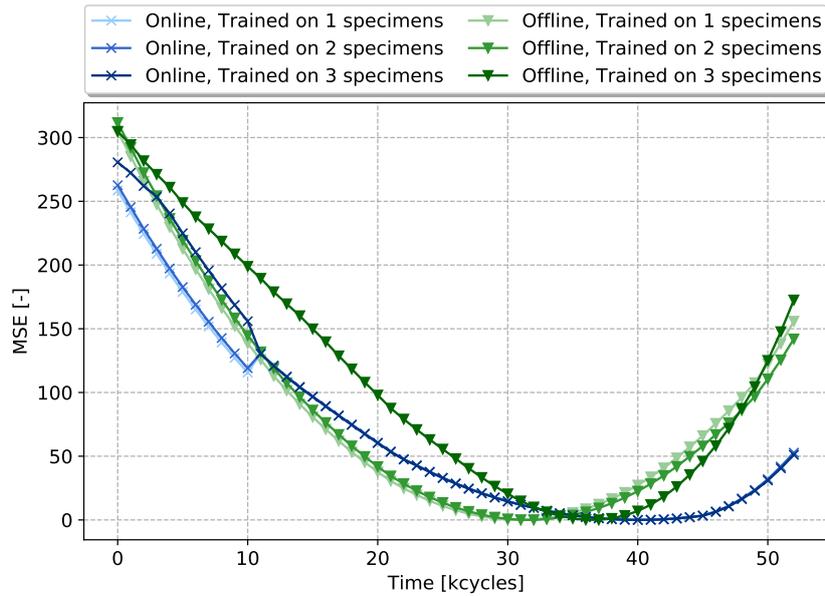


Figure 4.40: Errors of the IIM for different amounts of training data, using the normalised data and SVR base predictor.

increasing training data. While other factors could explain the lack of improvement, such as insufficiently trendable data or the model's complexity requiring a much larger dataset, the model's simplicity is likely the primary issue. Given that only 9 parameters can be updated, it is probable that the model is not complex enough.

4.5.5. Concluding Remarks about the Results of the IIM

Based on the analysis of the differences in results and performance of the IIM with respect to specific changes, a more general evaluation of the IIM as an SLP can be provided. This evaluation will help in addressing the research question in the conclusions.

The primary limitations of the IIM are as follows:

- The IIM requires high levels of monotonicity and linear smoothness to make accurate predictions throughout the lifespan of the specimens.
- Towards the end of the specimens' life, the IIM's predictions degrade significantly, while the predictions of the base predictors improve in accuracy.
- The model's simplicity allows it to perform well with minimal training data. However, this simplicity also demands very high-quality input data and contributes to the inaccuracy of final predictions.

5

Conclusions and Recommendations

The aim of this thesis was to answer the following research question:

How can the SRUL, of three open-hole coupons that are connected at the top and bottom, be predicted by the use of SLP techniques, such as the IIM, based on prognostic models that are trained on single open-hole coupons?

1. *How could the IIM be modified to be used for the given aerospace structure?*
2. *How does the IIM model respond to different types of input data and base predictors?*
3. *What is the behaviour of the prediction error based on a varying amount of training data?*

To achieve this, the IIM was adapted for application to aerospace structures. Two base predictors, namely Support Vector Regression (SVR) and the Hidden Markov Model (HMM), were selected to provide remaining useful life (RUL) predictions for each component. These RUL predictions were inputs to the IIM to estimate the overall system's RUL (SRUL). A specimen consisting of open-hole coupons connected at the top and bottom was subjected to fatigue loading until failure, and the SRUL was estimated using Digital Image Correlation (DIC) strain data. Two types of data were used as inputs for the models: the average strain data directly obtained from the DIC and the normalised data, generated by applying the cumulative sum and normalising it to represent a health indicator (HI).

To answer the first research question, the IIM was adapted by shifting its core functionality to use RUL predictions from base predictors instead of operability measures. The environmental contribution was excluded, as it fell outside the scope of this research. Due to the limited data available for aerospace structures, determining hyperparameters involved a significant amount of guesswork. The model yielded promising results for specific types of input data, where the prediction error was mostly lower than that of the individual base predictors. For future work, implementing advanced feature extraction techniques to generate health indicators (HIs) that enhance the monotonicity and linear smoothness of the $RULs_c$ would be valuable.

In response to the second research question, the IIM's performance varied considerably based on the type of input data. The model produced the best predictions when the input data was the most smooth and monotonic. Specifically, during the online learning phase, the IIM excelled with normalised data, such as that generated by the SVR model, where predictions were smooth. However, the HMM stage-wise predictions introduced large error spikes. The IIM's online error outperformed its offline error when using normalised SVR data. In contrast, the averaged strain data resulted in poor performance, primarily due to the base predictors providing inaccurate and erratic results. Therefore, it is advisable to select advanced, high-performing base predictors for improved outcomes.

Addressing the third research question, the prediction error exhibited minimal variation despite changes in training data. Sometimes, the offline error was marginally better, but this was rare. Two main factors underlie this behaviour: firstly, the model's complexity may not be sufficient to capture the degradation characteristics fully; secondly, the online learning phase quickly mitigates the influence of offline training, rendering the volume of training data less impactful. This suggests that when exploring

different SLP models for aerospace structures, their complexity should still be considered to ensure they can be trained on a limited number of degradation histories, but that a small increase in complexity over the IIM might improve the predictions.

To fully address the research question, the SRUL of the proposed specimen can be predicted using the SVR as a base predictor, with normalised data representing an HI as input to a modified version of the IIM. The online training phase successfully adjusted the IIM to the specific characteristics of the specimen. However, the behaviour of the SRUL prediction error remains unsatisfactory, suggesting that certain improvements and considerations should be considered moving forward.

One limitation persisted despite changes in the amount of training data or input data: prediction accuracy occasionally declined towards the end. In certain cases, across various combinations of data and base predictor types, the error increased during the later stages of the predictions. Although the base predictors became more accurate near the end of their predictions, the IIM did not account for this, resulting in elevated error levels.

As previously discussed, increasing the amount of training data had little impact on the online learning phase, as it quickly overrides the connections learned during offline training. Additionally, the error peaks observed near the end of the predictions can be attributed to the model's lack of memory. Therefore, it is recommended that SLP models for aerospace structures incorporate some form of embedded memory. This addition would allow the online and offline training phases to reinforce each other, capturing temporal dependencies more effectively and increasing the weight of base predictors toward the end of predictions. While adding memory is likely to increase model complexity, it could assist in identifying trends that may shift or persist over time. It should be investigated whether this added complexity becomes a limitation.

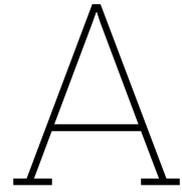
Although the original IIM from [9] served as a guide, the adaptation of the IIM remains open to further investigation. Modifying the formula or the training scheme can lead to significant changes in the results. For instance, the relative degradation term, which has undergone several iterations, is not fixed and may be replaced with a similar alternative that could yield better performance. Additionally, altering the online training schemes could introduce different influences on the model's behaviour. These possibilities highlight areas where further exploration and adjustments may enhance the model's effectiveness.

The IIM was initially selected as the SLP model due to its promising properties compared to other existing models. Although its performance in this thesis was found to be insufficient, there are still many aspects worth exploring. Incorporating the suggested recommendations and investigating potential adaptations could significantly enhance its results or offer valuable insights into implementing SLPs for aerospace structures.

References

- [1] Alfredo Güemes et al. “Structural health monitoring for advanced composite structures: A review”. English. In: *Journal of Composites Science* 4.1 (2020). Publisher: MDPI AG. ISSN: 2504477X (ISSN). DOI: 10.3390/jcs4010013. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85085739465&doi=10.3390%2fjcs4010013&partnerID=40&md5=2f1f04efc40173205d94537680eb0fdc>.
- [2] Enrico Zio. “Some Challenges and Opportunities in Reliability Engineering”. en. In: *IEEE Transactions on Reliability* 65.4 (Dec. 2016), pp. 1769–1782. ISSN: 0018-9529, 1558-1721. DOI: 10.1109/TR.2016.2591504. URL: <http://ieeexplore.ieee.org/document/7530921/> (visited on 11/17/2023).
- [3] Agnes A.R. Broer, Rinze Benedictus, and Dimitrios Zarouchas. “The Need for Multi-Sensor Data Fusion in Structural Health Monitoring of Composite Aircraft Structures”. English. In: *Aerospace* 9.4 (2022). Publisher: MDPI. ISSN: 22264310 (ISSN). DOI: 10.3390/aerospace9040183. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85128230733&doi=10.3390%2faerospace9040183&partnerID=40&md5=caf3cc6e139c7316a4dee6908ba5c3f3>.
- [4] Ferhat Tamssaouet et al. “System-level failure prognostics: Literature review and main challenges”. en. In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 237.3 (June 2023). Publisher: SAGE Publications, pp. 524–545. ISSN: 1748-006X. DOI: 10.1177/1748006X221118448. URL: <https://doi.org/10.1177/1748006X221118448> (visited on 10/24/2023).
- [5] Oscar Carpentier. *System-Level Prognostics for Aerospace Structures: A Literature Study*. Feb. 2024.
- [6] Leonardo Ramos Rodrigues. “Remaining Useful Life Prediction for Multiple-Component Systems Based on a System-Level Performance Indicator”. In: *IEEE/ASME Transactions on Mechatronics* 23.1 (Feb. 2018). Conference Name: IEEE/ASME Transactions on Mechatronics, pp. 141–150. ISSN: 1941-014X. DOI: 10.1109/TMECH.2017.2713722. URL: <https://ieeexplore.ieee.org/abstract/document/7944653> (visited on 10/24/2023).
- [7] Hamed Khorasgani, Gautam Biswas, and Shankar Sankararaman. “Methodologies for system-level remaining useful life prediction”. In: *Reliability Engineering & System Safety* 154 (Oct. 2016), pp. 8–18. ISSN: 0951-8320. DOI: 10.1016/j.res.2016.05.006. URL: <https://www.sciencedirect.com/science/article/pii/S0951832016300503> (visited on 10/27/2023).
- [8] Georgios Galanopoulos et al. “A data driven methodology for upscaling remaining useful life predictions: From single- to multi-stiffened composite panels”. In: *Composites Part C: Open Access* 11 (July 2023), p. 100366. ISSN: 2666-6820. DOI: 10.1016/j.jcom.2023.100366. URL: <https://www.sciencedirect.com/science/article/pii/S2666682023000221> (visited on 09/22/2023).
- [9] Ferhat Tamssaouet et al. “A Fresh new look for system-level prognostics: Handling multi-component interactions, mission profile impacts, and uncertainty quantification”. en. In: *International Journal of Prognostics and Health Management* 12.2 (Sept. 2021). Number: 2. ISSN: 2153-2648. DOI: 10.36001/ijphm.2021.v12i2.2777. URL: <https://papers.phmsociety.org/index.php/ijphm/article/view/2777> (visited on 10/30/2023).
- [10] Nick Eleftheroglou and Theodoros Loutas. “Fatigue damage diagnostics and prognostics of composites utilizing structural health monitoring data and stochastic processes”. en. In: *Structural Health Monitoring* 15.4 (July 2016). Publisher: SAGE Publications, pp. 473–488. ISSN: 1475-9217. DOI: 10.1177/1475921716646579. URL: <https://doi.org/10.1177/1475921716646579> (visited on 12/29/2023).
- [11] Nick Eleftheroglou, Dimitrios Zarouchas, and Rinze Benedictus. “An adaptive probabilistic data-driven methodology for prognosis of the fatigue life of composite structures”. English. In: *Composite Structures* 245 (2020). ISSN: 0263-8223. DOI: 10.1016/j.compstruct.2020.112386.

- [12] Mariana Salinas-Camus and Nick Eleftheroglou. "Uncertainty in Aircraft Turbofan Engine Prognostics on the C-MAPSS Dataset". en. In: *PHM Society European Conference* 8.1 (June 2024). Number: 1, pp. 10–10. ISSN: 2325-016X. DOI: 10.36001/phme.2024.v8i1.4007. URL: <https://www.papers.phmsociety.org/index.php/phme/article/view/4007> (visited on 09/04/2024).
- [13] Mariette Awad and Rahul Khanna. "Hidden Markov Model". en. In: *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Ed. by Mariette Awad and Rahul Khanna. Berkeley, CA: Apress, 2015, pp. 81–104. ISBN: 978-1-4302-5990-9. DOI: 10.1007/978-1-4302-5990-9_5. URL: https://doi.org/10.1007/978-1-4302-5990-9_5 (visited on 08/05/2024).
- [14] Richard Durbin et al. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. en. 1st ed. Cambridge University Press, Apr. 1998. ISBN: 978-0-521-62041-3. DOI: 10.1017/CB09780511790492. URL: <https://www.cambridge.org/core/product/identifier/9780511790492/type/book> (visited on 07/02/2024).
- [15] Dimitrios Zarouchas and Nick Eleftheroglou. "In-situ fatigue damage analysis and prognostics of composite structures based on health monitoring data". In: *Fatigue Life Prediction of Composites and Composite Structures (Second Edition)*. Ed. by Anastasios P. Vassilopoulos. Vol. 20. Woodhead Publishing Series in Composites Science and Engineering. Woodhead Publishing, Jan. 2020, pp. 711–739. ISBN: 978-0-08-102575-8. DOI: 10.1016/B978-0-08-102575-8.00020-6. URL: <https://www.sciencedirect.com/science/article/pii/B9780081025758000206> (visited on 09/25/2023).
- [16] Emad A. Mohammed, Christopher Naugler, and Behrouz H. Far. "Chapter 32 - Emerging Business Intelligence Framework for a Clinical Laboratory Through Big Data Analytics". In: *Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology*. Ed. by Quoc Nam Tran and Hamid Arabnia. Emerging Trends in Computer Science and Applied Computing. Boston: Morgan Kaufmann, Jan. 2015, pp. 577–602. ISBN: 978-0-12-802508-6. DOI: 10.1016/B978-0-12-802508-6.00032-6. URL: <https://www.sciencedirect.com/science/article/pii/B9780128025086000326> (visited on 09/12/2024).
- [17] Mariette Awad and Rahul Khanna. "Support Vector Regression". en. In: *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Ed. by Mariette Awad and Rahul Khanna. Berkeley, CA: Apress, 2015, pp. 67–80. ISBN: 978-1-4302-5990-9. DOI: 10.1007/978-1-4302-5990-9_4. URL: https://doi.org/10.1007/978-1-4302-5990-9_4 (visited on 08/02/2024).
- [18] Ferhat Tamssaouet, Thi Phuong Khanh Nguyen, and Kamal Medjaher. "System-level Prognostics Based on Inoperability Input-output Model". en. In: *Annual Conference of the PHM Society* 10.1 (Sept. 2018). Number: 1. ISSN: 2325-0178. DOI: 10.36001/phmconf.2018.v10i1.487. URL: <https://papers.phmsociety.org/index.php/phmconf/article/view/487> (visited on 10/26/2023).
- [19] Yennhi95zz. #4. *A Beginner's Guide to Gradient Descent in Machine Learning*. en. May 2023. URL: <https://medium.com/@yennhi95zz/4-a-beginners-guide-to-gradient-descent-in-machine-learning-773ba7cd3dfe> (visited on 05/31/2024).



Raw Strain-data of Specimens

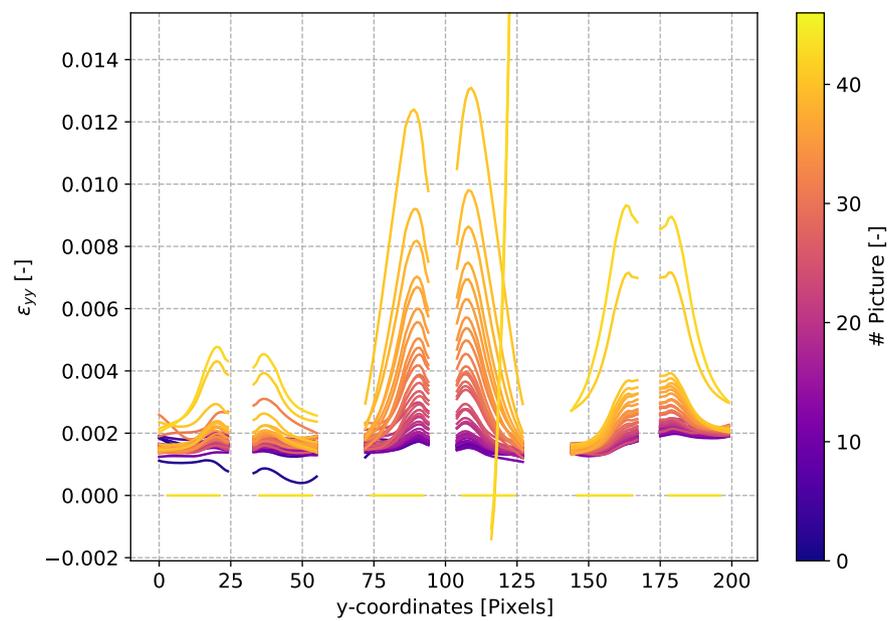


Figure A.1: Extracted strain data from DIC measurement, specimen three.

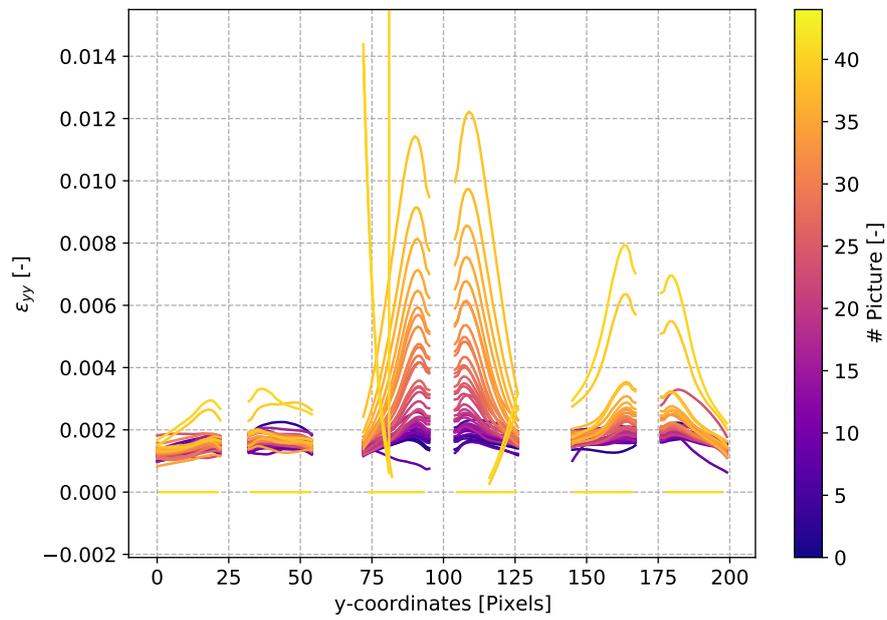


Figure A.2: Extracted strain data from DIC measurement, specimen four.

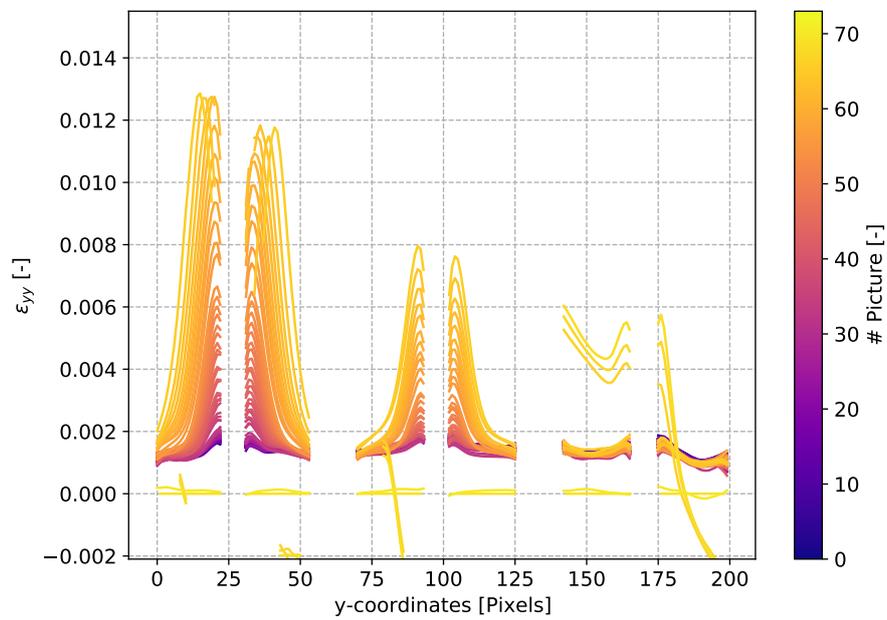


Figure A.3: Extracted strain data from DIC measurement, specimen five.

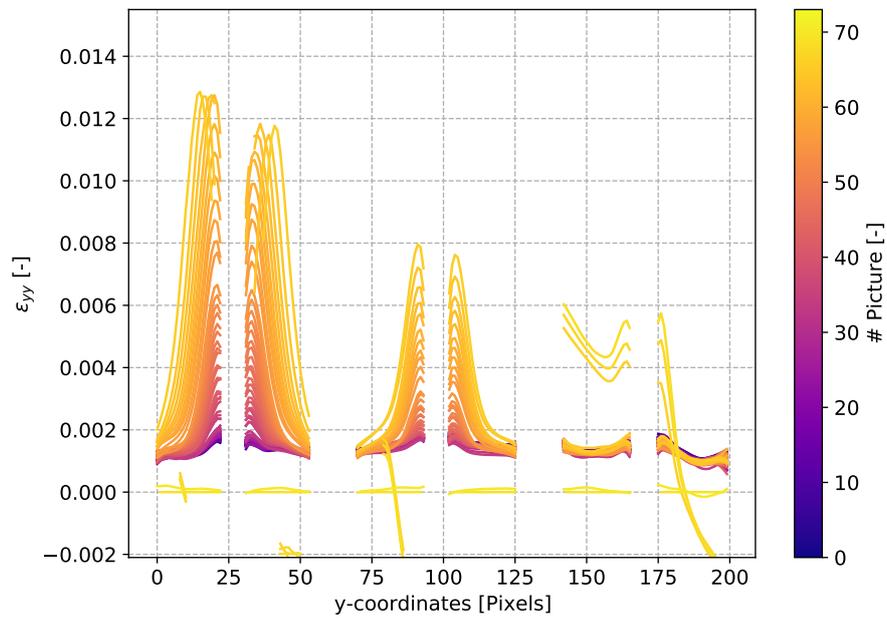


Figure A.4: Extracted strain data from DIC measurement, specimen six.

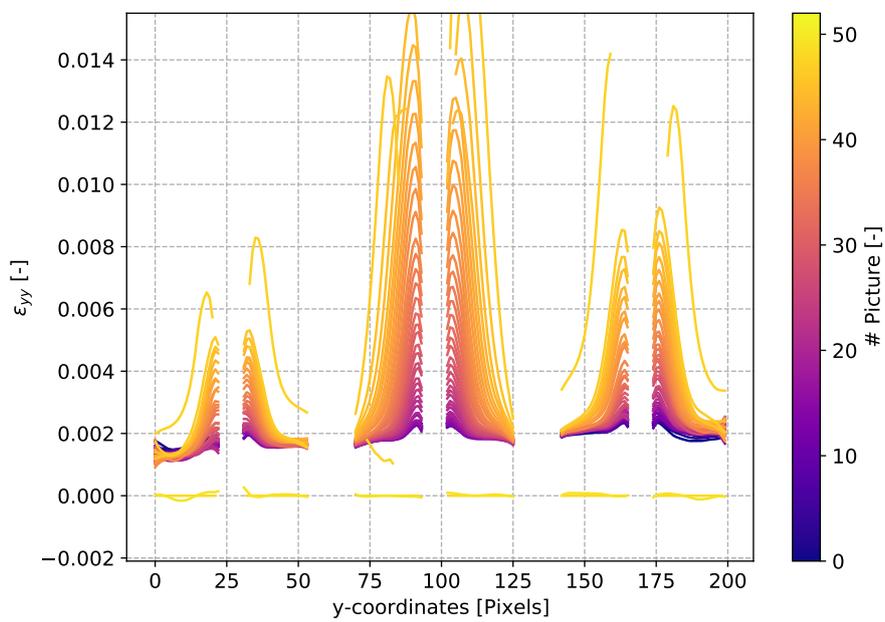


Figure A.5: Extracted strain data from DIC measurement, specimen seven.

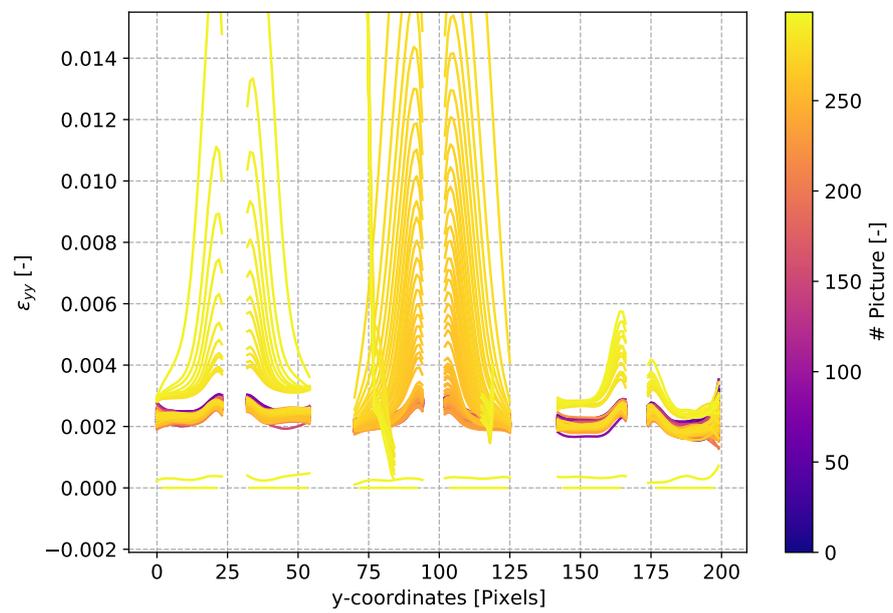


Figure A.6: Extracted strain data from DIC measurement, specimen eight.

B

Results of the base predictors for the Coupons

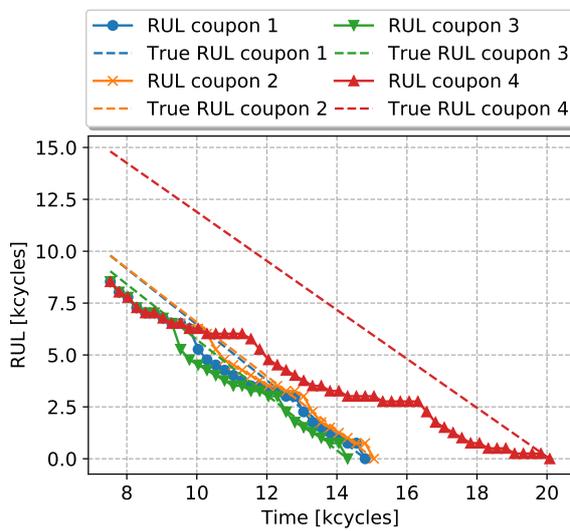


Figure B.1: RUL prediction of coupons 1 until 4 using the HMM.

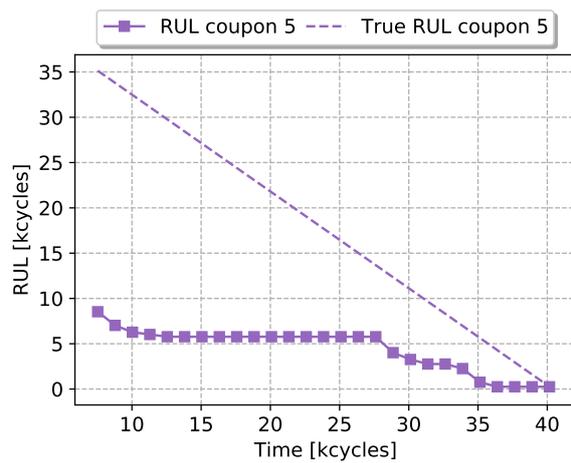


Figure B.2: RUL prediction of coupon 5 using the HMM.

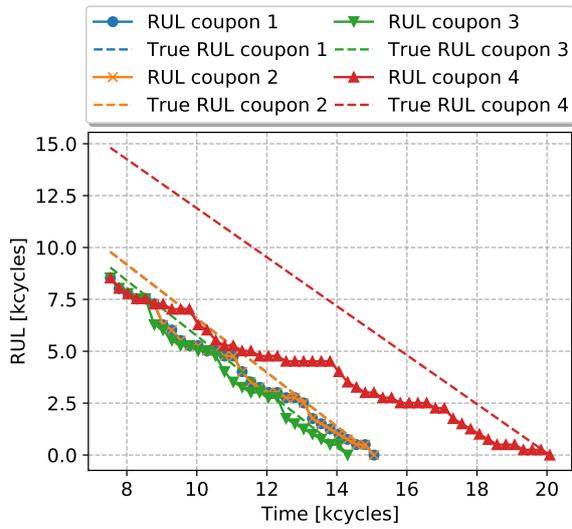


Figure B.3: RUL prediction of coupons 1 until 4 using the HMM trained on the normalised data.

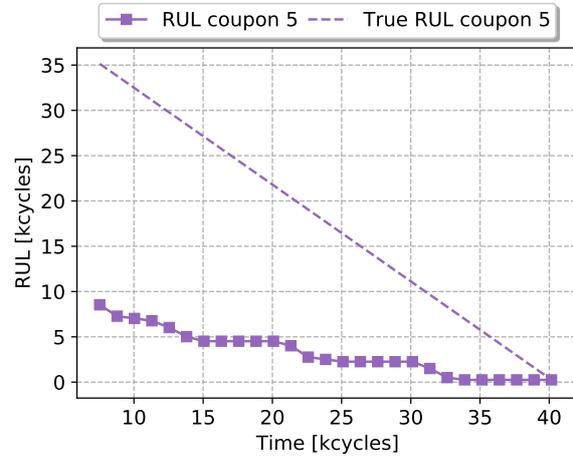


Figure B.4: RUL prediction of coupon 5 using the HMM trained on the normalised data.

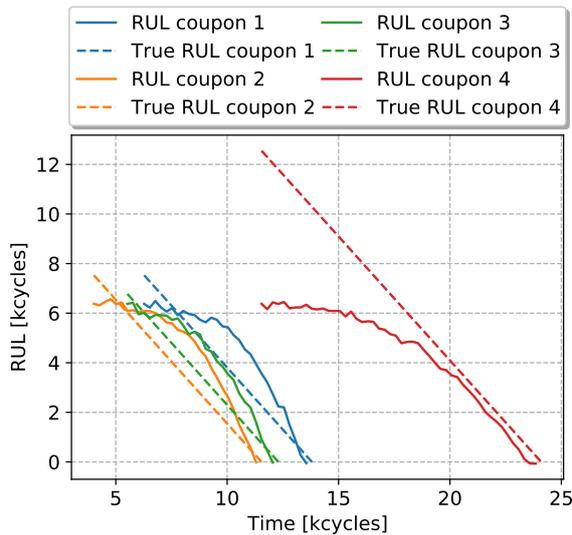


Figure B.5: RUL prediction of coupons 1 until 4 using the SVR.

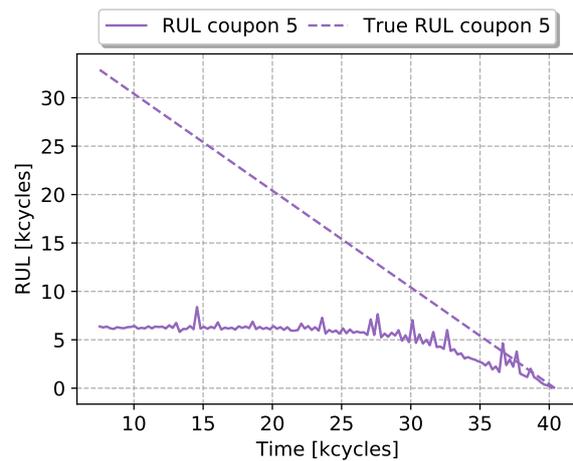


Figure B.6: RUL prediction of coupon 5 using the SVR.

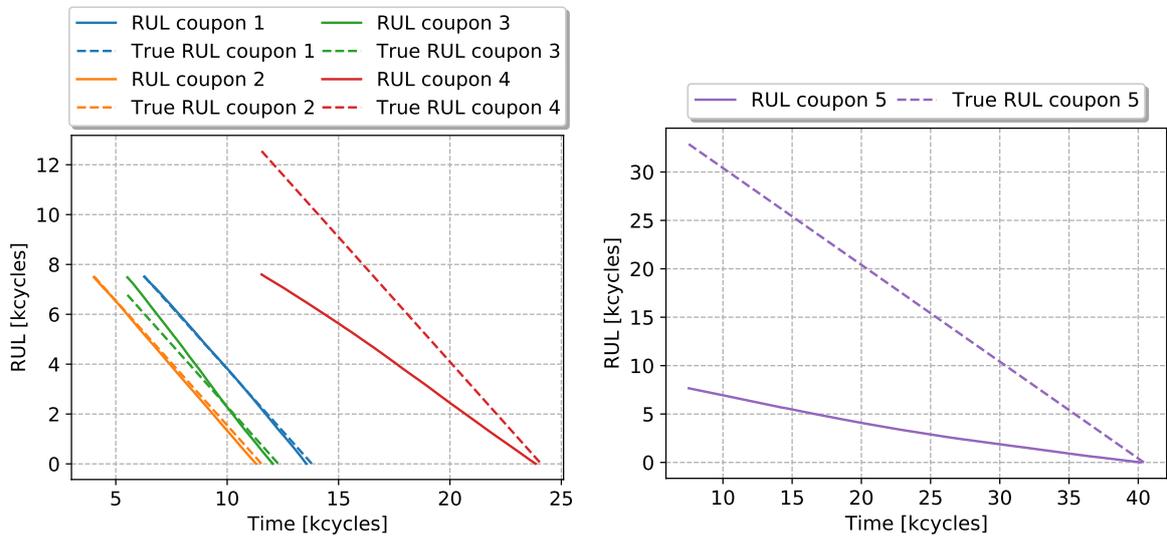
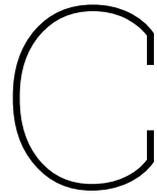


Figure B.7: RUL prediction of coupons 1 until 4 using the SVR trained on the normalised data. **Figure B.8:** RUL prediction of coupon 5 using the SVR trained on the normalised data.



Predictions of the HMM Base Predictor of the Specimen

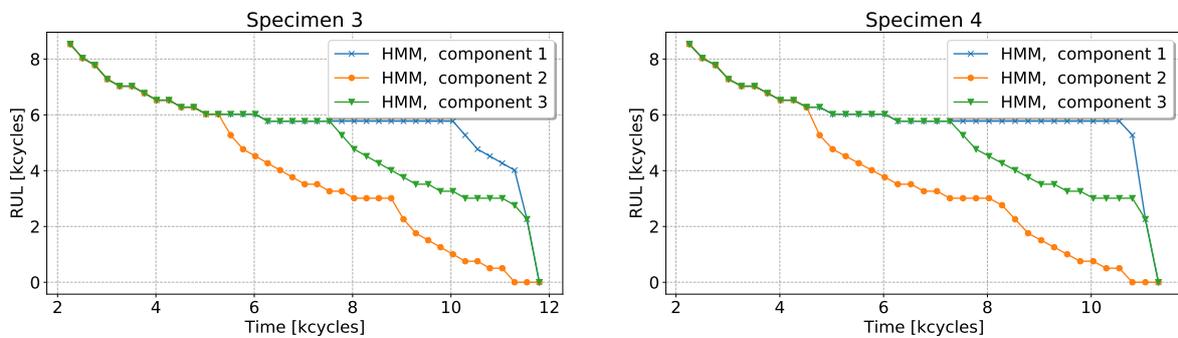


Figure C.1: RUL predictions of the HMM base predictor using the averaged strain data of specimens three and four.

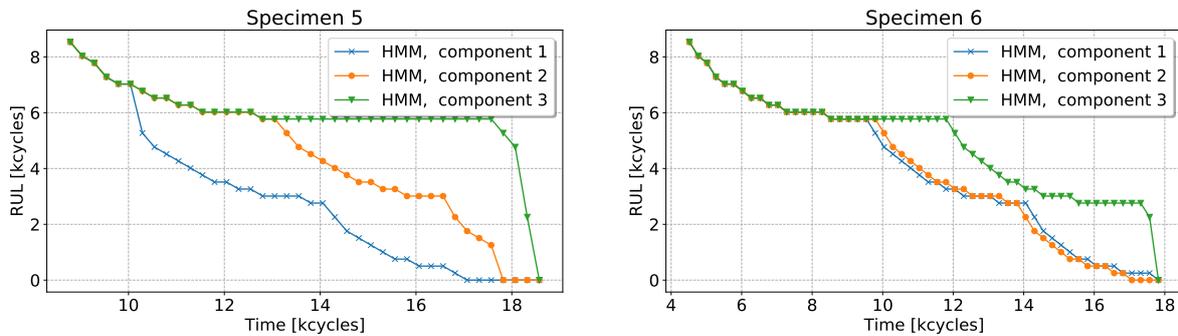


Figure C.2: RUL predictions of the HMM base predictor using the averaged strain data of specimens five and six.

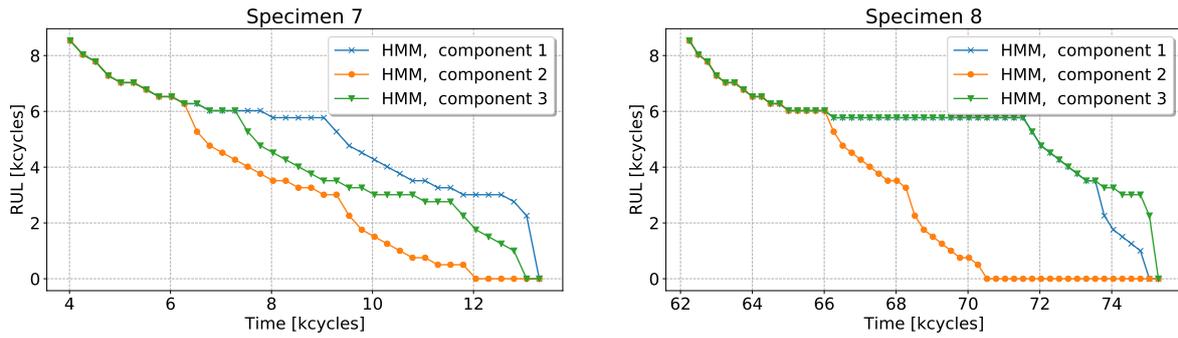


Figure C.3: RUL predictions of the HMM base predictor using the averaged strain data of specimens seven and eight.

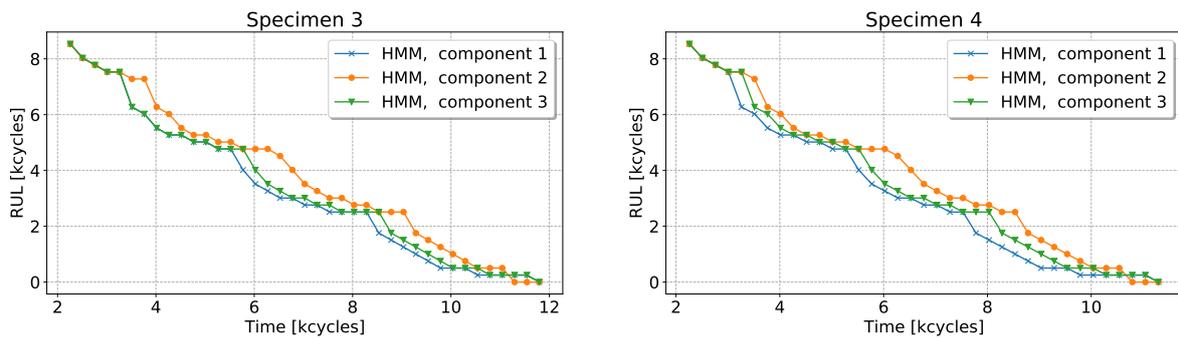


Figure C.4: RUL predictions of the HMM base predictor using the normalised data of specimens three and four.

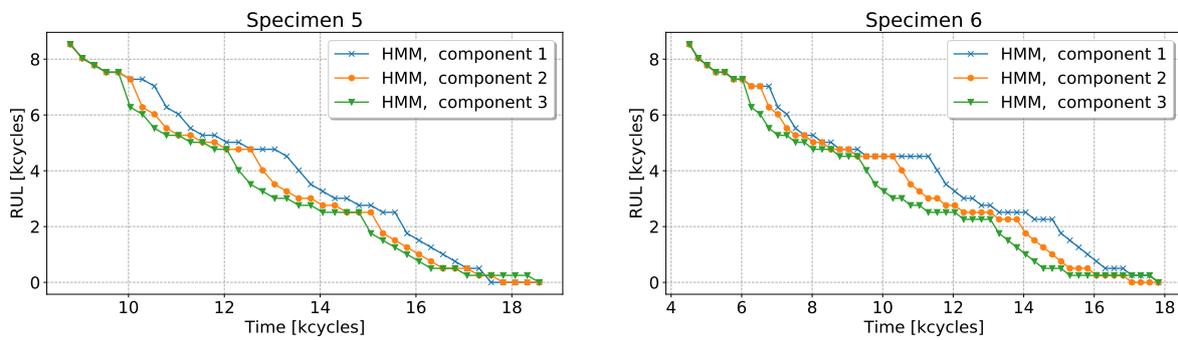


Figure C.5: RUL predictions of the HMM base predictor using the normalised data of specimens five and six.

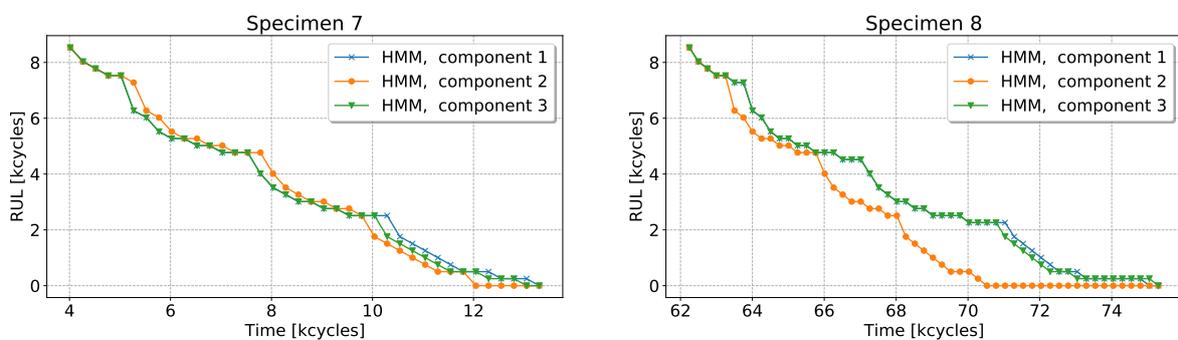


Figure C.6: RUL predictions of the HMM base predictor using the normalised data of specimens seven and eight.

D

Predictions of the SVR Base Predictor of the Specimen

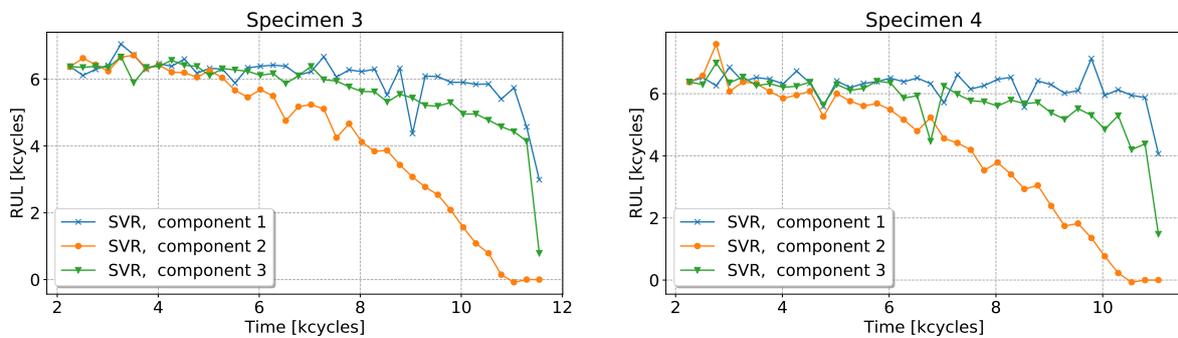


Figure D.1: RUL predictions of the SVR base predictor using the averaged strain data of specimens three and four.

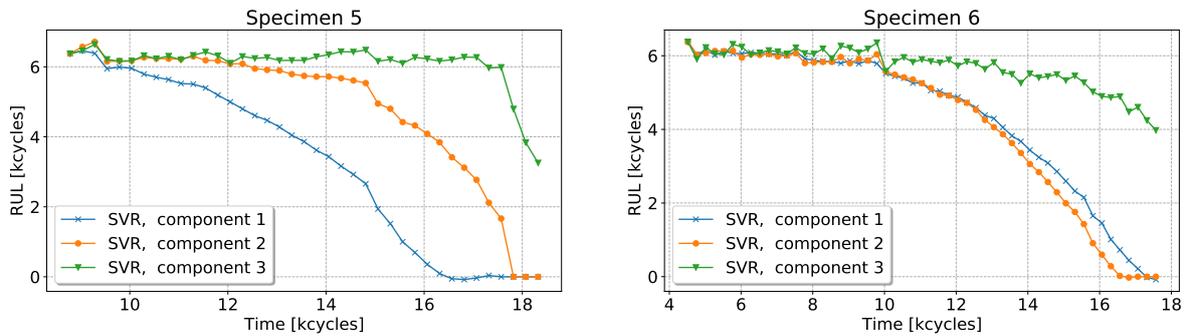


Figure D.2: RUL predictions of the SVR base predictor using the averaged strain data of specimens five and six.

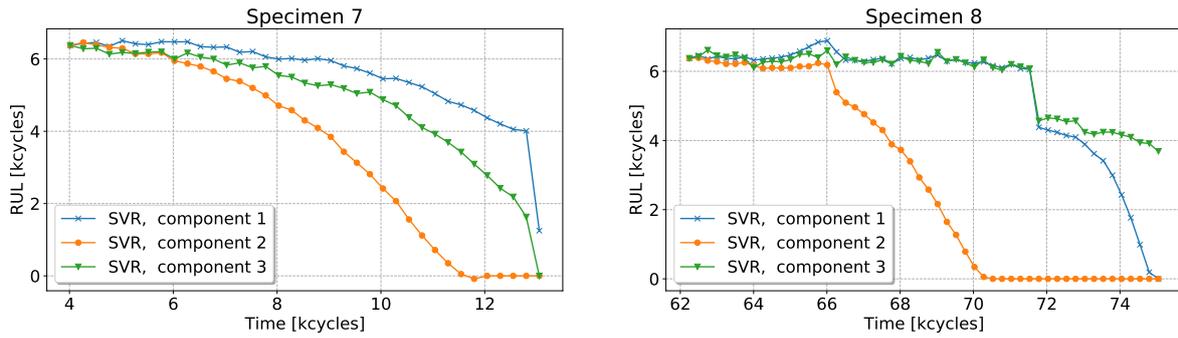


Figure D.3: RUL predictions of the SVR base predictor using the averaged strain data of specimens seven and eight.

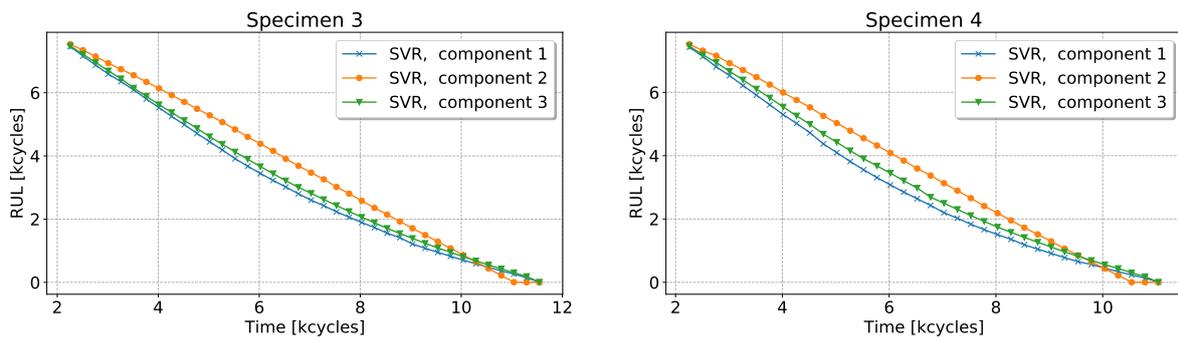


Figure D.4: RUL predictions of the SVR base predictor using the normalised data of specimens three and four.

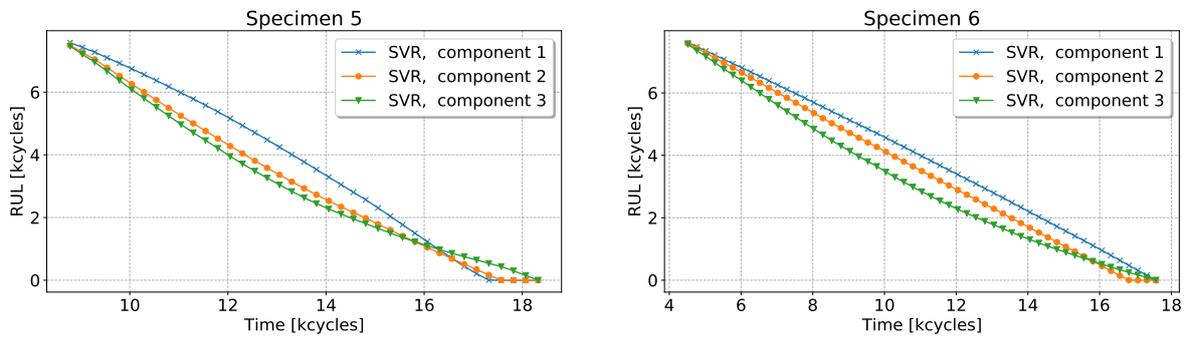


Figure D.5: RUL predictions of the SVR base predictor using the normalised data of specimens five and six.

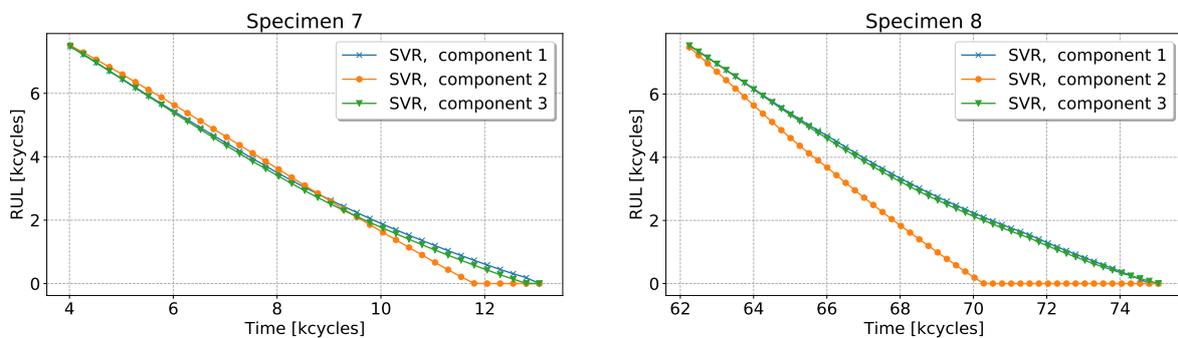


Figure D.6: RUL predictions of the SVR base predictor using the normalised data of specimens seven and eight.

E

Prediction Results

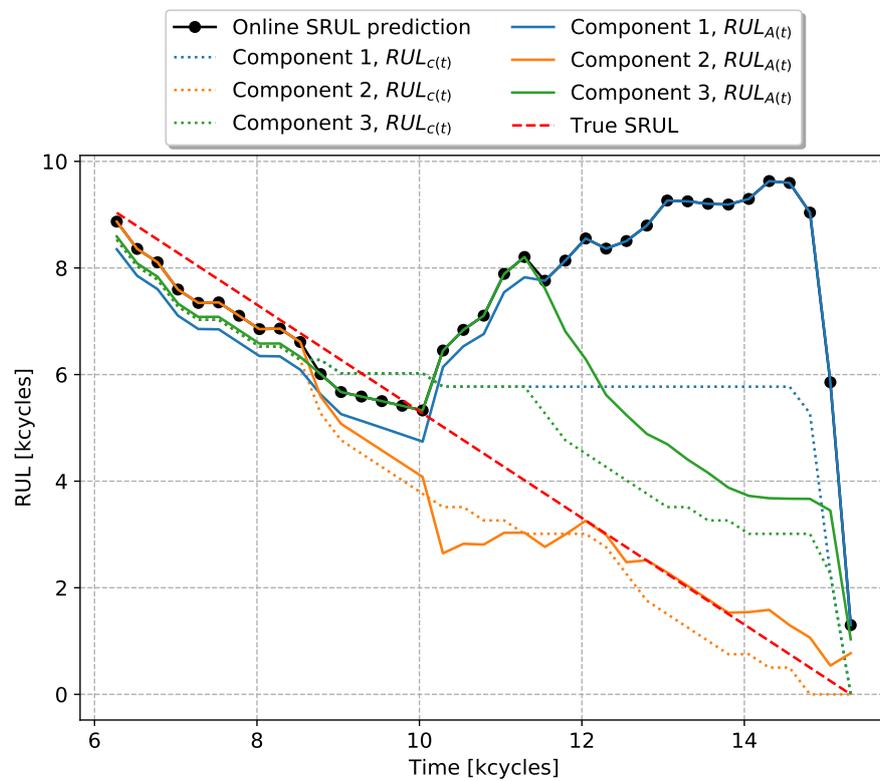


Figure E.1: Results of the IIM trained online on specimen four. Using averaged strain data and RULs provided by the HMM.

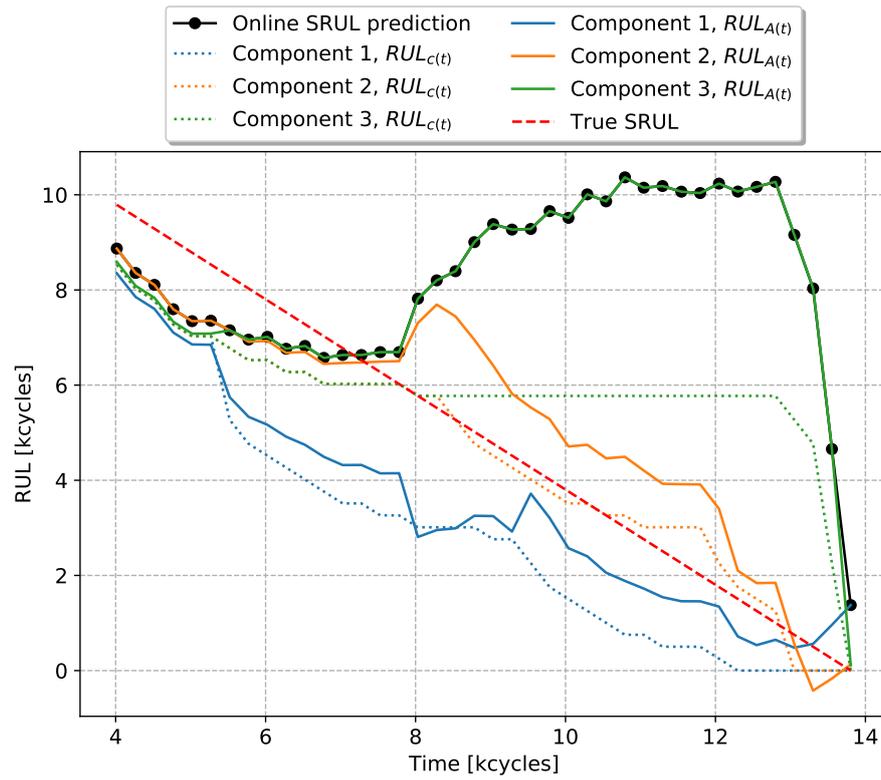


Figure E.2: Results of the IIM trained online on specimen five. Using averaged strain data and RULs provided by the HMM.

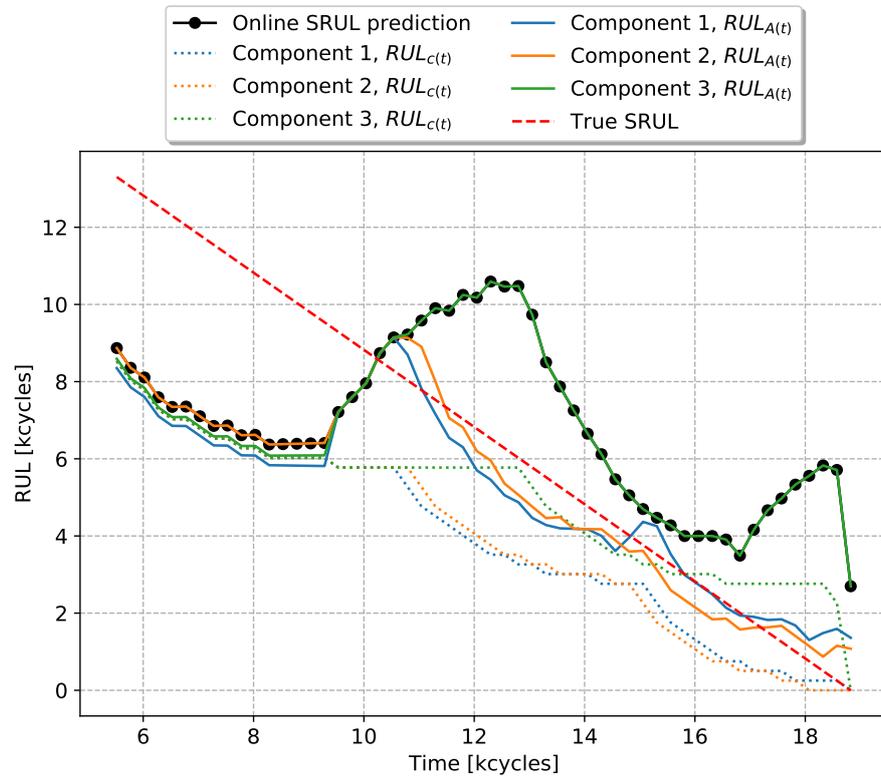


Figure E.3: Results of the IIM trained online on specimen six. Using averaged strain data and RULs provided by the HMM.

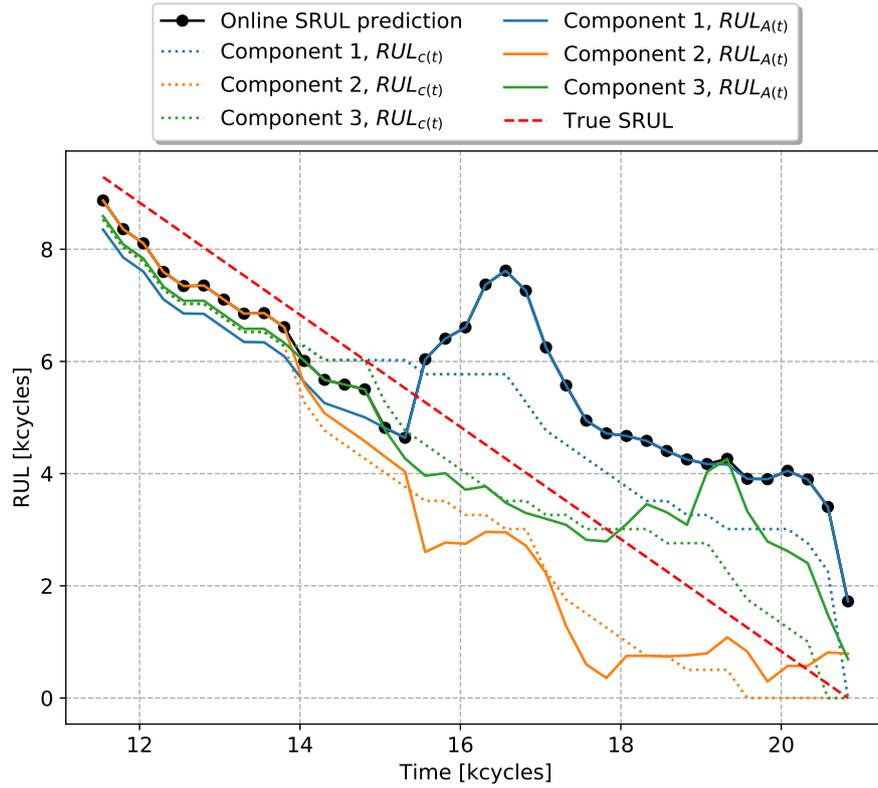


Figure E.4: Results of the IIM trained online on specimen seven. Using averaged strain data and RULs provided by the HMM.

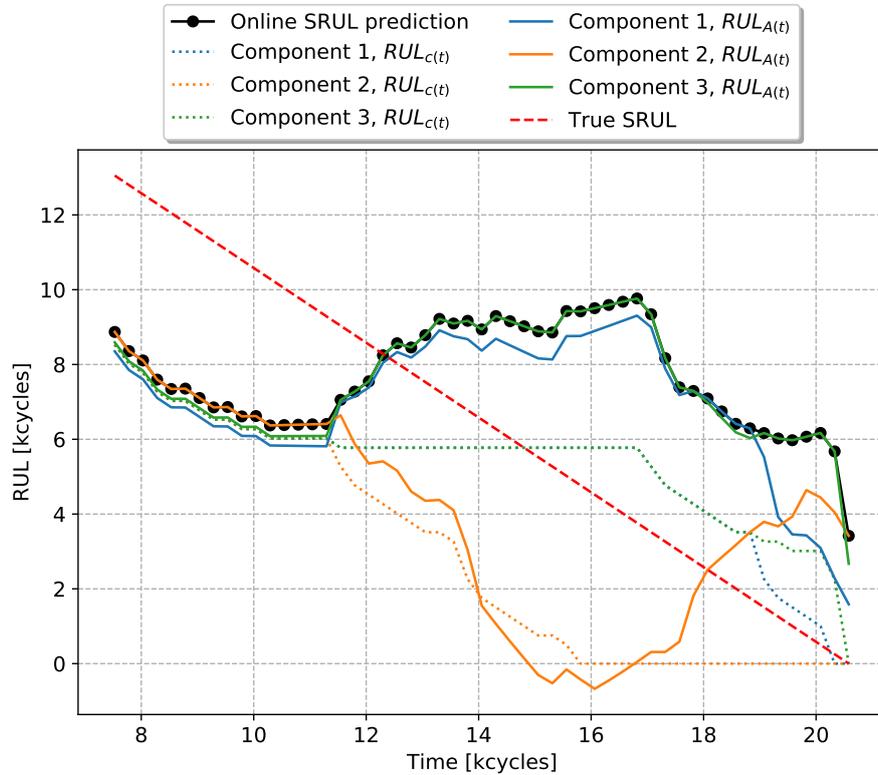


Figure E.5: Results of the IIM trained online on specimen eight. Using averaged strain data and RULs provided by the HMM.

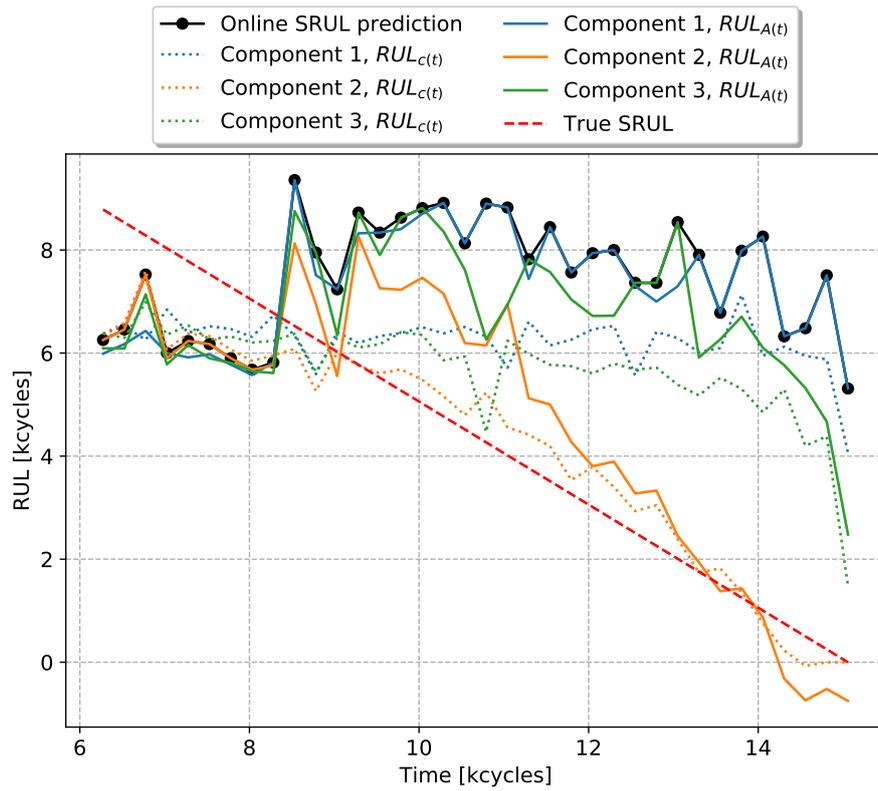


Figure E.6: Results of the IIM trained online on specimen four. Using averaged strain data and RULs provided by the SVR.

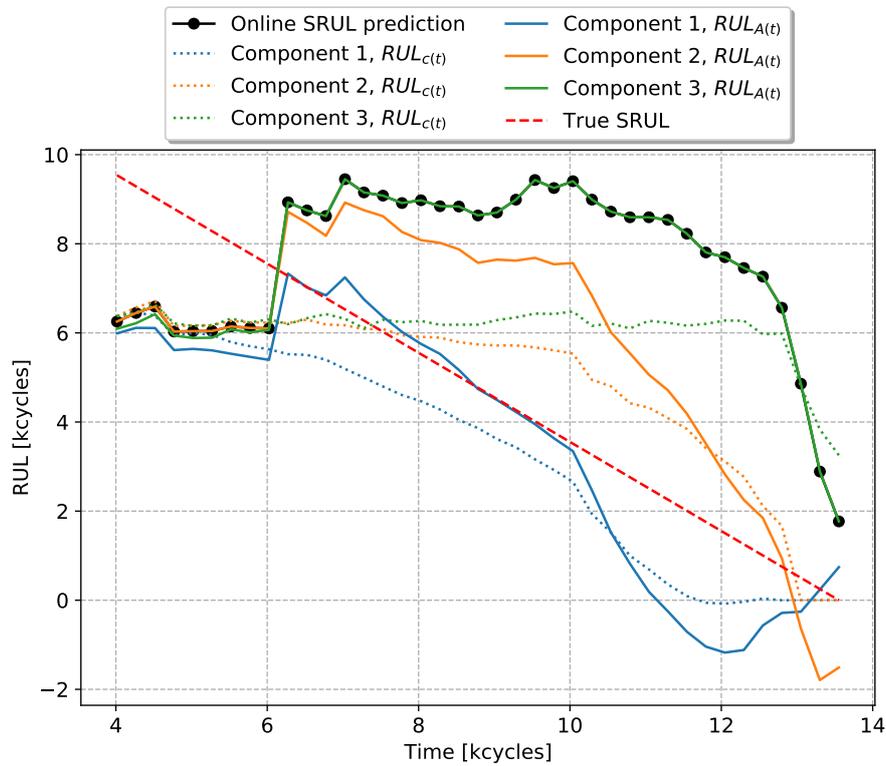


Figure E.7: Results of the IIM trained online on specimen five. Using averaged strain data and RULs provided by the SVR.

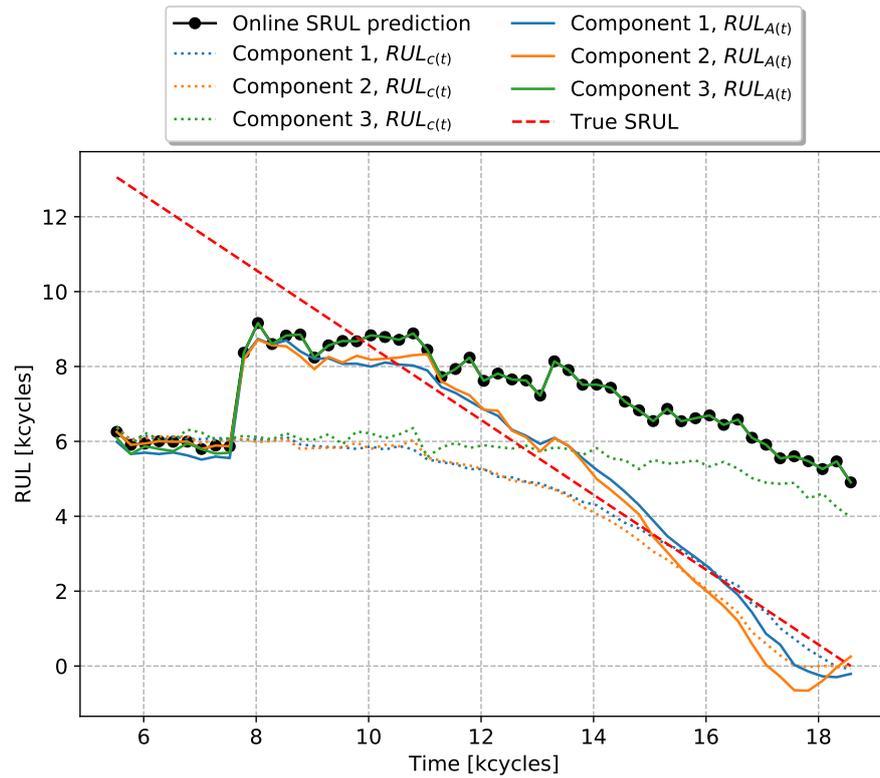


Figure E.8: Results of the IIM trained online on specimen six. Using averaged strain data and RULs provided by the SVR.

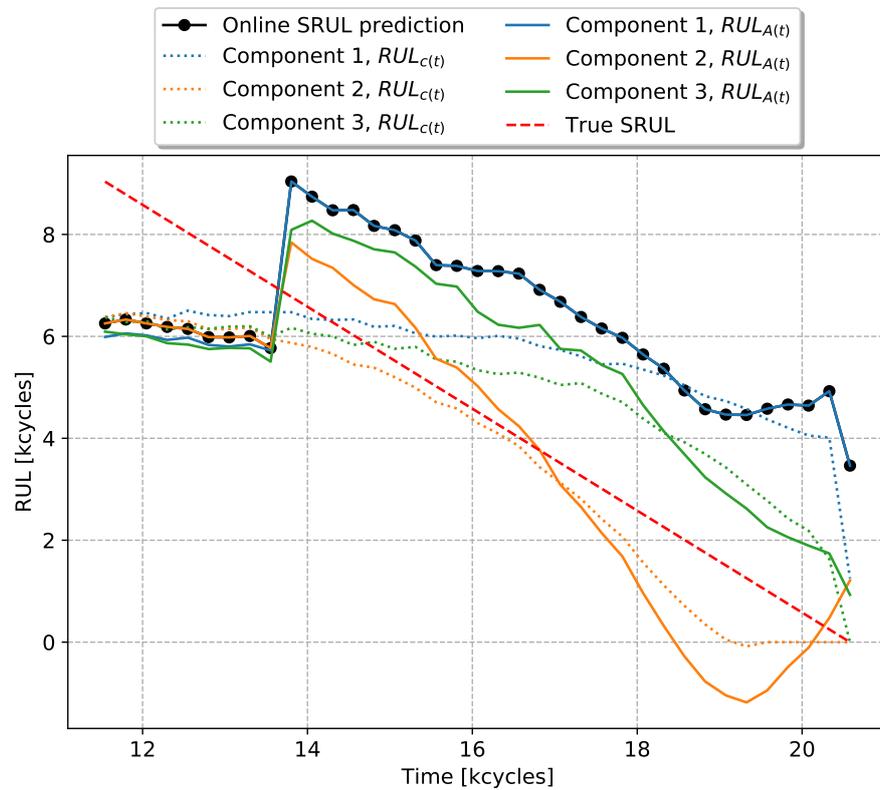


Figure E.9: Results of the IIM trained online on specimen seven. Using averaged strain data and RULs provided by the SVR.

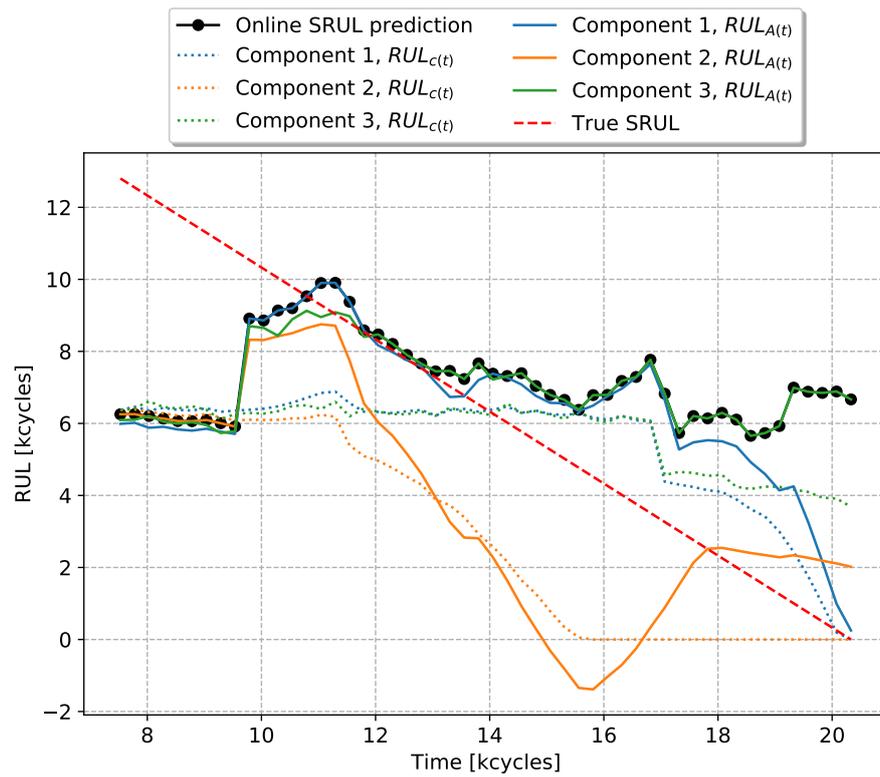


Figure E.10: Results of the IIM trained online on specimen eight. Using averaged strain data and RULs provided by the SVR.

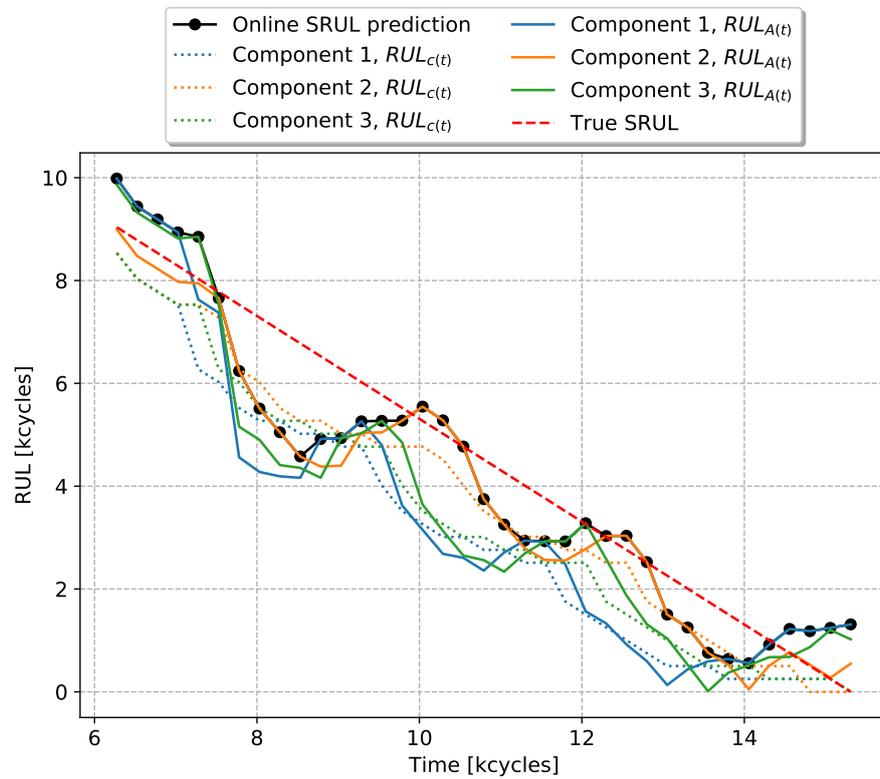


Figure E.11: Results of the IIM trained online on specimen four. Using normalised data and RULs provided by the HMM.

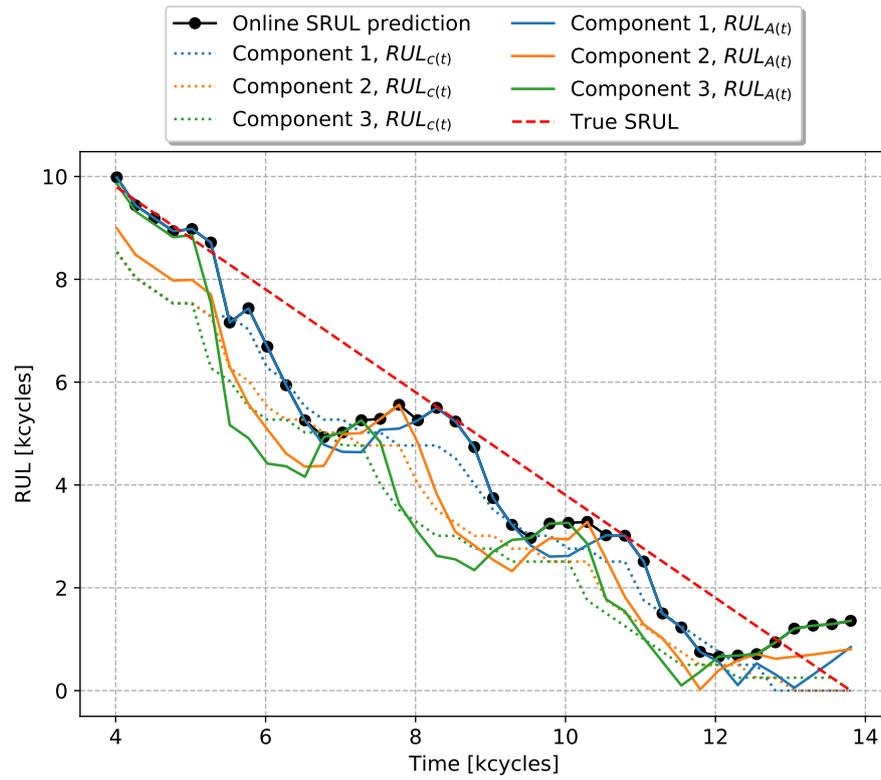


Figure E.12: Results of the IIM trained online on specimen five. Using normalised data and RULs provided by the HMM.

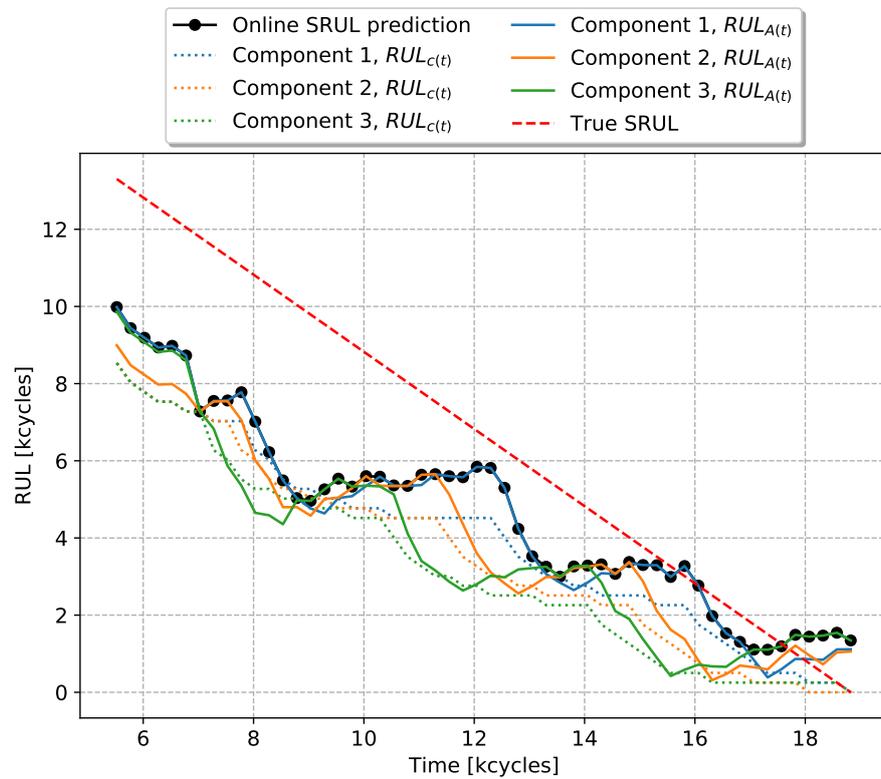


Figure E.13: Results of the IIM trained online on specimen six. Using normalised data and RULs provided by the HMM.

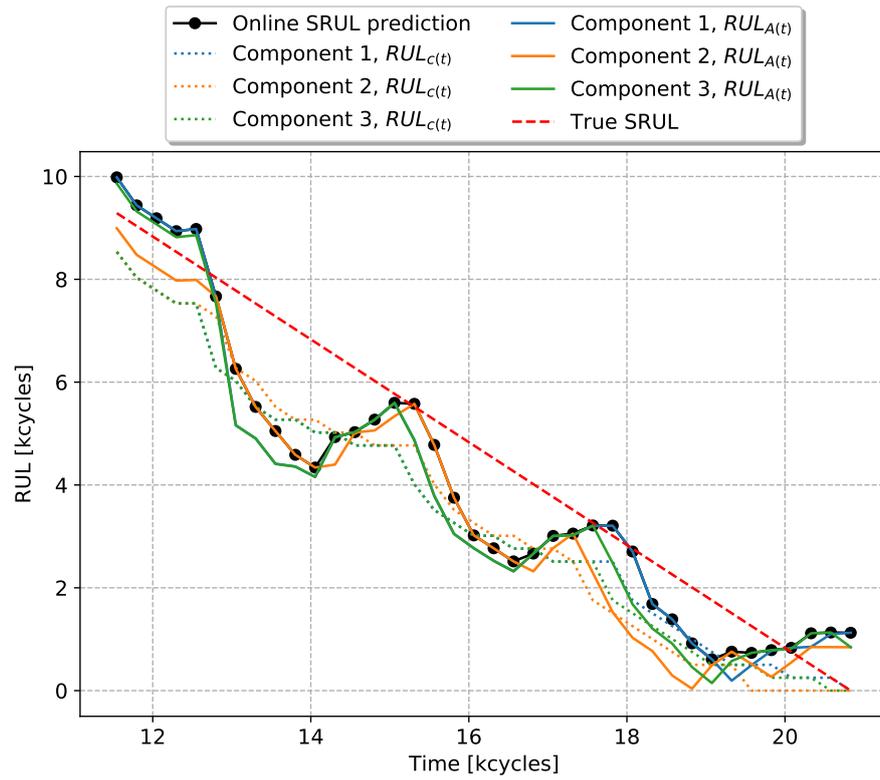


Figure E.14: Results of the IIM trained online on specimen seven. Using normalised data and RULs provided by the HMM.

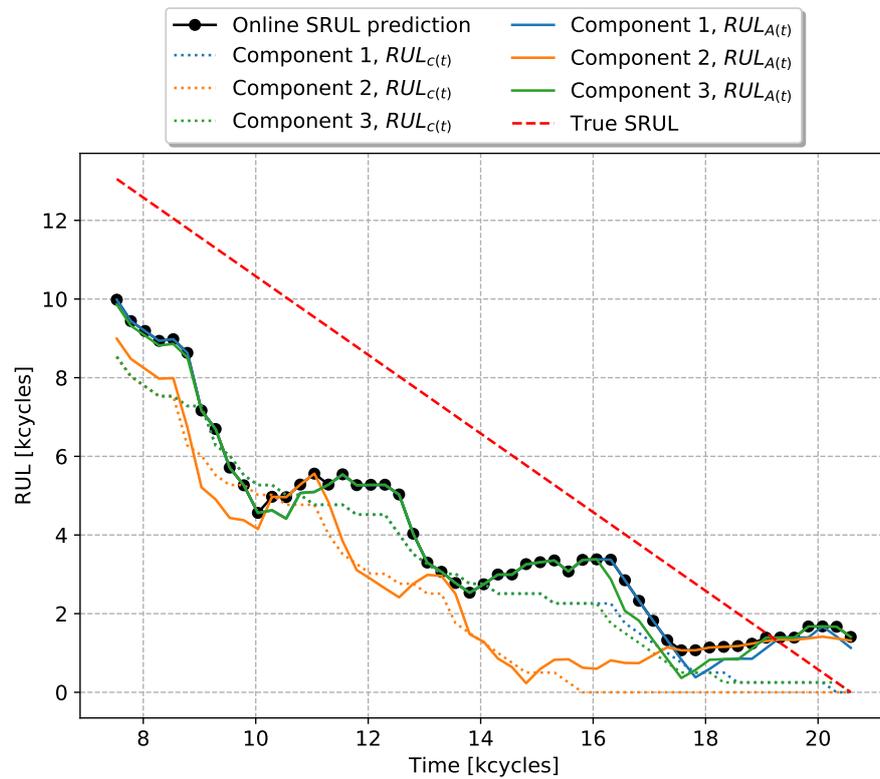


Figure E.15: Results of the IIM trained online on specimen eight. Using normalised data and RULs provided by the HMM.

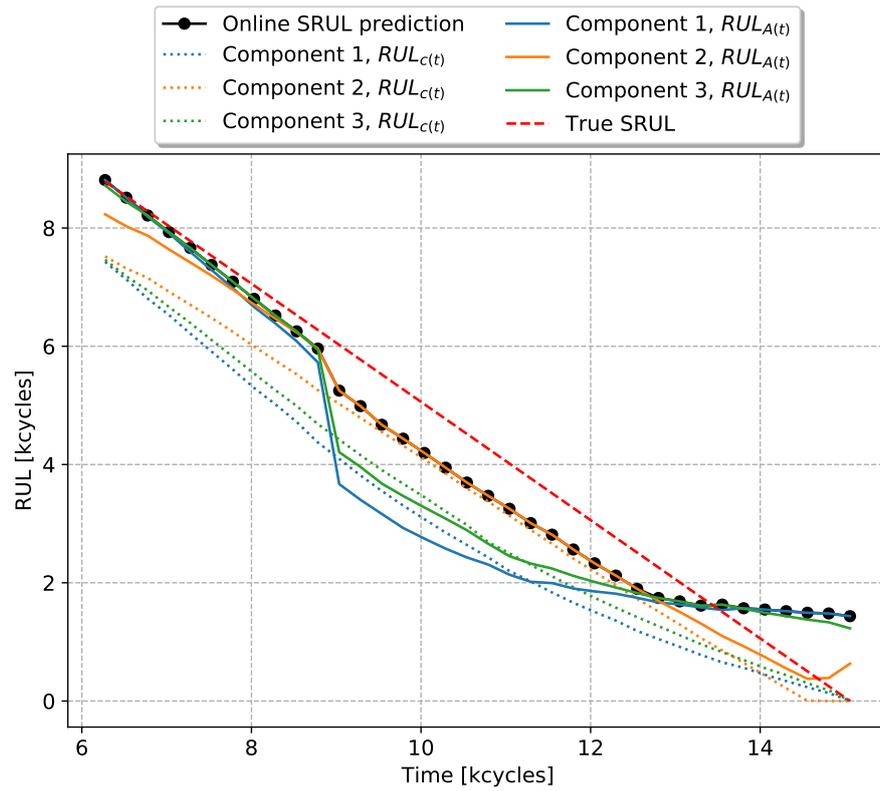


Figure E.16: Results of the IIM trained online on specimen four. Using normalised data and RULs provided by the SVR.

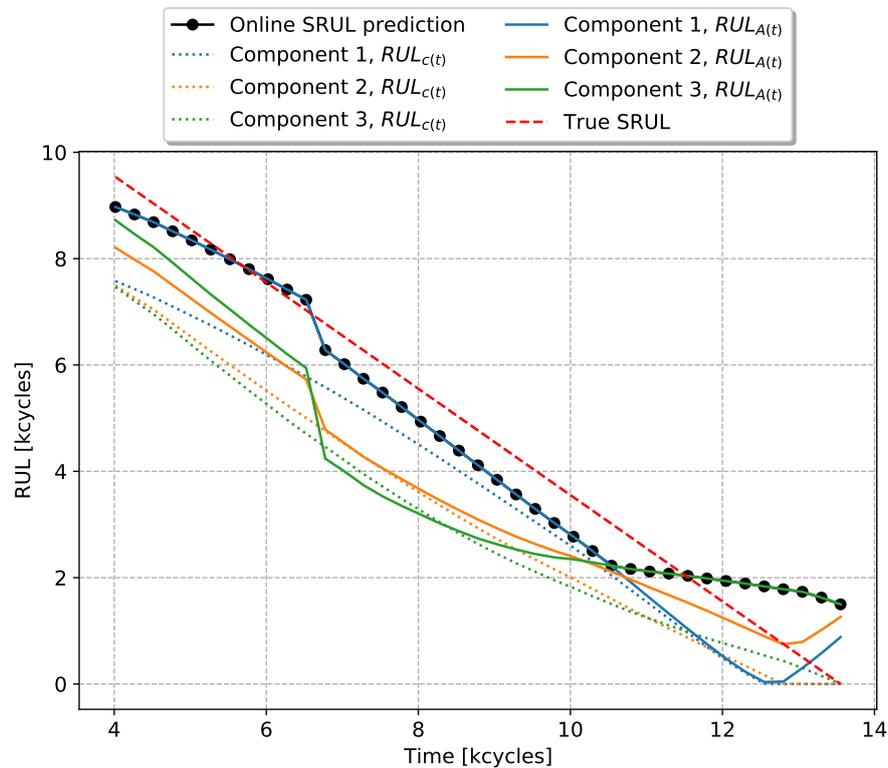


Figure E.17: Results of the IIM trained online on specimen five. Using normalised data and RULs provided by the SVR.

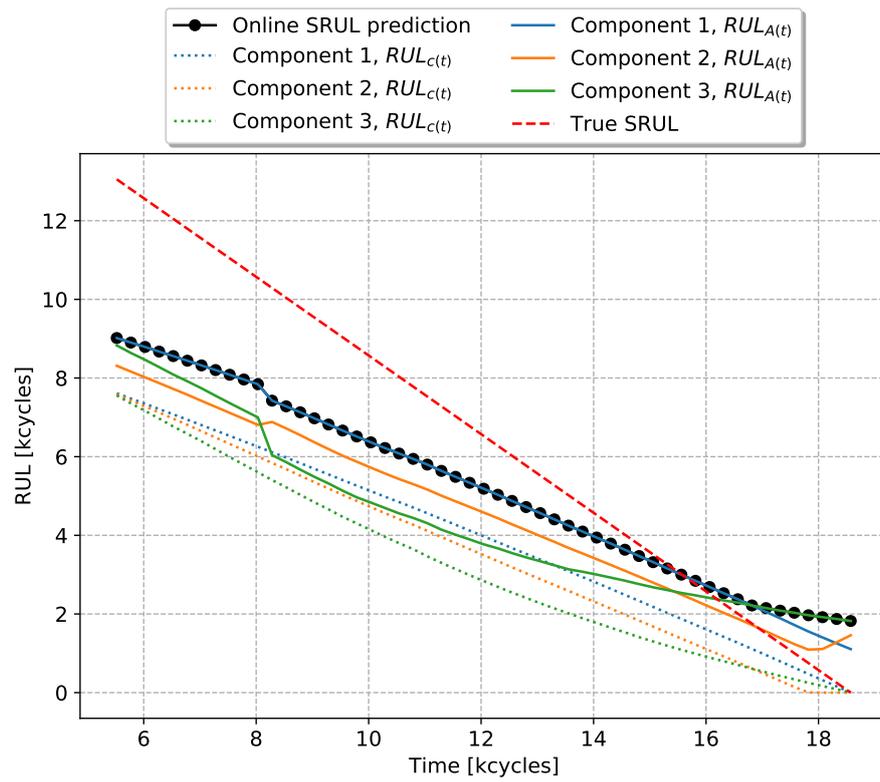


Figure E.18: Results of the IIM trained online on specimen six. Using normalised data and RULs provided by the SVR.

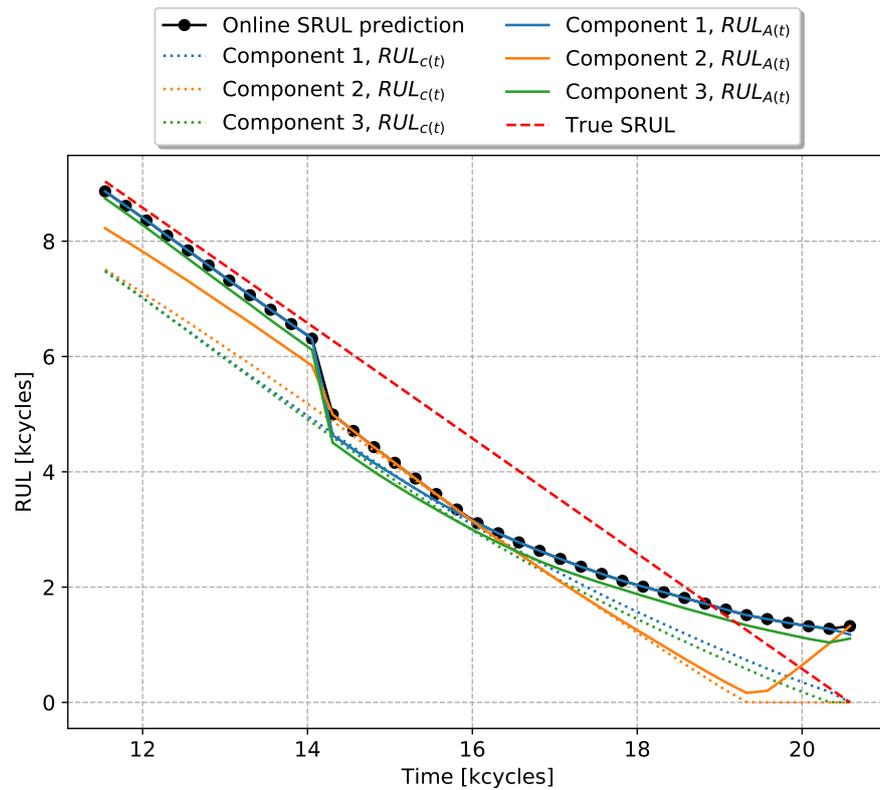


Figure E.19: Results of the IIM trained online on specimen seven. Using normalised data and RULs provided by the SVR.

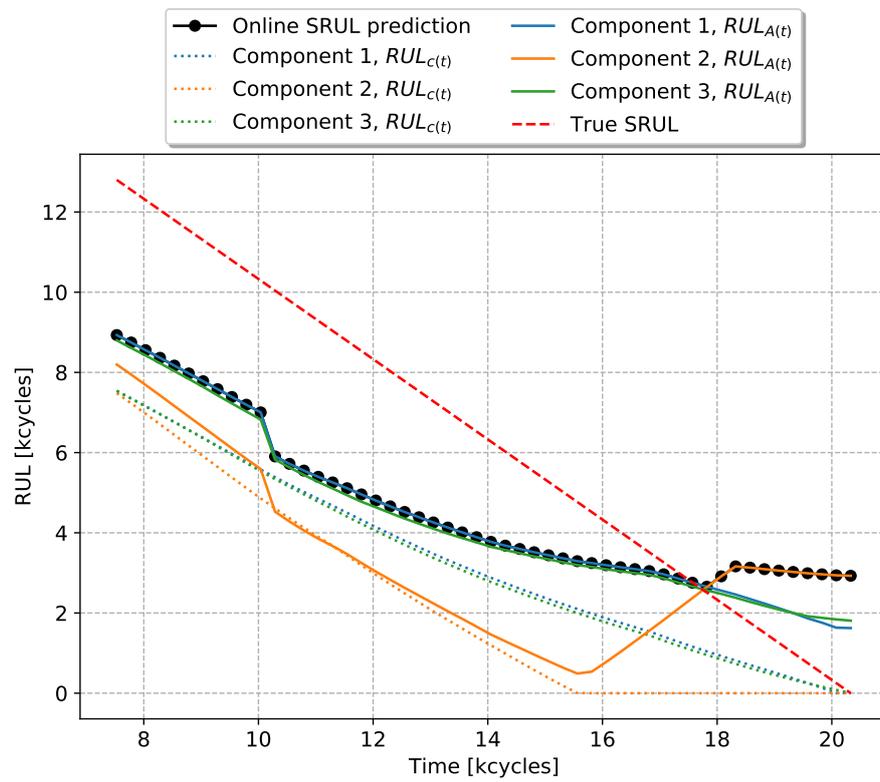


Figure E.20: Results of the IIM trained online on specimen eight. Using normalised data and RULs provided by the SVR.

F

Error of the IIM VS Base Predictors

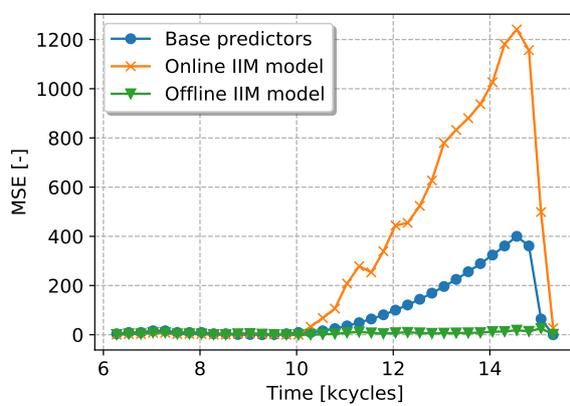


Figure F.1: Error of the HMM base predictor and the IIM of specimen four using the averaged strain data.

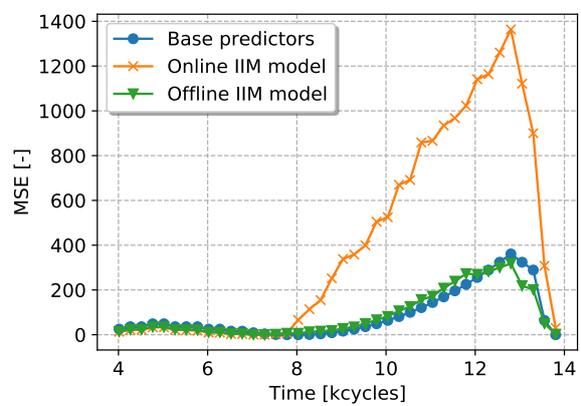


Figure F.2: Error of the HMM base predictor and the IIM of specimen five using the averaged strain data.

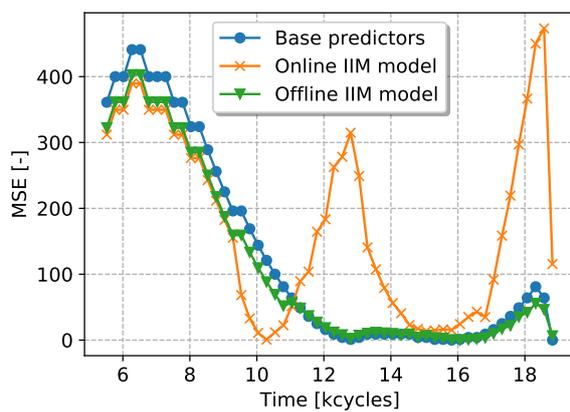


Figure F.3: Error of the HMM base predictor and the IIM of specimen six using the averaged strain data.

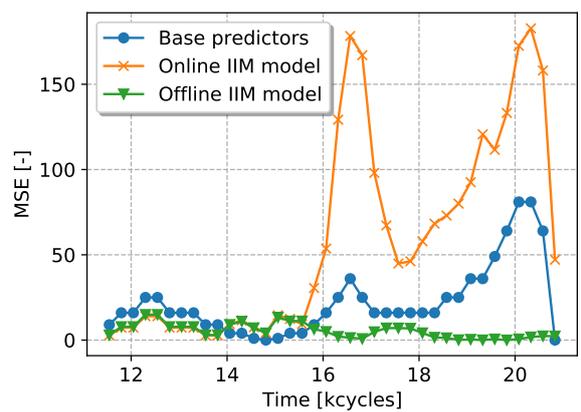


Figure F.4: Error of the HMM base predictor and the IIM of specimen seven using the averaged strain data.

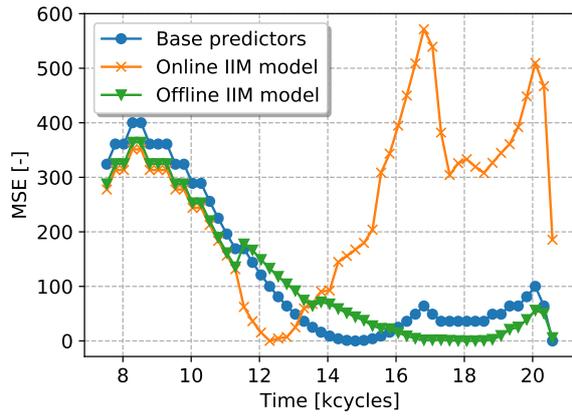


Figure F.5: Error of the HMM base predictor and the IIM of specimen eight using the averaged strain data.

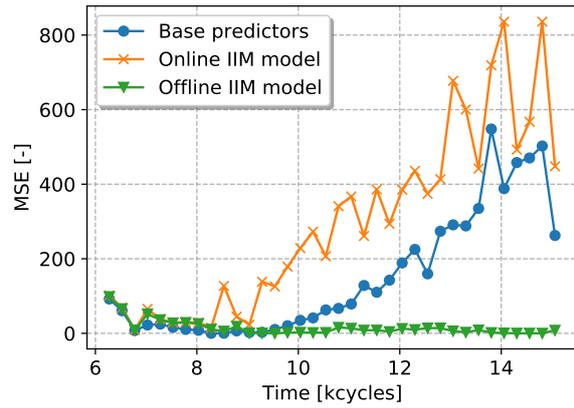


Figure F.6: Error of the SVR base predictor and the IIM of specimen four using the averaged strain data.

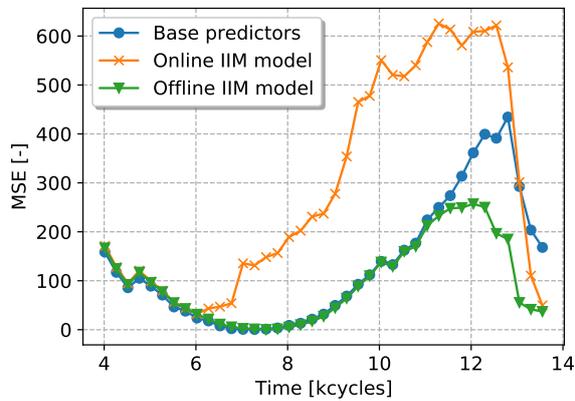


Figure F.7: Error of the SVR base predictor and the IIM of specimen five using the averaged strain data.

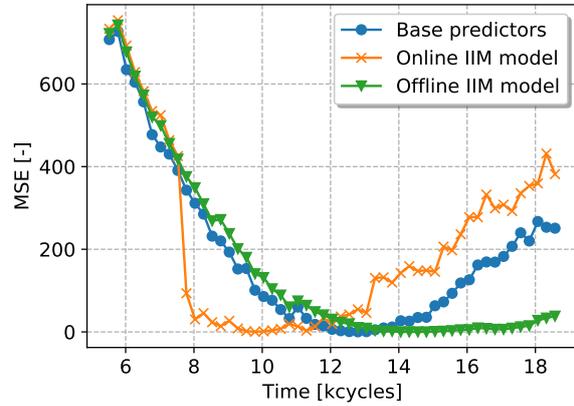


Figure F.8: Error of the SVR base predictor and the IIM of specimen six using the averaged strain data.

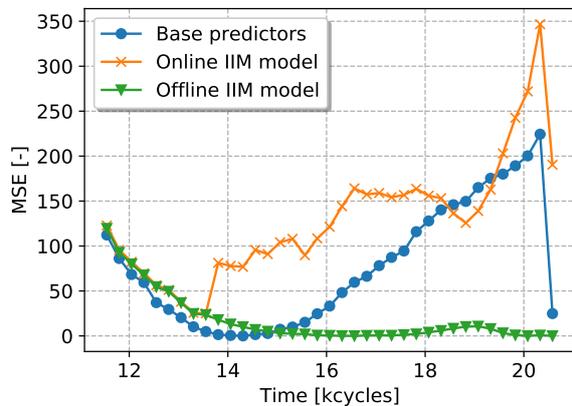


Figure F.9: Error of the SVR base predictor and the IIM of specimen seven using the averaged strain data.

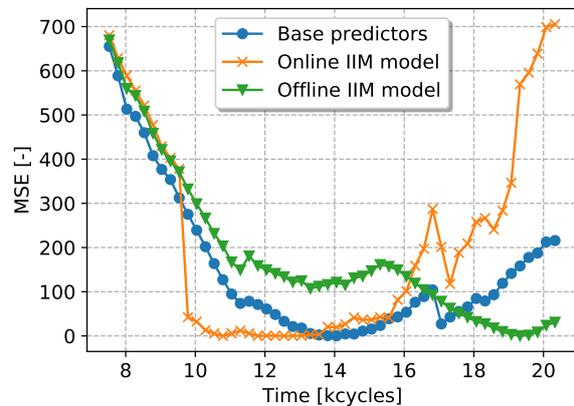


Figure F.10: Error of the SVR base predictor and the IIM of specimen eight using the averaged strain data.

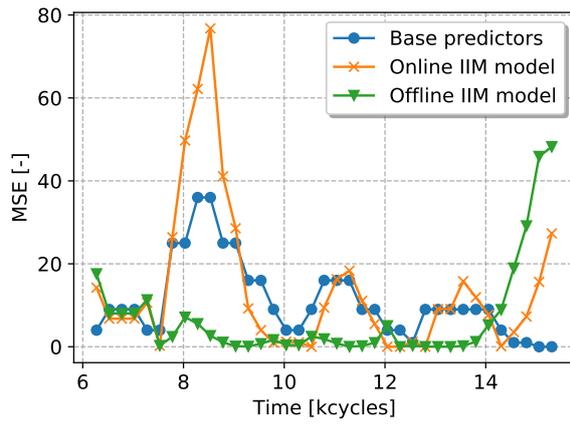


Figure F.11: Error of the HMM base predictor and the IIM of specimen four using the normalised data.

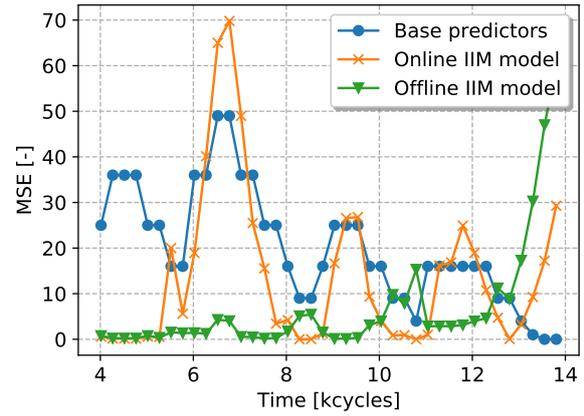


Figure F.12: Error of the HMM base predictor and the IIM of specimen five using the normalised data.

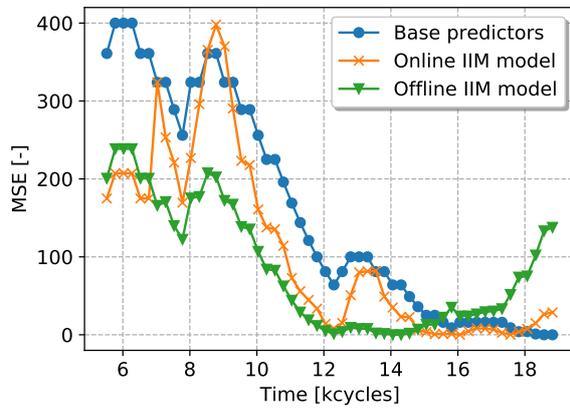


Figure F.13: Error of the HMM base predictor and the IIM of specimen six using the normalised data.

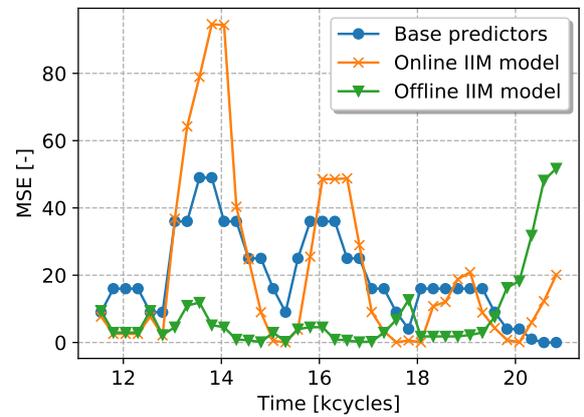


Figure F.14: Error of the HMM base predictor and the IIM of specimen seven using the normalised data.

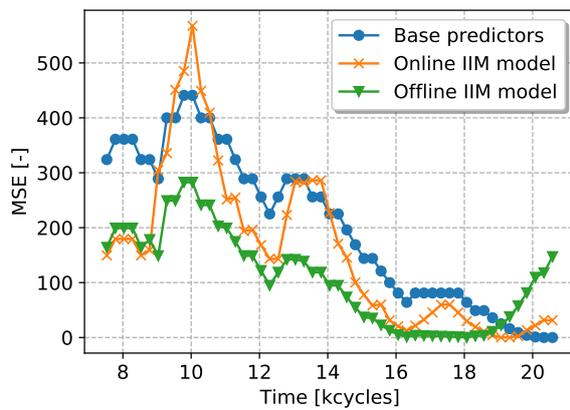


Figure F.15: Error of the HMM base predictor and the IIM of specimen eight using the normalised data.

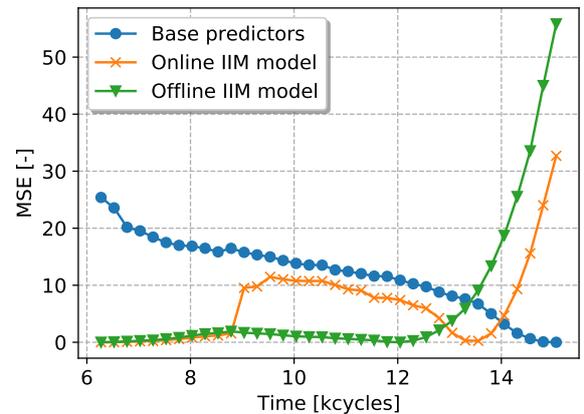


Figure F.16: Error of the SVR base predictor and the IIM of specimen four using the normalised data.

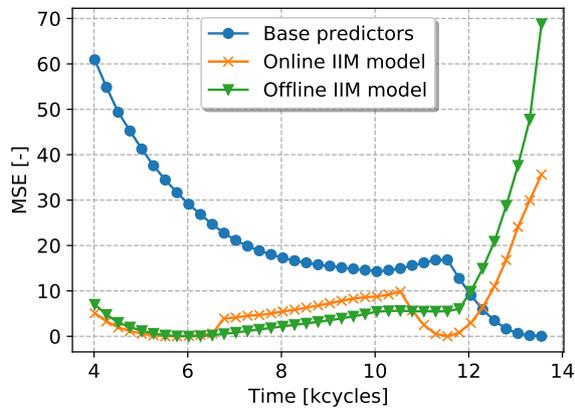


Figure F.17: Error of the SVR base predictor and the IIM of specimen five using the normalised data.

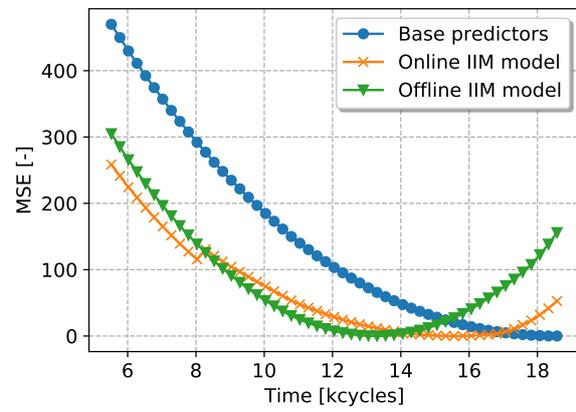


Figure F.18: Error of the SVR base predictor and the IIM of specimen six using the normalised data.

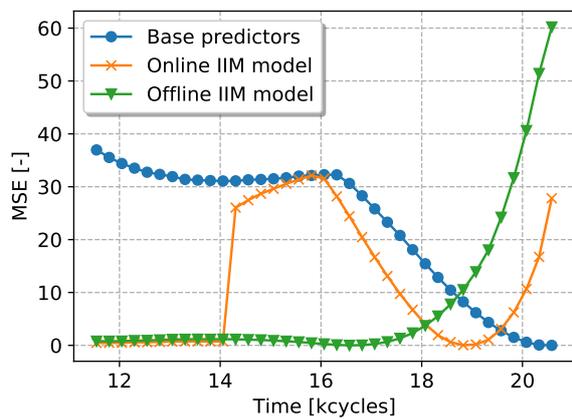


Figure F.19: Error of the SVR base predictor and the IIM of specimen seven using the normalised data.

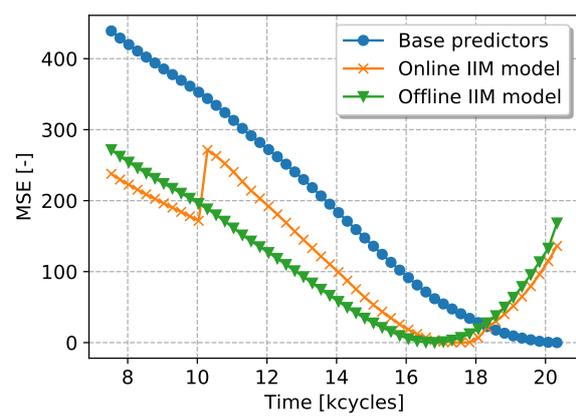


Figure F.20: Error of the SVR base predictor and the IIM of specimen eight using the normalised data.

G

Error of the IIM using both the SVR and HMM Base Predictor

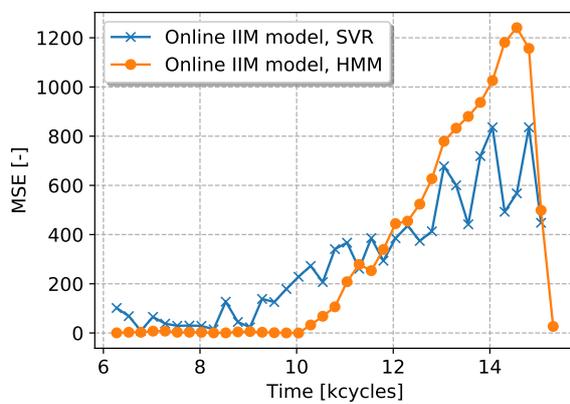


Figure G.1: Error of the IIM using both the HMM and the SVR model of specimen four using the averaged strain data.

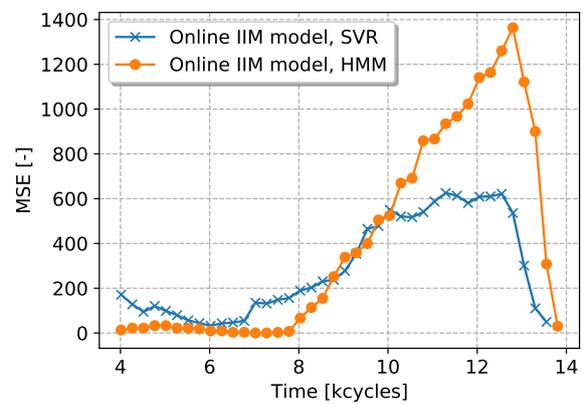


Figure G.2: Error of the IIM using both the HMM and the SVR model of specimen five using the averaged strain data.

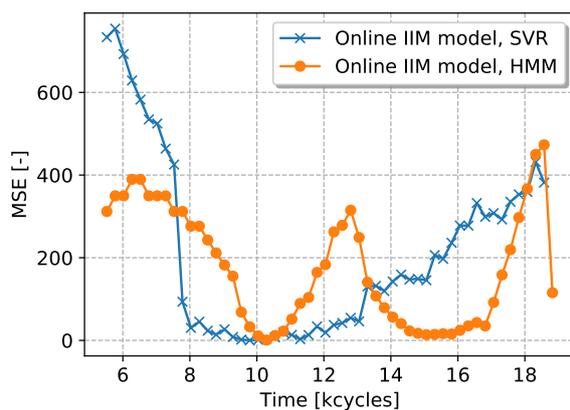


Figure G.3: Error of the IIM using both the HMM and the SVR model of specimen six using the averaged strain data.

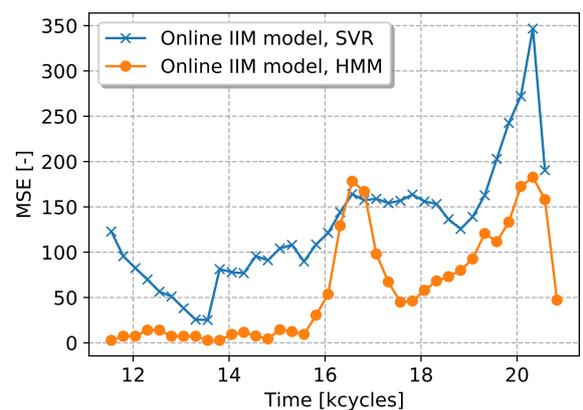


Figure G.4: Error of the IIM using both the HMM and the SVR model of specimen seven using the averaged strain data.

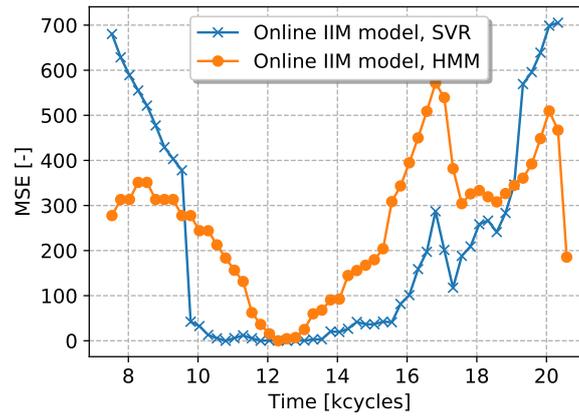


Figure G.5: Error of the IIM using both the HMM and the SVR model of specimen eight using the averaged strain data.

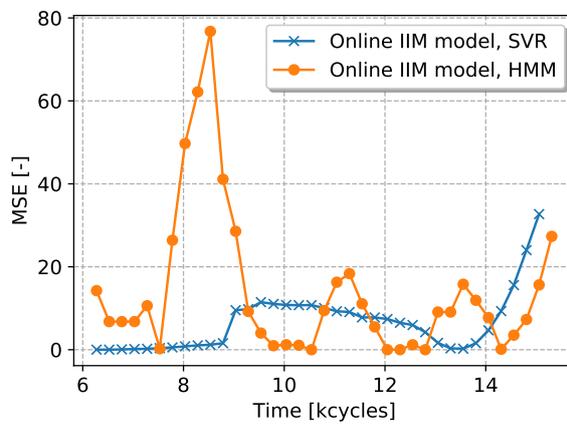


Figure G.6: Error of the IIM using both the HMM and the SVR model of specimen four using the normalised data.

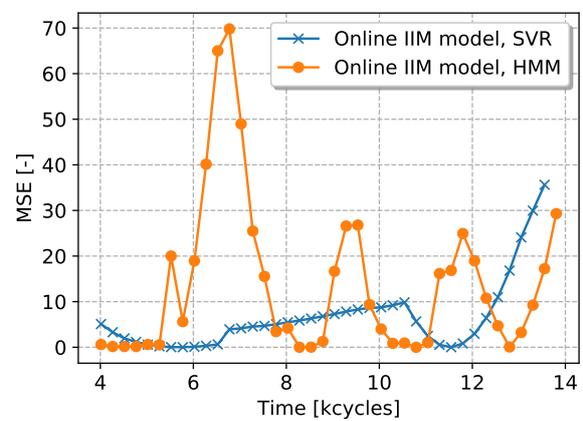


Figure G.7: Error of the IIM using both the HMM and the SVR model of specimen five using the normalised data.

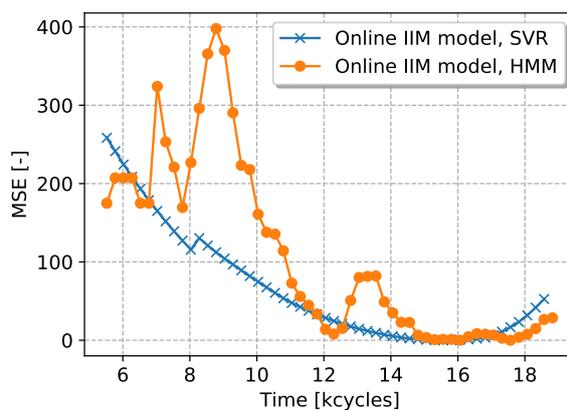


Figure G.8: Error of the IIM using both the HMM and the SVR model of specimen six using the normalised data.

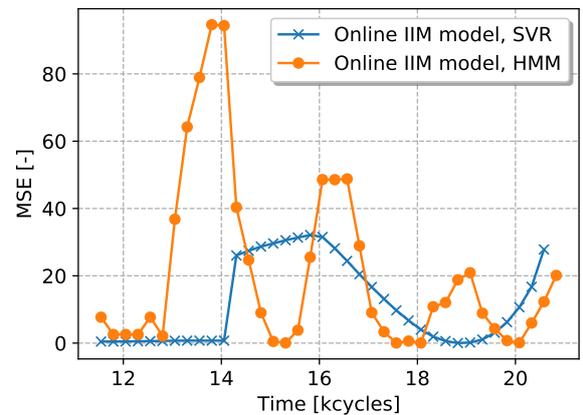


Figure G.9: Error of the IIM using both the HMM and the SVR model of specimen seven using the normalised data.

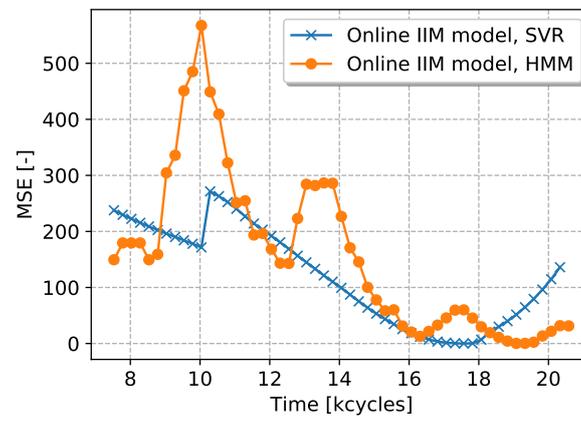
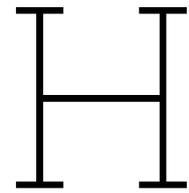


Figure G.10: Error of the IIM using both the HMM and the SVR model of specimen eight using the normalised data.



Error of the IIM Normalised VS Averaged Strain

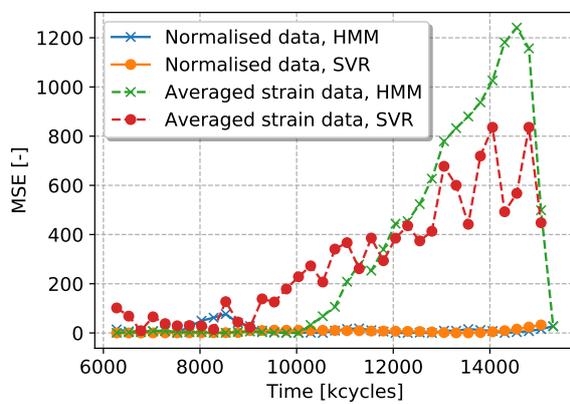


Figure H.1: Error of the IIM for both the HMM and the SVR model of specimen four using both types of data.

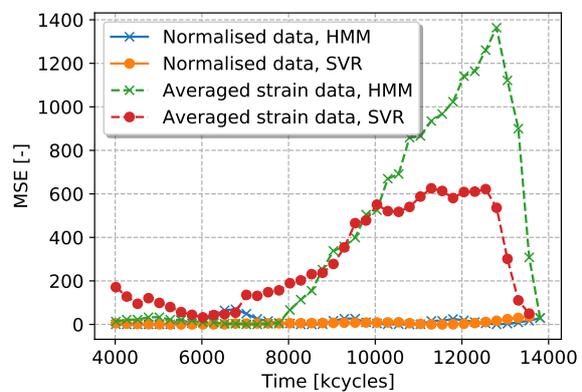


Figure H.2: Error of the IIM for both the HMM and the SVR model of specimen five using both types of data.

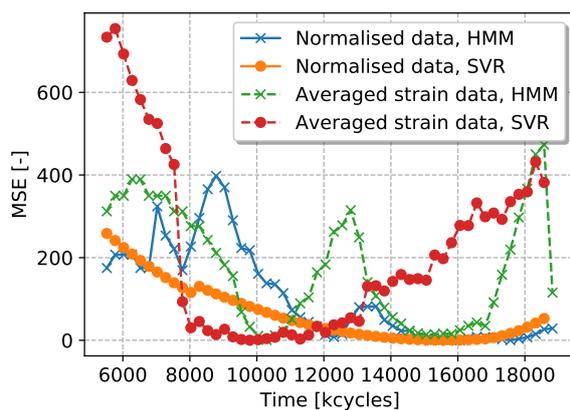


Figure H.3: Error of the IIM for both the HMM and the SVR model of specimen six using both types of data.

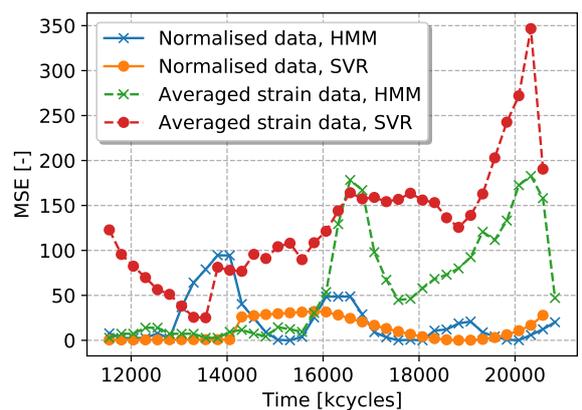


Figure H.4: Error of the IIM for both the HMM and the SVR model of specimen seven using both types of data.

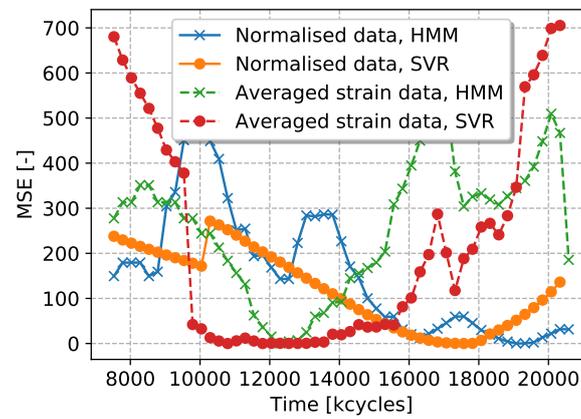


Figure H.5: Error of the IIM for both the HMM and the SVR model of specimen seven using both types of data.

Error of the IIM with Different Amount of Training Data

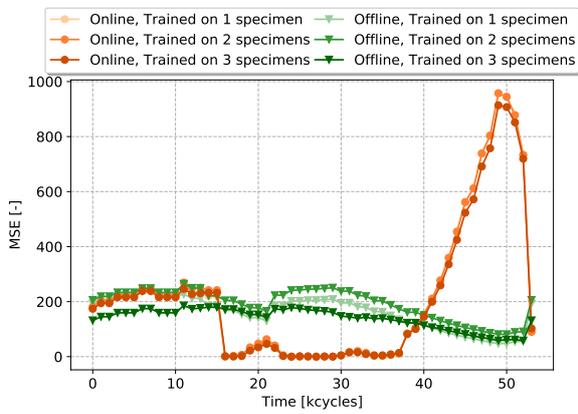


Figure I.1: The MSE of the IIM trained with different amounts of data for specimen six using the HMM model and averaged strain data.

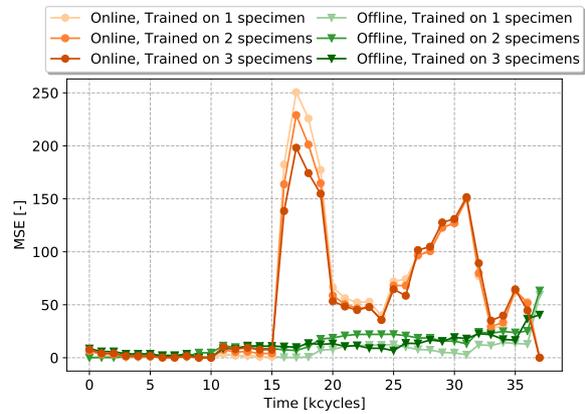


Figure I.2: The MSE of the IIM trained with different amounts of data for specimen seven using the HMM model and averaged strain data.

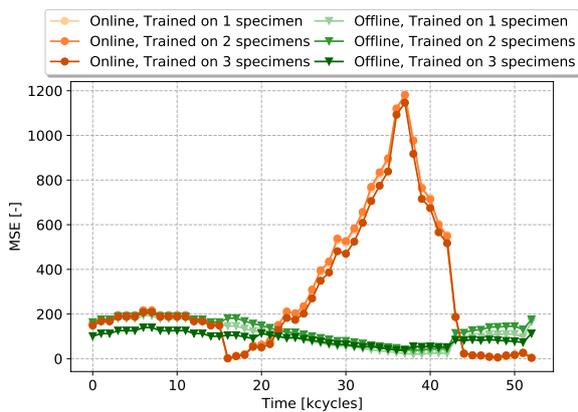


Figure I.3: The MSE of the IIM trained with different amounts of data for specimen eight using the HMM model and averaged strain data.

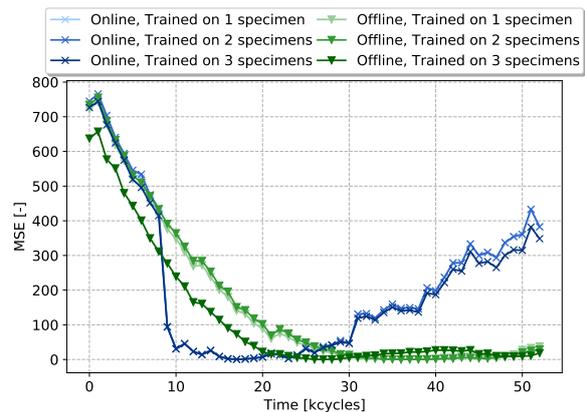


Figure I.4: The MSE of the IIM trained with different amounts of data for specimen six using the SVR model and averaged strain data.

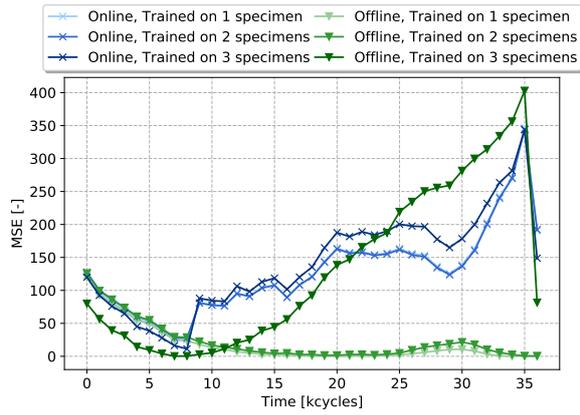


Figure I.5: The MSE of the IIM trained with different amounts of data for specimen seven using the SVR model and averaged strain data.

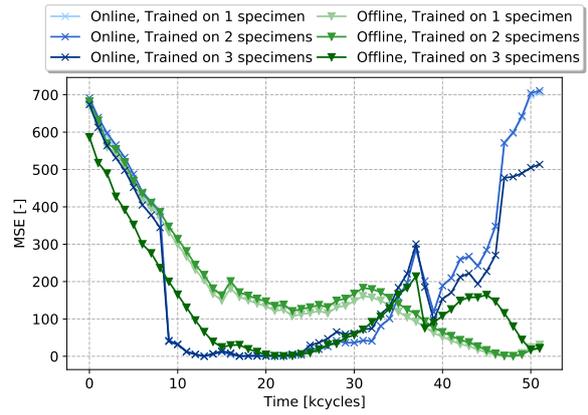


Figure I.6: The MSE of the IIM trained with different amounts of data for specimen eight using the SVR model and averaged strain data.

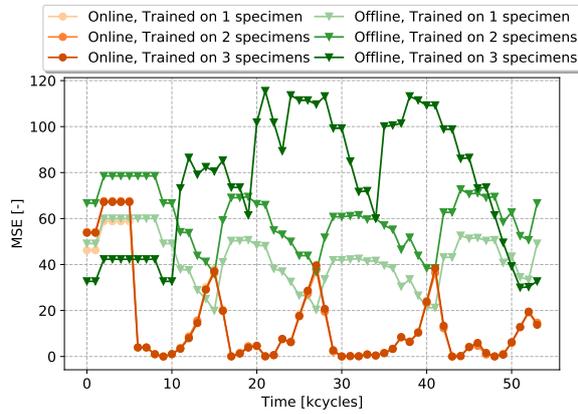


Figure I.7: The MSE of the IIM trained with different amounts of data for specimen six using the HMM model.

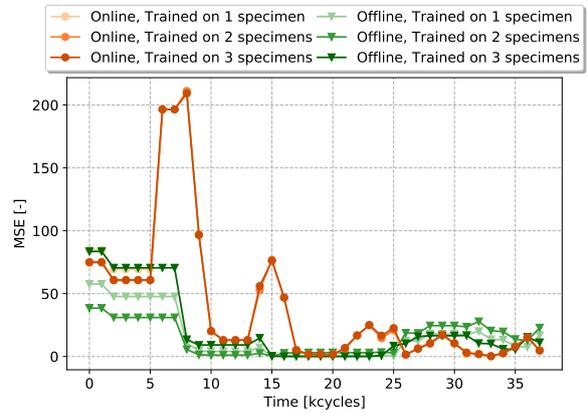


Figure I.8: The MSE of the IIM trained with different amounts of data for specimen seven using the HMM model.

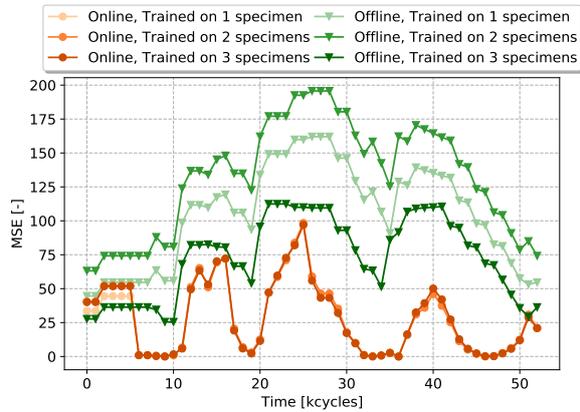


Figure I.9: The MSE of the IIM trained with different amounts of data for specimen eight using the HMM model.

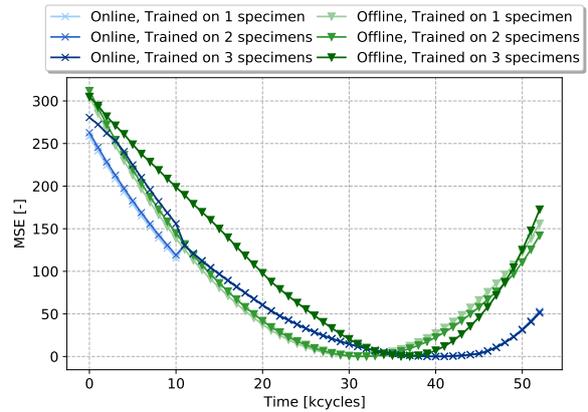


Figure I.10: The MSE of the IIM trained with different amounts of data for specimen six using the SVR model.

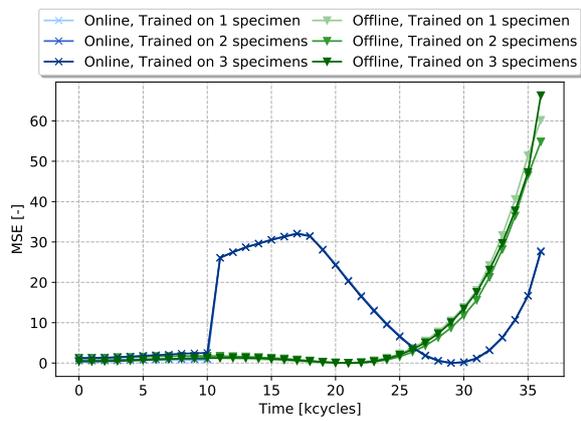


Figure I.11: The MSE of the IIM trained with different amounts of data for specimen seven using the SVR model.

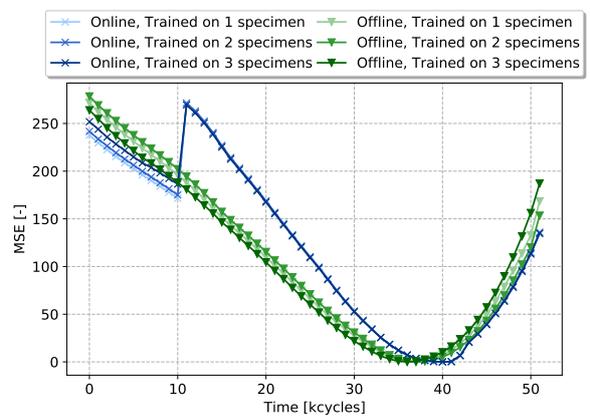


Figure I.12: The MSE of the IIM trained with different amounts of data for specimen eight using the SVR model.