Transmissibility-Based Monitoring and Combination of Damage Feature Decisions within a Holistic Structural Health Monitoring Framework

Von der Fakultät für Bauingenieurwesen und Geodäsie
der Gottfried Wilhelm Leibniz Universität Hannover zur
Erlangung des akademischen Grades
Doktor-Ingenieur
Dr.-Ing.

genehmigte Dissertation von
Dipl.-Ing. STAVROULA TSIAPOKI

Erscheinungs- bzw. Druckjahr 2018

Referent: Prof. Dr.-Ing. habil. RAIMUND ROLFES
Korreferent: Prof. JEROME PETER LYNCH,
University of Michigan

Tag der Promotion: 06.08.2018

Abstract

Over the past forty years, intensive research has been carried out in the field of structural health monitoring (SHM), since the identification of damage at an early stage contributes to avoiding structural failure and reducing maintenance costs. In particular, the monitoring of wind turbines has gained special interest, since there is an increasing number of installed wind turbines worldwide and a large number of wind turbines which have reached or will soon reach their design lifespan.

This thesis focuses on vibration-based SHM methods, which observe features describing the dynamics of a structure. Moreover, this work is based on the conception that the consideration of SHM should not only involve the observation of damage-sensitive features, but should also address further aspects, such as the effect of environmental and operational conditions (EOCs) and the statistical pattern recognition approaches used for decision making.

Wind turbines are complex structures which operate in a challenging environment. Most of the vibration-based approaches rely on assumptions which are violated, for example, during the operation of a wind turbine, raising doubts concerning their accuracy. Furthermore, there is a plethora of damage-sensitive features, alternatively called condition parameters (CPs), which can be used to assess the state of a structure. However, up to the present moment, little research has been conducted on the combination of damage feature selected and on the exploitation of decision making processes for improving the detection rates of damage when it exists.

This work introduces a new vibration-based CP, which does not rely on any significant assumptions. The new CP is based on an output-only version of an autoregressive model with exogenous input (ARX), which is essentially a transmissibility function (TF) model. The poles of the model are therefore called TF poles. The proposed CP is based on the observation of TF pole migration due to structural changes. Several experimental datasets are used to explore the sensitivity of TF poles to damage, while the concept of implementing TF poles as a CP in unsupervised mode is described. The new CP is integrated into a three-tier SHM framework which performs data normalizaton (tier 1), extracts the CP for analysis (tier 2) and subsequently makes use of hypothesis testing (tier 3). This framework using TF poles is validated on the fatigue test data of a full-scale rotor blade.

This work also proposes the implementation of adaptive boosting (AdaBoost) for the combination of decisions obtained from several damage features in order to attain a new and more accurate decision rule. The proposed concept is integrated into the aforementioned three-tier SHM framework and is used for combining the decisions of vibration-based damage features. However, the proposed concept can be implemented after any SHM process, even if other SHM approaches are employed. The concept of implementing Adaboost within the three-tier SHM framework is outlined and validated on the data of an operating

3 kW wind turbine.

Finally, different damage features, including the proposed CP, are compared with respect to their sensivity to damage and sensitivity to EOC variability based on rotor blade fatigue tests.

Zusammenfassung

Die Überwachung von Tragwerken und Bauteilen zur frühzeitigen Erkennung von Schäden, auch Structural Health Monitoring (SHM) genannt, kann zur Vermeidung von Strukturschäden und zur Senkung von Instandhaltungskosten beitragen. Insbesondere das Monitoring von Windenergieanlagen hat in den letzten Jahren stark an Interesse gewonnen, da die Anzahl sowohl von installierten Windenergieanlagen (WEA), als auch von WEA, welche deren Entwurfslebensdauer erreicht haben oder bald erreichen werden, ständig zunimmt.

Diese Dissertation setzt den Schwerpunkt auf schwingungsbasierte Verfahren, die Merkmale beobachten, welche das dynamische Verhalten der Struktur beschreiben. Darüber hinaus basiert diese Arbeit auf der Konzeption, dass sich die Betrachtung von SHM Methoden nicht nur auf die Beobachtung von Schadensindikatoren beschränkt, sondern auch weitere Aspekte mitberücksichtigen soll. Diese sind beispielsweise der Einfluss von Umwelt- und Betriebsbedingungen, sowie die Anwendung von statistischen Ansätzen zur Mustererkennung, die der Entscheidungsfindung dienen.

WEA sind komplexe Strukturen, welche in einem herausfordenden Umfeld betrieben werden. Die meisten schwingungsbasierten Verfahren beruhen auf Annahmen, die während des Betriebs einer WEA nicht eingehalten werden. Dies lässt Zweifel an der Genauigkeit dieser Verfahren aufkommen. Darüber hinaus gibt es eine Vielzahl von Schadensindikatoren, die auch als Zustandsparameter bezeichnet und zur Bewertung des Zustands der Struktur verwendet werden können. Bisher wurde jedoch wenig Forschung zur Kombination der Entscheidungen von Zustandsparametern (CPs von engl. condition parameter), sowie der Ausnutzung von Entscheidungsfindungsprozessen zur Verbesserung der Detektionsrate betrieben.

Diese Arbeit stellt einen neuen schwingungsbasierten CP vor, der auf keinen signifikanten Annahmen beruht. Der neue CP basiert auf einer Output-Only Variante eines autoregressiven Modells mit exogenem Eingang (ARX), das im Wesentlichen ein Transmissibility Function (TF) Modell ist. Die Pole des Modells werden daher als TF-Pole bezeichnet. Der vorgeschlagene CP basiert auf der Beobachtung der TF-Polmigration infolge struktureller Veränderungen. Mehrere experimentelle Datensätze werden verwendet, um die Sensitivität von TF-Polen zu untersuchen und in diesem Zusammenhang ihre Charakteristika zu verstehen. Das Konzept der Implementierung des CPs im unüberwachten Lernmodus wird beschrieben und der neue Schadensindikator in einem dreistufigen SHM-Framework integriert. Dieses umfasst dabei die Datennormalisierung (Stufe 1), die Berechnung des CPs (Stufe 2) und die Anwendung verschiedener Einstellungen bei der Entscheidungsfindung (Stufe 3). Der vorgeschlagene CP und die Integration in das SHM-Framework werden auf Basis von Messdaten eines Rotorblatt-Ermüdungstests validiert.

Des Weiteren wird in dieser Arbeit der Adaptive Boosting (AdaBoost) Algorithmus für die Kombination der Entscheidungen mehrerer Schadensindikatoren

vorgestellt. Das Ziel dieser Kombination ist es, eine neue und genauere Entscheidungsregel zu entwerfen. Das vorgeschlagene Konzept wird in das dreistufige SHM-Framework integriert, um die Entscheidungen der schwingungsbasierten CPs, welche vom SHM-Framework erhalten werden, zu kombinieren. Im Allgemeinen ist es möglich, AdaBoost nach jedem SHM-Prozess zu implementieren, selbst wenn andere SHM-Ansätze verwendet werden. AdaBoost schließt an das dreistufige SHM-Framework an. Dieses Konzept wird im Detail dargestellt und die Kombination der beiden Algorithmen auf Basis von experimentellen Daten einer 3 kW WEA werden validiert.

Abschließend werden unterschiedliche Schadensindikatoren, einschließlich des vorgeschlagenen CPs, hinsichtlich ihrer Empfindlichkeit gegenüber Schädigung und Schwankungen der Betriebs- und Umgebungsbedingungen auf der Grundlage des Rotorblatt-Ermüdungstests verglichen.

Acknowledgements

This thesis is the result of my activity as a PhD student and research assistant at the Institute of Structural Analysis of the Leibniz Universität Hannover. During my time at ISD, I had the opportunity to work mainly on one research project, which was funded by the German Federal Ministry for Economic Affairs and Energy and offered valuable material for this work. This thesis is not just the closing milestone of six years at the pleasant environment of the university, but also represents the end of a very challenging and mind-bending journey, during which one's limits and strength are often tested.

First of all, I would like to sincerely thank my advisor, Professor Raimund Rolfes, for giving me the opportunity to pursue a PhD on such an interesting and contemporary research topic and for the constant guidance on my thesis and further career steps. I am most greatful for his support, kind demeanor and for his contentment while seeing me grow in this procedure.

An important part of this thesis was excited and initiated during my research stay at the Laboratory for Intelligent Systems Technology of the University of Michigan. Words cannot express my gratitude and appreciation for Professor Jerome Lynch. His priceless supervision and the very interesting discussions during my time in Ann Arbor contributed immensely to my further development and are reflected in this work. Professor Lynch believing in my potential was one of the greatest sources of motivation.

A special place in these acknowledgements should undoubtedly be held for Moritz Häckell for his scientific and moral support especially at some very critical stages of this procedure. If it hadn't been for Moritz, I would not have made it to the end. I also feel the need to thank Moritz Fricke for his invaluable help during my first months at ISD and during the challenging preparations of the experiments and measurements. Other colleagues who supported me in several ways are Christian Claußen, Michael Treiber and, of course my project partner, Thomas Krause. I would also like to thank Omid Bahrami from the University of Michigan for the interesting scientific discussions that shed light into many aspects of my research, but mostly for making me feel a part of the laboratory during my stay in Ann Arbor. Moreover, many thanks are owed to Alexandros Antoniou, who advised me at many stages of this procedure and kindly cared for my progress and well-being.

At this point, I feel obliged to mention Professor Dimitrios Saravanos and Professor Theodoros Philippidis from the Department of Mechanical Engineerning and Aeronautics of the University of Patras for exciting my interest in scientific work and opening new horizons.

The gain of amazing friends during these years is of equal or even greater significance than the gain of knowledge. Many thanks to Cristina Diaz, Cristian Gebhardt, Sina Hühne, Jose Reinoso, Magoscha Szalyga, Benjamin Reil, Roozbeh Nabavi, Mehdi Garmabi, Sander van den Broek, Dorian Pache

and Timo Rogge. It was a pleasure spending time with you at the university and it is a greater pleasure to still enjoy your friendship. In my heart there is a special place for Ayan Haldar and Karsten Schröder, who supported me immensely both on a scientific and personal level especially during the last part of this procedure. Sharing office with you was a great pleasure and a time I look back on with nostalgia.

My friends outside ISD and especially Elisabet Zarkou, Nathalie Reeb and Amelie Fau are also owed many thanks for constantly encouraging me, giving me positive energy and making life in Hanover more joyful. My sincere thanks go also to Violetta Bennich, who not only proofread this thesis and significantly improved its quality in terms of language, but also encouraged me more than she can imagine during the last "fatigued" months. Finally, I am especially thankful to George Balokas for his unconditional support, for believing in the value of my work and for pulling me from the long working days back to a normal and carefree life. I am grateful and apologize for the hard time I might have given him.

Last but not least, I would like to thank my lovely parents and my sister, Aphrodite, for their love, tireless support throughout my entire life, trust and absolute respect for me and my decisions. I am so grateful you offer me a base to return to, always reminding me what really matters in life.

Stavroula Tsiapoki
Höchberg, November 2018

To my parents, Heleni and Lazaros,
to my sister, Aphrodite,
to Freddie.

I am the wisest man alive for I know one thing; that I know nothing.
Socrates c. 470-399 B.C.

Contents

List of Figures

List of Tables

Acronyms

AdaBoost adaptive boosting
ANN artificial neural network
ANNN auto-associative neural network
AP affinity propagation
AR autoregressive
ARARX autoregressive autoregressive with exogenous input
ARMA autoregressive with moving average
ARMAX autoregressive with moving average and exogenous input
ARX autoregressive with exogenous or extra input
ASCE American Society of Civil Engineers
AuC area under curve

BC Bhattacharyya coefficient

CL center line
CM condition monitoring
CP condition parameter
CT computer tomography

DOF degrees of freedom
DP damage prognosis

EFDD enhanced frequency domain decomposition
EMA experimental modal analysis
EOC environmental and operational condition

FDD frequency domain decomposition
FP false positive
FRF frequency response function

GPR Gaussian process regression

HT hypothesis testing

IRF impulse response function

LANL Los Alamos National Laboratory
LCL lower control limit
LS-SVR least squares support vector regression
LTS least trimmed squares

MA moving average
MAC modal assurance criterion
MDOF multi degree of freedom
ML machine learning
MRI magnetic resonance image
MSD Mahalanobis squared distance
MSE mean squared error

NDE non-destructive evaluation
NN neural network

OMA operational modal analysis

PCA principal component analysis
PDF probability density function
PSD power spectral density
PZT piezoelectric

RBF radial basis function
ROC receiver operating characteristic

SCADA supervisory control and data acquisition
SDOF single degree of freedom
SHM structural health monitoring
SK spectral kurtosis
SOP self-organizing maps
SPC statistical process control
SPR statistical pattern recognition
SSI stochastic subspace identification
SVD singular value decomposition
SVM support vector machine
SVR support vector regression

TEMP triangulation-based extraction of modal parameters
TF transmissibility function
TP true positive

UCL upper control limit

VAR vector autoregressive

List of Symbols

a significance level
$1 - a$ confidence interval
a_i coefficients of system output
a_t weight of the base classifiers in the AdaBoost classifier
b_i coefficients of system input
c damping constant
c_{cr} critical damping
d euclidean distance
$\bar{d}$ distance between the point of ROC curve and the diagonal
f_i natural frequency in Hz
$f(k)$ force varying with time in discrete time or system input
fn_i false negative detections
fp_i false positive detections
$f(t)$ force varying with time in continuous time
$\mathbf{f}(\mathbf{t})$ force vector varying with time
$h(t)$ transfer function in time domain
$h_t(\boldsymbol{\chi}_i)$ hypothesis of base classifier for data point $\boldsymbol{\chi}_i$
j imaginary unit of complex number
k stiffness as well as integer for discrete signals
m system mass
n model order
n_a order of the polynomial $A(q)$ containing the output coefficients
n_b order of the polynomial $B(q)$ containing the input coefficients
n_d number of dimensions
n_{in} number of inputs
n_{ch} number of output channels
n_t number of data points and number of training samples in AdaBoost
p_i poles (roots of the transfer function denominator/characteristic equation)
p_χ probability density function
q forward shift operator
r_n standardized normal variable
r_M^2 Mahalanobis squared distance
s complex number in the Laplace domain

$t_{i\%,n_s-1}$ i^{th} percentile of the t-distribution
tn_i true negative detections
tp_i true positive detections
$u(t)$ displacement varying with time in continuous time
$\mathbf{u(t)}$ displacement vector varying with time in continuous time
$\dot{u}(t)$ velocity varying with time in continuous time
$\mathbf{\dot{u}(t)}$ velocity vector varying with time in continuous time
$\ddot{u}(t)$ acceleration varying with time in continuous time
$\mathbf{\ddot{u}(t)}$ acceleration vector varying with time in continuous time
$v(t)$ measurement in state-space representation in continuous time
v_k measurement vector in state-space representation in discrete time
$w(t)$ process noise in state-space representation in continuous time
w_k process noise in state-space representation in discrete time
x normally distributed variable
$x(t)$ state vector in state-space representation in continuous time
x_k state vector in state-space representation in discrete time
y_i^h labels or hypotheses of weak classifiers used as input in AdaBoost
y_k model output in discrete time
$\mathbf{y}_k$ vector with model outputs in discrete time
z_i zeros (roots of the transfer function numerator)

$\mathbf{A_c}$ system matrix in continous state-space representation
$\mathbf{A_d}$ system matrix in discrete state-space representation
A_q polynomial with output coefficients for backward shift operator q
A_z polynomial with output coefficients for backward shift operator z
$\mathbf{B_c}$ input matrix in continous state-space representation
$\mathbf{B_d}$ input matrix in discrete state-space representation
C damping matrix
$\mathbf{C_c}$ output matrix in continous state-space representation
$\mathbf{C_d}$ output matrix in discrete state-space representation
$\mathbf{D_c}$ direct transition matrix in continous state-space representation
$\mathbf{D_d}$ direct transition matrix in discrete state-space representation
D_q denominator polynomial of noise transfer function in the generalized model structure
$\mathcal{D}_t$ distribution with probability that a point is chosen as training point in the next round of AdaBoost
FN_i false negative rate
FP_i false positive rate
$F(s)$ system excitation or system input in the Laplace domain
$F(z)$ system excitation or system input in the z-domain
$F(\omega)$ system excitation or system input in the frequency domain
G_q noise transfer function on the generalized model structure
H_0 null hypothesis

H_1 alternative hypothesis
H_{00} correct inference or true positive
H_{10} type error I or false positive
H_{01} type error II or false negative
H_{11} correct inference or true positive
$H(s)$ transfer function in the Laplace domain
$H(z)$ transfer function in the z-domain
$H(\omega)$ transfer function in the frequency domain
H_q input transfer function in the generalized model structure
$\mathcal{H}(\boldsymbol{\chi})$ hypothesis of the AdaBoost classifier
J Youden index
K stiffness matrix
LCL_n lower control limit
L_q denominator polynomial of input transfer function in the generalized model structure
$\mathbf{M}$ mass matrix
N number of degrees of freedom
S matrix with singular values obtained from singular value decomposition
T number of rounds in AdaBoost
T_{ij} transmissibility function between responses at locations i and j when input is applied at l
U left singular vectors obtained from singular value decomposition
UCL_n upper control limit
$U(s)$ system displacement in the Laplace domain
$U(z)$ system displacement in the z-domain
$U(\omega)$ system output in the frequency domain
V right singular vectors obtained from singular value decomposition
$X_{i\%}$ i^{th} percentile
$\mathbf{Y}_h$ Block Hankel matrix
$\mathbf{Y}_{hp}$ past row spaces of Block Hankel matrix
$\mathbf{Y}_{hf}$ furture row spaces of Block Hankel matrix

$\delta(t)$ damping ratio
ϵ_t error rate of base classifier in AdaBoost
ζ damping ratio
λ roots of the characteristic equation in time domain
μ mean value of a univariate distribution
$\boldsymbol{\mu}$ mean value of a multivariate distribution
σ standard deviation
σ^2 variance
σ_χ^2 variance of sampled normal distribution
τ_i a specific time
$\boldsymbol{\chi}_i$ n_d-dimensional feature used as input in AdaBoost

ψ_i eigenvectors from modal analysis
ω_d damped natural frequency or oscillation frequency
ω_n undamped natural frequency

$\hat{\Gamma}_i$ observability matrix
$\Delta\theta_i$ rotation of ellipse or difference between two angles
$\Delta\sigma_i$ difference between two standard deviations
$\hat{O}_i$ controlability matrix
Σ covariance matrix of a multivariate distribution

$|\cdot|$ absolute value
$(\cdot)^{-1}$ inverse of a matrix
$(\cdot)^T$ transpose of a matrix
$*$ convolution
$E(\cdot)$ expectation value
$erf(\cdot)$ error function
$exp(\cdot)$ exponential
$\mathcal{F}(\cdot)$ fourier operator
$Im(\cdot)$ imaginary part of a complex number
$\mathcal{L}(\cdot)$ laplace operator
$ln(\cdot)$ natural logarithm
$max(\cdot)$ maximum value
$min(\cdot)$ minimum value
$P(\cdot)$ probability
$Re(\cdot)$ real part of a complex number
$sign(\cdot)$ sign
$\mathcal{Z}(\cdot)$ z-transform operator

Chapter 1

Introduction

1.1 Structural Health Monitoring

Aerospace, civil and mechanical engineering infrastructure experiences inevitable ageing as well as damage accumulation due to the impact of the environment and extreme load events. Such events are earthquakes for civil infrastructure and 50-year storms for wind turbines. Changes in the material and/or geometric properties of a structure, including changes to the boundary conditions and system connectivity, adversely affect the system's performance and are defined as damage [1]. Failure occurs when the extent of damage is such that the functionality of the structure is affected. Structural failure and high repair and maintenance costs can be avoided by identifying damage existence at an early stage.

Over the past 40 years, research has been carried out in the field of damage identification by employing several adherent disciplines: structural health monitoring (SHM), condition monitoring (CM), non-destructive evaluation (NDE), statistical process control (SPC) and damage prognosis (DP). The process of implementing sensors to periodically observe a structure and autonomously extract information about its structural integrity is called structural health monitoring (SHM). While SHM usually refers to online monitoring (i.e., monitoring during the operation of the structure), NDE refers to evaluating structural components without damaging them. NDE usually involves offline testing under laboratory conditions and monitoring at a local level, but can also be performed while the structural component is operating. CM is employed for damage identification on rotating machinery. SPC originates from the field of quality control in industrial engineering and is employed to monitor changes in a process, which, in structural engineering applications, can be related to damage. In contrast to all the other disciplines, DP is employed after damage identification and deals with the estimation of the remaining useful life of the structure. Material characteristics, surrounding and operational conditions, as well as the tolerable damage extent of a structure contribute to the selection of the discipline, which is most suitable for monitoring.

Vibration-based SHM methods and NDE techniques are the most commonly used approaches for monitoring structural components. Vibration-based methods are based on the principle that changes in the physical properties of the system, i.e., damping and stiffness, cause detectable changes in the modal parameters and in other damage-sensitive features which describe the dynamics of the system [2]. Some of the most commonly used NDE techniques are ultrasonic testing, guided wave testing, electromagnetic testing, acoustic emission (AE) testing, infrared and thermal testing and computer tomography (CT) scanning. NDE techniques can be active or passive in nature. For instance, AE testing identifies damage by measuring elastic waves, which are generated by structural changes, such as material cracks and debondings [3]. While AE is a passive technique, ultrasonic testing is active, since it employs both actuators and receivers [4]. Actuators transmit ultrasonic waves into the material and sensors capture these waves at another location, with changes in wave characteristics indicating damage.

Most of the NDE techniques which are widely used for monitoring structural components are local approaches and employ many sensors on the structure (for instance AE and ultrasonic testing). This allows for the detection of relatively small damage, damage localization and, in some cases, even quantification. However, the capability to detect damage is highly dependent on appropriate sensor placement, which requires prior information about the damage location. On the other hand, vibration-based methods are used to monitor the structure at a global level providing information about the overall structural state. Vibration-based methods require fewer sensors and no previous knowledge about the anticipated damage location. Nowadays, the cost of SHM systems is proportional to the number of applied sensors. The robustness of vibration-based methods combined with the fact that they require few sensors renders them the most commonly applied methods for monitoring structures in operation. However, the question remains as to which types of damage can be detected by these approaches.

1.2 Wind Turbine Monitoring

Over the past decades, wind turbines have gained special interest due to the ongoing transition of energy production from fossil fuels to renewable energy sources. Depending on the assumptions on fossil fuel consumption, different estimations occur regarding fossil fuels depletion dates. Oil is estimated to deplete between 2044 and 2084, natural gas between 2046 and 2116 and coal between 2116 and 2316 [5]. A further motivation for the transition to renewable, clean energy sources is the negative environmental impact of fossil fuel consumption. Worldwide, efforts are being made towards the development and integration of renewable energy technologies that make use of wind, solar and geothermal energy, as well as ocean energy or hydropower. Europe in particular has a leading role in the integration of alternative sources of energy. The European Union's renewable energy directive sets a binding target of 20% final energy consumption from renewable sources by 2020

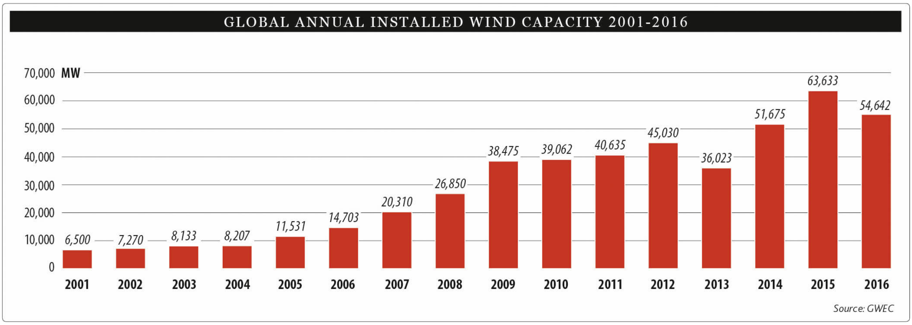


Figure 1.1: Annual installed wind capacity in MegaWatt for years 2001 to 2016 (source http://gwec.net).

and at least 27% by 2030.

Wind energy is a key component in the production of electricity from renewable sources and is currently one of the most established. Data on the global annual installed wind capacity show that more than 54 GW of wind power was installed across the global market in 2016, while the cumulative capacity from 2001 to 2016 reaches a total of 486.8 GW (see Figure 1.1). This implies that there is an increasing number of installed wind turbines, which have to be monitored in order to increase the availability and effectiveness, as well as decrease the cost of operation and maintenance. In addition, there is a large number of ageing wind turbines, which have reached or will soon reach their 25-year design lifespan. Especially these structures have to be monitored in order to assess whether they are able to further operate safely.

Wind turbines are complex structures that consist of structural components, rotating machinery and electrical components. The structural components are the blades and the support structure, which is understood as the complete structure below the nacelle, regardless of the type of structure. For instance, for onshore wind turbines the support structure is merely the tower, whereas for offshore wind turbines the support structure is composed of the tower and the monopile, tripile or jacket structure. SHM is used to monitor the structural components and CM to monitor the rotating machinery and electrical components [6]. Wind turbines are equiped with sensors which measure quantities describing the environmental and operational conditions (EOCs) of the wind turbine. Some examples are environmental data (such as the wind speed and wind direction), data describing the electrical characteristics of the turbine (such as the power output), control variables (such as the pitch angle of the rotor blades, the generator speed and the yaw angle), as well as temperatures at different components of the turbine [7]. These data are collected by the turbine supervisory control and data acquisition (SCADA) system and can be used to monitor the individual turbines or the entire wind farm. Today, SHM systems that monitor wind turbine structural components are available on the market, but planned visual inspections are still necessary.

One of the biggest challenges in wind turbine monitoring is the permanently changing environment in which the turbines operate. Therefore, data normalization based on the prevalent EOCs is essential, since it allows to distinguish between structural changes due to damage and structural changes due to variations in the EOCs. Most SHM systems for wind turbines observe the natural frequencies to identify structural changes. Other damage-sensitive features from different theoretical principles are a result of intensive research that has been conducted in order to obtain features that are more sensitive to damage than the rather insensitive natural frequencies. Another essential aspect of SHM is the decision making process and the boundaries or rules that are used in order to decide whether the structure has been subjected to changes. Over the past few years, there is an increasing preference for a more holistic monitoring approach that takes into account all the aforementioned aspects. The following section provides a literature review of state-of-the-art SHM concepts, damage-sensitive features for monitoring structures with a focus on wind turbines and decision making approaches.

1.3 State of the Art in Vibration-Based SHM

This section presents the state of the art in vibration-based approaches currently employed for monitoring structures. First, SHM concepts that provide an overall description of monitoring processes are presented. All presented concepts converge on the need for data normalization, the extraction of damage-sensitive features and the decision making process. Therefore, the section continues with data normalization techniques for the compensation of environmental and operational variability. These are distinguished into approaches, which do or do not require EOC measurements. Furthermore, an overview of the vibration-based methodology for damage detection is provided, followed by an overview of the state of the art in statistical pattern recognition for decision making in SHM. Thereafter, a brief overview of ensemble methods, which are used to combine decisions of different approaches, is given.

The fundamental idea of vibration-based damage identification is that structural change due to damage results in a change in the structure's dynamic behavior. Feature extraction refers to the process of transforming the measured data into some alternative form, where the correlation with the damage can be more readily observed. Subsequently, physics-based or data-based models are fitted to the measured system response and damage features are calculated. In the present work, damage features are alternatively called condition parameters (CPs). The term CP refers to any feature that is used to describe and monitor the condition of the structure. Hence, in this work, the terms damage feature and CP are used as equivalent.

The modal parameters of a structure are the most commonly used damage feature. Modal parameters can be estimated by three types of methods: the theoretical modal analysis, the experimental modal analysis (EMA) and the operational modal analysis (OMA), which can be used as an equivalent of output-only methods [8, 9, 10]. The theoretical modal analysis, which assumes the knowledge of mass, stiffness and damping matrices, constitutes an eigenvalue problem. EMA makes use of system responses in combination with information on the applied excitation to the structure. Over the past few years, OMA has received increased interest, since it does not require knowledge of the excitation and is thus suitable for monitoring structures under operational conditions. The presented state of the art focuses mainly on output-only vibration-based damage features.

Some other damage-sensitive features (or CPs) besides the modal parameters are: statistical moments of the responses, model errors (i.e., the difference between a model prediction and the actual quantity estimated), residues and model parameters. The majority of these features stem from system identification methods, which are based on nonparametric or parametric models. Parametric models describe the process behavior with a finite number of parameters. They are typically more elaborate and are characterized by higher computational complexity than nonparametric models, but are in general capable of achieving better performance [11]. Some examples of parametric models are autoregressive models and state-space models. On the other hand, nonparametric models generally require an infinite number of parameters to describe the process exactly. They offer simplicity and computational efficiency. Some examples of nonparametric models are the power spectral density (PSD) and the frequency response function (FRF). However, other models and approaches, which do not relate to system identification, are also used for damage detection, some examples of those being statistical moments of the measured responses and models based on artificial neural networks (ANN).

In the following subsections, CPs based on nonparametric methods are divided into (i) methods that identify modal parameters, (ii) methods that employ transmissibility measures and (iii) methods based on model residues. CPs based on parametric methods are divided into (i) methods which observe modal parameters and model coefficients, (ii) approaches which observe transmissibility measures, (iii) approaches which observe the migration of transfer function poles and (iv) approaches based on model residues. A summary of this categorization of damage features is presented in Figure 1.2.

After the computation of damage-sensitive features, their values are assessed in order to obtain a decision for the structural state. Therefore, statistical pattern recognition (SPR) approaches and approches that are based on neural network frameworks are employed. In SPR, different approaches may be applied depending on whether the problem is one of supervised or unsupervised machine learning. In supervised machine learning, the true class of the data is presented to the algorithm in training, while in unsupervised machine learning the true class is not available. Hence, in the following subsection, there is a distinction between novelty detection algorithms, classification algorithms and regression algorithms. Figure 1.3 provides an overview on the types of pattern recognition, as well as the available SPR algorithms, which

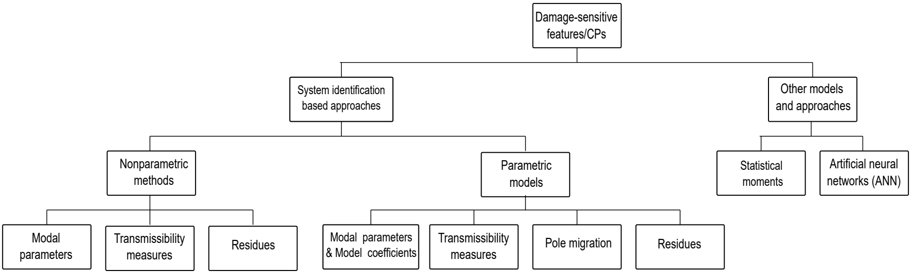

Damage-sensitive features/CPs
System identification based approaches
Other models and approaches
Nonparametric methods
Parametric models
Statistical moments
Artificial neural networks (ANN)
Modal parameters
Transmissibility measures
Residues
Modal parameters & Model coefficients
Transmissibility measures
Pole migration
Residues

Figure 1.2: Classification of damage-sensitive features.

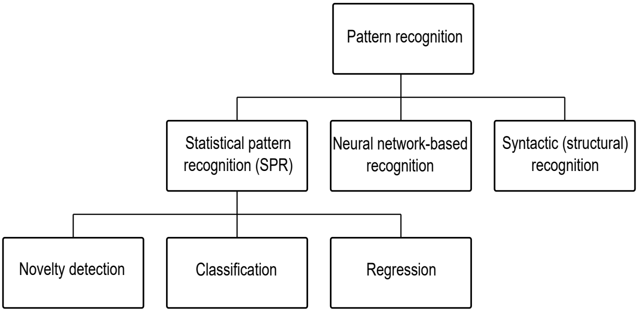

Pattern recognition
Statistical pattern recognition (SPR)
Neural network-based recognition
Syntactic (structural) recognition
Novelty detection
Classification
Regression

Figure 1.3: Decision making approaches based on pattern recognition.

are used in SHM for decision making.

1.3.1 SHM Concepts

In 2007, Worden et al. published the fundamental axioms and principles of SHM, summarizing thereby the extensive literature that had been developed up to that point in the field of damage identification [12]. Out of the seven axioms presented in that work, two are given special attention in the scope of the present thesis. Axiom II states that "The assessment of damage requires a comparison between two system states". Axiom IV has two parts and states that: (i) "Sensors cannot measure damage. Feature extraction through signal processing and statistical classification is necessary to convert sensor data into damage information" and (ii) "Without intelligent feature extraction, the more sensitive a measurement is to damage, the more sensitive it is to changing operational and environmental conditions". Rytter divides the damage detection process into four levels: damage detection (level I), damage localization (level II), damage assessment or quantification (level III) and damage prognosis (level IV), which refers to the estimation of residual life [13].

Several concepts that describe the damage identification process can be found in the literature. Farrar et al. define SHM processes in terms of a four-step statistical pattern recognition paradigm [14]. The suggested four-step process includes: (i) operational evaluation, (ii) data acquisition, normalization and cleansing, (iii) feature extraction and information condensation and (iv) statistical model development for feature discrimination. An alternative concept is provided in [4]. It is suggested that every SHM system should comprise three components regardless of the physical principle of its operation: data acquisition, signal processing tools and predictive models, and identification algorithms. A similar approach, the waterfall data fusion process model, is suggested by Bedworth et al. [15]. It includes the steps of sensing, signal processing, feature extraction, pattern recognition, situation assessment and decision making.

A framework for statistical time-series SHM methods is presented in [11]. The framework consists of the baseline phase and the inspection phase, both based on: (i) random excitation and/or response signals, (ii) statistical model building and (iii) statistical decision making under uncertainties. First, structural states are defined for the healthy state, the unknown to-be-tested state, and for damage types A, B and so forth. In both baseline and inspection phases, preprocessing, e.g., filtering, downsampling and normalizing, is followed by time series modelling, that implies the extraction of characteristic quantities that are analogous to damage features. Finally, in the inspection phase, statistical decision making is performed in the sense of hypothesis testing.

A modular SHM framework for the holistic long-term evaluation of structures is presented by Häckell in [16]. The framework consists of three tiers: (i) application of machine learning algorithms for data clustering based on EOCs, (ii) feature extraction and (iii) hypothesis testing (HT). Various damage features, also referred to as condition parameters (CPs), are calculated and different probabilistic models and settings for hypothesis testing are examined, resulting each time in a new realization of the SHM framework. The modular nature of the framework allows for many realizations, which contain different approaches along the three tiers. The SHM framework has been employed in several structures. In [17], the three-tier SHM framework is employed for monitoring the tower of a 5 MW offshore wind turbine of the alpha ventus offshore wind farm. In that study, there was no damage to be found and the focus lied on investigating the effect of varying EOCs on the structural dynamics, and on exploring the contribution of machine learning and data clustering to the increase of detection performance. Two further applications involve the Los Alamos National Laboratory (LANL) three-story building structure and the LANL 3.2 kW experimental wind turbine [16]. In both cases, the framework was employed to detect the artificial reversible damage, which had been induced to the structures. In [18], the framework is employed to detect real, fatigue-induced damage and ice accretion on a 34 m rotor blade of a wind turbine.

The aforementioned concepts describe the damage identification process, each in a different manner. While the concepts seem to differ, it can be said that they all converge on three stages: (i) the stage of preprocessing, including data acquisition, data cleansing and data normalization, (ii) the stage of feature extraction, including the implementation of signal processing tools, predictive models and identification methods and (iii) the stage of decision making, which includes the implementation of statistical models, pattern recognition and hypothesis testing.

1.3.2 Data Normalization

Data normalization is essential in order to distinguish between changes in the observed damage features that are caused by damage and changes that are caused by variation of the EOCs. For instance, in [19], it is observed that changing thermal conditions cause alterations in the modal properties of highway bridges ranging from 5% to 10%. The effect of operational conditions is shown in [20], where the natural frequencies of a short span bridge change up to 5.4% due to traffic loading. Worden and Farrar [21] provide a thorough overview of data normalization techniques. In Figure 1.4, this overview is summarized and the techniques are distinguished based on the availability of EOC measurements.

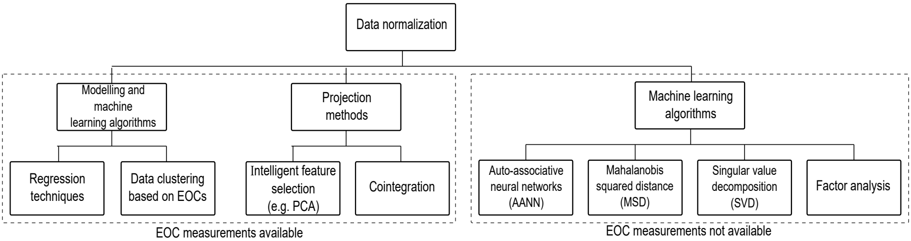

Data normalization
Modelling and machine learning algorithms
Projection methods
Machine learning algorithms
Regression techniques
Data clustering based on EOCs
Intelligent feature selection (e.g. PCA)
Cointegration
Auto-associative neural networks (AANN)
Mahalanobis squared distance (MSD)
Singular value decomposition (SVD)
Factor analysis
EOC measurements available
EOC measurements not available

Figure 1.4: Data normalization techniques categorized based on the availability of EOC measurements.

1.3.2.1 Data Normalization without EOC Measurements

When EOC measurements are not available, data normalization can be achieved by employing machine learning algorithms that develop a model which describes the influence of varying EOCs on the damage-sensitive features. These might be: auto-associative neural networks (AANNs), singular value decomposition (SVD) of matrices containing characteristic properties of the structure (for instance, natural frequencies, mode shapes and model coefficients) and algorithms, which employ factor analysis or the Mahalanobis squared distance (MSD). All of the aforementioned methods are based on the assumption that damage causes changes in the features, which are in some way orthogonal to the changes caused by EOC variations [21].

As their name implies, the output of AANNs reproduces patterns presented in the input layer. The special attribute of AANNs is a bottleneck in the hidden layers, which consist of fewer nodes than the input and output layers [22]. This architecture forces the network to learn the significant characteristics of the features, which, in the context of data normalization, are the correlations between the features of the training data. The trained network should be able to quantify the unmeasured sources of variability (such as EOCs), which affect the structural response. This variability is represented at the bottleneck output where the number of nodes should be equal to the number of unobserved independent factors which influence the structural response. In [23], an AANN is used along with other pattern recognition algorithms to detect damage on a 9 m long blade, which was tested under fatigue loading in LANL. On the other hand, the basic idea behind the employment of SVD for data normalization is that the rank of the state matrix, which contains some characterictic vectors of the structure, remains constant regardless of the EOCs, but increases in case of damage. The ability of SVD to identify damage is demonstrated on data from a finite element model of a truss structure and on the experimental data of a notched cantilever beam [24].

MSD is a multivariate measure of distance, which takes into account the covariance among the variables. It has an inherent normalization function, since, in its definition, the mean differences are divided by the covariance matrix. Factor analysis is a mathematical model that attempts to describe the correlation between a set of observed variables using a small number of unobserved underlying factors [25]. In [26], both MSD and factor analysis are employed for data normalization for the example of the three-story LANL structure. Finally, in [27], a data normalization procedure based on the minimization of the Euclidean distance is suggested, in order to minimize false positive warnings caused by the effects of EOCs.

1.3.2.2 Data Normalization with EOC Measurements

When EOC measurements are available, regression techniques, which model the relationships between the EOCs and the structural responses, as well as data clustering algorithms can be employed, with the latter being widely used in the monitoring of wind turbines in operation. Finally, some alternative approaches for data normalization, which involve projection are: (i) intelligent feature selection (i.e., the selection of features that are insensitive to damage), (ii) principal component analysis (PCA) and (iii) cointegration, with the latter being more suitable for unsupervised SHM applications [21].

Regression or heteroassociation deals with constructing a map between a group of continuous input variables and continuous output variables on the basis of a set of samples by minimizing an objective function [28]. Regression analysis includes linear, polynomial and autoregressive (AR) models, as well as machine learning algorithms such as support vector regression (SVR) and neural networks (NN). One of the most common applications of regression analysis in SHM involves the definition of the dependency between EOCs and the observed damage features.

A simple, linear relationship between traffic load and natural frequencies, which can be enhanced with a temperature-dependent variable, is used in [29] to predict the natural frequencies of the Tamar bridge. In [30], an adaptive filter is presented, which functions as a multiple linear regression model. The filter takes temporal and spatial temperature profiles as inputs and delivers the first natural frequency. The concept is validated on data from the Alamosa Canyon bridge and it is shown that it can reproduce the natural variability of the frequencies. A robust regression tool, least trimmed squares (LTS), is proposed in [31] as a means for characterizing and distinguishing the influence of EOCs on the structural response of bridges and civil engineering structures in general. In [32], a model is created for the 3.2 kW LANL wind turbine and Gaussian process regression (GPR) is used to extract features from the responses that correlate with EOCs and, at the same time, are insensitive to damage. The presented features correlate with the rotor angular velocity and nacelle yaw angle. Subsequently, measurement data of the real turbine is clustered by using these features instead of the actual EOCs. Least squares support vector regression (LS-SVR), a variation of SVR, is used in [33] to characterize the relationship between turbine power and weather variables, as well as to contribute to wind power production monitoring.

Data clustering is widely used in SHM of wind turbines in operation. Manual clustering or a variety of automatic clustering algorithms can be employed to cluster the structural data based on the EOCs. A review of clustering algorithms is provided in [34]. There are many different ways of categorizing clustering algorithms. According to the most general, clustering algorithms can be distinguished into hierarchical methods and partitional or point assignment methods. Hierarchical methods yield a dendrogram representing the nested grouping of patterns and similarity levels at which groupings change. Initially, each point is a distinct cluster and the two nearest clusters are repeatedly combined into one. Partitional clustering algorithms obtain a single partition of the data. They are cluster-based in the sense that a certain set of clusters is maintained, while each point is assigned to its nearest cluster.

In [35], operational cases are defined manually, resulting in clusters in the rotor speed-wind speed space, and data is normalized within each cluster. In [36], affinity propagation (AP) is employed to cluster the structural data of the 3.2 kW LANL wind turbine based on environmental data. AP is a hierarchical clustering algorithm, which takes as input measures of similarity between pairs of data points, which are called preferences, and exchanges information between them. The algorithm considers all data points as potential exemplars and hence does not require a predefined number of clusters [37]. In [16] the same clustering method is employed to cluster the structural data of a 5 MW offshore wind turbine based on EOCs. K-means clustering, a partitioning clustering algorithm, which requires predefinition of the cluster number, is used to cluster tower acceleration data of a 1.5 MW wind turbine, based on wind speed and drivetrain acceleration data [38]. Self-organizing maps (SOPs) are a type of artificial neural network (ANN), which gives an intuitevely appealing two-dimensional map of a multidimensional dataset. In [39], SOPs are used on SCADA data for damage detection on the gearbox of a 600 kW wind turbine.

1.3.3 Vibration-Based Damage Features from Non-Parametric Methods

1.3.3.1 Modal Parameters

Peak picking and the frequency domain decomposition (FDD) are two of the simplest methods employed for the identification of the modal parameters. In peak picking, the frequency response function (FRF) of the system is calculated and the peaks of the frequency spectrum are identified as the natural frequencies. Damping can be estimated by the half-power method, while the mode shapes are obtained from the amplitudes and phase angles for different degrees of freedom (DOFs). A disadvantage of peak picking is that the input of the system has to be measured in order to build the FRF. On the other hand, FDD is an output-only method that computes the power spectral density (PSD) of the signals and then applies singular value decomposition (SVD) to extract the natural frequencies and mode shapes. EFDD is an enhanced version of FDD which allows for the estimation of damping ratios. FDD assumes an independent white-noise input, i.e., constant PSD for the input. FDD was introduced in 2000 by Brincker et. al. and verified on a two-story building model [40]. In [41], FDD is employed to identify the modal parameters of a bridge.

Stiffness reduction due to damage causes decrease of the natural frequencies and changes in the mode shapes. Monitoring the changes of the identified mode shapes is among the most commonly used approaches for detecting damage in structures. In particular, mode shapes, curvature mode shapes and strain energy mode shapes are often used for damage localization. A disadvantage of these approaches is that a large number of sensors is required in order to capture the mode shapes. In [42], changes in curvature mode shapes are used for damage localization and are compared to displacement mode shapes.

Damping is also employed for damage detection. However, inaccuracies and errors are introduced during the estimation of damping values as well as in the definition of damping in the first place [43].

Another metric for damage identification, which belongs to the non-parametric approaches is the accumulated energy. Accumulated energy refers to the integral of power spectral densities over the measured frequency range. It serves as an indicator of energy distribution within a signal and takes into account changes in the entire frequency range instead of just single peaks. In [16], the mean frequency for 90-100% of accumulated energy of a single sensor is used as a damage feature for monitoring an offshore wind turbine.

While the modal parameters are one of the most powerful representations of system dynamics and exhibit sensitivity to structural changes, they also exhibit variability to EOC changes. The variability of the natural frequencies and mode shapes of the Alamosa Canyon bridge, which are caused by EOC changes, are discussed in [44]. This work shows that changes in the enviromental and service conditions, such as thermal gradients and traffic loads, introduce significant variability of the modal parameters and suggest that the effect of these variability sources to the modal parameters be quantified prior to the damage identification process. As a result, special attention has to be paid when employing the modal parameters as a damage indicator, in order to identify changes that are caused by effects other than damage.

1.3.3.2 Transmissibility Measures

Transmissibility functions are defined as the ratio between two response spectra. They are employed to extract damage-sensitive features, especially when the excitation of a structure is not available or cannot be measured. Johnson et. al. use auto- and cross-power spectral densities to compute the transmissibility functions between different degrees of freedom (DOFs) of the structure [45]. The transmissibilities are further normalized by computing their logarithms. Relative changes in both the transmissibility functions and their logarithms are computed for the baseline state and for the current datasets. Relative changes in all DOFs are then summed up to build a damage index for damage detection and localization on the example of the LANL three-story building. Furthermore, it is proven that the transmissibility functions are insensitive to nonlinearities at the structural boundaries as the operational conditions change. In [46], the ratio of two response spectra is used to build a transmissibility function and the difference between the transmissibility functions of different DOFs is used to calculate the system zeros and their properties. It is shown that the modal parameters of a system can be obtained from the suggested approach, i.e., by combining transmissibility measurements from different loading conditions. Devriendt et al. [46] employ the logarithm-related damage index of Johnson et al. [45] along with the comparison of transmissibility functions of different states to show that damage indicators based on transmissibility measurements can also be used for the challenging case of changing loading conditions. In particular, this is achieved by considering small frequency bands around the resonance frequencies of the structure. Similarly, the ratio of frequency responses is used in [47] to detect and localize damage on a 40 m rotor blade. FRFs are used in [48] to build transmissibilities on a plate with stiffeners that simulates a metallic aircraft wingbox. The transmissibilities are evaluated by three novelty detection algorithms: outlier analysis, density estimation and an auto-associative neural network (AANN).

1.3.3.3 Residues

Some direct ways for calculating residues for damage detection can be defined by the comparison of distances, FRF and frequency spectra. For example, in [49], FRFs are compared for damage detection on a truss structure. Furthermore, in [24], the amplitude of FRF is used to build characteristic vectors, which are arranged in a matrix. Singular value decomposition (SVD) is used to obtain the matrix singular values and matrix rank, which are used to build a damage index. The residual nature of the damage index is due to the fact that the matrix rank is related to the undamaged structure.

1.3.4 Vibration-Based Damage Features from Parametric Methods

1.3.4.1 Modal Parameters and Model Coefficients

The stochastic subspace identification (SSI) technique was introduced in 1996 by Van Overschee and De Moor [50]. It is a time domain method that uses output-only measurements and describes a dynamic system as a state-space model. The approach is based on the assumption of white-noise excitation and linearity. SSI can be either covariance-driven or data-driven. Covariance-driven SSI makes use of the output correlations for building the Hankel matrix. The factorization property might then be used to obtain the controllability and observability matrices and to extract the modal properties. In the data-driven approach, the weighted Hankel matrix is built directly from measured signals. Then, the factorization property might be used to obtain the observability matrix and the Kalman filter state sequence. One of the advantages of the SSI technique is that many model orders can be derived by a single block matrix, resulting in a low computation time. In [51], both SSI schemes are used for the operational modal analysis of an aerospace vehicle. An application of SSI in footbridge monitoring can be found in [52]. In [17], the triangulation-based extraction of modal parameters (TEMP) is proposed for reducing the solutions obtained from data-driven SSI. The combination of data-driven SSI is verified on tower data from a wind turbine of the alpha ventus offshore wind farm. This combination is also employed in [18] for identifying the modal parameters of a 34 m rotor blade.

In signal processing, autoregressive (AR) models use time series of structural responses to build a model described by model coefficients. This model is used to predict the present time step based on past values of the signal and subsequently to identify the model coefficients of the system. Precisely, the present output is expressed as a linear combination of past outputs. Many variations of AR models are used for system identification, such as vector autoregressive models (VAR), moving average (MA) models, autoregressive moving average (ARMA) models and autoregressive models with exogenous (or extra) input (ARX). The nature of all the aforementioned models in the context of system identification is presented thoroughly in [53]. VAR models use more than one channel. MA models estimate the present output as a linear combination of only past inputs, that refer to a stochastic term. ARMA models are combinations of AR and MA models, and offer higher accuracy than MA models for a much lower model order. ARX models predict present outputs based on past output values and values of an exogenous source, which is usually the excitation. While AR, MA and ARMA models are stochastic time series models and assume stationarity, the ARX model is not based on this assumption [54].

Bodeux et al. employ an ARMA vector model in order to identify the modal parameters and perform damage detection on the benchmark of a testing steel building [55]. In [56], a VAR model is used to identify the modal parameters of an offshore wind turbine. In [57], SSI is used to identify the natural frequencies of the Z-24 bridge, which are regarded as a time series of measurement values. At the next step, an ARX model is built using temperature measurements as input and the natural frequency time series as output, creating thereby a model that includes the thermal dynamics of the bridge.

Model coefficients can also be directly used as damage features. In [58], AR parameters are used as damage-sensitive features for the LANL three-story frame structure. In [59], the Mahalanobis squared distances of AR coefficients are used for detecting damage on concrete beams.

1.3.4.2 Transmissibility Measures

When ARX models are used as output-only models by solely considering system responses, they are essentially transformed to transmissibility function models. In [60], an ARX model, which involves only acceleration signals is considered for damage identification. Based on numerical examples, the model residual error, i.e., the difference between the measured signal of the actual state and the predicted signal, as well the residual error standard deviation are shown to be damage-sensitive features. For a multi-DOF (MDOF) system, the standard deviation of the residual error can also serve for damage localization, since higher values are observed close to damage. Moreover, for an undamped system, the model parameters are expressed in terms of stiffness. A similar formulation is provided in [61], where the parameters of an ARMAX model are expressed as a function of stiffness and damping. In [62], an ARMA model is treated as an ARX model and two damage localization indices are defined based on the AR coefficients of the model. The two indices are tested on a numerical simulation of the American Society of Civil Engineers (ASCE) benchmark structure. The results show that one of them is capable of localizing even minor damage, while the other localizes significant damage, but is non-conclusive for minor damage.

A frequency domain ARX model, which is able to distinguish between nonlinear and linear damage, is used in [63]. Despite the fact that the model is a transmissibility model in frequency domain, it falls into the category of parametric methods, since the model is described by the coefficients of the AR and the exogenous part. The AR coefficients are used to characterize nonlinear damage, whereas the exogenous coefficients are used to characterize linear damage. The approach is verified on a 4-DOF simulation model and on the LANL three-story building structure. The same frequency domain ARX model is used in [64] to detect damage on the same three-story structure based on the impedance of piezoelectric (PZT) sensors that are used as both sensors and actuators.

1.3.4.3 Migration of Transfer Function Poles

Transfer functions have the system output in the numerator and the system input in the denominator. The roots of the denominator characteristic equation are called poles. Poles constitute the solutions which maximize the transfer function, since for these values the transfer function tends to infinity. In structural dynamics, poles correspond to the natural frequencies and the damping ratios of the structure. On the other hand, the roots of the numerator are called zeros. Zeros are the solutions for which the transfer function is equal to zero, i.e., the values for which the system attenuates the input. In control systems engineering, poles and zeros are a means of designing and controlling the response characteristics of a system. Hence, also in structural engineering, pole migration on the complex plane can serve as an indicator of structural changes, since poles encapsulate information on both natural frequencies and damping ratios.

The concept of transfer function pole migration for SHM is introduced by Lynch in [65]. Pole migration in ARX models is employed in a cantilever aluminum plate to distinguish between healthy and damaged poles. The plate is damaged by hack saw cuts and is actively sensed by PZT pads. In [66], the same concept is used for detecting damage on the IASC-ASCE benchmark structure. Poles from each state are clustered for all states. It is shown that pole cluster migration is sensitive to damage and that the degree of pole cluster migration is affected by the extent of damage. In [67], transfer function pole migration is successfully applied for damage characterization on the Z24 bridge data. A damage index is built as a weighted sum of three indices, which are based on three different types of classifiers: (i) a classifier based on the nearest mean, (ii) the perceptron classifier and (iii) the mean separation distance criterion. The classifiers are used to distinguish between healthy and damaged poles.

1.3.4.4 Residues

As an alternative to the classical modal parameters, residues from the covariance-driven SSI technique are used to detect changes in the system's behavior. The most commonly used SSI residue was introduced by Baseville [68]. First, the Hankel matrix of output covariances is built and SVD is performed to estimate the left nullspace of the Hankel matrix. Residues can be formed by multiplying the nullspace matrix of the reference Hankel matrix by the Hankel matrix of the current dataset. This residual damage feature shows sensitivity to damage but also to varying EOCs. Therefore, data normalization is necessary prior to its deployment. In [56], the aforementioned residue is employed along with manual data clustering for monitoring an onshore wind turbine. Another residual vector, which is supposed to be robust to changes in excitation is introduced by Döhler [69]. This formulation is based on the fact that the Hankel matrix has the same nullspace matrix as the matrix containing the left singular vectors. Thus, a new residue is built by multiplying the left singular vectors by the Hankel matrices of the new datasets. Both residues are used for detecting damage and ice on rotor blades [18] and for detecting structural changes on the LANL three-story test structure [16].

Residual damage features can also be calculated from VAR models. For this purpose, the time series of a measured response is compared to the time series estimated by the autoregressive (AR) coefficients. One residue is the fitted coefficient of determination, which makes use of the actual time series values (e.g., acceleration signals), the model estimation and the mean value of the measured signal [70]. Another residue makes use of the model error covariance matrix and is based on Box's M-Test, which compares the error covariance matrix of the current dataset with that of the reference dataset [71, 56]. The residual error of a two-stage AR-ARX model, similar to an ARMA model, is used in [27] to identify damage on an 8-DOF mass-spring system. The prediction error of the AR model is calculated and is used as input in an ARX model for the prediction of the system responses. The residual error of the ARX model is then used as a damage-sensitive feature.

1.3.5 Other Approaches and Algorithms

Statistical moments also provide information about the characteristics of measured responses. The first two moments, mean and standard deviation, provide infromation on signal characteristics. The third moment, skewness, is a measure of the asymmetry of a probability density function (PDF). The fourth moment, kurtosis, indicates whether the data is peaked or flat compared to the normal distribution. Spectral kurtosis (SK) is an alternative form of kurtosis in frequency domain, which refers to the kurtosis of a signal's frequency components. For a random signal, SK equals zero, for a stationary harmonic process SK equals -1 and for a non-stationary signal SK is greater than zero [72]. SK can be used as a metric for the deviation from Gaussianity enabling the detection of a structural damage event, which is supposed to cause impulse loading or nonlinearities and subsequently a change of the PDF. An alternative technique for the estimation of the underlying PDF of a random variable employs the kernel density estimator [73].

Artificial neural networks (ANNs) are valuable models widely used in SHM. ANNs express an unknown relationship between the input and the output variables by means of linear and nonlinear relations between the hidden layers [74]. Depending on the selected network structure, many network types occur, which may be used for different purposes, e.g., linear or nonlinear autoregressive ANNs, autoassociative ANNs, radial basis function (RBF) NNs etc. Hence, only a few examples are presented here. In [23] an autoassociative neural network (AANN) and a RBF network, which is also used for auto-association, are used for damage detection on a 9 m rotor blade. While the hidden layers of a neural network compute the nonlinear function of the scalar product between the input vector and a weight vector, the hidden neurons of RBF networks are activated by a nonlinear function of the distance between the input vector and a weight vector. Similarly, an AANN is employed in [26] for detecting damage on the LANL three-story structure.

1.3.6 Statistical Pattern Recognition for Decision Making

Based on the values of the extracted damage-sensitive features or CPs, a decision has to be made about the condition of the structure. More specifically, the damage feature values have to be mapped to particular states of the structure, which are alternatively called classes. In the simplest case, these classes are the "healthy" state and the "damaged" state. Further cases include different structural states, extents of damage or different damage locations. The process of recognizing underlying relationships between damage feature values and structural states, and, subsequently, assigning specific structural states to damage feature values is called pattern recognition. Pattern recognition comprises three theoretical frameworks: the statistical, the syntactic (or structural) and the neural network-based framework [21]. The statistical and the neural network-based frameworks are the two most commonly used in SHM. Some neural network-based approaches for damage detection were presented briefly in section 1.3.5. Syntactic pattern recognition is applied when patterns are difficult to quantify as feature vectors. Instead, it enables describing large sets of complex patterns by using small sets of simple patterns. Some application fields for syntactic pattern recognition are computer vision and image processing.

Statistical pattern recognition (SPR) stems from the fields of statistics and machine learning. Machine learning can be supervised or unsupervised, depending on whether the true state (or class) of the data is available in training. For instance in terms of damage detection, in supervised machine learning, healthy and damaged datasets are used in training. In unsupervised machine learning, on the other hand, only data from the normal condition or the baseline state of the structure is used in training. Supervised machine learning applies to all the four SHM levels, i.e., damage detection, localization, quantification and prognosis, while unsupervised machine learning applies to level I and sometimes to level II. Ideally, SPR algorithms should return their decision with a confidence interval. Depending on the available type of data and information, and, subsequently, the type of machine learning, the following three types of SPR algorithms can be employed: (i) novelty detection algorithms, (ii) classification algorithms and (iii) regression algorithms.

Novelty detection algorithms use unsupervised learning and are employed in two-class problems, where classes are either "healthy" or "damaged". The output indicates deviations from the baseline state or the normal operational condition. Novelty detection can be carried out by defining decision boundaries on damage feature distribution obtained in training and by performing hypothesis testing. Decision boundaries may be demonstrated in control charts based on the methodology of statictical process control (SPC). Usually, the decision is returned with a confidence interval. Further approaches, which can be employed for novelty detection are one-class support vector machines (SVMs) and novelty indices based on distance measures, such as the Euclidean distance and the Mahalanobis squared distance (MSD).

Classification algorithms use supervised learning and are used in two-class or multi-class problems. In this case, the output is the class label (e.g., "healthy" or "damaged"). It is common practice to perform hypothesis testing by setting decision boundaries based on the probabilities of the individual classes. Further classification algorithms include SVM, neural networks and decision trees [75].

Regression algorithms use supervised learning and output continuous variables. Regression analysis includes linear models, polynomial models and AR models, as well as support vector machines (SVMs) and neural networks [16]. Since the present work focuses on novelty detection and classification problems, only a brief overview of the state of the art for these two types of algorithms is given.

In an earlier section, it was shown that distance measures can be used for data normalization. In the following bibliographical references, distance measures are used to build novelty indices. Surace et al. use the Euclidean distance between transmissibility functions to detect damage on an offshore platform model and a finite element model of an aircraft wing [76]. The MSD of AR parameters is used for feature classification in [58], where a three-story frame structure is tested under operational and environmental variability. In [77], the MSD of an error deriving from principal component analysis (PCA) is used for damage detection on a footbridge. In [16], MSD is used to build a hybrid condition parameter (CP). Multidimensional CPs, which are composed of one-dimensional CPs, are evaluated based on the MSD for detecting damage on the 3.2 kW LANL wind turbine.

Häckell et al. employs statistical process control and hypothesis testing in the third tier of a modular SHM framework based on unsupervised learning. The upper and lower control limits (UCL and LCL) of control charts are set for various CPs and a series of confidence intervals before hypothesis testing [36]. SPC is also employed in [78]. Control charts are used to observe the AR coefficients during quasi-static cyclic tests of reinforced-concrete bridge columns. An extended reference to hypothesis testing in unsupervised mode is given in [11]. The null hypothesis is defined based on the PDF of damage features and current datasets are evaluated with respect to that for damage detection and localization. This is shown for a series of parametric and non parametric approaches, such as the PSD-based method, the FRF magnitude-based method, the modal parameter based method and the model residue-based method.

One-class SVMs are variations of SVMs which can be used in unsupervised machine learning. One-class SVMs allow the classification of new data points based only on data from one class. In SHM, this implies the classification of a current dataset based on information from the healthy state only. The advantage of this approach is that one-class SVMs are able to create nonlinear decision boundaries by employing kernel functions. In [79], the concept is described and verified on a steel frame structure. On the other hand, SVMs can be used for supervised damage detection. Kernel functions transform points into other spaces allowing for nonlinear classifiers and decision boundaries. The application of SVMs is presented in [80] for damage detection on bridges, and in [81] for damage classification on ball bearings, in order to distinguish between different damage scenarios, and damage localization on a two-dimensional cantilever truss structure.

1.3.7 Combining Damage Features Decisions with Ensemble Methods

Regardless of the examined damage feature, classifiers evaluate the results and deliver a decision about the state of the structure. Classifiers map data to classes, depending on the corresponding damage feature values. In damage detection problems, the classes are labeled as "healthy" or "damaged" (level I of damage detection processes according to Rytter [13]). In damage localization problems, the classes comprise different damage locations (level II), while in damage characterization problems, the classes contain different types of damage (level III). Parallel observation of different damage features and, subsequently, classifiers can be beneficial and offer a better overview of the structural state. However, there are often discrepancies among the decisions of classifiers even on the exact same dataset, resulting in different detections for each damage feature.

Ensemble methods deal with improving the performance of algorithms by combining the output of several classifiers. In the context of SHM, this refers to combining decisions of different damage features. There are numerous classifier combination methods, e.g., majority vote, bagging, boosting and model averaging [82]. Boosting algorithms aim at improving the performance of learning algorithms by combining multiple base classifiers. The result is an ensemble classifier, whose joint decision offers a better performance than any of the individual classifiers [75]. Base classifiers are alternatively referred to as weak classifiers or learners, with term "weak" emphasizing the mediocrity in performance and the term "base" connoting their use as building blocks [83]. In Adaptive Boosting (AdaBoost), base classifiers are trained in sequence, resulting in a strong classifier, which has a form of a weighted sum [84]. AdaBoost is a widely used meta-algorithm, with numerous applications in classification problems, mainly in the field of image processing.

Some studies in the fields of face and object detection can be found in [85, 86, 87]. Applications are also found in other fields, such as recognition of urban building structures [88], bankruptcy forecasting [89] and medical imaging [90]. For instance, in [90], AdaBoost is employed for the identification of Alzheimer's disease by evaluating the hippocampal volume from magnetic resonance images (MRI). An application in civil engineering and, more specifically, in constructions can be found in [91], where AdaBoost is used for selecting an appropriate retaining wall method suitable for particular construction site conditions. Ten factors, which contribute to selecting the retaining wall method are selected. Based on these factors, which serve as weak classifiers, AdaBoost chooses between six wall retaining methods.

However, up to now, AdaBoost has been used in only a few studies in the field of SHM, i.e., for damage detection, localization and classification. A simplified version of AdaBoost is utilized in [67] for damage detection based on the observation of transfer function pole migration. Three approaches, serving as weak classifiers, classify the poles of test datasets with respect to the baseline healthy state. Subsequently, the decisions of the three weak classifiers are weighted, in order to combine their results in a single damage index. Real AdaBoost, a modified version of AdaBoost, is used in [92] to detect and locate damage on a bridge based on acceleration signals acquired by a wireless sensor network. As opposed to AdaBoost, where weak hypotheses are equal to +1 or -1, in RealAdaBoost, the weak hypotheses may be real numbers. Damage classification of reinforced concrete bridge deck damage is performed in [93]. Crack characteristics are extracted from digital images of concrete decks. These characteristics are then used in AdaBoost, in order to classify the damage level. In [94], AdaBoost is used to distinguish between the two most common types of damage in metallic structures: cracks and corrosion. Four different signal processing methods, in time and frequency domain, are evaluated with respect to their suitability for damage classification. Spectrograms are proven to be the most suitable method for damage classification and are used as weak classifiers in AdaBoost, which is capable of correctly classifying the two types of damage.

1.4 Research Objectives

Over the past two decades, more holistic SHM approaches have progressively gained interest. Such concepts were presented in section 1.3.1. In this sense, the present dissertation is based on the conception that the consideration of SHM problems should not be restricted to the observation of damage features, but should also address further aspects, such as the effect of EOCs and the approaches used for decision making. The main focus of this work lies in wind energy applications and in the exploitation of structural response data for the detection of structural changes. However, the applicability of the presented research is not limited to the field of wind energy but also expands into other fields.

Vibration-based SHM offers a plethora of approaches, which can be used for assessing the state of a structure. Modal parameters, model coefficients and model residues are some of the most widely used damage features. The majority of approaches are often based on assumptions, which are violated in real-life applications and during the operation of structures. Two examples are the assumption of stationarity for vector autoregressive models and the assumption of white-noise excitation for the stochastic subspace identification approach. During the operation of a wind turbine, for instance, both aforementioned assumptions are violated due to the rotor frequencies and the varying EOCs. The violation of these assumptions is a limiting factor, which raises doubts about the accuracy of some methods and sometimes does not allow their deployment. In sections 1.3.3 and 1.3.4, it was shown that research has been performed on system pole migration and transmissibility function models. However, up to now, no studies have been conducted on the migration of poles obtained from transmissibility functions.

It can be stated that it is unlikely that a damage feature is sensitive to all damage scenarios, since each damage feature exhibits different sensitivity to different types of damage or environmental settings. For instance, there are often discrepancies between the desicions of two damage features even on the exact same dataset. Parallel observation of different damage features can be advantageous and offer a better overview of the structural state. Yet, up to the present moment, little research has been conducted on the combination of damage feature decisions and on the exploitaton of decision making processes involved in one damage feature for improving the detection rate of another damage feature.

The first research objective of this thesis is the development of a damage feature, which does not rely on any significant assumptions. For this purpose, a modified autoregressive model with exogenous input (ARX) is considered. In the classical ARX model, external forces are used as input to the model. Due to the lack of this information for many applications, especially for an operating wind turbine, structural responses measured in one position of the structure are used as input to predict the responses in other positions. As a result, the modified ARX model becomes a transmissibility function model, whose poles do not correspond to the system's natural frequencies and damping. Instead, they represent the zeros of the system, which constitute the positions, in which the system attenuates the input.

Structural changes affect sytem poles and zeros, which, for the case of the modified ARX model, will be called transmissibility function (TF) poles and TF zeros. Thus, structural changes cause migration of TF poles on the complex plane, on which they are presented. The monitoring of TF poles can be beneficial because they do not involve system excitation and, at the same time, encapsulate frequency and damping information. The objective of this work is to understand the mechanism behind TF pole migration and to develop a new damage feature based on it, which is appropriate for online monitoring, i.e., for monitoring in unsupervised mode.

Another research objective of this thesis lies in the combination of the decisions of different damage features. In the context of SHM, classifiers are rules, which map data to specific classes, such as "healthy" or "damaged" based on damage feature values. Boosting algorithms aim at improving the performance of learning algorithms by combining multiple base classifiers. In this work, adaptive boosting (AdaBoost) is proposed for building a strong classifier based on the classifiers of individual damage features. The objective is to combine the decisions of various damage features and to exploit them in order to build a more powerful decision rule and improve detection performance. The suggested concept can be employed following any SHM process, which evaluates damage features and provides a decision regarding the structural state. The focus of this work lies in employing AdaBoost after the application of the three-tier SHM framework, which was presented in section 1.3.1. The modular nature of the three-tier SHM framework is ideally suited for boosting, because each connection between CPs and HT can be conceived as a classifier which constitutes a weak classifier in the boosting algorithm. Aim of this thesis is to exploit information from the SHM framework classifiers in order to compose a strong classifier which has the form of a weighted sum and is able to classify values of different CPs.

Several databases have been analyzed in order to conduct these studies. These include the experimental data of test structures and real structures, as well as data from real structures in operation. In particular, for the investigation of AdaBoost, the deployment of several damage features was required. All these analyses were carried out in the context of a holistic SHM procedure, which takes into account the variation of EOCs. A byproduct of these analyses is information on the detection performance of different damage features under consideration of EOCs variability. Therefore, a further goal of the present thesis is to perform a comparative study of sensitivity to damage and EOCs variability for different damage features.

1.5 Organization of the Thesis

Chapter 2 provides the theoretical background for the proposed methods and can be divided into two parts. The first part briefly presents the basic theory of structural dynamics, provides an overview of system identification methods and describes the classical and the modified ARX models. Special focus is placed on system poles and zeros. The second part of the chapter presents the theoretical background for decision making processes employed in SHM and more specifically for the new TF pole migration-based damage feature. Moreover, this part of the chapter provides definitions on boosting algorithms. In particular, the AdaBoost algorithm is described in detail.

Chapters 3 and 4 introduce the innovations of this thesis. Chapter 3 introduces a new damage feature, which is based on TF pole migration. Moreover, it sets the basis for the deployment of the new damage feature in the context of an SHM scheme, which evaluates current datasets with respect to datasets used in the training phase. Chapter 4 describes the integration of AdaBoost with a holistic SHM concept. Special focus is placed on the interface between such an SHM concept and AdaBoost. Furthermore, metrics facilitating the evaluation of boosting performance are defined.

Chapter 5 presents a sensitivity analysis of TF pole migration. This chapter provides insight into the factors, which affect TF pole positions, as well as the sensitivity of TF poles to damage and EOCs changes. This analysis is performed for the database of the LANL three-story test structure and for two databases obtained from a full-scale rotor blade test. The results of this chapter serve as a guide for developing a monitoring strategy within the context of an SHM concept which evaluates datasets in an unsupervised mode, i.e. without having trained with damaged datasets.

The validation of the new damage feature based on TF pole migration is presented in Chapter 6. The proposed damage feature is employed within the context of the three-tier SHM framework, which was presented in section 1.3.1. For this purpose, two databases are analyzed. The first contains datasets from a fatigue test of a full-scale rotor blade and the other experimental datasets of an operating small-scale wind turbine.

The validation of the conjunction of AdaBoost with a three-tier SHM framework is presented in Chapter 7. The concept introduced in Chapter 4 is employed on the simulated datasets of an operating small-scale wind turbine and subsequently on the experimental datasets of a small-scale wind turbine.

Chapter 8 examines the quantifiability of sensitivity to damage and EOCs variations of different damage features. Several damage features, including the one proposed in this thesis, are evaluated on various databases and compared to each other with respect to sensitivity to damage and sensitivity to varying EOCs.

Chapter 9 serves as a summary and provides a discussion on the main conclusions of the present thesis.

Chapter 2

Theoretical Background

This chapter briefly presents the theoretical background upon which the content of the following chapters builds. The theory presented in this chapter is divided into three parts: structural dynamics, system identification and statistical pattern recognition (SPR) for decision making.

Firstly, the basics of structural dynamics are provided, including the equations of motion for continuous and discrete systems, as well as the definitions of the impulse response function, the frequency response function and the transfer function. Subsequently, the notion of transmissibility is introduced and the pole-zero representation of systems is described. The second part of the chapter provides an overview on nonparametric and parametric system identification methods. The general model structure for system identification is presented, followed by the ARX model and the state-space representation. Furthermore, the procedure of determining the modal parameters with stochastic subspace identification (SSI) is described. The first two parts of the chapter lay the foundation for the proposed damage feature, which is introduced in Chapter 3 and is based on the migration of transmissibility function poles.

The last part of the chapter focuses on statistical pattern recognition (SPR) approaches which are applied in SHM for decision making and hypothesis testing, as well as on ensemble methods which are employed for combining the decisions of individual algorithms. The procedure of monitoring structural changes within the context of unsupervised learning is presented by defining the probability density functions for the univariate and the multivariate cases, and for defining thresholds which serve as decision boundaries. Finally, a brief overview of ensemble methods is given followed by the detailed description of adaptive boosting (AdaBoost). The last part of the chapter provides the background required for hypothesis testing for the damage feature introduced in Chapter 3, and for understanding the concept of combining decisions presented in Chapter 4.

2.1 Structural Dynamics and System Transfer Functions

2.1.1 Equation of Motion for Continuous Systems

The motion of a linear, time-invariant, mechanical system can be derived from Newton's second law by taking into account an internal restoring force and a damping force. Hence, the equation of motion of a single degree of freedom (SDOF) dynamic system in continuous time is

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = f(t) \tag{2.1}$$

with m being the system mass, c the damping constant and k the stiffness. Dividing by the mass m yields

$$\ddot{u}(t) + 2\zeta\omega_n\dot{u}(t) + {\omega_n}^2 u(t) = \frac{1}{m}f(t) \tag{2.2}$$

where $\omega_n = \sqrt{k/m}$ is the undamped natural frequency, $\zeta = c/c_{cr} = c/2\zeta\omega_n$ is the damping ratio and c_{cr} is the critical damping. When no external forces are applied, the second part of the equation is zero.

$$\ddot{u}(t) + 2\zeta\omega_n\dot{u}(t) + {\omega_n}^2 u(t) = 0 \tag{2.3}$$

The general solution of the equation of motion for free vibration is:

$$u(t) = Le^{\lambda t} \tag{2.4}$$

Substituting the general solution in Eq. 2.3 results in the so-called characteristic equation $\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0$, which yields two roots λ_1 and λ_2

$$\lambda_{1,2} = -\omega_n\zeta \pm j\omega_n\sqrt{1-\zeta^2} = -\omega_n\zeta \pm j\omega_d \tag{2.5}$$

with $\omega_d = \omega_n\sqrt{1-\zeta^2}$ being the damped natural frequency or oscillation frequency. If $\zeta > 1$, the roots are two real numbers and the system is overdamped, i.e., it does not vibrate and damping forces govern the response. If $\zeta = 1$, the solutions are two equal real roots and the system is critically damped, i.e., it does not vibrate and returns to the equilibrium faster than for the overdamped case. If $\zeta < 1$, the two roots are complex conjugate and the system is underdamped, i.e., it vibrates around its equilibrium points with decreasing amplitude.

The equation of motion of a MDOF system with N degrees of freedom is given by

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t) \tag{2.6}$$

where $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K} \in R^{NxN}$ are the symmetric mass, damping and stiffness matrices. $\mathbf{u}$, $\dot{\mathbf{u}}$, $\ddot{\mathbf{u}} \in R^{Nx1}$ are the displacement, velocity and acceleration vectors and $\mathbf{f} \in$

R^{Nx1} is the force vector. Usually, the assumption of proportional damping is made. Proportional damping is defined as a dissipative situation, where the viscous damping **C** is directly proportional to the stiffness matrix, the mass matrix or to a linear combination of both.

$$\mathbf{C} = c_1\mathbf{M} + c_2\mathbf{K} \tag{2.7}$$

When no external forces are applied the second part of Eq. 2.6 is zero. For undamped systems, the system properties can be calculated by solving the following eigenvalue problem of the form

$$(\mathbf{K} - \lambda_i^2\mathbf{M})\boldsymbol{\psi_i} = 0 \tag{2.8}$$

where λ_i are the eigenvalues and $\boldsymbol{\psi_i}$ the corresponding eigenvectors. The aforementioned eigenvalue problem is transformed into the special form $(\mathbf{A} - \lambda^2\mathbf{I}) = 0$ and the natural frequencies are obtained as the eigenvalues of matrix **A**. The eigenvectors, on the other hand, correspond to the mode shapes of each natural frequency.

Finally, the eigenvalues of a linear, damped MDOF dynamic system depend on the circular natural frequency and damping and have the same form as the eigenvalues of the SDOF system

$$\lambda_i = -\omega_{ni}\zeta_i \pm j\omega_{ni}\sqrt{1-\zeta_i^2} = -\omega_{ni}\zeta_i \pm j\omega_{di} \tag{2.9}$$

Natural frequencies and damping can be extracted from the eigenvalues as follows

$$\omega_{ni} =\mid \lambda_i \mid= 2\pi f_i \qquad , \qquad \zeta_i = -\frac{Re(\lambda_i)}{\mid \lambda_i \mid} \tag{2.10}$$

with f_i being the natural frequency in Hz.

2.1.2 Equation of Motion for Discrete Systems

Since real systems are continuous, they may be described by the continuous equation of motion. However, when the input and output signals of a system are recorded by sensors and a data acquisition unit, the obtained signals are discrete. Suppose that the input and output signals are recorded at a sequence of equally spaced times t_i, where $t_i = (i-1)\Delta t$. Δt is the sampling interval and $f_s = 1/\Delta t$ is the sampling frequency. Then, for the SDOF case, the equation of motion of Eq. 2.1 can be written as:

$$m\ddot{u}(t_i) + c\dot{u}(t_i) + ku(t_i) = f(t_i) \tag{2.11}$$

Finite difference methods are used to approximate discrete derivatives [95]. Hence, a posible way of approximating the displacement derivatives is by means of the backward difference

$$\dot{u}(t_i) \approx \frac{u(t_i) - u(t_i - \Delta t)}{\Delta t} = \frac{u_i - u_{i-1}}{\Delta t} \tag{2.12}$$

$$\ddot{u}(t_i) \approx \frac{u_{i+1} - 2u_i + u_{i+1}}{\Delta t^2} \tag{2.13}$$

Substituting the derivative approximations and rearranging yields

$$u_i = \big(2 - \frac{c\Delta t}{m} - \frac{k\Delta t^2}{m}\big)u_{i-1} + \big(\frac{c\Delta t}{m} - 1\big)u_{i-2} + \frac{\Delta t^2}{m} f_{i-1} \tag{2.14}$$

which can be rewritten as

$$u_i = a_1 u_{i-1} + a_2 u_{i-2} + b_1 f_{i-1} \tag{2.15}$$

Eq. 2.15 is the discrete time representation of the SDOF system with the constants a_1, a_2 and b_1 being defined in Eq. 2.14. Hence, it can be stated that the displacement at time i can be estimated by the current value of the input at $i-1$ and the displacement values at $i-1$ and $i-2$.

2.1.3 Impulse Response Function

The impulse response function (IRF) $h(t-\tau)$ is the response of a dynamic system to an impulse force $f(t) = \delta(t-\tau)$ applied at time τ. Any force $f(t)$ which varies arbitrarily with time can be represented by a series of infinitesimally small impulses. The response of a linear dynamic system to an impulse applied at time τ with amplitude $f(\tau)\delta\tau$ is this amplitude multiplied by the IRF

$$du(t) = [f(\tau)\delta\tau]h(t-\tau), \qquad \text{for} \qquad t > \tau \tag{2.16}$$

The system response at time t can be expressed as the summation of the responses to a series of impulses that represent the original forcing function

$$u(t) = \int_{-\infty}^{+\infty} f(\tau)h(t-\tau)\delta\tau = h(t) * f(t) \tag{2.17}$$

Eq. 2.17 is known as the convolutional integral and may be applied only for linear dynamic systems, since it is based on the superposition principle.

2.1.4 Frequency Response Function and Transfer Function

The analytical solution of the convolution integral in time domain becomes very extensive, especially when the system is subject to more complex excitation forces. Convolution in time domain corresponds to simple multiplication in frequency domain. Therefore, the Fourier transfrom is employed to transform signals into the frequency domain. An implicit assumption in the Fourier analysis is that the signal of interest is periodic. The Fourier transform of a time $u(t)$ is defined as

$$\mathcal{F}\{u(t)\} = U(\omega) = \int_{-\infty}^{+\infty} u(t)e^{-j\omega t}dt \tag{2.18}$$

Employing the Fourier transform on the IRF yields the frequency response function (FRF)

$$U(\omega) = H(\omega)F(\omega) \iff H(\omega) = \frac{U(\omega)}{F(\omega)} \tag{2.19}$$

The frequency response of a system can be graphically represented by the Bode plot, consisting of the magnitude plot and the phase plot, which show the magnitude and the phase shift of the frequency response as a function of frequency.

However, not all responses are periodic. For aperiodic signals, the Laplace transfrom can be employed. The Laplace transform of a function $u(t)$ is defined as

$$\mathcal{L}\{u(t)\} = U(s) = \int_{-\infty}^{+\infty} u(t)e^{st}dt \tag{2.20}$$

where s is a complex number. It may be observed that the Fourier transform is a subcase of the Laplace transform for $s = j\omega$. The transfer function is then defined as the ratio of the transformed response $U(s)$ and the transformed excitation $F(s)$

$$H(s) = \frac{U(s)}{F(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + ... + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + ... + a_1 s + a_0} \tag{2.21}$$

$H(s)$ is a complex function that is represented as a surface in the Laplace domain. The numerator order m is equal to the number of numerator roots, while the denominator order n corresponds to the number of denominator roots. Since the FRF is a subcase of the transfer function for $s = j\omega$, the FRF is essentially a slice of the transfer function. Factorization of the polynomials in the numerator and denominator yields

$$H(s) = \frac{\prod_1^m (s - z_i)}{\prod_1^n (s - p_i)} \tag{2.22}$$

where the zeros z_i are the roots of the numerator equation and the poles p_i are the roots of the characteristic equation in the denominator.

The Laplace transform is used to transform continuous signals from time domain to frequency domain. The equivalent transformation for discrete signals is the z-transform, which is defined as

$$\mathcal{Z}\{u[k]\} = U(z) = \sum_{-\infty}^{+\infty} u[k]z^{-k} \tag{2.23}$$

where k is an integer and z is a complex number. One of the most important properties of the z-transform is the time shifting property, according to which a time delay of i samples in time domain corresponds to multiplication by z^{-n} in the z-domain

$$u[k-i] = z^{-i}U(z) \tag{2.24}$$

The transfer function for discrete signals in z-domain can be defined accordingly to that for continuous systems as

$$H(z) = \frac{U(z)}{F(z)} \tag{2.25}$$

where $U(z)$ is the z-transform of the system response and $F(z)$ is the z-transform of the system input.

If the degree of the numerator polynomial is smaller than or equal to the degree of the denominator polynomial, then the system is proper. In this case, $H(\infty) < \infty$, which implies that the value of the transfer function is bounded to values smaller than infinity as frequency tends to infinity. On the other hand, if the degree of the numerator polynomial is smaller than that of the denominator polynomial, then the system is strictly proper and the value of the transfer function for frequency values tending to infinity is zero ($H(\infty) = 0$). It can be stated that all physical systems are strictly proper.

2.1.5 Transmissibility Function

The ratio between the spectra of two responses is defined as the transmissibility function. If the input force $f(t)$ is applied at DOF l, and $U_{il}(s)$, $U_{jl}(s)$ are two responses measured at locations i and j respectively, then the transmissibility function between these two responses is given by

$$T_{ij(l)}(s) = \frac{U_{il}(s)}{U_{jl}(s)} \tag{2.26}$$

By making use of the transfer function definition, the responses can be expressed as a function of the transfer function and the applied force. Since the excitation force appears in both the numerator and the denominator, it can be eliminated, reducing the transimissibility function to the ratio between two transfer functions

$$T_{ij(l)}(s) = \frac{H_{il}(s)F_l(s)}{H_{jl}(s)F_l(s)} = \frac{H_{il}(s)}{H_{jl}(s)} \tag{2.27}$$

As a result, transmissibility functions are independent of the amplitude of the applied force $f(t)$ but dependent on the location of force application l. Transmissibility functions are especially useful when the excitation forces of the structure are not measured or are not measurable.

2.1.6 Poles and Zeros of a System

The roots of the transfer function characteristic equation, i.e., the roots of the denominator polynomial, are called poles and correspond to the values of s in the Laplace domain and z in the z-domain for which the transfer function becomes infinite. Poles can be used to extract the natural frequencies and damping of the system. Hence, the location of system poles is defined by the system properties and is independent of the input. The roots of the transfer function numerator are called zeros, since they represent the values of s and z for which the transfer function is equal to zero. Thus, zeros correspond to the values for which the system attenuates the input. Unlike poles, the location of zeros is defined by properties which relate the system input to the system output. Poles and zeros are either real numbers or appear in complex conjugate pairs. They are essential for understanding the system behavior and can be used to design the system response.

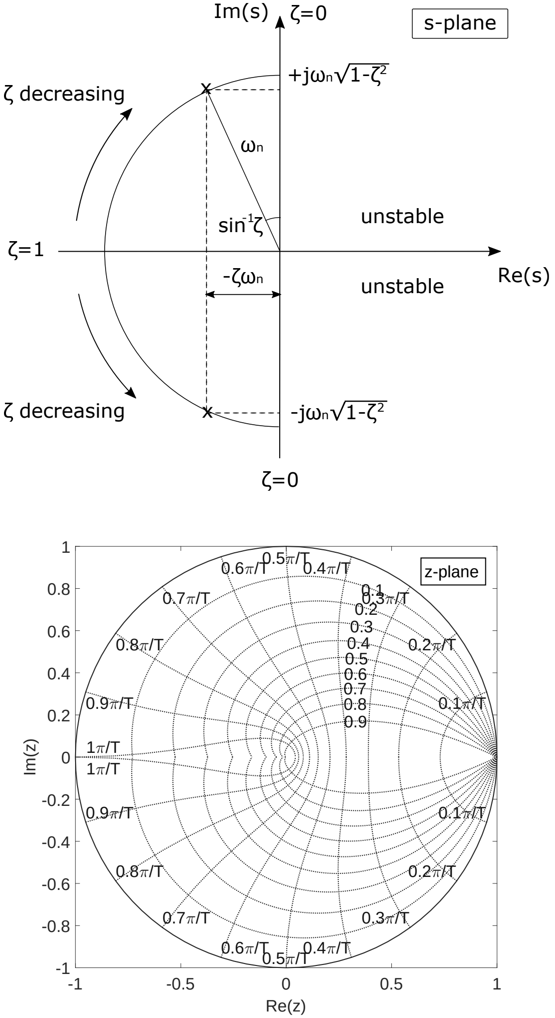
Pole-zero plots, which represent the locations of poles and zeros on the complex plane, are used to graphically represent system dynamics and provide qualitative insights into the response characteristics of a system. For continuous signals, poles and zeros are plotted upon the complex s-plane, which is shown in Figure 2.1(a). If all poles are located in the left half plane, the system is stable. Stable systems are defined as systems which have bounded output when subjected to bounded input. Complex poles in the left half plane correspond to oscillation with decreasing amplitude, whereas poles on the real axis have damping ratio equal to 1 and correspond to a critically damped system. On the other hand, poles located in the right half plane indicate that the system has unstable modes. Unstable modes are modes for which bounded input results in unbounded output.

Poles and zeros of discrete signals can be plotted upon the z-plane, which is shown in Figure 2.1(b). Systems with poles located inside the unit circle are stable, whereas systems with poles outside the unit circle have unstable modes. Figure 2.1(b) contains countour lines of constant frequency and damping, which provide visual reference to pole frequency and damping values. These contour lines are essentially a grid of natural frequencies from 0 to π in steps of $\pi/10$ and a grid of damping ratios from 0 to 1 in steps of 0.1.

Causal or nonanticipatory systems depend on past and current inputs but not on future inputs. Noncausal or anticipatory systems, on the other hand, are systems whose output not only depends on past and current inputs, but also on future inputs. If the system input can be uniquely determined by the output, the system is invertible. For invertible systems, multiplication of the system matrix by its inverse yields the unit matrix.

A linear time-invariant system is called a minimum-phase system if the system and its inverse are stable and causal. In the s-domain, minimum-phase systems have all poles and zeros in the left half plane. Analogously, a discrete time system is defined as minimum-phase if all poles and zeros lie inside the unit circle. The term "minimum-phase" is used because these systems have the least phase lag among all systems with the same magnitude response. Systems which are stable and causal but have unstable inverses are called nonminimum-phase systems. These systems are characterized by the existence of zeros in the right half plane in the s-domain and outside the unit circle in the z-domain.

The term "nonminimum-phase system" stems from the fact that these systems introduce additional phase lag, while the magnitude of the response remains the



(b)

Figure 2.1: Pole-zero plots in the s-domain (a) and the z-domain (b).

same. This phase lag is obvious when considering the step response function of a nonminimum-phase system, which exhibits initial undershoot, zero crossing or overshoot [96]. Initial undershoot occurs when the step response initially takes values which do not lie between the initial value and the asymptotic value, and then reaches this target value asymptotically. This behavior is equivalent to a response error. In terms of structural response, the structure moves first in the opposite direction of the applied load and reaches the maximum value of the step response with some delay. This structural behavior occurs, for instance, in feedback control systems and in structures subjected to forced excitation. Zero crossing refers to the situation in which the step response passes through the value of zero. Finally, overshoot refers to the situation in which the step response takes values greater than and/or smaller than the asymptotic value of the step response.

In particular, it can be shown that the number of nonminimum-phase zeros has different effects on the system response [96]. Table 2.1 summarizes the effect of nonminimum phase zeros for proper and strictly proper systems. The number of zeros and their nature (real or complex numbers) highly affects the step response of the system. More specifically, the number of zeros determines whether the step response has an initial undershoot, a zero passing or an overshoot. Here, an odd number of zeros implies that there is one real nonminimum zero, whereas an even number of zeros implies that there are only complex nonminimum zeros.

Table 2.1: Effect of nonminimum-phase zeros on the response of strictly proper and exactly proper systems. Effect distinguished into undershoot, zero crossing and overshoot.

	Initial undershoot	Zero crossing	Overshoot
Strictly proper	If and only if $H(z)$ has an odd number of positive zeros	If $H(z)$ has at least one positive zero	If $H(z) - H(\infty)$ has at least one positive zero
Exactly proper	If and only if $H(z) - H(\infty)$ has an odd number of positive zeros	If $H(z)$ has at least one positive zero	If $H(z) - H(\infty)$ has at least one positive zero

2.2 System Identification Methods

System identification deals with the problem of building mathematical models of dynamical systems based on input and output signals. System identification methods are distinguished into nonparametric and parametric. Parametric methods employ models, which have a certain structure and are associated with the determination of a set of parameters. Nonparametric methods, on the other hand, do not employ a finite-dimensional parameter vector to describe system dynamics. Instead, they employ system responses to represent the system in a graphic or informative manner.

In 1999, Ljung introduced a generalized model structure which, when simplified, can yield the most commonly used models for system identification, especially the ones belonging to the parametric family [53]. The model structure is given by

$$A(q)y(k) = \frac{B(q)}{L(q)}f(k) + \frac{C(q)}{D(q)}n(k) = H(q)f(k) + G(q)n(k) \tag{2.28}$$

where

$$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + ... + a_{n_a} q^{-n_{n_a}} \tag{2.29}$$

$$B(q) = 1 + b_1 q^{-1} + b_2 q^{-2} + ... + b_{n_b} q^{-n_{n_b}} \tag{2.30}$$

and n_a and n_b are the model orders of polynomials $A(q)$ and $B(q)$, respectively. q is a forward shift operator defined as $qy(t) = y(t+1)$. Similarly, we can define the backward shift operator $q^{-1}y(t) = y(t-1)$, which is equivalent to the shift property of the z-operator presented in section 2.1.4. Eq. 2.28 is composed of a deterministic part expressed by term $H(q)f(k)$ and a stochastic part expressed by $G(q)n(k)$. $H(q)$ is the input transfer function, since it describes the relationship between the input $f(k)$ and the output $u(k)$. The input transfer function can be decomposed into polynomials $B(q)$, $L(q)$ and $A(q)$. In the second term, $G(q)$ is the noise transfer function, which relates the white-noise $n(k)$ to the output $y(k)$. This structure constitutes the representation of all models employed for system identification, both nonparametric and parametric. The block diagram of the general structure model is shown in Figure 2.2, while Table 2.2 presents the model equations of the most common linear models, which can be expressed as subcases of the general structure model.

Nonparametric methods, which are relatively simple and computationally efficient, are applied in both time and frequency domains. Figure 2.3 offers an overview of nonparametric methods. Some examples of time domain parametric methods are the impulse response function, the step response function and approaches, which employ correlation analysis in the time domain. Frequency domain methods include, among others, sine-wave testing, correlation analysis, Fourier analysis and spectral analysis (i.e., power spectral density).

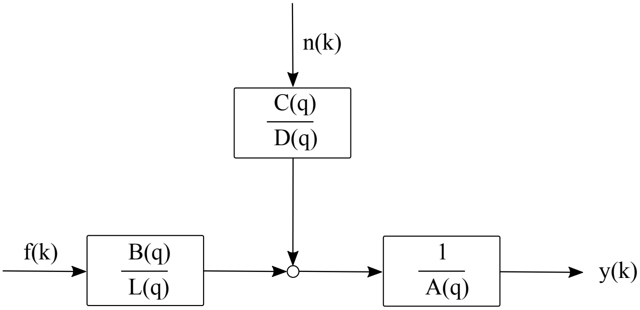


Figure 2.2: Block diagram of the general structure model.

Table 2.2: Model equations of the most common linear models belonging to time series models, equation error models and output error models.

Model type	Model name	Model equations
time series	MA	$y(k) = C(q)n(k)$
	AR	$y(k) = 1/D(q)n(k)$
	ARMA	$y(k) = C(q)/D(q)n(k)$
equation error	ARX	$y(k) = B(q)/A(q)u(k) + 1/A(q)n(k)$
	ARMAX	$y(k) = B(q)/A(q)u(k) + C(q)/A(q)n(k)$
	ARARX	$y(k) = B(q)/A(q)u(k) + 1/D(q)A(q)n(k)$
output error	OE	$y(k) = B(q)/L(q)u(k) + n(k)$
	BJ	$y(k) = B(q)/L(q)u(k) + C(q)D(q)n(k)$

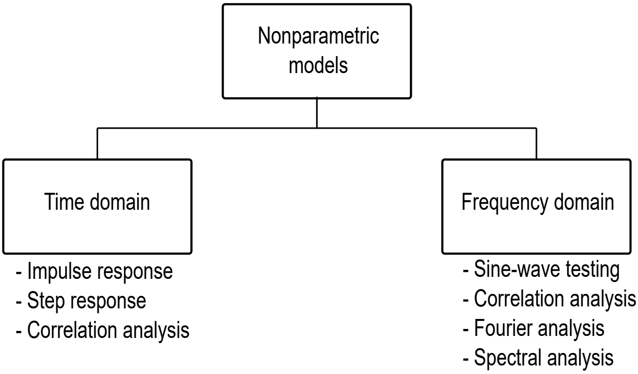


Figure 2.3: Overview of nonparametric system identification methods.

Parametric methods can de distinguished into so-called transfer function models (i.e., input/output models) and state-space models. Transfer function models are further classified into: (i) time series models, (ii) equation error models and (iii) output error models [53, 54] (see Figure 2.4). Time series models take into account only the time series of interest without considering any input. Hence, in Eq. 2.28, $f(k)$ is equal to zero and the models become fully stochastic. Autoregressive (AR), moving average (MA) and autoregressive models with moving average (ARMA) are the most well-known time series models. Equation error models are characterized by the fact that the noise term $n(k)$ does not directly influence the output $y(k)$ but instead enters the model before the $1/A(q)$ filter. Moreover, in equation error models, the term $1/A(q)$ is common in both the deterministic process model and the stochastic noise model. These model assumptions are reasonable if the noise enters the process at an early stage, so that its frequency characteristic is shaped by the process dynamics. Some of the most common equation error models are the autoregressive model with exogenous input (ARX), the autoregressive model with moving average and exogenous input (ARMAX) and the autoregressive autoregressive models with exogenous input (ARARX). Output error models are characterized by noise models which are independent of the deterministic process models, i.e., do not contain the process dynamics. Therefore, they are more suitable for systems in which noise is primarily measurement noise, which typically disturbs the output directly. The output error model and the Box-Jenkins model also belong to this family.

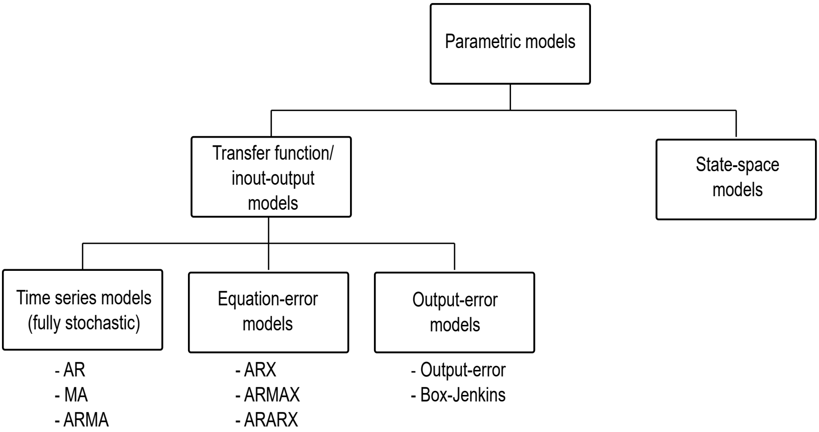


Figure 2.4: Overview of parametric system identification methods.

2.2.1 The ARX Model

In Table 2.2, it was shown that ARX models can be expressed as a subcase of the general model structure. The block diagram of this model is shown in Figure 8.1. Taking into account that the shift operator q is equivalent to the shift property of the z-operator, the ARX model equation can be written as

$$y(k) = \frac{B(z)}{A(z)}u(k) + \frac{1}{A(z)}n(k) = H(z)u(k) + G(z)n(k) \tag{2.31}$$

with

$$A(z) = 1 + a_1z^{-1} + a_2z^{-2} + ... + a_{n_a}z^{-n_{n_a}} \tag{2.32}$$

$$B(z) = 1 + b_1z^{-1} + b_2z^{-2} + ... + b_{n_z}q^{-n_{n_b}} \tag{2.33}$$

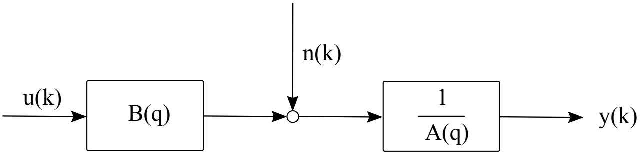


Figure 2.5: Block diagram of the ARX model.

The autoregressive part of the model is related to the transfer function between input and output ($H(z)$), as well as to the noise transfer function $G(z)$. Thus, the deterministic and the stochastic part of the ARX model possess identical denominator dynamics represented by $1/A(z)$.

An alternative expression of the ARX model is given by a linear difference equation, which weighs the past observations of the system output y and those of the system input f, to make a prediction of the system output at a given time step k

$$y(k) = \sum_{i=1}^{n_a} a_iy(k-i) + \sum_{i=1}^{n_b} b_if(k-i) \tag{2.34}$$

Coefficients a_i are the weights on the past observations of the system output, whereas coefficients b_i are the weights of the past observations of the system input. Writing Eq. 2.34 in a more analytical form yields

$$y(k) + a_1y(k-1) + ... + a_{n_a}y(k-n_a) = b_1f(k-1) + ... + b_{n_b}f(k-n_b) \tag{2.35}$$

This representation of the ARX model exhibits similarity to the most general expression of the equation of motion for discrete dynamic systems (see Eq. 2.11). This similarity proves the generality and significance of the ARX model. The least squares algorithm is used to fit the model to a set of data by identifying the values of coefficients a_i and b_i. In order to achieve that Eq. 2.35 is rewritten as

$$y(k) = -a_1 y(k-1) - ... - a_n y(k-n_a) + b_1 f(t-1) + ... + b_{n_b} f(k-n_b) \tag{2.36}$$

Subsequently, we introduce vector θ and matrix ϕ, which contain the adjustable parameters of the model and the past input and output values, respectively:

$$\theta = [a_1 \quad a_2 \quad ... \quad a_{n_a} \quad b_1 \quad ... \quad b_{n_b}]^T \tag{2.37}$$

$$\phi(k) = [-y(k-1) \quad ... \quad -y(k-n_a) \quad f(k-1) \quad ... \quad f(k-n_b)] \tag{2.38}$$

Hence, the output at an instance k can be expressed as a function of θ and ϕ

$$y(k) = \phi^T \theta = \hat{y}(k|\theta) \tag{2.39}$$

where $\hat{y}(k|\theta)$ is the estimate of the system output and is written in this way to emphasize that the calculation of $y(k)$ from past data depends on the parameters in θ. The estimate of θ is then given by

$$\hat{\theta} = (\phi^T \phi)^{-1} \phi^T y \tag{2.40}$$

2.2.2 State-Space Representation

State-space models are an alternative way of representing dynamic systems. They can be derived either from the differential equation in time domain or by the transfer function representation in the Laplace domain. Therefore, the auxiliary vector

$$x(t) = \begin{bmatrix} u(t) \\ \dot{u}(t) \end{bmatrix} \in R^{2N\text{x}1} \tag{2.41}$$

is introduced, where x is the state vector, u the displacement and N the number of DOFs. Hence, the second order equation of motion for a MDOF dynamic system in continuous time (see Eq. 2.6) can be simplified in the following pair of first order equations

$$\dot{\mathbf{x}}(t) = \mathbf{A_c x}(t) + \mathbf{B_c f}(t) + \mathbf{w}(t) \tag{2.42a}$$

$$\mathbf{y}(t) = \mathbf{C_c x}(t) + \mathbf{D_c f}(t) + \mathbf{v}(t) \tag{2.42b}$$

which describes the relationship between the input signal $f(t)$, the output signal $y(t)$ and the noise signals $w(t)$ and $v(t)$. Noise terms $w(t)$ and $v(t)$ represent process noise and measurement noise respectively. Both terms are assumed to be zero-mean, stationary, white-noise vectors. $A_c \in R^{2N\text{x}2N}$ is the system matrix, $B_c \in R^{2N\text{x}n_{in}}$ the input matrix, $C_c \in R^{n_{ch}\text{x}2N}$ the output matrix and $D_c \in R^{n_{ch}\text{x}n_{in}}$ the direct transition matrix, with n_{in} being the number of inputs and n_{ch} the number of output channels. The system and input matrices can be expressed as a funtion of the system properties

$$\mathbf{A_c} = \begin{bmatrix} 0 & I \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \tag{2.43}$$

$$\mathbf{B_c} = \begin{bmatrix} 0 \\ \mathbf{M}^{-1} \end{bmatrix} \tag{2.44}$$

Subsequently, the system transfer function of Eq. 2.21 can be written in terms of system matrices

$$H(s) = \mathbf{C_c}(s\mathbf{I} - \mathbf{A_c})^{-1}\mathbf{B_c} + \mathbf{D_c} \tag{2.45}$$

Hence, the properties of the system are derived by solving the following eigenvalue problem

$$det(s\mathbf{I} - \mathbf{A_c}) = 0 \tag{2.46}$$

In discrete time, the state-space equation is given by

$$\mathbf{x}_{k+1} = \mathbf{A_d}\mathbf{x}_k + \mathbf{B_d}\mathbf{f}_k + \mathbf{w_k} \tag{2.47a}$$

$$\mathbf{y}_k = \mathbf{C_d}\mathbf{x}_k + \mathbf{D_d}\mathbf{f}_k + \mathbf{v_k} \tag{2.47b}$$

while the transfer function results from Eq. 2.45 by replacing s with the z-opertaor.

2.2.3 SSI for the Identification of Modal Parameters

Data-driven SSI is a common time-domain identification technique used to extract the modal parameters, i.e., modal frequencies, damping ratios and mode shapes. Data-driven SSI was introduced in 1996 by Overschee and de Moor as an alternative to the covariance-driven SSI [50].

In discrete time, the system response is represented by the data matrix, which is composed of n_{ch} channels and n_t data points

$$\mathbf{Y} = [\mathbf{y}_1 \quad \mathbf{y}_2 \quad \dots \quad \mathbf{y}_{n_t}] = \begin{bmatrix} \mathbf{y}(1) \\ \mathbf{y}(2) \\ \vdots \\ \mathbf{y}(n_{ch}) \end{bmatrix} \in \mathbb{R}^{n_{ch} \times n_t} \tag{2.48}$$

The block Hankel matrix $\mathbf{Y}_h \in \mathbb{R}^{2 \cdot n_{ch} \cdot i \times n_t - 2i}$ is assembled by $2i$ data blocks with $n_t - 2i$ samples of the output matrix

$$\mathbf{Y}_{h(0,2i-1)} = \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{j-1} \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_j \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{i-1} & \mathbf{y}_i & \cdots & \mathbf{y}_{i+j-2} \\ \mathbf{y}_i & \mathbf{y}_{i+1} & \cdots & \mathbf{y}_{i+j-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{2i-2} & \mathbf{y}_{2}i & \cdots & \mathbf{y}_{2i+j-3} \\ \mathbf{y}_{2i-1} & \mathbf{y}_{2i+1} & \cdots & \mathbf{y}_{2i+j-2} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{hp} \\ \mathbf{Y}_{hf} \end{bmatrix} \tag{2.49}$$

The upper half part of the Hankel matrix is denoted by $\mathbf{Y_{hp}}$ and contains past row spaces, whereas the lower half part is denoted by $\mathbf{Y_{fp}}$ and contains future row spaces. The projection of the future row spaces onto the past row spaces defines the projection matrix

$$E(\mathbf{Y}_{hf}|\mathbf{Y}_{hp}) = \mathbf{Y}_{hf}\mathbf{Y}_{hp}^T(\mathbf{Y}_{hp}\mathbf{Y}_{hf}^T)^{-1}\mathbf{Y}_{hp} \tag{2.50}$$

The projection matrix is then decomposed using singular value decomposition (SVD) with the resulting matrices **U** and **V** containing the singular vectors and the diagonal of matrix **S** containing the singular values.

$$E(\mathbf{Y}_{hf}|\mathbf{Y}_{hp}) = \mathbf{USV}^T \tag{2.51}$$

These matrices are used for the definition of the observability and controllability matrices as follows:

$$\hat{\mathbf{\Gamma}}_\mathbf{i} = \mathbf{US}^{1/2} \tag{2.52a}$$

$$\hat{\mathbf{O}}_i = \mathbf{S}^{1/2}V^T \tag{2.52b}$$

System matrix $\mathbf{A}_d$ is obtained by the lower blocks of $\hat{\mathbf{\Gamma}}_i$. Finally, solving the eigenvalue problem

$$(\mathbf{A}_c - \lambda_i I)\psi_\mathbf{i} = 0 \tag{2.53}$$

yields the eigenvalues λ_i and eigenvectors $\psi_\mathbf{i}$, which are given from Eq. 2.9 and can be used to determine the circular frequencies and the corresponding damping ratios as in Eq. 2.10.

The order n of the data-driven SSI model is equal to the number of block rows $2i$ multiplied by the number of channels n_{ch}. However, when calculating the model solutions for a model order n, the model is also solved for all lower model orders. Hence, data-driven SSI allows for the fast calculation of a series of model orders, the solutions of which can be plotted on stabilization diagrams. Depending on the model order, a number of solutions is calculated, which can be either physical or mathematical. The triangulation-based extraction of modal parameters (TEMP) serves for distinguishing between physical and mathematical solutions and for the automatic identification of stable paths in the stabilization diagram[17]. For this purpose, Delaunay triangulation of the solutions in the frequency-damping plane is performed and the solutions are reduced based on the following criteria:

1. mean phase divergence within a single eigenvector,
2. relative frequency difference between two solutions in %,
3. relative damping difference between two solutions in %,
4. modal assurance criterion (MAC) value between two solutions and
5. minimum number of model orders participating in a path object.

2.3 Statistical Pattern Recognition for Decision Making

Section 1.3.6 provided a brief overview of pattern recognition and statistical pattern recognition (SPR) approaches employed in SHM. Originating from the fields of machine learning and statistics, pattern recognition can be distinguished into: (i) SPR, (ii) neural network-based pattern recognition and (iii) syntactic or structural pattern recognition, with the first two types being applied to SHM. As shown in Figure 1.3, SPR algorithms can be further categorized depending on the available data and the type of machine learning (supervised or unsupervised). The three categories of SPR are: (i) novelty detection algorithms, (ii) classification algorithms and (iii) regression algorithms. Novelty detection algorithms are used in unsupervised machine learning for two-class problems. The most common case comprises training with healthy data and assessing current data, which can be labeled either "healthy" or "damaged". Classification algorithms use supervised machine learning and can be employed in two-class or multi-class problems. Finally, regression algorithms are used in supervised machine learning and deliver continuous outputs.

The present work makes use of novelty detection algorithms based on the probability density function of the healthy datasets which are considered in training. In the training phase, healthy data is used to calculate the probability density function (PDF) of one or more CPs. In testing, new datasets are assessed with respect to whether they belong to the healthy state or come from the damaged state of the structure. For one-dimensional CPs, univariate probabilistic models are used. On the other hand, the bivariate PDF is employed for two-dimensional CPs and the multivariate PDF is used for CPs of higher dimensions. Therefore, this section introduces the univariate and multivariate PDFs, and describes the definition of decision boundaries for hypothesis testing. Subsequently, receiver operating characteristic (ROC) curves are described. ROC curves allow for the overall assessment of a damage feature with respect to the detection performance. Furthermore, the basic definitions of ensemble methods are presented, followed by a detailed description of AdaBoost.

2.3.1 Hypothesis Testing for Novelty Detection

2.3.1.1 Univariate Damage Features

The probability density function (PDF) of the normally distributed variable $\mathrm{x} \in \mathcal{X}$ can be described by the mean value μ and the standard deviation σ or the variance σ^2 as

$$p(\mathrm{x}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-1/2((\mathrm{x}-\mu)^2/\sigma^2)} \tag{2.54}$$

where

$$\mu = E[\mathrm{x}] = \sum_{\mathrm{x}\in\mathcal{X}} \mathrm{x}p(\mathrm{x}) \tag{2.55}$$

$$\sigma^2 = Var[\mathrm{x}] = E[(\mathrm{x}-\mu)^2] = \sum_{\mathrm{x}\in\mathcal{X}} (\mathrm{x}-\mu)^2 p(\mathrm{x}) \tag{2.56}$$

Normally distributed data tend to cluster around the mean. Numerically, the probabilities obey

$$P[|\mathrm{x}-\mu| \leq \sigma] \simeq 0.68 \tag{2.57a}$$

$$P[|\mathrm{x}-\mu| \leq 2\sigma] \simeq 0.95 \tag{2.57b}$$

$$P[|\mathrm{x}-\mu| \leq 3\sigma] \simeq 0.997 \tag{2.57c}$$

A natural measure of the distance from x to the mean μ is the distance $|\mathrm{x}-\mu|$ measured in units of standard deviation

$$r = \frac{|\mathrm{x}-\mu|}{\sigma} \tag{2.58}$$

A standardized normal variable $r_n = (\mathrm{x}-\mu)/\sigma$ has a zero mean and unit standard deviation, so the normalized PDF can be written as

$$p(r_n) = \frac{1}{\sqrt{2\pi}} e^{-r_n^2/2} \tag{2.59}$$

The error function is a finite integral of the Gaussian distribution defined as

$$erf(r_n) = \frac{2}{\sqrt{\pi}} \int_0^{r_n} e^{-\mathrm{x}^2} dx \tag{2.60}$$

The error function corresponds to the area under the standardized Gaussian PDF between $-\sqrt{2}r_n$ and $\sqrt{2}r_n$. That is, if x is a standardized Gaussian random variable, $Pr[|\mathrm{x}| \leq \sqrt{2}r_n] = erf(r_n)$. Thus, the complementary probability $1 - erf(u)$ is the probability that a sample is chosen, with $|\mathrm{x}| > \sqrt{2}r_n$.

In this sense, for each CP, a PDF can be fitted over all data instances considered in training. For a normal distribution, decision boundaries can be defined by the confidence interval upper and lower limits. In terms of statistical process control

(SPC), these decision boundaries are called upper control limit (UCL) and lower control limit (LCL). The confidence interval is an estimated range of the PDF $[\alpha, \beta]$, which is likely to include a value, for instance a CP, and which is calculated from a given set of sampled data. The confidence level $1 - a$ gives the confidence or the probability $P(W) \in [0, 1]$ that a CP lies within the confidence interval, so that

$$P(W) = P(\alpha < X_i < \beta) = 1 - a \tag{2.61}$$

where a is the significance level. The confidence interval can be conceived as a range of values on the horizontal axis of the PDF function, while the confidence level is the area under the PDF curve within this interval. The upper and lower limits of the two-sided confidence interval $(1 - a)$ for the mean value of a CP, μ_X, are then defined as

$$LCL_n = \mu_X - t_{\frac{a}{2}\%, n_s - 1} \sigma_X \tag{2.62a}$$

$$UCL_n = \mu_X + t_{\frac{a}{2}\%, n_s - 1} \sigma_X \tag{2.62b}$$

taking into account the variance of the sampled normal distribution

$$\sigma_X^2 = \frac{1}{n_s - 1} \sum (\mathrm{x}_i - \bar{\mathrm{x}})^2 \tag{2.63}$$

and the $(1 - \frac{a}{2})^{th}$ percentile of the t-distribution with $n_s - 1$ DOF given by $t_{\frac{a}{2}, n_s - 1}$. t-distributions are formed by taking many samples from a normal distribution and the values of $t_{\frac{a}{2}, n_s - 1}$ can be found in tables. In many applications, these factors are equal to three, setting the control limits to distance $\pm 3\sigma$ from the mean value.

If the PDF of a variable is unknown, percentiles can be used for selecting the decision boundaries as a function of the significance level a, where $a =$ $(1 - \text{confidence level})$. In this case, the center line (CL) is the 50^{th} percentile and the LCL and UCL are percentiles $X_{a/2\%}$ and $X_{1-a/2\%}$ respectively.

$$LCL_d = X_{a/2\%} \tag{2.64a}$$

$$UCL_d = X_{1-a/2\%} \tag{2.64b}$$

2.3.1.2 Multivariate Damage Features

Multivariate features are composed of η normally distributed random variables, each with its own mean value and variance, so that $p_{\mathrm{x}_i}(\mathrm{x}_i) \sim N(\mu_i, \sigma_i^2)$. For this case, the mean value becomes a vector containing the mean values of the d individual variables, and the covariance becomes a matrix $\in R^{\eta \mathrm{x} \eta}$ [97]. The mean value is

$$\boldsymbol{\mu} = E[\mathbf{x}] = \int_{-\infty}^{+\infty} \mathbf{x} p(\mathbf{x}) d\mathbf{x} \tag{2.65}$$

the covariance matrix is

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{12}^2 & \cdots & \sigma_{1\eta}^2 \\ \sigma_{21}^2 & \sigma_2^2 & \cdots & \sigma_{2\eta}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{\eta 1}^2 & \sigma_{\eta 2}^2 & \cdots & \sigma_\eta^2 \end{bmatrix} \tag{2.66}$$

$$\text{with} \qquad \mathbf{\Sigma} = E[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T] = \int_{-\infty}^{+\infty} (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})p(\mathbf{x})\mathbf{dx} \tag{2.67}$$

and the PDF for a multivariate normal distribution with a mean $\boldsymbol{\mu}$ and a covariance matrix $\mathbf{\Sigma}$ is given by

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\eta/2}|\mathbf{\Sigma}|^{1/2}} exp\Big[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\Big] \tag{2.68}$$

Multivariate normal data tend to cluster about the mean vector μ, forming an ellipsoidally shaped cloud, whose principal axes are the eigenvectors of the covariance matrix. The measure of the distance from $\mathbf{x}$ to the mean $\boldsymbol{\mu}$ is provided by the Mahalanobis squared distance (MSD) from $\mathbf{x}$ to $\boldsymbol{\mu}$

$$r_M^2 = (\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \tag{2.69}$$

As a result, $\mathbf{r}_n = \mathbf{\Sigma}^{-1/2}(\mathbf{x}-\boldsymbol{\mu})$ is the analogous expression for $r_n = (\mathrm{x}-\mu)/\sigma$.

Contour plots of constant density are ellipsoids defined by $\mathbf{x}$, so that

$$(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = \kappa^2 \tag{2.70}$$

These ellipsoids are centered at $\boldsymbol{\mu}$ with axes $\pm\kappa\sqrt{\boldsymbol{\lambda}_i^m}\mathbf{e}_i^m$, where $\mathbf{\Sigma}\mathbf{e}_i^m = \lambda_i^m\mathbf{e}_i^m$, for $i = 1, 2, ..., \eta$. For the η-dimensional normal distribution, the MSD follows a χ^2-distribution, so that $\kappa^2 = \chi_p^2(a)$ can be defined, with $\chi_p^2(a)$ being the 100^{th} percentile of the χ^2-distribution. This choice leads to contours, which contain $(1-a)$ of the probability.

The equivalent of the confidence interval for multivariate distributions is the confidence level. For the multivariate case confidence regions can also be defined; these serve as boundaries for hypothesis testing. It can be shown that the confidence level of a multivariate CP is given by

$$P(n_s(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}(\mathbf{x}-\boldsymbol{\mu})) \leq \frac{(n_s-1)\eta}{n_s-\eta}F_{\eta,n_s-\eta}(a) = 1-a \tag{2.71}$$

where $F_{\eta,n_s-\eta}(a)$ denotes critical values of the F-distribution, which can be found in tables. The F-distribution is employed when dealing with statistics formed by ratios of variance estimates and is used extensively in the analysis of variance applications.

2.3.2 Receiver Operator Characteristic Curves

If the calculated values lie within the LCL and UCL for the univariate case or within the ellipsoid in the multivariate case, the structure is assumed to be healthy. Otherwise, it is assumed that there is a deviation from the healthy state and, consequently, that the structure is damaged. In statistical hypothesis testing, the first outcome is the null hypothesis H_0 and the second outcome is the alternative hypothesis H_1. The CPs are evaluated with respect to the correct hypothesis, that is, with respect to the actual structure state. Finally, hypothesis testing delivers four possible cases, which include two error types:

1. H_{00}: H_0 is accepted, while it is true (Correct inference - true negative or tn)
2. H_{10}: H_0 is rejected, while it is true (Type error I - false positive or fp)
3. H_{01}: H_0 is accepted, while it is false (Type error II - false negative or fn)
4. H_{11}: H_0 is rejected, while it is false (Correct inference - true positive or tp)

The percentage of fp and tp detections can be used as a measure for assessing detection performance. These counts strongly depend on the chosen value for a, since a defines the value range for the CP values linked to a healthy condition. If these parameters are analyzed for different a-values, receiver operating characteristic (ROC) curves can be built. ROCs illustrate the true positive rate (TP or H_{11}) against the false positive rate (FP or H_{10}) for different a-values, providing a measure for the overall detection performance of a CP. The TP and FP rates are defined as

$$FP_i = \frac{fp_i}{fp_i + tn_i} \tag{2.72}$$

$$FN_i = \frac{fn_i}{fn_i + tp_i} \tag{2.73}$$

with $TP = 100 - FN$ and $TN = 100 - FP$. The closer an ROC point is to point (TP, FP)=(100,0), i.e., the upper left corner of the ROC plot, the better the detection performance is.

One metric for evaluating the performance of HT based on the ROC curve is the Youden index (J), which describes the maximum distance between the ROC curve and the diagonal (Equation 2.74). Another performance indicator is the minimum distance to the optimum located at the upper left corner (Equation 2.75)

$$J = max\{H_{11}(a) - H_{10}(a)\} \tag{2.74}$$

$$\bar{d} = min\{\sqrt{H_{01}(a)^2 + H_{10}(a)^2}\} \tag{2.75}$$

Finally, a scalar index regarding the performance of HT (and thus, the detection performance of a CP) is the area under curve (AuC), which is the area under the ROC curve. AuC values $\mathcal{A}$ take values between 0 and 1. AuC values greater than 0.9 ($\mathcal{A} > 0.9$) indicate high accuracy, AuC values between 0.9 and 0.7 ($0.7 < \mathcal{A} < 0.9$) indicate moderate accuracy, and values between 0.7 and 0.5 ($0.5 < \mathcal{A} < 0.7$) indicate low accuracy.

2.3.3 Ensemble Methods for the Combination of Classifiers

Any rule or algorithm which implements classification is called a classifier. Ensemble methods, which are also called classifier combination techniques, address the problem of decision fusion; that is, combining decisions of different rules or classifiers. The objective of ensemble methods is to improve performance by combining the outputs of several classifiers. The most important classifier combination methods are the following: product rule, sum rule, min-max-median, majority vote, borda count, combination of class predictors, stacked general, mixture of experts, bagging, boosting and model averaging [82]. In all cases, the classifiers which are intended to be combined are called component classifiers.

Classifier combination schemes can be categorized based on different characteristics, such as:

1. Space of classifiers. The component classifiers may be designed in the same or different feature spaces.
2. Combiner level. The combiner may operate at the raw data level, the feature level or the decision level.
3. Degree of training. The combiner may be fixed or trainable, i.e., adjustable based on the knowledge of the training data.
4. Form of individual component classifiers. The considered classifiers may either be of the same or of dissimilar form (for instance, all classifiers are neural networks or all classifiers are decision trees).
5. Structure of the multiple classifier system. The structure may be parallel (all component classifiers are passed into the combiner at once), serial (component classifiers are invoked sequentially) or hierarchical (component classifiers are combined in a hierarchy).
6. Optimization scheme. The classifiers and the combiner may be optimized separately or simultaneously.

In particular, boosting algorithms aim at improving the performance of learning algorithms by combining multiple base classifiers. These classifiers are usually inaccurate or perform poorly. Base classifiers are alternatively referred to as weak classifiers or learners, with the term "weak" emphasizing their mediocre performance and the term "base" connoting their use as building blocks [83]. The terms "weak" and "base" do not refer to the component classifiers, but to the ones constructed by using the training set of the boosting algorithm. However, the weak classifiers are built based on the decisions of the component classifiers[1]. The component classifiers considered in boosting are designed in a common feature space. The combiner operates at the feature level and is trainable. All component classifiers have the same form, while the

[1] In literature, the term "weak classifier" is often used to describe the component classifier. In order to avoid confusion, in this work, the term "weak" refers to the classifiers selected during the training phase of AdaBoost, which are dependent on the component classifier output.

structure of the multiple classifier system is serial. Finally, the constituent classifiers of the combining scheme (i.e., the weak classifiers) are optimized.

2.3.4 Adaptive Boosting (AdaBoost)

AdaBoost is a widely used boosting algorithm with numerous applications in classification problems. It trains a sequence of base classifiers in rounds (or iterative calls) based on the decisions of the component classifiers. Weights are assigned to the base classifiers depending on their performance on the training set. These weights define the importance of the base classifiers, i.e., the contribution of the base classifiers to the final strong classifier, which is built as a weighted sum. Each base classifier is weaker than the ensemble classifier. Therefore, base classifiers are also called weak classifiers. Adaptive boosting is based on the assumption that each of the weak hypotheses is at least slightly more accurate than random guessing. For two-class problems, this implies that each error rate should be at most 1/2. Given this assumption, which is termed the weak learning condition, and sufficient data, AdaBoost can produce a final hypothesis with arbitrarily small generalization error [83].

The training set used for AdaBoost contains CP values which correspond to *tp* and *tn* detections, as well as their decisions. In each iteration, one data point of the training set is selected as a training point and the rest constitute test points. Subsequently, weights are assigned to the training points, with misclassified points receiving higher weights. These weights determine the probability that a point is selected for a training set for the subsequent weak classifier. Thereby, emphasis is placed on the data points which are more difficult to be correctly classified. The advantage of AdaBoost lies in this iterative manner of learning. In testing, the final classification decision for a test point is based on a discriminant function, which is the weighted sum of the binary outputs given by the cascade of classifiers obtained in training.

In the training phase of AdaBoost, n_s sets of training examples are used in the algorithm as input

$$(\boldsymbol{\chi}_1, y_1^h), ..., (\boldsymbol{\chi}_{n_s}, y_{n_s}^h) \tag{2.76}$$

where $\boldsymbol{\chi}_i \in X$ are n_d-dimensional features and $y_i^h \in \{-1, +1\}$ are the corresponding labels or hypotheses of the component classifiers. At the first iteration, the assumption is made that the error rates of the n_s training examples are uniformly distributed. Thus, the distribution is initialized, so that all training examples have the same error rate:

$$\mathcal{D}_1(i) = 1/n_s \text{ for } i = 1, ..., n_s \tag{2.77}$$

The error rate provided by distribution $\mathcal{D}(i)$ corresponds to the importance of classifying a point correctly, as well as the probability that a data point is chosen as a training point in the next iteration. For $t = 1, 2..., T$, where T is the user-defined number of AdaBoost rounds and subsequently the number of base classifiers which form the final strong classifier, the following steps are repeated:

1. A training point is chosen based on distribution $\mathcal{D}_t$ and the weak learner $\mathcal{C}_t$ is trained.
2. The test data points are evaluated by C_t and its weak hypothesis h_t is calculated.
3. The training error of $\mathcal{C}_t$ on X is calculated using $\mathcal{D}_t(i)$.
4. The weight of hypothesis h_t to the overall classifier a_t is calculated as

$$a_t = \frac{1}{2} ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \tag{2.78}$$

   It is desirable that the classifier have a small error rate. Thus, small training errors ϵ_t result in a higher contribution to the ensemble classifier.

5. The distribution weights are updated using Equation 2.79b, where N_t is a normalization factor chosen, so that the new distribution $\mathcal{D}_{t+1}$ sums up to 1 ($\sum_{i=1}^{m} \mathcal{D}_{t+1}(i) = 1$).

$$\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i) \exp(-a_t y_i^h h_t(\boldsymbol{\chi}_i))}{N_t} \tag{2.79a}$$

$$= \begin{cases} e^{-a_t}, & \text{if } h_t(\boldsymbol{\chi}_i) = y_i^h \\ e^{a_t}, & \text{if } h_t(\boldsymbol{\chi}_i) \neq y_i^h \end{cases} \tag{2.79b}$$

The final hypothesis for the evaluation of a data point x is calculated as the sign of the weighted sum of all the weak hypotheses:

$$\mathcal{H}(\boldsymbol{\chi}) = sign \left(\sum_{t=1}^{T} a_t h_t(\boldsymbol{\chi}) \right) \tag{2.80}$$

In terms of optimization, AdaBoost is a procedure of minimizing the following exponential loss function:

$$\frac{1}{n_s} \sum_{i=1}^{n_s} exp(-y_i^h F_l(\boldsymbol{\chi}_i)) \tag{2.81}$$

2.4 The Three-Tier SHM Framework

The SHM framework and its three constituent parts are shown in Figure 2.6. The framework comprises the following tiers: (i) application of machine learning (ML) algorithms for data clustering based on EOCs, (ii) extraction of damage-sensitive features, which are also referred to as CPs and (iii) hypothesis testing (HT). All framework tiers are performed during both training and testing phases.

In tier 1, datasets are normalized and assigned to clusters according to the prevailing EOCs and, in tier 2, CPs are calculated for each cluster. CPs are features extracted from the measured system response data, which are observed for detecting structural changes. They are damage sensitive but, at the same time, can be sensitive to changes in the prevailing EOCs. CPs can be either absolute or relative. Absolute CPs are calculated by single datasets. The most common absolute CPs are the natural frequencies. Relative or reference-based CPs are formed as residues by comparing characteristics of a dataset with a reference value calculated from another dataset and instance. Two examples of reference-based CPs are residues which derive from vector autoregressive (VAR) models and residues from the stochastic subspace identification (SSI) approach. Finally, in HT (tier 3) different probabilistic models are employed and decision thresholds are defined based on confidence intervals. The CPs are evaluated assuming that they follow a Gaussian distribution and a discrete distribution within each cluster.

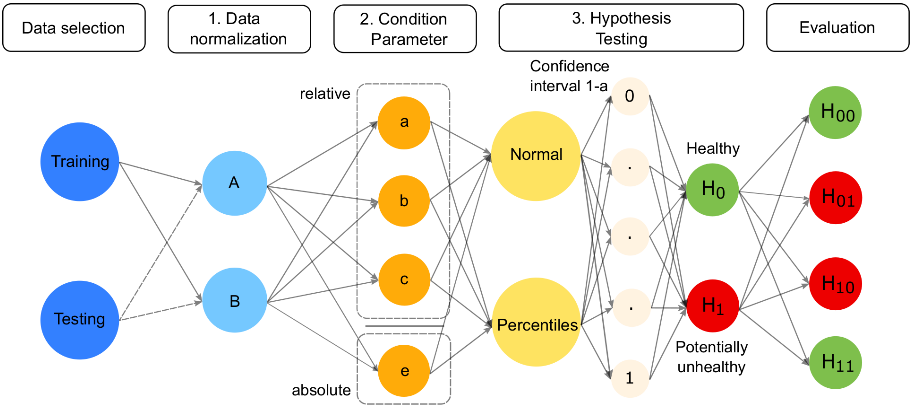


Figure 2.6: The three-tier SHM framework.

Chapter 3

SHM using Transmissibility Function Poles

In the previous chapter, it was shown that the ARX model is a powerful tool for modeling dynamic systems. The significance of this model becomes obvious when comparing its structure to the equation of motion for discrete dynamic systems. ARX models belong to the family of equation error models, which, unlike the time series models (AR, MA, ARMA etc.), are not fully stochastic. Thereby, they are not based on the assumption of stationarity, which refers to time series with constant statistical moments over time. The assumption of stationarity is often violated, especially for structures which are surrounded by a varying environment, such as wind turbines. Therefore, the implementation of the ARX model can be advantageous, since it involves solely the assumption that the noise vector is white noise. Another powerful tool for the representation of a system are transfer function poles and zeros, which not only serve for the extraction of the modal parameters but also offer a valuable insight into the system's response.

The objective of the present chapter is to introduce a new damage feature, which is based on observing the poles of an output-only ARX model. Output-only ARX models are especially useful when the system excitation is not available. This modified model constitutes a transmissibility function (TF) model, the solutions of which can be used to detect structural changes. However, due to the replacement of the exogenous input with a response of the structure, the solutions of the denominator do not correspond to system poles. Therefore, these solutions are referred to as transmissibility function (TF) poles. TF poles can be conceived as a two-dimensional feature which encapsulates frequency and damping information. SHM using TF pole migration can be performed either in supervised or in unsupervised mode. The supervised mode requires datasets from both the healthy state and the damaged state of the structure. On the other hand, in unsupervised mode, TF poles of the healthy state are used as reference for the assessment of TF poles of current datasets. In the current thesis, both cases are considered but the focus lies on exploring pole migration in unsupervised mode, since this mode is more appropriate for implementing SHM in real structures.

The chapter begins with the presentation of the output-only ARX model.

This model is then expressed by means of a state-space representation, in order to show that TF poles are essentially system zeros. Furthermore, the concept of observing TF pole migration in supervised mode is outlined and some simple metrics for the quantification of TF pole migration are defined. Finally, the concept is described for the unsupervised case within the context of a three-tier SHM framework.

3.1 The Output-Only ARX Model

The ARX model is expressed by a linear difference equation, which consists of n_b weighted observations of the system input $f(k)$ and n_a weighted observations of the system output $y(k)$ (see section 2.2.1). For a discrete system, the ARX model is written as

$$y(k) + a_1 y(k-1) + ... + a_{n_a} y(k-n_a) = b_1 f(k-1) + ... + b_{n_b} f(k-n_b) \tag{3.1}$$

The model is composed of an autoregressive part expressed by coefficients a_i, and an exogenous part expressed by coefficients b_i.

When the system input is not available, the exogenous part of the equation can be built by using response signals instead of input signals, yielding thereby an output-only model. If we use response signal y_p as model output and response signal y_q as model input, Eq. 3.1 can be rewritten as

$$y_p(k) + a_1 y_p(k-1) + ... + a_{n_a} y_p(k-n_a) = b_1 y_q(k-1) + ... + b_{n_b} y_q(k-n_b) \tag{3.2}$$

The transfer function of the output-only ARX model is essentially a transmissibility function, since it is equal to the ratio of two responses

$$H(z) = \frac{Y_p(z)}{Y_q(z)} = \frac{b_1 z^{-1} + ...b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + ...a_{n_a} z^{-n_a}} \tag{3.3}$$

Hence, the output-only ARX model constitutes a transmissibility function model. If n_{ch} response signals are available, one of them can be used as a model input and the rest $n_{ch} - 1$ as the model outputs. As a result, n_{ch} different output-only MDOF ARX models emerge, each describing the different ratios of the structural responses.

Section 1.3.4.2 presented the state of the art on CPs which originate from parametric models and are based on transmissibility measures. Therein, some works which employ the aforementioned output-only ARX model were presented [60, 61, 62]. For instance, in [60], the model error of the output-only ARX model and its standard deviation are used for damage detection, while the model error of the different MDOF models is used for damage localization.

3.2 Transmissibility Function Poles

The modification of the ARX model input transformed the transfer function into a transmissibility function (TF). Therefore, the solutions of the TF denominator do not correspond to the system poles. In order to gain insight regarding the nature of these solutions, the classical ARX model is expressed as a state-space model and the transfer function is considered. The state equation, which describes a state-space representation of a linear system in discrete time, is given by the following pair of equations

$$\mathbf{x}_{k+1} = \mathbf{A_d}\mathbf{x}_k + \mathbf{B_d}\mathbf{f}_k + \mathbf{w_k} \tag{3.4a}$$

$$\mathbf{y}_k = \mathbf{C_d}\mathbf{x}_k + \mathbf{D_d}\mathbf{f}_k + \mathbf{v_k} \tag{3.4b}$$

where system matrix $A_d \in R^{2N\text{x}2N}$, input matrix $B_d \in R^{2N\text{x}n_{in}}$, output matrix $C_d \in R^{n_{ch}\text{x}2N}$ and direct transition matrix $D_d \in R^{n_{ch}\text{x}n_{in}}$. The number of inputs is denoted by n_{in} and the number of outputs by n_{ch}.

In order to obtain the equivalent state-space model of the classical ARX model, a signal-flow graph is drawn, as shown in Figure 3.1. Signal-flow diagrams visually represent the interactions between system components. Block diagrams are a subcase of simulation diagrams which consist of unidirectional, operational blocks and are adequate for the description of the relationship between the input and the output of a system. Signal-flow diagrams, on the other hand, are capable of describing more complex interrelationships between the system components. They are able to simulate the signal flow within a process and can be used in computer tools, such as Simulink [98]. Figure 3.1 contains two summing points, while operator z^{-1} indicates time shift of one sample (i.e., $x(k+1)z^{-1} = x(k)$).

The outputs of the shift registers ($x(k+1)$, $x(k+2)$, ..., $x(k+n_a)$) are chosen as state variables. Subsequently, the state equation can be written in the controllable canonical form

$$\begin{Bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+n_a) \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & -a_1 & -a_2 & \cdots & -a_{n_a} \end{bmatrix} \begin{Bmatrix} x(k) \\ x(k+1) \\ \vdots \\ x(k+n_a-1) \end{Bmatrix} + B_d f_k \tag{3.5}$$

where the system parameters a_1 to a_{n_a} are arranged across the last row of system matrix A_d. Eigenvalue decomposition of matrix A_d yields the system properties, i.e., the natural frequencies and corresponding damping ratios. Subsequently, the transfer function of the state-space model can be written as

$$\begin{aligned} H(z) =& \frac{Y(z)}{F(z)} = \left|C_d(zI - A_d)^{-1}B_d + D_d\right| \\ =& \frac{C_d adj(zI - A_d)^{-1}B_d + D_d det(zI - A_d)}{det(zI - A_d)} \end{aligned} \tag{3.6}$$

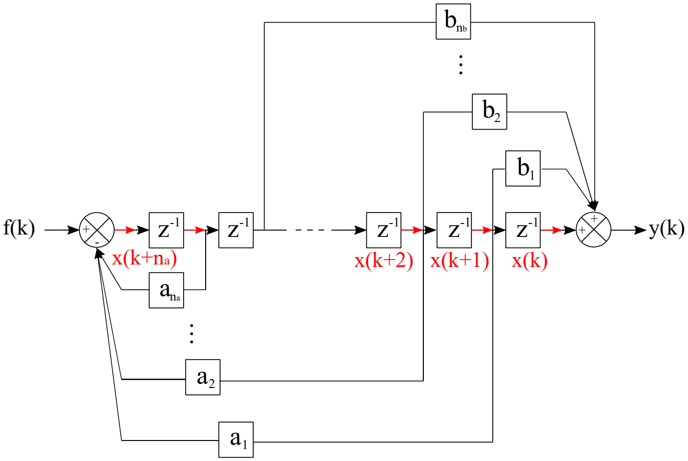


Figure 3.1: Signal-flow diagram with the set of differential equations describing the ARX model. The output of each shift register (z^{-1}) is chosen as a state variable in the state-space representation.

Usually, the direct transition matrix D_d is equal to zero simplifying the transfer function, which can be written as a fraction with the denominator D_H and the numerator N_H

$$H(z) = \frac{C_d adj(zI - A_d)^{-1} B_d}{det(zI - A_d)} = \frac{N_H}{D_H} \tag{3.7}$$

The characteristic equation of the denominator contains the system matrix A_d, while the numerator is expressed in terms of the system matrix A_d, the input matrix B_d and the output matrix C_d. In Eq. 3.6 and 3.7, $adj(\cdot)$ denotes the adjugate and $det(\cdot)$ the determinant of a matrix.

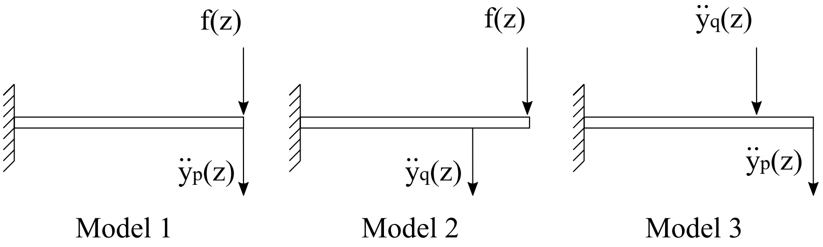


Figure 3.2: Model 1 using force $f(z)$ as input and acceleration $\ddot{y_p}(z)$ as output. Model 2 using force $f(z)$ as input and acceleration $\ddot{y_q}(z)$ as output. Model 3 using acceleration $\ddot{y_q}(z)$ as input and acceleration $\ddot{y_p}(z)$ as output.

Let models 1, 2 and 3 be three ARX models. Model 1 receives force $f(z)$ as input and outputs acceleration $\ddot{y_p}(z)$. Model 2 receives force $f(z)$ as input and outputs acceleration $\ddot{y_q}(z)$, and model 3 receives acceleration $\ddot{y_q}(z)$ as input and outputs acceleration $\ddot{y_p}(z)$ (see Figure 3.2). The transfer functions of the three models are

$$H_1(z) = \frac{\ddot{Y}_p(z)}{F(z)}, \qquad H_2(z) = \frac{\ddot{Y}_q(z)}{F(z)}, \qquad H_3(z) = \frac{\ddot{Y}_p(z)}{\ddot{Y}_q(z)} \tag{3.8}$$

Hence, H_3 can be written as a function of H_1 and H_2.

$$H_3(z) = \frac{\ddot{Y}_p(z)}{\ddot{Y}_q(z)} = \frac{H_1(z)F(z)}{H_2(z)F(z)} = \frac{\frac{N_{H_1}}{D_{H_1}}}{\frac{N_{H_2}}{D_{H_2}}} = \frac{N_{H_1}}{N_{H_2}} \tag{3.9}$$

The transfer function denominator contains only the system matrix and, hence, the system properties (see Eq. 3.7). Since the system is the same for models 1 to 3, the denominator can be eliminated. Finally, H_3 is written as the ratio of the numerator polynomial of model 2 (N_{H_2}) to the numerator polynomial of model 1 (N_{H_1}). As a result, the characteristic equation solutions of H_3 do not yield the system poles, but the zeros of model 1.

3.3 SHM based on Transmissibility Function Pole Migration

The objective of this chapter is to introduce a novel damage feature which is based on the migration of TF poles. In the previous section, it was shown that the TF denominator polynomial does not yield the system poles as in the case of the transfer function. Instead, the solutions of the TF denominator are zeros of another model containing the input. Hence, TF pole locations are defined by the properties which relate the system input to the system output and are dependent on the location of the input force application but not on its amplitude.

TF pole migration may be investigated either in supervised mode or in unsupervised mode. The supervised mode refers to two-class problems, where a number of datasets from the baseline and the damaged state are considered in the training phase and current datasets are assigned to one of these classes in the testing phase. This mode is useful for detecting and quantifying damage based on TF pole migration. The unsupervised mode, on the other hand, comprises the assessment of the current datasets based only on the knowledge of the baseline state, and is more appropriate for implementing TF pole migration within the context of an SHM framework that is used for monitoring structures in operation. While the observation of TF poles in supervised mode is briefly outlined, the present thesis focuses on exploring TF pole migration in unsupervised mode.

3.3.1 Supervised Mode

Suppose that datasets are available from several states of the structure, which include the healthy state, the damaged state and other states of interest. In this case, TF poles from all the states can be plotted upon the z-plane in order to investigate the effect of damage or varying EOCs on the TF pole locations. Furthermore, several criteria can be used to classify the TF poles of new datasets as healthy or damaged.

Firstly, the appropriate model order n_a of the output-only ARX model has to be determined. For this purpose, models of different orders are built and the mean squared errors (MSEs) of their predictions are calculated for a training dataset. As the model order increases, the MSE will be exponentially reduced. The calculated models, which were fitted to the training dataset, are then used to predict the system's response for a validation dataset. The MSE of the validation dataset decreases exponentially, but starts to increase at higher model orders for which the model starts to overfit the training dataset by capturing its noise processes. The MSEs of a training dataset and a validation dataset are depicted in Figure 3.3 for a series of model orders. A range of optimal model orders can be obtained by comparing the MSE curve of the training set with that of the validation set. More specifically, the optimal model order lies around the model orders for which the MSE of the training set keeps dropping, while the MSE of the validation set starts to increase. The order of the exogenous part n_b can be set equal to the model order

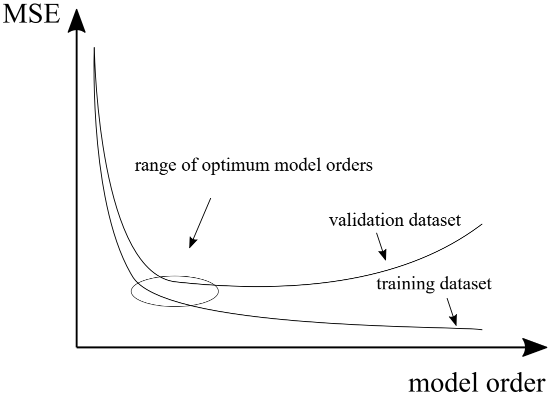


Figure 3.3: Model mean squared error vs model order. Interval of orders with reducing MSE of the training set and increasing MSE of the validation set define the range of optimal model orders.

n_a, because, for the output-only ARX model, there is no delay between the input and the output (since they are both structural responses).

After the appropriate model order has been selected, the TF poles of the baseline state are calculated and plotted upon the z-plane. The number of resulting TF poles is dependent on the selected model order. TF poles usually appear in conjugate pairs but the solutions might occasionally contain poles located on the real axis. These TF poles are ignored because their locations are not consistent for different datasets and because they are expected to create complications at the subsequent step of clustering. Conjugate and real TF poles which lie outside the unit circle constitute nonminimum-phase zeros. As mentioned in section 2.1.6, nonminimum-phase zeros imply that the system response is subjected to undershoot, overshoot or zero crossings and reaches the target response value with some delay. Therefore, these TF poles are not considered in the further analysis, either.

If more datasets from the baseline state are analyzed, their TF poles can be superimposed on the same plane. TF poles with similar frequency and damping content aggregate in groups. Subsequently, clustering algorithms may be employed to build clusters of TF poles characterized by similar frequency and damping. In this work, two clustering algorithms are used in order to cluster TF poles on the z-plane: k-means clustering and affinity propagation. k-means clustering is a partitional clustering algorithm, i.e., it requires a predefined number of clusters. This number of clusters is maintained throughout the clustering process. The algorithm aims at finding an assignment of data points to clusters, so that the sum of the squared distances between each data point and its closest cluster centroid is minimum [22]. The implementation of k-means for the clustering of TF poles of different states allows for the definition of a certain number of clusters which remains the same for all states. Affinity propagation (AP), on the other hand, is a clustering algorithm which uses measures of similarity between pairs of data points and exchanges information between them in order to build clusters. These measures of similarity are dubbed preferences. The algorithm considers all data points as potential exemplars and, therefore, does not require a predefined number of clusters [37]. The choice between k-means clustering and AP is made, depending on their efficiency on the exanimed data.

A certain number of clusters with TF poles and their centroids are obtained, regardless of the clustering algorithm employed. These clusters can be described by a normal bivariate distribution by calculating the mean value and standard deviation. For instance, level curves, which are essentially ellipses, can be plotted using the mean value ($\boldsymbol{\mu}$) and three times the standard deviation of each pole cluster ($\mathbf{3}\boldsymbol{\sigma}$). The same procedure can be followed for the datasets of the damaged state.

Figure 3.4 shows an example of level curves for a set of healthy and damaged TF pole clusters. The TF poles are obtained from the modal test data of a 34 m wind turbine rotor blade. TF poles of the healthy state are represented by the blue points and the cyan level curves, while TF poles of the damaged state are shown by the green points and the red ellipses. In this example, the locations of all five TF pole clusters can be simultaneously observed for damage identification. The clusters of the healthy and damaged state can be exploited (i) to describe and quantify pole cluster migration due to structural changes and (ii) to evaluate the TF poles of new datasets by classifying them as healthy or damaged.

In order to describe and quantify pole migration, three simple metrics can be defined: (i) the Euclidean distance d between the cluster means of two states (μ_h for the healthy state and μ_c for the current state), (ii) the difference between the standard deviation both in the major and in the minor direction of the ellipse ($\Delta\sigma_{major} = \sigma_{1c} - \sigma_{1h}$ and $\Delta\sigma_{minor} = \sigma_{2c} - \sigma_{2h}$) and (iii) the rotation of the ellipse ($\Delta\theta = \theta_c - \theta_h$), i.e., the difference between θ_c and θ_h, which are the angles between the horizontal axis and the ellipse major direction for the current state and the healthy state, respectively. The components, based on which these metrics are composed, are shown in Figure 3.5 for one pole cluster.

Furthermore, the TF pole clusters of the healthy and damaged states serve as the training data of a classification algorithm. Hence, TF poles of new datasets are evaluated based on whether they belong to the healthy or the damaged pole cluster. For this purpose, more complex metrics are required. For instance, in [67], transfer function poles are assessed by a damage index which is built as a weighted sum of three indices. More specifically, these indices are based on three different types of classifiers: (i) a classifier based on the nearest cluster mean, (ii) the perceptron classifier and (iii) the mean separation distance criterion. However, achieving a high detection accuracy and quantification in the supervised mode is beyond the scope of the present thesis, which focuses on employing TF pole migration for detecting structural changes in unsupervised mode.

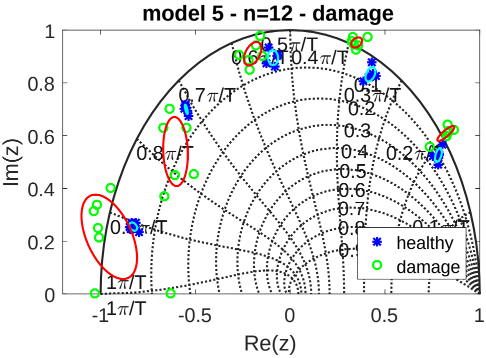


Figure 3.4: Healthy and damaged TF poles obtained from the modal test data of a rotor blade plotted upon the z-plane. TF poles clustered with k-means clustering. Cyan ellipses describe the TF poles of healthy datasets and red ellipses describe the poles of damaged datastes.

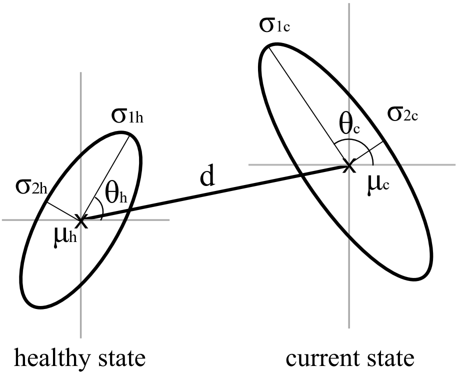


Figure 3.5: Ellipses describing the TF poles of the baseline state (subscript h) and the current state (subscript c). The figure depicts: the cluster means (μ_h and μ_c), the standard deviations in the principle axis direction (σ_{1h} and σ_{2h}) and in the perpendicular direction (σ_{1c} and σ_{2c}), the angle of the ellipse principle direction with respect to the horizontal axis (θ_h and θ_c) and the Euclidean distance d between the two means.

3.3.2 Unsupervised Mode

In most practical implementations of SHM, only undamaged datasets are available. Therefore, most SHM concepts focus on employing SHM methods which assess datasets based exclusively on information gained from the undamaged structure. In this section, the concept of detecting structural changes based on the locations of TF poles is presented in the context of unsupervised machine learning. More specifically, in the current thesis, TF pole migration is investigated in the context of the three-tier SHM framework, which was presented in section 2.4. [36].

Firstly, the appropriate model order is selected by comparing the MSE of a training dataset to that of a validation dataset, as described in the previous section. In the training phase, datasets from different healthy system states are assigned to clusters depending on their EOCs (tier 1-ML). Data clusters can either be predefined by the user (manual clustering) or generated by automatic clustering methods. CPs are calculated for the data contained in each cluster (tier 2-CP). In HT, probabilistic models are set up for the CPs of each cluster obtained in ML based on the EOCs (tier 3-HT).

In the testing phase, new incoming datasets are assigned to the existing clusters (tier 1-ML) and their CPs are calculated (tier 2-CP). Relative CPs are calculated using information from the reference datasets defined during the training phase. In this work, the reference values of each cluster occur as the mean values of the corresponding quantities coming from all the datasets contained in each cluster. Finally, in HT, the extracted CPs are compared to cluster statistics and are evaluated within different probabilistic models for different confidence intervals (tier 3-HT).

At this point, a new CP, which makes use of the TF pole locations, is defined and assessed using the three-tier SHM framework. It was previously shown that TF poles are essentially system zeros, and since the location of system zeros is affected by the input location, data clustering is necessary. Hence, in the training phase, datasets are grouped in clusters either by manual clustering or by automatic clustering algorithms in order to compensate the effect of prevailing EOCs. In tier 2, output-only ARX models are built for datasets obtained from different healthy states, i.e., datasets from the healthy structure acquired under different EOCs. The TF poles of all datasets are calculated and superimposed on the z-plane for each cluster obtained during tier 1. Subsequently, for each clustering case of tier 1, TF poles of the healthy states are clustered upon the z-plane using a clustering algorithm, such as k-means clustering or AP. Thus, several TF pole clusters emerge. Each TF pole cluster is described by the mean value $\boldsymbol{\mu}$ and the $3\boldsymbol{\sigma}$ level curves of a bivariate normal distribution. Alternatively, the level curves can represent other multiples of the standard deviation $\boldsymbol{\sigma}$ or the 100^{th} percentile of the bivarite distribution. The two variables of the normal distribution are, in this case, the real part and the imaginary part of the TF poles.

In HT, decision boundaries can be defined for a series of confidence regions. As shown in section 2.3.1.2, the Mahalanobis squared distances (MSDs) of the points of a bivariate normal distribution follow a χ^2-distribution. Consequently, the decision boundaries of the new CP correspond to contours of constant probability $(1 - a)$, which are obtained as the percentiles of the χ^2-distribution (see Eq. 2.71). This selection of decision boundaries results in accurate detections, assuming that the TF poles of each cluster are normally distributed. If this assumption of Gaussianity does not hold, other decision boundaries have to be defined. To be more specific, the MSD between each point in a TF pole cluster and the distribution of the TF pole cluster is calculated. The percentiles of the MSDs of the TF poles contained in a cluster can be used as thresholds for classifying the TF poles of new datasets based on their MSD value.

In testing, new datasets are evaluated one at a time. First, the datasets are assigned to the existing clusters of tier 1, depending on the EOCs they correspond to. Subsequently, the output-only ARX model is built for each test dataset and the TF poles are calculated (tier 2). The TF poles of the new datasets are assigned to their closest TF pole clusters and the MSDs between the TF poles and the corresponding clusters are calculated. In HT, the datasets are assessed for two cases: (i) for the assumption that TF poles within a cluster are normally distributed and (ii) for the case that the TF pole distribution deviates from Gaussianity. In the first case, the MSDs of the current datasets are assessed with respect to the percentiles of the χ^2-distribution. In the latter case, the MSDs of the current dataset are assessed with respect to the percentiles of the MSDs within the clusters obtained in training. Hence, the MSD can be conceived as a new CP, which indicates whether the location of the TF poles has changed significantly. Figure 3.6 depicts the concept of evaluating the TF pole of a dataset by calculating the MSD to its closest cluster. The current dataset can be assessed for a series of confidence regions, which are equivalent to confidence intervals for univariate distributions.

This procedure is followed for all TF poles of a dataset, i.e., for n_{cl} TF poles, where n_{cl} is the number of pole clusters, and yields n_{cl} damage indices, which can either be conceived as seperate CPs or can be used to build one CP. In the latter case, the CP hypothesis is (i) the majority vote of the decisions derived from HT of the individual pole clusters and (ii) H_1, i.e., a positive detection is returned, if at least one of the TF pole clusters indicates changes of the TF pole location.

Several output-only ARX models can be built, when several structural responses are available. Each of these uses different inputs and outputs and, therefore, expresses another relation between input and output. As a result, the TF poles of each model are different. The number of TF poles, on the other hand, depend on the order of the ARX model. Hence, the number of TF poles are the same regardless of the clustering case used in tier 1 of the SHM framework. The clustering case, however, affects the cluster characteristics, i.e., the cluster mean and standard deviation.

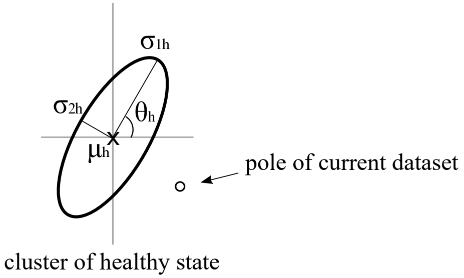


Figure 3.6: Ellipse describing the TF poles of the healthy state, and TF pole of the current dataset, which is evaluated with respect to the distribution of the healthy state. The figure depicts: the cluster means (μ_h), the standard deviations in the principle axis direction (σ_{1h}) and in the perpendicular direction (σ_{2h}) and the angle of the ellipse principle direction with respect to the horizontal axis (θ_h)

3.4 Summary

This chapter introduced a new CP for monitoring vibrating structures by observing the TF pole locations of an output-only ARX model. The output-only ARX model, which constitutes a transmissibility function model, was introduced. The output-only ARX model is a powerful tool for modeling dynamic systems, because it requires no measurements of excitation forces and it is not based on any significant assumptions, such as the assumptions of stationarity and white-noise excitation. A state-space representation was used to show that TF poles do not correspond to the transfer function poles but are essentially zeros. In general, transfer function zeros have not been given the same attention as system poles. However, zeros are equally important, since they represent frequency and damping values for which the system attenuates the input. The advantage of exploiting zeros for SHM lies in the fact that they encapsulate both frequency and damping information, and that they do not contain the input. Instead, they are defined by properties which define the relationship between system input and system output and are only dependent on the input location.

The concept of SHM based on TF pole migration was presented for the cases of supervised and unsupervised machine learning, with the focus lying on the latter. Regardless of the type of machine learning algorithms, the appropriate model order has to be selected. This is achieved by fitting ARX models of different orders to a training dataset and using these models to calculate the model MSE for a training dataset and a validation dataset. Comparison between the MSEs of the training dataset and the validation dataset yields a range of appropriate orders, which can be considered for selecting the optimum model order. In chapter 5, it will be shown that the criterion for selecting the optimal model order within this range is the aggregation of poles and the consistency of their values when evaluating several datasets of the same state. After the optimal model order has been selected, the output-only ARX model can be built and the TF pole clusters of several datasets can be calculated. Depending on the selected model order, several TF poles are obtained for each dataset. Real TF poles and TF poles corresponding to nonminimum-phase zeros are eliminated before proceeding with the further analysis.

The implementation of TF pole migration for detecting damage in the context of supervised machine learning is not in the scope of this thesis, so it was only briefly mentioned. In supervised mode, datasets from both healthy and damaged states can be used (i) to investigate and quantify TF pole migration due to structural changes and (ii) to assess the state of new datasets by classifying their TF poles as "healthy" or "damaged". When several datasets from each state are evaluated, clustering algorithms are implemented to create TF pole clusters, which contain poles with similar properties. TF pole clusters can be described by a bivariate normal distribution and, more specifically, by their mean value μ and the three times their standard deviation σ. The cluster mean μ and the contour representing 3σ are represented by ellipses. The investigation and quantification of pole migration is achieved by defining simple metrics. In this chapter, three metrics were defined based on (i) the distance between the means of the healthy cluster and the means of the damaged cluster, (ii) the changes in standard deviation in the major and minor directions of the ellipses and (iii) the rotation of the ellipses. However, in order to assess new datasets and assign them to either the "healthy" or the "damaged" clusters, more complex metrics are required.

The main contribution of this chapter lies in the conception of a novel CP based on TF pole migration, and in integrating it into an SHM framework. The framework employs unsupervised machine learning and consists of the following tiers: (i) application of machine learning algorithms for data clustering based on EOCs, (ii) extraction of CPs and (iii) hypothesis testing (HT). In unsupervised mode, datasets from the healthy structure are used as a reference for the assessment of new datasets. More specifically, the implementation of the new CP along the three tiers of the SHM framework can be summarized as follows:
In the training phase,

- Tier 1. Training datasets from different states of the pristine structure are clustered based on the EOCs they belong to.
- Tier 2. The model order is selected and output-only ARX models are fitted to the training datasets. The TF poles are calculated and superimposed upon the z-plane for each cluster of tier 1. After eliminating the real TF poles and the TF poles corresponding to nonminimum-phase zeros, the remaining TF poles are clustered upon the z-plane (for each cluster of tier 1). The MSD between each TF pole and the cluster it belongs to is calculated.
- Tier 3. Decision boundaries are defined for a series of confidence regions:
(i) For the case that TF poles contained in a cluster follow a bivariate normal distribution. The decision boundaries are the percentiles of the χ^2-distribution.

(ii) For the opposite case. The decision boundaries are the percentiles of the MSD values of the poles contained within each cluster.

In testing,

- Tier 1. Test datasets are assigned to the clusters obtained during the training phase depending on the prevailing EOCs.
- Tier 2. Output-only ARX models are fitted to the test datasets and the TF poles are calculated. After eliminating the real TF poles and the TF poles corresponding to nonminimum-phase zeros, each TF pole is assigned to its adjacent cluster for further comparison. The MSD of each TF pole cluster to the cluster it was assigned to is calculated.
- Tier 3. The MSD values are assessed (i) with respect to the percentiles of the χ^2-distribution, when assuming that the TF poles follow a bivariate normal distribution, (ii) with respect to the percentiles of the MSDs calculated in the training phase, when assuming that the TF poles are not normally distributed.

Depending on the selected model order, several pole clusters are obtained. All datasets usually yield the same number of TF poles, which have to be evaluated by means of comparison to the healthy pole clusters. Consequently, observing pole migration results in a number of damage indices which is equal to the number of pole clusters. These damage indices can be either conceived as seperate CPs or can be used to build one CP. The decision of this compact CP (i) can result in a detection if at least one of the pole clusters yields a detection and (ii) can be the majority vote of the decisions derived from HT on the individual pole clusters.

Chapter 4

Improving Detection Performance by Combining Damage Feature Decisions with AdaBoost

Inferences about the structural state, either at a local or at a global level, are provided by implementing different SHM algorithms and approaches. Within each SHM approach there is also a plethora of damage-sensitive features which can be extracted and analyzed for detecting structural changes. However, it is unlikely that a damage feature is capable of detecting all damage scenarios, whereas each damage feature exhibits different sensitivity to damage or environmental settings. Parallel observation of several damage features derived from the same method (or, better yet, from different methods) is therefore beneficial and offers a better overview of the structural state. After damage-sensitive features have been extracted, certain thresholds, decision boundaries or rules are implemented in order to ascertain whether the structure has been subjected to changes.

Any rule or algorithm which implements classification constitutes a classifier. In this sense, the process of assessing the state of a structure by implementing a certain rule can be conceived as the implementation of a classifier. While the combination of features and their decisions has been studied intensively in many fields, such as that of image processing, it has not been given much attention in the context of SHM. Ensemble methods, which aim at improving the performance of learning algorithms, are employed in order to levage a combination of feature decisions. In particular, boosting algorithms constitute a category of ensemble methods which aim at improving the performance of component learning algorithms by combining multiple base classifiers.

AdaBoost is a boosting algorithm characterized by an adaptive manner of learning which places focus on data points that are hard to classify correctly. In AdaBoost, component classifiers are combined in order to improve their performance. In the context of SHM, component classifiers correspond to the classifiers used during a monitoring process, including the definition of decision boundaries or the implementation of a decision rule on the extracted CPs. AdaBoost uses the decisions of these component classifiers to build weak (or base) classifiers, which compose the strong classifier. In the current work, the term "component classifier" is used for the classifiers which are obtained from monitoring processes and are intended to be combined, whereas the term "weak" or "base classifier" is used for those emerging during AdaBoost. It has to be noted that each round of AdaBoost produces a weak classifiers based on the decisions of the component classifiers.

In this chapter, AdaBoost is suggested for combining the decisions of different damage features. The objective of this combination is to improve detection performance. First, the applicability conditions of AdaBoost are discussed, focusing on the nature of features which may be combined and on the weak learning condition. The feature space upon which the component classifiers were designed is decisive for their combinability using AdaBoost. The weak learning condition, on the other hand, sets an upper limit on the detection performance of the component classifiers, which may be further used in boosting.

Furthermore, the concept of integrating AdaBoost with SHM concepts is outlined based on the three-tier SHM framework presented in [36]. The main contribution of this chapter is the integration of AdaBoost with the three-tier SHM framework. Several framework realizations can be obtained by considering different settings along the three tiers of the framework. Each realization requires another classifier, while the classifier of a realization is obtained by implementing decision boundaries on the distributions of the calculated CP values. This classifier is a component classifier which can be used to train AdaBoost. Special attention is given to the description of the interface between the two algorithms, as well as to the notions of the component classifier and the base or weak classifier within this conjunction.

This chapter also addresses some important aspects regarding the implementation of AdaBoost, such as (i) the effect of overlapping CP distributions, (ii) the selection of training data when few datasets are available and (iii) the need for cross-validation. Especially the effect of overlapping CP distributions is described in detail, since it turns out to be a decisive factor for boosting performance. Finally, metrics for the assessment of boosting performance are defined.

While the presented concept is discussed for vibration-based SHM methods, it is also applicable to other SHM approaches, such as acoustic emission and wave propagation, as long as the applicability conditions are satisfied.

4.1 Applicability Conditions of AdaBoost

4.1.1 Space of Damage Features

As all boosting algorithms, AdaBoost has the following attributes:

- The component classifiers have to be designed on the same feature space (i.e., the features combined should have the same dimensions).
- The combiner is trainable and operates at the feature level.
- All component classifiers have the same form.
- The structure of the multiple classifier is serial.

- The part of the combining scheme which is optimized comprises the constituent or weak classifiers.

Consequently, AdaBoost is capable of combining damage features with the same dimensions, i.e., one-dimensional CPs with one-dimensional CPs (CPs described by only one numerical value), two-dimensional CPs with two-dimensional CPs (CPs described by two numerical values), etc. It is possible, for instance, to combine a one-dimensional CP derived from the residues of a VAR model with a one-dimensional CP derived from the residues of the SSI approach. On the other hand, it is not possible to combine a one-dimensional CP derived from the residues of a VAR model with the two-dimensional CP (TF pole positions are complex numbers) introduced in chapter 3.

4.1.2 Weak Learning Condition

A central assumption of the study of boosting is the weak learning assumption. The weak learning assumption states that component learners produce a hypothesis which is at least slightly better than random guessing. This weak hypothesis and the error rate refer to the examples used for testing these argorithms. For two-class problems, this implies that each component classifier used for boosting should have an error rate below 0.5, where the error rate (i.e., the ratio of false detections to the total number of tested datasets n_s and, subsequently, to the total number of detections) is:

$$\epsilon = \frac{fn + fp}{n_s} = \frac{fn + fp}{fp + fn + tp + tn} \tag{4.1}$$

As a result, the error rate of each component classifier ϵ_t should be at most $1/2 - \gamma$ for any $\gamma > 0$, resulting in the following probability:

$$\mathbf{Pr}_{i\ D}[h(x_i) \neq y_i] <= \frac{1}{2} - \gamma \tag{4.2}$$

The weak learning condition is useful for assessing classifiers and it can be employed as a criterion for excluding the use of a classifier which performs worse than random guessing.

4.2 Integrating AdaBoost in an SHM Framework

The contribution of this chapter lies in introducing the integration of AdaBoost with SHM concepts. In this section, AdaBoost is integrated with the three-tier SHM framework presented in section 1.3.1. This is achieved by training the AdaBoost classifier with the results of framework classifiers. The objective of this integration is to boost the framework performance by combining information from framework classifiers of different realizations. The resulting strong classifier is capable of classifying values of different CPs which might be based on different theoretical principles. The coupling of the three-tier SHM framework with AdaBoost is depicted in Figure 4.1.

The three-tier SHM framework is illustrated in Figure 4.2. After data clustering has been performed based on the prevailing EOCs, various CPs are calculated. Therefore, CP values are highly dependent on the selected data clustering settings. Thereafter, different probabilistic models and confidence intervals are used in hypothesis testing to return a decision regarding the structural state. Several framework realizations may be obtained by considering (i) different clustering cases (i.e., automatic clustering algorithms or manual clustering), (ii) different CPs and (iii) different distributions, based on which the decision boundaries are defined (i.e., Gaussian distribution or discrete distribution). When performing analyses for various confidence intervals, these realizations can be observed as a function of the confidence interval. One possible realization of the SHM framework is indicated in Figure 4.2 by the dashed arrows which connect clustering case A (tier 1), the first relative CP (tier 2), the normal distribution, one specific value of confidence interval and the output H_0 (tier 3). Each realization requires another classifier. Each combination of CP and HT, including a specific confidence interval, generates a classifier of the SHM framework. Due to the framework structure, CPs are a function of data clustering performed on tier 1. That is, two clustering cases result in different CP values even when observing the same CP. Therefore, the classifier does not involve tier 1.

The error rates of the framework classifiers have to be calculated before proceeding with boosting in order to test whether the weak learning assumption is satisfied. According to the weak learning assumption, only the SHM framework classifiers which have an error rate smaller than 0.5 can be further considered for boosting. The error rate limit set by the weak learning assumption can also be used as a criterion to eliminate framework realizations with detection performances worse than random guessing. The error rate of a classifier is the ratio of false detections (fp and fn) to the total number of datasets (see Eq. 4.1). The results of SHM processes for different confidence intervals are usually presented in receiver operating characteristic curves (ROCs), which depict the performance in terms of true positive (TP) and false positive (FP) rates for a series of confidence interval values, providing thereby a more holistic image of the detection performance. It could be mistakenly considered that the $x = y$ line on the ROC plot represents error rates equal to 0.5. In order to avoid confusion, it has to be emphasized that the error rate is expressed as a function of the fp and fn detections and not as a function of FP and FN rates, which are given

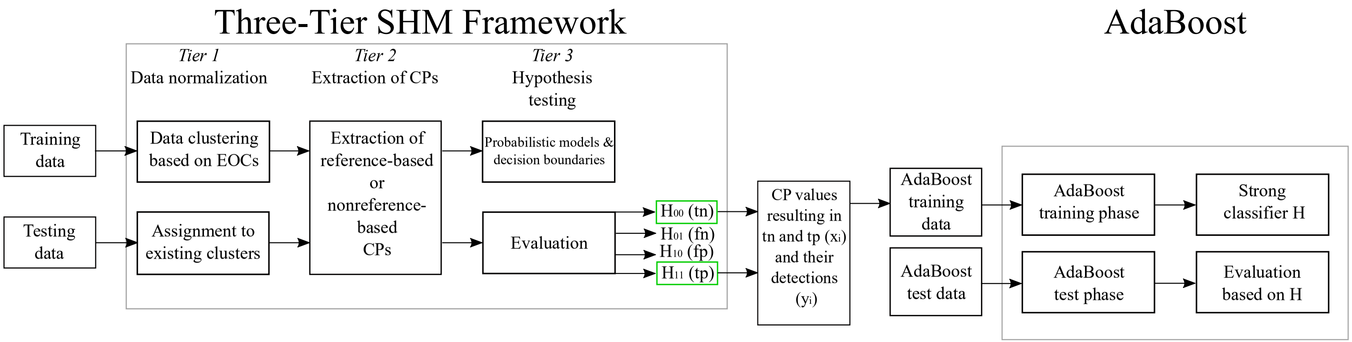


Figure 4.1: Coupling of the three-tier SHM framework with AdaBoost.

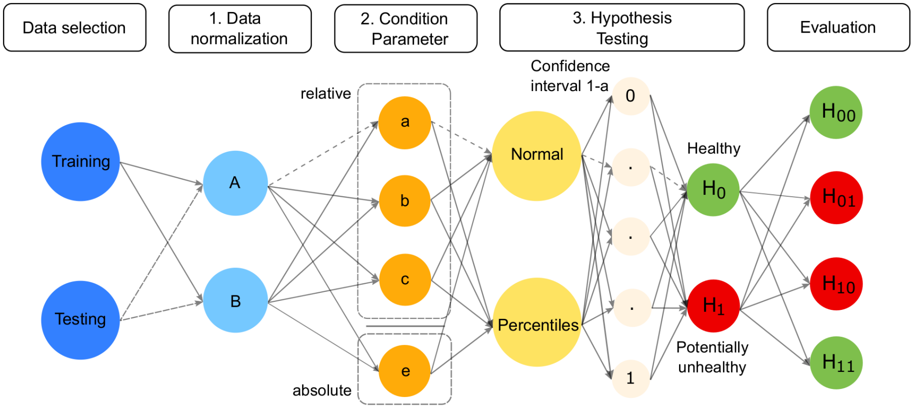


Figure 4.2: The three-tier SHM framework. One possible realization of the SHM framework is indicated by the path of dashed arrows connecting clustering case 1 (tier 1), the first relative CP (tier 2), the normal distribution in HT, one value of confidence interval and the output H_0 (tier 3). Each realization results in another classifier. The classifier of a realization is defined by implementing decision boundaries on the distribution of the calculated CPs.

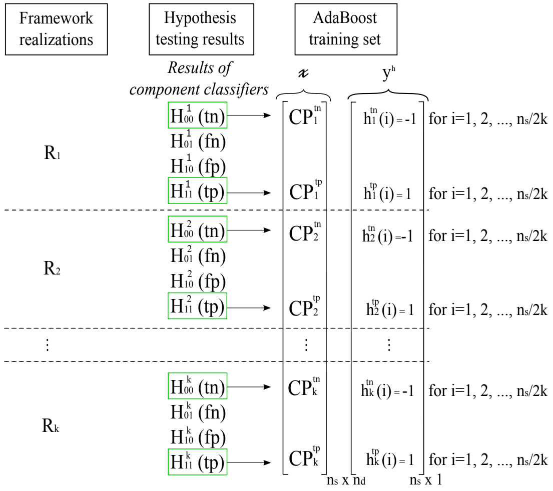


Figure 4.3: AdaBoost training set. Input vector χ contains CP values of different framework realizations which result in true detections. Input vector y^h contains the labels for each CP value (-1 for the healthy state, +1 for the damaged state). If the CPs are n_d-dimensional, CP_k^{tn} and CP_k^{tp} have dimensions $n_s/2k$ x n_d, where n_s is the total number of CP values with tp and tn detections and k is the number of realizations. h_k^{tn} and h_k^{tp} have dimensions $n_s/2k$ x 1 and contain elements equal to -1 and +1, respectively.

by

$$FP_i = \frac{fp_i}{fp_i + tn_i} \tag{4.3}$$

$$FN_i = \frac{fn_i}{fn_i + tp_i} \tag{4.4}$$

where tp and tn are the true positive and true negative detections, respectively. Subsequently, the true positive rate (TP) and the true negative rate (TN) are $TP = 100 - FN$ and $TN = 100 - FP$.

Figure 4.1 presents the interface between the framework and the AdaBoost algorithm. In the AdaBoost stage, CP values which result in true positive (tp or H_{11}) and true negative detections (tn or H_{00}) are used as input to the boosting algorithm along with the corresponding framework decisions (-1 for datasets of the healthy structure and +1 for datasets of the damaged structure). Only the true detections of different realizations are used to build the training set of AdaBoost, as shown in Figure 4.3. The CP values corresponding to true detections form input vector $\boldsymbol{\chi}$, whereas their corresponding detections form vector y^h. These two vectors are used as input to the AdaBoost algorithm. CP values with tn detections represent the healthy structure, while CP values with tp detections represent the damaged structure. Hence, CP values of different realizations corresponding to tp and tn detections are a feature which AdaBoost is to classify correctly.

At this point, the definitions handled in this chapter are summarized in order to facilitate the comprehension of the boosting process.

- Realization of the SHM framework: Defined by one set of settings along the three tiers.
- Classifier of a framework realization: Defined by the implementation of decision boundaries on the calculated CP values. The decision boundaries are based on the assumption of a distribution and a specific confidence interval, while the CP values are a function of the clustering case used in tier 1.
- Component classifier: One of the classifiers intended to be combined in AdaBoost. In the context of this chapter, a classifier of the SHM framework is a component classifier.
- Training set (of AdaBoost): Composed of n_s values of CPs which deliver true detections, as well as their detections (-1 for tn, i.e., for the healthy structure, and +1 for tp detections, i.e., for the damaged structure). The training set contains n_s values obtained from different framework realizations.
- Traning point or traning sample (of AdaBoost): One CP value along with the corresponding detection (+1 or -1).
- Training rounds of AdaBoost: The numbers of iterations in the boosting algorithm. In each training round one training point is chosen and a weak classifier which classifiers this point correctly is defined.

- Base or weak classifiers: Classifiers generated by the boosting algorithm in each training round. These classifiers are generated based on information of the component classifiers.

In the training phase of AdaBoost, successive base classifiers are trained to distinguish between the CPs of the healthy structure and the CPs of the damaged structure. Weights are assigned to the base classifiers depending on their performance on the test data points of each round and the strong classifier is defined as the weighted sum of their hypotheses. In the testing phase of AdaBoost, CP values which were not used in training are classified by the ensemble classifier. The decision of the ensemble classifier regarding the state of the structure for each tested CP value is the sign of the weighted sum forming the strong classifier.

The training phase of AdaBoost is described in detail in Figure 4.4. Let the training set contain n_s training samples, which are CP values labeled either with -1 if they correspond to the healthy state or with +1 if they correspond to the damaged structure. In each boosting round, one sample is selected as the training point, which has to be classified correctly, and the remaining $n_s - 1$ points constitute test points. Moreover, in each round, distribution $\mathcal{D}$ assigns each sample a weight, which corresponds to the importance of classifying this point correctly. The weights of distribution $\mathcal{D}$ can alternatively be conceived as the probability that a given point is chosen as the training point in the next iteration. In the first boosting round, distribution $\mathcal{D}$ is initialized, so that all samples have the same weight. Subsequently, the point resulting in the smallest error rate is chosen as the training point and the remaining $n_s - 1$ points are chosen as test points. Weak classifier $\mathcal{C}_1$ is then defined based on this training and test data selection. Thereafter, hypothesis $h_1(\boldsymbol{\chi})$ and error rate ϵ_1 of classifier $\mathcal{C}_1$ are calculated. Weight a_1, i.e., the weight assigned to classifier $\mathcal{C}_1$ in the strong classifier, is then calculated based on ϵ_1. High error rates result in small values of a_1, whereas the desired classifiers with low error rates obtain a large weight in the strong classifier.

In the second boosting round, probability $\mathcal{D}$ is updated based on weight a_1 of the previous weak classifier ($\mathcal{C}_1$), i.e., based on whether the samples were correctly classified by $\mathcal{C}_1$. Points correctly classified by $\mathcal{C}_1$, i.e., points for which $h_1(\boldsymbol{\chi}) = y_i^h$, are assigned a probability equal to e^{-a_1}. In this way, correctly classified samples are assigned a small probability to be selected as the training sample in the next round. On the contrary, points wrongly classified by $\mathcal{C}_1$, i.e., points for which $h_1(\boldsymbol{\chi}) \neq y_i^h$, are assigned a probability equal to e^{a_1}. Consequently, wrongly classified points are assigned a high probability to be selected as the training sample in the next round. The sample with the highest probability is then selected as the training sample and the remaining $n_s - 1$ points are selected as test points. Weak classifier $\mathcal{C}_2$ is subsequently defined and its hypothesis $h_2(\boldsymbol{\chi})$ and error rate ϵ_2 are calculated, allowing for the determination of the weight of $\mathcal{C}_2$ in the strong classifier (a_2). In this manner, the algorithm adapts the training samples and thereby places the focus on points which are difficult to be correctly classified.

This procedure is followed for T boosting rounds, resulting in a strong classifier.

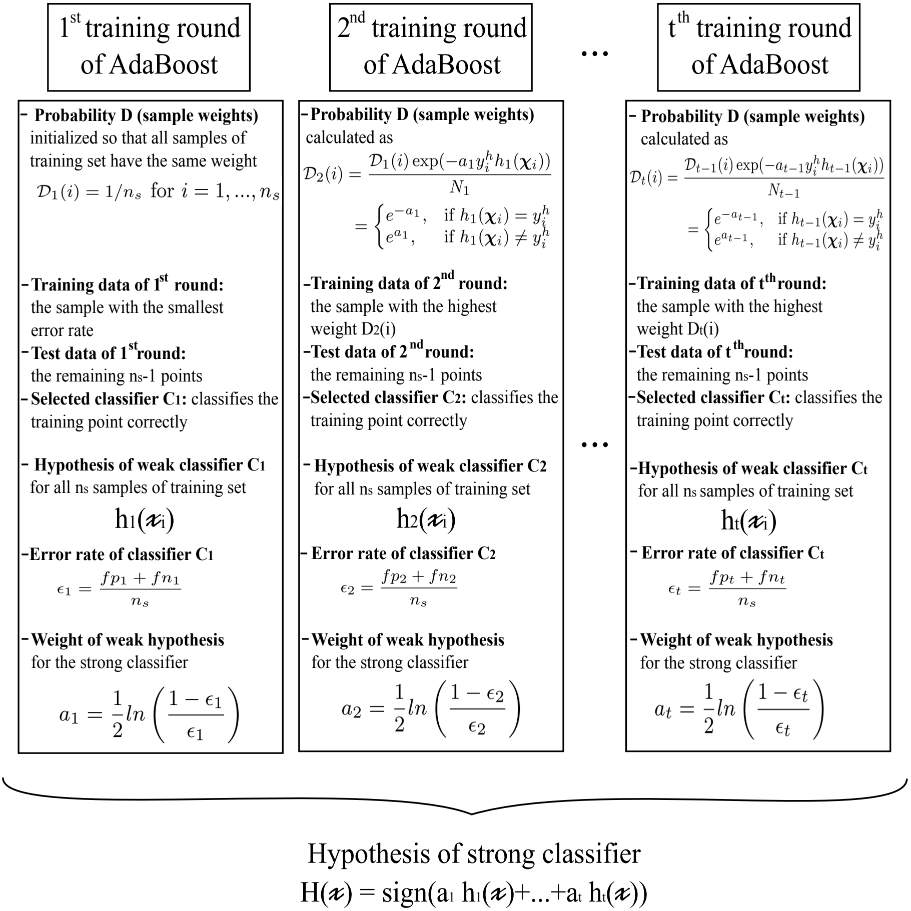

1^{st} training round of AdaBoost
2^{nd} training round of AdaBoost
...
t^{th} training round of AdaBoost
- Probability D (sample weights) initialized so that all samples of training set have the same weight
$\mathcal{D}_1(i) = 1/n_s$ for $i = 1, ..., n_s$
-Training data of 1^{st} round: the sample with the smallest error rate
-Test data of 1^{st}round: the remaining ns-1 points
-Selected classifier C1: classifies the training point correctly
- Hypothesis of weak classifier C1 for all ns samples of training set
$h_1(\chi_i)$
-Error rate of classifier C1
$\epsilon_1 = \frac{fp_1 + fn_1}{n_s}$
-Weight of weak hypothesis for the strong classifier
$a_1 = \frac{1}{2} ln \left(\frac{1-\epsilon_1}{\epsilon_1} \right)$
-Probability D (sample weights) calculated as
$\mathcal{D}_2(i) = \frac{\mathcal{D}_1(i) \exp(-a_1 y_i^h h_1(\chi_i))}{N_1} = \begin{cases} e^{-a_1}, & \text{if } h_1(\chi_i) = y_i^h \\ e^{a_1}, & \text{if } h_1(\chi_i) \neq y_i^h \end{cases}$
-Training data of 2^{nd} round: the sample with the highest weight D2(i)
-Test data of 2 ndround: the remaining ns-1 points
-Selected classifier C2: classifies the training point correctly
- Hypothesis of weak classifier C2 for all ns samples of training set
$h_2(\chi_i)$
- Error rate of classifier C2
$\epsilon_2 = \frac{fp_2 + fn_2}{n_s}$
-Weight of weak hypothesis for the strong classifier
$a_2 = \frac{1}{2} ln \left(\frac{1-\epsilon_2}{\epsilon_2} \right)$
...
- Probability D (sample weights) calculated as
$\mathcal{D}_t(i) = \frac{\mathcal{D}_{t-1}(i) \exp(-a_{t-1} y_i^h h_{t-1}(\chi_i))}{N_{t-1}} = \begin{cases} e^{-a_{t-1}}, & \text{if } h_{t-1}(\chi_i) = y_i^h \\ e^{a_{t-1}}, & \text{if } h_{t-1}(\chi_i) \neq y_i^h \end{cases}$
-Training data of t^{th}round: the sample with the highest weight Dt(i)
-Test data of t^{th}round: the remaining ns-1 points
-Selected classifier Ct: classifies the training point correctly
-Hypothesis of weak classifier Ct for all ns samples of training set
$h_t(\chi_i)$
- Error rate of classifier Ct
$\epsilon_t = \frac{fp_t + fn_t}{n_s}$
-Weight of weak hypothesis for the strong classifier
$a_t = \frac{1}{2} ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$
Hypothesis of strong classifier
$H(\chi) = sign(a_1 h_1(\chi)+...+a_t h_t(\chi))$

Figure 4.4: Training rounds of Adaboost. In each round t, one classifier emerges with a hypothesis $h_t(\boldsymbol{\chi_i})$ and a weight for the strong classifier a_t. The strong classifier is obtained as a weighted sum of T weak classifiers, where T is the total number of boosting rounds.

The hypothesis of the strong classifier is given as the sign of the weighted sum of the weak classifier hypotheses obtained in each boosting round. In the testing phase, a vector $\boldsymbol{\chi}$ containing new CP values is input in AdaBoost. The strong classifier is then used to evaluate these new CP values.

4.3 Effect of Overlapping CP Distributions

Prior to training the AdaBoost classifier, the CP set of each realization is normalized separately with respect to its minimum and maximum values, so that $CP_i \in [0, 1]$. The box plots presented in Figure 4.5 represent the percentiles of the normalized CP values of two hypothetical CPs, CP_1 and CP_2. More specifically, the CP values are distinguished into the CPs resulting in *tp*, *tn*, fp and fn detections. The central red lines of the boxes represent the medians, while the left and right edges of the boxes represent the 25^{th} and the 75^{th} percentiles, respectively. Outliers are indicated by red + symbols. By observing the percentiles of the CP values, conclusions can be drawn about the overlaps of the corresponding distributions. For instance, overlaps of *tp* and *tn* distributions occur within the distribution of one CP, when data clustering has been performed in tier 1.

When applying boosting to a damage detection problem, the improvement of detection performance can be decomposed into the reduction of the fp and fn rates. Hence, it is desirable to transform fp detections into *tn* detections (H_{10} into H_{00}) and fn detections into *tp* detections (H_{01} into H_{11}). This can be ideally achieved when:

- CP values with fp detections of one realization overlap with CP values with *tn* detections of the other realization, so that H_{10} is transformed to H_{00}, and when
- CP values with fn detections of one realization overlap with CP values with *tp* detections of the other classifier, so that H_{01} is transformed to H_{11}.

Some of the overlaps can be exploited to enhance the boosting performance of AdaBoost. For instance, in the example presented in Figure 4.5, evaluating CP values of the fp (H_{10}) distribution of CP_1 in testing is expected to result in *tn* (H_{00}) detections, since these values were designated as healthy by the *tn* detections of CP_2 in training. As a consequence, the fp detections of the CP_1 are expected to be reduced. Accordingly, evaluating CP_2 values which delivered fn (H_{01}) detections is expected to give *tp* (H_{11}) detections, causing a reduction of the fn rate of CP_2. In this manner, the true detections of one CP are exploited in order to reduce the false detections of the other. The positive effect and the negative effect of overlaps between the hypothesis testing results of CP_1 and CP_2 are summarized in Table 4.1.

Histogram comparison methods can be employed to extract a metric for the quantification of the overlap between the individual distributions. The Bhattacharyya

Table 4.1: Effect of overlaps between CP distributions corresponding to hypothesis testing results tp (H_{11}) , tn (H_{00}), fp (H_{10}) and fn (H_{01}). Symbol + indicates positive, - negative, n+ neutral-to-positive and n- negative-to-neutral boosting effect.

	tp_2	tn_2	fp_2	fn_2
tp_1	+	-	-	+
tn_1	-	+	+	-
fp_1	-	+	n-	n-
fn_1	+	-	n-	n-

coefficient is a metric which expresses the overlap between two statistical samples or distributions [99]. It involves a form of integration of the overlaps of two probability distributions p_a and p_b. The two distributions are split into a number of bins and the coefficient is calculated using the number of values in each common bin. For discrete samples, the Bhattacharyya coefficient (BC) is defined as:

$$BC(p_a, p_b) = \sum_{i=1}^{n_{bins}} \sqrt{p_{a_i} p_{b_i}} \tag{4.5}$$

where n_{bins} is the number of histogram bins. When the number of overlapping samples of each distribution is normalized with respect to the total number of values in the distribution, BC ranges from 0 to 1. Hence, BC values equal to 0 indicate no overlap between two histograms, whereas values equal to 1 indicate complete overlap between two histograms.

However, the distribution of CP values may have high divergences from the ideal case presented in Figure 4.5, resulting in smaller overlaps with positive effect or bigger overlaps with negative effect (see Table 4.1). Figure 4.6 shows an example with no overlap between the tn detections of CP_2 and the fp detections of CP_1. A possible way to overcome the lack of positive overlaps is to exclude specific CP values from the training and testing phase. This can be achieved, for instance, by omitting from the analysis CP values belonging to a specific cluster. Thereby, it may be investigated, whether leaving out certain clusters from the analysis results in the desirable overlaps.

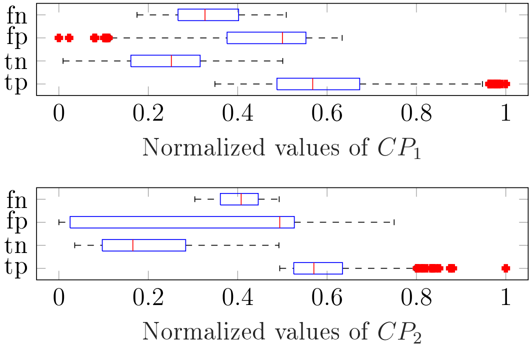


Figure 4.5: Box plots of normalized CP_1 and CP_2 values corresponding to tp, tn, fp and fn detections for example 1, which has high positive effect overlaps. For instance, some positive overlaps are observed: (i) between the fp detections of CP_1 and the tn detections of CP_2 (contribution to reducing the fp detections of CP_1) and (ii) between the fn detections of CP_2 and the tp detections of CP_1 (contribution to reducing the fn detections of CP_2).

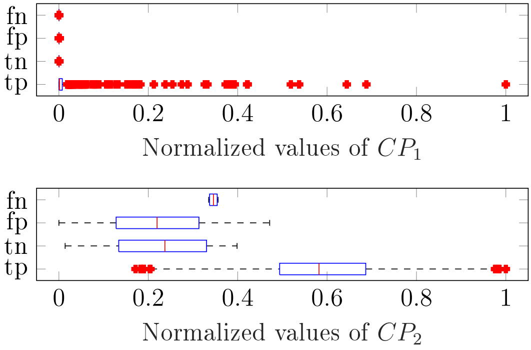


Figure 4.6: Box plots of normalized CP_1 and CP_2 values corresponding to tp, tn, fp and fn detections for example 2, which has low positive effect overlaps. The missing overlap between the tn detections of CP_2 and the fp detections of CP_1 can have a negative effect in boosting.

4.4 Training Data Selection

The training set of AdaBoost consists of *tp* and *tn* detections of framework realizations. Training the classifier equally with respect to the number of *tp* and *tn* instances, as well as with respect to the number of CP values from the individual realizations, is decisive for the performance of the new classifier. Therefore, the assumption was made that the training set should contain approximately the same number of datasets from each realization. Simultaneously, for each realization the number of *tp* detections has to be approximately equal to the number of *tn* detections.

The schematic diagram in Figure 4.7 describes the procedure followed for the selection of the training data. Let k be the number of realizations considered for boosting. Each realization i consists of a set containing n_{tp_i} CP values with *tp* detections and n_{tn_i} CP values with *tn* detections. If there is a sufficient number of *tp* and *tn* detections, a user-defined number of minimum training data n_{sel} is selected. Otherwise, if there is a small number of *tp* and *tn* detections and, subsequently, a small number of training data, the minimum number n_{min} among the number of CPs with *tp* and *tn* detections of all k realizations is determined. Then, systematic sampling is used to sample each set of CPs. Choosing systematic sampling over random sampling for the selection of training data offers a more representative set of CP values and thus helps to avoid bias. In systematic sampling, elements are chosen from an ordered list of values [100].

The process starts by randomly selecting an element from the list. Then, the sampling interval s is defined as

$$s = round\Big(\frac{N_p}{n_{sel}}\Big) \tag{4.6}$$

where N_p is the size of the population and n_{sel} is the number of elements to be selected. As a result, every s^{th} element is chosen. In the context of AdaBoost, the ordered list of values corresponds to percentiles of each set containing CP values with *tp* or *tn* detections. Thus, the size of the population is $N_p = 100$ and the number of elements to be selected is either $n_{sel} = n_{min}$ if sufficient training data is available, or $n = n_{min}$ if few datasets are available for training. In the current work, instead of randomly selecting the first element, it was defined as the 1^{st} percentile. The sampling interval s is then defined and the remaining elements are chosen. Finally, the training set of AdaBoost contains $2kn_{min}$ CP values and their labels -1 or +1 for the healthy state and the damaged state, respectively. Prior to training, the selected values of each CP should be normalized separately with respect to their minimum and maximum values, so that the normalized values lie between 0 and 1. Accordingly, the CP values included in the test set should be normalized separately with respect to the minimum and maximum values obtained from the individual CPs of the training set.

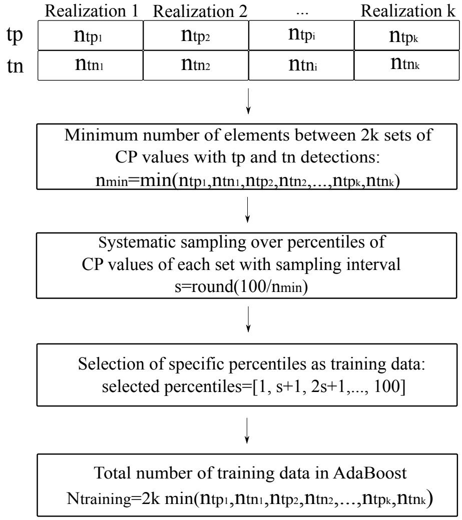

Realization 1
Realization 2
...
Realization k
tp
tn
ntp1
ntp2
ntpi
ntpk
ntn1
ntn2
ntni
ntnk
Minimum number of elements between 2k sets of CP values with tp and tn detections:
nmin=min(ntp1,ntn1,ntp2,ntn2,...,ntpk,ntnk)
Systematic sampling over percentiles of CP values of each set with sampling interval s=round(100/nmin)
Selection of specific percentiles as training data: selected percentiles=[1, s+1, 2s+1,..., 100]
Total number of training data in AdaBoost Ntraining=2k min(ntp1,ntn1,ntp2,ntn2,...,ntpk,ntnk)

Figure 4.7: Selection of training datasets for AdaBoost using systematic sampling when few datasets are available. k is the number of realizations and consequently the number of features input in AdaBoost. n_{tp_i} and n_{tn_i} are the numbers of CP values delivering tp and tn detections.

4.5 Cross-Validation

Systematic sampling was chosen over random sampling for the sake of consistency among the investigated boosting examples (see Figure 4.7). However, the test error occuring when testing the remaining datasets does not necessarily represent the classifier generalization error, i.e., the error obtained when testing an independent test sample. Cross-validation is a widely used method for estimating the prediction error (also called average generalization error) of a supervised learning algorithm. It is employed especially when few training datasets are available. k-fold cross-validation is one of the simplest and most commonly used approaches for the estimation of prediction error. In k-fold cross-validation, the available set is randomly divided into k folds of equal size. First, $k-1$ folds are used in training and the algorithm is evaluated on the remaining fold. This is repeated k times, each time leaving one fold out to be used as validation set. Finally, the estimated performance is the mean value of these k error rates. TP and FP rates can also be used for the assessment of the classifier performance. Therefore, when few datasets are available, training datasets are selected according to Figure 4.7 and the generalization error is estimated using cross-validation.

4.6 Boosting Metrics

The error rate and the TP and FP rates are metrics which enable the assessment of boosting performance. The error rate is the ratio of false detections (fp and fn) to the total number of datasets. In the ideal case, boosting is achieved when the error rate of the AdaBoost classifier is smaller than the error rates of the individual framework classifiers. If k realizations are used in boosting, this can be expressed as:

$$\Delta\epsilon_i = \epsilon_{AdB} - \epsilon_i < 0, \text{for } i = 1, 2, .., k \tag{4.7}$$

On the other hand, the FP rate is the ratio of the fp detections to the total number of healthy datasets (see Equation 4.3). Similarly, the TP rate is the ratio of the tp detections to the total number of damaged datasets. In the ideal case, the TP rate of the AdaBoost classifier is higher than the TP rates of the framework classifiers and the FP rate is smaller than the FP rates of the framework classifiers. Thus, in terms of TP and FP rates, the AdaBoost classifier outperforms the SHM classifiers when Equations 4.8 and 4.9 are simultaneously satisfied.

$$\Delta TP_i = TP_{AdB} - TP_i > 0, \text{for } i = 1, 2, .., k \tag{4.8}$$

$$\Delta FP_i = FP_{AdB} - FP_i < 0, \text{for } i = 1, 2, .., k \tag{4.9}$$

Observing the TP and FP rates offers a more detailed insight than the error rate. As a consequence, two possible ways of improving the detection performance are defined:

- $\Delta TP_i > 0$ and $\Delta FP_i < 0$: Optimal case.
- $\Delta TP_i > -10$ and $\Delta FP_i < 10$: Suboptimal case. The TP rate of AdaBoost is either higher than the TP rates of the AdaBoost classifiers or up to 10 percentage points lower. At the same time, the FP rate of AdaBoost is either lower than the FP rates of the framework classifiers or up to 10 percentage points higher. This case allows for small deteriorations of some rates as long as the other rates are significantly improved.

4.7 Summary

This chapter suggested AdaBoost for combining the decisions of various damage features for increasing detection performance, and introduced the concept of integrating of AdaBoost with an SHM framework. In SHM applications, decision making or the classification of a dataset as healthy or damaged is achieved by extracting damage-sensitive features and by defining decision boundaries. This process can be conceived as the implementation of a classifier.

The applicability conditions of AdaBoost extend in two directions. On the one hand, the features have to stem from similar spaces and should have the same form. As a result, AdaBoost can only be used to combine CPs which have the same dimensions. On the other hand, the weak learning assumption sets a lower limit on the detection performance of component classifiers which may be used in boosting. The weak learning assumption states that the error rate of component classifiers on their original test set has to be smaller than 0.5. Moreover, the weak learning assumption can be used to exclude framework classifiers which perform worse than random guessing and help to eliminate the corresponding framework paths. These two conditions define the boundaries for the applicability of AdaBoost and specify which damage features may be used in boosting.

The contribution of this chapter lies in introducing the integration of AdaBoost with an SHM framework. In order to demonstrate this conjunction, the three-tier SHM framework presented in section 1.3.1 was used. Different realizations emerge by selecting different settings in each framework tier. These realizations may result in different decisions regarding the structural state. Each connection of a CP and HT, including a specific value of confidence interval, can be conceived as a classifier. AdaBoost may be employed to combine the classifiers and thereby the decisions of various framework realizations.

The integration of AdaBoost with the SHM framework and the interface between these two components were described in detail. Special focus was placed on the overlaps of CP distributions. More specifically, the distributions of normalized CP values, which correspond to tp, tn, fp and fn detections, were considered. The effect of overlaps between these distributions was discussed, since they affect boosting performance. The overlaps were quantified by means of the Bhattacharyya coefficient, which offers a metric for comparing two histograms.

Some other important aspects which have to be considered when applying AdaBoost were discussed. One of these is the selection of training data when few datasets are available. It was suggested that the training data should contain approximately the same number of CP instances from each realization and, moreover, approximately the same number of CP instances corresponding to tp and tn detections. The second aspect addressed refers to the need for cross-validation, which is essential for estimating the generalization error of the algorithm, especially when few datasets are available. For this purpose, k-fold cross-validation, which is one of the simplest and most commonly used approaches, was suggested.

Finally, metrics which enable the evaluation of boosting performance were defined. The first metric compares the error rates of framework realizations to the error rates of AdaBoost. In addition, a metric which compares the changes in TP and FP rates was suggested. The advantage of the latter is that it expresses boosting performance as a function of the familiar TP and FP rates, which are extensively used for the evaluation of SHM processes.

The concept of employing AdaBoost to boost the detection performance of SHM processes was presented for vibration-based damage features. However, the coupling of AdaBoost with SHM concepts is also applicable for other SHM approaches, as long as these features satisfy the applicability conditions of AdaBoost.

Chapter 5

Sensitivity Analysis of ARX Transmissibility Function Pole Migration

In chapter 3, the concept of monitoring structural changes using transmissibility function (TF) pole migration was presented and a new CP was proposed. This chapter presents a sensitivity analysis of TF pole migration. The objective of this chapter is, on the one hand, to provide insight on which factors affect TF pole positions, and, on the other hand, to explore the sensitivity of TF pole positions to structural changes and varying EOCs. A further goal is to draw information regarding the mechanics behind pole migration, which is valuable for developing a monitoring strategy in an unsupervised mode. More specifically, the information extracted can be used for the effective deployment of the new pole migration-based CP within the context of the three-tier SHM framework, which involves training with healthy datasets and testing with new incoming datasets, which are potentially healthy or damaged.

First, the effect of the selected model order on TF pole positions is investigated and it is examined whether pole positions provide additional information for the selection of the optimal model order. In addition, it is investigated whether model order selection negatively affects the final detection performance of the proposed CP. Data from other states involving structural changes, such as damage or nonlinearities, as well as variations of EOCs and loading are analyzed. The purpose of this analysis is to show that pole positions vary when the structure is subjected to damage and EOC changes and to examine whether these two effects can be distinguished. One of the structural responses is used as an input in the ARX model, rendering it a single-input multiple-output (SIMO) model. The effect of the input channel on pole migration and damage detection is also investigated, since several SIMO ARX models occur when several responses are available. Finally, the relationship between TF poles and system poles is presented briefly.

5.1 Sensitivity Analysis Factors and Analyzed Databases

The selected model order, the structural state and the surrounding conditions (EOCs) are some factors which are expected to affect TF pole locations on the z-plane. Examining the effect of the model order can provide useful information for the selection of the optimal model order. Examining the effect of damage and comparing it to the effect of varying EOCs can provide with information on whether data clustering is required and on whether TF pole positions allow the distinction between structural changes and changes caused by EOC variations.

In order to expound pole migration and provide the needed insight for the subsequent employment of the proposed CP, a sensitivity analysis was performed. Several datasets of the baseline state were used to compute TF poles, which were compared to TF poles resulting from other structural states. In order to investigate all aforementioned aspects, three databases were used. The first database contained experimental data from a test structure subjected to damage scenarios, as well as to structural changes which intended to simulate EOC variations. The other two databases contained experimental data from a full-scale rotor blade test, which included an ice accretion test and a fatigue test resulting in real structural damage. Table 5.1 presents the databases, the corresponding excitation types and structural states, as well as the factors investigated within the sensitivity analysis.

The three-story frame structure provided by the Los Alamos National Laboratory (LANL) was the first database used in the sensitivity analysis. The structure base was excited by a shaker and datasets were collected at the baseline state, at the damaged state, for varying EOCs and for the combined case of damage and varying EOCs. This database is used to show the impact of model order and input channel in the output-only ARX model. Moreover, it allows to examine the impact of damage and varying EOCs, as well as the distinguishability of these two processes. The second database contained datasets from the modal test of a 34 m rotor blade. The blade was excited in the first edgewise mode shape (second natural frequency) and datasets were collected at the undamaged state, at four steps of added mass and at the damaged state. Added masses simulated ice accretion, while the damaged state involved damage at the trailing edge due to fatigue. The third database contained data from a fatigue test of the same 34 m rotor blade, which was harmonically excited in the edgewise direction until the occurrence of damage at the trailing edge. Different load levels were applied during the experiment, while microcracks caused by fatigue were observed prior to the occurrence of damage. Thus, the effect of varying loads, nonlinearities (such as fatigue microcracks) and damage were examined. It was also investigated whether it is possible to distinguish between structural changes and varying loads.

In the modified SIMO ARX model presented in Chapter 2, structural responses are used as an input instead of some other exogenous input, such as system excitation. If n_{ch} is number of measured signals, one of them is the input of the ARX model and the remaining $n_{ch} - 1$, are the output. In this way, n_{ch} possible ARX models are obtained. In the following sections, model i is the model in which channel i is used as the input. For instance, if 12 signals are available, model 5 corresponds to the model which uses channel 5 as input and channels 1 to 4 and 6 to 12 as output.

Table 5.1: Databases presented and analyzed in this chapter. Excitation type, available structural states and factors investigated within each analysis are given for each database.

Database	Excitation	States	Factors/effects investigated
I. LANL 3-story building	Base excitation by a shaker	1. Baseline 2. Changes in EOCs (simulated by added mass and reduced stiffness) 3. Damage 4. Damage and EOCs	- model order - input channel - varying EOCs - correlation of input channel & location of structural changes - simulated damage - distinction between damage and varying EOCs
II. Modal test on a 34 m rotor blade	Manual excitation of the first flap mode	1. Baseline 2. Ice step 1 3. Ice step 2 4. Ice step 3 5. Damage at the training edge	- real damage - added mass (ice accretion) - input channel (quantifiability of effect)
III. Fatigue test on a 34 m rotor blade	Harmonic loading in edgewise direction	1. Baseline 2. Increasing load levels 3. Fatigue signs/Nonlinearities 4. Damage at the training edge 5. Damage propagation	- increasing load - nonlinearities (fatigue signs) - damage at the trailing edge - damage propagation - distinction between damage and increasing load level

5.2 TF Pole Sensitivity to Simulated Damage - LANL 3-Story Building

5.2.1 Experimental Setup and Database

LANL offers open access data from several experimental tests, such as the database of a three-story building structure, which served for the validation of the LANL statistical pattern recognition paradigm, as well as for the validation of approaches attempting to detect damage under varying EOCs without EOC measurements. The experimental setup and results from these studies are reported thoroughly in [73].

The three-story building structure consists of aluminum plates and columns, which are assembled using bolted joints with a rigid base. Each plate (30.5x30.5x2.5cm) represents one floor and is connected to the top and bottom aluminum plates by four aluminum columns (17.7x2.5x0.6cm). An additional center column (15.0x2.5x2.5cm) is suspended from the top floor in order to induce nonlinear behaviors when it contacts a bumper mounted on the second floor. The extent of nonlinearity can be varied by changing the bumper's position. The entire structure slides on rails, allowing movement only in one direction (x-direction). The test structure and the basic dimensions of its components are presented in Figure 5.1.

The structure was excited at the base floor by an electromagnetic shaker with a band-limited random excitation in the range of 20-150 Hz. This excitation signal was chosen in order to avoid the rigid body modes of the structure lying below 20 Hz. A force transducer was attached at the end of a stringer to measure the input force from the shaker. Both the structure and the shaker were mounted on a base plate (76.2x30.5x2.5cm). Four accelerometers with a nominal sensitivity of 1000 mV/g were measuring the response of each floor. The accelerometers were mounted in the middle of each floor and were therefore uncapable of capturing torsion modes of the structure. Datasets were collected at a sampling frequency of 322.58 Hz.

The detailed test plan is presented in Table 5.2. Four conditions were investigated during the experiment, each including different states. 10 measurements were recorded for each state. The tested conditions were the following:

1. The baseline condition (State #13). The suspended column and bumber were members of the structure but had no contact during the excitation.
2. The baseline condition when the structure has undergone changes in stiffness by EOC effects. Such variations usually cause changes in stiffness or mass in a linear manner. Hence, tests were performed with different mass and stiffness conditions. Mass modifications involved adding a 1.2 kg mass to the first floor and the base (States #01 and #02). Stiffness modifications involved the stiffness reduction by 50% in selected columns (States #17, #18 and #21 to #24).
3. State conditions with nonlinearities imposed by the bumper (State #08 to #12). The gap between the bumper and the suspended column was varied (0.05, 0.10, 0.13, 0.15 and 0.20 mm) in order to introduce different degrees of nonlinearity.
4. State conditions with nonlinearities and EOC effects (States #14 to #16). This group involves both added masses and nonlinearities induced by the bumper. The aim of this study is to detect damage when the structure has undergone structural changes due to EOCs.

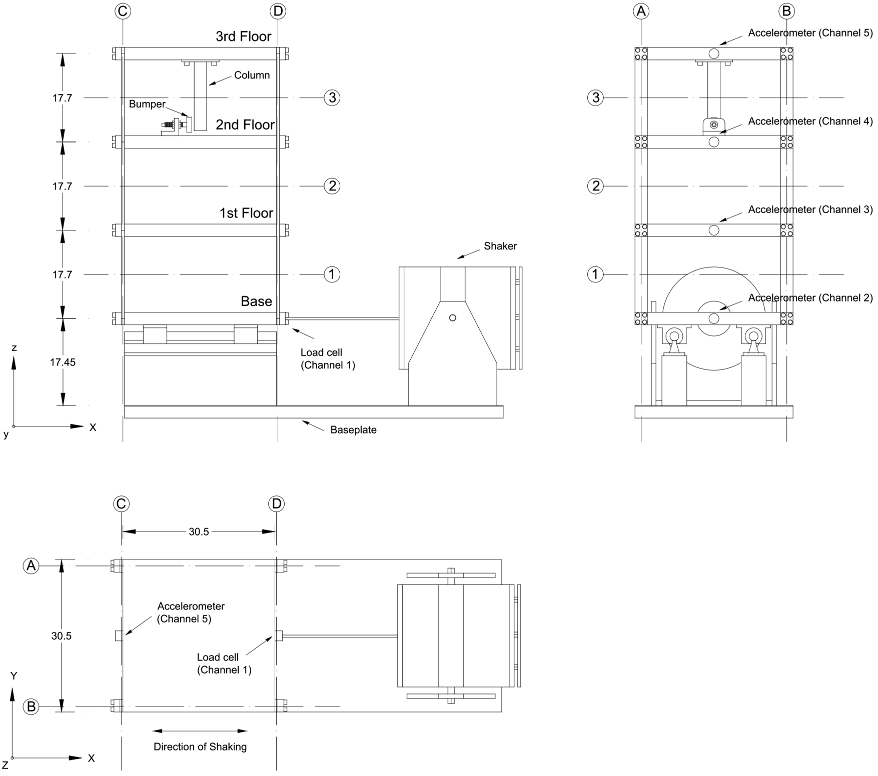


Figure 5.1: Experimental setup for the LANL three-story building structure. Front, side and bottom views. Dimensions shown in cm. (source [73])

Table 5.2: Overview of systems states and database for the LANL three-story building structure ([73]).

Label	State condition	Structural state
State#01	Mass on the 1st floor	EOCs
State#02	Mass at the base	EOCs
State#08	Gap=0.13mm	damage
State#09	Gap=0.10mm	damage
State#10	Gap=0.05mm	damage
State#11	Gap=0.15mm	damage
State#12	Gap=0.20mm	damage
State#13	Baseline condition	baseline
State#14	Gap=0.20mm + mass on the 1st floor	damage and EOCs
State#15	Gap=0.10mm + mass on the first floor	damage and EOCs
State#16	Gap=0.20mm + mass at the base	damage and EOCs
State#17	Column: 1BD - 50% stiffness reduction	EOCs
State#18	Column: 1AD + 1BD - 50% stiffness reduction	EOCs
State#21	Column: 3BD - 50% stiffness reduction	EOCs
State#22	Column: 3AD + 3BD - 50% stiffness reduction	EOCs
State#23	Column: 2AD + 2BD - 50% stiffness reduction	EOCs
State#24	Column: 2AD - 50% stiffness reduction	EOCs

5.2.2 Effect of Model Order

Table 5.3 shows which columns are adjacent to each of the floors, as well as which channels measure the responses of each floor. The channel numeration is given as documented by the LANL in [73]. The third and fourth columns of the table show which channels are the input of the four output-only ARX models. The procedure of estimating the optimal order of the ARX model was presented in Chapter 2. A range of potentially optimal model orders was extracted from Figure 3.3. The computation of the TF poles for all model orders within this range, reveals the sensitivity of TF poles to model order and can provide an indication about the appropriate model order.

Table 5.3: Correspondence between the channels of the LANL three-story model and the input channels of ARX models 1 to 4. Corresponding floors and adjacent columns presented in the first two columns. Channel numbers of experimental setup listed according to LANL documentation. Channel 1 contains the force applied to the structure and is thus excluded.

Floor	Adjacent columns	Channel on this floor (used as input in the ARX model)	Number of ARX model
Base	1	Channel 2 (accelerometer)	1
1st floor	1, 2	Channel 3 (accelerometer)	2
2nd floor	2, 3	Channel 4(accelerometer)	3
3rd floor	3	Channel 5 (accelerometer)	4

In Figure 5.2 the MSEs for a training dataset and a validation dataset of models 1 to 4 are presented as a function of model order. The range of potentially optimal model orders is obtained by comparing the MSE curve of the training dataset with that of the validation dataset. The optimal model order lies around the point where the MSE of the training set keeps dropping, while the MSE of the validation set starts to increase. For model 1, the range of optimal model orders lies between 4 and 15. It may be observed that all models have approximately the same ranges of optimal orders, which indicates that the optimal order is independent of the input channel.

TF poles from the baseline state are presented in Figure 5.3 for model order 10. For model 1, some poles are located on the real axis and outside the unit circle. This phenomenon also occurs in model 2. In this case, the real TF poles are negative and have higher absolute values than the corresponding TF poles of model 1. Conjugate poles of models 3 and 4 are also located outside the unit circle. This phenonemon intensifies as the input channel number increases.

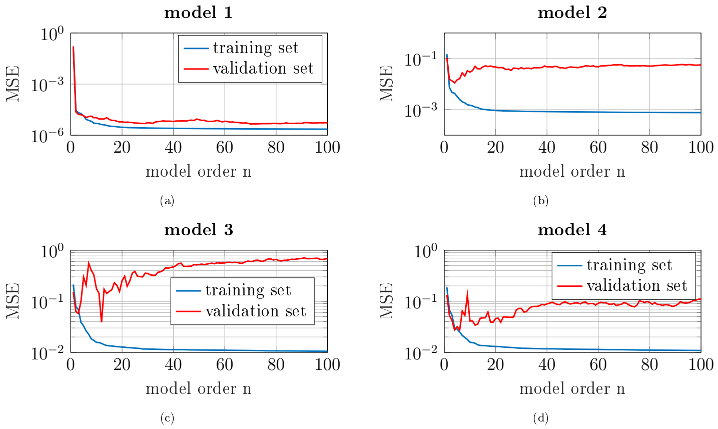
Models 1 to 4 result in different TF poles, despite the fact that the mechanical systems remain the same. This can be attributed to the fact that the output-only ARX model is a transmissibility function model and that the TF poles are zeros (section 3.2). The location of zeros is defined by properties which relate the system input to the system output. Since models 1 to 4 use different inputs and outputs, their TF poles describe different relationships and, therefore, differ.

Poles located outside the unit circle imply that the system has an unstable inverse. Unstable inverse systems in discrete time are equivalent to nonminimum-phase systems in continuous time, so their TF poles correspond to nonminimum-phase zeros (see section 2.1.6) and imply that the structure moves in the opposite direction of the applied load causing a delay in the expected response. This phenomenon is usual in cases of forced excitation or cycling loading. In this chapter, only the poles which lie inside and on the unit circle, as well as poles lying outside the unit circle but aggregating with TF poles inside it are considered. Solutions lying far outside the unit circle are not considered for further investigation.

Figure 5.2(a) gives a range of possible optimal model orders for model 1. TF poles of the system at the baseline state have been computed for a series of model orders to investigate the effect of model order. In Figure 5.4, TF poles of model 1 are presented for orders 5 to 12. As mentioned earlier, only solutions located inside, on and close to the outer part of the unit circle are considered. As the model order increases from 5 to 10, TF poles aggregate. Further increase of the model order results in additional poles on the real axis and inside the unit circle and, subsequently, in lower aggregation (see orders 11 and 12). Therefore, model order 10 is selected as optimal. The optimal model orders for models 2 to 4 were estimated in the same way (see TF poles for orders 6 to 12 in Appendix A, Figures A.1 to A.3). Finally, model order equal to 10 was selected for models 1 and 2 and model order equal to 11 was selected for models 3 and 4. Selecting the model order resulting in the best pole aggregation facilitates the assignment of new poles to the existing pole clusters.

A counterexample is shown in Figure 5.4(d) for model order 8, which results in low levels of aggregation. In this case, challenges occur at two levels: (i) pole clustering is less obvious than in the optimal case and (ii) even if successful pole clustering is achieved, the evaluation of the TF poles of a new dataset with respect to these pole clusters remains challenging.

In the current work, the selection of the optimal model order from a range of potentially appropriate orders was performed manually by observing the aggregation level of the TF poles. This process could be automatized by calculating the TF poles for some model orders (i.e., for the orders indicated when plotting the MSE against the model orders), clustering the TF poles on the z-plane and selecting the order which yields the smallest standard deviations within the clusters. That is, the standard deviation of the clustered TF poles can be used as a measure of aggregation.



(a)

(b)

(c)

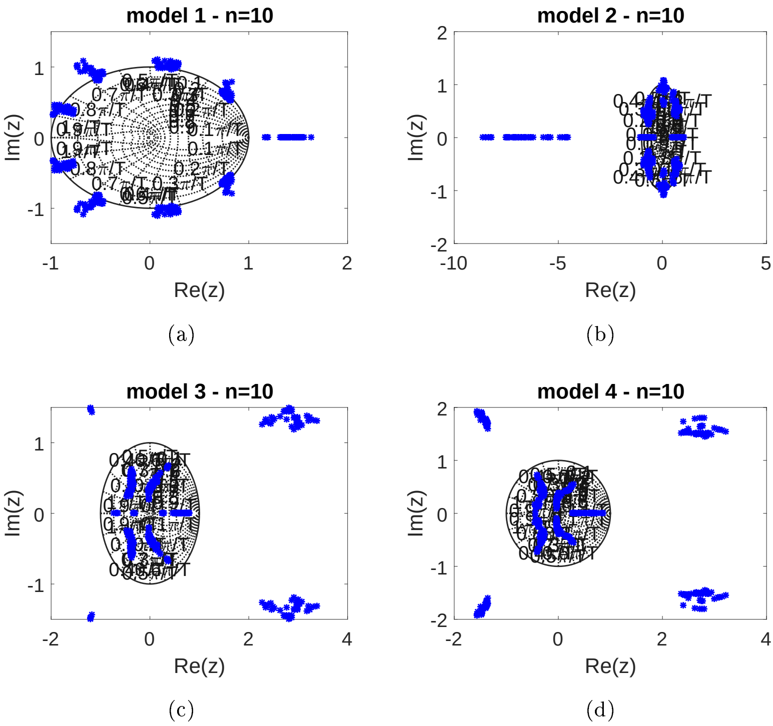
(d)

Figure 5.2: Mean squared error (MSE) of models 1 to 4 as a function of the model order for a training set and a validation set. Optimal model orders are found around the point where the error of the validation set starts increasing while the error of the training set keeps dropping, i.e., between model orders 4 and 15.

5.2.3 Effect of EOCs

After the optimal model order has been selected, the TF poles of all states are calculated. In this section, the results are presented for states, which simulate changes in the EOCs by adding mass to the structure and by reducing the stiffness of specific columns. TF poles of the healthy state along with TF poles for added masses are shown in Figure 5.5. Since the solutions are conjugate complex numbers, only solutions with positive imaginary part are presented. For models 1 and 2, added mass on the first floor of the test structure results in poles which are linearly separable by the poles of the baseline state. Added mass on the base floor does not result in distinct clusters, but still causes pole migration, i.e., changes of both frequency and damping. As for the direction of pole migration, no pattern is observed. Models 3 and 4 result in smaller variations of TF pole positions, but slight pole displacements are observed.

The TF poles of the states with reduced stiffness are presented accordingly in Figure 5.6. The majority of states cause pole migration with respect to the baseline state for all models. The solutions of models 1 and 2, however, exhibit higher sensitivity to stiffness reduction than those of models 3 and 4. Moreover, each pole cluster delivers different information. In model 1, for instance, the left pole cluster (at $0.9\pi/T$) delivers information on the simultaneous stiffness reduction in columns 3AD and 1BD, but does not experience pole migration for the stiffness reduction in column 1BD. On the other hand, the pole located close to $0.7\pi/T$ offers the reverse information. This means that parallel observation of all pole clusters offers more



(a)

(b)

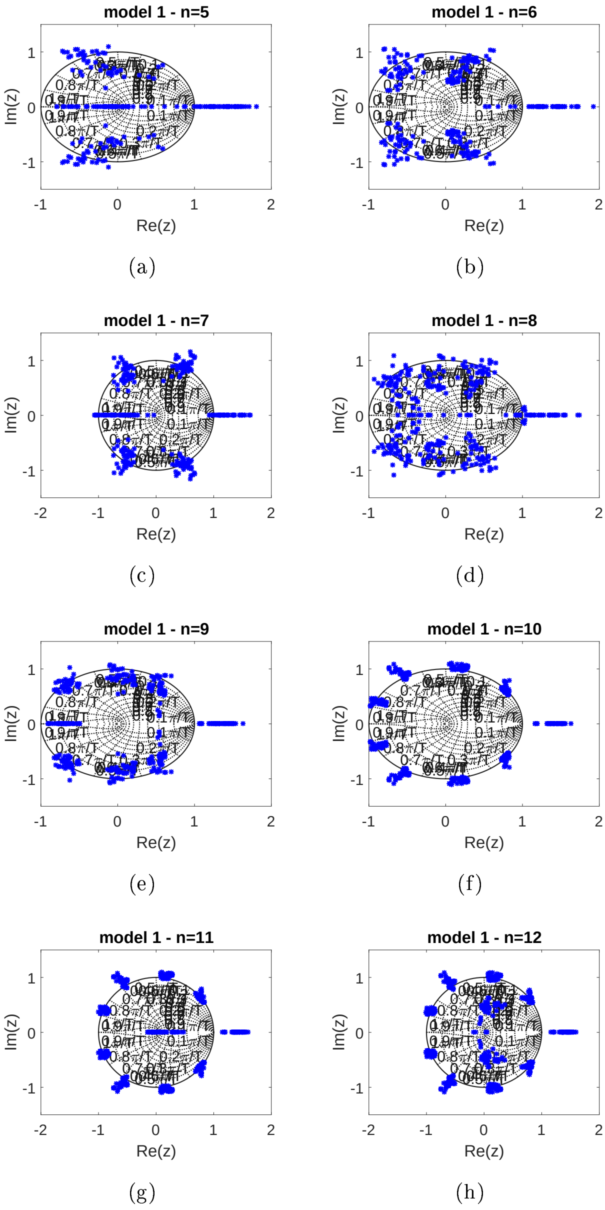
(c)

(d)

Figure 5.3: TF Poles of models 1 to 4 at the baseline state for model order 10. TF Poles far outside the unit circle correspond to nonminimum-phase zeros and indicate delay in the response. The effect intensifies with increasing number of input channel.

information and allows to capture different structural changes or effects. For models 1 and 2, no pattern is observed regarding the direction of pole migration, but both damaged pole clusters of model 3 migrate along the frequency contours towards higher damping ratios. This effect is partially observed also in the case of model 4. Poles from states involving simultaneous stiffness reduction of 3AD and 1BD, as well as columns 1AD and 1BD migrate not only along the frequency contours towards higher damping ratios but also towards higher frequencies (see right pole clusters of model 4).

The analysis of these states shows that the positions of TF pole clusters are affected by both mass increase and stiffness reduction of various columns. These linear changes of mass and stiffness have been used to simulate structural changes due to varying EOCs. Thus, it has been shown qualitatively that TF poles are sensitive to varying EOCs. As a result, data normalization based on EOCs is crucial, when datasets are acquired under varying EOCs. It has also been observed that



(a) (b)

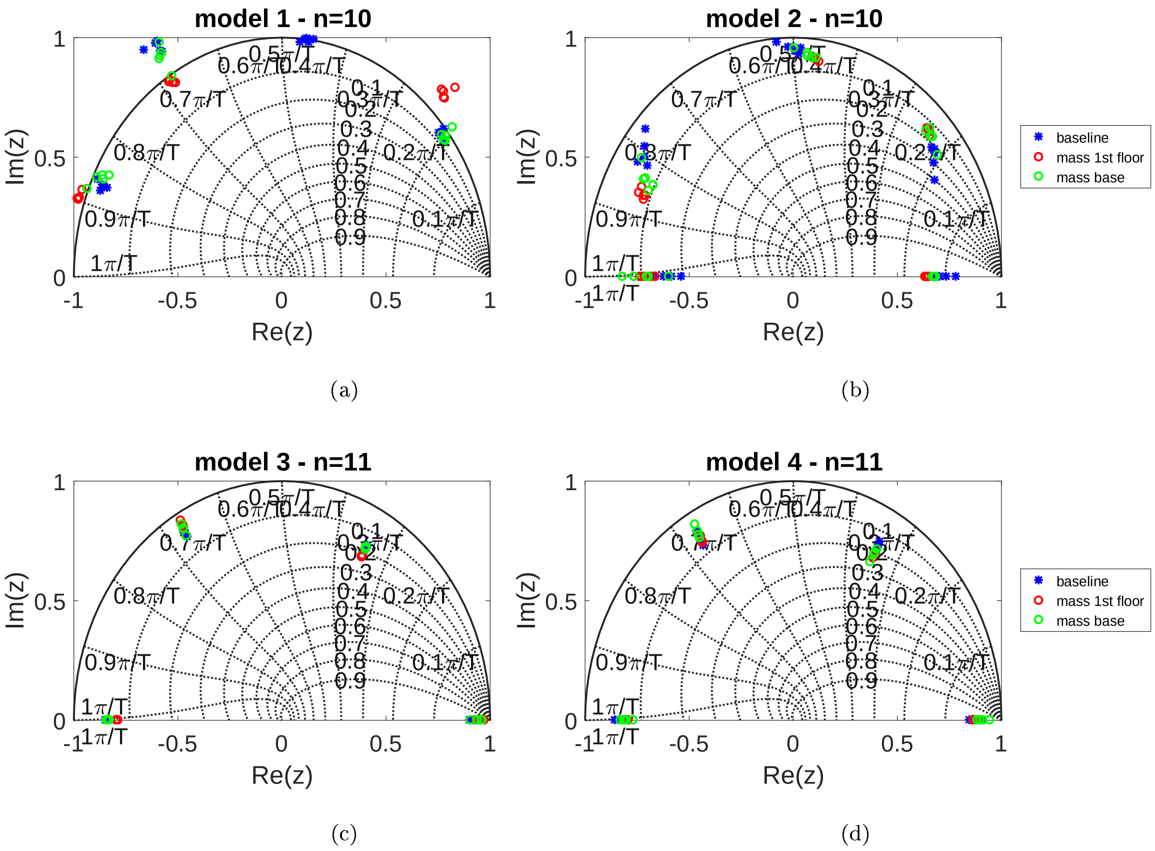
(c) (d)

(e) (f)

(g) (h)

Figure 5.4: TF poles of model 1 at the baseline state for the potentially optimal model orders of the output-only ARX model (orders 6 to 12 according to Figure 5.2(a)). Model order 10 provides the highest level of pole aggregation and is thus selected as the optimal model order.

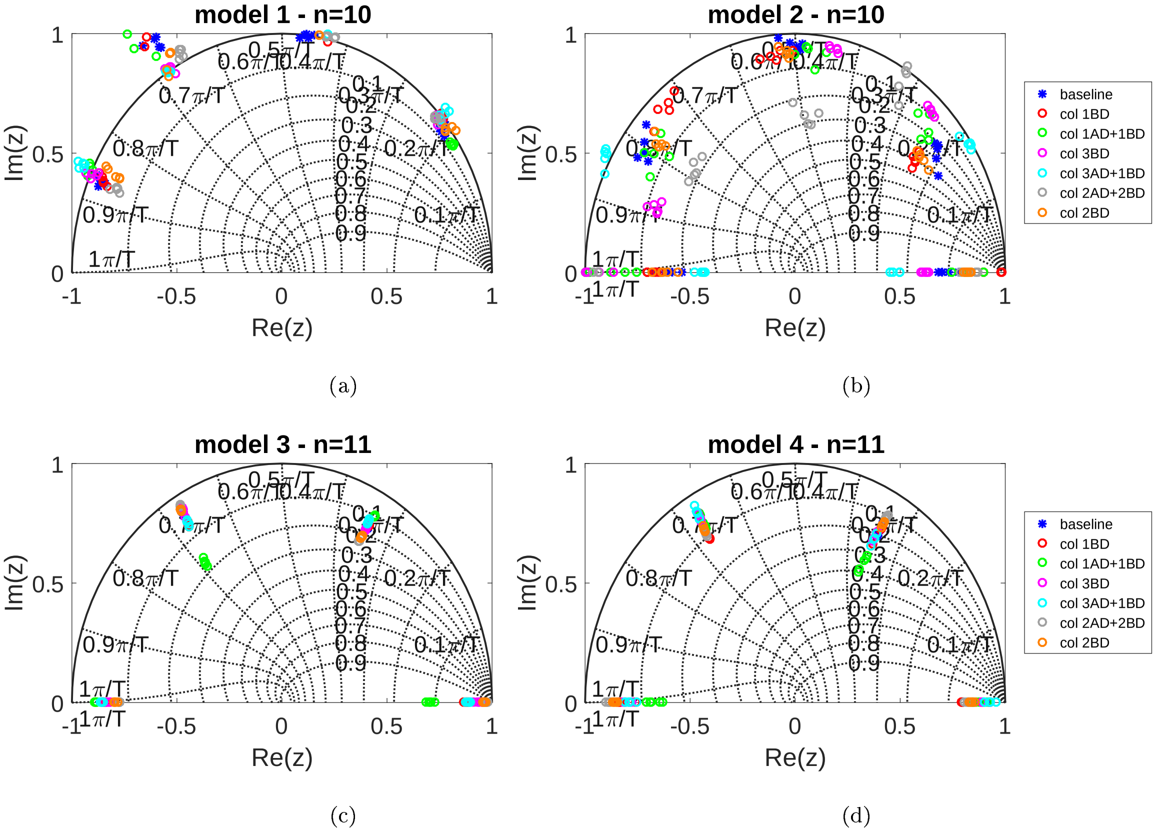
at least one pole cluster within a model exhibits changes due to increased mass or reduced stiffness. This indicates that parallel observation of all pole clusters offers the capture of different effects.



(a) (b)

(c) (d)

Figure 5.5: TF poles from the healthy state and from states with added mass, which simulate EOC variations (States #01 and #02). TF poles of models 1 to 4 shown for their optimal model orders (order 10 for models 1 and 2, and order 11 for models 3 and 4).



(a)

(b)

(c)

(d)

Figure 5.6: TF poles from the healthy state and from states with reduced stiffness, which simulate EOC variations (States #17, #18 and #21 to #24). TF poles of models 1 to 4 shown for their optimal model orders (order 10 for models 1 and 2, and order 11 for models 3 and 4).

5.2.4 Effect of Input Channel

Another aspect which has to be examined is whether the model sensitivity to structural changes is higher when the model input channel lies close to these changes. In particular, it is examined whether stiffness reductions located close to the model input channel cause higher pole cluster displacements than stiffness reductions at other locations. If the input channel has an effect on TF pole migration, then the TF poles corresponding to structural changes in the adjacent columns should migrate more than the TF poles corresponding to changes in the other columns.

The input of model 1 is channel 2, which contains the acceleration of the base floor, while the base floor is adjacent to column 1 (see Table 5.3). Figure 5.6(a) shows that model 1 results in four TF poles per dataset, while TF poles with frequency higher than $0.6\pi/T$ exhibit higher migrations than the TF poles with lower frequency (see poles at $0.2\pi/T$ and at $0.48\pi/T$). The TF poles located at $0.7\pi/T$ indicate that the highest changes with respect to the baseline state are given for changes in column 3BD, for simultaneous changes in columns 2AD and 2BD, as well as for simultaneous changes in columns 1AD and 1BD. These involve changes in columns 1, 2 and 3, while only column 1 is adjacent to the base floor. The input of model 2 is channel 3, which contains the acceleration of the 1^{st} floor, while the 1^{st} floor is adjacent to columns 1 and 2. The TF poles of model 2 are more sensitive to structural changes (see Figure 5.6(b)), while the highest displacements are given for simultaneous changes in columns 2AD and 2BD, for changes in column 3BD and for simultaneous changes in columns 3AD and 1BD. These involve changes in columns 1, 2 and 3, while the adjacent columns of the 1^{st} floor are columns 1 and 2. The input of model 3 is channel 4, which contains the acceleration of the 2^{nd} floor, while the 2^{nd} floor is adjacent to columns 2 and 3. Figure 5.6(c) shows that the TF poles of model 3 migrate less than in the case of models 1 and 2. The highest migrations are observed for simultaneous changes in columns 1AD and 1BD (see TF poles at $0.7\pi/T$), despite the fact that the 2^{nd} floor is adjacent to columns 2 and 3.

Some clusters exhibit high sensitivity to stiffness reductions in columns which are adjacent to the model input, whereas some do not. Hence, the conclusion can be drawn that models with input channels adjacent to the damage location do not necessarily result in higher TF pole displacements on the z-plane than the other models. This result implies that the observation of TF pole migration enables only damage detection but no damage localization.

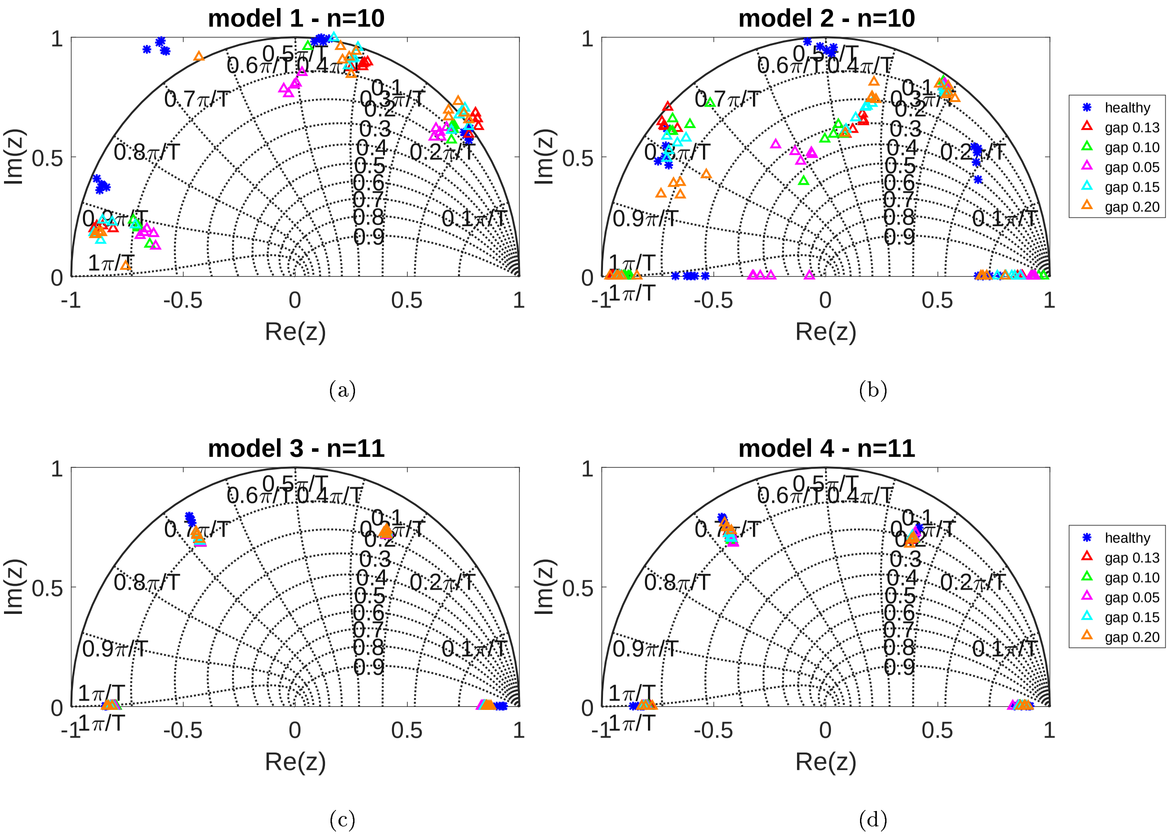
5.2.5 Effect of Damage

The effect of damage was simulated by inducing nonlinearities of different extent. TF poles from these states were calculated and plotted on the z-plane superimposed with TF poles from the baseline state. Figure 5.7 shows the TF poles of models 1 to 4 for their optimal model orders. The solutions of model 1 aggregate in four clusters. Clusters of all damaged states have moved significantly with respect to the baseline state. While TF poles of the two left pole clusters from the baseline state are distinct from the damaged poles, the other two clusters are subjected to smaller diplacements. The baseline TF poles of model 2 with positive imaginary part aggregate in three pole clusters. Here, the location of TF poles is also affected by the different damaged states. Especially the two right clusters are subjected to high displacements due to damage. In this case, it is not possible to assign the damaged TF poles to one of the healthy pole clusters. However, this is not an obstacle for the damage detection process, since the TF poles of the damaged states are not located inside any of the baseline TF pole groups. Therefore, they are identified as damaged, regardless of the baseline pole cluster they are compared to.

The solutions of models 3 and 4 with positive imaginary part aggregate in two pole clusters. For both models, the left pole cluster is subjected to changes, which are, however, much smaller than these of models 1 and 2. As in the case of added masses, pole migration takes place along the frequency contours towards higher damping ratios. However, the positions of the damaged TF poles of the right pole cluster are almost identical to those of the baseline state and are therefore not easily identified as damaged.

It has been shown that nonlinearities of different extent which are intended to simulate structural damage cause pole migration. Two of the four available models exhibit higher sensitivity to damage than the other two. It is not possible, however, to visually quantify the extent of nonlinearity based on the pole cluster displacement.

Each model exhibits different sensitivity to damage. This can be attributed to the fact that TF poles are essentially zeros. The zeros of a system express the relationship between system input and system output, while they depend on the input location. As a result, the TF poles of the models differ between them and do not have the same sensitivity to damage, because they use other input and output channels and, therefore, represent another input-output relationship. A further factor which affects the sensitivity to damage is the model order, which affects the level of TF pole aggregation. Models 1 and 2 was examined for order 10 and models 3 and 4 for order 11, resulting in TF pole clusters with different levels of aggregation.



(a)

(b)

(c)

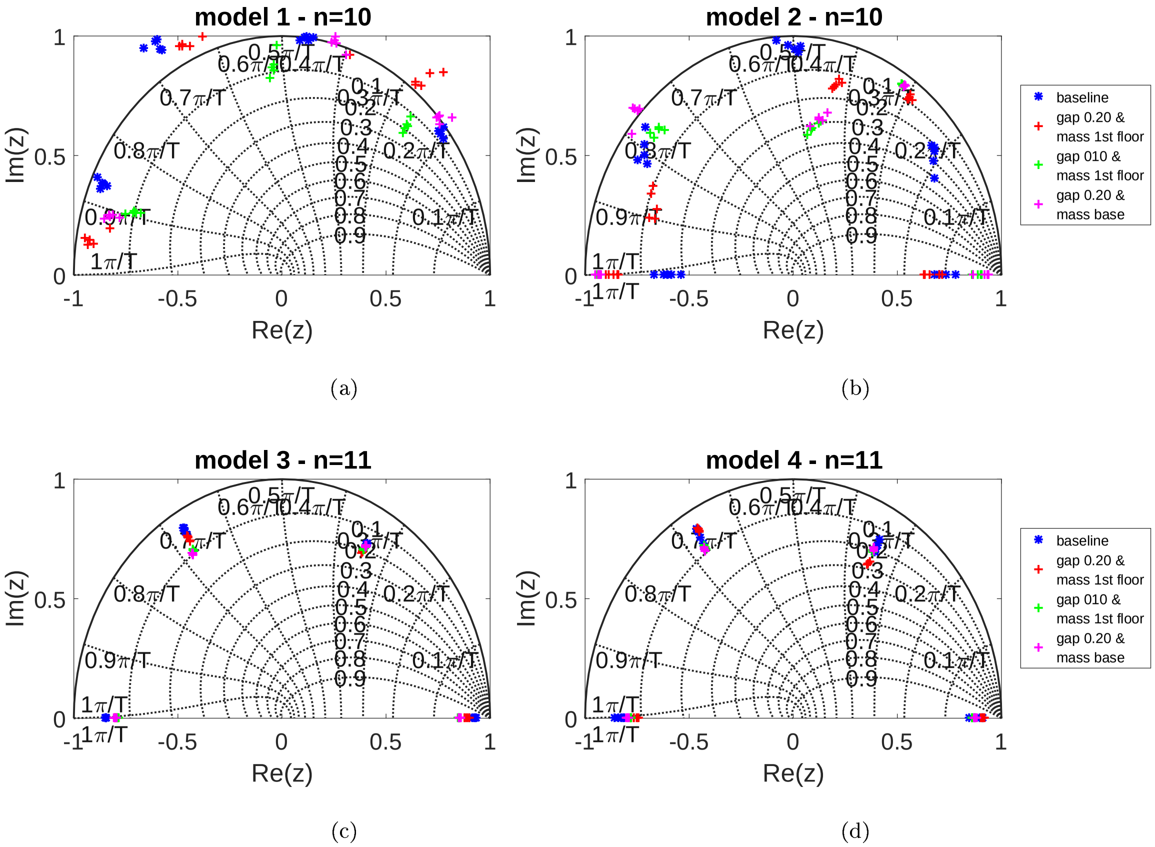
(d)

Figure 5.7: TF poles from the healthy state and from states with nonlinearity, which simulate damage (States #08 to #12). TF poles of models 1 to 4 shown for their optimal model orders (order 10 for models 1 and 2, and order 11 for models 3 and 4).

5.2.6 Effect of Damage Accompanied by EOC Changes

Nonlinearities combined with mass increase and stiffness reduction were applied to the test structure in states #14 to #16. These states simulate the existence of damage for different EOCs. Figure 5.8 shows TF poles from (i) the baseline state, (ii) the state with a gap equal to 0.02 mm and added mass on the first floor, (iii) the state with a gap equal to 0.10 mm and mass on the first floor and (iv) the state with a gap equal to 0.02 mm and mass on the base. The TF poles of model 1 are shown in Figure 5.8(a). TF poles in all three cases ((ii), (iii) and (iv)) deviate from the baseline state and, at the same time, differ from each other.

As in the individual cases of added mass and reduced stiffness, deviations from the baseline state are higher for models 1 and 2 than for models 3 and 4. For models 1 and 2, TF poles of states #14 to #16 deviate significantly from the baseline state. The majority of clusters are linearly separable from the baseline state. However, there is no trend regarding the direction of pole migration, i.e., towards higher or lower frequencies or towards higher or lower damping ratios. On the other hand, pole displacements of models 3 and 4 are smaller than these of models 1 and 2. As for the direction of migration, TF poles migrate towards higher damping ratios with a constant frequency.



(a) (b)

(c) (d)

Figure 5.8: TF poles from the healthy state and from states with stiffness reduction and nonlinearities, which at the same time simulate EOC variations and damage (States #14 to #16). TF poles of models 1 to 4 shown for their optimal model orders (order 10 for models 1 and 2 and, and order 11 for models 3 and 4).

5.3 TF Pole Sensitivity to Real Damage - Modal Test of a Full-Scale Rotor Blade

5.3.1 Full-Scale Rotor-Blade Test

A full-scale blade test was performed on a 34 m rotor blade made of glass fiber/epoxy at the blade center of IWES Bremerhaven, Germany, in June 2014. The objective of the experiment was to carry out modal tests, as well as a fatigue test in the edgewise direction, while continuously measuring the structural responses. The collected data were intended to employ and validate methodologies for ice and damage detection in rotor blades.

First, a modal test involving manual excitation of the first three natural frequencies was performed on the baseline state. The same modal test was performed with added masses, which simulated ice accretion. The added masses were attached to the blade tip in four steps. After the modal tests on the healthy blade and on the "iced" blade, a fatigue test in edgewise direction was carried out for over 1 million cycles. The fatigue test ended with damage at the trailing edge due to fatigue. The damage was located 6 meters from the blade root. Subsequently, the load frame was removed and the modal test was performed anew on the damaged blade. The detailed test plan is presented in Table 5.4.

As it can be seen in Figure 5.9, accelerometers and geophones were distributed in six positions along the blade in order to measure the structural responses. Twelve sensors were available in total, since two sensors were installed in each position in order to measure the responses in both edgewise and flapwise directions. Sensors in positions P1 to P3 were installed inside the blade, on the web located towards the trailing edge. Sensors in positions P4 to P6 were installed on the outside, on the pressure side shell. Velocity signals measured at positions P1 and P2 were derived in order to obtain acceleration signals, which were then used for analysis. The exact sensor locations, sensor types and signals used in the analysis are presented in Table 5.5.

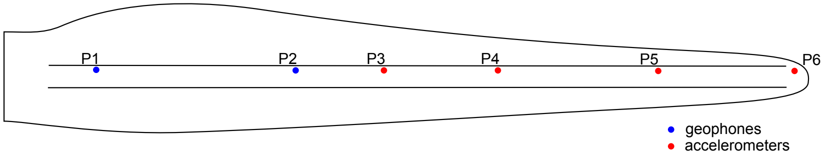


Figure 5.9: Rotor blade with sensor types and sensor positions. Two sensors were installed in each position recording structural responses in both edgewise and flapwise directions (12 sensors in total).

Table 5.4: Rotor blade test plan including excitation type, test parts and blade state for each of the four test setups.

	Excitation type	Part/Applied load level	Blade state/Added mass
Setup 1 Baseline	MODAL TEST Manual excitation	Test on healthy state	healthy
Setup 2 Ice detection	MODAL TEST and ICE Manual excitation	Ice test 1 (Setup 2.1)	+0.1% of blade mass
		Ice test 2 (Setup 2.2)	+0.3% of blade mass
		Ice test 3 (Setup 2.3)	+0.6% of blade mass
		Ice test 4 (Setup 2.4)	+0.9% of blade mass
Setup 3 Fatigue test	FATIGUE TEST Cyclic excitation in edgewise direction (load frame attached to blade)	Step 1 - 70%	healthy insignificant delaminations and cracks
		Step 2 - 75.6%	
		Step 3 - 81.1%	
		Step 4 - 90%	
		Step 5 - 95.6%	
		Step 6 - 105%	
		Step 7 - 110%	
		Step 8 - 120%	fatigue microcracks & insignificant delaminations and cracks
		Step 9 - 130%	
		Step 10 - 140%	
		Step 11 - 170%	damage at trailing edge damage length of 440 mm
		Step 12 - 50%	damaged - total damage length not inspected
		Step 13 - 70%	damaged - total damage length not inspected
		Step 14 - 90%	damaged - total damage length of 443 mm
		Step 15 - 105%	damaged - total damage length of 532 mm
		Step 16 - 115%	damaged - total damage length of 554 mm
		Step 17 - 130%	damaged - total damage length of 731 mm
Setup 4 Damaged	MODAL TEST Manual excitation	Test on damaged state	damaged - damage length of 724 mm

Table 5.5: Sensor positions, sensor types and measured signals.

Sensor position	P1	P2	P3	P4	P5	P6
Distance from root	3.7 m	12 m	15.5 m	20.5 m	27.4 m	34.2 m
Location	inside	inside	inside	outside	outside	outside
Sensor type	geophone	geophone	accelerometer	accelerometer	accelerometer	accelerometer
Signal used in analysis	acceleration	acceleration	acceleration	acceleration	acceleration	acceleration

5.3.2 Modal Test: Experimental Setup and Database

Modal testing is widely used for identifying the modal parameters of a structure, namely natural frequencies, mode shapes and damping ratios. In this section, data from the manual excitation of the first natural frequency of the blade are examined. The first natural frequency corresponds to the first mode shape in flapwise direction. The blade was manually excited and then released in order to obtain free vibration response. The procedure was repeated six times.

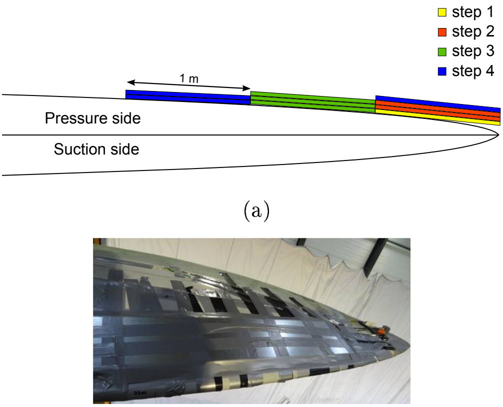
Since blade tips in operation have a higher angular speed and reach higher elevation than the blade root, ice usually accumulates towards the blade tip and less towards the blade root. Moreover, ice accumulates at the leading edge of the blade tip and mostly on the pressure side [101]. Hence, it can be concluded that ice accretion takes place on the leading edge following the direction from blade tip to blade root, resulting in a triangular load.

For the experimental setup, icing was realized by adding extra masses to the blade surface. Nine steel sheets, each weighing 4.8 kg and with dimensions of 1000x200x1.5 mm, were placed and fixed to the blade in four steps. Thus, an ice accretion event was simulated, taking place from the blade tip to the blade root and resulting in a pseudo-triangular load. The ice accretion test steps (setups 2.1 to 2.4), including the exact application positions and weights, are shown in Table 5.6 and Figure 5.10(a). Figure 5.10(b) depicts the actual steel plates which were attached to the blade tip.

The rotor blade was excited at its first and second mode shapes flapwise, as well as the first mode shape edgewise, and was released to obtain the free vibration response. This was repeated six times for each of the aforementioned mode shapes. Figure 5.11 depicts the acceleration time series in both flapwise and edgewise directions recorded in sensor position P6, which was located close to the blade tip. All the rounds of excitation of the second mode shape in the flapwise direction are shown exemplarily in Figure 5.11. While no sensor errors are to be found in the flapwise signals, sensor errors are observed in the edgewise direction in two of the six excitation rounds. Therefore, these rounds are omitted from the analysis. Such sensor errors only occurred a few times, but corroborate the need for repetitions of the applied excitation. After the data were cleansed, the free vibration response, which in Figure 5.11 is indicated by the red rectangles, was selected for further analysis.

Table 5.6: Ice accretion steps and added masses in the rotor blade ice detection test (setup 2).

Step	Number of added steel sheets	Position in m	Total number of steel sheets	Total added weight in kg	Total added weight in % of blade mass
1	1	33-34	1	4.8	0.1%
2	2	33-34	3	14.4	0.3%
3	3	32-33	6	28.8	0.6%
4	3	31-32, 33-34	9	43.2	0.9%



(a)

(b)

Figure 5.10: Ice accumulation steps (a) and steel sheets mounted on the blade surface for the simulation of ice accretion (b).

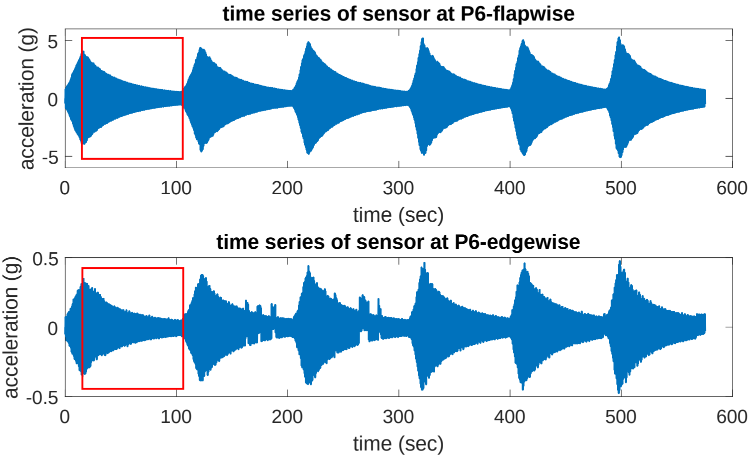


Figure 5.11: Acceleration time series recorded in sensor position P6, 33 m from the blade root. Acceleration shown in both flapwise direction (upper plot) and edgewise direction (lower plot). Sensor errors are observed in the signals of edgewise acceleration during excitation rounds two and three. Red rectangles indicate the free vibration response, which was used in further anaylsis.

5.3.3 Relationship between System Poles and ARX TF Poles

This section focuses on the investigation of the relationship between the system poles and the TF poles obtained from the output-only ARX model. The same procedure was followed for the selection of the optimal model order, as for the LANL three-story model. The MSEs of a training set and a validation set were computed for model orders 1 to 100. These curves provided a range of potentially optimal model orders. The TF poles were then computed for the model orders of this range and the order which offered the highest level of aggregation within each pole cluster was chosen. The optimal model orders of all 12 models are presented in Table 5.7. Models 2 to 11 have optimal model order 12, while models 1 and 12 have optimal model order 13.

In the classical ARX model, which uses exogenous or extra input, the poles of the transfer function denominator correspond to the system poles. These are used to identify the system's natural frequencies and damping ratios. The transmissibility function poles, as shown in Chapter 3, do not correspond to the system poles but to system zeros. Devrient et. al showed published a work which shows that the modal parameters can be identified by combining tranmissibility measurements under different conditions [46]. In order to prove that, they showed that the values of the transmissibility function depend on the excitation location. Moreover, they showed that the transmissibility function values obtained from different loading conditions are the same at the system poles, regardless of the system excitation. Hence, some of the TF poles are in deed the sytem poles.

In order to identify the modal parameters, the modal test data from the healthy state were analyzed with data-driven SSI and the TEMP algorithm. The first seven natural frequencies of the rotor blade were successfully identified. They corresponded to the first four mode shapes in flapwise direction and the first three mode shapes in edgewise direction. The identified natural frequencies, the corresponding damping ratios and types of mode shape are presented in Table 5.8. Natural frequencies and damping ratios were obtained as mean values of the solutions of natural frequency and damping.

The system poles of models 5 and 12, as obtained from the SSI and TEMP, and the ARX TF poles are shown exemplarily in Figure 5.12. For model 5, none of the system poles coincides with the TF poles. The only similarity between the two types of solutions lies around $0.35\pi/T$ (approximately 17.5 Hz), where the system poles have the same frequency content as the TF poles. This occurs due to the fact that the system poles lie on the same frequency contour as the TF poles but at higher damping ratios. For model 12, on the other hand, some system poles coincide with the TF poles around $0.1\pi/T$ (approximately 5 Hz) and between $0.2\pi/T$ and $0.3\pi/T$ (between 10 and 15 Hz). In Table 5.8, it was shown that the second mode shape in edgewise direction is equal to 5.64 Hz, whereas the fourth mode shape in flapwise direction and the third mode shape in edgewise direction are equal to 11.63 Hz and 13.22 Hz, respectively. Hence, model 12 is a case which confirms that some of the TF poles are equal to system poles. Devriendt et al. showed this for different loading conditions, whereas in this work it was shown for different input channels under the same loading conditions.

Table 5.7: Optimal model orders for models 1 to 12 of the rotor blade modal test database.

model	1	2	3	4	5	6	7	8	9	10	11	12
optimal model order	13	12	12	12	12	12	12	12	12	12	12	13

Table 5.8: Modal parameters from impulse excitation for the healthy state (setup 1).

natural frequency	1	2	3	4	5	6	7
mode shape	$1^{st}flap$	$1^{st}edge$	$2^{nd}flap$	$2^{nd}edge$	$3^{rd}flap$	$4^{th}flap$	$3^{rd}edge$
f in Hz	1.069	1.678	3.153	5.642	6.611	11.635	13.226
damping ratio	0.778%	3.643%	1.219%	0.303%	0.575%	0.398%	0.01%

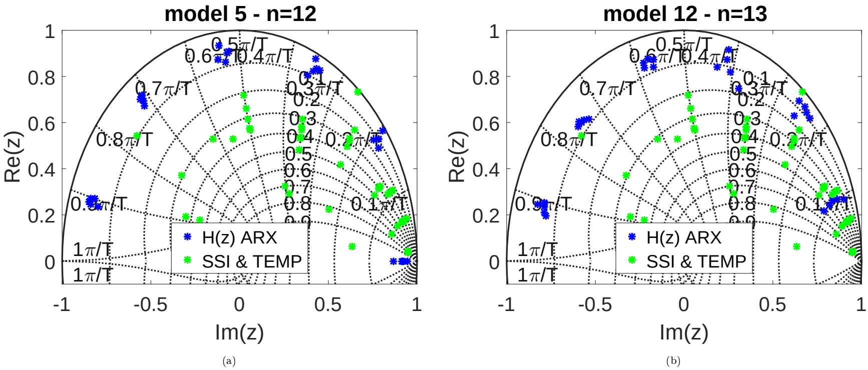


Figure 5.12: ARX TF poles and system poles for the healthy state computed by data-driven SSI and the TEMP algorithm for model 5 (a) and model 12 (b).

5.3.4 Detection and Quantifiability of Damage and Ice Accretion

TF poles were computed for all blade states, i.e., for the healthy state, the four steps of ice accretion and the damaged state. Figure 5.13 shows exemplarily the TF poles of model 5. The TF poles of models 1 to 12 can be found in Appendix A (see Figures A.4 to A.7). In Figure 5.13, the solutions aggregate in five clusters, while the highest displacements are observed for the damaged poles. Ice steps 1 to 4 also result in poles which have migrated from the healthy state. However, the three inner pole clusters exhibit higher sensitivity to both damage and added mass than the two outer pole clusters, which are located close to $0.2\pi/T$ and $0.9\pi/T$.

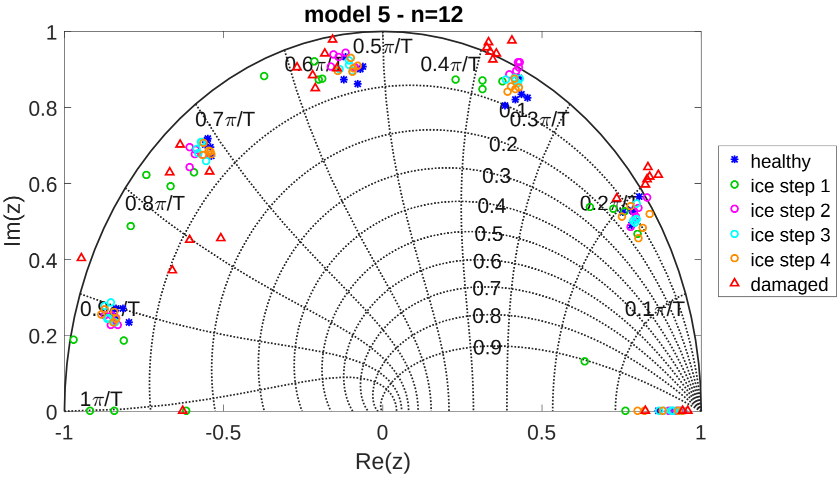


Figure 5.13: TF poles of model 5 from the healthy state, the four "iced" states and the damaged state. TF poles shown for the optimal model order n=12.

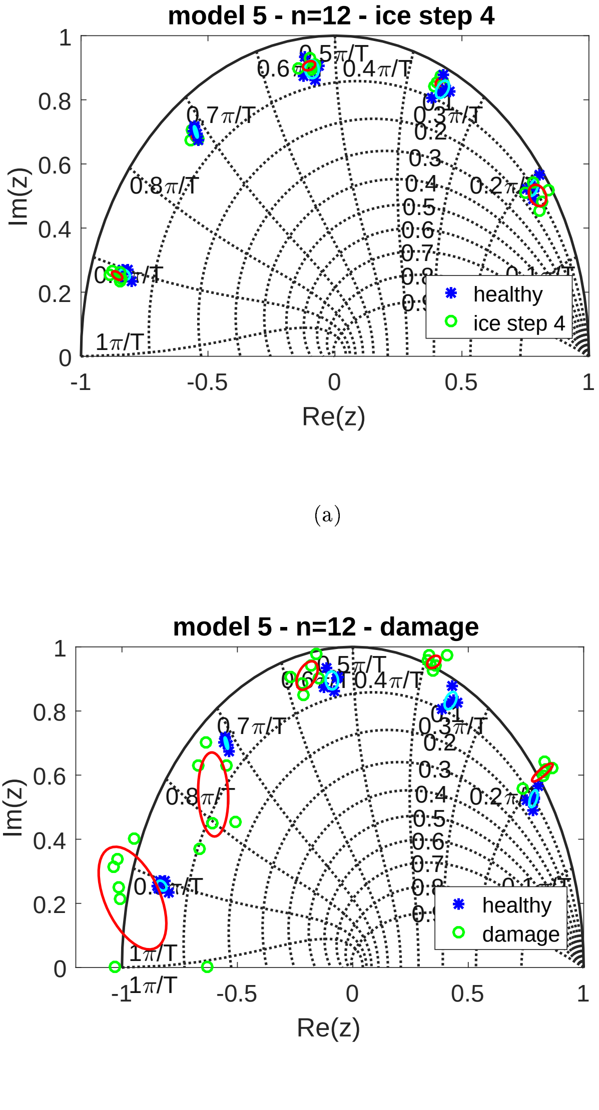
Subsequently, the TF poles from all states were clustered using affinity propagation (AP), and bivariate normal probability density functions were estimated for each pole cluster. Figure 5.14(a) shows the clustered poles of the healthy state superimposed with the poles of ice step 4, while Figure 5.14(b) shows the clustered poles of the healthy state superimposed with the poles of the damaged state. The elliptical level curves in Figure 5.14 represent the distance equal to three times the standard deviation from the mean of the estimated normal density function. Figure 5.14(a) shows that TF pole clusters of the iced state have migrated with respect to the pole clusters of the healthy state. TF pole displacements are relatively small, while the solutions of the two aforementioned states have comparable standard deviations. In Figure 5.14(b), it can be seen that the damaged TF poles exhibit higher migration than the TF poles of ice step 4. As for the change in standard deviation, the clusters of the damaged state located at frequencies lower than $0.5\pi/T$ (equal to 5 Hz) exhibit significantly higher variance than the corresponding clusters of the healthy state. Hence, it can be stated that the damage at the trailing edge causes higher pole migration than the added mass of ice step 4. At the same time, damage causes the TF pole clusters of higher frequencies to have bigger variances than the clusters of both the healthy state and the state of ice step 4.

Three metrics were used for the characterization and quantification of pole migration: (i) the distance between the cluster means of the healthy state and the state to be examined (d), (ii) the difference between the standard deviation both in the major and in the minor direction of the ellipse ($\Delta\sigma_1$ and $\Delta\sigma_2$) and (iii) the rotation of the ellipse ($\Delta\theta$), i.e., the difference between the angle of the ellipse major direction of the healthy state, and the angle of the ellipse major direction of the state to be examined. Figures 5.15 to 5.17 present these three metrics for all TF pole clusters of models 1 to 12. For this purpose, the clusters are numerated starting from frequency equal to 0 and moving towards frequency equal to $1\pi/T$, i.e., starting from the point with coordinates (1,0) and moving towards the point with coordinates (-1,1).

The distance between the mean of a pole cluster at the healthy state and the mean of the corresponding pole cluster at the new state is indicated by d_i, where i is the cluster number. Figure 5.15 presents the cluster-mean distances for ice step 4 and for the damaged state. Each subplot contains the values d_i for all 12 models. The cluster-mean distances of clusters 1 to 5 are shown for ice step 4 on the left part of Figure 5.15. The cluster-mean distances are greater than zero for all pole clusters and models, except for cluster 2 of model 3 and cluster 3 of model 8. The maximum distance between the cluster means is provided by model 1 for clusters 1, 3, 4 and 5, and by model 10 for cluster 2. The extent to which TF poles migrate does not depend on the model input, since the highest migrations are observed for different models. Furthermore, the TF pole clusters of ice state 4 do not migrate as far as the clusters of the damaged state. In particular, cluster-mean distances of ice step 4 reach up to 0.08, while cluster-mean distances of the damaged state reach up to 0.8.

The second metric deployed for the description of pole migration ($\Delta\theta$) is presented for all pole clusters and all models in Figure 5.16. As in the previous plot, the left part of the figure shows cluster rotations for ice step 4, whereas the right part of the figure shows cluster rotations for the damaged state. While no trend can be observed with respect to the model order number or pole cluster number, Figure 5.16 shows that $\Delta\theta$ values are greater than zero for most of the cases. Hence, TF pole clusters experience rotations that range from approximately -200° to +200°.

Standard deviation changes are shown in Figure 5.17 for all pole clusters and models. Each subplot shows $\Delta\sigma$ in the direction of both major and minor axes of the ellipse describing the pole cluster. Standard deviation changes for ice step 4 and for the damaged state are presented in the left and right parts of the figure. For ice step 4, most of TF pole clusters exhibit changes in variation, since $\Delta\sigma$ values are different from zero. However, the major axis of cluster 2 and the minor axis of cluster 1 exhibit differences in standard deviations which are very close to zero. Standard deviation changes for the damaged state, on the other hand, are in general greater than zero, implying that the damaged solutions have higher variances than the solutions of the baseline state. $\Delta\sigma$ values of the damaged state have a higher order of magnitude than the correspoding values of ice step 4. In particular, $\Delta\sigma$ values of ice step 4 range between -0.04 and 0.02, while $\Delta\sigma$ values of the damaged cases range between -0.01 and +0.17. As in the case of the distance metric, no pattern is observed that connects the model number to changes in standard deviation.



(a)

(b)

Figure 5.14: (a) TF poles of model 5 for the healthy state and for ice step 4 (n=12). (b) TF poles of model 5 for the healthy state and the damaged state (n=12). Healthy state indicated by blue TF poles and cyan ellipses. Ice step 4 and damaged state indicated by green TF poles and red ellipses.

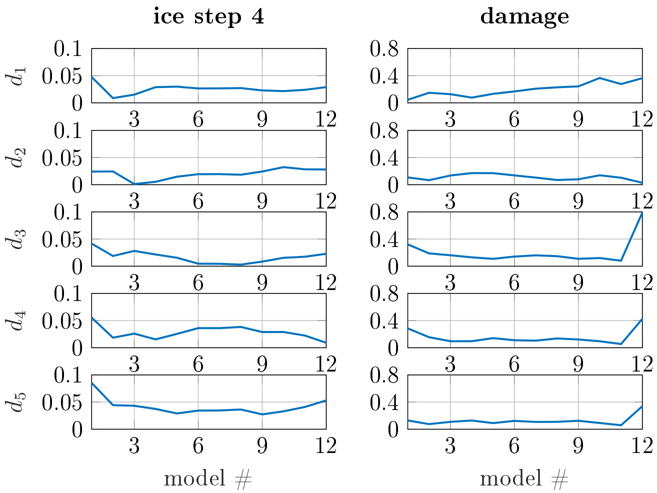


Figure 5.15: Cluster-mean distances (d_i) for ice step 4 (left) and for the damaged state (right), where i=1, 2, ..., 5 is the cluster number. Each subplot contains the distances of cluster i for models 1 to 12. Distances calculated with respect to the healthy state clusters.

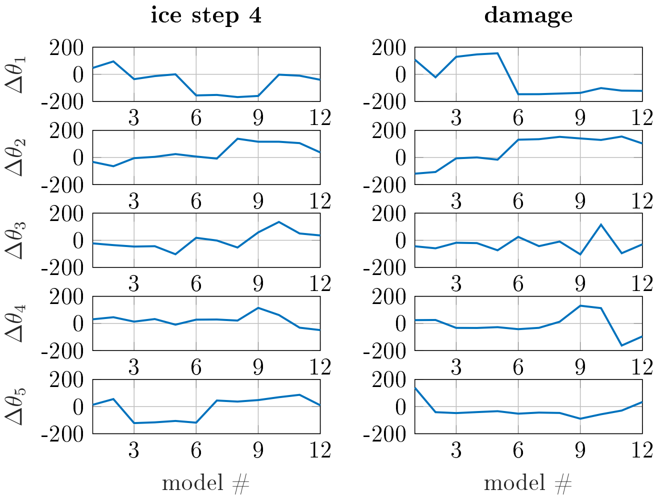


Figure 5.16: Ellipse rotation ($\Delta\theta_i$) for ice step 4 (left) and for the damaged state (right), where i=1, 2, ..., 5 is the cluster number. Each subplot contains the ellipse rotation of cluster i for models 1 to 12. Changes calculated with respect to the clusters of the healthy state.

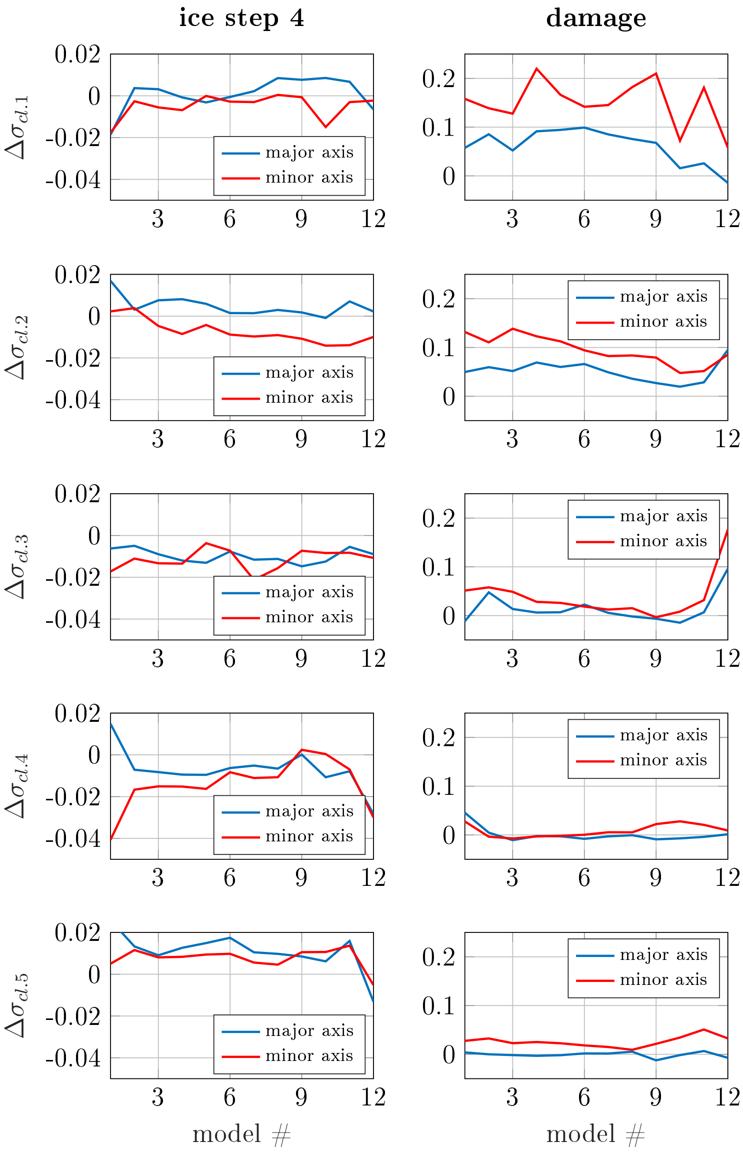


Figure 5.17: Standard deviation changes ($\Delta\sigma_{cl.i}$) for ice step 4 (left) and for the damaged state (right), where i=1, 2, ..., 5 is the cluster number. Each subplot contains the changes in standard deviation in the major axis and in the minor axis of cluster i for models 1 to 12.

5.4 TF Pole Sensitivity to Real Damage - Fatigue Test of a Full-Scale Rotor Blade

5.4.1 Fatigue Test: Experimental Setup and Database

An edgewise fatigue test was carried out after the modal test on setups 1 and 2 (i.e., on the healthy state and ice accretion) was completed. Its objective was to apply cyclic loading in edgewise direction in order to introduce slow and steady failure, while continuously monitoring the structural responses. For this purpose, gradually increasing dynamic loads were applied 17.5 m from the blade root by means of a load frame for approximately 1.3 million cycles. A hydraulic cylinder was connected to the load frame and actuated the blade close to its first mode shape in edgewise direction (second natural frequency of the structure) with horizontal, cyclic forces. The test setup, including the load frame and the driving structure, is shown in Figure 5.18.

No information regarding the design loads or material layup for this specific blade type was available. Instead, 10 million cycle equivalent loads of the successor blade were used for load calculation [102]. The successor blade had the same length and similar design. According to the IEC technical specification, partial safety factors have to be used in such tests for the blades to withstand higher test loads. These are the partial safety factors (i) for the consequence of failure γ_n, (ii) for blade-to-blade variation γ_s and (iii) for errors in fatigue formulation γ_{ef}. They are equal to 1.15, 1.10 and 1.05, respectively, and result in the total safety factor γ, which was applied to the design loads

$$\gamma = \gamma_n \cdot \gamma_s \cdot \gamma_{ef} = 1.15 \cdot 1.1 \cdot 1.05 \approx 1.33. \tag{5.1}$$

Moreover, an additional factor had to be applied to the design loads in order to account for the lower number of cycles. Instead of loading with 10 million cycles, it was planned to load the blade with approximately 1 million cycles. This assumption took into account the available testing days and the fact that the first edgewise natural frequency was approximately 1.5 Hz. The material lifespan N_l in cycles for sinusoidal fatigue loading with mean stress σ_m is determined by the S-N curve equation as

$$N_l = \left(\frac{\sigma_u}{\sigma_a}\right)^b \tag{5.2}$$

where σ_a is the stress amplitude, σ_u is the ultimate stress and b is the curve slope, which is chosen empirically [103]. According to the blade design document, the S-N curve slope could be assumed to be equal to 10. Thus, the additional safety factor for design loads was calculated as the ratio of the number of cycles for which the equivalent loads were available (N_d) to the planned number of cycles (N_p)

$$\sqrt[b]{(N_d/N_p)} = \sqrt[10]{(10^7/10^6)} \simeq 1.26. \tag{5.3}$$

In addition to the sensors applied, strain gauges were distributed along the blade providing strain values, which would serve as driving values for load application and

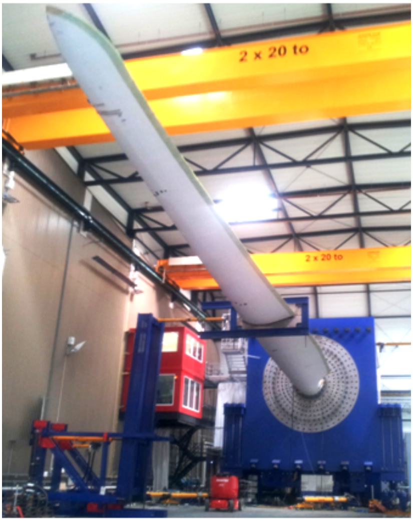


Figure 5.18: Test setup for fatigue test with load frame attached 17.5 m from blade root. Loading structure on the left.

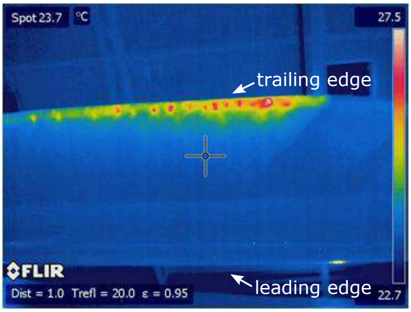


Figure 5.19: Thermal image of the pressure side shell from root (right side) up to 11 m from the blade root (left side) at the beginning of step 4 of the fatigue test (DEL 90%). Hot spots at trailing edge coincide with final damage location.

for stopping loading in case strain exceeded safe levels. Before the test started, the blade was quasi-statically pulled to the side in edgewise direction to determine the relationship between the bending moments and measured strain values. However, the fact that the design loads did not correspond to the blade which was in fact tested but to its successor blade, led to some necessary modifications to the plan during the experiment. The final damage equivalent loads (DELs) applied during the test are presented in Table 5.4 and refer to the measured strain 2 m from the blade root at the trailing edge side.

During the experiment, visual inspections of the blade interior and of the trailing edge were performed at least once a day. For this purpose, the outer coating was removed along the trailing edge. Furthermore, the blade was inspected with an infrared camera, which indicated areas where heat was produced due to friction caused by cyclic excitation. Figure 5.19 shows a thermal image of the pressure side shell up to approximately 11 m from the blade root after load step 4, which corresponds to DEL of 90%. The hot spots shown in Figure 5.19 coincide with the actual, final damage location.

In Figure 5.20, the applied DEL and the accumulated damage are presented as a function of the number of cycles. The depicted accumulated damage is a theoretical prediction calculated using the Palmgren-Miner rule. Load levels in the diagram

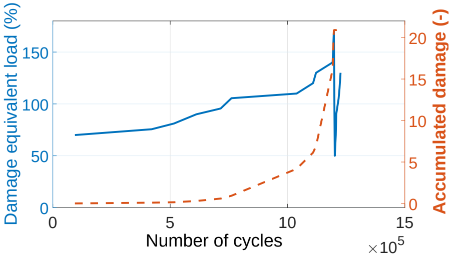


Figure 5.20: Damage equivalent load (DEL) and accumulated damage during the fatigue test. DEL corresponds to the reference point for load application located 2 m from the blade root, whereas the accumulated damage corresponds to the location of the trailing edge damage, 6 m from the blade root.

correspond to the reference point for load application, which was at the radius 2 m from the blade root, whereas accumulated damage is given for the radius 6 m from the blade root, where the significant damage finally occurred. For the first test day, DEL was set to 70% and the test was run until the next morning. Inspections after loading with this DEL did not reveal any damage. Then, the load was increased to 75.6%. For the first six steps presented in Table 5.4 no significant damage was observed during the inspections. Merely small delaminations and cracks had formed on the structure. Step 7 with DEL of 110% lasted 2.5 days approximately. During the inspection, which took place directly after step 7, many microcracks of the matrix were observed along the fibers over the entire blade surface. Local debonding between resin and fibers is a sign of fatigue in composite materials, which occurs prior to damage and which can be associated with crazing of thermoplastic materials [104]. This phenomenon intensified during steps 8, 9 and 10, after which the number of microcracks increased. The assumption is made that the data recorded during step 7 (DEL 110%) are not damaged, since it is unknown when exactly during this step the formation of the microcracks began.

During step 11 (DEL 170%), the blade failed with a crack at the trailing edge, approximately 6 m from the root. The failure extended to both suction and pressure sides of the trailing edge and a had total length of 440 mm on both shells. Since it was intended to cause slow damage propagation, the experiment was continued but with significantly reduced loads (step 12, DEL 50%). Subsequently, the load was again gradually increased up to 130%. After step 14 (DEL 90%), a 3 mm propagation was observed. After step 15 (DEL 105%), crack propagation was equal to 89 mm, resulting in a total crack length of 532 mm. After step 16, damage propagated further by 22 mm and after step 17 by another 177 mm, resulting in a total crack length of 731 mm. Summarizing, the blade condition during the fatigue test can be divided into four states:

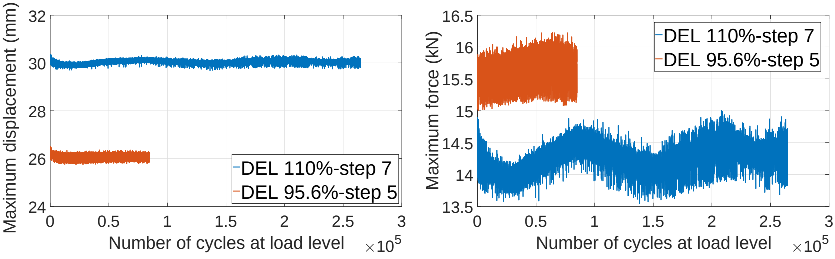
1. Healthy state (steps 1-7)
2. Fatigue microcracks and insignificant damage, i.e., small delaminations and small material or bondline cracks (steps 8-10)
3. Trailing edge damage (end of step 11)
4. Crack propagation (steps 12-17)

The fatigue test was controlled by two overlapping cycles. The first was displacement-controlled and the other ensured that certain force thresholds were not exceeded. Figure 5.21(a) presents the maximum displacement applied to the load frame 17 m from the blade root. The orange curve depicts the maximum displacement during step 5 with DEL 95.6%, while the blue curve depicts the maximum displacement during step 7 with DEL 110%. Maximum forces at the load frame are shown for the same steps in Figure 5.21(b). While the force fluctuations observed for DEL 95.6% are negligible, significant fluctuations of maximum forces can be observed for DEL 110%. However, changes of the maximum displacement for DEL 110% are much smaller than the force fluctuations. Taking into account that the test was mainly displacement-controlled, the fluctuation of maximum forces during loading with DEL 110% suggests that the blade was subjected to structural changes at that load level. This finding is in agreement with the visual inspection after DEL 110%, which indicated fatigue microcracks in the material.

Furthermore, it may be observed that the maximum forces during loading with DEL 110% are lower than the maximum forces during loading with DEL 95.6%, despite the fact that the load level is higher. DEL 95.6% started after approximately 715,200 loading cycles, whereas DEL 100% starts after approximately 1039000 loading cycles. Composite laminates are subjected to stiffness degradation during fatigue loading, while the F_{max}/F_{min} ratio decreases with increasing number of cycles. The reduced maximum forces during DEL 110% can therefore be attributed to stiffness degradation of the rotor blade structure.

Acceleration signals were available at all six positions shown in Figure 5.9. At a certain time point during the fatigue test, the geophone at position P2 provided erroneous signals due to overmodulation. TF ARX poles examined in this work are calculated by signals from all sensor positions. Removing one signal from only one part of the database (in this case, a specific load step), would result in noncomparable poles between different parts of the fatigue test. Thus, the two channels of sensor position P2 were excluded from the complete analysis and signals from positions P1 and P3 to P6 were finally used for the analysis employing the SHM framework.

Gradual increase in the applied load during the fatigue test led to a corresponding increase in the vibration amplitude. Data normalization was performed in order to distinguish between damage feature changes caused by damage and structural changes resulting from load variation. More specifically, signals were standardized by subtracting the mean value of the signal and then by dividing the result by the mean value of the standard deviations of all signals. Thereby, each acceleration signal had zero mean and constant variance throughout the entire fatigue test, while the relationship between the channels was preserved.



(a)

(b)

Figure 5.21: Maximum displacement (a) and maximum force at load frame (b) during steps 5 and 7 of fatigue test (DEL 95.6% and 110%).

5.4.2 Detection of Nonlinearities Due to Fatigue and Damage

In the previous section, it was mentioned that two out of the twelve channels had to be removed from the analysis due to erroneous signals. The remaining 10 channels were subsequently used to obtain 10 SIMO ARX models. The same procedure as in the case of the two previous databases was followed for the selection of the optimal model order. First, the MSEs of a training set and a validation set were calculated for model orders 1 to 100 and the comparison of the two curves provided a range of potentially optimal model orders. TF pole clusters were then calculated for the model orders included within that range and the model order offering the highest level of pole aggregation was selected as the optimal model order. Table 5.9 shows the selected model orders for models 1 to 10. Order equal to 11 was chosen for model 1 and order equal to 10 was chosen for models 2 to 6 and 10. Models 7 to 9 had an optimal order equal to 8.

The analysis of the LANL three-story building showed that TF pole positions are affected by EOC variations. The fatigue test database contains a series of applied load levels at different structural states. The load levels constitute the varying EOCs. Unless data normalization or standardization is performed, only load levels available at both the healthy state and the damaged state are comparable. These are load levels 70%, 90% and 105%. As mentioned in the fatigue test description, the datasets were standardized in order to compensate the effects of increasing loads. Figures 5.22(a) and 5.23(a) exemplarily show the TF poles of these load levels for models 9 and 10, with load level 70% of the healthy state being the baseline state. The TF poles of the remaining models are presented in Appendix A (see Figures A.8 to A.11).

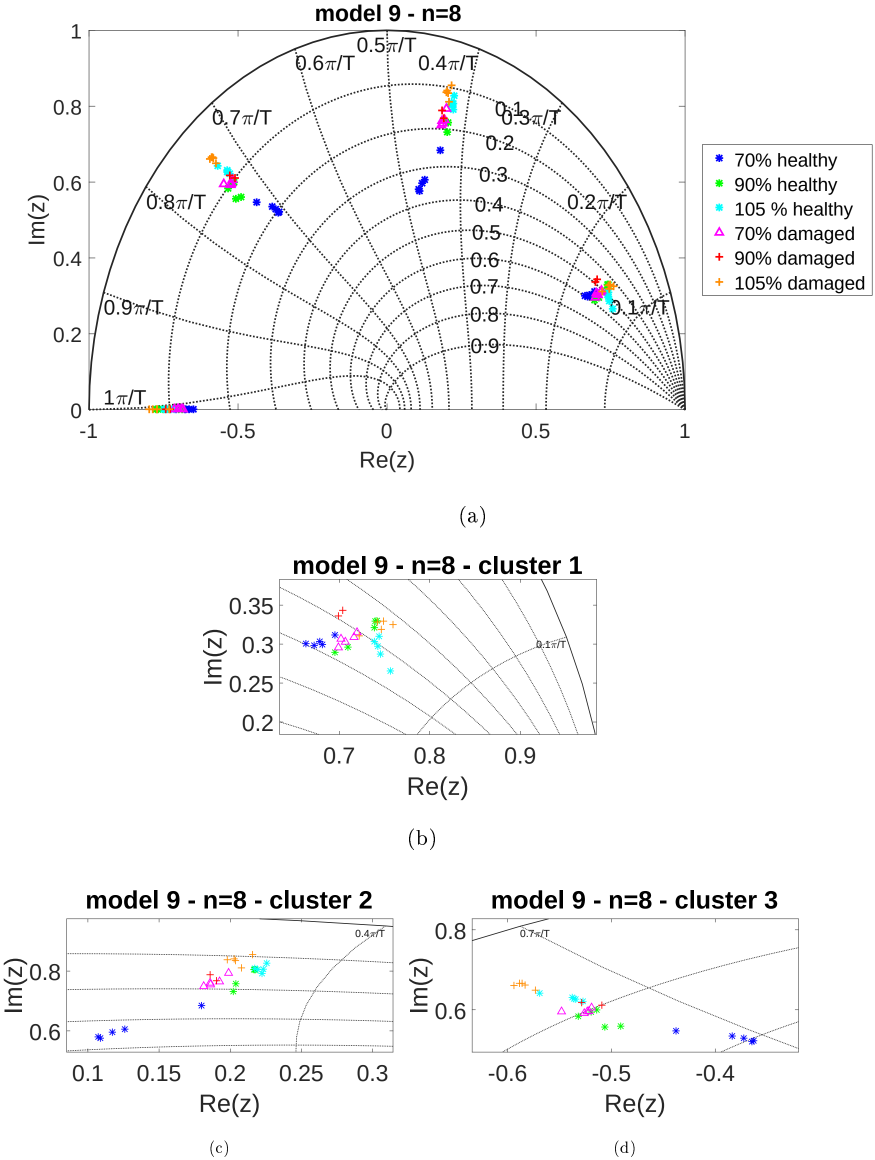
The TF poles of model 9 aggregate in three clusters (see Figure 5.22(a)). Figures 5.22(b) to 5.22(d) provide closeups of these three clusters. Pole migration is observed in all of them but it is more intensive for the second and the third.

Especially these two clusters exhibit significant pole migrations with respect to the baseline state (DEL 70% at the healthy state). Moreover, the pole locations of all the presented load levels and states differ from each other. For instance, the poles of DEL 70% at the healthy state differ from the poles of DEL 70% at the damaged state. In addition, the poles of 90% healthy differ from those of 90% damaged and the poles of 105% with fatigue micro cracks differ from those of 105% damaged. These pole migrations show that damage identification is possible based on the pole positions. Differences between poles of 70% healthy, 90% healthy and 105% fatigue micro cracks show that, despite data standardization, pole locations are affected by load variations (and eventually by structural changes caused by fatigue which cannot be inspected visually). However, pole migrations do not have the same extent in all pole clusters, so the advantage of observing all pole clusters simultaneously is accentuated.

The TF poles of model 10 are presented in Figure 5.23(a). In this case, poles aggregate in four clusters, which are shown in detail in Figures 5.23(b) to 5.23(e). Pole displacements are smaller than those of model 9, but damage identification is still possible, since at least one of the four pole clusters exhibits migrations.

Table 5.9: Optimal model orders of models 1 to 12 for the rotor blade fatigue test database.

model	1	2	3	4	5	6	7	8	9	10
optimal model order	11	10	10	10	10	10	8	8	8	10



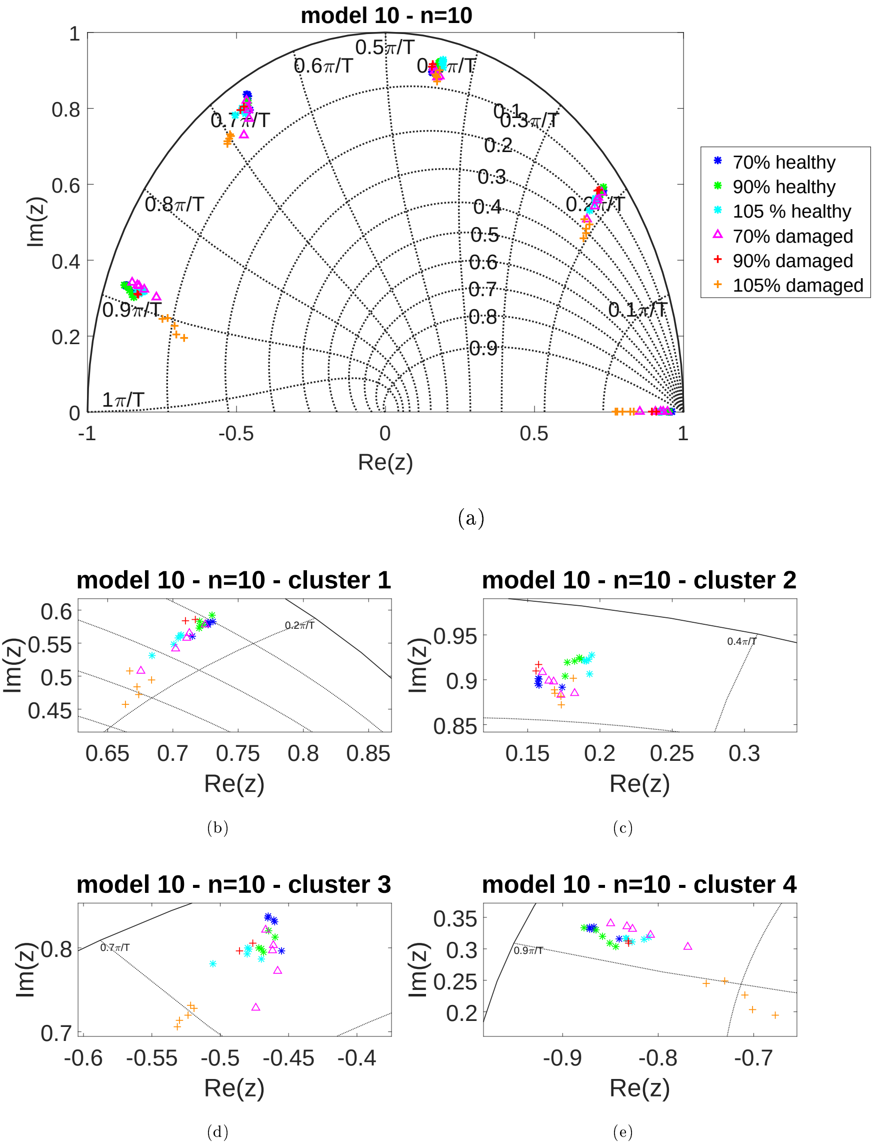
(a)

(b)

(c)

(d)

Figure 5.22: (a) TF poles of model 9 with order 8 for load levels 70%, 90% and 105% at the healthy state (steps 1, 4 and 6) and for the same load levels at the damaged state (steps 13, 14 and 15). Closeups of cluster 1 (at $0.15\pi/T$) (b), cluster 2 (at $0.4\pi/T$) (c) and cluster 3 (at $0.7\pi/T$) (d).



(a)

(b)

(c)

(d)

(e)

Figure 5.23: (a) TF poles of model 10 with order 10 for load levels 70%, 90% and 105% at the healthy state (steps 1, 4 and 6) and for the same load levels at the damaged state (steps 13, 14 and 15). Closeups of cluster 1 (at $0.2\pi/T$) (b), cluster 2 (at $0.45\pi/T$) (c), cluster 3 (at $0.7\pi/T$) (d) and cluster 4 (at $0.9\pi/T$) (e).

5.5 Summary and Conclusions

In this chapter, three databases were analyzed in order to expound TF pole migration and to investigate the effect of various factors on the positions of ARX TF poles. First, the effect of model order was explored. It was shown that the model order significantly affects pole aggregation. Proper selection of the model order is therefore crucial, since it can contribute to eliminating effects which might negatively influence the final detection performance. More specifically, a high level of pole aggregation, i.e., low variance within each pole cluster, is of great importance when evaluating new datasets with respect to pole clusters obtained from baseline datasets in the training phase. The following procedure is proposed for the selection of the optimal model order:

1. Build a SIMO ARX model and compute the MSEs of a training dataset and a validation dataset for a series of model orders.
2. Define a range of potentially optimal model orders from the graph in which the MSEs are plotted against the model order. The optimal model orders are found around the point where the MSE of the training error keeps dropping, while the MSE of the validation set starts to increase.
3. For this range of model orders, calculate the TF poles and plot them on the z-plane.
4. Select the model order which results in the highest level of pole aggregation. This process could be automatized by clustering the TF poles on the z-plane for all model orders obtained during the previous step, and selecting the model order which provides the clusters with the smallest standard deviations.

The effect of other factors was subsequently investigated. The following points summarize the conclusions drawn from the sensitivity analysis, which was performed in this chapter:

- Model order affects pole positions and pole aggregation.
- Proper model order selection enables more effective pole clustering, which facilitates the evaluation of new datasets with respect to clusters obtained in the training phase.
- All n SIMO output-only ARX models, which use one responses of a structure as input and $n-1$ as output have approximately the same optimal model order.
- Poles outside the unit circle imply that the system has an unstable inverse. Unstable inverse systems in discrete time are equivalent to nonminimum-phase systems in continuous time. Such systems move in the opposite direction of the applied load, when subjected to a step load. Thus, they reach the maximum value of the step response function with delay. These poles, which

are common in the case of forced excitation, are not accounted for during the damage detection process.

- It is recommended that real TF poles and TF poles corresponding to nonminimum-phase zeros be removed from further analysis in order to facilitate clustering on the z-plane.
- EOC changes affect TF pole positions, making data normalization or data clustering based on EOCs necessary.
- Damage causes pole migration. Damage of different types and extent results in different TF pole positions. Damage quantification could not be proven.
- Added masses on the structure cause pole migration. Different amounts of added mass result in different pole positions. Quantification of the amount of added mass could not be proven.
- Not all pole clusters exhibit the same sensitivity to structural changes. Hence, parallel observation of all pole clusters is beneficial and enables the capture of different effects.
- There is no correlation between the damage location and the performance of the model with input channels adjacent to the damage location. Therefore, damage localization could not be performed.

These conclusions provide an insight into the phenomenon of pole migration. The information extracted from this preliminary analysis is utilized in the next chapter for the effective deployment of the suggested CP, which is based on TF pole migration (CP^{ARX}).

Chapter 6

Validation of the ARX Transmissibility Function Feature

The concept of monitoring structural changes by observing the TF poles of an output-only ARX model was introduced in chapter 3. A preliminary analysis of this concept was presented in chapter 5, where the sensitivity of TF poles was examined with respect to several factors. The factors considered were the model order, the input channel, as well as the effect of damage and varying EOCs. The concept of observing TF pole migration was presented for application in the context of both supervised and unsupervised machine learning. As for the supervised mode, three metrics were used in order to describe TF pole migration due to structural changes. The Euclidean distance between the cluster mean of the healthy state TF poles and the cluster mean of the current state TF poles was one of the three metrics. However, these metrics are only useful for application in the supervised mode.

One of the main objectives of this thesis is to validate the new CP within an SHM framework which can be employed to detect structural changes when only datasets from the intact structure are used in the training phase of the monitoring process (supervised learning). In this mode, the datasets from the baseline state are used to build TF pole clusters corresponding to the intact structure. Subsequently, the TF poles of a current dataset are evaluated based on the MSD of current TF poles to the clusters of the baseline state. The novel CP will be called CP_d^{ARXi}, where i indicates the channel used as input channel and subscript d the settings selected in hypothesis testing.

In the present chapter, CP^{ARX} is validated on the database obtained from a rotor blade fatigue test. For this purpose, the new damage feature is implemented using the three-tier SHM framework presented in section 1.3.1. The following sections provide a detailed outline of the implementation of CP^{ARX} throughout the three framework tiers. Different models are built by exploiting all sensor signals as input channels. These models are analyzed for different clustering cases, while special focus is placed on examining the distribution of TF poles. The MSD is used to build CP^{ARX}, which is assessed for different confidence regions. The results are presented in ROCs, while AuC values are employed to compare the models obtained from different settings. Finally, the MSEs of the considered models are presented as an additional feature for damage detection.

6.1 The Rotor Blade Fatigue Test

A detailed description of the entire rotor blade test was provided in section 5.3.1, while a thorough description of the fatigue test was given in 5.4.1. Nevertheless, the experimental setup and the database of the fatigue test are briefly reviewed below for clarity reasons.

In June 2014, a 34 m wind turbine rotor blade made of glass fiber/epoxy was tested in order to validate methodologies for damage and ice detection on rotor blades. The blade test included (i) a modal test at the healthy state and at the "iced" state, (ii) a fatigue test in the edgewise direction until the occurrence of damage, and (iii) a modal test after damage occurrence. The blade was instrumented with 12 sensors, which were installed in six positions along the blade length, measuring responses in both the flapwise and edgewise directions (see Figure 6.1). Sensors in positions P1 and P2 were measuring vibration velocity, while sensors in positions P3 to P6 were measuring acceleration. The velocity signals from positions P1 and P2 were then derived to obtain the acceleration signals, which were used in the analysis. Details about the sensor positions and the measured signals are provided in Table 6.1. Due to sensor errors, the signals of the sensor in position P2 were excluded from the analysis, resulting in totally 10 signals.

The blade was excited in the edgewise direction by means of a load frame, which was attached at a section 17 m from the blade root. The applied load was gradually increased until the blade failed at the trailing edge, 6 m from the blade root, after approximately one million loading cycles. Thereafter, the load was reduced and was gradually increased anew. The plan of the fatigue test is outlined in Table 6.2, where the applied loads and the state of the blade are presented. The blade condition during the fatigue test can be divided into four states:

1. Healthy state (steps 1-7) [1]
2. Fatigue microcracks and insignificant damage, i.e., small delaminations and small material or bondline cracks (steps 8-10)
3. Trailing edge damage (end of step 11)
4. Crack propagation (steps 12-17)

The database obtained contained 1167 datasets, with 357 training and 810 testing datasets, each having a duration of 10 minutes. Only datasets from the healthy

[1] The first microcracks due to fatigue were observed after step 7. This phenomenon intensified during steps 8-10. Since it is unknown when exactly during step 7 the formation of the microcracks began, these datasets are regarded as healthy, and only datasets from the beginning of step 7 are used in training.

state of the structure, i.e., from steps 1 to 7, were selected as training datasets. Since it is unknown when exactly during step 7 (load level 110%) the formation of microcracks began, only few data recorded at the beginning of this load step were selected as training datasets. Consequently, the training datasets were composed of 50% of the datasets from steps 1 to 6, and 34 datasets from the beginning of step 7. The test data contained the remaining 50% of the healthy datasets, as well as data from steps 8-17, which correspond to the existence of microcracks caused by fatigue and to damage at the trailing edge. Selecting the test data in this manner allows us to explore the ability of the investigated CP to recognize which datasets belong to the healthy or to an abnormal state. The signals were standardized by subtracting the mean value of the signal and dividing the result by the mean value of the standard deviations of all signals. Thereby, each acceleration signal had zero mean and constant variance throughout the entire fatigue test, while the relationship between the channels was preserved. Datasets were collected at a sampling rate of 1200 Hz but were downsampled to 50 Hz for the analysis.

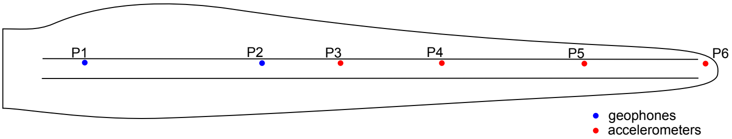


Figure 6.1: Rotor blade with sensor types and sensor positions.

Table 6.1: Sensor positions, sensor types and measured signals.

Sensor position	P1	P2	P3	P4	P5	P6
Distance from root	3.7 m	12 m	15.5 m	20.5 m	27.4 m	34.2 m
Location	inside	inside	inside	outside	outside	outside
Sensor type	geophone	geophone	accelerometer	accelerometer	accelerometer	accelerometer
Signal used in analysis	acceleration	acceleration	acceleration	acceleration	acceleration	acceleration

6.2 Implementation in the Three-Tier SHM Framework

6.2.1 Training Phase

Variation in the prevalent EOCs can cause changes in TF pole positions and, subsequently in CP^{ARX}. In addition to data normalization, which minimizes the effect of varying EOCs, datasets can be assigned to clusters based on their EOCs. In this chapter, these will be called ML clusters, since they are obtained during tier 1 of the SHM framework, which is also referred to as ML[2]. The only varying operational condition is the applied load level, so the different cases occurring during the fatigue test are distinct. Hence, employing automatic clustering methods is not necessary in this analysis. In tier 1 of the SHM framework, two cases of manual clustering were examined: (i) Manual 1 (Man_1), in which the datasets were not clustered and, thus, formed one big ML cluster, and (ii) Manual 2 (Man_2), which refers to clustering according to the load level (7 ML clusters corresponding to steps 1 to 7). However, not all training load levels exist in both the training and the testing phases. In such cases, each test dataset was assigned to the cluster with the closest load level. For instance, load levels 120% to 170% were assigned to the ML cluster containing datasets with load level 110%. The fact that not all load levels are available in both training and

Table 6.2: Plan of rotor blade fatigue test including applied damage equivalent loads (DEL) and corresponding blade state.

Fatigue test	Applied load level	Blade state
Cyclic excitation in edgewise direction (load frame attached to blade)	Step 1 - 70%	healthy insignificant delaminations and cracks
	Step 2 - 75.6%	
	Step 3 - 81.1%	
	Step 4 - 90%	
	Step 5 - 95.6%	
	Step 6 - 105.5%	
	Step 7 - 110%	
	Step 8 - 120%	fatigue microcracks & insignificant delaminations and cracks
	Step 9 - 130%	
	Step 10 - 140%	
	Step 11 - 170%	damage at trailing edge damage length of 440 mm
	Step 12 - 50%	damaged - total damage length not inspected
	Step 13 - 70%	damaged - total damage length not inspected
	Step 14 - 90%	damaged - total damage length of 443 mm
	Step 15 - 105%	damaged - total damage length of 532 mm
	Step 16 - 115%	damaged - total damage length of 554 mm
	Step 17 - 130%	damaged - total damage length of 731 mm

[2] The term "ML cluster" refers to clusters obtained during tier 1 of the SHM framework, while the term “TF pole cluster" refers to the clusters obtained on the z-plane

Table 6.3: Optimal orders for models 1 to 10 for the rotor blade fatigue test database. The number of TF pole clusters remains constant regardless of the clustering case used in tier 1 of the SHM framework.

model	1	2	3	4	5	6	7	8	9	10
optimal model order	11	10	10	10	10	10	8	8	8	10
number of TF pole clusters on upper z-plane	4	3	3	3	3	3	3	3	3	4

testing phases emphasizes the necessity for data standardization before clustering in tier 1.

Since 10 acceleration signals were available, 10 different SIMO output-only ARX models could be built. The optimal model order was determined for each one of the 10 models, as described in section 5.4.2. Table 6.3 shows the selected orders for models 1 to 10. Model 1 has order equal to 11, models 2 to 6 and 10 have orders equal to 10 and model 9 has order equal to 8.

The output-only ARX models were built and the TF poles were calculated. In chapter 3, it was suggested that real TF poles and TF poles corresponding to nonminimum-phase zeros be eliminated before proceeding with the clustering of the solutions on the z-plane. Since the models of the fatigue test did not yield any nonminimum-phase zeros, only the real TF poles were excluded from further analysis. Subsequently, the TF poles of all training datasets were superimposed on the z-plane for each ML cluster obtained during tier 1. That is, for Manual 1, the poles of all training datasets were plotted at the same time on the z-plane, while for Manual 2, the TF poles of the training datasets contained in each of the seven ML clusters were plotted separately. The TF poles were then clustered on the z-plane using k-means clustering, which requires predefinition of the number of clusters. k-means clustering was chosen over AP for this application, since it resulted in better clustering results on the z-plane.

Figure 6.2 shows the clustered TF poles of models 1 to 10 for Manual 1, which refers to all datasets belonging to one big ML cluster. The figure depicts only the conjugate solutions with positive imaginary part, omitting the symmetric solutions. The TF poles of models 1 and 10 aggregate in four clusters in the upper part of the z-plane, i.e., in eight clusters on the entire z-plane. On the other hand, the TF poles of models 2 to 9 aggregate in three clusters in the upper part of the z-plane and, hence, in six clusters on the entire plane. Figure 6.3 exemplarily shows the clustered TF poles of models 1 to 10 for ML cluster 1 of Manual 2, i.e., for the datasets acquired at load level 70%. Figures 6.2 and 6.3 show that the TF pole clusters of Manual 2 exhibit a higher level of aggregation, i.e., have smaller standard deviations. The clustered TF poles of all models for ML clusters 2 to 7 of Manual 2 are given in Appendix B.

It has to be noted that the number of clusters obtained from a model, and subsequently the number of TF poles per dataset, depends only on the selected model order. This implies that the number of TF pole clusters of a model is the same re-

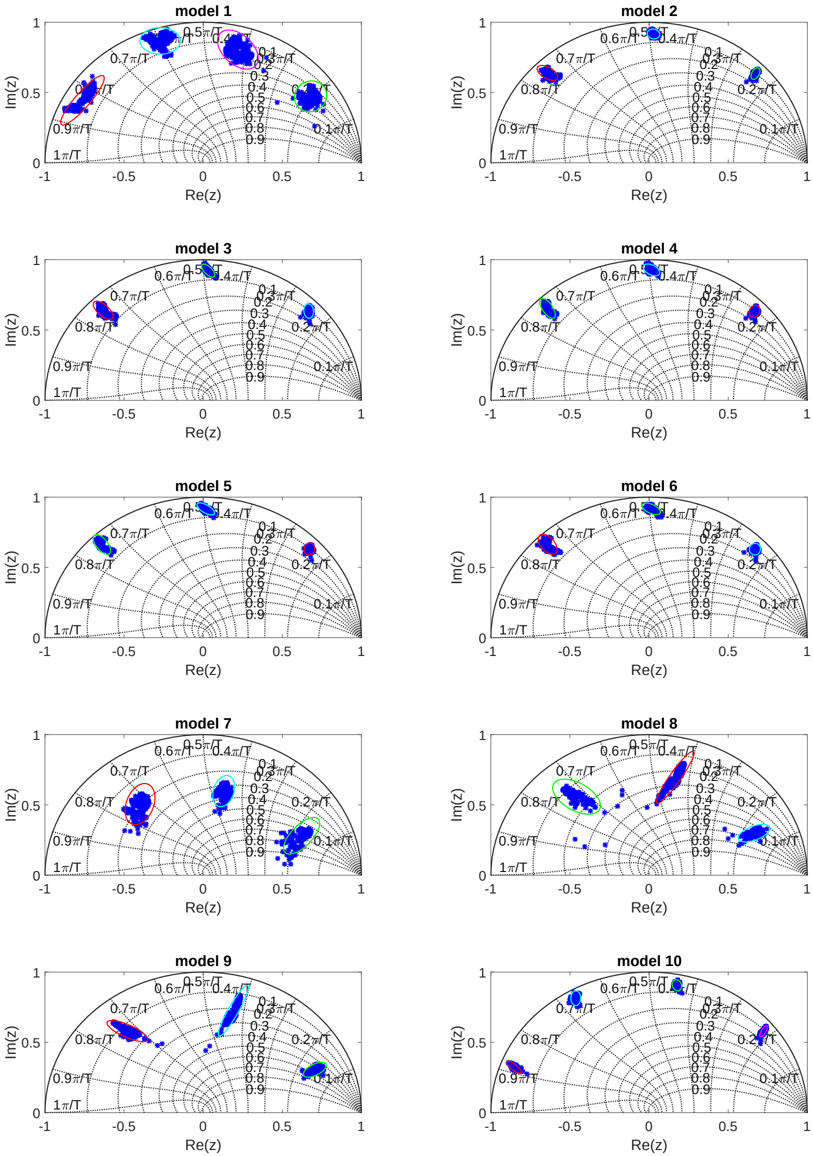


Figure 6.2: TF poles of models 1 to 10 for Manual 1. The TF poles are clustered on the z-plane using k-means clustering.

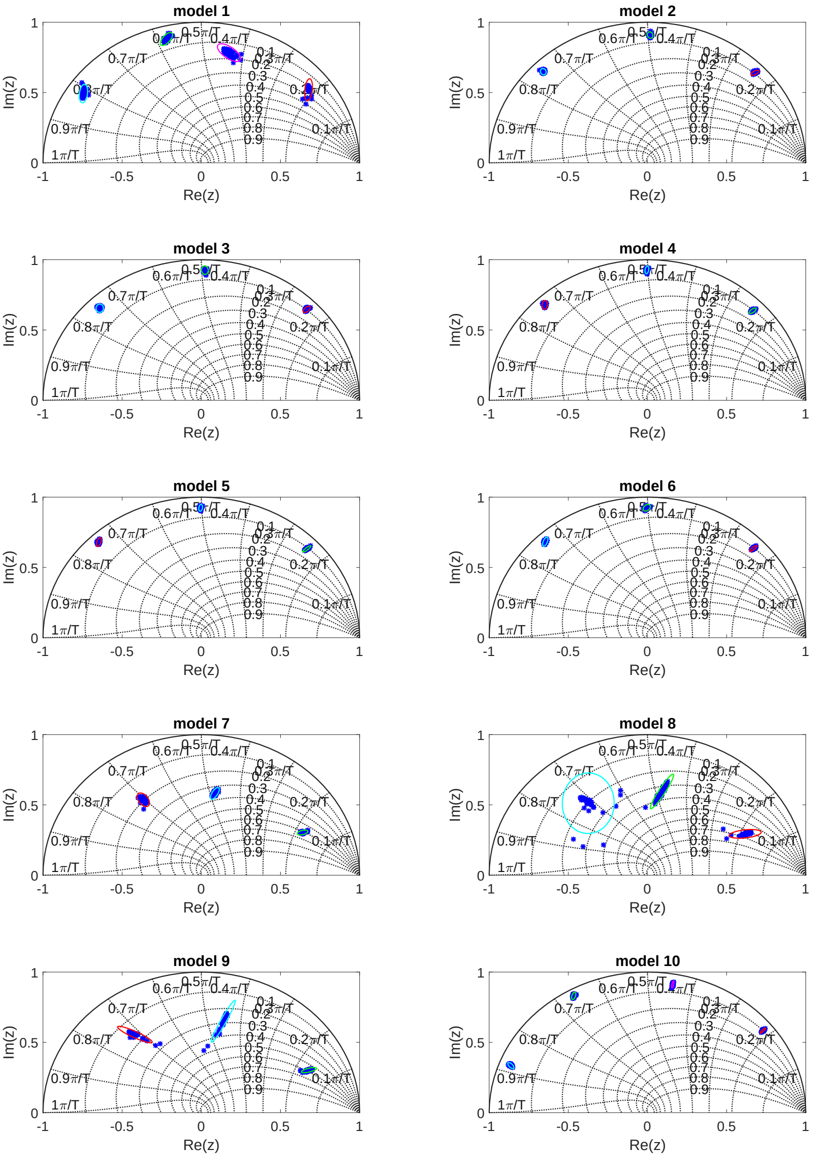


Figure 6.3: TF poles of models 1 to 10 for ML cluster 1 of Manual 2 (i.e., for data of load level 70%). The TF poles are clustered on the z-plane using k-means clustering.

gardless of the clustering case used in tier 1 (see Figure 6.2, Figure 6.3 and figures in Appendix B), while the clustering cases affect the cluster attributes, such as cluster location and standard deviation. Hence, it can be stated that, within a model, each clustering case results in the same number of TF poles, but these clusters have different locations. Moreover, it can be stated that each model results in different TF pole positions, since it expresses different relations between input and output.

Subsequently, the MSD between each TF pole and the cluster it belongs to is calculated. The percentiles of these MSD values serve as decision boundaries for different confidence regions, when assuming in HT that the MSDs of the TF poles within a cluster deviate from the bivariate normal distribution. The decision boundaries for the assumption of normally distributed MSD values of the TF poles are equal to the percentiles of the χ^2-distribution.

The MSDs of the TF poles contained within a cluster were plotted against the quantiles of the χ^2-distribution, in order to examine whether the MSDs follow a normal distribution. Figure 6.4 shows the MSD values of the three clusters of model 6, plotted against the χ^2-distribution quantiles. Line $x = y$ indicates Gaussianity and serves as a reference. That is, if the MSD percentiles of a cluster are located on the $x = y$ line, then the TF poles of this cluster are normally distributed. While the MSD quantiles of TF pole cluster 2 for Manual 2 approximate the quantiles of the χ^2-distribution (MSD values are close to the $x = y$ line), the remaining cases deviate significantly from the quantiles of the χ^2-distribution. In particular, the MSD quantiles of cluster 3 for Manual 1 have high values, showing that the TF poles of this cluster are not normally distributed and have outliers.

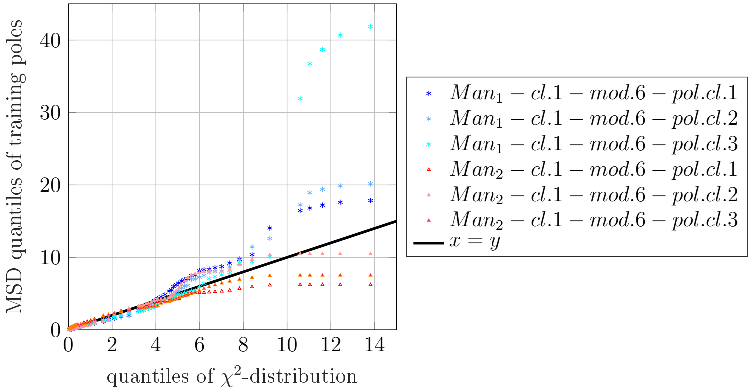


Figure 6.4: MSD percentiles of TF pole clusters obtained in the training phase, plotted against the percentiles of the χ^2-distribution. MSD values correspond to the TF poles of model 6 for Manual 1 and Manual 2.

6.2.2 Testing Phase

In testing, new datasets were evaluated one at a time. They were first assigned to one of the ML clusters obtained in tier 1. The ARX models were then fitted to the test datasets and their TF poles were calculated. Most of the datasets yielded as many TF poles as the number of TF pole clusters obtained in training. In that case, the TF poles were simply assigned to their adjacent TF pole clusters, and their MSDs to the TF pole clusters they had been assigned to, were calculated. Some datasets of models 2 to 9 yielded more TF poles than the existing number of TF pole clusters. As a result, more than one pole was assigned to a specific cluster. For instance, if four TF pole clusters are available from training and the current dataset yields five TF poles, then two TF poles have to be assigned to one of the clusters. In that case, the TF pole with the largest MSD with respect to that specific cluster was retained and the remaining were omitted from the analysis. In this manner, a penalty was given to this dataset for yielding an additional TF pole, while the number of TF poles was maintained equal to the number of clusters.

In HT, two cases were investigated by comparing the MSDs of the test datasets (i) to the quantiles of the χ^2-distribution and (ii) to the quantiles of the MSDs obtained during the training phase. The assessment of a dataset in the testing phase yields as many decisions as the number of clusters in the upper part of the z-plane. For instance, if four TF pole clusters are available from training, four decisions are obtained. These decisions can either be observed separately or can be condensed in one damage index. The decisions of the individual TF poles can be condensed in one damage index in two ways:

1. The decision about the state of a dataset is obtained as the majority vote of the individual TF pole decisions. For instance, let a dataset yield four TF poles which are assessed with respect to the clusters obtained in training. The dataset is identified as damaged if at least three of the four TF poles result in a positive detection (H_1, i.e., the assessment indicates that the TF poles belong to the damaged state).
2. A dataset is identified as damaged if at least one of the TF poles indicates damage. For instance, let a dataset yield four TF poles which are assessed with respect to the clusters obtained in training. The dataset is identified as damaged if at least one of the four TF poles deviates from its baseline cluster and results in a positive detection (H_1, i.e., the assessment indicates that the TF poles belong to the damaged state).

Each of the datasets was assessed for several confidence regions. The results of those analyses were plotted in ROCs, providing an overall picture of the detection performance of CP^{ARX}.

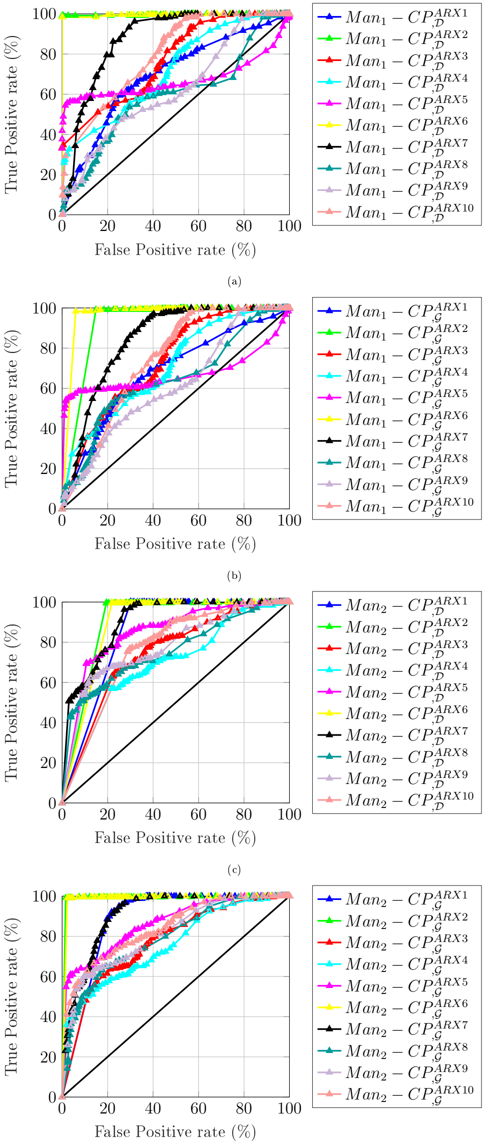
Figures 6.5 to 6.8 show the ROCs for the values of CP^{ARX} calculated under the assumption that a positive detection is obtained, if the assessment of the TF poles indicates damage. The condensed results of all available models are presented for different framework settings. The results of the considered models are denoted in the figure legends as CP_d^{ARXi}, with $i = 1, 2, ..., 10$ indicating the model number and d the distribution assumption ($d = \mathcal{D}$ for the discrete distribution and $d = \mathcal{G}$ for the Gaussian distribution). The ROCs of Manual 1 are presented for the assumption of Gaussianity in subplot (a) and for the definition of decision boundaries based on percentiles in subplot (b). Similarly, subplots (c) and (d) present the ROCs of Manual 2 for the assumption of Gaussianity and for the discrete case, respectively.

The difference between the four figures lies in the definition of the damaged datasets. More specifically:

- in Figure 6.5 datasets after the beginning of load level 120% are regarded as damaged, i.e., after step 7. This includes the early stage of fatigue signs and the damage at the trailing edge (damage occurrence and propagation).
- in Figure 6.6 datasets after the beginning of load level 130% are regarded as damaged, i.e., after step 8. This includes fatigue signs at an intermediate stage and the damage at the trailing edge.
- in Figure 6.7 datasets after the beginning of load level 140% are regarded as damaged, i.e., after step 9. This includes fatigue signs shortly before damage and the damage at the trailing edge.
- in Figure 6.8 datasets after the beginning of load level 170% are regarded as damaged, i.e., after step 10. This includes some datasets prior to the damage at the trailing egde as well as the damage itself.

The ROC plots are used to compare the overall performance of the different framework realizations constructed using various settings along the three tiers. With respect to tier 1, the ROCs of Manual 1 can be compared to those of Manual 2. Figure 6.5 shows that the ROCs of Manual 2 contain points which are located closer to the upper left corner and have higher TP rates. Hence, Manual 2 outperforms Manual 1. Furthermore, the ROCs can be compared with respect to the settings of tier 3, i.e., the definition of decision boundaries based on the assumption of a bivariate normal distribution and based on the quantiles of the MSDs within a cluster. For Manual 1, the selected distribution does not significantly affect the ROCs (see comparison between Figures 6.5(a) and 6.5(b)). This can be attributed to the fact that Manual 1 involves only one big ML cluster. When more datasets are available, the distribution of the TF poles approximates the Gaussian distribution. However, the effect of the selected distribution is obvious for Manual 2 when comparing Figure 6.5(c) to Figure 6.5(d). The ROCs corresponding to decision boundaries based on the discrete distribution exhibit higher TP rates than the ROCs corresponding to the assumption of Gaussian distribution. The aforementioned patterns regarding the settings of tiers 1 and 3 are also observed in Figures 6.6 to 6.8, which are obtained for different definitions of damaged datasets.

Moreover, the ROCs can be observed with respect to the model number, i.e., with respect to the channel used as input in the SIMO ARX model. The ROCs of models 2, 6 and 7 provide the best performance regardless of the case shown in Figure 6.5. This



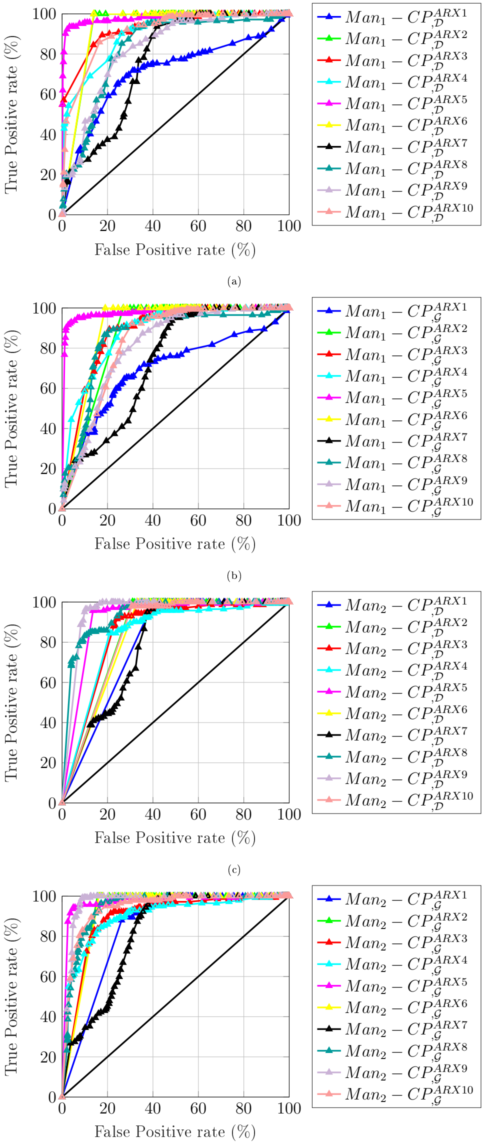
(a)

(b)

(c)

(d)

Figure 6.5: ROCs of CP^{ARX} computed under the assumption that a positive detection is obtained if at least one of the TF poles indicates damage. Datasets after the beginning of load level 120% (step 8) are regarded as damaged. CP^{ARX} of models 1 to 10 shown for (a) Manual 1 and the discrete case in HT, (b) Manual 1 and the Gaussian case in HT, (c) Manual 2 and the discrete case in HT and (d) Manual 2 and the Gaussian case in HT.



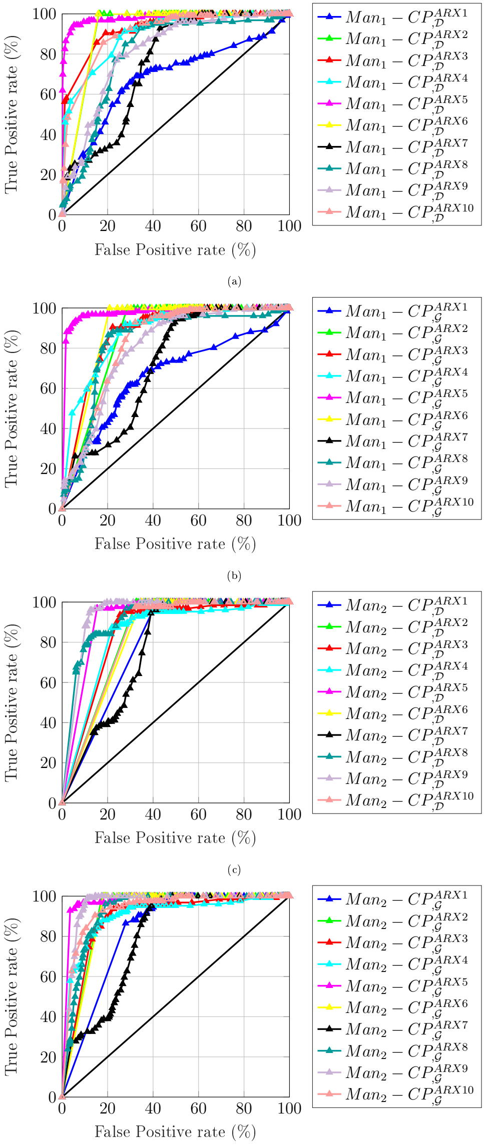
(a)

(b)

(c)

(d)

Figure 6.6: ROCs of CP^{ARX} computed under the assumption that a positive detection is obtained if at least one of the TF poles indicates damage. Datasets after the beginning of load level 130% (step 9) are regarded as damaged. CP^{ARX} of models 1 to 10 shown for (a) Manual 1 and the discrete case in HT, (b) Manual 1 and the Gaussian case in HT, (c) Manual 2 and the discrete case in HT and (d) Manual 2 and the Gaussian case in HT.



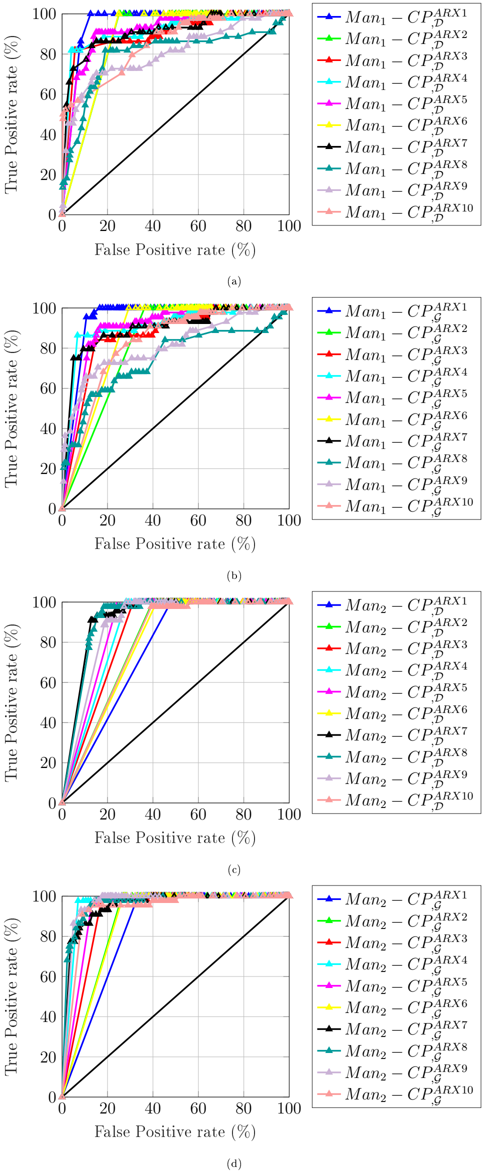
(a)

(b)

(c)

(d)

Figure 6.7: ROCs of CP^{ARX} computed under the assumption that a positive detection is obtained if at least one of the TF poles indicates damage. Datasets after the beginning of load level 140% (step 10) are regarded as damaged. CP^{ARX} of models 1 to 10 shown for (a) Manual 1 and the discrete case in HT, (b) Manual 1 and the Gaussian case in HT, (c) Manual 2 and the discrete case in HT and (d) Manual 2 and the Gaussian case in HT.



(a)

(b)

(c)

(d)

Figure 6.8: ROCs of CP^{ARX} computed under the assumption that a positive detection is obtained if at least one of the TF poles indicates damage. Datasets after the beginning of load level 170% (step 11) are regarded as damaged. CP^{ARX} of models 1 to 10 shown for (a) Manual 1 and the discrete case in HT, (b) Manual 1 and the Gaussian case in HT, (c) Manual 2 and the discrete case in HT and (d) Manual 2 and the Gaussian case in HT.

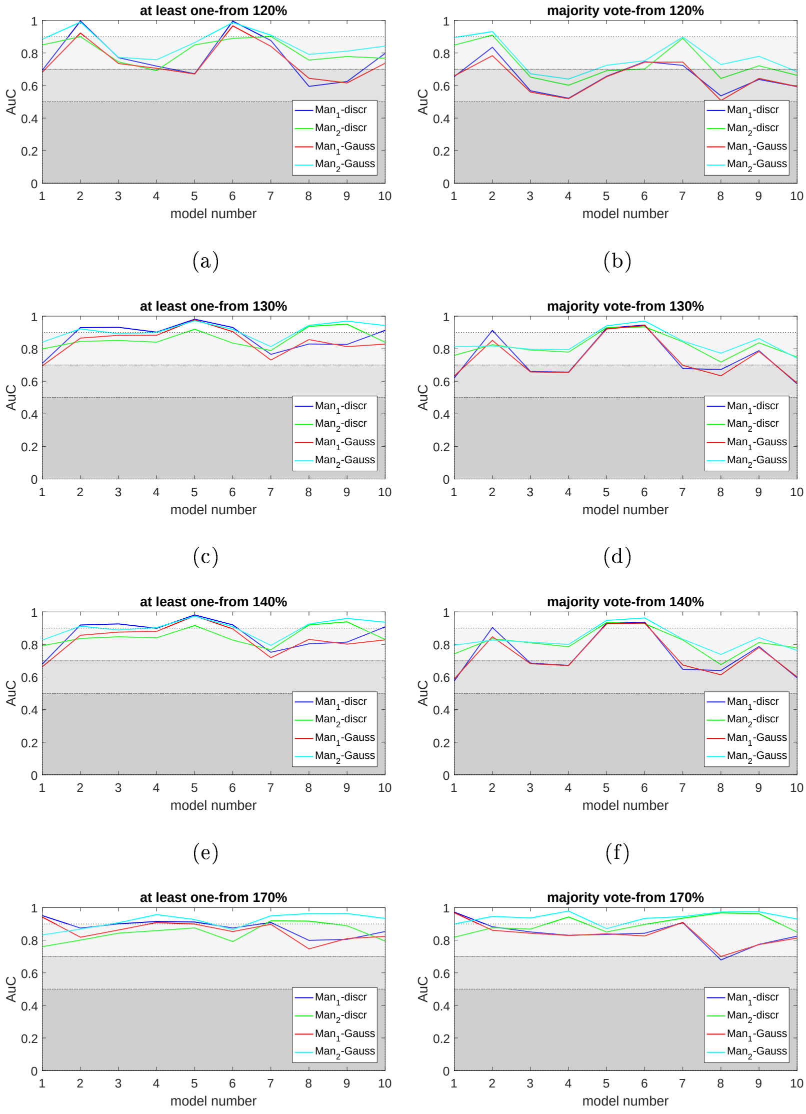
figure depicts the performance of CP^{ARX}, when assuming that damaged datasets contain the initiation of fatigue microcracks (beginning of load level 120%). However, when considering that the damaged datasets start at load level 130%, models 2, 5 and 6 yield the best detections for Manual 1, while models 5, 8 and 9 yield the best detections for Manual 2 (see Figure 6.6). The same holds for the damaged datasets starting at load level 140% (see Figure 6.7). Finally, Figure 6.8 shows that models 1, 4 and 5 give the best ROC curves for Manual 1, and models 7, 8 and 9 for Manual 2. Hence, it can be concluded that the models which deliver the best detections vary depending on the type of changes the structure is subjected to. Some models (i.e., some input channels) deliver better detections when the structure is subjected to changes due to fatigue, and other models detect damage at the trailing edge with higher accuracy.

The area under the ROC curves (AuC) is a metric which allows for the comparison of two ROCs. Observation of the ROCs in the four consecutive figures reveals that AuC values increase from Figure 6.5 to Figure 6.8. As a result, the conclusion can be drawn that CP^{ARX} is capable of detecting both the fatigue signs and the damage at the trailing edge, but performs better at detecting the latter. This observation is quantified in Figure 6.9, which presents the values of AuC for models 1 to 10. The first column of Figure 6.9 shows the AuC values of the ROCs described above. These correspond to detections which are positive, if at least one of the TF poles of the tested dataset is identified as damaged. The second column corresponds to detections obtained as the majority vote of the detections of all the TF poles of a dataset. Both columns contain the four variations of damaged datasets definition rowwise, i.e., results for the damaged datasets from load levels 120%, 130%, 140% and 170% are presented in rows 1 to 4, respectively. Each subplot contains the AuC values for all combinations between the settings of tier 1 and tier 3 (Manual 1 and Manual 2 for tier 1, and Gaussian and discrete for tier 3).

When detection is obtained if at least one of the TF poles is identified as damaged, AuC values increase as the assumption of damaged datasets moves from load level 120% to load level 170% (see first row of Figure 6.9). More specifically, when the boundary between healthy and damaged datasets lies at the beginning of load level 120%, AuC values range between 0.6 and 1. It is reminded that AuC values below 0.7 indicate low accuracy, while values greater than 0.9 indicate high accuracy. AuC values for the boundary between healthy and damaged datasets lying at the beginning of 130% and 140% are slightly better, ranging from 0.67 to 1. Finally, when the boundary between healthy and damaged datasets lies at the beginning of load level 170%, the AuC has values greater than 0.72, with many being approximately 0.9.

This pattern is also observed when the detection regarding a dataset is obtained as the majority vote of all TF pole detections (second column of Figure 6.9). However, in this case, the AuC values in the second column are slightly lower than those in the first column, showing that the fusion of all the TF pole detections in one detection as the majority vote of all TF the pole detections is less efficient than the other option.

The conclusion can therefore be drawn that not all TF poles of a damaged



at least one-from 120%

majority vote-from 120%

(a)

(b)

at least one-from 130%

majority vote-from 130%

(c)

(d)

at least one-from 140%

majority vote-from 140%

(e)

(f)

at least one-from 170%

majority vote-from 170%

(g)

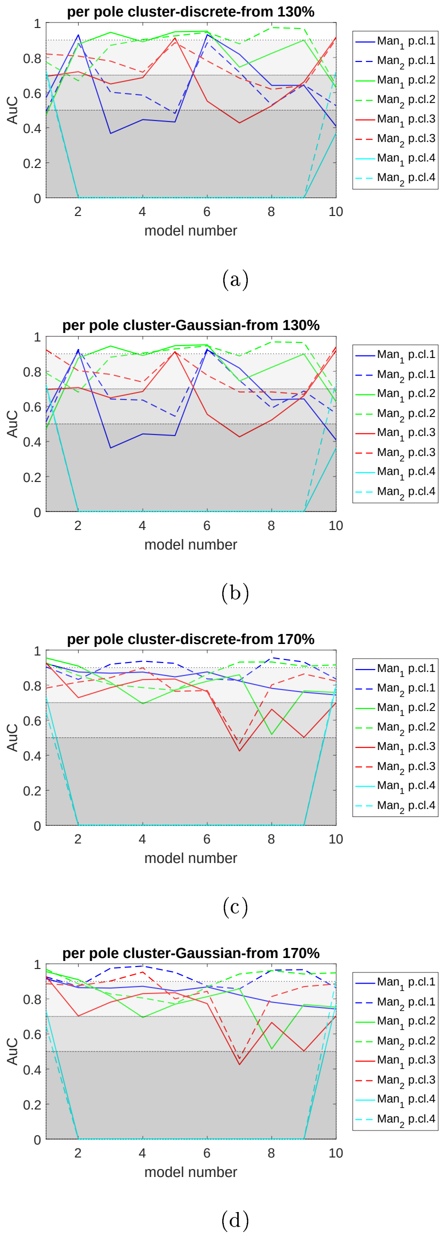
(h)

Figure 6.9: AUCs of models for Manual 1 and Manual 2 in tier 1, and discrete and Gaussian case in HT. Left column: a positive detection of CP^{ARX} yields, if at least one of the TF poles of the test dataset indicates damage. Right column: hypothesis of CP^{ARX} given as the majority vote of the individual decisions of the TF poles. Various assumptions of damaged datasets (starting from 120%, 130%, 140% and 170%) presented rowwise. $AuC > 0.9$ indicates high accuracy, $0.7 < AuC < 0.9$ indicates mediocre accuracy and $0.5 < AuC < 0.7$ indicates low accuracy.

dataset yield detections. In general, it can be stated that less than 50% of the TF poles of a damaged dataset indicate divergence from the healthy state. This can be seen in Figure 6.10, which shows the performance of the individual TF pole clusters for both cases of HT. The upper subplots present the AuC values of the models for the assumption that the damaged datasets start at the beginning of load level 130%. The left subplot depicts the results for defining decision boundaries in HT based on the quantiles of the MSD values, while the right subplot shows the results when assuming Gaussianity. The two subplots show that TF pole cluster 2 offers on average the highest AuC values. At the same time, it can be seen that no TF pole cluster yields high AuC values for all the models simultaneously. AuC values of TF pole 4 are equal to zero for models 2 to 9, since, as it was shown earlier, four clusters are only available for models 1 and 10.

The lower subplots of Figure 6.10 show the AuC values of models 1 to 10 for the assumption that the damaged datasets start at load level 170%. These subplots show that only the TF pole clusters 1 and 2 of Manual 2 deliver high AuC values for all considered models. The other curves do not indicate high AuC values. These results emphasize that it is advantageous to design CP^{ARXi} to return positive detection if at least one of the TF poles yields detections.

The control charts in Figures 6.11 and 6.12 present the MSEs for the discrete case in HT. The results of Manual 1 are shown in Figure 6.11, while the results of Manual 2 are given in Figure 6.12. The UCL and LCL correspond to the 5^{th} and the 95^{th} percentiles of the MSE of the output-only ARX models which were fitted on the training datasets. The figures are divided into subplots for clarity, with the first subplot containing steps 1 to 10 of the fatigue test and the second subplot containing steps 11 to 17. The MSE for Manual 1 exhibits significant changes from the beginning of load level 105% (see Figure 6.11(a)). On the other hand, the MSE for Manual 2 exceeds the LCL after load level 130%. The model MSE can be used in addition to CP^{ARX} in order to obtain a better insight into the changes observed in the TF pole locations.



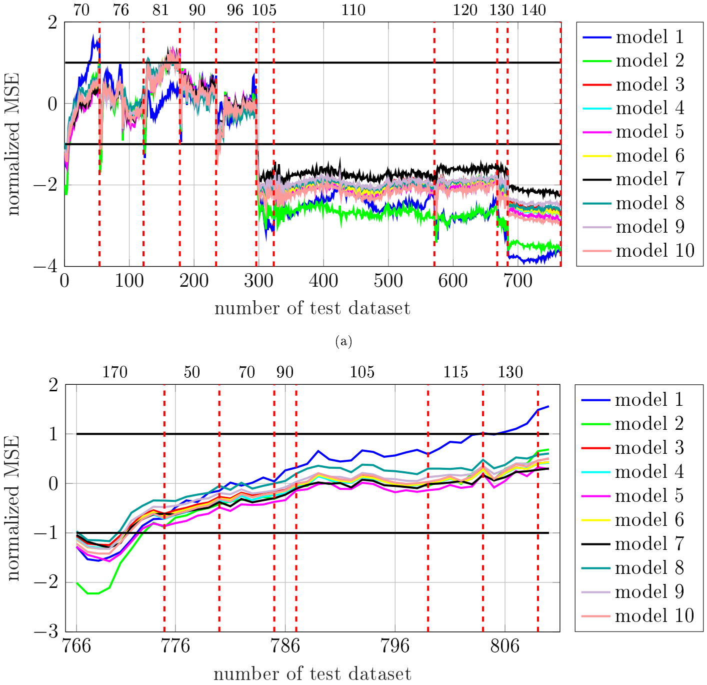
(a)

(b)

(c)

(d)

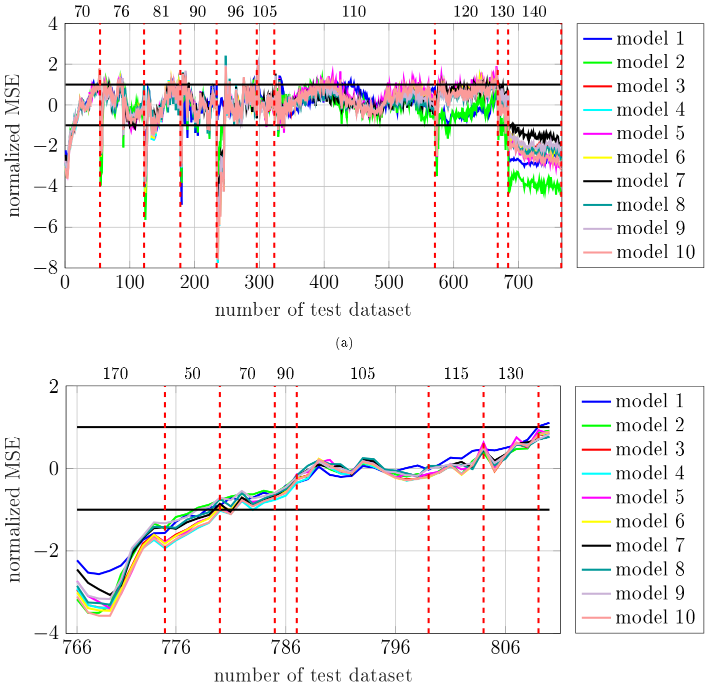
Figure 6.10: AuC values of individual TF pole clusters of models 1 to 10 for Manual 1 and Manual 2 in tier 1, and for discrete and Gaussian cases in HT. AuCs assuming that damaged datasets start at 130% for the discrete case (a) and the Gaussian case (b). AuCs assuming that damaged datasets start at 170% for the discrete case (c) and the Gaussian case (d).



(a)

(b)

Figure 6.11: Model MSEs for Manual 1 and for discrete distribution in HT. (a) Steps 1 to 10 and (b) steps 11 to 17 of fatigue test. Load levels shown over the figure. Upper control limit (UCL) and lower control limit are indicated by the horizontal lines at $y = 1$ and $y = -1$. UCL and LCL given as the 5^{th} and 95^{th} percentiles, respectively.



(a)

(b)

Figure 6.12: Model MSEs for Manual 2 and for discrete distribution in HT. (a) Steps 1 to 10 and (b) steps 11 to 17 of fatigue test. Load levels shown over the figure. Upper control limit (UCL) and lower control limit are indicated by the horizontal lines at $y = 1$ and $y = -1$. UCL and LCL given as the 5^{th} and 95^{th} percentiles, respectively.

6.3 Summary and Conclusions

In this chapter, the proposed damage feature (CP^{ARX}) was implemented for monitoring vibrating structures with unsupervised learning algorithms. The concept was integrated into an SHM framework which is composed of three tiers: (i) data clustering according to the prevailing EOCs, (ii) the extraction of damage-sensitive features and (iii) hypothesis testing. CP^{ARX} was implemented within this framework and was validated on the data obtained from a full-scale rotor blade fatigue test. The fatigue test involved excitation in the edgewise direction for over one million loading cycles. The experiment concluded with a damage at the trailing edge, while microcracks attributed to fatigue were observed prior to blade failure.

The implementation of CP^{ARX} in the training phase of the monitoring framework can be summarized as follows:

- In tier 1, two clustering cases were examined: (i) Manual 1, which did not involve data clustering and (ii) Manual 2, which involved manual clustering based on the load level applied during the experiment.
- In tier 2, the ten acceleration signals were used one at a time as input channels of an output-only ARX model which predicted the other nine structural responses. The transmissibility functions (TF) of the models were built and their TF poles were calculated. The TF poles of the datasets included within each ML cluster obtained in tier 1 were superimposed on the z-plane and clustered using k-means clustering. Subsequently, the MSD of each TF pole to the cluster it belonged to was calculated. These MSD values constitute the values of CP^{ARX} for the training datasets.
- In hypothesis testing (tier 3), two cases were investigated. The first assumed that TF poles contained within a cluster followed the bivariate normal distribution. Decision boundaries were defined as the percentiles of the χ^2-distribution, since the MSDs of the data points following a Gaussian distribution follow the χ^2-distribution. The second assumed that TF poles contained within a cluster were not normally distributed. Decision boundaries were defined as the percentiles of the MSD values obtained in training.

Some of the TF clusters were exemplarily analyzed with respect to their distribution. The MSD values of the TF pole clusters were compared to the percentiles of the χ^2- distribution in order to investigate whether the TF poles followed a bivariate normal distribution. It was shown, that while some TF pole clusters approximated the Gaussian distribution, there were also clusters which deviated significantly from Gaussianity. Therefore, investigating both cases in HT is essential for defining the appropriate decision boundaries and obtaining more accurate detections.

In testing, each test dataset was analyzed for clustering cases Manual 1 and Manual 2, and was assigned to one of their ML clusters (tier 1). Subsequently, the output-only ARX models were fitted to the test datasets and their TF pole clusters were calculated. Those were assigned to one of the TF pole clusters, which had been derived in the training phase, using k-means clustering. The MSDs of the testing TF poles to the clusters they had been assigned to were calculated and were assessed with respect to the two settings of HT for a series of confidence regions. Finally, the hypotheses of the individual TF poles of test dataset were fused into one hypothesis (i) by generating the hypothesis of CP^{ARX} as the majority vote of the decisions regarding the individual TF poles and (ii) by generating one positive detection for CP^{ARX}, if at least one of the TF poles indicated damage.

The results were plotted in ROCs in order to evaluate the overall detection performance of CP^{ARX} for different framework realizations. Futhermore, the AuC values of the presented ROCs were used to compare the framework realizations analyzed. Since the fatigue experiment included structural changes due to fatigue and due to damage at the trailing edge, various definitions regarding the damaged datasets were examined. The ROCs of the first case involving fusion of decisions were presented in detail (i.e., for generating one positive detection for CP^{ARX}, if at least one of the TF poles indicated damage). Some models of Manual 1 which were evaluated for the discrete case in HT provided the perfect detection, i.e., 100% TP rate and 0% FP rate, while the remaining models had lower AuC values. That occurred when assuming that the damaged datasets contained an early stage of fatigue. However, the overall performance of the models improved as the border of the damaged datasets moved from the early fatigue stage to the damage at the trailing edge. For instance, detecting fatigue microcracks at an intermediate stage and the damage at the trailing edge was performed simultaneously by all the models with higher accuracy than in the aforementioned case. Consequently, CP^{ARX} is capable of detecting structural changes due to fatigue as well as damage at the trailing edge.

The models which perform well for the early fatigue signs differ from those which perform well in detecting the damage at the trailing edge. While no conclusion could be drawn with respect to which channels and, consequently, which directions and locations of measurement yield the best results, this observation emphasizes the need for investigating all possible models obtained from different input signals. The decisions of the individual output-only ARX models could be exploited in order to build a single CP. This could be achieved in the same manner as for the TF poles, i.e., either by obtaining the decisions as the majority vote of the individual models or by obtaining one positive detection, if at least one of the models indicates damage.

Moreover, it was shown that the performance of realizations involving no data clustering according to EOCs (Manual 1) are independent of the decision boundaries set in HT. That is, the performance of Manual 1 is independent of whether the decision boundaries are set based on the discrete or the Gaussian assumption for the TF poles contained within a cluster. For Manual 2, though, the case considered in HT affects the detection performance. While the AuCs for both the discrete and the Gaussian case of Manual 2 are high, it is the discrete case which offers significantly higher TP rates.

AuC values were presented for both cases considered in HT, for both decision fusion cases and for three considerations regarding the definition damaged datasets. In the first case, i.e., when the damaged datasets included fatigue signs at an early stage and the damage at the trailing edge (load level 120%), the AuC values ranged from 0.58 to 1. When assuming that the damaged datasets included fatigue signs at an intermediate stage and the damage at the trailing edge (load levels 130% and 140%), the AuC values ranged between 0.68 and 1. Finally, when assuming that the damaged datasets contained damage at the trailing edge, AuC values were greater than 0.7, with the majority of AuC values lying around 0.9. The other fusion case which generated the hypothesis of CP^{ARX} as a majority vote yielded lower AuC values than those of the first fusion case. In addition, the AuC values of the individual TF pole clusters were presented, showing that the TF poles of a damaged dataset did not simultaneously exhibit variations due to structural changes. Hence, it is suggested that CP^{ARX} be designed to return a positive detection when at least one of the TF poles deviates from the baseline state.

Summarizing, it has been proven that the deployment of the novel CP in unsupervised mode is possible and yields good detection performances for real structural changes of different types, i.e., for detecting structural damage (such as damage at the rotor blade trailing edge) and for detecting structural changes due to fatigue. Furthermore, this chapter validates the concept of implementing CP^{ARX} within the context of an SHM framework, which takes into account different aspects, such as data normalization and different approaches to the definition of decision boundaries.

Chapter 7

Validation of AdaBoost on the Tower of a 3 kW Wind Turbine

The concept of employing adaptive boosting (AdaBoost) in order to build a more effective decision rule by combining the decisions of various damage features was presented in Chapter 4. The concept was implemented within the context of a three-tier SHM framework, which results in several realizations depending on the settings of each tier, i.e., (i) the settings used for data clustering based on the EOCs, (ii) the investigated damage feature and (iii) the settings defining the decision boundaries in hypothesis testing. In the current chapter, the integration of AdaBoost with the SHM framework is validated on the support structure of an operational 3 kW wind turbine located in Los Alamos National Laboratory (LANL).

The rotor blade fatigue test database, which was analyzed in the previous two chapters, involved damage at the trailing edge and structural changes due to fatigue. The distinction between the healthy datasets and the damaged datasets can be subjective for this database. For instance, one might consider that only the damage at the trailing edge constitutes damage, or that structural changes due to fatigue also constitute damage. On the other hand, the LANL wind turbine structure was subjected to reversible damage, while the healthy state and the damaged state are totally distinct. This database was chosen over the database of the rotor blade fatigue test for the validation of AdaBoost, since, in this first application of AdaBoost, it is important to have a clear and objective distinction between healthy and damaged datasets.

AdaBoost constitutes a supervised learning algorithm. As any machine learning algorithm, AdaBoost requires a sufficient number of training datasets in order to yield a representative estimate of the algorithm's performance. The database of the 3 kW wind turbine contains relatively few damaged datasets. Therefore, a numerical model of the wind turbine was created by the Laboratory for Intelligent Systems Technology (LIST) at the University of Michigan and was used to generate structural responses. The generated data were employed in order to implement the SHM framework and, subsequently, AdaBoost, allowing for the investigation of AdaBoost's performance while excluding the negative effect of having few datasets.

Both the synthetic and the operational database were analyzed using the SHM framework for different settings along the three tiers. Thus, several realizations of the SHM framework occurred with different detections and performances. Four realizations of the synthetic database were selected and were combined pairwise. The boosting performance was assessed by comparing: (i) the TP rate and the FP rate of the SHM framework to those of AdaBoost and (ii) the error rate of the SHM framework to the error rate of AdaBoost. Furthermore, an investigation was carried out to identify the attributes of those sets which resulted in boosting or yielded better boosting results. The findings of this investigation were used to define a way of achieving better boosting results by omitting CPs that correspond to specific EOC clusters.

The same analysis was carried out for four realizations of the operational database. The low number of datasets contained in this database raised the need for cross-validation. Hence, k-fold cross-validation was employed in order to obtain a more representative estimate of AdaBoost's prediction error. Furthermore, the analysis was extended to the combination of three framework realizations.

7.1 Experimental Setup and Analyzed Databases

Experimental data from the LANL 3 kW wind turbine was used to validate the integration of AdaBoost into the three-tier SHM framework. The initial purpose of that measurement campaign was to verify and test the Martlet wireless sensor nodes in situ, as well as to apply the triangulation-based extraction of modal parameters (TEMP) and the aforementioned three-tier SHM framework in operation. A detailed description of the Martlet wireless sensors can be found in [105]. The TEMP algorithm is presented in [17], while the analysis of the LANL turbine data with the use of the SHM framework can be found in [36].

The LANL wind turbine is a Whisper 500 two-bladed turbine with a 12 m high steel support structure. The turbine has a rotor diameter of 4.5 m and rated power of 3 kW at 10.5 m/s wind speed. The operational wind speeds of the system range from 3.4 to 55 m/s, while the normal rotor speed is at 500 rpm (8.2 Hz). For maintenance and experimental purposes, the tower can be tilted around a pivot point located at one third of the tower height. During the experiment, damage was introduced to the tower by modifying the bolt used to fix the support structure. The bolt was replaced by a longer one with a spring. Hence, the boundary conditions were slightly modified from fixed to only partially fixed and nonlinearities were introduced.

Three sensors were installed above and three below the pivot point of the tower (see Figure 7.1). Each of the six sensor nodes consisted of a tri-axial acceleration sensor recording tower lateral accelerations in two orthogonal directions, with the third direction being neglected. Datasets were collected at a sampling rate of 500 Hz and partially at 1 kHz. Information about the weather conditions was obtained from a weather station located 50 m north of the turbine. It included 15-minute mean values and standard deviations of wind speed, wind direction and temperature.

Datasets were collected at the undamaged state and under two abnormal conditions, which were applied both separately and simultaneously. The first involved bolt loosening for the modification of the boundary conditions and the second involved artificially induced break scenarios. 367 datasets were collected in total, with 325 corresponding to the healthy structure and 42 corresponding to the damaged structure. The main wind direction (south and south-southeast) caused the tower to lean against the substructure rather than against the bolt or spring. Therefore, the damage severity is asssumed to be very small for the overall dynamic behavior. Moreover, during a significant part of the measurement, the wind speed was relatively low, resulting in a lower excitation level. A detailed description of the database can be found in [36].

Some drawbacks of this measurement campaign are (i) the limited EOC collection (e.g., lack of information regarding the rotor speed, which has a dominating effect), (ii) the weakness of the induced damage scenario and (iii) the fact that only 15-minute mean values of the environmental data were available, while the system is subject to very fast rotor speed and nacelle orientation variations. At the same time, however, it is due to these experimental setup attributes and the potentially entailed mediocre detection performance that this example is ideally suited for the application of boosting techniques.

The experimental data contained only 42 damaged datasets. This number is relatively low for machine learning applications. A numerical model of the 3 kW turbine was built in order to perform a preliminary study on the integration of AdaBoost with the SHM framework with sufficient data. The model was used to generate datasets corresponding to the responses of the real structure. The integration of AdaBoost with the SHM framework was first validated on the model and, subsequently, on the operational data of the real structure.

7.2 Validation on Simulated Data of a 3 kW Wind Turbine

The numerical model for the LANL wind turbine was provided by the Laboratory for Intelligent Systems Technology (LIST) of the University of Michigan [32]. The model was constructed using FAST (Fatigue, Aerodynamics, Structures and Turbulence) [106], TurbSim [107], and MATLAB. FAST is a software developed by the National Renewable Energy Laboratory (NREL) and is particularly useful for modeling and analysis of wind turbines. TurbSim is another software released by NREL which is capable of modeling full-field turbulent wind structures.

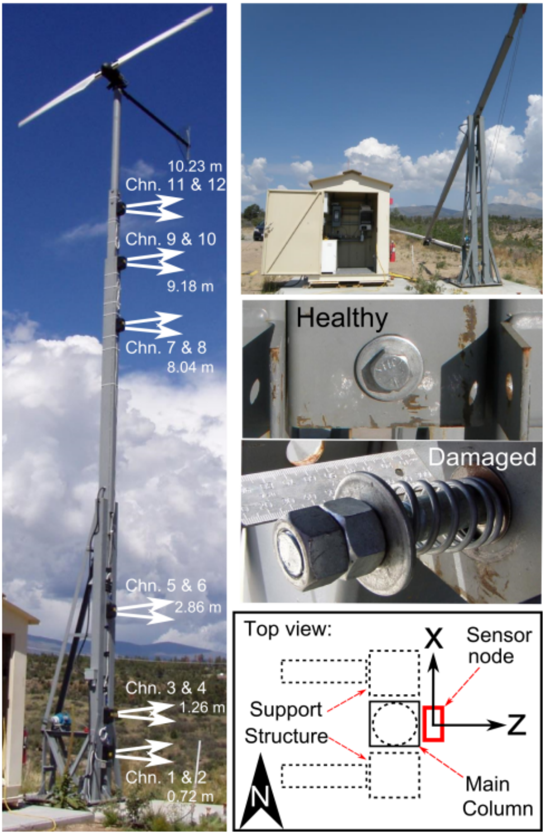


Figure 7.1: The LANL wind turbine.

FAST uses cantilever beams to model the turbine tower, since almost all large-scale wind turbines are supported by cantilever towers. However, the support structure of the LANL wind turbine involves unique two-pin supported boundary conditions and cannot be modeled by FAST. Therefore, a numerical model of the structure was built in MATLAB. Wind statistics of the LANL site were retrieved from an online meteorological database[1] and were input to TurbSim to generate full-field stochastic time series. Those time series were then used as inputs in FAST in order to simulate the response of the nacelle. The nacelle support loads were used as input to the MATLAB tower model for the computation of the tower response. The model had been designed to output accelerations at the same locations as the sensors installed on the real wind turbine. The calculation of the tower responses based on the na-

[1] http://environweb.lanl.gov/weathermachine/

celle loads obtained by FAST is a simplifying assumption which results in reduced accuracy. Another simplification concerns the rotor blades. The blade profile and the mass distribution in the model are approximations of blade profile and the mass distribution of the real Whisper 500, since the exact properties were not available. However, this representation is sufficient for the purpose of this study, since the consideration of the model intended primarily to generate a sufficient number of datasets from a model that approximates as much as possible the experimental structure.

The synthetic dataset consisted of 1400 undamaged datasets for the training phase as well as 1200 undamaged datasets and 900 damaged datasets for the testing phase. The datasets were generated with a sampling frequency of 500 Hz and each one had a duration of 30 seconds. The damage scenario consisted of replacing the rigid pin support of the tower in x-direction with a spring (see Figure 7.1), thus being similar to the experimental damage scenario.

7.2.1 Analysis using the SHM Framework

The synthetic datasets were downsampled to 100 Hz and analyzed using the SHM framework for various clustering cases and CPs. In the first tier, the structural responses were clustered according to yaw angle and rotor speed values provided by the model. Four clustering cases were investigated: Man1, Man2, Man3 and AP1. In Man1, all datasets were contained in one cluster, while Man2 and Man3 involved manual clustering which resulted in 9 and 26 clusters, respectively (see Figure 7.2). For AP1, affinity propagation was employed resulting in 53 clusters. Affinity propagation (AP) is a clustering algorithm which uses measures of similarity between pairs of data points and exchanges information between them in order to build clusters. These measures of similarity are dubbed preferences. The algorithm considers all data points as potential exemplars and, therefore, does not require a predefined number of clusters [37]. All the clustering cases and the corresponding number of clusters are given in Table 7.1. The cluster geometries on the rotor speed - yaw angle space are shown in Figure 7.2.

In the second tier, three residual CPs were calculated: CP^M and CP^{R^2}, which derive from a vector autoregressive (VAR) model, and CP^ϵ, which derives from the stochastic subspace identification approach (SSI). CP^M makes use of the VAR model error covariance matrix and is based on Box's M-Test, which compares the error covariance matrices of two data instances, namely the error covariance matrix of the dataset to be tested, and that of the reference dataset [71]. CP^{R^2} employs the coefficient of determination according to Neter, which makes use of the actual time series values (e.g., acceleration signals), the model estimation and the mean value of the measured signal [70]. The last residual CP, CP^ϵ, is proposed by Baseville [68] and is based on the interpretation that the singular vectors matrix U, which is obtained by the singular value decomposition of the covariance Hankel matrix, is the left nullspace of the Hankel matrix. Thus, the Hankel matrix and U are used to build a residue.

In the third tier, decision boundaries were defined using both the assumptions

Table 7.1: EOCs accounted for clustering, settings and resulting number of clusters for the four clustering cases used in the analysis of simulated data using the SHM framework.

Clustering case	Manual 1	Manual 2	Manual 3	AP 1
EOCs for clustering	-	wind speed/rotor speed	wind speed/rotor speed	wind speed/rotor speed
Number of clusters	1	9	26	53

Table 7.2: Database information and SHM framework settings for the analysis of the synthetic data.

Number of channels	12			
Sampling frequency	100 Hz			
Number of datasets	3500			
Training datasets	1400			
Test datasets	2100			
Clustering cases	Man 1	Man 2	Man 3	AP1
Number of clusters	1	9	26	53
Settings for condition parameters				
CP^M, CP^{R^2}	AR-model order=14			
CP^ϵ	time shifts=4 significant columns=24			
Analyzed distribution types: discrete and Gaussian a-values for hypothesis testing: a=[0.5:0.5:10, 11:1:20, 25:5:45]%;				

of the Gaussian and discrete distributions for the CPs. The combinations of the aforementioned tier settings resulted in different realizations of the SHM framework, which were evaluated for 48 confidence intervals. Table 7.2 summarizes the database information and analysis settings. The ROC curves of CP^ϵ, CP^M and CP^{R^2} for all possible realizations are presented in Appendix C (see Figure C.1).

AdaBoost is capable of combining only damage features with the same dimensions. For instance, a one-dimensional CP can be combined only with another one-dimensional CP and a two-dimensional CP can be combined only with another two-dimensional CP. This chapter focuses on the simplest case, i.e., on the combination of one-dimensional features by examining the aforementioned CPs (CP^ϵ, CP^M and CP^{R^2}). The new damage feature which was introduced in the previous chapter (CP^{ARX}) is not considered in this chapter, since it is two-dimensional. However, it would be possible to employ AdaBoost for combining CP^{ARX} values obtained from different output-only models.

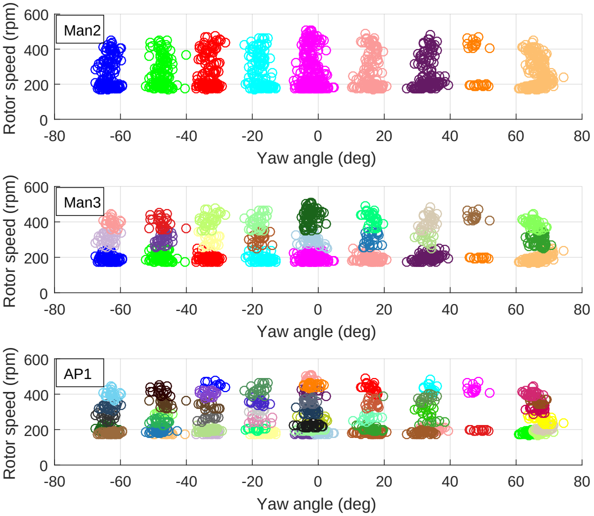


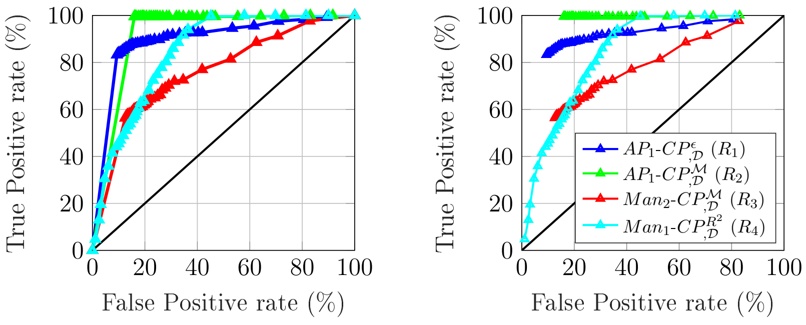
Figure 7.2: Cluster geometries for clustering cases Man2, Man3 and AP1 on the rotor speed - yaw angle space based on 15-minute mean values of the training data EOCs.

7.2.2 AdaBoost with Two Realizations

The implementation of AdaBoost for the combination of two framework realizations was carried out for different realization pairs. Four realizations were selected and investigated pairwise in order to assess the performance of AdaBoost:

- Realization 1 (R_1): CP^{ϵ} with clustering using AP1 and discrete distribution in HT,
- Realization 2 (R_2): CP^{M} with clustering using AP1 and discrete distribution in HT,
- Realization 3 (R_3): CP^{M} with clustering using Man2 and discrete distribution in HT,
- Realization 4 (R_4): CP^{R^2} with clustering using Man1 and discrete distribution in HT.

Before proceeding with boosting, the error rates of the aforementioned realizations were calculated in order to specify for which ROC points the weak learning assumption held. 35 ROC points were available in total. The weak learning condition was satisfied by 34 ROC points for R_1 and R_2, 35 ROC points for R_3 and 34 ROC points for R_4. Figure 7.3(a) shows the ROC curves of the selected realizations, while Figure 7.3(b) shows the ROC points of the selected ROCs with error rate smaller than 0.5. Those



(a)

(b)

Figure 7.3: (a) ROC curves of the realizations selected for the analysis of the simulated data using the SHM framework (R_1 to R_4). (b) Points of the ROC curves of R_1 to R_4 that satisfy the weak learning assumption.

realizations were combined pairwise, forming six sets: R_1-R_2, R_1-R_3, R_1-R_4, R_2-R_3, R_2-R_4, R_3-R_4. The cases which are to be investigated in AdaBoost are formed by combining the ROC points of two realizations, as long as they satisfy the weak learning condition. In the training phase, 300 CP values were used in 2000 rounds to train the AdaBoost classifier. The training data consisted of only tp and tn detections of the two realizations. Those were sampled by the CP sets with systematic sampling as described in section 4.4.

These four realizations were selected on the one hand, because they cover all examined CPs (CP^{ϵ}, CP^{M} and CP^{R^2}) and all examined clustering cases (Man1, Man2 and AP1) and, on the other hand, because they yield ROCs of different form. For instance, R_1 has high TP rates for all confidence intervals and the FP rates increase with decreasing confidence interval $(1-a)$, whereas R_4 has low TP rates and low FP rates for lower confidence intervals and with increasing confidence interval.[2]

7.2.2.1 Results

The results of employing AdaBoost with two realizations are presented in Table 7.3 for the six considered realization pairs. The table presents the number of investigated sets for each of the realization pairs. For instance, the cases of pair R_1-R_2 occurred by combining 34 ROC points of R_1 with 34 ROC points of R_2 and resulted in 1156 combinations to be examined. The detections of AdaBoost were assessed based on the boosting metrics introduced in Section 4.6, which are obtained (i) by the simul-

[2] The confidence interval $(1-a)$ decreases as one moves from the left to the right side of the ROC plot.

taneous comparison of the framework TP and FP rates to the TP and FP rates of AdaBoost and (ii) by the comparison of the SHM framework error rate to the error rate of AdaBoost. In addition, Table 7.3 presents the number of cases which resulted in boosting based on those two metrics.

The obtained number of sets resulting in boosting differs depending on the employed criterion, i.e., on whether the error rate or the combination of the TP and FP rates was used for the assessment of the detection performance. These discrepancies can be attributed to the definitions of these three rates, since the error rate essentially includes the TP and FP rates expressed in a different manner. When observing the error rate differences, only a small number of sets result in boosting. More specifically, only 2 out of 1190 sets of R_1-R_3 and 229 out of 1190 sets of R_2-R_3 result in boosting. The optimal case of obtaining simultaneous improvement of both the TP rates and the FP rates ($\Delta TP_i = TP_{AdB} - TP_i > 0$ and $\Delta FP_i = FP_{AdB} - FP_i < 0$, for $i = 1, 2$) is achieved for 346 cases of R_3-R_4. The suboptimal case, which allows for small deteriorations of the TP or the FP rate ($\Delta TP_i = TP_{AdB} - TP_i > -10$ and $\Delta FP_i = FP_{AdB} - FP_i < 10$, for $i = 1, 2$), is achieved for 35 sets of R_2-R_4 and 330 sets of R_3-R_4.

Table 7.3: Number of sets satisfying the weak learning assumption and number of sets resulting in boosting according to the error rate metric $\Delta\epsilon$ and according to the combination of TP and FP rate metrics for all realization pairs - synthetic data.

Realization set	R_1-R_2	R_1-R_3	R_1-R_4	R_2-R_3	R_2-R_4	R_3-R_4
Number of sets after the weak learning condition	1156	1190	1156	1190	1156	1190
Results based on ΔTP and ΔFP metrics						
$\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2$	0	0	0	0	0	346
$\Delta TP_i > -10$ and $\Delta FP_i < 10$ for $i = 1, 2$	0	0	0	0	35	330
Results based on $\Delta\epsilon$ metric						
$\Delta\epsilon_i < 0$ for $i = 1, 2$	51	2	5	229	354	14

7.2.2.2 Quantification of Performance Improvement

The detailed results of employing AdaBoost for pair R_3-R_4 are presented in this section in order to provide a quantification of the improvement which can be achieved. Table 7.4 presents the mean values and the standard deviations of the differences between the TP and FP rates of the AdaBoost classifier and the corresponding rates of realization pair R_3-R_4. These statistical values refer to the sets which result in improved detection performance according to the different boosting metrics. In the optimal case ($\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2$), the TP rate of AdaBoost is on average 8.14 percentage points higher than the TP rate of realization R_3, and 15.92 percentage points higher than the TP rate of R_4. On the other hand, the FP rate of AdaBoost is on average 11.20 percentage points lower than the FP rate of R_3 and 6.46 percentage points lower than the FP rate of R_4. For the suboptimal case ($\Delta TP_i > -10$ and $\Delta FP_i < 10$ for $i = 1, 2$), AdaBoost outperforms both framework classifiers in terms of the TP and FP rates, but the standard deviations are higher than in the optimal case mentioned above.

Furthermore, the TP and FP rate differences are presented for the results obtained based on the error rate differences. For these sets, the FP rate of AdaBoost is on average 57.04 percentage points lower than the FP rate of R_3 and 14.84 percentage points lower than the FP rate of R_4. As for the TP rates, AdaBoost TP rate is on average 36.46 percentage points higher than the TP rate of R_4, but only 2.06 percentage points lower than the TP rate of R_3. These results reflect the fact that the definition of the error rate does not distinguish between fp and fn detections. Consequently, the error rate may reflect the compensation between a low fp rate and a high fn rate or between a high fp rate and a low fn rate.

Table 7.4: Mean values and standard deviations of the differences between AdaBoost TP rate and TP rates of SHM framework classifiers (third column), and between AdaBoost FP rate and FP rates of SHM framework classifiers (fourth column) for realization pair R_3-R_4 of the simulated data. Mean values and standard deviations of the changes are given in percentage points for the sets resulting in boosting according to (i) the TP and FP rate changes and (ii) the error rate changes.

Boosting metrics for $i = 1, 2$	Number of sets resulting in boosting	$(\mu_3 \pm \sigma_3)$, $(\mu_4 \pm \sigma_4)$ of TP % changes with respect to R_3 and R_4	$(\mu_3 \pm \sigma_3)$, $(\mu_4 \pm \sigma_4)$ of FP % changes with respect to R_3 and R_4
$\Delta TP_i > 0$ and $\Delta FP_i < 0$	346/1190	(8.14±3.66), (15.92±7.46)	(-11.20±5.82), (-6.46±3.32)
$\Delta TP_i > -10$ and $\Delta FP_i < 10$	330/1190	(5.78±7.16), (25.39±21.50)	(-17.90±13.40), (-2.91±10.20)
$\Delta \epsilon_i < 0$	14/1190	(-2.06±7.88), (36.46±46.50)	(-57.04±10.68), (-14.84±43.94)

7.2.2.3 Identification of Sets Resulting in Boosting

Table 7.3 presents the number of sets which resulted in boosting for all six considered realization pairs and for both boosting metrics. When considering that improvement is achieved when $\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2$, boosting is obtained for only 346 sets of R_3-R_4. When considering that $\Delta TP_i > -10$ and $\Delta FP_i < 10$ for $i = 1, 2$ constitutes improvement, boosting is obtained for 35 sets of R_2-R_3 and 330 sets of R_3-R_4. When considering that improvement is achieved when $\Delta\epsilon_i < 0$ for $i = 1, 2$, boosting is obtained for 51, 2, 5, 229, 354 and 14 sets of the six realization pairs. Thus, it may be observed that not all sets yield a new decision rule which outperforms the framework classifiers. Therefore, it has to be investigated, which set attributes render them eligible for AdaBoost.

The attributes which potentially affect the final boosting performance when analyzing a set are:

- the number of training datasets,
- the coordinates of the ROC points which are combined, i.e., the TP rate, the FP rate and, indirectly, the confidence interval of the individual framework realizations forming a set to be analyzed in AdaBoost, and
- the overlaps of the CP values of the two realizations on the CP space.

The negative effect of using few training datasets can be ignored in the case of this specific database, since a sufficient number of training datasets were used for training AdaBoost. Moreover, all sets of the six realizations contained exactly the same number of training datasets. In order to investigate the effect of the coordinates of the combined ROC points, the sets which resulted in boosting can be plotted upon the matrix containing the ROC point combinations. The selected confidence interval $(1 - a)$ affects the decision boundaries and results in some TP and FP rates. In general, as the confidence interval decreases, the range of CPs regarded as "healthy" decreases and higher FP rates occur. Hence, ROC points located in the left part of the ROC plot are associated with high confidence intervals, while the ROC points in the right part of the plot are associated with low confidence intervals. The form of the ROC curve, however, depends on the distribution and the nature of the calculated CP.

Figure 7.4 depicts the locations of the sets which result in boosting according to the error rate metric ($\Delta\epsilon_i < 0$ for $i = 1, 2$) for each of the realization pairs. White elements represent the sets resulting in boosting, while black elements depict the remaining sets. The locations of the sets resulting in boosting vary for each of the realization pairs. Consequently, it can be stated that the boosting performance of an analyzed set is independent of the coordinates of the ROC points.

Similarly, the boosting results according to the TP and FP rates are plotted upon the matrix containing all the possible sets of each realization (see Figure 7.5). The sets are categorized into: (i) sets which do not result in boosting (indicated by blue elements or number 0 in the colorbar), (ii) sets resulting in the optimal case

$\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2$ (cyan elements or number 1 in the colorbar), (iii) sets resulting in the suboptimal case $\Delta TP_i > -10$ and $\Delta FP_i < 10$ for $i = 1, 2$ (yellow elements or number 2 in the colorbar) and (iv) sets resulting in improvement of some rates but deterioration of other rates (dark red elements or number 3 in the colorbar). The locations of these four cases on the ROC combination matrix differ for the six realization pairs. Hence, the conclusion can be drawn that the ROC point coordinates of the combined realizations do not correlate with the boosting performance or with the boosting cases, regardless of the boosting metric considered.

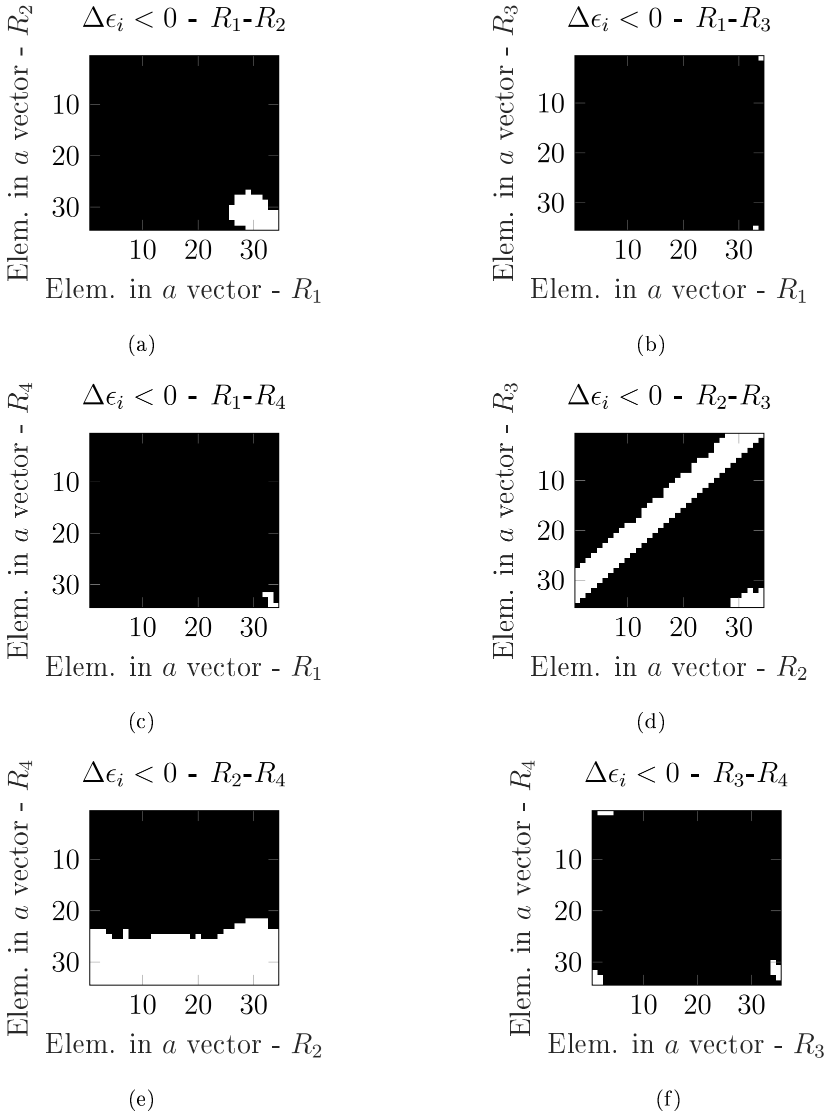
The Bhattacharyya coefficient (BC) was calculated for all the sets of CP values analyzed using AdaBoost in order to quantify the overlap between the two distributions. The values of each CP were divided into values corresponding to tp, tn, fp and fn detections of the framework. Subsequently, the overlaps between those sets of the two CPs were calculated. An example is given in Table 7.5, where the BC values are presented for the two sets shown in the boxplots of Figures 4.5 and 4.6. Figure 4.5 corresponds to a set yielding the optimal case for R_3-R_4, i.e., to $\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2$. Pair R_3-R_4 involves the combination of CP^M values with CP^{R^2} values. Figure 4.6, on the other hand, shows the boxplots for a set of R_1-R_2 which yields boosting of some rates but causes deterioration of other rates. The CP values combined in this figure were derived from CP^ϵ and CP^M. The BC values of Figure 4.5 are given in the left part of the table, while the BC values of Figure 4.6 are given in the right part of table. It may be observed that the overlaps in the left part of the table are significantly higher than those in the right part. Not-a-number BC values in the table indicate the absence of one of the two distributions which are used for the calculation of the overlaps.

The difference in overlaps between the two cases can be attributed to the nature of the CPs considered. Unnormalized CP^M values are close to zero in the healthy state and increase in the damaged state. On the contrary, CP^{R^2} values are equal to one in the healthy state and decrease in the case of structural changes. As for the CPs involved in R_1-R_2, CP^ϵ values are close to zero for the healthy datasets and increase significantly for the damaged datasets. On the other hand, CP^M values are zero for the healthy datasets but in the case of structural changes exhibit increases which are smaller than those of CP^ϵ values. It can be concluded that the nature of the combined CPs as well as their sensitivity to damage affect their relative positioning on the normalized CP space and, subsequently, the boosting performance. After normalization, the considered CPs are observed in the same range but these trends between datasets from the healthy structure and datasets from the damaged structure are preserved.

The values of the Bhattacharyya coefficient were calculated for the individual sets of tp, tn, fp and fn detections. The results for the individual sets are presented in Appendix C for all the realization pairs (see Figures C.2 to C.7). In order to obtain one single metric for the overlap between the CP values of two realizations, the BC values of the individual sets were summed up, resulting in a sum of overlaps. This sum is plotted upon the matrix containing the combinations of ROC points in order to investigate the correlation between the boosting performance and the overlap of the two CPs (see Figure 7.6). The results can be summarized as follows:

- Realization pairs R_1-R_2 and R_1-R_3 have lower BC sums than the other pairs, with BC values lying below 5. These two realization pairs contain very few pairs which result in boosting according to the error rate metric. With respect to the TP rate and FP rate metrics, these two realization pairs result in the improvement of some rates and deterioration of others (see Table 7.3).
- The BC sums of realization pair R_1-R_4 are on average slightly higher than 5. This pair contains very few sets which exhibit boosting according to the error rate metric, and none according to the TP and FP rate metrics.
- The BC sums for pair R_2-R_3, on the other hand, are slightly lower than 10. These values are significantly higher than the aforementioned ones and reflect the fact that this pair contains some sets which exhibit improvement of their detection performance based on the error rate metric. As for the TP rate and FP rate metrics, this pair results in improvement of some metrics and deterioration of others.
- The BC sums of realization pair R_2-R_4 lie between 7.5 and 10. This pair results in boosting for 354 sets according to the error rate metric, while 35 sets result in the suboptimal boosting case, in which $\Delta TP_i > -10$ and $\Delta FP_i < 10$ for $i = 1, 2$.
- Finally, the BC sums of realization pair R_3-R_4 lie between 7.5 and 12, while the cases located on the same matrix elements as those yielding the optimal boosting case (see Figure 7.5(f)) have BC sums greater than 10. This pair yields few sets which result in boosting according to the error rate metric. Additionally, more than 300 sets result in the optimal boosting case and more than 300 result in the suboptimal boosting case.

The interpretation of the maps containing the sum of all BCs indicates that there is a correlation between the overlaps of the CP values being combined within a set and the boosting performance. More specifically, the higher the values of the BC sum, the higher the possibility that boosting is achieved. These conclusions have been drawn by observing the sum of all possible overlaps within a set, i.e., by observing a sum that encapsulates all overlaps in one value. In general, it is challenging to define a BC sum threshold which indicates which sets are expected to result in boosting. However, as can be seen in the results summarized above, BC sum values greater than 8 are very likely to yield good boosting performance. It has to be noted that the sum of BC values contains both advantageous and disadvantageous overlaps (see Table 4.1). As a result, the overlapping of the different CP values is a complex phenomenon, difficult to quantify and to enclose in one metric.



(a) (b)

(c) (d)

(e) (f)

Figure 7.4: Combinations resulting in boosting according to the error rate metric ($\Delta\epsilon_i < 0$ for $i = 1, 2$) plotted upon the matrix containing the sets of ROC points of two realizations - synthetic database. $1 - a$ is the confidence interval and a is the significance level. White elements indicate the sets resulting in boosting.

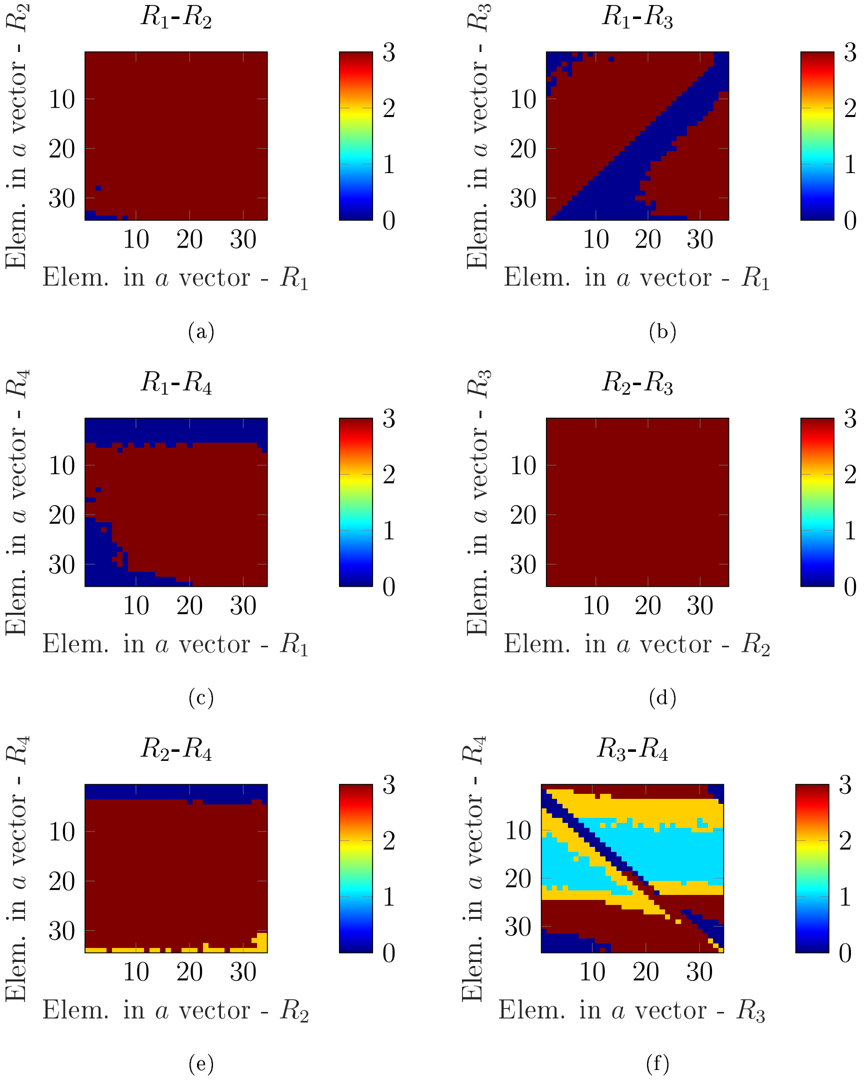
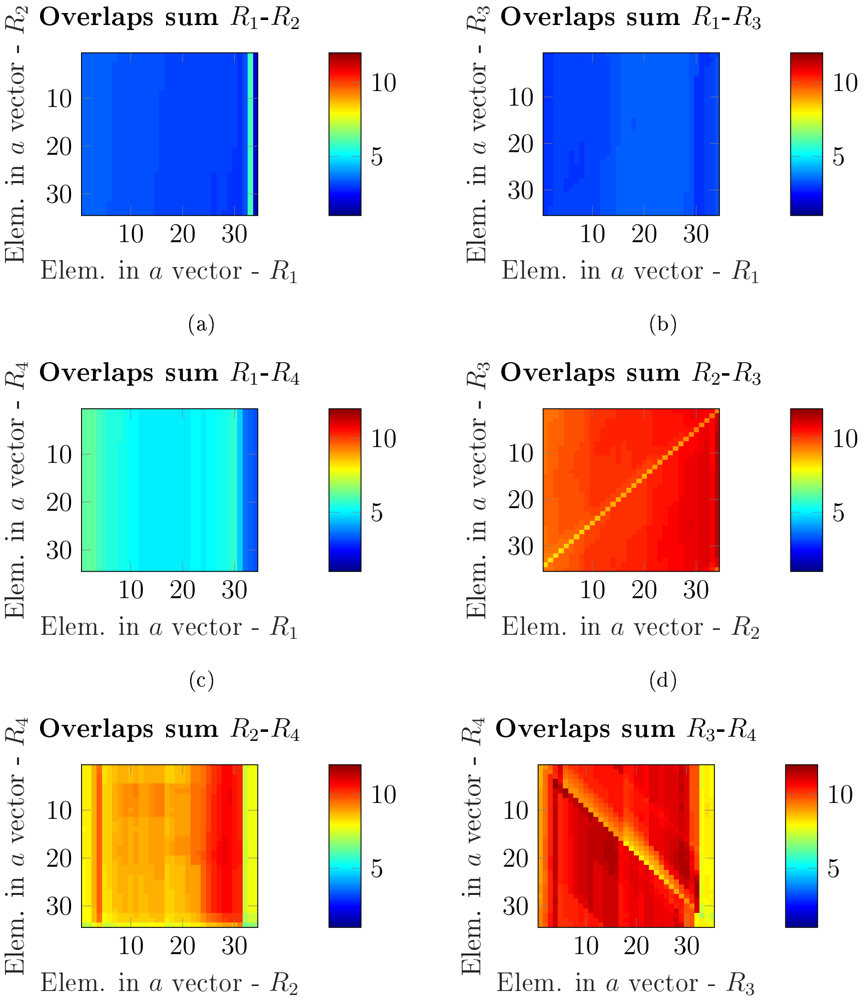


Figure 7.5: Boosting results according to TP and FP rate changes, plotted upon the matrix containing the sets of the ROC points of two realizations. $1-a$ is the confidence interval and a is the significance level. Blue elements indicate no boosting (0 in colorbar). Cyan elements indicate the optimal case $\Delta TP_i > 0$ and $\Delta FP_i < 0$ (1 in colorbar). Yellow elements indicate the suboptimal case $\Delta TP_i > -10$ and $\Delta FP_i < 10$ (2 in colorbar). Dark red elements indicate all other sets which yield improvement of the one rate but deterioration of the other rate (3 in colorbar). Results presented for the synthetic database.



(a) (b)

(c) (d)

(e) (f)

Figure 7.6: Sum of Bhattacharyya coefficients of all overlaps (i.e., overlaps between the tp, tn, fp and fn detections of the two CPs), plotted upon the matrix containing the sets of the ROC points of two realizations. $1 - a$ is the confidence interval and a is the significance level. Results presented for the synthetic database.

Table 7.5: Overlaps of HT results of CP_1 and CP_2 according to Bhattacharyya coefficient (BC). Example 1 has higher overlaps and is expected to result in good boosting performance (left part). Example 2 has lower overlaps and is expected to have mediocre boosting performance (right part). Elements with positive overlaps marked with gray.

	BC values in Figure 4.5				BC values in Figure 4.6			
	tp_2	tn_2	fp_2	fn_2	tp_2	tn_2	fp_2	fn_2
tp_1	0.9370	0.3598	0.6805	0.8685	0.5278	0.3126	0.4439	0.7071
tn_1	0.3727	0.9047	0.5409	0.8530	NaN	0.0930	0.2176	NaN
fp_1	0.8482	0.7190	0.9161	0.7424	NaN	0.0930	0.2176	NaN
fn_1	0.4461	0.6836	0.4289	0.9642	NaN	0.0930	0.2176	NaN

7.2.2.4 Improvement of CP Overlapping

The results of the application of AdaBoost for different realizations of the synthetic database were presented in the previous section. It was shown that boosting performance depends on the relative positions of the CP values on the CP space and on the overlaps between the CP values of the two realizations which correspond to specific hypothesis testing results (tp, tn, fp and fp detections). The optimum case, in which the AdaBoost classifier has a higher TP rate and a lower FP rate than both framework classifiers, was only achieved for one of the six realization pairs. The suboptimal case, which allows for up to 10% deterioration of some rates, occurred for only two pairs. In the remaining cases, boosting in terms of one rate was accompanied by deterioration with respect to the other rate.

In this section, it is investigated whether omitting CP values corresponding to specific EOC clusters helps overcome the absence of optimal overlaps and delivers better boosting performance than when including the entire CP sets. This investigation was carried out by successively omitting the CP values of each cluster from the training set and test set, altering thereby the distribution of the normalized CP values on the CP space. In other words, the clusters were successively switched off. For instance, R_1 and R_2 had 53 clusters each and formed pair R_1-R_2, which was examined for 1156 sets of ROC points. First, the clusters of R_1 were successively switched off one at a time. The training set was then selected with systematic sampling and normalized for AdaBoost. The same procedure was followed for all sets of ROC points. Similarly, all the 53 clusters of R_2 were successively omitted from the analysis before AdaBoost was employed.

Table 7.6 summarizes the percentage of sets resulting in the optimal and suboptimal boosting cases when entire CP sets are considered (see the second and third columns), as well as when switching off clusters (see the last four columns). The number of clusters which are switched off per realization is given in the fourth column, while the last four columns show the percentage of sets resulting in boosting when the clusters of the one realization (R_i) or of the other (R_j) are switched off. For instance, for pair R_1-R_3, 14.66% of the 1190 sets result in the optimal boosting case when switching off the clusters of R_1, and 26.69% when switching off the clusters of R_3. The suboptimal case occurs for 21.36% of the sets when switching off the clusters of R_1 and for 0.87% of the sets when switching off the clusters of R_3. The optimal boosting case of increasing the TP rate and simultaneously decreasing the FP rate is attained for all realization pairs except for pair R_2-R_3, when switching off the clusters of R_3 (see the third and fourth columns of Table 7.6). By omitting CP values of specific clusters, the distributions and overlaps of the CP values are altered. As a consequence, boosting performance is in some cases better than when applying AdaBoost to the entire CP sets.

The extent to which boosting performance is improved is presented in parallel coordinate plots in Appendix C. Parallel coordinate plots are a useful tool for the visualization of multivariate data and for the comparison of many variables. Each variable constitutes a coordinate and has its own axis. All axes are plotted next to each other, while the values corresponding to the same set are connected by lines. The parallel coordinate plots in Appendix C present the following for each realization pair R_i-R_j: (i) the cluster number of a realization which, when switched off, results in boosting (first coordinate), (ii) the corresponding significance level values for R_i (second coordinate), (iii) the corresponding significance level values for R_j (third coordinate), (iv) the changes in the TP rate of R_i achieved by boosting (fourth coordinate), (v) the changes in the TP rate of R_j achieved by boosting (fifth coordinate), (vi) the changes in the FP rate of R_i achieved by boosting (sixth coordinate) and (vii) the changes in the FP rate of R_j achieved by boosting (seventh coordinate). In Appendix C, we also present the identity and the location on the EOC space of the clusters which have to be excluded from the analysis in order to obtain improvement in boosting performance.

Table 7.6: Percentage of sets resulting in the optimal and suboptimal boosting cases when all CP values are considered (second and third column) and when cluster CP values of the synthetic data are successively switched off (last four columns). Results presented for all realization pairs and for switching off clusters of R_i and R_j. Number of clusters in each realization is shown in the fourth column. The optimal case ($\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2$) occurs for all examined cases except for switching off clusters of R_2 in R_1-R_2.

	Optimal case	Suboptimal case		Optimal case when switching off clusters of		Suboptimal case when switching off clusters of	
R_i-R_j	Number of sets resulting in optimal case	Number of sets resulting in suboptimal case	Number of clusters in (R_i, R_j)	R_i	R_j	R_i	R_j
R_1-R_2	0/1156 (0%)	0/1156 (0%)	(53, 53)	5.49%,	5.49%	-	-
R_1-R_3	0/1190 (0%)	0/1190 (0%)	(53, 9)	14.66%	26.69%	21.36%	0.87%
R_1-R_4	0/1156 (0%)	0/1156 (0%)	(53, 1)	20.99%	-	10.23%	-
R_2-R_3	0/1190 (0%)	0/1190 (0%)	(53, 9)	5.85%	-	4.95%	0.26%
R_2-R_4	0/1156 (0%)	35/1156 (3.02%)	(53, 1)	10.83%	-	21.27%	-
R_3-R_4	346/1190 (29.07%)	330/1190 (27.73%)	(9, 1)	32.22%	-	21.81%	-

7.3 Validation on the Data of a 3 kW Wind Turbine

The main objective of this chapter is to implement AdaBoost on damage features obtained from real structural data. For this purpose, the tower response data of an operating 3 kW wind turbine were analyzed using the SHM framework for different settings and condition parameters. In the previous section, simulated data from a structure which approximated the 3 kW wind turbine were analyzed to validate the implementation of AdaBoost and to extract useful conclusions about its application in damage features obtained from real data.

The analysis presented in the previous section was repeated for the experimental setup of the small wind turbine, which was described in Section 7.1. First, the response data of the tower were analyzed using the three-tier SHM framework for several settings. Four realziations were selected to be examined pairwise in AdaBoost, resulting in six realization pairs. This section outlines the results of employing AdaBoost for two realizations, the quantification of the achieved improvement of detection performance as well as the calculation of the Bhattacharyya coefficient, which serves as an indicator of the CP overlaps. Moreover, the section includes the implementation of k-fold cross-validation with three folds, which provides an estimate of the algorithm's generalization error, as well as the deployment ofAdaBoost for combining three framework realizations.

7.3.1 Analysis using the SHM Framework

The operational data of the LANL 3 kW wind turbine were analyzed with the use of the SHM framework. The collected data were downsampled to 100 Hz and analyzed for various clustering cases and CPs. The database contained 367 datasets, with 325 datasets from the healthy state and 42 datasets from the damaged state. 204 out of the 325 healthy datasets were used in training, while the remaining 163 healthy and damaged datasets were used in testing.

In addition to the analysis without clustering, which is referred to as Man1, four different clustering cases were examined. The first involved manual clustering based on wind speed and temperature 15-minute averages, and resulted in 3 clusters (clustering case Man2). In the second clustering case, datasets were grouped using the affinity propagation (AP) algorithm, again based on wind speed and temperature 15-minute averages (clustering case AP1), resulting in 5 clusters. In the remaining clustering cases, datasets were clustered with AP based on wind speed and wind direction 15-minute averages (clustering cases AP2 and AP3) and resulted in 3 and 6 clusters, respectively. Figure 7.7 depicts the training and test data, as well as the healthy and damaged data, on the wind speed - temperature space and wind speed-wind direction space. The clustered EOC data is shown in Figure 7.8 for Man2, AP1, AP2 and AP3.

In the second tier, the residual CPs used were the same as those presented in the analysis of the simulated data. Those were CP^M and CP^{R^2}, which derive from a VAR model, as well as CP^ϵ, which derives from the SSI approach. In the third tier, the decision boundaries were defined based on the assumption that the CPs follow the Gaussian distribution. Those combinations of settings along the three tiers resulted in different realizations of the SHM framework. Each realization was tested for 48 confidence intervals. Table 7.7 summarizes the information of the database and the analysis settings.

Table 7.7: Settings for the analysis of the LANL wind turbine database using the SHM framework.

Number of channels	12
Sampling frequency	100 Hz
Number of datasets	367
Training datasets	204
Testing datasets	163
Classification names	Man1, Man2, AP1, AP2, AP3
Number of clusters	1, 3, 5, 3, 6
Settings for condition parameters	
CP^M/CP^{R^2}	AR-model order=30
CP^ϵ	time shifts=2 significant columns=20
Analyzed distribution types: Gaussian a-values for hypothesis testing: a=[0.1:0.1:0.4, 0.5:0.5:10, 11:1:20, 25:5:45, 55:5:75, 80:1:90, 90.5:0.5:95, 95.6:0.1:99.9]%	

Table 7.8: EOCs accounted for clustering, settings and resulting number of clusters for the five clustering cases used in the analysis of the database within the SHM framework.

Clustering case	EOCs for clustering	Settings	Number of clusters
Man1	-	-	1
Man2	wind speed/ temperature	-	3
AP1	wind speed/ temperature	Pref.=-4	5
AP2	wind speed/ wind direction	Pref.=-12	3
AP3	wind speed/ wind direction	Pref.=-11	6

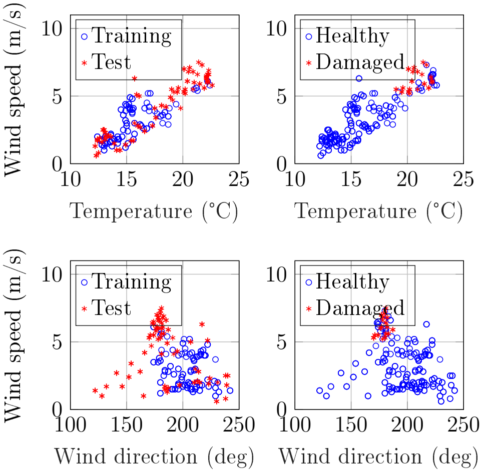


Figure 7.7: Training and test (left plots) and healthy and damaged datasets (right plots) on the wind speed - temperature space (upper plots) and wind speed - wind direction space (lower plots). The data points depict 15-minute averages of the entire environmental data of the LANL wind turbine.

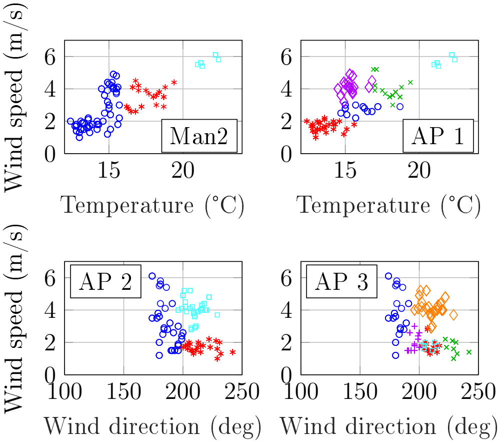


Figure 7.8: Cluster geometries for clustering cases Man2, AP1, AP2 and AP3 on the wind speed - temperature space (upper plots) and the wind speed - wind direction space (lower plots). The data points depict 15-minute averages of enviromnental data of the LANL wind turbine which were used in training.

7.3.2 AdaBoost with Two Realizations

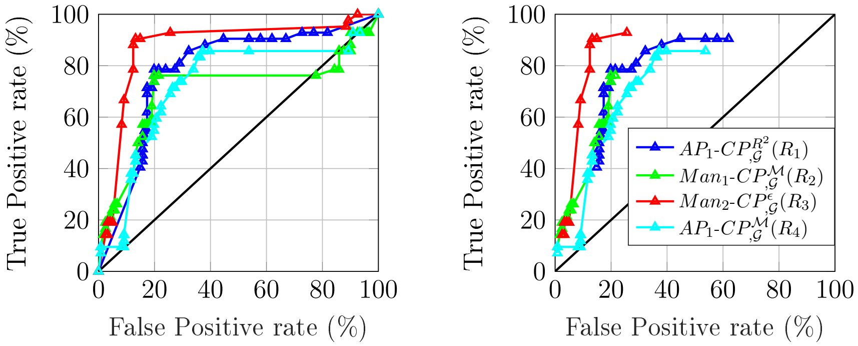
In the previous section, the operational data were analyzed using the SHM framework and the results were presented in ROCs. The following four realizations were selected for the boosting stage:

- Realization 1 (R_1): CP^{R^2} with clustering using AP1 and Gaussian distribution in HT,
- Realization 2 (R_2): CP^{M} with clustering using Man1 and Gaussian distribution in HT,
- Realization 3 (R_3): CP^{ϵ} with clustering using Man2 and Gaussian distribution in HT and
- Realization 4 (R_4): CP^{M} with clustering using AP1 and Gaussian distribution in HT.

Before proceeding with boosting, the error rates of all the ROC curve points were calculated in order to examine whether the related classifier satisfied the weak learning assumption. The ROCs of the selected realizations are shown in Figure 7.9(a), while the ROC points satisfying the weak learning assumption are presented in Figure 7.9(b). Points with error rate smaller than 0.5 lie under the $x = y$ line and in the area with FP rates higher than 65%. The weak learning assumption is satisfied for 23 out of 48 ROC points for R_1, 21 for R_2, 22 for R_3 and 24 for R_4.

All the ROC point combinations are presented in Figure 7.11 as elements of the matrix which contains the a values combinations of realizations R_1 and R_2. Black elements indicate the combinations satisfying the weak learning assumption. Gray elements indicate the combinations which satisfy the weak learning assumption but deliver detections identical to previous combinations and are thus omitted. Combinations which do not satisfy the weak learning assumption are represented by white elements.

As opposed to the synthetic database, the operational database contains relatively few datasets since there are only 42 damaged datasets. Therefore, there can be at most 42 *tp* detections. The training data were selected using systematic sampling, so that the number of *tp* detections would be approximately equal to the number of *tn* detections for both realizations (see concept in Figure 4.7). The final number of training data of R_1-R_2 is presented exemplarily in Figure 7.10 for all possible combinations of the ROC points of R_1 and R_2. R_1 has a dominant effect on the determination of the number of training datasets, as manifested by the vertical stripes in the matrix. This implies that the lowest number of *tp* or *tn* detections was mainly defined by R_1. The maximum number of training datasets is 134 and occurs for higher a values of both R_1 and R_2, with the effect of R_1 being again prevalent.



(a) (b)

Figure 7.9: (a) ROC curves of the selected realizations (R_1 to R_4) for the analysis of the operational data with the SHM framework. (b) Points of the ROC curves of R_1 to R_4 that satisfy the weak learning assumption.

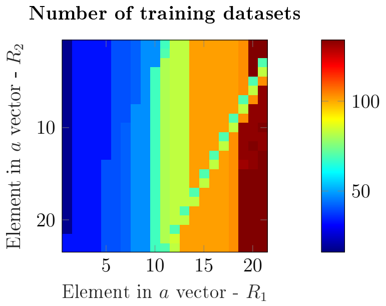


Figure 7.10: Number of AdaBoost training datasets for all possible combinations of a values between R_1 and R_2 of the LANL wind turbine data. a the significance level and $1 - a$ is the confidence interval. Vertical stripes manifest the dominance of R_1 regarding the determination of the number of training datasets.

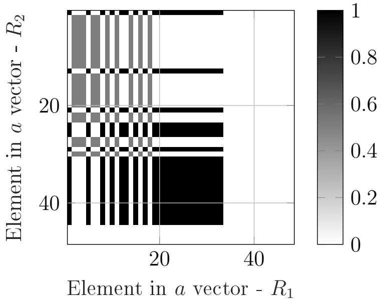


Figure 7.11: All combinations of confidence intervals $(1 - a)$ for pair $R_1 - R_2$ - LANL wind turbine data. Black elements represent the combinations which satisfy the weak learning assumption and which were further used in boosting. Gray elements correspond to the combinations which satisfy the weak learning assumption but deliver solutions identical to other combinations. White elements represent the combinations that do not satisfy the weak learning assumption.

7.3.2.1 Results

Table 7.9 summarizes the results for all the realization pairs considered. The number of combinations which satisfy the weak learning assumption is presented along with the number of sets which result in boosting according to the error rate metric and the TP and FP rates (optimal case and suboptimal case). The optimal boosting case ($\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2$) is obtained for four out of the six realization pairs and, more specifically, for 12 sets of R_1-R_2, 41 sets of R_1-R_4, 48 sets of R_2-R_4 and 11 sets of R_3-R_4. The suboptimal case occurs for all pairs and, more specifically, for 135 sets of R_1-R_2, 14 sets of R_1-R_3, 205 sets of R_1-R_4, 34 sets of R_2-R_3, 79 sets of R_2-R_4 and 36 sets of R_3-R_4. As for the error rate metric, no sets result in boosting.

Table 7.9: Number of sets satisfying the weak learning assumption and number of boosting cases according to the error rate metric $\Delta\epsilon$ and to the combination of the TP and FP rate metrics for all realization pairs - operational data of the LANL wind turbine.

Realization set	R_1-R_2	R_1-R_3	R_1-R_4	R_2-R_3	R_2-R_4	R_3-R_4
Number of sets after the weak learning condition	483	462	504	506	552	528
Results based on ΔTP and ΔFP metrics						
$\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2$	12	0	41	0	48	11
$\Delta TP_i > -10$ and $\Delta FP_i < 10$ for $i = 1, 2$	135	14	205	34	79	36
Results based on $\Delta\epsilon$ metric						
$\Delta\epsilon_i < 0$ for $i = 1, 2$	0	0	0	0	0	0

7.3.2.2 Quantification of Performance Improvement

The results of realization pair R_2-R_4 are presented in detail in this section in order to gain insight into the extent to which AdaBoost improves the detection performance of the SHM framework. Table 7.10 provides the number of sets resulting in the optimal and suboptimal boosting cases, as well as the mean values and standard deviations of the TP and FP rate changes with respect to the rates of R_2 and R_4. The optimal boosting case is obtained for 48 out of the 552 sets which satisfy the weak learning assumption. The TP rate of AdaBoost is on average 14.73 percentage points higher than the TP rate of R_2, with a standard deviation of 12.93, and 24.38 percentage points higher than the TP rate of R_4, with a standard deviation of 14.27. The FP rates, on the other hand, decrease on average by 4.42 percentage points compared to R_2 and by 5.21 percentage points compared to R_4. The suboptimal boosting case is obtained for 70 out of 551 sets. In this case, the TP rates have an average increase of 12.45 and 29.63 percentage points compared to R_2 and R_4, respectively. However, the FP rates exhibit an average increase of 5.45 percentage points compared to R_2, and an average increase of 2.52 percentage points compared to R_4.

The differences of the TP and FP rates are presented in Figure 7.12 for the optimal and suboptimal cases as well as for boosting cases 3 and 4, which involve the improvement of some rates but the deterioration of other rates. More specifically, boosting case 3 involves improvement of the TP rates and either improvement or deterioration of the FP rates ($\Delta TP_i > 0$ and $\Delta FP_i > 0$ or $\Delta FP_i > 0$), whereas boosting case 4 involves improvement of the FP rates and either improvement or deterioration of the TP rates ($\Delta TP_i < 0$ or $\Delta TP_i > 0$ and $\Delta FP_i < 0$). The left subplot shows the rate differences with respect to R_2, while the right subplot shows the differences with respect to R_4. The optimal case in both plots is indicated by the blue marks. The results of the optimal case show that the AdaBoost classifier has a lower FP rate than both R_2 and R_4, with the differences reaching up to 20 percentage points. As for the TP rates, the optimal case delivers AdaBoost TP rates, which are up to 40 percentage points higher than the TP rates of R_2 (upper plot) and up to 45 percentage points higher than the TP rates of R_4 (lower plot).

Table 7.10: Mean values and standard deviations of the differences between AdaBoost TP rate and TP rates of SHM framework classifiers (third column), and between AdaBoost FP rate and FP rates of SHM framework classifiers (fourth column) for realizations R_2 and R_4 of the operational data. Mean values and standard deviations are given in percentage points for the four boosting cases according to TP and FP rate changes.

Boosting metrics for $i = 1, 2$	Number of sets resulting in boosting	($\mu_2\pm\sigma_2$) for TP % changes with respect to R_2 and R_4	($\mu_4\pm\sigma_4$) for FP % changes with respect to R_2 and R_4
$\Delta TP_i > 0$ and $\Delta FP_i < 0$	48/552	(14.73±12.93), (24.38±14.27)	(-4.42±2.80), (-5.21±1.66)
$\Delta TP_i > -10$ and $\Delta FP_i < 10$	79/552	(12.45±7.16), (29.63±18.72)	(5.45±12.87), (2.52±11.53)

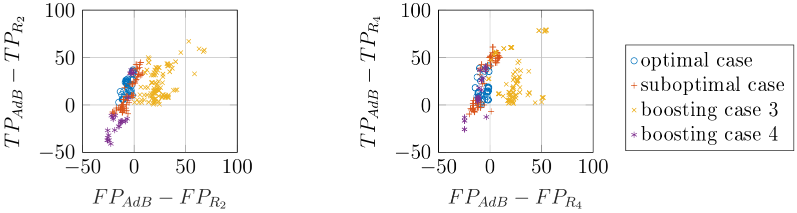


Figure 7.12: TP rate and FP rate differences between AdaBoost classifier and R_2 classifier (left plot) and between AdaBoost classifier and R_4 classifier (right plot) for pair R_2-R_4 of the LANL wind turbine data. Optimal case: $\Delta TP_i > 0$ and $\Delta FP_i < 0$. Suboptimal case: $\Delta TP_i > -10$ and $\Delta FP_i < 10$. Boosting case 3: $\Delta TP_i > 0$ and $\Delta FP_i > 0$ or $\Delta FP_i > 0$. Boosting case 4: $\Delta TP_i < 0$ or $\Delta TP_i > 0$ and $\Delta FP_i < 0$.

7.3.2.3 Identification of Sets Resulting in Boosting

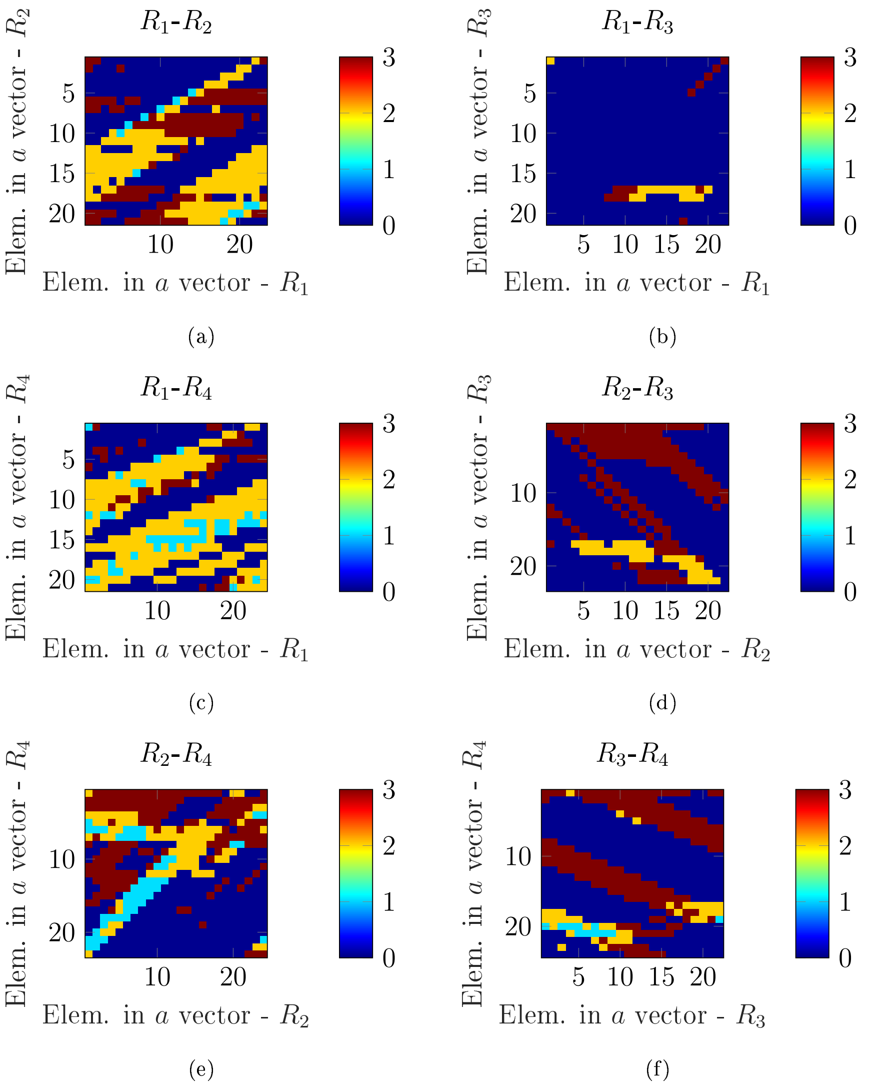
In Figure 7.13, the boosting results according to the TP and FP rates metric are plotted upon the matrix containing the ROC point combinations of the investigated realizations. More specifically, these maps show the locations of the sets which result in (i) no boosting (blue elements or 0 in the colorbar), (ii) the optimal boosting case $\Delta TP_i > 0$ and $\Delta FP_i < 0$ (cyan elements or 1 in the colobar), (iii) the suboptimal boosting case $\Delta TP_i > -10$ and $\Delta FP_i < 10$ (yellow elements or 2 in the colorbar) and (iv) boosting of detection performance with respect to some rates and deterioration of performance with respect to other rates (dark red or 3 in the colorbar). As in the case of the simulated database, no regularity is to be observed with respect to the location of the elements of the aforementioned cases. Thus, it is confirmed that the coordinates of the ROC points which are being combined with AdaBoost do not correlate with the performance of the algorithm.

The analysis of the synthetic dataset showed that the overlapping of the CP values which correspond to the different hypothesis testing results constitute the decisive attribute which enhances detection performance. As in the previous analysis, the CP values of the two realizations used in AdaBoost were divided into CPs yielding tp, tn, fp and fn detections. Subsequently, the overlaps between those sets of the two realizations were quantified by means of the Bhattacharyya coefficient, resulting in 16 BC values. The BC values of all the sets examined within each realization are presented in Appendix C (see Figure C.11 to Figure C.16). For each set, the 16 individual BC values were summed up yielding overlap sums which are presented in Figure 7.14. Realization pairs R_1-R_2, R_1-R_4 and R_2-R_4 have higher BC sums than the other three realization pairs:

- The BC sums of R_1-R_2 lie on average around 8. This pair results in 12 sets of the optimal boosting case and 135 sets of the suboptimal boosting case.
- The BC sums of R_1-R_4 are greater than 8. This pair results in 41 sets in the optimal boosting case and 205 sets in the suboptimal boosting case.

- The BC sums of R_2-R_4 range from 7.5 to 12. This pair results in 48 sets in the optimal boosting case and 79 sets in the suboptimal boosting case.
- Realization pairs R_1-R_3 and R_2-R_3 have BC sums which range from 4 to 7.5, with most values lying around 7. These pairs result in 14 and 34 cases in the optimal and the suboptimal case, respectively.
- The BC sums of R_3-R_4 range between 4.5 and 7.5. This pair results in 11 sets in the optimal boosting case and 36 sets in the suboptimal boosting case.

These results are in agreement with the results of the simulated database since the realization pairs yielding in the optimal case and the suboptimal case are characterized by sets with high BC sums. The only exception is realization pair R_3-R_4, which delivers relatively low BC sum values despite the fact that some sets do yield the optimal and the suboptimal case results. Summarizing, it was shown that there is a correlation between high BC sum values and boosting performance. However, the fact that the sets with the highest BC sums do not coincide with the sets resulting in the optimal boosting case implies that the BC sum selected to be computed does not capture precisely the composition of the overlaps which result in boosting.



(a)

(b)

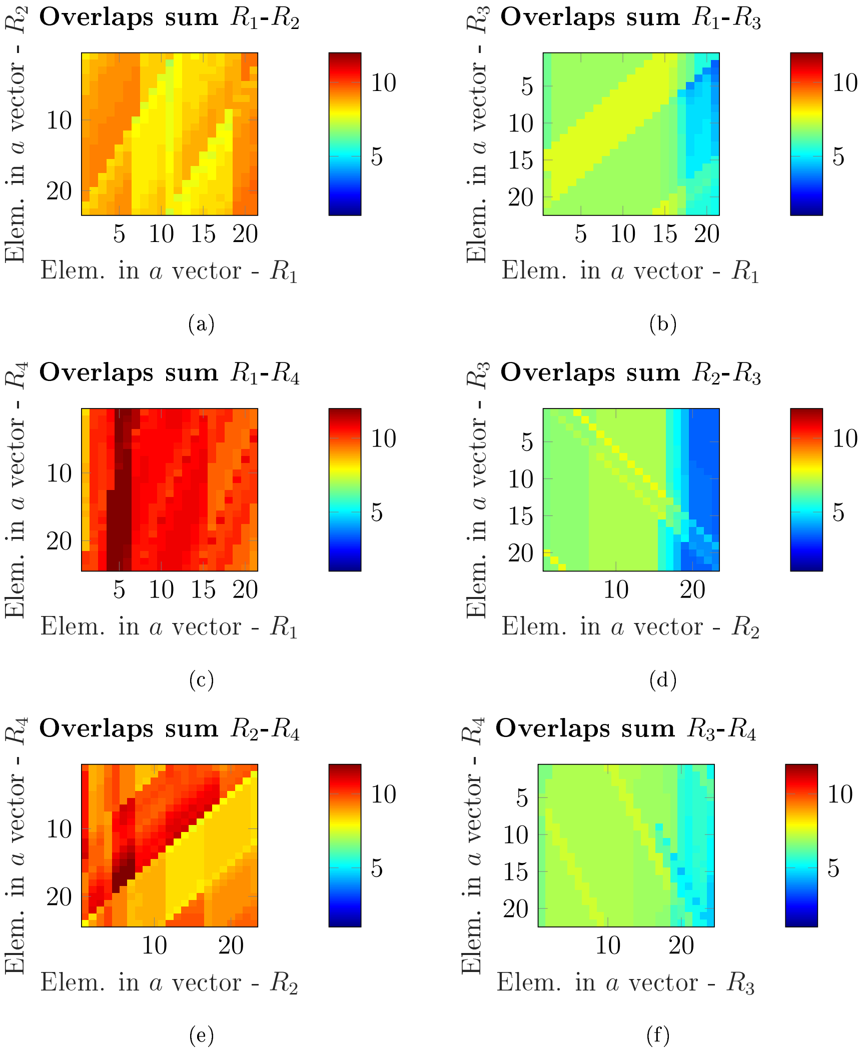
(c)

(d)

(e)

(f)

Figure 7.13: Boosting results according to TP and FP rate changes plotted upon the matrix containing the sets of the ROC points of the two realizations. $1 - a$ is the confidence interval and a is the significance level. Blue elements indicate no boosting (0 in colorbar). Cyan elements indicate the optimal case $\Delta TP_i > 0$ and $\Delta FP_i < 0$ (1 in colorbar). Yellow elements indicate the suboptimal case $\Delta TP_i > -10$ and $\Delta FP_i < 10$ (2 in colorbar). Dark red elements indicate all the other sets, which yield improvement of the one rate but deterioration of the other rate (3 in colorbar). Results presented for the database of the LANL wind turbine data.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7.14: Sum of Bhattacharyya coefficients of all overlaps (i.e., overlaps between the tp, tn, fp and fn detections of the two realizations), plotted upon the matrix containing the sets of the ROC points of the two realizations. $1 - a$ is the confidence interval and a is the significance level. Results presented for the LANL wind turbine data.

7.3.2.4 Improvement of CP Overlapping

The results of the synthetic dataset showed that omitting CP values of specific EOC clusters can be beneficial for boosting. This investigation was repeated for the operational database of the 3 kW wind turbine. The CP values of each cluster were successively omitted from the training and test sets and AdaBoost was employed. For instance, realization pair R_1-R_3 was formed by R_1 and R_3, which contained 5 clusters and 4 clusters, respectively. First, the clusters of R_1 were successively switched off and AdaBoost was performed for each reduced set. When switching off R_1 for R_1-R_3, 0.04% of the 462 combinations resulted in the optimal boosting case and 0.07% in the suboptimal boosting case. Subsequently, the clusters of R_3 were switched off one at a time and AdaBoost was employed for each set.

Table 7.11 summarizes the percentage of sets resulting in the optimal and the suboptimal boosting cases when the entire CP sets are considered (see second and third column), as well as when switching off clusters (see last four columns). The number of clusters which are switched off per realization are given in the fourth column, while the last four columns show the percentage of sets resulting in boosting when the clusters of the one realization (R_i) or the clusters of the other realization (R_j) are switched off. For instance, none of the 462 sets of realization pair R_1-R_3 result in the optimal boosting case and only 14 out of the 462 sets (i.e., 3.03% of the sets) result in the suboptimal boosting case when the entire CP set is considered. Realization R_1 has 5 clusters which may be switched off in order to improve the CP overlapping, whereas realization R_3 has 4. When switching off the clusters of R_1, 0.04% of the sets yield the optimal boosting case and 0.07% of the sets yield the suboptimal case. When switching off the clusters of R_3, 0.02% of the sets yield the optimal boosting case and 0.38% of the sets yield the suboptimal case. The detection performance is improved for all realization pairs (see third to sixth columns of Table 7.11) by switching off clusters. It is remarkable that the optimal case ($\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2$) is achieved even for pairs R_1-R_3 and R_2-R_3, which, as shown in Figure 7.13, did not deliver these optimal results when initially examined using the entire CP value set. The extent to which boosting performance is improved is presented in parallel coordinate plots in Appendix C (see Figure C.17).

Table 7.11: Percentage of sets resulting in the optimal and suboptimal boosting cases when all CP values are considered (second and third column) and when cluster CP values of the synthetic data are successively switched off (last four columns). Results presented for all realization pairs and for switching off clusters of R_i and R_j. Number of clusters in each realization is shown in the fourth column. Boosting case 1 ($\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2$) occurs for all cases examined except for switching off clusters of R_2 in R_2-R_4.

	Optimal case	Suboptimal case		Optimal case when switching off clusters of		Suboptimal case when switching off clusters of	
R_i-R_j	Number of sets resulting in optimal case	Number of sets resulting in suboptimal case	Number of clusters in (R_i, R_j)	R_i	R_j	R_i	R_j
R_1-R_2	12/483 (2.48%)	135/483 (27.95%)	(5, 1)	0.41%,	-	3.96%	-
R_1-R_3	0/462 (0%)	14/462 (3.03%)	(5, 4)	0.04%	0.02%	0.07%	0.38%
R_1-R_4	41/504 (8.13%)	205/504 (40.67%)	(5, 5)	6.87%	6.87%	2.71%	2.71%
R_2-R_3	0/506 (0%)	34/506 (6.71%)	(1, 4)	-	0.90%	-	0.51%
R_2-R_4	48/552 (8.69%)	79/552 (14.31%)	(1, 5)	-	9.88%	-	1.50%
R_3-R_4	11/528 (2.08%)	36/528 (6.81%)	(5, 4)	1.28%	1.21%	1.17%	0.58%

7.3.3 Cross-Validation

All the results presented so far were obtained by selecting the training data according to the scheme of Figure 4.7. In particular, the results of the operational database were obtained using at most 140 training datasets. This number is relatively low for supervised learning algorithms. In order to estimate the generalization error of AdaBoost, 3-fold cross-validation was performed for all the sets of the operational database examined. The available CP set was randomly divided into three folds - two were used in training, while the remaining one was used in testing. The same procedure was repeated three times, each time leaving out one fold to be used as validation set.

The cross-validation results are presented in Figure 7.15 in terms of the mean error rate, mean TP rate and mean FP rate of the three folds. The realizations are presented rowwise, i.e., the legend on the right contains the curve identities of the entire row. In some cases, dividing the set into three folds resulted in a very small number of datasets. Therefore, only the sets with more than 40 initial training datasets were cross-validated. As a consequence, cross-validation was performed for 336 out of 483 sets of R_1-R_2, 147 out of 462 sets of R_1-R_3, 420 out of 504 sets of R_1-R_4, 128 out of 506 sets of R_2-R_3, 361 out of 552 sets of R_2-R_4 and 140 out of 528 sets of R_3-R_4.

The mean error rate of the three folds is presented in Figure 7.15(a) for all the realization pairs. In general, AdaBoost error rates are higher than those of the framework realizations, but for some sets the error rate of AdaBoost is lower than that of either one or both framework classifiers. The mean TP rates of the

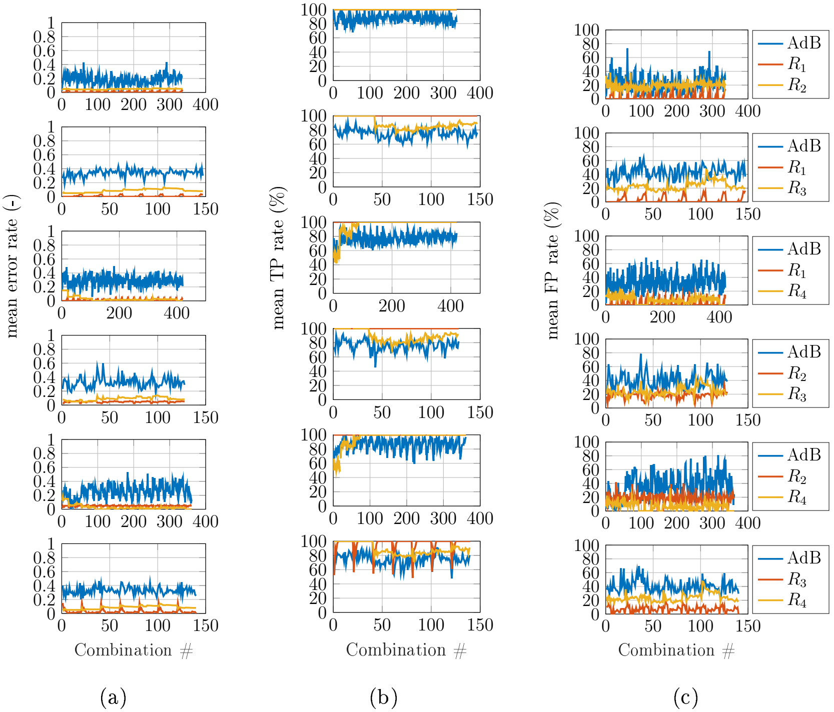


Figure 7.15: Mean error rate (a), mean TP rate (b) and mean FP rate (c) of the 3 folds. Realization pairs presented rowwise. The legends indicate the realization pair presented in each row.

three folds are shown in Figure 7.15(b). The TP rates of AdaBoost are generally lower than the TP rates of the individual framework classifiers. However, AdaBoost outperforms the framework classifiers for more sets in terms of TP rate than in terms of error rate. In particular, AdaBoost exhibits high TP rates for many sets of R_1-R_3 and R_3-R_4. Finally, the mean FP rates are presented in Figure 7.15(c). As in the previous figures, AdaBoost FP rates are on average higher than the FP rates of the framework realizations, but there are many combinations for which AdaBoost outperforms the framework classifiers. The cross-validation results are in accordance with the results obtained for selecting the training dataset based on the scheme of Figure 4.7, which were presented earlier in this chapter.

7.3.4 AdaBoost with Three Realizations

Subsequently, AdaBoost was employed in order to build a new classifier by combining the CP values and the decisions of three framework realizations. Four triads occur when combining the four realizations selected: R_1-R_2-R_3, R_1-R_2-R_4, R_1-R_3-R_4 and R_2-R_3-R_4. Each realization was evaluated for 48 confidence intervals, resulting in a total number of $48^3 = 110592$ possible sets. However, the weak learning assumption was only satisfied for 23, 21, 22 and 24 ROC points respectively, significantly reducing the number of combinations. Finally, 10262 sets were analyzed for R_1-R_2-R_3, 11592 for R_1-R_2-R_4, 11088 for R_1-R_3-R_4 and 10000 for R_2-R_3-R_4.

Table 7.12 provides an overview of the number of sets satisfying the weak learning assumption and the number of sets resulting in the optimal and suboptimal boosting cases. The optimal boosting case is obtained for only 82 sets of R_1-R_2-R_4, while the suboptimal case is obtained for all the triads. More specifically, the suboptimal boosting case is obtained for 99 combinations of R_1-R_3-R_3, 1705 combinations of R_1-R_2-R_4, 160 combinations of R_1-R_3-R_4 and 421 combinations of R_2-R_3-R_4.

In Table 7.13, the mean values and standard deviations of the TP and FP rate differences between the AdaBoost classifier and the framework classifiers are listed for the example of R_1-R_2-R_4. It is evident that the TP rates of the sets with $\Delta TP_i > 0$ and $\Delta FP_i < 0$ for $i = 1, 2, 3$ increased significantly. The TP rate of AdaBoost is on average 24.11 percentage points higher than the TP rate of R_1, 36.29 percentage points higher than that of R_2 and 24.28 percentage points higher than that of R_4. As for the sets with $\Delta TP_i > -10$ and $\Delta FP_i < 10$ for $i = 1, 2, 3$, i.e., when minor deteriorations of some rates are tolerated, the AdaBoost classifier outperforms the framework classifiers in terms of TP rates, with increases which are lower than those of the optimal case but reach on average up to 23.47 percentage points. The FP rate of AdaBoost, on the other hand, is on average 2.98 percentage points lower than that of R_2 and 3.87 percentage points lower than that R_4, but is on average higher than that of R_1 by 3.44 percentage points.

Table 7.12: Number of combinations satisfying the weak learning assumption and number of combinations resulting in the optimal and suboptimal boosting cases - AdaBoost with three realizations for the LANL wind turbine.

Triads	R_1-R_2-R_3	R_1-R_2-R_4	R_1-R_3-R_4	R_2-R_3-R_4
Number of sets satisfying the weak learning condition	10626	11592	11088	12444
$\Delta TP_i > 0$ and $\Delta FP_i < 0$	0	82	0	0
$\Delta TP_i > -10$ and $\Delta FP_i < 10$	99	1705	160	421

Table 7.13: Mean values and standard deviations of the TP and FP rate differences for all boosting cases of triad R_1-R_2-R_4 of the operational data.

Boosting case	$(\mu_1 \pm \sigma_1)$, $(\mu_2 \pm \sigma_2)$, $(\mu_4 \pm \sigma_4)$ TP % changes with respect to R_1, R_2 and R_4	$(\mu_1 \pm \sigma_1)$, $(\mu_2 \pm \sigma_2)$, $(\mu_4 \pm \sigma_4)$ FP changes with respect to R_1, R_2 and R_4
$\Delta TP_i > 0$ and $\Delta FP_i < 0$	(24.11±14.32), (36.29±11.28), (24.28±12.22)	(-5.63±3.51), (-7.75± 2.25), (-8.49±5.05)
$\Delta TP_i > -10$ and $\Delta FP_i < 10$	(7.94±14.69), (23.47±14.15), (19.93±20.12)	(3.44±6.05), (-2.98±5.65), (-3.87±7.03)

7.4 Summary and Conclusions

The concept of implementing AdaBoost in order to combine the decisions of different damage features and to create a new and more effective decision rule was validated in this chapter. A three-tier SHM framework was used to analyze structural data and to deliver decisions regarding the structural state. By selecting different settings along the three tiers, several framework realizations were obtained. The classifiers of these realizations were used as component classifiers to train AdaBoost.

The integration of AdaBoost with the three-tier SHM framework was validated on the operational data of a 3 kW wind turbine located at Los Alamos National Laboratory (LANL), as well as on synthetic data generated by a numerical model of the LANL wind turbine. The synthetic and operational databases were analyzed using the SHM framework for various CPs, clustering and HT settings. Four realizations were selected for the boosting stage, where they were combined pairwise to train AdaBoost classifiers. The error rate, the TP rate and the FP rate of the AdaBoost classifier were compared to those of the individual framework classifiers. For the operational data, which contained few datasets, 3-fold cross-validation was employed in order to obtain an estimation of the classifier generalization error. Furthermore, AdaBoost was implemented for combining three framework realizations.

It was shown that changes in the TP rate and the FP rate are more suitable metrics for the assessment of boosting performance than changes in the error rate. The superiority of the TP and FP rates is due to the notion of error rate, which contains both false positive and false negative detections. As a result, measuring the detection performance with respect to the error rate does not offer information on the nature of the false detections, a piece of information which is very useful in the context of SHM.

The framework realizations were evaluated for a series of confidence intervals, which yield different detections. At the boosting stage, all the confidence intervals analyzed were taken into account resulting in a large number of sets for investigation. Improved detection performance was achieved for a relatively small number of those sets. However, when boosting was achieved, it was entailed by substantial improvement of the TP and FP rates. The improvements of the TP and FP rates were presented exemplarily for some realizations. Tables 7.10 and 7.4 present the improvements for a realization pair of the simulated data and for a realization pair of the operational data, whereas Table 7.13 presents the improvements for a triad.

These tables indicate that the TP rates of the realizations involved have improved more than the FP rates. The average of TP improvements reaches up to approximately 30% for the operational realization pair and up to 36% per realization when combining three realizations. The average FP improvements, on the other hand, are lower, reaching up to approximately 5% when combining two realizations and 8.5% when combining three realizations. Table 7.10 shows, however, that this is not the rule, i.e., AdaBoost does not perform better at improving TP rates than FP rates. This results from the fact that AdaBoost improves the TP rates of R_4 more than the TP rates of R_3, but at the same time improves the FP rates of R_3 more than the FP rates of R_4. Hence, it can be stated that AdaBoost improves both rates of a realization, while one of the two rates experiences higher improvements, depending on the application.

The most significant conclusion of this chapter is that boosting performance is not dependent on the coordinates of the ROC points characterizing the realizations to be combined, i.e., on the TP rate, the FP rate and, indirectly, on the confidence interval. Instead, boosting performance depends on the relative positioning of CP values on the CP space. This relative positioning can be expressed by the overlaps of CP distributions which correspond to tp, tn, fp, and fn detections. For the quantification of these overlaps, the Bhattacharyya coefficient was suggested. More specifically, 16 BC values which describe the overlaps between the detections of the two CPs considered, were calculated. Those coefficients were summed up in order to build one metric quantifying the overlaps between two sets of CPs and their detections. Realization pairs with high BC sums (values higher than 8) resulted in AdaBoost classifiers which outperformed the framework classifiers, or in classifiers which improved the one rate but caused deterioration of the other rate by up to 10 percentage points. Lower values of BC sums resulted in lower boosting performance, i.e., the improvement of only the TP rate or only the FP rate.

The sum of the Bhattacharyya coefficients were able to serve as an indicator for estimating in advance which CP sets were expected to deliver good boosting results. However, it was not possible to identify the sets resulting in the optimal boosting case based on the BC sum. In addition, it was not possible to define a threshold on the BC sum which is able to separate sets with good boosting potential from others. Hence, it can be concluded that the BC sum, i.e., the sum of the overlaps between all detections of two CPs, is not a proper metric for the identification of sets which yield the optimal boosting results. The drawback of the BC sum is that overlaps with positive effect are added to overlaps with negative effect. A more complex metric, such as the weighted sum of the individual BC values, could lead to the identification of the CP sets appropriate for boosting.

The shortcoming of not having the optimal overlaps within a set was remedied by omitting CP values belonging to specific EOC clusters from the training and the test sets of AdaBoost. Hence, it was shown that switching off clusters results in altered CP overlaps which can be beneficial for boosting. The training set is, in this manner, adjusted in order to achieve higher performance.

In this chapter, AdaBoost was used in order to boost the performance of classifiers obtained from an SHM framework which employs vibration-based damage features. However, AdaBoost can also be employed to combine damage features of any nature, such as features obtained from ultrasonic testing or acoustic emission testing. The availability of datasets from both the healthy and the damaged state of the structure is a prerequisite for the implementation of AdaBoost. At present, AdaBoost can only be employed in applications where damaged datasets are available. The deployment of the suggested concept in operation, when no information from the damaged structure is available, can be enabled by accurate numerical models of the structure to be investigated. More specifically, datasets can be generated from a model that is an accurate representation of the structure investigated. The datasets can be analyzed using different realizations of the SHM framework, and the AdaBoost classifier, which combines the decisions of the individual realizations, could be built. Subsequently, the AdaBoost classifier can be used in operation, in order to classify the CP values obtained from the analysis using the framework.

Chapter 8

Quantifiability of CP Sensitivity to Damage and EOC Variability

Structural health monitoring is implemented by measuring structural data and transforming them into some alternative form, where the correlation with damage is more apparent [21]. This is achieved by fitting physics-based or data-based models to the data. This process is called feature extraction, while the damage-sensitive features are simply referred to as damage features. Alternatively, these features are called condition parameters (CPs), implying their use to describe and monitor the condition of the structure. In this thesis, the terms "damage feature" and "condition parameter" are used as equivalent. Simultaneous observation of several CPs is beneficial since CPs deriving from different models capture different damage scenarios and effects. For instance, some CPs exhibit higher sensitivity to damage or to EOC changes than others, depending on the model they are obtained from as well as the entailed assumptions.

In this thesis, a new damage feature was introduced, while several other damage features were also implemented. CP^{ARX} was derived from an output-only ARX model, which constitutes a transmissibility function model, by observing the migration of the transmissibility function poles. That CP was implemented and validated in the context of unsupervised machine learning (see Chapters 5 and 6). Furthermore, CPs already existing in literature were employed and their classifiers were used to train AdaBoost in order to create a new decision rule. Two of those CPs were based on a VAR model (CP^{M} and CP^{R^2}), while the third CP was obtained from the SSI method (CP^{ϵ}). Finally, the most established and widely used damage features, i.e., the natural frequencies, were presented in chapter 5 for the modal test of a wind turbine rotor blade.

The objective of this chapter is to provide an overview of the sensitivity to damage and to changes in the EOCs for the aforementioned CPs. For this purpose, the theoretical background of the corresponding models and the entailed assumptions are mentioned. Datasets obtained from the modal test of a 34 m rotor blade are used in order to show the sensitivity of these CPs to structural damage and to added mass. In this manner, CP^{ARX} is compared to the natural frequencies and to the damping

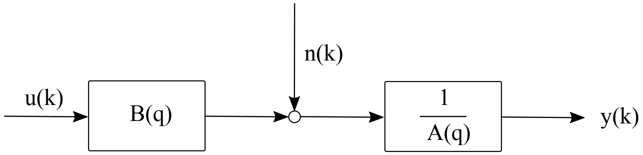


Figure 8.1: Block diagram of the ARX model.

ratios. Furthermore, datasets from the fatigue test of the same rotor blade are used in order to compare the CPs with respect to their sensitivity to damage and to EOC changes.

8.1 Considered Damage Features

CP^{ARX} is obtained from an output-only ARX model. The fact that structural responses are used instead of an exogenous input renders it a transmissibility function model. The solutions of the transmissibility function denominator are observed in order to identify structural changes. To be more specific, the TF poles of the training datasets are calculated and clustered in groups with similar properties. Subsequently, the TF poles of the current dataset are calculated and the MSD between these and the cluster they belong to is used as a CP. The only assumption involved in the ARX model is that the noise term $n(k)$ (i.e., the disturbances) enters the process at an early stage and, more specifically, before the system dynamics (see Figure 8.1). In real-life applications, though, disturbances are also observed at the system output, such as noise introduced by sensors or by the data acquisition system.

CP^M and CP^{R^2} are both derived from a VAR model. CP^M is built by employing a statistical M-test on the error covariance matrix. The M-test compares the error covariance matrix of two data instances, namely the error covariance matrix of the dataset to be tested, and that of the reference dataset [71]. According to Neter, CP^{R^2} employs the coefficient of determination (R^2), which makes use of the actual time series values (e.g., the acceleration signal), the model estimation and the mean value of the signal measured [70]. VAR models are time series models which only take into account the system output. These models assume that the output is stationary, i.e., that it is characterized by contstant statistical moments. This assumption is often violated in real applications. Nonstationarity of the time series observed (for instance structural responses) can be remedied to some extent by performing data clustering.

CP^{ϵ} is based on the interpretation that the singular vectors matrix U, which is obtained from the singular value decomposition of the covariance Hankel matrix, is the left nullspace of the Hankel matrix [68]. The Hankel matrix of the current datasets and matrix U_{ref} of a reference dataset are used to build a residue. In this work, the modal parameters, i.e., the natural frequencies and the damping ratios, were identified using the SSI approach, while the TEMP algorithm was used in order to reduce the number of solutions. The SSI approach assumes white-noise inputs. The assumption of white-noise excitation is violated in many real applications which include some dominating frequency, such as the rotor frequency of a wind turbine. Furthermore, both the measurement noise and the process noise are also assumed to be white.

Table 8.1 summarizes the characteristics of the aforementioned CPs by presenting the models they derive from and the corresonding assumptions.

Table 8.1: Condition parameters, model they are derived from and corresponding model assumptions.

condition parameter	model	model assumptions	derivation of CP from	quantity defining the CP
CP^{ARX}	output-only ARX	noise is white and enters the process before the system dynamics	transmissibility function poles	MSD of TF poles to the TF poles of the training set
CP^{M}	VAR	stationarity	model error	M-test on error covariances
CP^{R^2}	VAR	stationarity	time series and model estimate	coefficient of determination
CP^{ϵ}	SSI	white-noise excitation	Hankel matrix of current dataset and nullspace of a reference Hankel matrix	coefficient of determination
f, ζ	SSI	white-noise excitation	system matrix A	natural frequencies, and damping ratios

8.2 Sensitivity to Structural Changes

The modal test of the 34 m wind turbine, which was thoroughly presented in section 5.3.2, involved manual excitation of the structure in the healthy state, at the four steps of ice accretion and in the damaged state. That database was used in order to compare the CPs investigated with respect to their sensitivity to structural changes. In order to achieve that, the CP values were calculated for all the states and the mean values of the CPs were detemined within each state.

Table 8.2 offers a comparison between the changes in the CPs investigated for the excitation of the first natural frequency in the flapwise direction. The presented results include the changes in the first six natural frequencies and the corresponding damping ratios, the changes in all reference-based CPs (CP^M, CP^{R^2} and CP^ϵ) as well the changes in CP^{ARX5}, i.e., the changes in the MSDs of the TF poles of model 5. The i in $CP_{cl.i}^{ARX5}$ denotes the number of the TF pole cluster on the z-plane. The table presents the mean value of the CPs for the healthy state and the percentage change of the CP mean value for the other states.

All aforementioned condition parameters are subjected to changes due to added mass and due to the damage at the trailing edge. The mean values of $CP_{cl.i}^{ARX5}$ in the healthy state range from 1.293 to 1.876. The values of CP^{ARX5} change drastically both due to ice accretion and due to damage, with the percentage increases ranging from 15.18% to as much as 25211%. This implies that the MSDs of the TF poles at these states are significantly higher than the rather low MSDs of the healthy state, and that the TF poles have migrated significantly on the z-plane. The only decrease is observed for $CP_{cl.3}^{ARX5}$ at ice steps 2 and 3. The effect of damage is more obvious for clusters 1 and 3, while the remaining clusters exhibit higher sensitivity to added mass. One single CP could be built out of these five CPs by simply adding the percentage changes of the five clusters. This corresponds to simultaneously observing the MSDs of all TF poles with respect to the mean values of MSDs calculated for the clusters obtained from the healthy state.

The reference-based CPs are also subjected to changes due to ice accretion and damage. CP^M and CP^ϵ increase significantly due to structural changes, with the percentage changes of CP^M ranging between 641.3% for ice step 2 and 4271.9% for the damaged state, and the changes of CP^ϵ ranging between -445.51% and +1192.2%. CP^{R^2}, on the other hand, is less sensitive to structural changes, but still decreases between 5.39% for ice step 2 and 11.06% for the damaged state. Correlation between the changes in the reference-based CPs and the amount of added mass is available for setups 2.2, 2.3 and 2.4, but not for setup 2.1. Finally, it may be observed that reference-based CPs are more sensitive to damage at the trailing edge than to added mass since percentage differences are higher for setup 4 than for setups 2.1 to 2.4.

The natural frequencies and the damping ratios also present variations due to added mass and damage but are less sensitive than all aforementioned CPs. The reduction of the natural frequencies ranges from 0.05% to 3.63%. Furthermore, with increasing added mass, there is a decrease of natural frequencies for all the frequencies except for the 3^{rd}, 5^{th} and 6^{th}. The fact that most of the natural frequencies decrease as more mass is added to the blade shows that the natural frequencies have the potential of quantifying structural changes. Changes in damping ratios range from -33.03% to +822%, indicating that structural damping has high sensitivity to damage and that it can also be used as a CP. However, damping variations do not correlate with the amount of added mass, as it was observed for the frequencies.

Table 8.2: CP values and natural frequencies identified for the healthy state (setup 1) and percentage changes for ice accretion steps and the damaged state (setups 2 and 4), for manual excitation of the 1^{st} flapwise natural frequency.

	Setup 1 (healthy)	Setup 2.1 (ice step 1)	Setup 2.2 (ice step 2)	Setup 2.3 (ice step 3)	Setup 2.4 (ice step 4)	Setup 4 (damaged)
$CP_{cl.1}^{ARX5}$	1.293	+2728.60%	+642.70%	+196.04%	+49.16%	+3608.90%
$CP_{cl.2}^{ARX5}$	1.722	+3715.90%	+50.47%	+15.18%	+449.47%	+803.97%
$CP_{cl.3}^{ARX5}$	1.446	+1137.20%	+227.81%	-12.65%	-18.88%	+1311.1%
$CP_{cl.4}^{ARX5}$	1.876	+69466%	+6059.3%	+1752.70%	+330.38%	+25211.00%
$CP_{cl.5}^{ARX5}$	1.661	+7680.10%	+144.98%	+171.89%	+133.54%	+6648.6%
CP^M	285.34	+2275.6%	+641.3%	+1225.5%	+1387.2%	+4271.9%
CP^{R^2}	0.75	-7.88%	-5.39%	-5.75%	-8.22%	-11.06%
CP^{ϵ}	$7.21 * 10^{-6}$	+445.51%	+537.73%	+793.48%	+1011%	+1192.2%
f_1	1.068 Hz	-0.26%	-1.10%	-1.95%	-2.79%	-1.33%
ζ_1	0.82%	-16.46%	-17.85%	-19.80%	-26.16%	-5.87%
f_2	1.689 Hz	-0.70%	-2.44%	-3.03%	-3.63%	-2.42%
ζ_2	0.46%	+248.31%	+459.13%	+822.00%	+353.45%	+660.03%
f_3	3.132 Hz	-0.11%	-2.26%	-2.16%	-3.14%	-1.54%
ζ_3	0.38%	+442.00%	+634.22%	+8.80%	-16.80%	+276.89%
f_4	5.640 Hz	-0.05%	-0.91%	-1.65%	-2.26%	-1.37%
ζ_4	0.34%	+135.19%	-7.28%	-6.80%	+12.13%	12.13%
f_5	6.618 Hz	+0.23%	-0.88%	-1.05%	-1.45%	+0.05%
ζ_5	0.39%	+25.83%	+7.09%	+13.51%	-3.95%	-3.95%
f_6	11.596 Hz	-0.90%	-0.81%	-1.26%	-1.50%	-0.91%
ζ_6	0.55%	-30.14%	+28.77%	-33.03%	-17.20%	-17.20%

8.3 Sensitivity to EOC Changes

The database of the rotor blade fatigue test (see section 5.4.1) was used to investigate the sensitivity of the CPs to EOC changes. During the fatigue test, the rotor blade was excited at its first natural frequency in the edgewise direction for more than one million cycles. The load levels applied to the rotor blade constitute the only available varying EOC. The blade states during the experiment can be summarized as follows:

1. Healthy state (steps 1-7, i.e., load level ranging from 70% to 110%)
2. Fatigue microcracks and insignificant damage, i.e., small delaminations and small material or bondline cracks (steps 8-10, i.e., load level ranging from 120% to 140%)
3. Trailing edge damage (step 11, i.e., load level 170%)
4. Crack propagation (steps 12-17, i.e., load level ranging from 50% to 130%)

The database was analyzed using the three-tier SHM framework for different settings along the three tiers. Two data clustering cases were examined: (i) Man1, which did not involve data clustering and (ii) Man2, which involved data clustering according to load level. In the second tier, the reference-based CPs CP^M, CP^{R^2} and CP^ϵ as well as the CP introduced in this thesis (CP^{ARX}) were calculated. In HT, the desicion boundaries were defined based on (i) the assumption of the Gaussian distribution and (ii) the percentiles of the CPs. All CPs were evaluated for a series of confidence intervals. The settings used in the analysis of the database within SHM framework are presented in Table 8.3.

During the fatigue test, material flaws due to fatigue were observed before damage occurred at the trailing edge. Therefore, different definitions with respect to the damaged datasets were obtained depending on whether or not the fatigue effects were considered as damaged. The cases considered in this investigation are the following:

- The damaged datasets start at the beginning of load level 120%, i.e., after step 7. This case includes the early stages of fatigue as well as damage at the trailing edge.
- The damaged datasets start at the beginning of load level 130%, i.e., after step 8. This case includes fatigue signs at an intermediate stage as well as the damage at the trailing edge and its propagation.
- The damaged datasets start at the beginning of load level 170%, i.e., after step 10. This case includes few datasets prior to the damage at the trailing edge, as well as the damage and its propagation.

The results for CP_i^{ARX} were presented in chapter 6 using ROC curves and their AuC values (see Figures 6.5 to 6.9). Man2 of CP_i^{ARX} outperformed Man1, showing that data clustering is beneficial and that this CP exhibits some sensitivity to EOC changes. Moreover, it was shown that most of the AuC values increase as one moves the boundary of the damaged datasets from the beginning of 120% to the beginning of 170%. The detection performance of CP_i^{ARX} is generally very good since it has high AuC values, especially when detecting damage at the trailing edge (see Figure 6.9).

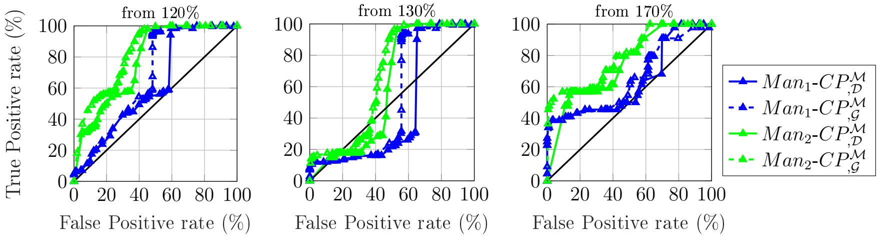
Table 8.3: Settings for analysis of setup 3 (fatigue test) within the SHM framework

Number of channels	10	
Sampling frequency	50 Hz	
Number of datasets	1162	
Training datasets	357	
Testing datasets	810	
Settings for system identification		
Data-driven SSI	time shifts=40, model orders=40-250	
TEMP	f_{crit}=20, ζ_{crit}=20, MAC=0.88, ϕ_{crit}=25, path length=20	
Classification names	Man1	Man2
Number of clusters	1	7
Settings for condition parameters		
CP^M/CP^{R^2}	AR-model order=35	
CP^ϵ	time shifts=4, significant columns=10	
CP_i^{ARX} for $i = 1, 2.., 10$	$n_a(i) = [11, 10, 10, 10, 10, 10, 8, 8, 8, 10]$	
Analyzed distribution types: discrete, Gaussian		
Alpha-values for hypothesis testing: a=[0.1:0.1:0.4, 0.5:0.5:10, 11:1:20, 25:5:45, 55:5:75, 80:1:90, 90.5:0.5:95, 95.6:0.1:99.9]%		

It can be said that CP_i^{ARX} is generally more sensitive to the damage at the trailing edge than to the structural changes due to fatigue. The only exceptions are models 2 and 6, which provide perfect detection when considering that the fatigue signs at an early stage also indicate damage. As far as the settings in HT are concerned, it was shown that the assumption of discrete distribution results in higher TP rates compared to the Gaussian distribution.

The ROCs of CP^M are presented in Figure 8.2 for different assumptions regarding the damaged datasets. Data clustering is beneficial when implementing CP^M since the results for Man2 outperform the results of Man1. This implies that CP^M also exhibits some sensitivity to varying EOCs. Nevertheless, its detection performance is relatively poor, as indicated by the small areas under the curve. The highest overall detection performance is achieved when the damaged datasets start at the beginning of load level 120%, while the second best overall detection performance is obtained when assuming that only the damage at the trailing edge is regarded as damage. Finally, it may be observed that the Gaussian distribution yields slightly better results than the discrete distribution.

Figure 8.3 presents the ROC curves of CP^{R^2} for the three definitions of dam-

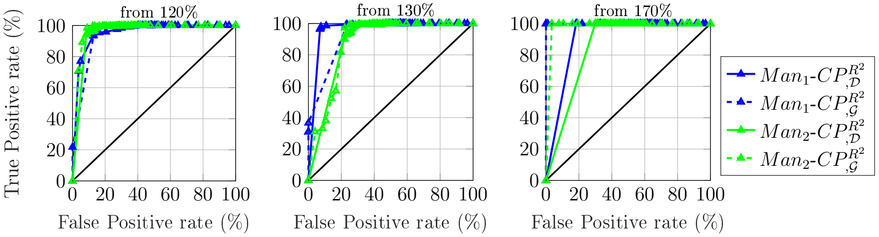


(a) (b) (c)

Figure 8.2: ROC curves of CP^M for the rotor blade fatigue test database, when assuming that damaged datasets start at the beginning of (a) load level 120% (step 12), (b) load level 130% (step 13) and (c) load level 170% (step 14). ROC curves presented for Man1 (no data clustering) and for Man2 (data clustering according to load level) as well as for the assumptions of the discrete and the Gaussian distributions in HT.

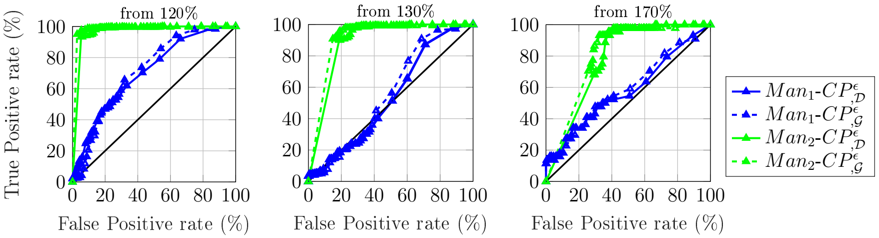
aged datasets. The overall detection capability of CP^{R^2} is excellent for all cases, as implied by the areas under the ROC curves. In particular, when assuming that the damaged datasets start at the beginning of load level 170%, there are no false negative detections (TP rate equal to 100%), while one case yields the perfect detection, i.e, 100% TP rate and 0% FP rate. The effect of data clustering as well as effect of the selected distribution are minimal. The fact that data clustering has no effect on detection performance shows that CP^{R^2} is not influenced by EOC changes.

The ROC curves of CP^{ϵ} are presented in Figure 8.4. For this CP, data clustering results in substantial improvement of the detection accuracy, showing that CP^{ϵ} is especially sensitive to EOC changes. The effect of the assumed distribution in HT, though, is minimal, with the two distributions yielding almost identical results. When the early fatigue signs are included in the damaged datasets, the detection capability of CP^{ϵ} for Man2 is almost excellent, with very high TP rates and the ROC points being very close to the ROC curve optimum (see Figure 8.4(a)). The results are slightly worse when assuming that the damaged datasets start at the beginning of load level 130% and deteriorate when the damaged datasets only include the damage at the trailing edge and its propagation. As a result, CP^{ϵ} performs better at detecting structural changes due to fatigue than at detecting the damage at the trailing edge.



(a) (b) (c)

Figure 8.3: ROC curves of CP^{R^2} for the rotor blade fatigue test database, when assuming that damaged datasets start at the beginning of (a) load level 120% (step 12), (b) load level 130% (step 13) and (c) load level 170% (step 14). ROC curves presented for Man1 (no data clustering) and for Man2 (data clustering according to load level) as well as for the assumptions of the discrete and the Gaussian distributions in HT.



(a) (b) (c)

Figure 8.4: ROC curves of CP^{ϵ} for the rotor blade fatigue test database, when assuming that damaged datasets start at the beginning of (a) load level 120% (step 12), (b) load level 130% (step 13) and (c) load level 170% (step 14). ROC curves presented for Man1 (no data clustering) and for Man2 (data clustering according to load level) as well as for the assumptions of the discrete and the Gaussian distributions in HT.

8.4 Conclusions and Discussion

Data from the modal test and the fatigue test of a 34 m rotor blade were used in order to compare different condition parameters with respect to their sensitivity to damage and their sensitivity to varying EOCs. The investigated CPs were CP_i^{ARX}, which was introduced earlier in this thesis, two CPs based on a VAR model (CP^M and CP^{R^2}), one CP based on the SSI method (CP^ϵ) as well as the natural frequencies and the damping ratios.

All aforementioned CPs were calculated in the modal test for the healthy state, the four steps of ice accretion and the damaged state. All CPs exhibited changes due to both added mass and damage. CP^{ARX}, the reference-based CPs and the damping ratios were shown to be more sensitive to structural changes than the natural frequencies. However, it was shown that the natural frequencies are also capable of quantifying the amount of added mass. While the natural frequencies are more sensitive to added mass than to damage, the reference-based CPs exhibit higher sensitivity to damage than to added mass. For CP^{ARX}, the sensitivity to damage and added mass varies depending on the TF pole cluster observed. Each feature reacts in a different way and with different sensitivity to damage and to change due to added mass. Hence, it can be concluded that parallel observation of multiple damage features is advantageous for SHM applications.

The three-tier SHM framework was employed in order to evaluate the fatigue test database. CP^{ARX} and the reference-based CPs were calculated in that context for two clustering cases and the results were presented in the form of ROC curves. It was shown that CP^{ARX} and CP^M exhibit moderate sensitivity to EOC changes since higher detection performance is obtained when the data are clustered according to the applied load level. On the other hand, the sensitivity of CP^{R^2} to varying EOCs is very low. It is evident that CP^ϵ exhibits very high sensitivity to damage since the results are significantly improved when performing data clustering according to the EOCs.

Chapter 9

Summary, Conclusions and Future Research

9.1 Summary

The research presented in this thesis is based on the perception that SHM should be implemented in a more holistic framework which takes into account the effect of the environmental and operational conditions and investigates damage-sensitive features deriving from multiple methods as well as several decision-making rules. A further perception is that the simultaneous observation of various damage features is advantageous since each damage feature is capable of capturing different effects. This thesis was motivated by the fact that a large number of the currently employed approaches in the field of SHM are based on assumptions which are often violated in real-life applications and by the necessity of exploiting the information provided by different damage features.

This research was carried out in three directions: (i) the conception and the investigation of a new condition parameter obtained from a robust model which is not based on any significant assumption, (ii) the exploitation of the decisions of several condition parameters in order to attain a more accurate decision rule and (iii) the comparative study of the sensitivity of various condition parameters.

9.1.1 Investigation of a Novel Condition Paremeter

A new CP capable of monitoring vibrating structures was introduced and investigated in the context of unsupervised machine learning. It was based on the concept of monitoring structural changes using transmissibility function pole migration and was derived from an output-only ARX model by observing its poles. The modified ARX model constitutes a transmissibility function model, while the transmissibility function (TF) poles are essentially system zeros. The new CP was built using the Mahalanobis squared distance and expresses the migration of the TF poles due to structural changes. The advantage of this new damage feature lies in the fact that (i) it does not rely on assumptions, such as these of stationarity or white-noise excitation,

(ii) it does not require knowledge of the system's excitation, and (iii) it encapsulates both frequency and damping information. Preliminary analyses were performed using several databases in order to investigate the effect of several factors, such as the model order and the input channel to the TF pole positions as well as to the final detection performance. Furthermore, the new CP was implemented within a three-tier SHM framework, which can be employed to monitor structures in unsupervised mode, i.e., when only data from the healthy structure are available during the training phase of the monitoring process. Finally, the concept was validated on the data of a rotor blade fatigue test.

9.1.2 Combination of Damage Feature Decisions

This work also addresses the implementation of adaptive boosting in order to combine the decisions of several damage features and to attain a new, more accurate decision rule. AdaBoost combines several classifiers in order to obtain a new decision rule, while the selection of the training data within the algorithm is adaptive in nature. The presented concept can be employed after any SHM procedure which delivers decisions regarding the structural state, as long as the damage features involved belong to the same feature space and have the same form. In this thesis, AdaBoost was integrated with a three-tier SHM framework, which yields different realizations depending on the selected settings along the three tiers. Each realization results in a classifier, which can be used as a component classifier in AdaBoost. The concept of implementing AdaBoost after the three-tier SHM framework was outlined and several aspects were discussed. Those involved (i) the assumptions of the algorithm, (ii) the selection of training data for the AdaBoost algorithm when few data sets are available, (iii) the necessity of cross-validation in order to obtain an estimate of the algorithm's generalization error and (iv) the boosting metrics used for the evaluation of the results. The integration of the SHM framework with AdaBoost was validated on the data of a 3 kW wind turbine.

9.1.3 Comparison of CP Sensitivity

Another objective was to provide an overview of the sensitivity of several damage features, including the one suggested in this thesis. For this purpose, the theoretical principles of the models, the involved assumptions and the quantities defining the condition parameters were briefly presented. The sensitivity of these condition parameters to structural changes and to varying environmental and operational conditions (EOCs) was investigated based on the modal test data and the data obtained from the fatigue test of a rotor blade. The CPs investigated were: the proposed CP^{ARX}, two CPs deriving from a VAR model (CP^{M} and CP^{R^2}), one CP based on the SSI method (CP^{ϵ}), as well as the natural frequencies and the damping ratios which were identified using the SSI method and the TEMP algorithm.

9.2 Conclusions

9.2.1 Investigation of a Novel Condition Parameter

The preliminary analysis regarding the sensitivity of TF poles offered an insight into the mechanism of TF pole migration. The selection of the model order was shown to be crucial, since it can contribute to avoiding effects propagating to the decision making stage. Therefore, a procedure for the selection of the optimal model order was proposed. That procedure was based on the model's mean squared error and the aggregation of the TF poles on the z-plane. The analysis showed that TF pole positions are affected by structural changes, such as damage and added mass, but are also affected by EOC changes. Hence, data clustering according to the prevailing EOCs is necessary and expected to enhance the detection performance. Depending on the selected model order, several TF poles are obtained. The investigations showed that not all TF poles exhibit the same sensitivity to structural changes. Hence, parallel observation of all TF poles is beneficial and enables the capture of different events. While it was shown that TF pole positions vary for different EOC settings and for different extents of damage, the potential of damage quantification could not be proven. The conclusions of this preliminary analysis served as a guideline for the implementation of the novel CP in the context of an SHM framework which employed unsupervised learning.

The new condition parameter (CP^{ARX}) was implemented in the context of the three-tier SHM framework and validated on the data of a rotor blade fatigue test. CP^{ARX} was built using the Mahalanobis squared distances (MSDs) between the TF poles of the current dataset and the corresponding TF pole clusters obtained in the training phase. A statistical analysis showed that some TF pole clusters approximate the Gaussian distribution, while others deviate significantly from Gaussianity.

The employment of CP^{ARX} yields very good performance for structural changes of different types. The ROC curves of the models analyzed show that high performance is obtained, with the ROC points lying close to the ROC optimum. The AuC values provide the overall performance of the model, i.e., the performance for different confidence regions. High AuC values show that the new CP is capable of detecting both the damage at the trailing edge and the fatigue signs prior to its occurrence. However, the CP proves more accurate at detecting fatigue signs at an intermediate stage, and even more accurate at detecting the damage at the trailing edge. The performance of realizations which do not involve data clustering according to the EOCs are idependent of the selected distribution type in HT, showing that the TF poles follow the Gaussian distribution when no data clustering is performed. On the contrary, the distribution considered in HT affects the detection performance when data clustering is performed, with the discrete case yielding higher TP rates.

All the signals measured were used to build output-only SIMO ARX models. The ROC curves of the individual models showed that they deliver different detections, while some models are more accurate at detecting damage and others are more accurate at detecting fatigue microcracks. No conclusions could be drawn with respect to which inputs (i.e., which locations or measurement directions) yield the highest performance. This finding, though, emphasizes the need for investigating all possible models simultaneously, since each model captures different effects.

Each dataset resulted in several TF poles, which were assessed. Hence, more than one detection was obtained for each dataset. Those decisions were combined (i) by obtaining the majority vote and (ii) by returning a positive detection if at least one of the TF poles was identified as damaged. The conclusion was reached that the decisions based on the majority vote yield lower AuC values than the ones of the more conservative case of obtaining a positive detection if at least one TF pole of a dataset deviates from the healthy state. This result is in agreement with the AuC values of the individual TF pole clusters, which show that the TF poles of a damaged dataset do not simultaneously exhibit variations due to structural changes.

9.2.2 Investigation with AdaBoost

It was shown that changes in the TP rate and the FP rate are more suitable metrics for the assessment of the boosting performance than the error rate. This lies in the fact that the notion of the error rate does not distinguish between false positive and false negative detections, while this information is significant for real-life SHM applications. The framework realizations which were combined using AdaBoost were evaluated for a series of confidence intervals, yielding a high number of available sets for investigation with AdaBoost. Relaltively few sets resulted in AdaBoost classifiers which improved the detection performance. However, when boosting was achieved, it entailed substantial improvement of both the TP and the FP rates.

A significant conclusion drawn from this investigation is that boosting perfomance is dependent on the relative position of the CP values on the CP space. This relative positioning can be decomposed to the overlaps of CP distributions corresponding to tp, tn, fp and fn detections. The Bhattacharyya coefficient (BC) was suggested for the quantification of those overlaps. The BC values of the aforementioned overlaps were summed up to build a BC sum. Realization pairs with high BC sums resulted in AdaBoost classifiers which outperformed the individual framework classifiers, and in classifiers which improved some rates but caused deterioration of other rates. On the other hand, lower values of BC sums resulted in poor performance which involved partial improvement of the detection accuracy or no boosting at all. While the BC served as a metric for the identification of the sets which were expected to yield good boosting results, the observed sum did not exactly represent the optimal cases. Nevertheless, it seems that the BC values can be used in advance to estimate whether a set is eligible for boosting.

CP values corresponding to specific clusters were omitted from the AdaBoost analysis in order to investigate whether the shortcoming of insufficient CP ovelaps could be compensated. In this manner, specific clusters were "switched off" and the CP overlaps were altered. The results of this investigation showed that that was an effective measure since many sets yielded AdaBoost classifiers which outperformed the individual framework classifiers.

In this work, AdaBoost was implemented for combining damage features obtained by vibration-based methods. However, the applicability of the suggested concept also extends to other SHM approaches which make use of different techniques and methodologies, such as acoustic emission or wave propagation.

9.2.3 Comparison of CP Sensitivity

Data from the modal test of a wind turbine rotor blade were used in order to investigate the sensitivity of several CPs to structural changes, such as added mass, which simulated ice accretion on the blade surface, and damage at the blade trailing edge. All investigated CPs were affected by structural changes. CP^{ARX}, the reference-based CPs deriving form the VAR model and the SSI model, as well as the damping ratios proved more sensitive to damage than the natural frequencies. However, the natural frequencies showed potential of quantification since most of the observed values decreased with the increasing amount of mass. Furthermore, it was shown that the natural frequencies are more sensitive to added mass, while the reference-based CPs are more sensitive to damage. As for CP^{ARX}, the sensitivity to added mass and damage varies for the different TF poles obtained from the model.

In order to investigate the sensitivity of the aforementioned CPs to varying EOCs, data from the fatigue test of a wind turbine rotor blade were analyzed using the SHM framework for several settings and for a series of confidence intervals. The results were presented in ROC curves, which provided information about the overall performance of the CPs. It was shown that CP^{ARX} and CP^{M} exhibit moderate sensitivity to varying EOCs, while CP^{R^2} is rather insensitive to such variations. CP^{ϵ}, on the other hand, exhibits high sensitivity to EOCs since the detection performance improves significantly with data clustering.

9.3 Future Research

9.3.1 Investigation of a Novel Condition Paremeter

Clustering the TF poles of the training set on the z-plane is the most challenging part of the implementation of CP^{ARX} for SHM. The selection of the appropriate model order offers better TF pole aggregation, which facilitates clustering. However, more complex EOC settings are expected to result in TF pole clusters which are not as distinct as those presented in the example in chapter 6. In this work, k-means clustering and affinity propagation were used. The disadvantage of k-means clustering is that it requires a predefined number of clusters, while affinity propagation does not always yield the desired outcome. Therefore, further research should be done in order to select a more appropriate clustering method.

CP^{ARX} was obtained by building SIMO output-only ARX models. Hence, one signal was used as input and the remaining channels were used as outputs, while the model expressed the transmissibility between those measures. In an extention of this work, CP^{ARX} can be built using MIMO ARX models, which represent more complex transmissibility measures.

The preliminary analysis regarding TF pole sensitivity, which was presented in chapter 5, included the superposition of the TF poles and the system poles obtained from the SSI and TEMP on the z-plane. This superposition revealed that, in terms of frequency, some of the TF poles are in fact the system poles, i.e., the frequency content of the TF poles equals the natural frequency. Future research should be carried out in order to explore the effect of structural changes on the relative positions of TF poles and on the system poles. As it was described in chapter 2, the system poles and system zeros correspond to the maxima and minima of the transfer function. Hence, investigating the relative changes between the TF pole positions and the pole positions can reveal infromation on the contributions of the mode shapes of the structure and on how these contributions are affected when structural changes occur.

9.3.2 Investigation with AdaBoost

AdaBoost used in conjunction with damage features delivering a decision regarding the structural state is applicable to any SHM method, provided that there are sufficient data for all damage features and that the damage features are obtained from the same feature space and have the same form. In the future, AdaBoost should be implemented to combine the decisions of damage features deriving from other SHM approaches.

Further work should explore the application of AdaBoost in operation. In the suggested concept of integrating AdaBoost with the SHM framework, the AdaBoost classifier is trained with *tp* and *tn* detections. Hence, information from the healthy structure and the damaged structure is required. However, this is not possible for an operating system which has not been subjected to damage or structural changes.

Future research should explore whether the integration of AdaBoost and the SHM framework can be trained for different damage scenarios using the generated data of an accurate structural model and then tested on a structure in operation. Moreover, it has to be examined whether the AdaBoost classifier which has been trained on a certain structure can accurately classify the CP values of another structure which is identical but has a different scale.

Furthermore, it should be investigated whether AdaBoost can be applied to an operating wind turbine even without the support of a structural model. For instance, when monitoring an operating, intact wind turbine, only datasets from the healthy structure are available. These datasets result in either tn or fp detections. After a certain time period of monitoring, information is gathered about CP values which correspond to tn or fp detections. These may be used in the AdaBoost algorithm to build a classifier which distinguishes between tn and fp values. The relative positioning of the CP values on the CP space can be further exploited, so that fp values are transformed into tn values. However, since this modified version only uses healthy datasets, the resulting classifier will be unable to detect damaged datasets and will be restricted to reducing the fp detections.

Machine learning approaches offer very powerful tools for describing underlying mechanisms and for making predictions based on data. The coupling of numerical or finite element models with the analysis of measured structural responses can be used in order to support the implementation of supervised machine learning approaches in operation. Damaged datasets are usually not available for real structures, limiting the application of supervised machine learning approaches in operation. Thus, the simulation of various damage scenarios in accurate models or in the so-called digital twin of the structure can be exploited in order to evaluate the state of the operating structure based on the combination of response data and model data.

Appendix A

Sensitivity Ananlysis of TF Poles

A.1 LANL Three-Story Model

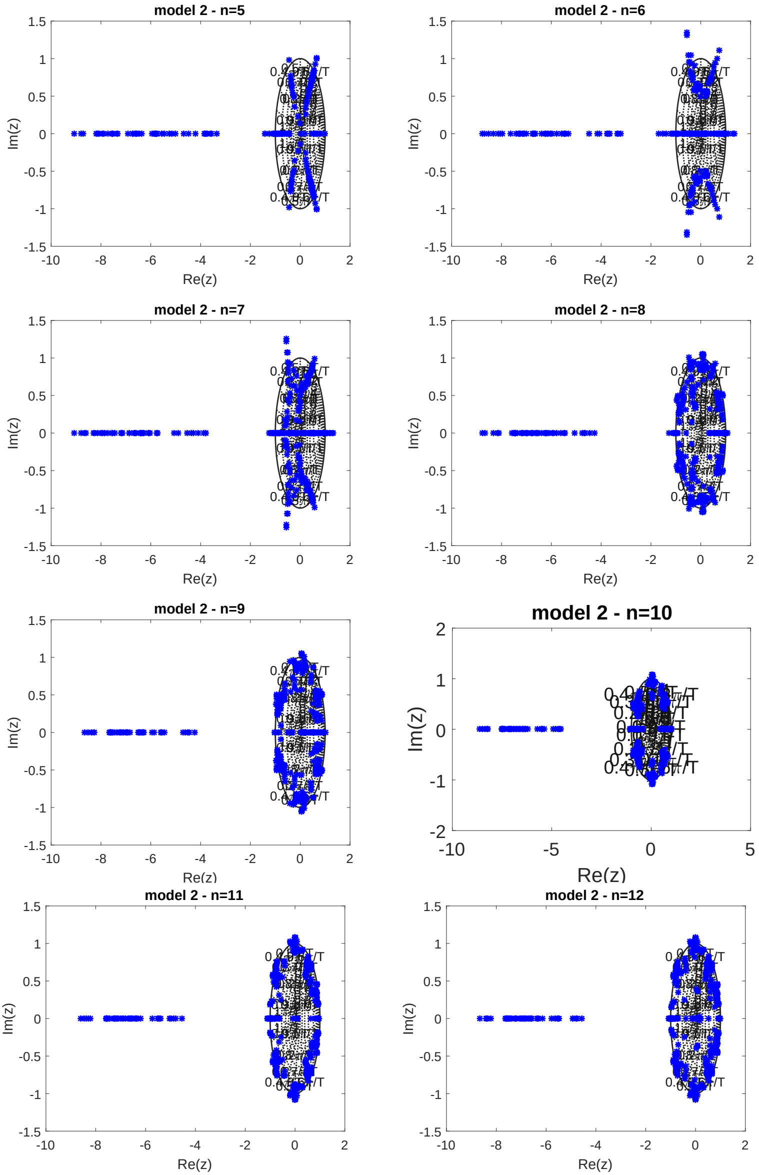


Figure A.1: TF poles of model 2 at the baseline state for the potentially optimal model orders of the output-only ARX model (6 to 12 according to Figure 5.2(b)). Model order 10 provides the highest level of pole aggregation and is thus selected as the optimal model order.

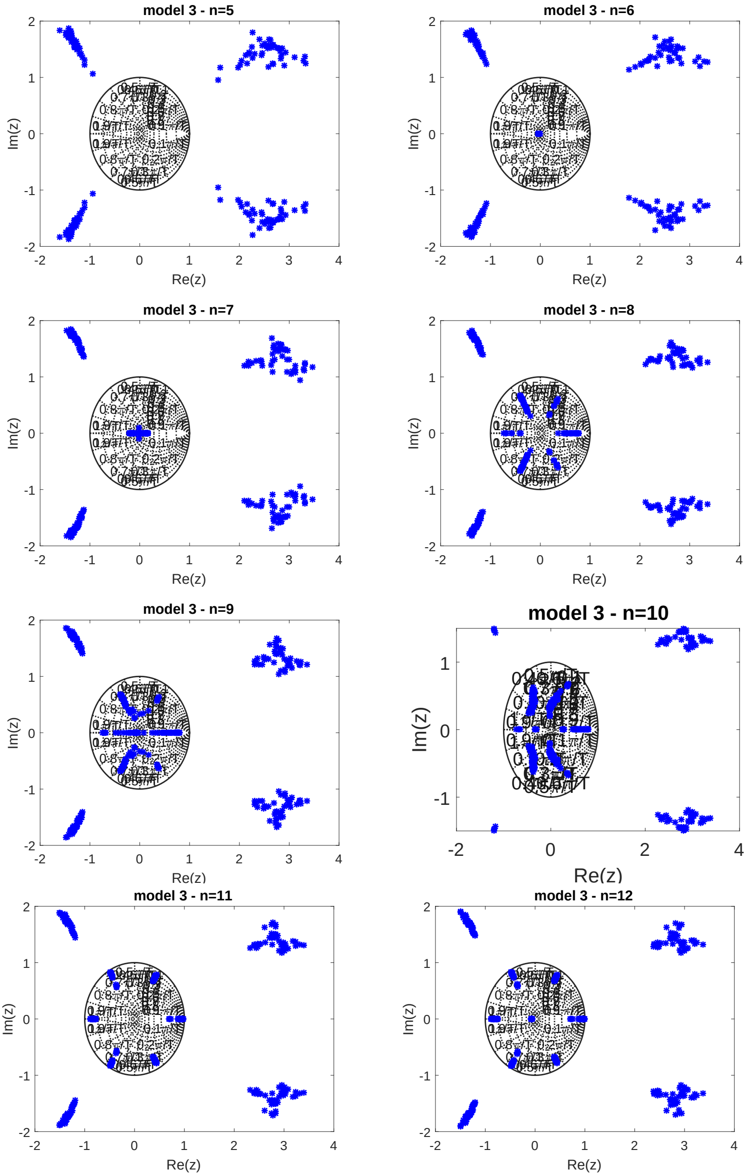


Figure A.2: TF poles of model 3 at the baseline state for the potentially optimal model orders of the output-only ARX model (6 to 12 according to Figure 5.2(c)). Model order 11 provides the highest level of pole aggregation and is thus selected as the optimal model order.

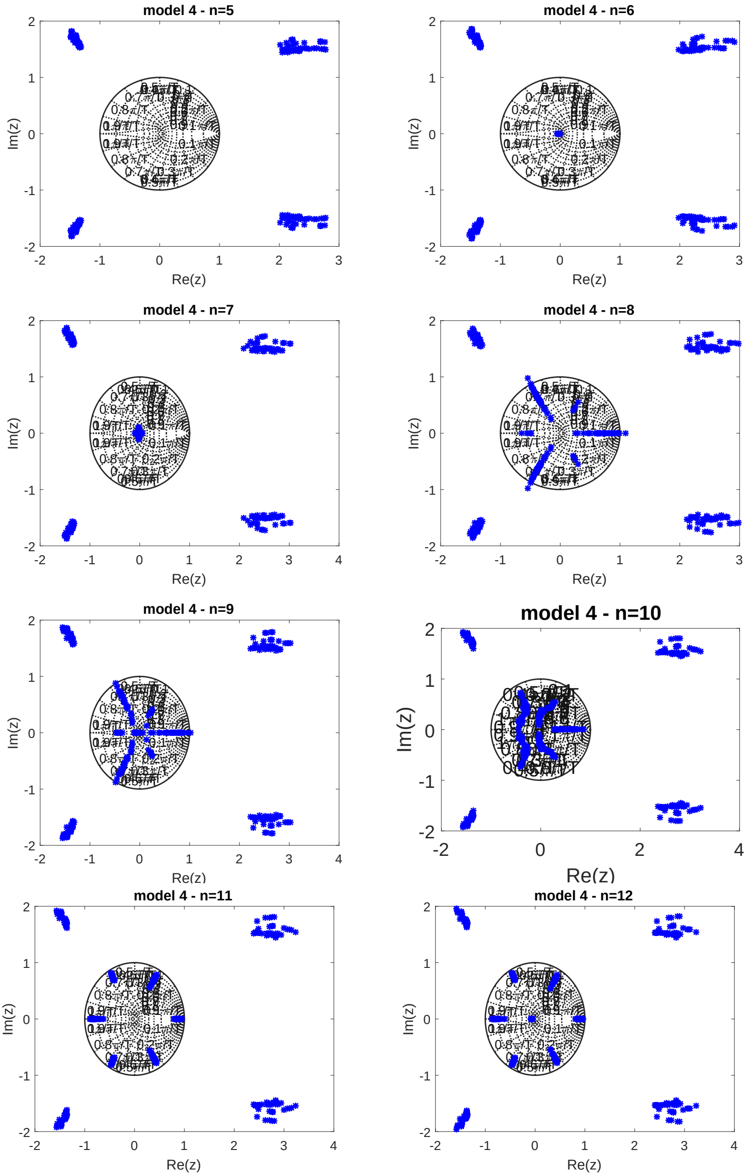


Figure A.3: TF poles of model 4 at the baseline state for the potentially optimal model orders of the output-only ARX model (6 to 12 according to Figure 5.2(d)). Model order 11 provides the highest level of pole aggregation and is thus selected as the optimal model order.

A.2 Rotor Blade Modal Test

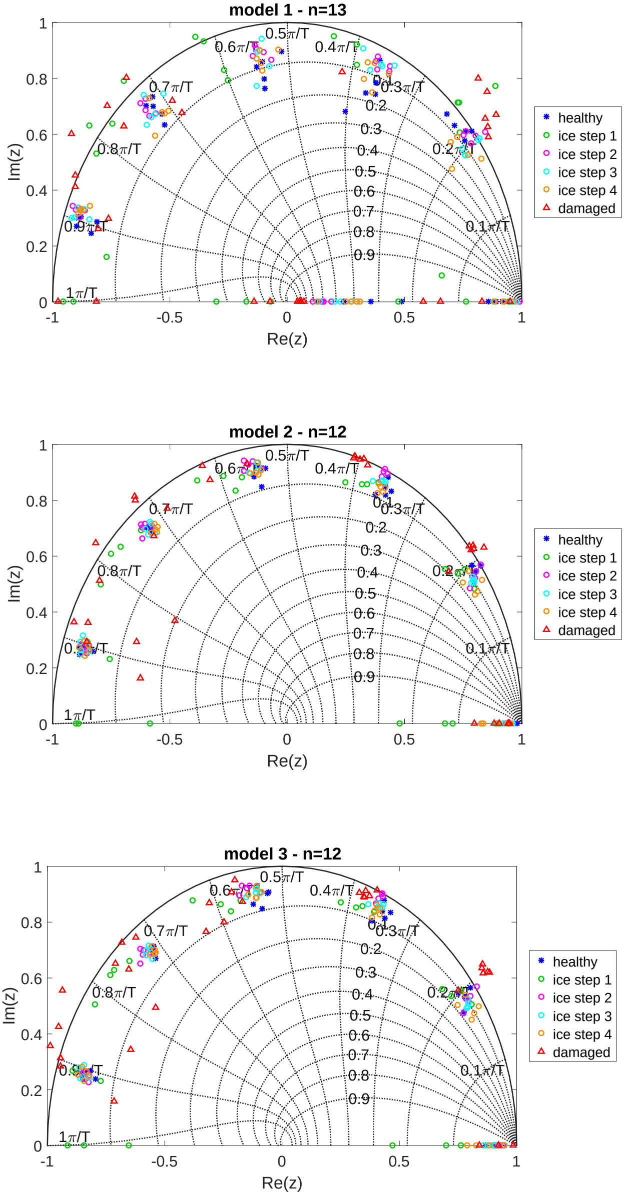


Figure A.4: TF poles of models 1 to 3 for the healthy state, the four iced states and the damaged state. Poles presented for the optimal model orders shown on top of the plots.

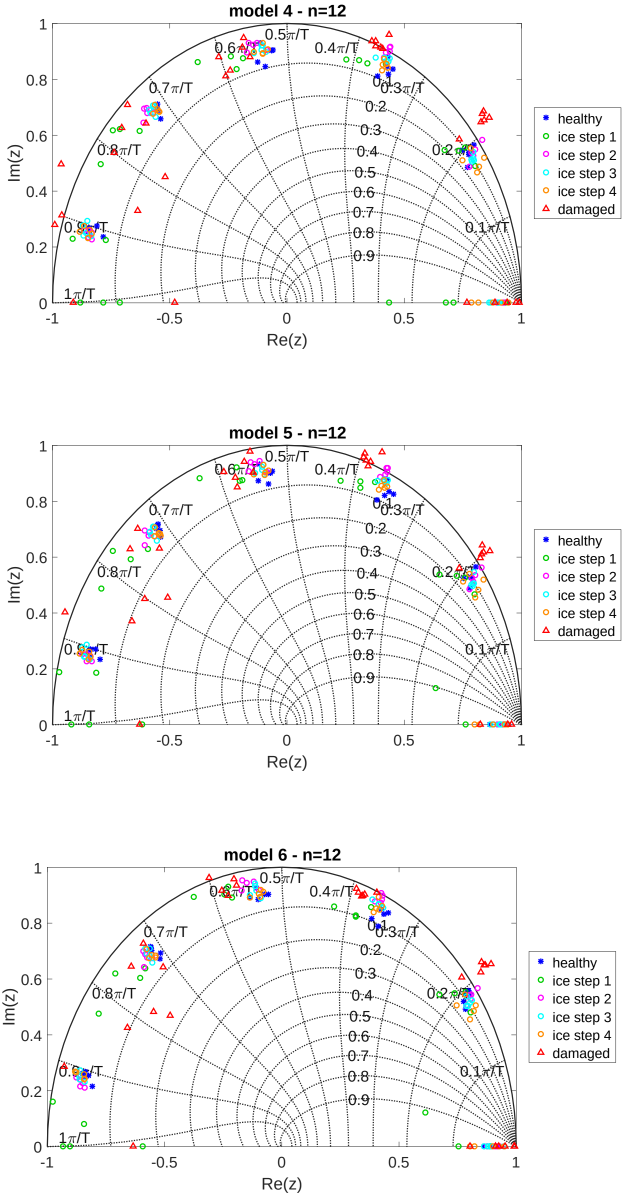


Figure A.5: TF poles of models 4 to 6 for the healthy state, the four iced states and the damaged state. Poles presented for the optimal model orders shown on top of the plots.

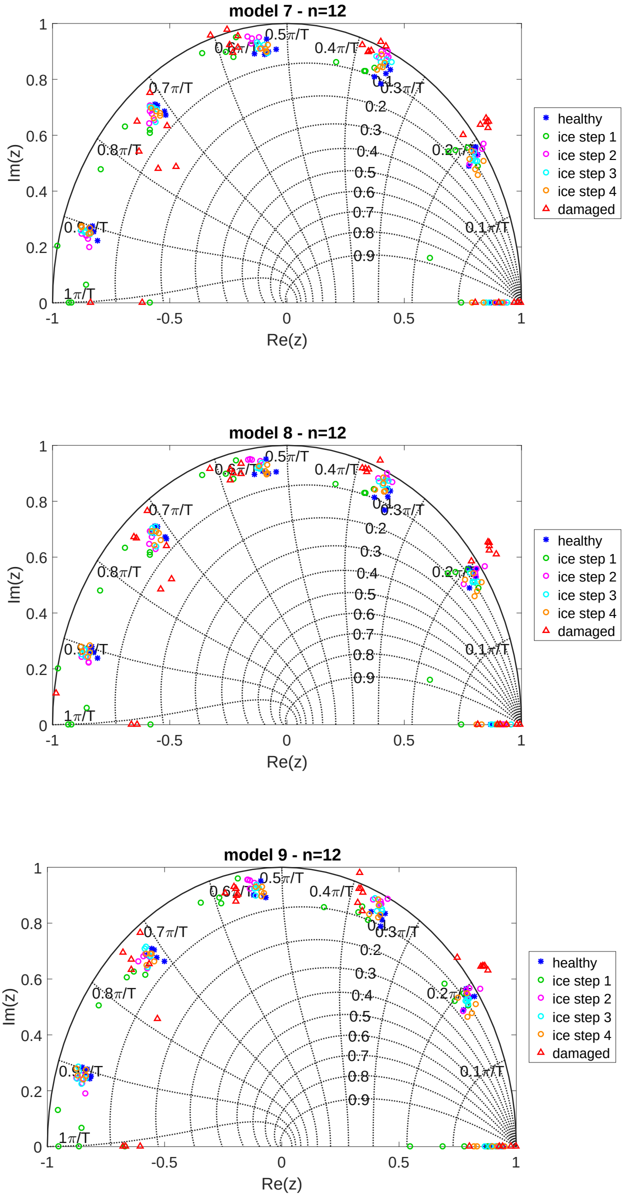


Figure A.6: TF poles of models 7 to 9 for the healthy state, the four iced states and the damaged state. Poles presented for the optimal model orders shown on top of the plots.

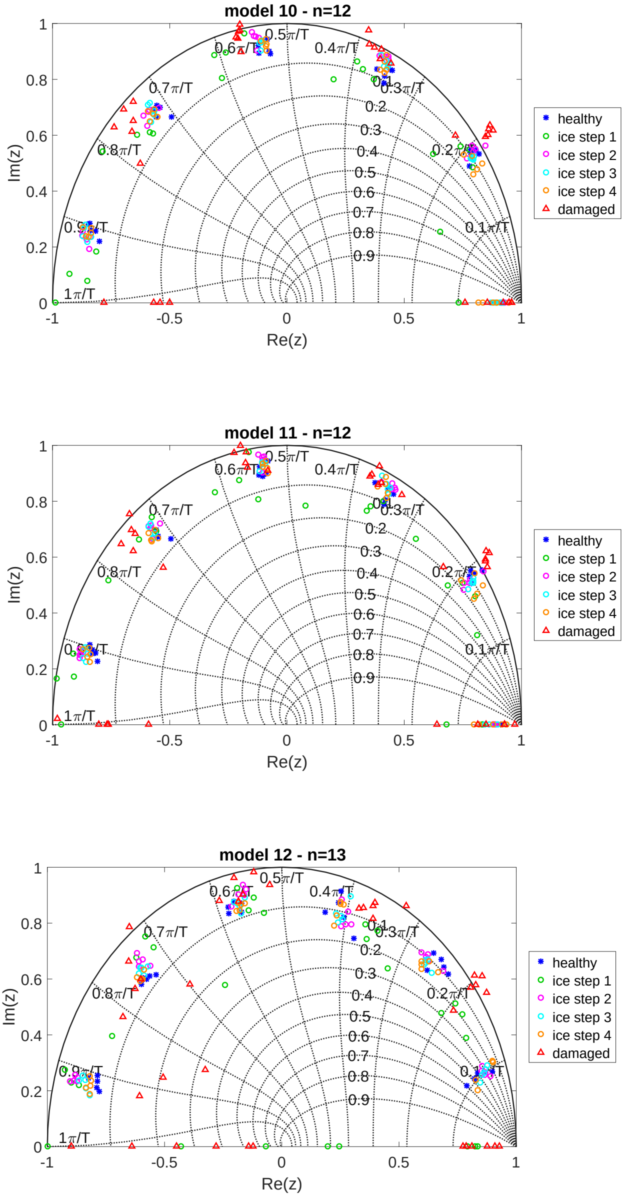


Figure A.7: TF poles of models 10 to 12 for the healthy state, the four iced states and the damaged state. Poles presented for the optimal model orders shown on top of the plots.

A.3 Rotor Blade Fatigue Test

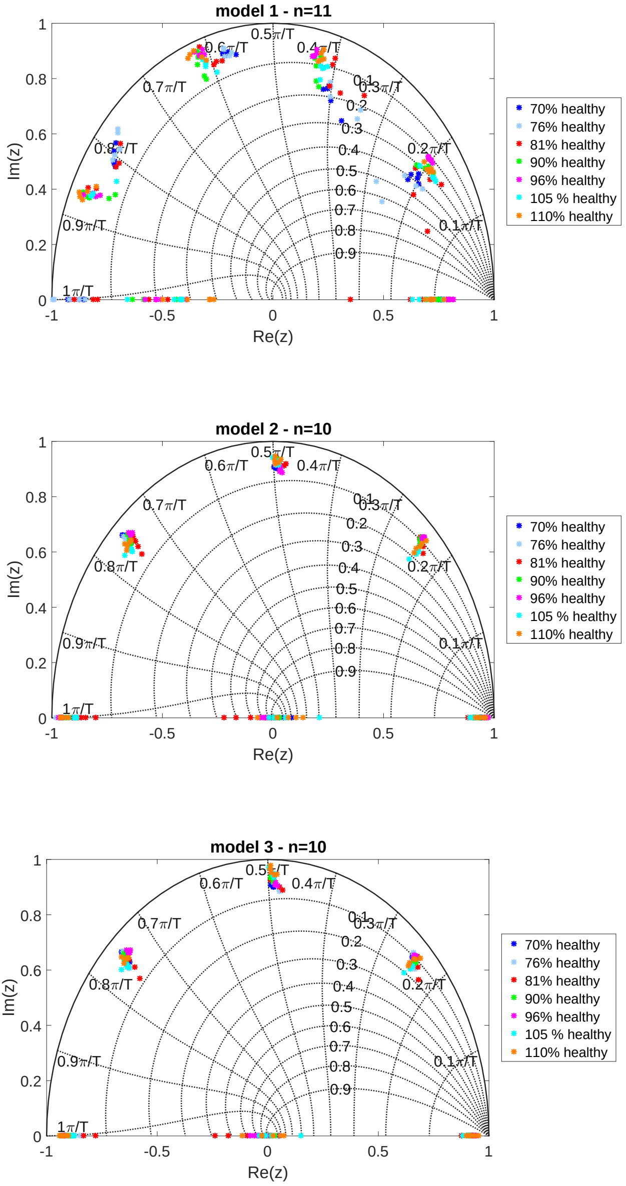


Figure A.8: TF poles of models 1 to 3 at their optimal model order for the rotor blade fatigue test. Results shown for the baseline state and for different load levels of the healthy state.

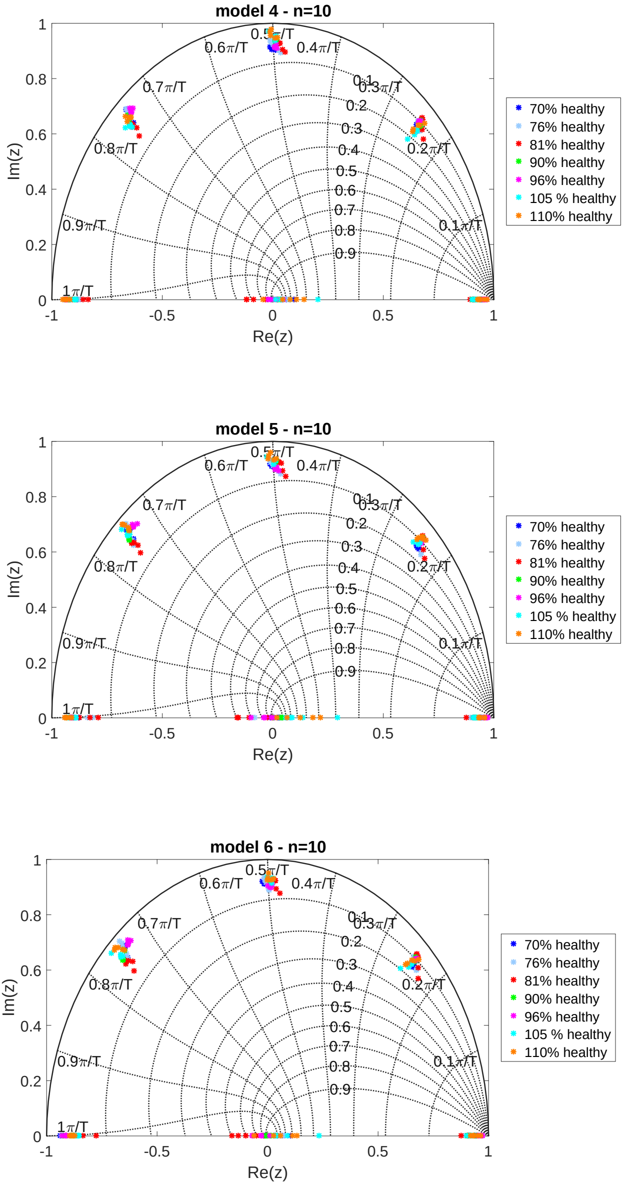


Figure A.9: TF poles of models 4 to 6 at their optimal model order for the rotor blade fatigue test. Results shown for the baseline state and for different load levels of the healthy state.

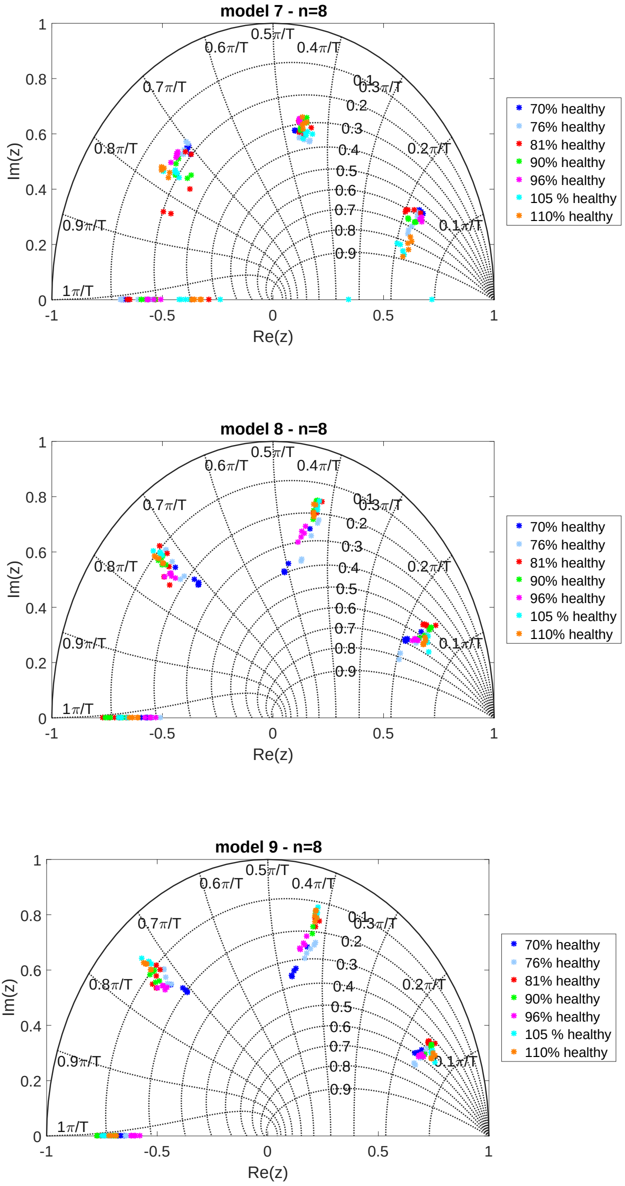


Figure A.10: TF poles of models 7 to 9 at their optimal model order for the rotor blade fatigue test. Results shown for the baseline state and for different load levels of the healthy state.

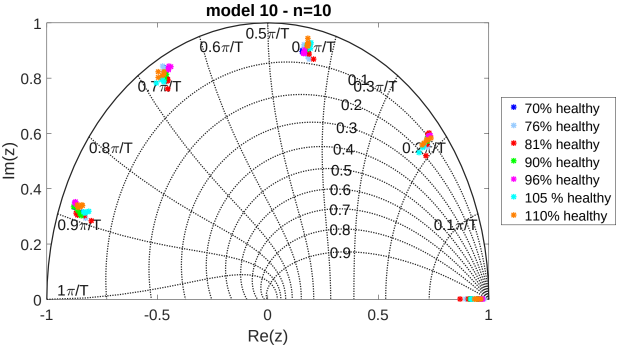


Figure A.11: TF poles of model 10 at its optimal model order for the rotor blade fatigue test. Results shown for the baseline state and for different load levels of the healthy state.

Appendix B

Validation of CP^{ARX}

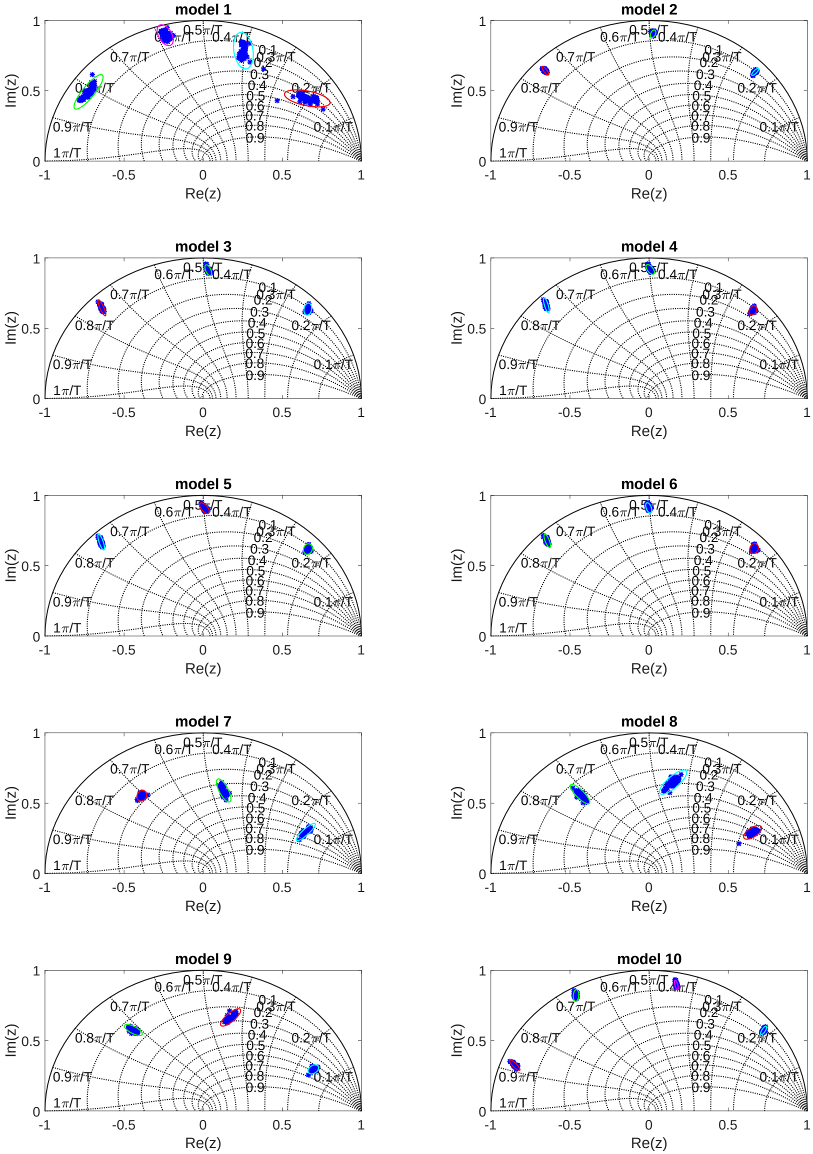


Figure B.1: TF poles of models 1 to 10 for ML cluster 2 of Manual 2. The TF poles are clustered on the z-plane using k-means clustering.

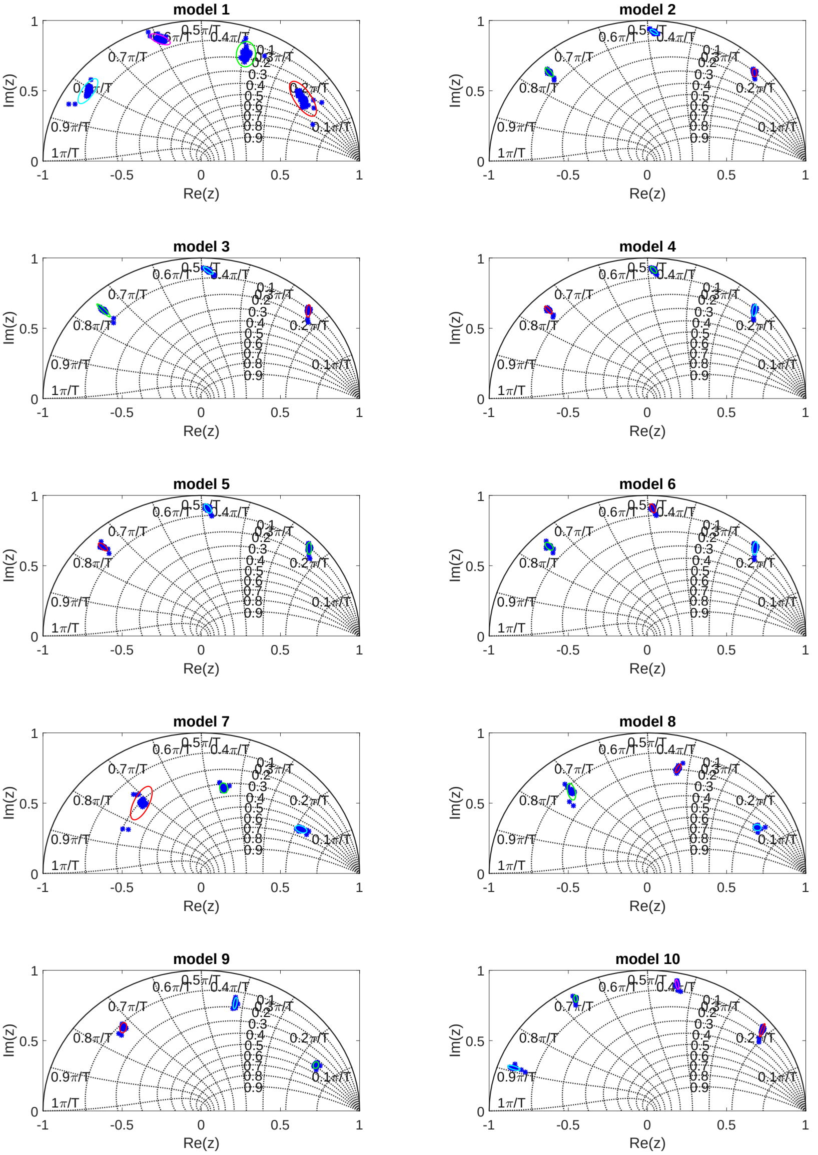


Figure B.2: TF poles of models 1 to 10 for ML cluster 3 of Manual 2. The TF poles are clustered on the z-plane using k-means clustering.

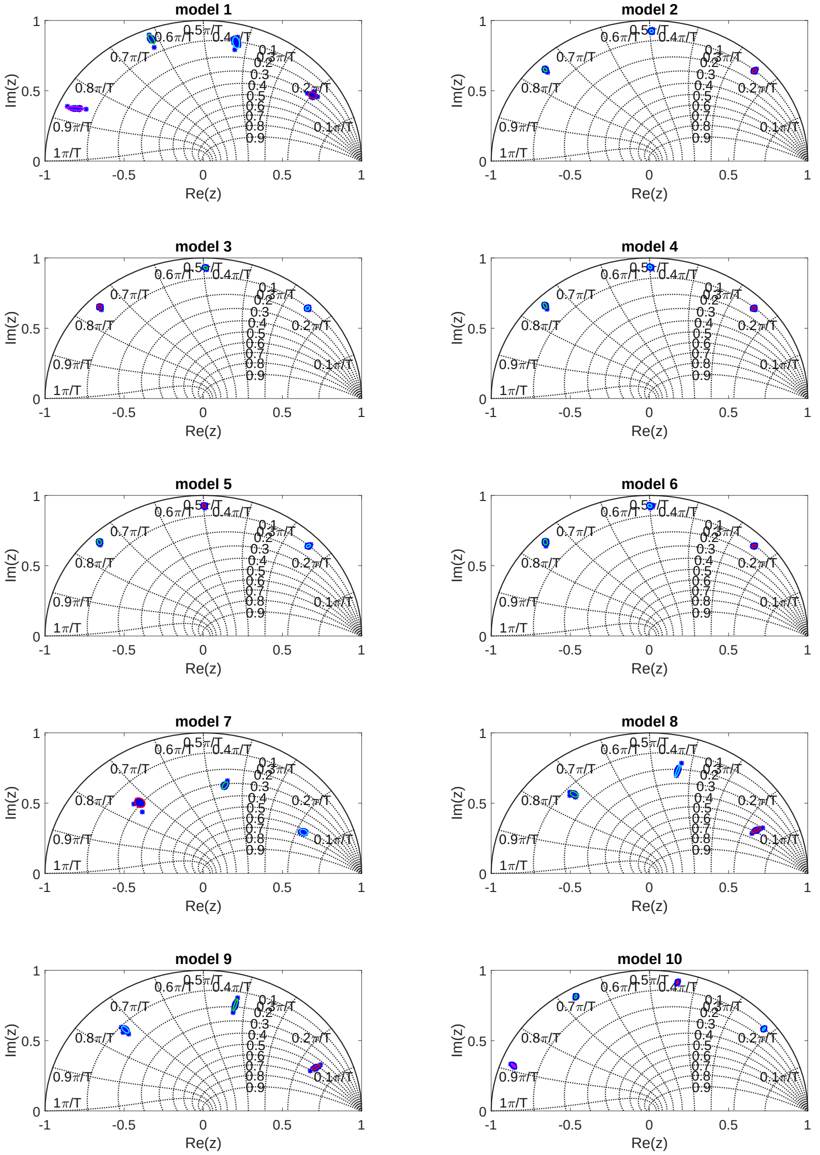


Figure B.3: TF poles of models 1 to 10 for ML cluster 4 of Manual 2. The TF poles are clustered on the z-plane using k-means clustering.

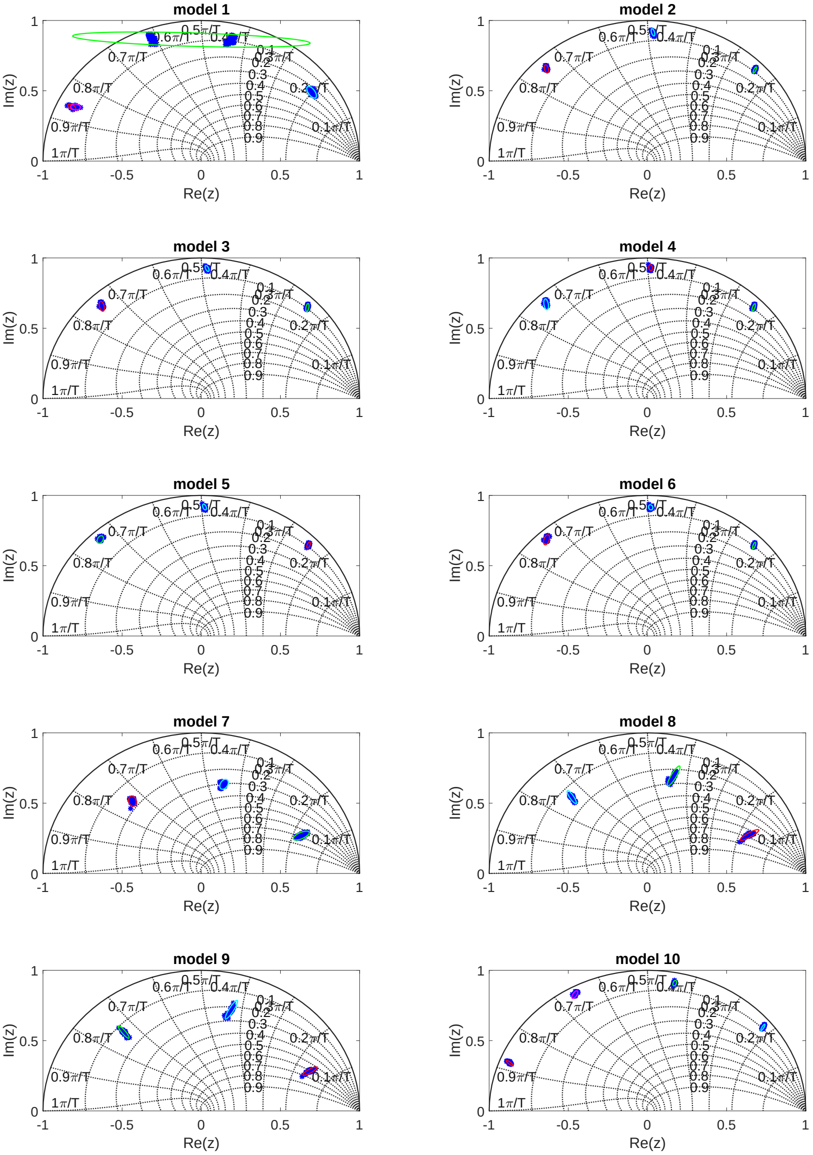


Figure B.4: TF poles of models 1 to 10 for ML cluster 5 of Manual 2. The TF poles are clustered on the z-plane using k-means clustering.

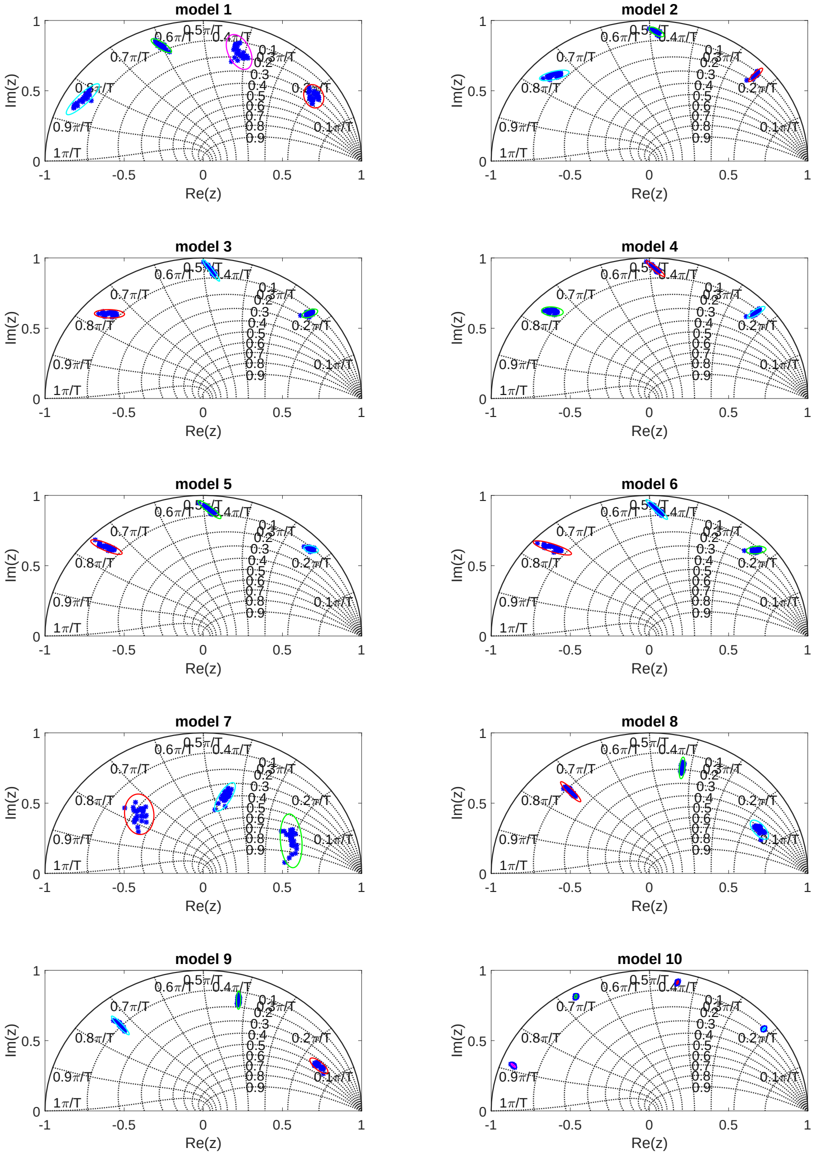


Figure B.5: TF poles of models 1 to 10 for ML cluster 6 of Manual 2. The TF poles are clustered on the z-plane using k-means clustering.

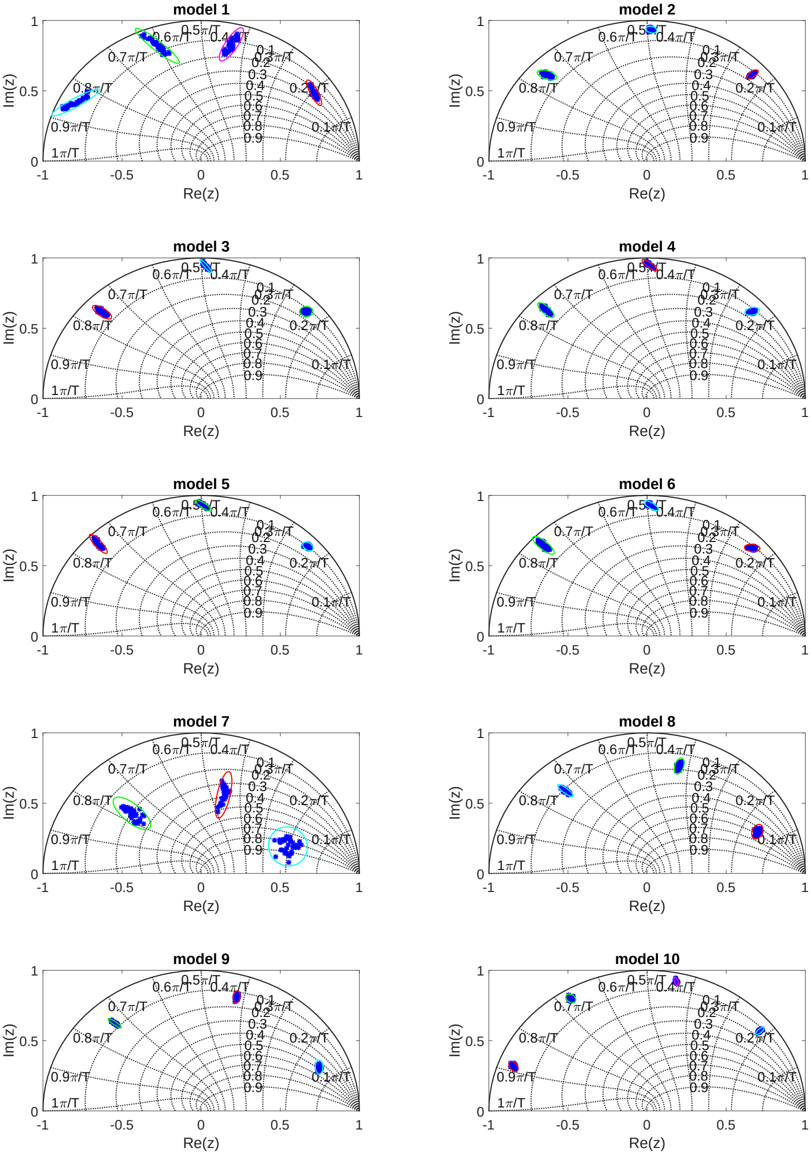


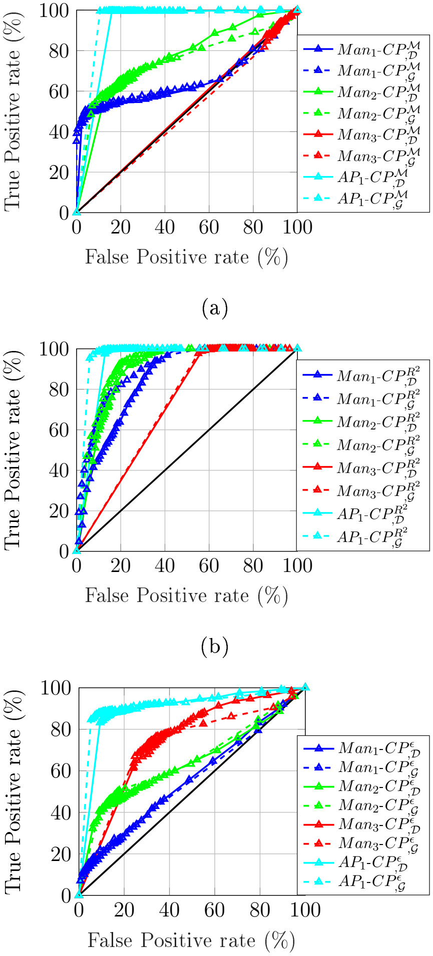
Figure B.6: TF poles of models 1 to 10 for ML cluster 7 of Manual 2. The TF poles are clustered on the z-plane using k-means clustering.

Appendix C

Validation of AdaBoost

C.1 Simulated Data of the LANL 3 kW Wind Turbine

Figure C.1. The solid curves correspond to decision boundaries based on percentiles, while the dashed lines to defining the decision boundaries for the assumption of Gaussian distribution. In all figures, the discrepancies between the two settings of the hypothesis testing tier are quite small. CP^M has a very high AuC value and 100% true positive detections for AP1 (see Figure C.1(a)). The second best results are obtained for Man2 and then for Man1, while Man3 exhibits very poor performance. In Figure C.1(b), CP^{R^2} shows very high true positive detections for all the ROC points of Man3 and AP1, and relatively high AuC for all the clustering cases except Man3. The performance of CP^{ϵ}, on the other hand, is positively affected by the increasing number of clusters, since the AuC is higher for clustering cases with more clusters (see Figure C.1(c)).



(a)

(b)

(c)

Figure C.1: ROC curves of CP^M (a) CP^{R^2} (b) and CP^{ϵ} (c) from the analysis of the simulated data using the SHM framework. Solid curves for discrete distribution and dashed curves for Gaussian distribution. Clustering cases indicated by different colors.

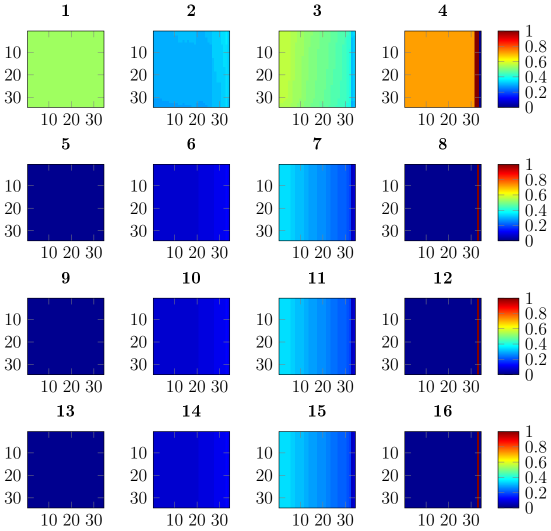


Figure C.2: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_1-R_2 of simulated data.

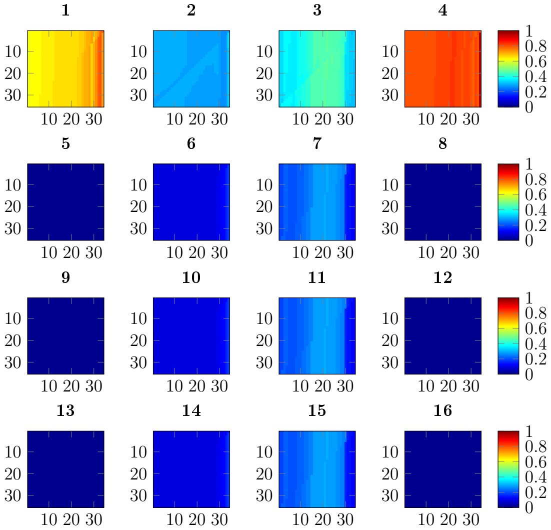


Figure C.3: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_1-R_3 of simulated data.

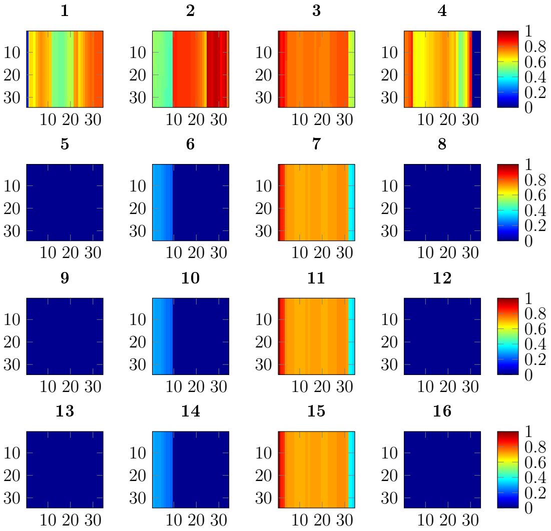


Figure C.4: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_1-R_4 of simulated data.

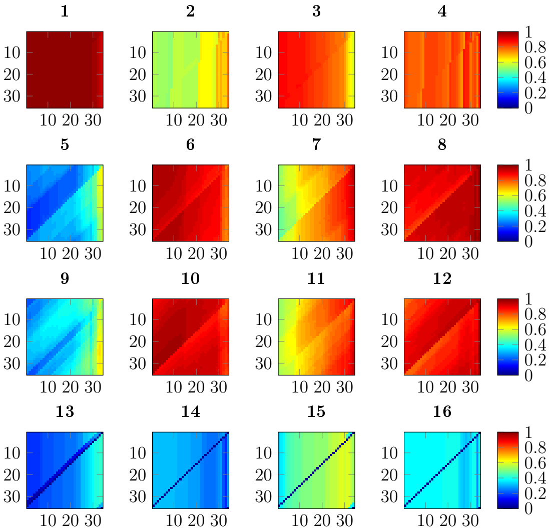


Figure C.5: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_2-R_3 of simulated data.

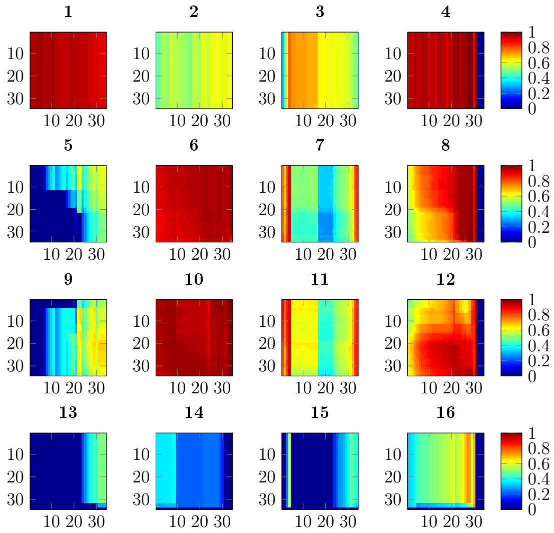


Figure C.6: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_2-R_4 of simulated data.

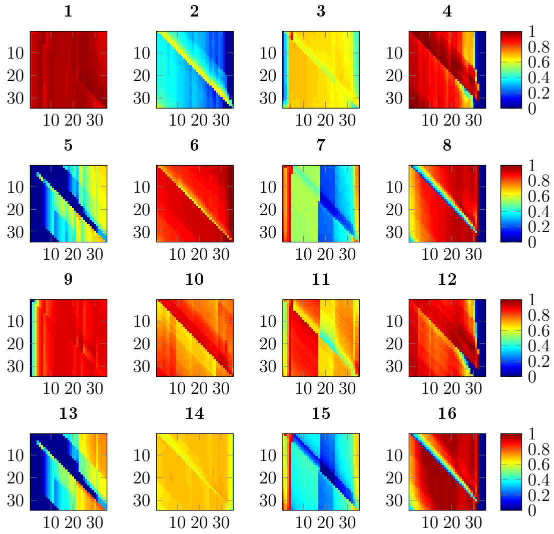
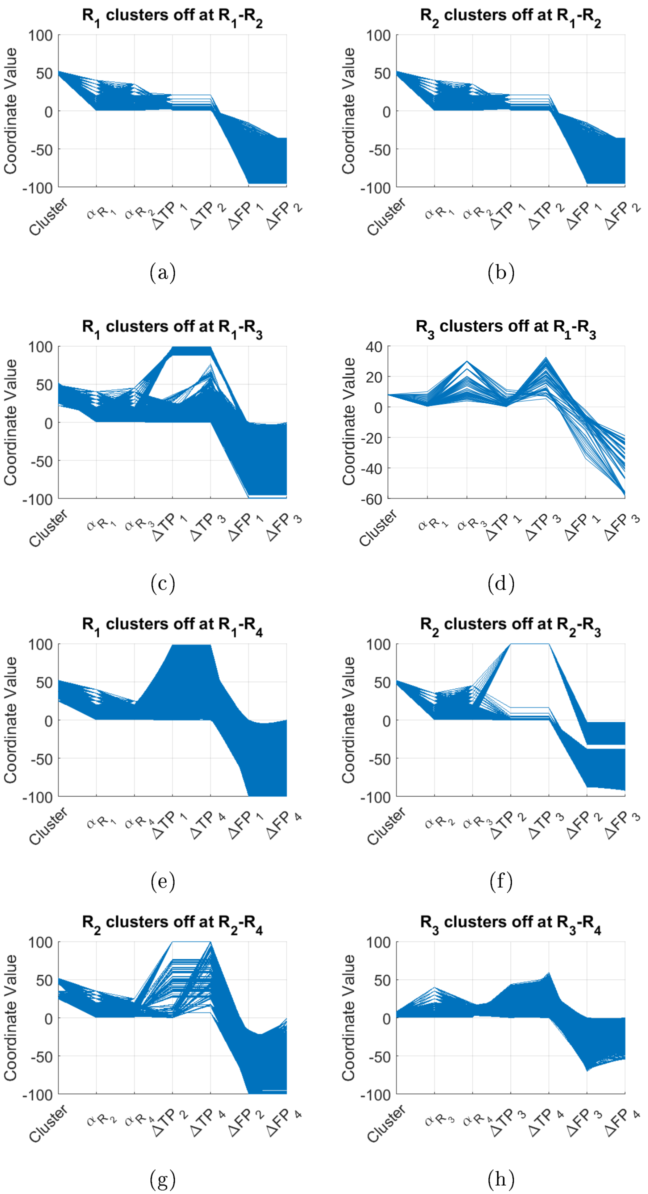


Figure C.7: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_3-R_4 of simulated data.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure C.8: Parallel coordinate plots with sets of: cluster number, corresponding a values and changes of TP and FP rates for the sets that result in boosting case 1 ($\Delta TP_i > 0$ and $\Delta FP_i < 0$) when switching off clusters of the synthetic data.

The first coordinate in the parallel coordinate plots in Figure C.8 shows for which cluster numbers boosting case 1 is achieved. The remaining coordinates indicate the corresponding values of level of significance and the resulting differences in TP and FP rates with respect to the two realizations. Figure C.8(a) summarizes the results for switching off clusters of R_1 in pair R_1-R_2. Boosting case 1 occurs when switching off clusters 47 to 52. These sets have different a values (a_{R_1} and a_{R_2}). This outcome is in accordance with the results presented earlier, which showed that the boosting cases are not dependent on the level of significance. The coordinates ΔTP_1 and ΔTP_2 range between 0 and 25 percentage points. On the other hand, ΔFP_1 and ΔFP_2 involve greate changes, which range from 20 to approximately 90 percentage points.

Likewise, in Figures C.8(b) and C.8(f), boosting case 1 is obtained when switching off clusters 47 to 52. In Figure C.8(c) and C.8(e), boosting case 1 is achieved when switching off clusters 22 and 25 to 52, while in Figure C.8(g) when switching off clusters 25 and 30 to 52. In all aforementioned figures, the clusters result from clustering with AP1. The cluster numbers of AP1 and their location on the rotor speed - yaw angle space are shown in Figure C.9. These clusters correspond to rotor speeds which range from 170 to 350 rpm and yaw angles which range from -65 to 70 degrees. In Figure C.8(d), on the other hand, only omitting the CP values of cluster 8 results in boosting case 1, while in Figure C.8(h) this happens for clusters 1 to 8. The clusters of these two figures occur from Man2.

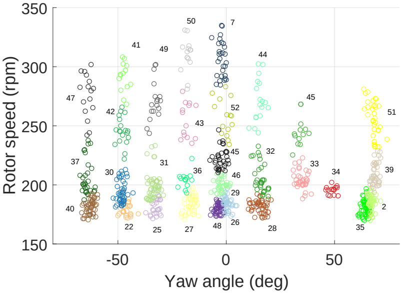
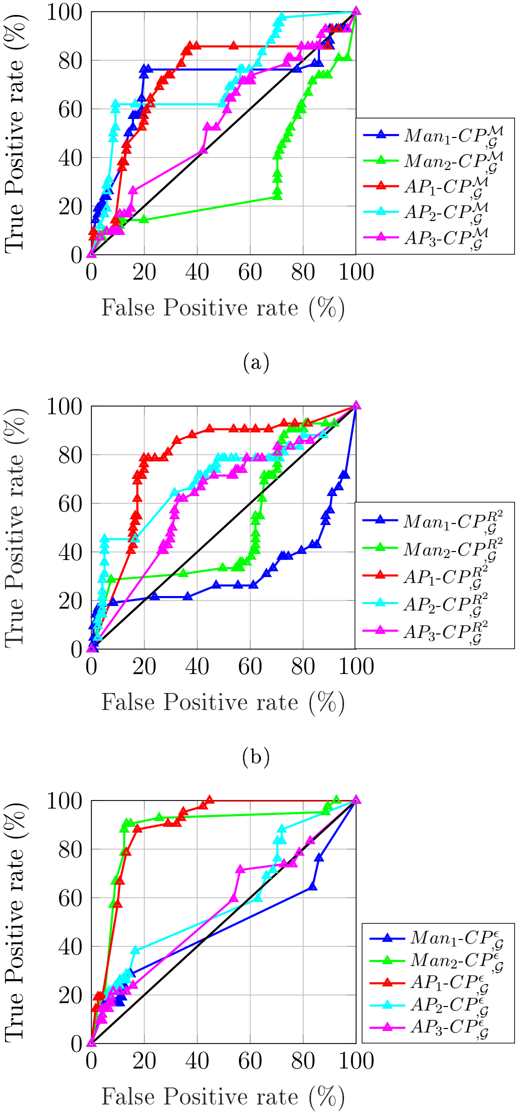


Figure C.9: Numbered clusters of AP1 on the rotor speed - yaw angle space for the simulated data.

The ROCs of the framework realizations are shown in Figure C.10. The comparison of the figures reveals that the three CPs deliver different detections for the same database. Furthermore, there is a prevalent effect of data clustering on the AuC, which reflects the overall detection performance. For instance, in the ROCs of CP^M in Figure C.10(a), clustering cases Man1, AP1 and AP2 exhibit a relatively good detection performance, AP3 has a mediocre detection performance, with the AuC value sightly higher than 0.5, and Man2 delivers poor detection performance, with the AuC value smaller than 0.5. For CP^{R^2}, on the other hand, AP1 yields the best detection results, being followed by AP2 and AP3, while Man1 and Man2 exhibit very poor performance. Finally, for CP^ϵ, Man2 and AP1 have high AuC values, while the ROCs of the remaining clustering cases are very close to the $x = y$ line.

The ROCs of the experimental data are not comparable to the ROCs of the simulated data, because different quantities were used in data clustering. The synthetic dataset was clustered according to wind speed and rotor speed data. However, the rotor speed was not available in the experimental setup. Measurements of wind speed, wind direction and temperature were used instead. Nevertheless, there are some consistencies between the ROCs of the synthetic and the experimental datasets. In both cases, the highest AuC is achieved for clustering with AP for all CPs. Moreover, clustering is always beneficial for CP^ϵ since Man1 has the lowest AuC value for both the synthetic and experimental datasets.

C.2 Operational Data of the LANL Wind Turbine



(a)

(b)

(c)

Figure C.10: ROC curves of CP^M (a), CP^{R^2} (b) and CP^{ϵ} (c) for the Gaussian distribution from the analysis of the experimental data within the SHM framework. Clustering cases indicated by different colors.

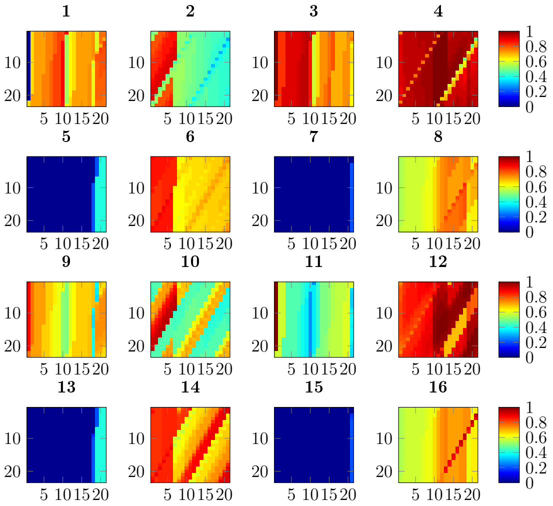


Figure C.11: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_1-R_2 for the LANL wind turbine.

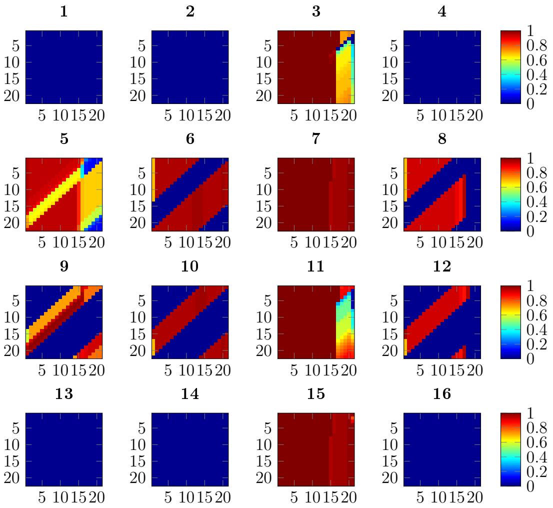


Figure C.12: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_1-R_3 for the LANL wind turbine.

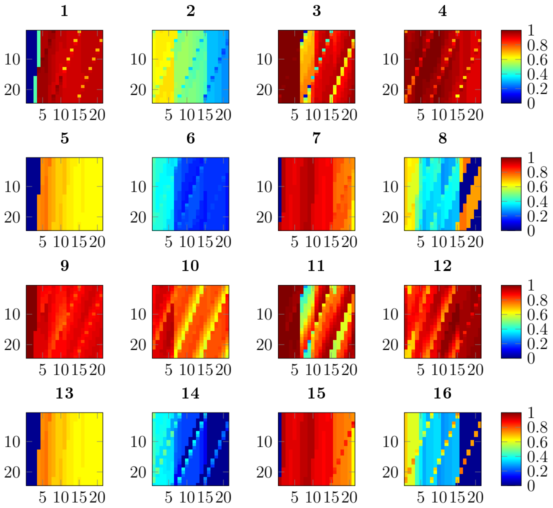


Figure C.13: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_1-R_4 for the LANL wind turbine.

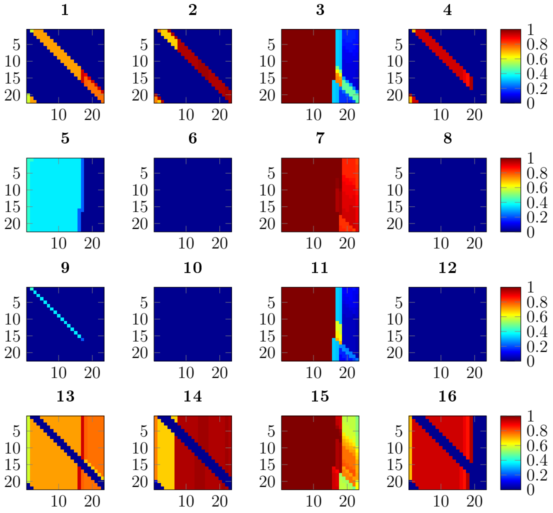


Figure C.14: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_2-R_3 for the LANL wind turbine.

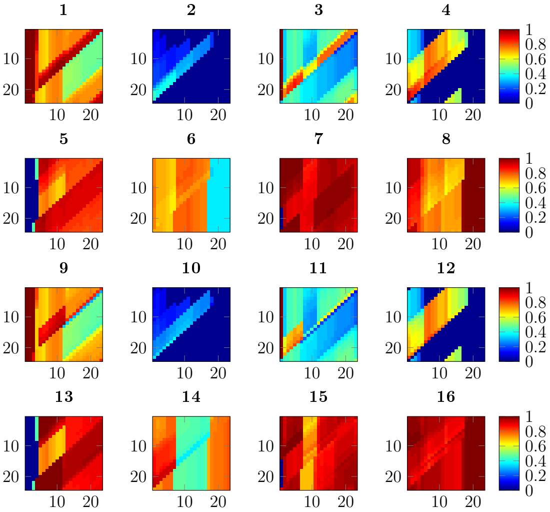


Figure C.15: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_2-R_4 for the LANL wind turbine.

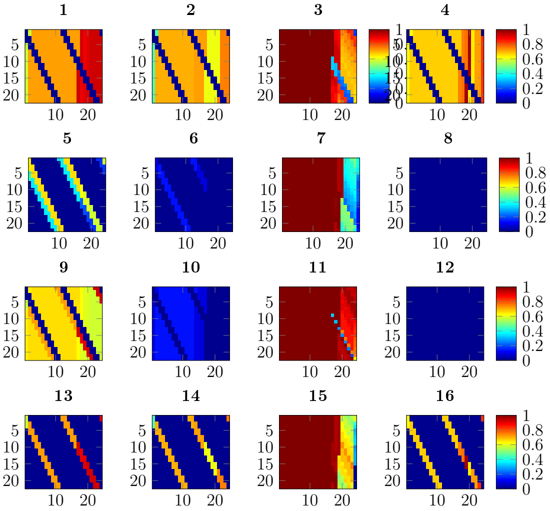


Figure C.16: Bhattacharyya coefficients of the overlaps between the CP values corresponding to different HT results (see Table 4.1). Realization pair R_3-R_4 for the LANL wind turbine.

The parallel coordinate plots in Figure C.17 show for which cluster numbers and a values boosting case 1 is achieved. In addition, the last four coordinates give the differences in TP and FP rates. For instance, in Figure C.17(a), the results are presented for switching off R_1 at R_1-R_3. Boosting case 1 is obtained when switching off clusters 1 and 2. Some values of ΔTP_2 are higher than those of ΔTP_1 by up to 30 percentage points. Thus, the AdaBoost classifier has a stronger effect on R_2 than on R_1. The same holds for the FP rates which are improved by up to approximately 10 percentage points. Also in Figures C.17(b), C.17(d), C.17(g) and C.17(h), switching off clusters 1 and 2 results in better performances. These two clusters are represented by the blue and red points of AP1 in Figure 7.8 and correspond to wind speeds equal to 1-3 m/s and temperatures equal to 12-19 °C. In Figures C.17(c), C.17(f) and C.17(i), boosting case 1 is obtained when switching off cluster 1 of Man2. This cluster is represented in Figure 7.8 by the blue points of Man2 and corresponds to wind speeds equal to 1-5 m/s and temperatures equal to 12-16°C. Hence, also for the experimental data, switching off clusters of specific EOCs and then employing AdaBoost results in better detection performance than that of the SHM framework. Furthermore, it is shown that switching off clusters may help exploit realization pairs that do not deliver satisfactory results when the entire sets are taken into account for boosting.

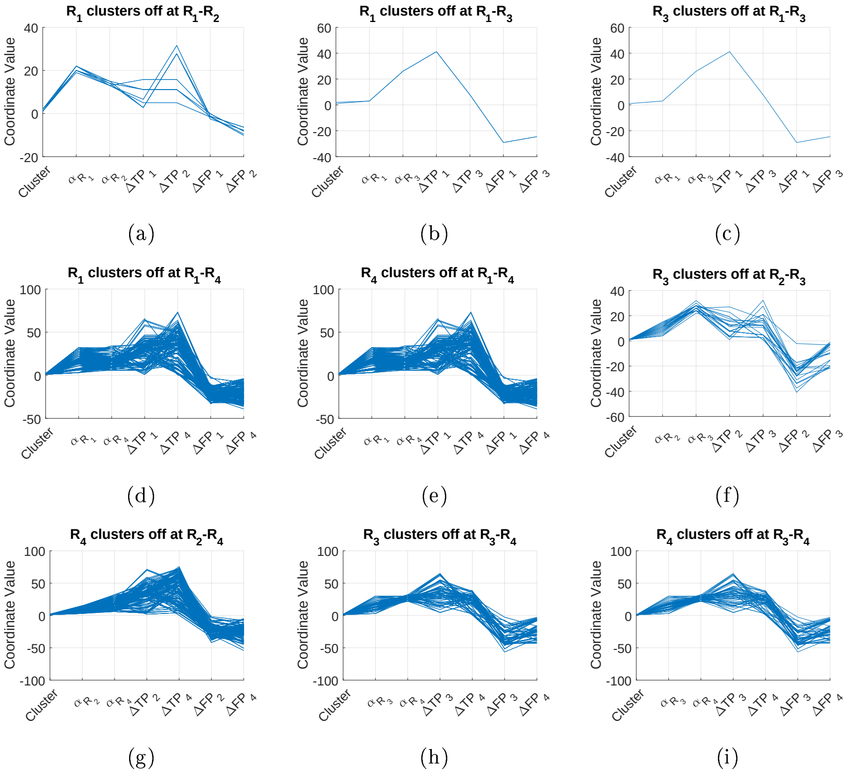


Figure C.17: Parallel coordinate plots with sets of: cluster number, corresponding a values and changes in TP and FP rates for the combinations which result in boosting case 1 ($\Delta TP_i > 0$ and $\Delta FP_i < 0$) when switching off clusters of the experimental data.

Bibliography

[1] C.R. Farrar and K. Worden. “An introduction to structural health monitoring”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365.1851 (2007), pp. 303–315.

[2] C.P. Fritzen, M. Klinkov, and P. Kraemer. “Vibration-Based Damage Diagnosis and Monitoring of External Loads”. In: *New Trends in Structural Health Monitoring*. Ed. by Friedrich Pfeiffer, Franz G. Rammerstorfer, Jean Salençon, Bernhard Schrefler, Paolo Serafini, Wieslaw Ostachowicz, and J. Alfredo Güemes. Vol. 542. Vienna: Springer Vienna, 2013, pp. 149–208.

[3] D. Zarouchas, A. Antoniou, F. Sayer, D. Van Hemelrijck, and A. van Wingerde. “Structural Integrity Assessment of Blade’s Subcomponents using Acoustic Emission Monitoring.” In: *Experimental and Applied Mechanics, Volume 6*. Ed. by Tom Proulx. Springer New York, 2011, pp. 511–518.

[4] L.P. Pahlavan. “Wave propagatin in thin walled composite structures: application to structural health monitoring”. PhD thesis. Delft, the Netherlands: TU Delft, 2012.

[5] S. Shafiee and E. Topal. “When will fossil fuel reserves be diminished?” In: *Energy Policy* 37.1 (2009), pp. 181–189.

[6] M.L. Wang, J.P. Lynch, and H. Sohn, eds. *Sensor technologies for civil infrastructures*. Woodhead Publishing series in electronic and optical materials number 55, 56. Elsevier, Woodhead Publishing, 2014.

[7] J. Tautz-Weinert and S.J. Watson. “Using SCADA data for wind turbine condition monitoring – a review”. In: *IET Renewable Power Generation* 11.4 (2017), pp. 382–394.

[8] A.K. Chopra. *Dynamics of structures: theory and applications to earthquake engineering*. 4th ed. Prentice Hall, 2012.

[9] D.J. Ewins. *Modal testing: theory, practice, and application*. 2nd ed. Mechanical engineering research studies 10. Research Studies Press, 2000.

[10] C. Rainieri and G. Fabbrocino. *Operational Modal Analysis of Civil Engineering Structures*. Springer New York, 2014.

[11] S.D. Fassois and F.P. Kopsaftopoulos. “Statistical Time Series Methods for Vibration Based Structural Health Monitoring”. In: *New Trends in Structural Health Monitoring*. Ed. by Friedrich Pfeiffer, Franz G. Rammerstorfer, Jean Salençon, Bernhard Schrefler, Paolo Serafini, Wieslaw Ostachowicz, and J. Alfredo Güemes. Vol. 542. Springer Vienna, 2013, pp. 209–264.

[12] K. Worden, C.R. Farrar, G. Manson, and G. Park. “The fundamental axioms of structural health monitoring”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463.2082 (2007), pp. 1639–1664.

[13] A. Rytter. “Vibration based inspection of civil engineering structures”. PhD thesis. Aalbork, Denmark: Aalborg University, Dept. of Building Technology and Structural Engineering, 1993.

[14] R. Farrar C and H. Sohn. “Pattern recognition for structural health monitoring”. In: *Workshop on Mitigation of Earthquake Disaster by Advanced Technologies*. Las Vegas, NV, USA, Dec. 2000.

[15] M. Bedworth and J. O'Brien. "The Omnibus model: a new model of data fusion?" In: *IEEE Aerospace and Electronic Systems Magazine* 15.4 (2000), pp. 30–36.

[16] M.W. Häckell. "A Hollistic Evaluation Concept for Long-Term Structural Health Monitoring". PhD thesis. Hannover, Germany: Institute of Structural Analysis, Leibniz Universität Hannover, 2015.

[17] M.W. Häckell and R. Rolfes. "Monitoring a 5MW offshore wind energy converter — Condition parameters and triangulation based extraction of modal parameters". In: *Mechanical Systems and Signal Processing* 40.1 (2013), pp. 322–343.

[18] S. Tsiapoki, M.W. Häckell, T. Grießmann, and R. Rolfes. "Damage and ice detection on wind turbine rotor blades using a three-tier modular structural health monitoring framework". In: *Structural Health Monitoring: An International Journal* (2017).

[19] J.M. Ko and Y.Q. Ni. "Technology developments in structural health monitoring of large-scale bridges". In: *Engineering Structures* 27.12 (2005), pp. 1715–1725.

[20] C.Y. Kim, D.S. Jung, N.S. Kim, S.D. Kwon, and M.Q. Feng. "Effect of vehicle weight on natural frequencies of bridges measured from traffic-induced vibration". In: *Earthquake Engineering and Engineering Vibration* 2.1 (2003), pp. 109–115.

[21] C.R. Farrar and K. Worden. *Structural Health Monitoring: A Machine Learning Perspective.* John Wiley & Sons, Ltd, 2012.

[22] C.M. Bishop. *Neural networks for pattern recognition.* Oxford : New York, 1995.

[23] N. Dervilis, M. Choi, S.G. Taylor, R.J. Barthorpe, G. Park, C.R. Farrar, and K. Worden. "On damage diagnosis for a wind turbine blade using pattern recognition". In: *Journal of Sound and Vibration* 333.6 (2014), pp. 1833–1850.

[24] R. Ruotolo and C. Surace. "USING SVD TO DETECT DAMAGE IN STRUCTURES WITH DIFFERENT OPERATIONAL CONDITIONS". In: *Journal of Sound and Vibration* 226.3 (1999), pp. 425–439.

[25] J. Kullaa. "Is temperature an essential measurement in Structural Health Monitoring?" In: *4th International Workshop on Structural Health Monitoring.* Stanford University, CA, USA, 2003.

[26] E. Figueiredo, G. Park, C.R. Farrar, K. Worden, and J. Figueiras. "Machine learning algorithms for damage detection under operational and environmental variability". In: *Structural Health Monitoring: An International Journal* 10.6 (2011), pp. 559–572.

[27] H. Sohn and C.R. Farrar. "Damage diagnosis using time series analysis of vibration signals". In: *Smart Materials and Structures* 10.3 (2001), pp. 446–451.

[28] K. Worden and G. Manson. "The application of machine learning to structural health monitoring". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365.1851 (2007), pp. 515–537.

[29] E.J. Cross, K.Y. Koo, J.M.W. Brownjohn, and K. Worden. "Long-term monitoring and data analysis of the Tamar Bridge". In: *Mechanical Systems and Signal Processing* 35.1-2 (2013), pp. 16–34.

[30] H. Sohn, M. Dzwonczyk, E.G. Straser, A.S. Kiremidjian, K.H. Law, and T. Meng. "An experimental study of temperature effect of modal parameters of the Alamosa Canyon bridge". In: *Earthquake Engineering and Structural Dynamics* 28 (1999), pp. 879–897.

[31] N. Dervilis, K. Worden, and E.J. Cross. "On robust regression analysis as a means of exploring environmental and operational conditions for SHM data". In: *Journal of Sound and Vibration* 347 (2015), pp. 279–296.

[32] O. Bahrami, S. Tsiapoki, M.B. Kane, J.P. Lynch, and R. Rolfes. "Extraction of environmental and operational conditions of wind turbines using tower response data for structural health monitoring". In: *11th International Workshop on Structural Health Monitoring.* Stanford University, CA, USA, 2017.

[33] N. Yampikulsakul, E. Byon, S. Huang, S. Sheng, and M. You. "Condition Monitoring of Wind Power System With Nonparametric Regression Analysis". In: *IEEE Transactions on Energy Conversion* 29.2 (June 2014), pp. 288–299.

[34] A.K. Jain, M.N. Murty, and P.J. Flynn. "Data clustering: a review". In: *ACM Computing Surveys* 31.3 (1999), pp. 264–323.

[35] W. Weijtjens, T. Verbelen, G. De Sitter, and C. Devriendt. "Foundation structural health monitoring of an offshore wind turbine—a full-scale case study". In: *Structural Health Monitoring: An International Journal* 15.4 (2016), pp. 389–402.

[36] M.W. Hackell, R. Rolfes, M.B. Kane, and J.P. Lynch. "Three-Tier Modular Structural Health Monitoring Framework Using Environmental and Operational Condition Clustering for Data Normalization: Validation on an Operational Wind Turbine System". In: *Proceedings of the IEEE* 104.8 (2016), pp. 1632–1646.

[37] B.J. Frey and D. Dueck. "Clustering by Passing Messages Between Data Points". In: *Science* 315.5814 (2007), pp. 972–976.

[38] Z. Zhang and A. Kusiak. "Monitoring Wind Turbine Vibration Based on SCADA Data". In: *Journal of Solar Energy Engineering* 134.2 (2012), p. 021004.

[39] K. Kim, G. Parthasarathy, O. Uluyol, W. Foslien, S. Sheng, and P. Fleming. "Use of SCADA Data for Failure Detection in Wind Turbines". In: *Proceedings of 2011 Energy Sustainability Conference and Fuel Cell Conference*. Washington DC, USA, Aug. 2011.

[40] R. Brincker, L. Zhang, and P. Andersen. "Modal identification of output-only systems using frequency domain decomposition". In: *Smart Materials and Structures* 10 (2001), pp. 441–445.

[41] S. Soyoz and M.Q. Feng. "Long-Term Monitoring and Identification of Bridge Structural Parameters". In: *Computer-Aided Civil and Infrastructure Engineering* 24.2 (2009), pp. 82–92.

[42] A.K. Pandey, M. Biswas, and M.M. Samman. "Damage detection from changes in curvature mode shapes". In: *Journal of Sound and Vibration* 145.2 (1991), pp. 321–332.

[43] C. Williams and O.S. Salawu. "Damping as a damage indication parameter". In: *15th International Modal Analysis Conference*. Orlando, USA, 1997, pp. 1531–6.

[44] C.R. Farrar, S.W. Doebling, P.J. Cornwell, and E.G. Straser. "Variability of modal parameters measured on the Alamosa Canyon Bridge". In: *International Modal Analysis Conference*. 1996.

[45] T.J. Johnson and D.E. Adams. "Transmissibility as a Differential Indicator of Structural Damage". In: *Journal of Vibration and Acoustics* 124.4 (2002), p. 634.

[46] C. Devriendt and P. Guillaume. "Identification of modal parameters from transmissibility measurements". In: *Journal of Sound and Vibration* 314.1-2 (2008), pp. 343–356.

[47] W. Yang, Z. Lang, and W. Tian. "Condition Monitoring and Damage Location of Wind Turbine Blades by Frequency Response Transmissibility Analysis". In: *IEEE Transactions on Industrial Electronics* 62.10 (2015), pp. 6558–6564.

[48] K. Worden, G. Manson, and D. Allman. "EXPERIMENTAL VALIDATION OF A STRUCTURAL HEALTH MONITORING METHODOLOGY: PART I. NOVELTY DETECTION ON A LABORATORY STRUCTURE". en. In: *Journal of Sound and Vibration* 259.2 (2003), pp. 323–343.

[49] F.P. Kopsaftopoulos and S.D. Fassois. "Vibration based health monitoring for a lightweight truss structure: Experimental assessment of several statistical time series methods". In: *Mechanical Systems and Signal Processing* 24.7 (2010), pp. 1977–1997.

[50] P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems*. Springer US, 1996.

[51] M. Goursat, M. Döhler, L. Mevel, and P. Andersen. "Crystal Clear SSI for Operational Modal Analysis of Aerospace Vehicles". In: *Structural Dynamics, Volume 3.* Ed. by Tom Proulx. Springer New York, 2011, pp. 1421–1430.

[52] P. Moser and B. Moaveni. "Environmental effects on the identified natural frequencies of the Dowling Hall Footbridge". In: *Mechanical Systems and Signal Processing* 25.7 (2011), pp. 2336–2357.

[53] Lennart Ljung. *System identification: theory for the user.* 2nd ed. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999.

[54] O. Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models.* Springer, 2011.

[55] J.B. Bodeux and J.C. Golinval. "Application of ARMAV models to the identification and damage detection of mechanical and civil engineering structures". In: *Smart Materials and Structures* 10.3 (2001), pp. 479–489.

[56] P. Kraemer. "Schadendiagnoseverfahren für die Zustandsüberwachung von Offshore-Windenergieanlagen". PhD thesis. Siegen, Germany: Universität Siegen, 2011.

[57] B. Peeters and G. De Roeck. "One-year monitoring of the Z24-Bridge: environmental effects versus damage events". In: *Earthquake Engineering & Structural Dynamics* 30.2 (2001), pp. 149–171.

[58] E. Figueiredo, J. Figueiras, G. Park, C.R. Farrar, and K. Worden. "Influence of the Autoregressive Model Order on Damage Detection: Influence of the autoregressive model order on damage detection". In: *Computer-Aided Civil and Infrastructure Engineering* 26.3 (2011), pp. 225–238.

[59] A. Clement and S. Laurens. "Vibration-based damage detection in a concrete beam under temperature variations using AR models and state-space approaches". In: *Journal of Physics: Conference Series* 305 (2011), p. 012040.

[60] Y. Lu and F. Gao. "A novel time-domain auto-regressive model for structural damage diagnosis". In: *Journal of Sound and Vibration* 283.3-5 (2005), pp. 1031–1049.

[61] C.G. Lee and Yun C.B. "Parameter identification of linear structural dynamic systems". In: *Computers & Structures* 40.6 (1991), pp. 1475–1487.

[62] K.K. Nair, A.S. Kiremidjian, and K.H. Law. "Time series-based damage detection and localization algorithm with application to the ASCE benchmark structure". In: *Journal of Sound and Vibration* 291.1-2 (2006), pp. 349–368.

[63] D.E. Adams and Charles R. Farrar. "Classifying Linear and Nonlinear Structural Damage Using Frequency Domain ARX Models". In: *Structural Health Monitoring: An International Journal* 1.2 (2002), pp. 185–201.

[64] T.R. Fasel, H. Sohn, G. Park, and C.R. Farrar. "Active sensing using impedance-based ARX models and extreme value statistics for damage detection". In: *Earthquake Engineering & Structural Dynamics* 34.7 (2005), pp. 763–785.

[65] Jerome P. Lynch. "Linear classification of system poles for structural damage detection using piezoelectric active sensors". In: *Proceedings of SPIE.* Ed. by S.C. Liu. Vol. 5391. San Diego, CA, 2004.

[66] J.P. Lynch. "Damage Characterization of the IASC-ASCE Structural Health Monitoring Benchmark Structure by Transfer Function Pole Migration". In: *Structures Congress 2005.* New York, NY: American Society of Civil Engineers, 2005, pp. 1–10.

[67] R. Andrew Swartz and J.P. Lynch. "Damage Characterization of the Z24 Bridge by Transfer Function Pole Migration". In: *Proceedings of the International Modal Analysis Conference (IMAC) XXVI.* Orlando, Florida, USA, 2008.

[68] M. Basseville, L. Mevel, and M. Goursat. "Statistical model-based damage detection and localization: subspace-based residuals and damage-to-noise sensitivity ratios". In: *Journal of Sound and Vibration* 275.3-5 (2004), pp. 769–794.

[69] M. Döhler, L. Mevel, and F. Hille. "Subspace-based damage detection under changes in the ambient excitation statistics". In: *Mechanical Systems and Signal Processing* 45.1 (2014), pp. 207–224.

[70] J. Neter, W. Wasserman, and M.H. Kutner. *Applied linear statistical models.* 4th ed. Irwin, 1996.

[71] G.E.P Box, G.M. Jenkins, G.C. Reinsel, and G.M. Ljung. *Time series analysis: forecasting and control.* Wiley, 2016.

[72] V. Vrabie, P. Granjon, and C. Serviere. “Spectral kurtosis: from definition to application”. In: *6th IEEE International Workshop on Nonlinear Signal and Image Processing.* Grado-Trieste, Italy, 2003.

[73] E. Figueiredo, G. Park, J. Figueiras, C.R. Farrar, and K. Worden. *Structural Health Monitoring Algorithm Comparisons Using Standard Data Sets.* Tech. rep. LA-14393. Los Alamos National Laboratory, 2009.

[74] “An overview of wireless structural health monitoring for civil structures”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365 (2007), pp. 345–372.

[75] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* John Wiley & Sons, 2012.

[76] C. Surace and K. Worden. “Novelty detection in a changing environment: A negative selection approach”. In: *Mechanical Systems and Signal Processing* 24.4 (2010), pp. 1114–1128.

[77] W.H. Hu, C.s Moutinho, F. Magalhaes, E. Caetano, and A. Cunha. “Analysis and Extraction of Temperature Effects on Natural Frequencies of a Footbridge based on Continuous Dynamic Monitoring”. In: *IOMAC’09.* May 2009, pp. 55–62.

[78] H. Sohn, J.A. Czarnecki, and C.R. Farrar. “Structural Health Monitoring Using Statistical Process Control”. In: *Journal of Structural Engineering* 126.11 (2000), pp. 1356–1363.

[79] J. Long and O. Buyukozturk. “Automated Structural Damage Detection Using One-Class Machine Learning”. In: *Dynamics of Civil Structures, Volume 4.* Ed. by Fikret Necati Catbas. Springer International Publishing, 2014, pp. 117–128.

[80] Y. Bao, C. Song, W. Wang, T. Ye, L. Wang, and L. Yu. “Damage Detection of Bridge Structure Based on SVM”. In: *Mathematical Problems in Engineering* 2013 (2013), pp. 1–7.

[81] K Worden and A.J. Lane. “Damage identification using support vector machines”. In: *Smart Materials and Structures* 10.3 (2001), pp. 540–547.

[82] A.R. Webb and Keith D. Copsey. *Statistical pattern recognition.* 3. ed. Wiley, 2011.

[83] R.E. Schapire and Y. Freund. *Boosting: foundations and algorithms.* Adaptive computation and machine learning series. MIT Press, 2012.

[84] Yoav Freund and Robert E Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.

[85] P. Viola and M. Jones. “Fast and robust classification using asymmetric adaboost and a detector cascade”. In: *NIPS 2002.* MIT, 2002, pp. 1311–1318.

[86] P. Viola and M.J. Jones. “Robust Real-Time Face Detection”. In: *International Journal of Computer Vision* 57.2 (2004), pp. 137–154.

[87] B. Wu, H. Ai, C. Huang, and S. Lao. “Fast rotation invariant multi-view face detection based on real adaboost”. In: IEEE, 2004, pp. 79–84.

[88] S. Steiniger, T. Lange, D. Burghardt, and R. Weibel. “An Approach for the Classification of Urban Building Structures Based on Discriminant Analysis Techniques”. In: *Transactions in GIS* 12.1 (2008), pp. 31–59.

[89] E. Alfaro, N. García, M. Gámez, and D. Elizondo. “Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks”. In: *Decision Support Systems* 45.1 (2008), pp. 110–122.

[90] M. Rangini and W. Jiji. “Identification of Alzheimer’s Disease using Adaboost Classifier”. In: *Proceedings of the International Conference on Applied mathematics and Theoretical Computer Science*. 2013, p. 229.

[91] Y. Shin, D.W. Kim, J.Y. Kim, K.I. Kang, M.Y. Cho, and H.H. Cho. “Application of AdaBoost to the Retaining Wall Method Selection in Construction”. In: *Journal of Computing in Civil Engineering* 23.3 (2009), pp. 188–192.

[92] Z. Li, J. Guo, W. Liang, X. Xie, G. Zhang, and S. Wang. “Structural Health Monitoring Based on RealAdaBoost Algorithm in Wireless Sensor Networks”. In: *Wireless Algorithms, Systems, and Applications*. Vol. 8491. Springer International Publishing, 2014, pp. 236–245.

[93] H. Furuta, H. Hattori, and Dam M. Frangopol. “Damage Assessment of reinforced concrete bridge decks using AdaBoost”. In: *The 3rd international ASRANet Colloquium*. Glasgow, UK, 2006.

[94] D. Kim and M. Philen. “Damage classification using Adaboost machine learning for structural health monitoring”. In: *Proceedings of SPIE*. Ed. by M. Tomizuka, C.B. Yun, V. Giurgiutiu, and J.P. Lynch. Vol. 7981. 2011, 79812A.

[95] S. Mazumder. *Numerical Methods for Partial Differential Equations: Finite Difference and Finite Volume Methods*. Elsevier Science, 2015.

[96] J.B. Hag and D.S. Bernstein. “Nonminimum-phase zeros - much to do about nothing - classical control - revisited part II”. In: *IEEE Control Systems Magazine* 27.3 (2007), pp. 45–57.

[97] R.A. Johnson and D.W. Wichern. *Applied multivariate statistical analysis*. 6th ed. Upper Saddle River, N.J: Pearson Prentice Hall, 2007.

[98] R.C. Dorf and R.H. Bishop. *Modern Control Systems*. Pearson Prentice Hall, 2008.

[99] T. Kailath. “The Divergence and Bhattacharyya Distance Measures in Signal Selection”. In: *IEEE Transactions on Communications* 15.1 (1967), pp. 52–60.

[100] R.K. Som. *Practical sampling techniques*. 2nd ed., rev. and expanded. Statistics, textbooks and monographs v. 148. M. Dekker, 1996.

[101] N. Bose. “Icing on a small horizontal-axis wind turbine — Part 1: Glaze ice profiles”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 45.1 (1992), pp. 75–85.

[102] Fraunhofer IWES. *Proposed and Actual Loading*. Technical Report IWES-B-A-13-100-013. Bremerhaven, Germany.

[103] S. Suresh. *Fatigue of Materials*. 2nd ed. Cambridge University Press, 1998.

[104] R.P. Kambour. “A review of crazing and fracture in thermoplastics”. In: *Journal of Polymer Science: Macromolecular Reviews* 7.1 (1973).

[105] M.B. Kane. “Wirelessly Enabled Control of Cyber-Physical Infrastructure Systems”. PhD thesis. Ann Arbor, MI, USA: University of Michigan, Department of Civil, Environmental Engineering, Department of Electrical Engineering, and Computer Science, 2014.

[106] J.M. Jonkman and Marshall L. Buhl Jr. *FAST User’s Guide*. Technical Report NREL/EL-500-38230. Golden, Colorado: National Renewable Energy Laboratory, Aug. 2005.

[107] B.J. Jonkman and Marshall L. Buhl Jr. *TurbSim User’s Guide: Version 1.06.00*. Technical Report NREL/TP-500-39797. Golden, Colorado: National Renewable Energy Laboratory, Sept. 2006.