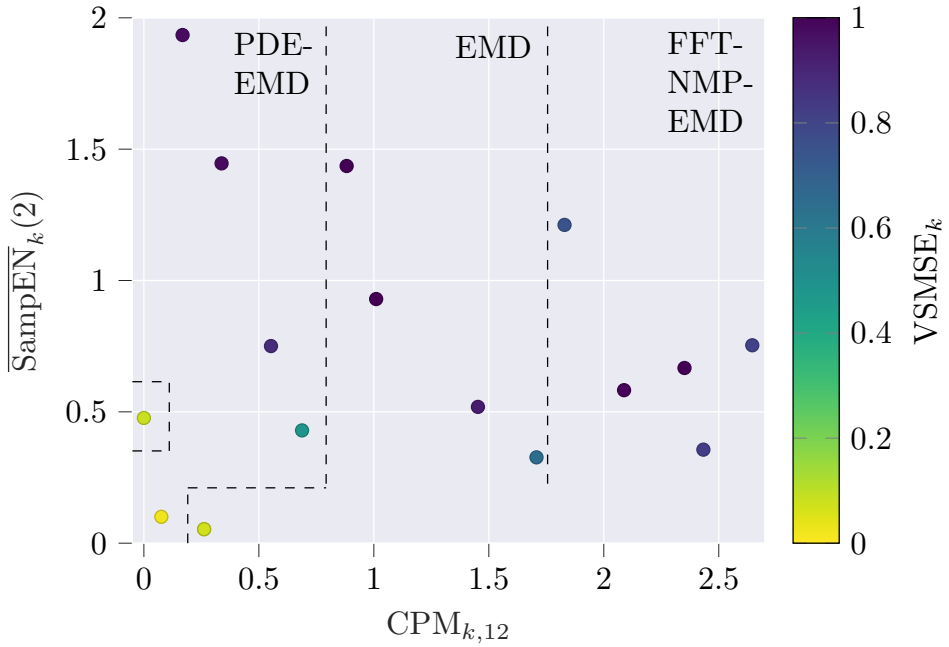
Adaptive Data Analysis

Theoretical Results and an Application to Wind Power Forecasting

Master's Thesis

Mathematical Engineering



Aalborg University
Mathematical Engineering

This project has been written in LaTeX with figures produced in TikZ and Matplotlib unless otherwise stated. Scripts have been made using Python 3.9. In case of stains, please note the washing guide below.

AALBORG UNIVERSITY
STUDENT REPORT

Mathematical Engineering
Aalborg University
http://www.aau.dk

Title:
Adaptive Data Analysis

Subtitle:
Theoretical Results and an Application to Forecasting of the Danish Wind Power Production

Project Period:
Spring Semester 2022

Project Group:
math-22-mattek-10-1217b

Participants:
Andreas Anton Andersen
Martin Voigt Vejling
Morten Stig Kaaber

Supervisors:
Christophe Biscio
Petar Popovski
Tobias Kallehauge

Copies: 1

Number of Pages: 113

Date of Completion:
June 1, 2022

Abstract:

In this thesis, different adaptive decomposition methods are implemented as a pre-processing step for one hour ahead forecasting algorithms in an online setup on the Danish wind power production. Four adaptive decomposition methods have been explored in this thesis, i.e. the empirical mode decomposition (EMD), two compressive sensing based methods, and a partial differential equation based method dubbed PDE-EMD. Additionally, the ability of the EMD in an offline setup has been tested. Relevant theory related to the implemented methods are presented and theoretical properties relating to the uniqueness of the compressive sensing based methods are proven. Initially, properties of the decomposition methods are tested both using simulated examples and the Danish wind power production data. Afterwards, the forecasting algorithms are tested on the data. Here the methods are compared to long short-term memory neural network and autoregressive baselines. For an online setup, it is found that using the PDE-EMD, the results are similar to that of the baselines, whereas the EMD based models and the compressive sensing based models perform worse. However, in an offline setup, the EMD method significantly outperforms the other forecasting algorithms. Based on the findings of this thesis, it is concluded that decomposition based forecasting works great in an offline setup, but more work is needed for it to be applicable in an online setup.

Preface

This thesis has been written in cooperation with Energinet which is the transmission system operator in Denmark.

The thesis has been made by a group master students studying Mathematical Engineering at Aalborg University. The group thanks the supervisors Christophe Biscio, Tobias Kallehauge, and Petar Popovski for their supervision throughout the semester. Additionally, the group thanks Energinet for a good cooperation during the semester and would especially like to thank the contact person Lasse Diness Borup. Furthermore, the group thanks CLAAUDIA for making the AI-Cloud service available for GPU computing and Strato service available for CPU computing.

The references throughout the thesis have been handled with the alphabetical IEEE-method with specified page numbers of the respective source. Further information about the sources may be found in the bibliography.

Symbols in boldface are used to denote vectors and matrices. All vectors are considered as column vectors unless otherwise specified. Types of spaces and mathematical quantities used in this report appear from the **Notation List**. Additionally, abbreviations used throughout the thesis are listed in the **Abbreviation List**.

The figures have been made using the Ti*k*Z package in LaTeX, and using the `matplotlib.pyplot` package in Python by the authors of the thesis unless otherwise stated. The scripts which have been used to obtain results in the thesis can be found in the GitHub repository.

Aalborg University, June 1, 2022

Andreas Anton Andersen
<aand17@student.aau.dk>

Martin Voigt Vejling
<mvejli17@student.aau.dk>

Morten Stig Kaaber
<mkaabe17@student.aau.dk>

Resumé på Dansk

Spektraldekomposition er et værktøj, som ofte bruges til at analysere signaler. Den mest kendte og nok også mest udbredte dekompositionsmetode er Fouriertransformationen. Dog har Fouriertransformationen sine begrænsninger, og er bedst egnet til lineære og stationære systemer. I denne specialeafhandling arbejdes der med signaler, som hverken er lineære eller stationære, og derfor anvendes de såkaldte adaptive dekompositionsmetoder, som netop kan anvendes til at opsplitte ikke-lineære og ikke-stationære signaler i komponenter, der tilnærmelsesvis har lokal smal båndbredde. I 1998 blev den adaptive dekompositionsmetode ved navn empirical mode dekomposition (EMD) introduceret, og siden da er der blevet introduceret yderligere adaptive dekompositionsmetoder.

I dette speciale er målet at lave prædiktioner én time frem af vindkraftproduktionen i en online opsætning, og til dette formål er fire forskellige adaptive dekompositionsmetoder introduceret. Den ene metode er baseret på partielle differentialligninger og kaldes PDE-EMD, to af metoderne er baseret på compressive sensing, og slutteligt er EMDen inkluderet. Disse metoder er anvendt som forbehandling på et vindkraftproduktionssignal, inden en prædiktion er lavet. Desuden er EMDen også testet i en offline opsætning.

For at introducere disse metoder er relevant teori vedrørende compressive sensing og partielle differentialligninger introduceret. Desuden er to teoremer omhandlende entydigheden af dekompositionen for compressive sensing metoderne bevist.

Dekompositionsmetodernes evne er testet ift. den resulterende opsplitnings ortogonalitet, metodernes evne til at begrænse randeffekter samt metodernes evne til at være konsekvente ved små ændringer i data. Ved disse undersøgelser ses det, at PDE-EMD metoden klarer sig bedst ift. at være konsekvent samt i at begrænse randeffekter, hvorimod metoden klarer sig dårligst ift. ortogonalitet.

Prædiktionerne laves ved at træne et long short-term memory (LSTM) neuralt netværk til at prædiktere hver komponent fra dekompositionerne, og prædiktionen fås ved at aggregere prædiktionerne fra hver komponent. Prædiktionsevnen for metoderne er sammenlignet med et LSTM neuralt netværk og en autoregressiv model. For den

offline metode ses det, at en klar forbedring kan opnås ved at bruge dekompositionsmetoder. For de online metoder viser resultaterne, at compressive sensing metoderne og EMDen klarer sig dårligere end baseline metoderne, hvorimod PDE-EMD metoden klarer sig lige så godt som baseline metoderne. Ved at analysere resultaterne ses det, at PDE-EMD metoden har en lav mean squared error (MSE) på hver komponent, dog bliver MSEen efter aggregering af komponenterne større end summen af MSEerne. Dette skyldes positive korrelationer i prædiktionsfejlen, hvilket kunne skyldes dekompositionens manglende ortogonalitet.

Abbreviation List

AIO	Average index of orthogonality.
AM-FM	Amplitude-modulated and frequency-modulated.
BPDN	Basis pursuit denoising.
CPM	Consistency performance measure.
CS	Compressive sensing.
DNN	Deep neural network.
EEEI	End effect evaluation index.
EMD	Empirical mode decomposition.
FFT	Fast Fourier transform.
FFT-NMP-EMD	Fast Fourier transform non-linear matching pursuit based empirical mode decomposition.
HHT	Hilbert-Huang transform.
IA	Instantaneous amplitude.
IF	Instantaneous angular frequency.
IMF	Intrinsic mode function.
IO	Index of orthogonality.
LSTM	Long short-term memory.
MIO	Maximum index of orthogonality.

MP	Matching pursuit.
MSE	Mean squared error.
NBIAS	Normalised bias.
NMAE	Normalised mean absolute error.
NMP	Non-linear marching pursuit.
NMP-EMD	Non-linear matching pursuit based empirical mode decomposition.
NP-hard	Non-deterministic polynomial-time hard.
NRMSE	Normalised root mean squared error.
NWP	Numerical weather predictions.
OMP	Orthogonal matching pursuit.
PDE	Partial differential equation.
PDE-EMD	Partial differential equation based empirical mode decomposition.
SampEN	Sample entropy.
SP	Sifting procedure.
VSMSE	Variance scaled mean squared error.

Notation List

Miscellaneous

$\mathbb{R}$	The set of real numbers.
$\mathbb{R}^+$	The set of non-negative real numbers.
$\mathbb{C}$	The set of complex numbers.
$\mathbb{N}$	The set of natural numbers.
$\boldsymbol{I}$	Identity matrix.
j	Imaginary unit.
C^n	n times continuously differentiable.
L^p	Space of p-order Lebesgue integrable functions.
ℓ^p	Space of p-order summable vectors.
$\|\cdot\|_p$	L^p-norm or ℓ^p-norm.
$\langle\cdot,\cdot\rangle$	L^2 or ℓ^2 inner product.
$\mathrm{Re}\{\cdot\}$	Real part of complex number.
$\mathcal{F}\{\cdot\}$	Fourier transform.
$\mathcal{H}\{\cdot\}$	Hilbert transform.
$\lvert\cdot\rvert$	Absolute value of a real number or cardinality of a set.
$\mathrm{diag}(\cdot)$	Diagonal matrix with $\cdot$ in the diagonal.

$P(t)$	Wind power at time t.
s	Sparsity of adaptive decomposition.
$c_k(t)$	The kth component of a decomposition.
$r(t) = c_{s+1}(t)$	Residual.
p	Amount of power history used.
$\boldsymbol{P}_t^p$	Window of wind power of length p ending at time t.
$\boldsymbol{c}_{k,t}^p$	Window of length p ending at time t of the kth component of a decomposition.
τ	Forecast horizon in samples.
$\hat{P}_t(t+\tau)$	τ-ahead forecast of P given time t.
$\hat{c}_{k,t}(t+\tau)$	τ-ahead forecast of component k given time t.
q	Window length of online decomposition.
$\omega(t)$	Instantaneous angular frequency.
$\theta(t)$	Instantaneous phase.
$a(t)$	Instantaneous amplitude.
Adaptive Decomposition Methods	
$y(t)$	Observed signal.
$y_A(t)$	$y(t)$ converted to an analytical signal.
$c(t)$	Intrinsic mode function.
$h(t)$	Intrinsic mode function candidate.
$m(t)$	Local mean.
$e_u(t)$	Upper envelope.
$e_l(t)$	Lower envelope.

Compressive Sensing with Time-Frequency Dictionaries

$[m]$	$\{1, 2, \ldots, m\}$.
$\operatorname{supp}(\boldsymbol{x})$	$\{j \in [m] : x_j \neq 0\}$.
T	Length of observation window.
L_θ	Reciprocal of the smallest scale of θ, i.e. $\frac{\theta(T)-\theta(0)}{2\pi}$.
$\mathcal{D}$	Dictionary of mono-components.
λ	Smoothness parameter.
$V(\theta; \lambda)$	Over-complete Fourier dictionary.
v_κ	Basis functions for V.
δ	Noise level/energy in residual.
a_k^l	Value of a after the kth outer iteration and lth inner iteration.
$\hat{\boldsymbol{a}}_k^{l+1}$	Representation of a_k^{l+1} in $V(\theta_k^l; \lambda)$.
f	Frequency.
$\chi_{V(\theta_k^l;\lambda)}$	Low-pass filter with cut-off frequency determined by the highest frequency in $V(\theta_k^l; \lambda)$.
$\hat{h}_k^l$	Frequency response of $\chi_{V(\theta_k^l;\lambda)}$.

Uniqueness of Compressive Sensing Algorithm

$\mathcal{D}_\varepsilon$	Dictionary of scale separated signals with separation factor ε.
$\mathcal{A}_{\varepsilon,f_r}$	Dictionary of well-separated signals with scale separation factor ε and frequency ratio f_r.
$\hat{(\cdot)}$	Fourier transform of $(\cdot)$.
ψ	Wavelet function.
$\mathcal{W}\{\cdot\}$	Wavelet transform.

Partial Differential Equation Based Adaptive Decomposition

u	Solution to partial differential equation.
S	Continuous-time spatial domain.
$S_T = S \times (0, T]$	Continuous-time spatial and time domain.
$C_m^n(S_T)$	Functions which are m times continuous differentiable in the temporal variable and n continuous differentiable in the spatial variables.
$\mathcal{S}$	Discrete-time domain.
ν	Metric.
$\overline{S}$	Closure of S.
B_S	Boundary of S.
K	Length of observation window.
T	Point of convergence for PDE solution.
u_i^j	Discrete observation $u(x_i, t_j)$.
$m(x) = u(x, T)$	Local mean.
α	Diffusivity constant.

Experimental Setup

$\boldsymbol{P}_t^q$	Window of wind power of length q ending at time t.
$\boldsymbol{c}_{k,t}^q$	kth component of a decomposition of $\boldsymbol{P}_t^q$.
$c_{k,m}(t)$	Observation at time t for the kth component in an adaptive decomposition of a window ending at time $m > t$.
ξ	Window shift used when assigning target value.

Contents

1. Problem Analysis

This thesis has been made in cooperation with Energinet with the purpose of researching different time series decomposition methods used as a pre-processing step in a forecasting algorithm for the Danish wind power production. This chapter is structured as follows: The primary tasks of Energinet are introduced in Section 1.1, afterwards, in Section 1.2, the data and some of the factors which contribute to the uncertainty in the data are described. Then, in Section 1.3, the problem of forecasting the wind power production and the principle behind decomposition methods for this setup are presented, and finally the problem statement is given in Section 1.4.

1.1 The Primary Tasks of Energinet

The primary task of Energinet is to solve the trilemma of energy: (i) Convert power systems to renewable energy (ii) while preserving the security of supply for consumers (iii) at a low price. [Ene22b]

The power in the power grid is dependent on both the time and the location. The power grid consists of a discrete set of nodes that are connected by edges given by the power cables. The nodes in the discrete domain include points where power enters the power grid, points where power cables meet, i.e. transformer stations, and points where power exits the power grid. Power can enter the power grid by being generated by an energy source or it can enter by being purchased from another nation. Power can exit the power grid by being consumed or by being sold to other nations [Ene22a]. Additionally, when power is transported via an edge in the power grid, a loss of energy occurs due to energy dissipation which is caused by a number of internal and external factors such as resistance, atmospheric conditions etc. A graph illustrating a simplified power grid is shown in Fig. 1.1.

To secure the supply for consumers, i.e. task (ii), it is seminal to ensure a balance between the energy supply and demand in the power grid [Ene21]. We pose this issue by the inequality [Ene22d]

$$|S - D| \leq \varepsilon \tag{1.1}$$

where D is the demand, S is the supply, and $\varepsilon > 0$ is the acceptable threshold for balance.

The supply is affected by power trading and power generation by sources such as

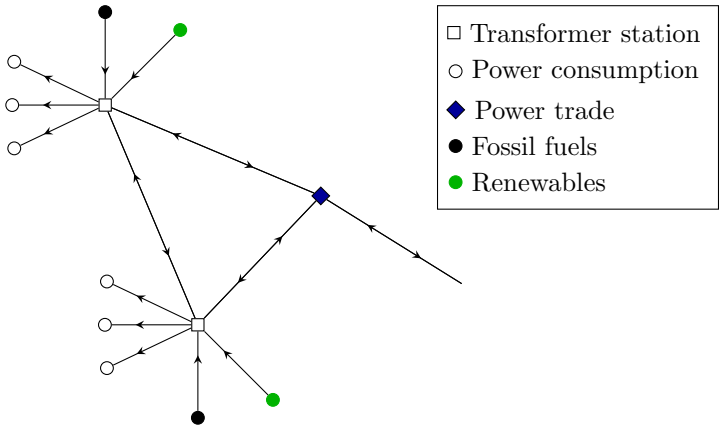


Figure 1.1: Schematic depiction of a simplified power grid. Arrows indicate the flow of power through the grid and the length of an edge indicates the distance between the nodes.

coal power, solar photovoltaic power, wind power etc. [Ene22c]. The power generation from fossil fuel based power sources such as coal power can be accurately predicted and controlled whereas the renewable sources such as wind power are largely dependent on atmospheric conditions and as such are difficult to predict. However, it is important to predict the power generation in the power grid in order to balance the supply and demand as to uphold Eq. (1.1) [Ene21].

Generally, the tasks of Energinet can be formulated as an optimisation problem based on solving the trilemma of energy. This is done by regulating the electricity market, planning the infrastructure, etc. [Ene22d; Ene21].

1.2 Data and System Uncertainties

In order to forecast the wind power production, it is necessary to understand the underlying system from which the wind power production data is sampled. The power produced by a wind turbine is dependent on several weather factors such as wind speed and air density. This can be seen by the theoretical relationship between wind speed v and wind power P [Jai16, p. 20]

$$P = \frac{1}{2}\rho_{\text{air}} A C v^3 \tag{1.2}$$

where ρ_{air} is the air density, A is the swept area, and C is a constant describing the physical properties of the wind turbine. The constant C is bounded by the Betz limit $C \leq \frac{16}{27}$ and states that ideally a wind turbine can be approximately 59% effective [Jai16, pp. 11-25]. As is apparent from Eq. (1.2), the wind power depends on weather conditions, namely air density and wind speed in the direction towards the blades of the wind turbine.

The power grid in Denmark is divided into 2 power grids called DK1 and DK2, respectively. This has been done for historical reasons and influences the data avail-

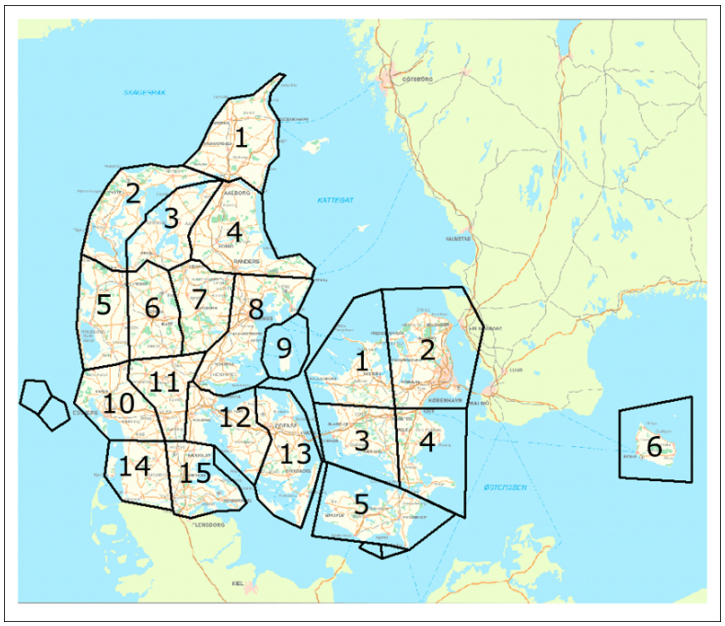


Figure 1.2: The sub-grids in DK1 and DK2. Taken from a presentation given by Energinet.

able to us. Geographically, DK1 covers the western part of Denmark, i.e. Jutland and Funen, whereas DK2 consists of the eastern part of Denmark, i.e. Zealand and the adjacent islands. Each power grid is further divided into a number of sub-grids. DK1 contains 15 sub-grids and DK2 contains 6 sub-grids yielding a total of 21 sub-grids. The sub-grid division can be seen in Fig. 1.2. [Ene22a]

The data available to us represents the wind power production in each of the 21 sub-grids provided in 5 minute intervals. The data as well as the underlying system are influenced by several factors which contribute to the uncertainty in the data. Firstly, a data acquisition process called SCADA-upscaling is used to collect the data. In this process, sensor measurements from some of the wind turbines in each sub-grid are used to approximate the actual wind power production in the sub-grid. This introduces uncertainty in the data since these measurements do not completely represent the entire sub-grid. Moreover, inaccuracy is expected from the sensor measurements themselves. Additionally, the data is affected by how the power grids have been regulated during the data acquisition. Specifically, in order to ensure balance in the power grid, the wind power production can be down-regulated. This down-regulation is given to us at power grid resolution, i.e. the down-regulation for DK1 and DK2 are available, but the down-regulation for each of the 21 sub-grids is not available. Finally, as can be seen in Eq. (1.2), weather conditions affect the wind power production and as such any uncertainty in the weather conditions also contributes to uncertainty in the wind power production.

1.3 Wind Power Production Forecasting

Wind power forecasting is an active research area with many existing methods in the literature. Different time horizons for forecasting yield different applications and also warrants the use of different methods. Some applications are related to the energy market, while others are based on specific tasks related to the wind turbines such as determining the pitch of the turbine blades and maintenance of the turbines. In general, the forecasting time horizons are split into four categories, i.e. very short term, short term, medium term, and long term. Very short term forecasts are forecasts up to 30 minutes ahead and can be used for regulation actions, real-time grid operations, market clearing, and turbine control. Short term forecasts range from 30 minutes to 6 hours and these can be used for load dispatch planning. Medium term forecasts range from 6 hours to 1 day and can be used for operational security in the electricity market, energy trading, and online and offline generating decisions. Finally, long term forecasts are forecasts which are longer than 1 day and these can be used for reserve requirements, maintenance schedules, optimum operating cost, and operation management. [Han+20; Ene19]

Broadly, the methods can be classified according to the explanatory variables used for the forecast. These classes are the physical numerical weather prediction (NWP) based methods [CD14; Sin16], the wind power history based methods [Nie+07; Liu+10; Tou+21], and methods using a combination of the two aforementioned inputs [MNN05; SH07; Zhe+13; Zha+19]. The term NWP refers to forecasts of meteorological variables such as wind speed, wind direction, air density, etc. using spatio-temporal computational fluid dynamics models [Qia+19]. Using NWPs is useful when doing medium and long term forecasts, however computing the NWPs is computationally expensive thereby limiting the resolution and introducing a delay in real-time applications. Moreover, for very short and short term forecasts, the wind power production history should be used to exploit the autocorrelations of the wind power production time series [Qia+19]. Additionally, methods using the wind speed history to forecast the wind speed followed by a mapping to the wind power production are also widely used in the literature and are referred to as indirect methods [Bok+19].

This thesis is focused on a one hour forecast, i.e. a short term forecast, as Energinet highlighted this time frame as being of interest. For the very short and short term time horizons, a common method used for forecasting is data-driven non-linear regression models using the wind power production history as the explanatory variable. Let $\{P(t)\}_{t=1}^{n}$ be the observed wind power production and let $\boldsymbol{P}_t^p = \big(P(t), P(t-1), \ldots, P(t-p+1)\big)^T$ for $p \geq 1$ and $t = p, p+1, \ldots, n$ be data vectors of p samples of wind power production history. Assume that the power history up to time t is known, then a τ-ahead forecast can be made as

$$\hat{P}_t(t+\tau) = f(\boldsymbol{P}_t^p; \phi) \tag{1.3}$$

for some model f with parameters ϕ. It should be noted that the combination of

wind power production history and NWPs is often used for short term forecasts in the literature. The benefit of using NWPs is dependent on the quality and resolution of the NWPs as well as the specific time horizons, i.e. the longer the time horizons, the more beneficial the use of NWPs can be. [Qia+19]

A variety of methods for designing the forecast model, also referred to as a forecaster, for very short and short term applications exists in the literature [Han+20]. The methods can be categorised into the classical statistical time series methods, e.g. autoregressive models [MNN05; Nie+07; PS+09; ZCA16; Lyd+16; AS17], and the more modern machine learning techniques, e.g. deep neural networks (DNNs) [BT07; SH07; Zha+19; Zha+20; Pen+20; Tou+21]. Additionally, a variety of pre-processing and post-processing techniques have been employed to improve the performance of the forecasting models [LC19]. In relation to forecasting, pre-processing encompasses any processing of the data prior to applying a forecaster. This includes outlier detection, denoising, dimensionality reduction, feature extraction, etc. [GLH15, pp. 10-16][Bis06, pp. 2-3][LC19]. Feature extraction refers to methods of transforming the data into a new space where ideally forecasting is easier. This includes decomposition methods which is a class of pre-processing methods that decomposes time series data into components for which patterns are easier to recognise than for the original time series [LC19]. Models combining two or more techniques are referred to as hybrid models [Bok+19]. In this thesis, hybrid decomposition based models using a DNN forecaster are considered, i.e. models using a time series decomposition method as a pre-processing step when doing forecasting with a DNN model. These are then compared to a baseline DNN forecaster and a baseline autoregressive forecaster.

1.3.1 Decomposition Based Models

It is generally recognised that wind power production time series are generated by a non-linear and non-stationary system and thereby the time series have a low predictability [Qia+19]. We also refer to the time series as being non-linear and non-stationary. Due to the low predictability, methods employing time series decomposition have been widely used for forecasting of wind power (or wind speed) in the literature with encouraging results [Liu+10; SV13; Zhe+13; MZR14; XHY19; LDB21]. These methods combine the capabilities of different approaches by decomposing the wind power production (or wind speed) time series into components which are more predictable than the original time series and subsequently doing the forecast.

With the consideration that the wind power production time series is non-linear and non-stationary, the use of adaptive data analysis methods is of interest. Using an adaptive data analysis method, a signal is decomposed into a sum of locally narrowband amplitude-modulated and frequency-modulated (AM-FM) signals and a residual, i.e. for some signal $P(t)$ the decomposition is made as

$$P(t) = \sum_{k=1}^{s} c_k(t) + r(t) \tag{1.4}$$

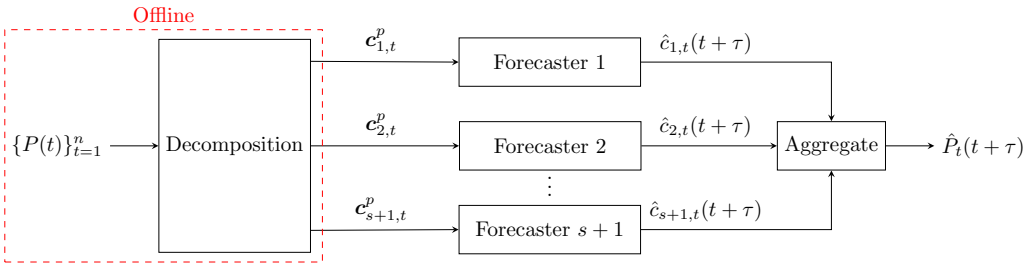


Figure 1.3: Block diagram of a typical decomposition based method.

for $t = 1, \ldots, n$ where c_k are the AM-FM signals for $k = 1, \ldots, s$ and r is the residual. We adopt the convention that r is also denoted c_{s+1}.

The most widely used structure of decomposition based methods is to construct a forecaster for each sub-series [Qia+19]. In this setup, it is assumed that the wind power production data $\{P(t)\}_{t=1}^{n}$ can be decomposed into the representation in Eq. (1.4). The components are then given as input to a forecaster each yielding the τ-ahead forecasts

$$\hat{c}_{k,t}(t+\tau) = f_k(\boldsymbol{c}_{k,t}^{p}; \phi_k)$$

for models f_k parameterised by ϕ_k and with $\boldsymbol{c}_{k,t}^{p} = \big(c_k(t), c_k(t-1), \ldots, c_k(t-p+1)\big)^T$ for $k = 1, \ldots, s+1$, $p \geq 0$, and $t = p, p+1, \ldots, n$. The forecast of the wind power production is then computed as

$$\hat{P}_t(t+\tau) = \sum_{k=1}^{s+1} \hat{c}_{k,t}(t+\tau). \tag{1.5}$$

A block diagram of this structure is shown in Fig. 1.3.

In 1998, the adaptive data analysis method called the empirical mode decomposition (EMD) was introduced by [Hua+98]. The EMD is an intuitive method which can be used to compute a decomposition on the form Eq. (1.4) where the AM-FM components are called intrinsic mode functions (IMFs). However, the EMD introduces several problems. Firstly, the method lacks a solid mathematical foundation. Furthermore, the EMD suffers from mode mixing and end effects.

To exemplify mode mixing and end effects, the EMD of the signal

$$y(t) = \cos\big((15 + 4.5\cos(1.5\pi))2\pi t\big) + \big(1.7 - 0.7\cos(0.4\pi t)\big)\cos(8\pi t) \tag{1.6}$$

for $t \in [0, 1.2]$ is shown in Fig. 1.4. This simulated example has been constructed with the consideration that in the middle of the observation window, the instantaneous frequencies of the two components are close. From Fig. 1.4, mode mixing is clearly observed around the time $t = 0.5$ when comparing IMFs 1 and 2. Moreover, end effects can be observed when considering the residual at $t > 1.1$.

In this thesis, alternatives to the EMD are investigated in an attempt to alleviate the end effect and mode mixing issues.

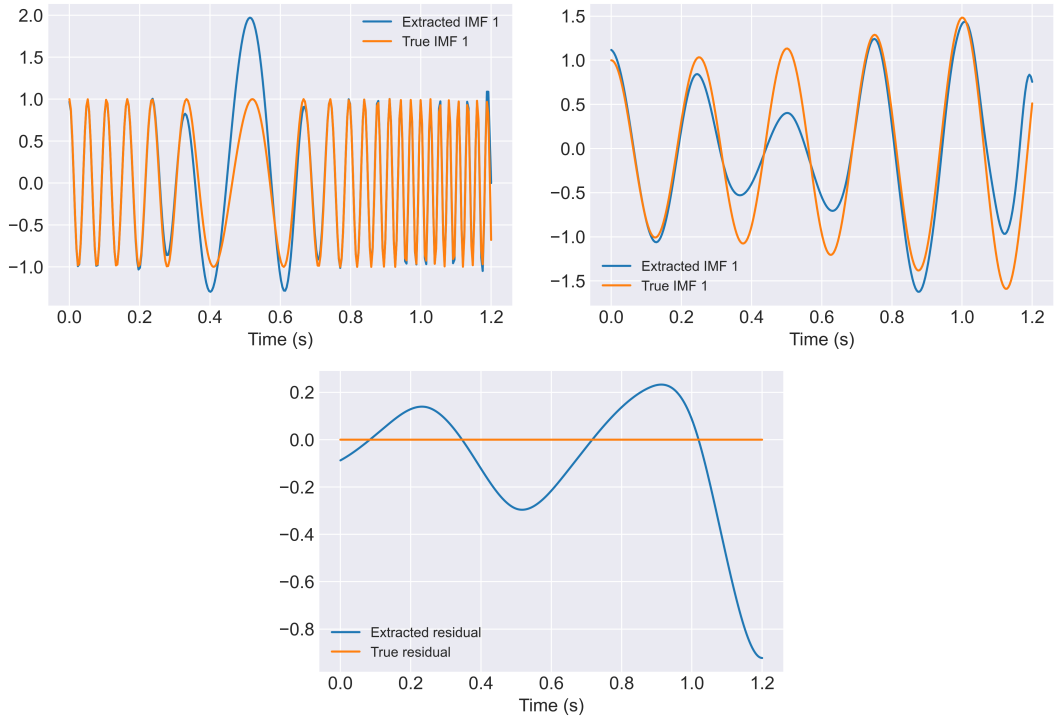


Figure 1.4: Example of mode mixing and end effects when using the EMD on the signal in Eq. (1.6).

1.3.2 Online Setup

Due to difficulties with adaptive data analysis in an online setup, typically in the literature the entire time series $\{P(t)\}_{t=1}^{n}$ is used for the decomposition as shown in Fig. 1.3. The decomposition is as such made with the assumption that the entire dataset is observed before making forecasts.

In practice, the entire dataset is not known beforehand. Hence, at each time where a forecast is to be computed, only the previous data can be used for the decomposition. This introduces both a computational challenge as well as a challenge to avoid compromising the quality of the decomposition method with the loss of information. The computational challenge can be addressed by introducing a windowing operation before the decomposition such that the decomposition of the time series is only based on the previous $q \geq p$ samples. This implies replacing the input $\{P(t)\}_{t=1}^{n}$ in Fig. 1.3 by the vector $\boldsymbol{P}_t^q = \big(P(t), P(t-1), \ldots, P(t-q+1)\big)^T$ for $q > 0$ and $t \geq q$ [Qia+19]. As for the quality of the decomposition, using a fraction of the entire data has a negative influence on the decomposition, particularly with regards to the low frequency components of the signal. Moreover, when using the EMD, end effects further degrades the decomposition at the boundary and the different decompositions computed at each step can produce inconsistent results due to mode mixing. These defects are detrimental to subsequent forecasting [SPC20].

To enable the use of decomposition based models for an online setup, an adaptive data analysis method which alleviates the mode mixing and end effects issues is needed. Adaptive decomposition of signal is an active research area with many promising developments towards alleviating the aforementioned issues of the EMD [DPB21; SECB22]. Investigating the recent developments in this field is a central focus of this thesis with the purpose of developing a novel hybrid decomposition based model applicable to online forecasting of wind power production.

1.4 Problem Statement

The preceding discussion leads to the following problem statement.

What is the effect of applying adaptive decomposition based models for online forecasting of wind power production compared to a purely deep neural network based model?

The rest of the thesis is structured as follows. Different adaptive decomposition methods are investigated in Chapter 2 facilitating a discussion regarding which methods to pursue further. In Chapter 3, the theory of compressive sensing is introduced in the context of time-frequency dictionaries leading to a description of an adaptive decomposition method. This is followed by theoretical results concerning uniqueness of the compressive sensing based adaptive decomposition methods in Chapter 4. Then in Chapter 5, theory regarding a specific type of partial differential equation, i.e. the heat equation, is introduced in order to present a partial differential equation based adaptive decomposition method. Having introduced the theoretical side of the adaptive decomposition methods, the experimental part of the thesis follows. This includes a description of the experimental setup in Chapter 6 followed by numerical experiments in Chapter 7. Finally, in Chapter 8, the thesis is concluded and directions to pursue to further the work are discussed in Chapter 9.

2. Adaptive Decomposition Methods

Spectral decomposition is a mathematically well established and powerful tool for analysis of signals. A spectral decomposition method of particular importance is the Fourier transform which yields the Fourier spectrum of a given signal. The Fourier transform is a linear transformation well suited for stationary signals. However, most signals stemming from natural phenomena are not stationary nor linear. To accommodate the non-stationarity, methods such as the short-time Fourier transform and the wavelet transform have been developed both empirically and mathematically. These methods are linear and well suited for piecewise stationary signals yielding a time-frequency representation. However, inherently these integral transform methods are limited by Heisenberg's uncertainty principle as a result of the convolution time-frequency duality [Fol92, pp. 232-233], imposing a bound on the possible time and frequency resolution. This presents a practical deficit in the analysis of non-linear and non-stationary signals since these signals often have a frequency representation which changes rapidly over time [HS14, p. 3].

In 1946, a definition of the concepts of instantaneous frequency (IF) and instantaneous amplitude (IA) was proposed by Gabor using the Hilbert transform [Gab46]. The key idea was to complexify the signal using the Hilbert transform by defining the so-called analytic signal $y_A(t) = y(t) + j\mathcal{H}\{y\}(t)$ where $y(t)$ is the observed signal and $\mathcal{H}$ is the Hilbert transform. For definitions of analytic signals and the Hilbert transform see Appendix A.1. Rewriting the complex-valued function $y_A(t)$ in polar form as $y_A(t) = a(t)\exp(j\theta(t))$ provides a definition of IF as $\omega(t) = \theta'(t)$ and IA as $a(t)$. These definitions, however, are rather restrictive since they only allow for a single frequency at any given time t. In 1998, the empirical mode decomposition (EMD) was introduced as an algorithm to decompose a signal into so-called intrinsic mode functions (IMFs) which, from an intuitive interpretation, mimics some properties of these single frequency modes $a(t)\exp(j\theta(t))$ which we also refer to as mono-components or locally narrowband amplitude-modulated and frequency-modulated (AM-FM) signals [Hua+98]. Combining this decomposition method with the Hilbert transform gave rise to the time-frequency representation method called the Hilbert-Huang transform (HHT) [HS14, ch. 1].

The focus of this thesis is an application of adaptive decomposition methods to forecasting and while the EMD has empirically shown to be a powerful tool for forecasting, several weaknesses with the classical EMD algorithm have been emphasised such as mode mixing, end effects, and a lack of mathematical foundation [HS14, ch. 1]. In light of these weaknesses, methods for adaptively decomposing a signal into a superposition of mono-components have been an active research area. Empirically, some of the developed methods have been applied successfully in forecasting applications, for instance forecasting wind power production, however many remain mostly untested [DPB21; Bok+19; Qia+19].

The remainder of the chapter is structured as follows. In Section 2.1, the problem is posed in a strict sense by discussing how the concept of mono-components can be posed in a mathematically well defined setting and in Section 2.2, the EMD algorithm is explained. Finally, in Section 2.3, existing methods are categorised and merits of the individual methods are discussed, facilitating a choice regarding which methods to pursue further.

2.1 The Adaptive Data Analysis Setup

Let $y(t)$ be a real-valued continuous-time signal observed on a finite interval $[0, T]$ and assume that it is a realisation of a non-linear and non-stationary stochastic process. The purpose is to decompose this signal into multi-scale features such that

$$y(t) = \sum_{k=1}^{s} c_k(t) + r(t) \tag{2.1}$$

for $t \in [0, T]$ where s is the number of AM-FM signals, $c_k(t) = a_k(t)\cos(\theta_k(t))$ are AM-FM signals, and r is the residual representing the trend and/or noise. To derive the time-frequency representation of this decomposition, the analytic signal approach is used to determine $a_k(t)$ and $\omega_k(t) = \theta_k'(t)$. From this demodulation, the signal can be represented as

$$y(t) = \mathrm{Re}\Big\{ \sum_{k=1}^{s} a_k(t) \exp\Big(j \int \omega_k(t) dt \Big) \Big\} + r(t). \tag{2.2}$$

This clarifies the interpretation of the HHT as a generalised Fourier expansion with Fourier coefficients and frequencies that are functions of time [HS14, p. 13]. With Eq. (2.2) in mind, the Hilbert-Huang spectrum is defined as

$$\mathrm{HHS}\{y\}(t, \omega) = \begin{cases} a_k(t), & \omega = \omega_k(t), \\ 0, & \text{otherwise}, \end{cases}$$

thereby providing a spectral analysis tool capable of accommodating non-stationary and non-linear signals. [HS14, pp. 12-14]

It is clear that the decomposition in Eq. (2.1) is not unique and can provide physically nonsensical decompositions. This motivates imposing restrictions on the functions a_k and θ_k. In [Hua+98], an operational definition is given by introducing the concept of IMFs. The IMFs are defined using envelopes, i.e. smooth curves outlining the extremes as stated below.

Definition 2.1 (Intrinsic Mode Function)
Let $c(t)$ be a signal observed for $t = 1, \ldots, n$ with n_e extrema and z_c zero crossings and let it be bounded by an upper envelope, $e_u(t)$, and a lower envelope, $e_l(t)$, defined by a smooth interpolation of the local maxima and minima of $c(t)$, respectively. Then $c(t)$ is an IMF if

$$|n_e - z_c| \leq 1$$

and if for $t = 1, \ldots, n$

$$m(t) = \Big|\frac{e_u(t) + e_l(t)}{2}\Big| = 0$$

where $m(t)$ is called the local mean. [Hua+98]

Subsequently, other classes of functions have been defined in a pursuit to develop a mathematically well defined class of mono-components. The most basic assumptions placed on the IA and the IF are that $a(t) \geq 0$ and $\omega(t) \geq 0$, respectively. These assumptions are based on providing a physically meaningful decomposition. In [Qia05; Qia06; Qia+09], a definition of mono-components is given based on these assumptions as well as an assumption related to unique modulation.

Definition 2.2 (Mono-component)
A signal $c(t) = a(t)\cos(\theta(t))$ is a mono-component if

- The IA $a : \mathbb{R} \to \mathbb{R}$ is non-negative.
- The instantaneous phase $\theta : \mathbb{R} \to \mathbb{R}$ is differentiable with non-negative IF $\omega = \theta' : \mathbb{R} \to \mathbb{R}$.
- The IA a and instantaneous phase θ are such that $\mathcal{H}\{a\cos(\theta)\}(t) = a(t)\sin(\theta(t))$.

The identity $\mathcal{H}\{a\cos(\theta)\}(t) = a(t)\sin(\theta(t))$ is motivated by the demodulation using the analytic signal since if the identity holds, then

$$c_A(t) = c(t) + j\mathcal{H}\{c\}(t) = a(t)\big(\cos(\theta(t)) + j\sin(\theta(t))\big) = a(t)\exp(j\theta(t)).$$

As such, this property ensures a uniquely determined modulation. To fulfil the identity $\mathcal{H}\{a\cos(\theta)\}(t) = a(t)\sin(\theta(t))$ it is sufficient that the Bedrosian identity $\mathcal{H}\{a\cos(\theta)\}(t) = a(t)\mathcal{H}\{\cos(\theta)\}(t)$ is fulfilled, and that $\mathcal{H}\{\cos(\theta)\}(t) = \sin(\theta(t))$. To fulfil the Bedrosian identity, it is required that the IA is band-limited to frequencies below the frequency domain support of the carrier wave $\cos(\theta(t))$ [Bed63]. The criterion $\mathcal{H}\{\cos(\theta)\}(t) = \sin(\theta(t))$ further provides a productive way of constructing mono-components. The Bedrosian identity in its most general form can be found in Theorem A.5.

Many other definitions of mono-components exist in the literature such as Fourier intrinsic band functions [Pus+17], δ-IMF [DAP13], ε-mono-components [HYY15; Hua+17], AM-FM$_\sigma$ [Guo+16], intrinsic mode type function [DLW11], scale separated signals [LSH17], see also [HS11; HHY17]. Similarities in the requirements among these different definitions include a non-negative IA and IF which both varies more slowly or is smoother than the carrier wave.

Additionally, assumptions can be imposed on the decomposition as a whole, e.g. orthogonality of mono-components [Hua+17; Pus+17; HK13; Sin+15] in order for the decomposition to preserve the energy in the signal and thereby providing a meaningful decomposition without redundancy, and separation of modes [LSH17; DLW11; Hua+17] in order to avoid overlap of the IF and thereby avoid mode mixing.

2.2 Empirical Mode Decomposition

In 1998, the adaptive decomposition method called the EMD was introduced [Hua+98]. In the EMD method, the procedure which is used to extract the IMFs is the sifting procedure (SP). Consider a signal $y(t)$. The EMD method works by identifying all the extrema of the signal and then using these extrema to fit an upper and a lower envelope by cubic spline interpolation [dBo78, pp. 39-45]. These envelopes are denoted $e_u(t)$ and $e_l(t)$, respectively. With the envelopes, a local mean is defined as

$$m_{1,0}(t) = \frac{e_u(t) + e_l(t)}{2}$$

and then the first potential IMF is found as

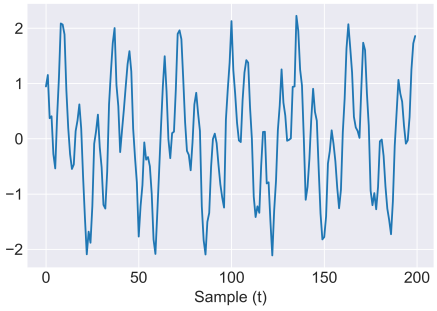
$$h_{1,1}(t) = y(t) - m_{1,0}(t).$$

This is the principle behind the SP in the EMD and is depicted in Fig. 2.1 where the signal in consideration is

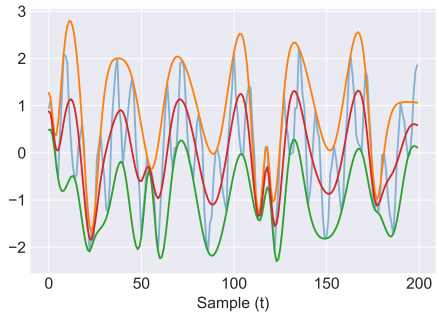
$$y(t) = \sin\left(t\frac{4\pi}{199}\right) + \cos\left(t\frac{14\pi}{199}\right) + 0.2\varepsilon(t)$$

with $\varepsilon_t \overset{iid.}{\sim} \mathcal{N}(0,1)$ for $t = 0, \ldots, 199$.

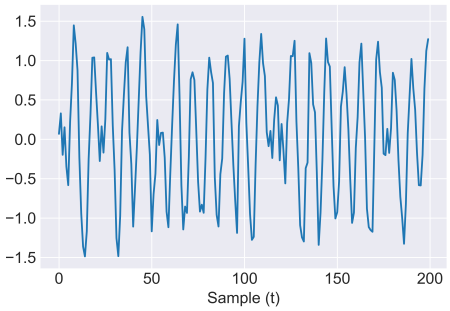
Ideally, $h_{1,1}(t)$ should be an IMF, however, this is often not the case due to overshoots and undershoots of the envelopes compared to the signal which can result



(a) Original signal $y(t)$.



(b) Upper envelope $e_u(t)$ (orange). Lower envelope $e_l(t)$ (green). Local mean $m_{1,0}(t)$ (red).



(c) The first potential IMF $h_{1,1}(t)$.

Figure 2.1: Depiction of one iteration of the SP.

in new extrema, shifts, or exaggerate existing extrema. Thus, it is likely that $h_{1,1}(t)$ is not an IMF and if this is the case, then the process is repeated with $h_{1,1}(t)$ treated as the signal. This process is repeated $j+1$ times until $h_{1,j+1}(t)$ is obtained as

$$h_{1,j+1}(t) = h_{1,j}(t) - m_{1,j}(t)$$

which is an IMF and then $h_{1,j+1}(t)$ is denoted as $c_1(t)$. Next the first residual $r_1(t)$ is computed as

$$r_1(t) = y(t) - c_1(t).$$

After the first iteration of the SP, $r_1(t)$ is treated as the new data and the SP is applied to $r_1(t)$ to obtain the second IMF $c_2(t)$ and a new residual $r_2(t)$ is then determined as

$$r_2(t) = r_1(t) - c_2(t).$$

This process is continued until iteration $s \in \mathbb{N}$ when a stopping criteria is met. Then the signal has been decomposed as in Eq. (2.1). [Hua+98, pp. 917-923]

The SP is summarised in Algorithm 1.

Algorithm 1 The Sifting Procedure

Input: Signal $y(t)$, sifting procedure stopping criteria.
1: Initialise $k = 0$; $h_k(t) = y(t)$.
2: **while** sifting stopping criteria **are** False **do**
3: Determine the local mean $m_k(t)$ of $h_k(t)$.
4: Compute IMF candidate $h_{k+1}(t) = h_k(t) - m_k(t)$.
5: Update $k \leftarrow k + 1$.
6: **end while**
Output: Mono-component $c(t) = h_k(t)$ and residual $r(t) = y(t) - c(t)$.

The EMD method has the advantage of being simple and the only hyperparameters are the stopping criteria. However, the EMD suffers from some known problems such as end effects, mode mixing, and a lack of mathematical foundation. Therefore, other methods for adaptive data analysis are presented in the following section which try to alleviate some of these problems.

2.3 Adaptive Decomposition Methods

A plethora of adaptive decomposition methods for decomposing a multi-component signal into a superposition of mono-components as in Eq. (2.1) have been developed since the introduction of the EMD. The main part of the EMD algorithm is the SP, and in many cases the SP still forms the foundation for the methods. The key step in the SP which allows for different approaches is the determination of the local mean. Some sifting based approaches follow this procedure by using different types of interpolation techniques, while other methods skip the step of finding local extrema and instead develop a procedure for directly determining the local mean.

Different criteria for classification of existing methods can be considered depending on the objective of the comparison. Based on the preceding discussion, one way of categorising the methods is depending on whether the method uses the SP or not. An entirely different criterion to use would be to categorise the methods by considering from which mathematical point-of-view the algorithms are derived.

Initially, we consider a categorisation which can broadly be used to distinguish between methods which have potential in terms of alleviating end effects and methods which inherently struggle with end effects. Consider the categorisation into sifting based methods and non-sifting based methods. The sifting category can be further sub-categorised into interpolation based methods, i.e. methods finding local extrema and then interpolating the extrema to find envelopes, and interpolation free methods, i.e. methods finding the local mean without using interpolation techniques. Inherently, the interpolation based methods have problems with end effects since extrema outside the observation window are unknown. This motivates us to not consider these methods in further detail. This includes the methods in [Smi05; DLN05; FO07; HK13; Yan+14; Li+18].

Considering interpolation free methods, we recognise 4 main classes of methods depending on the mathematical point-of-view of the algorithms. These are Fourier series based methods, operator based methods, partial differential equation (PDE) based methods, and compressive sensing (CS) based methods. The Fourier series and operator based methods are briefly mentioned in the following.

With regards to Fourier series based methods, in [Pus+17], a so-called Fourier decomposition method is introduced. Here the signal is decomposed into so-called Fourier intrinsic band functions based on a method derived using Fourier series. However, the method of [Pus+17] shows poor performance in terms of tone separation [Zho+22]. In [Zho+22], an expansion of this method is introduced, however, this method is non-causal and as such can only be applied in an offline setup. Additionally, in [HYY15], a method is introduced to extract ε-mono-components which combines the fields of Fourier theory and CS. However, [HYY15] provides little insight into the performance of the method and the practical implementation.

With the operator based methods, an operator is constructed and the idea is that the mono-components should be in the null space of the operator. Then the mono-components can be found using an algorithm called null space pursuit. Examples of these methods can be found in [Guo+16; PH08; HPH15; HPH13; PH10]. While some of the earlier operator based methods has issues with end effects and computational complexity, the method of [Guo+16] shows promise. However, due to time limitations this method has not been further investigated.

In the following, the PDE and CS based methods are introduced in further detail.

2.3.1 Partial Differential Equation Based Methods

The PDE based methods are a type of sifting based methods in which the local mean is determined by solving a linear PDE. In this section, the interpolation free PDE based methods are described. The PDE based methods have a stronger theoretical framework compared to the EMD. Additionally, by avoiding interpolation using the local extrema, the PDE based methods can avoid end effects to some extent [SECB22].

In [DAB09; DAB10; DAP13], they replace the local mean by an operator

$$m_\delta\{y\}(x) = \frac{1}{2}[\sup_{|\Delta|<\delta} y(x+\Delta) + \inf_{|\Delta|<\delta} y(x+\Delta)] \tag{2.3}$$

for $x \in \Omega \subset \mathbb{R}$ where $y : \Omega \to \mathbb{R}$ and $\delta > 0$. It is proven that the operator $m_\delta\{y\}(x)$ can be represented by a differential equation and from this a PDE is posed, the solution of which is a so-called δ-IMF. The PDE which is solved is

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{1}{\delta^2}u + \frac{1}{2}\frac{\partial^2 u}{\partial x^2} = 0, \\ u(x,0) = y(x) \end{cases}$$

where $x \in \Omega$, $u : \Omega \times \mathbb{R} \to \mathbb{R}$, $y(x)$ is the signal to be decomposed which is used as the initial condition, and the δ-IMF is $h_T(x) = u(x,T)$ for some $T > 0$. One of

the advantages of PDE based methods is that the end effects can be controlled by a proper choice of boundary conditions thereby forming an initial and boundary value problem. Furthermore, they show that the resulting PDE is a heat equation. This is used to show that the PDE has a unique solution and to prove numerical stability of the update scheme used to solve the PDE. Using this method, they obtain better results than the ones obtained using the classical EMD as they alleviate end effects and mode mixing.

In [WMV18], they propose a different heat equation than the one introduced in [DAP13]. They argue for this formulation by noting that the decomposition with the method of [DAP13] has three problems:

1. The parameter δ has to be fitted empirically and greatly influences the result.
2. Even if the frequency is extracted correctly there is no guarantee that the amplitude of the component is correct.
3. The heat equation of [DAP13] can sometimes have problems with instability of numerical methods for solving the PDE.

The method of [WMV18] is to solve the heat equation

$$\begin{cases} \frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \\ u(x,0) = y(x) \end{cases}$$

where $\alpha > 0$ and they define the local mean as $m(x) = u(x,T)$ for $T > 0$ chosen such that the PDE has converged. The heat equation is used to find the local mean due to properties ensuring a smooth solution which passes through the inflection points of the signal. The results of [WMV18] indicate that the method can alleviate the mode mixing problem of the EMD.

2.3.2 Compressive Sensing Based Methods

In the CS based methods, the decomposition of the signal y is formulated as a sparse optimisation problem where the solution is sought as a sparse decomposition into oscillatory components which belong to some dictionary $\mathcal{D}$. Hence, these methods are non-sifting based. The initial optimisation problem is formulated by considering a version of the ℓ^0 optimisation problem which is usually considered in CS with non-linear constraints. This optimisation problem is formulated as

$$\begin{aligned} &\text{Minimise} && s \\ &\text{Subject to} && y(t) = \sum_{k=1}^{s} a_k(t)\cos(\theta_k(t)) \\ & && a_k(t)\cos(\theta_k(t)) \in \mathcal{D}, \text{ for } k = 1, \ldots, s. \end{aligned} \tag{P0}$$

The Eq. (P0) optimisation problem is NP-hard and it is reformulated into a simpler problem which can be solved. The reformulated optimisation problem, the dictionary,

and the method used to solve the reformulated optimisation problem differ from article to article.

In [HS11], a method is introduced which searches for the sparsest representation of a multi-component signal where the IMF candidates are in the dictionary

$$\mathcal{D} = \{a(t)\cos(\theta(t)) : \frac{d\theta(t)}{dt} \geq 0,\ a(t) \text{ is smoother than } \cos(\theta(t))\}.$$

An ℓ^1 optimisation problem is formulated which uses the third order total variation. However, this optimisation problem is difficult and has to be solved recursively using an interior-point method. Through numerical examples, it is seen that the proposed method is less noise sensitive and reduces end effects compared to the classical EMD. However, the computational cost of using this method is relatively high. In [HS13b], the same dictionary is considered and a non-linear matching pursuit (NMP) algorithm is used to solve the optimisation problem. This method proves to be computationally efficient compared to that of [HS11], alleviates end effects, is more noise robust than ensemble EMD [WH09], and has a solid mathematical foundation as convergence and uniqueness of the decomposition have been proven [HST13; LSH15].

In [LSH17], recent developments following the method introduced in [HS13b] are given. The article introduces a two-level method consisting of a local algorithm and a global algorithm. The local algorithm is used to determine a piecewise constant phase function which is used for initialisation in the global algorithm and then the optimisation problem is approximately solved using the method introduced in [HS13b].

In [DSB19], methods using either orthogonal matching pursuit (OMP) or least angle regression are used to solve the CS optimisation problem in Eq. (P0). Using properties of OMP, convergence of the OMP algorithm is shown. Furthermore, they argue for convergence of the least angle regression method through convexity of the optimisation problem. The algorithms are tested both quantitatively and qualitatively on a variety of simulated signals and the results are compared to the classical EMD and the PDE based δ-EMD method introduced in [DAP13]. The results show good capability regarding the separation of the frequency components as well as tone separation, however in some of the tests amplitude attenuation occurs. Lastly, it is shown numerically that the algorithms are noise robust.

2.3.3 Summary

For the PDE based methods, the method introduced in [DAP13] seems to have a broader mathematical foundation compared to that of [WMV18]. However, as pointed out by [WMV18], the method has some shortcomings, specifically it is sensitive to the choice of δ, whereas the method of [WMV18] is less hyperparameter sensitive.

In terms of CS based methods, the method introduced in [HS11] is able to both alleviate end effects and is noise robust, however a more computationally effective method has been given in [HS13b] which generally shows better performance. Furthermore, the method of [LSH17] can be seen as an extension of the method in [HS13b]

which further improves performance but is more computationally complex. Finally, the method of [DSB19] also achieves good results, however, the method is hard to replicate as they do not disclose which dictionary they use.

Based on these considerations, the methods of [WMV18] and [HS13b] are decomposition methods of interest for this thesis. Introducing these methods in detail is the subject of Chapters 3 to 5.

3. Compressive Sensing with Time-Frequency Dictionaries

Compressive sensing refers to an area of mathematics which was originally motivated by applications in sampling theory. Collecting measurements via sensors that are sampling a physical process is common in many applications as is also the case with wind power production. In the classical Shannon sampling theory, the sample rate is required to be twice the bandwidth of the observed process. This can be rather restrictive and introduces an excessive sampling requirement for exact recovery of signals from the sampled data. However, it has been found that if the observed process is sparse with respect to some basis, then the sample rate can be lowered depending on the sparsity. [FR13, ch. 1]

In this chapter, the CS approach to adaptive time-frequency decomposition introduced in [HS13b] is presented. The underlying assumption is that the observed signal is sparse with respect to some dictionary consisting of potential mono-components defined via the so-called scale separation property which is formally introduced in Chapter 4. In this chapter, approximations of this dictionary are given based on Fourier bases in order to develop practical algorithms.

This chapter is structured as follows. In Section 3.1, the basic CS problem is discussed. Subsequently, in Section 3.2, the CS problem is related to time-frequency dictionaries and in Section 3.3, an algorithm called non-linear matching pursuit empirical mode decomposition (NMP-EMD) for adaptive data analysis is introduced. A special case of the algorithm is then considered in Section 3.4 leading to an efficient algorithm, called the fast Fourier transform non-linear matching pursuit empirical mode decomposition (FFT-NMP-EMD) algorithm. Then in Section 3.5, practical considerations concerning the introduced algorithms are given. Finally, in Sections 3.6 and 3.7, the progression of the algorithms are shown and the influence of a particularly important hyperparameter is analysed through synthetic examples.

3.1 Sparse Recovery from Redundant Dictionaries

For convenience, we define the notation $[m] = \{1, 2, \dots, m\}$ and the cardinality of a set S as $|S|$. Moreover, the support of a vector $\boldsymbol{x} = \big(x_1, \dots, x_m\big)^T \in \mathbb{R}^m$ is defined as $\operatorname{supp}(\boldsymbol{x}) = \{j \in [m] : x_j \neq 0\}$. The notion of sparsity of vectors is defined below.

Definition 3.1 (s-Sparse Vectors)
Let $\boldsymbol{x} \in \mathbb{R}^m$. If $\|\boldsymbol{x}\|_0 \leq s$ for $s \in \mathbb{N}$, then $\boldsymbol{x}$ is called an s-sparse vector. Moreover, let $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ where $\boldsymbol{y} \in \mathbb{R}^n$ is the observed data and $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ for $n < m$ is called a dictionary. If $\boldsymbol{x}$ is s-sparse, then $\boldsymbol{y}$ is s-sparse with respect to the dictionary $\boldsymbol{A}$. [FR13, p. 41]

The columns of $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ are denoted $(\boldsymbol{A})_j = \boldsymbol{a}_j$ for $j = 1, \dots, m$ and are also referred to as atoms.

The basic CS problem is motivated from an underdetermined set of linear equations

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

where $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{n \times m}$, and $\boldsymbol{x} \in \mathbb{R}^m$ for $n < m$. In CS, $\boldsymbol{y}$ is interpreted as the observed data and it is assumed that $\boldsymbol{x}$ is an s-sparse vector such that the observed data $\boldsymbol{y}$ is s-sparse with respect to the dictionary $\boldsymbol{A}$. To recover $\boldsymbol{x}$ from the observed data $\boldsymbol{y}$, the sparsity assumption of $\boldsymbol{x}$ is used and an ℓ^0-optimisation problem is posed as

$$\begin{aligned} \underset{\boldsymbol{z} \in \mathbb{R}^m}{\text{Minimise}} \quad & \|\boldsymbol{z}\|_0 \\ \text{Subject to} \quad & \boldsymbol{y} = \boldsymbol{A}\boldsymbol{z}. \end{aligned} \tag{P0}$$

In the presence of noise, a slightly modified problem is posed as

$$\begin{aligned} \underset{\boldsymbol{z} \in \mathbb{R}^m}{\text{Minimise}} \quad & \|\boldsymbol{z}\|_0 \\ \text{Subject to} \quad & \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_2 \leq \delta \end{aligned} \tag{P0$_\delta$}$$

where δ determines the noise tolerance in the optimisation problem. This P0 problem is in general NP-hard which means that the optimisation problem needs to be relaxed and formulated differently to provide a constructive way of solving the problem approximately [FR13, ch. 1]. Existing methods include basis pursuit, in which a convex relaxation of the ℓ^0-optimisation problem is considered; greedy algorithms, in which it is assumed that the overall problem can be solved accurately by iteratively solving sub-problems; and thresholding based methods, in which a thresholding operation is used to impose sparsity [FR13, ch. 3].

In this thesis, a greedy algorithm is used. Greedy algorithms in CS seek to approximately solve the P0 problem by building an approximate solution by iteratively updating the support set of the solution. The update in the greedy algorithm is

based on determining the locally optimal choice at each iteration. As such, in this process, atoms which match the data the best are found one-by-one. Examples of greedy algorithms commonly used in CS are matching pursuit (MP) and OMP. [MZ94, p. 3399-3400][FR13, pp. 65-66]

3.2 Sparse Time-Frequency Decomposition

Combining the ideas of the EMD with that of CS, an adaptive data analysis method can be developed. Consider an observed signal assumed to be non-linear, non-stationary, and with low predictability. Extraction of patterns from the observed signal motivates decompositions over large and redundant dictionaries of time-frequency atoms. Due to the adaptive nature of the data, it is required to adaptively learn the dictionary from the data. [MZ94]

In [HS13b], a method for this purpose is introduced. The method works by searching for the sparest representation of a signal over the largest possible dictionary consisting of IMFs. The method is said to be adaptive due to the fact that a large and highly redundant basis is used to obtain the sparsest decomposition.

The dictionary over which this problem is solved can be defined as follows

$$\mathcal{D} = \{a(t)\cos\big(\theta(t)\big) : \frac{d\theta(t)}{dt} \geq 0, a(t) \in V(\theta;\lambda), \frac{d\theta(t)}{dt} \in V(\theta;\lambda)\} \tag{3.1}$$

where $V(\theta;\lambda)$ is a set consisting of functions which are smoother than $\cos\big(\theta(t)\big)$ with smoothness parameter λ. Based on the approach in [HS13b], given a $\theta(t)$ then $V(\theta;\lambda)$ is constructed as the linear span of an over-complete Fourier basis defined as

$$V(\theta;\lambda) = \text{span}\{V_b(\theta;\lambda)\} \tag{3.2}$$

for

$$V_b(\theta;\lambda) = \big\{1, \cos\Big(\frac{k\theta}{\rho L_\theta}\Big), \sin\Big(\frac{k\theta}{\rho L_\theta}\Big) : k = 1, 2, \ldots, \lfloor \rho\lambda L_\theta \rfloor\big\} \tag{3.3}$$

where $L_\theta = \frac{\theta(T)-\theta(0)}{2\pi}$ for a signal observed in the interval $[0, T]$, $\rho > 1$ controls the over-completeness, and the parameter $\lambda \leq 1/2$ controls the smoothness of $V(\theta;\lambda)$. An over-complete representation is used in order to obtain a large dictionary which in turn gives the possibility for a sparse decomposition. Note that for $\rho = 1$, the basis reduces to the standard Fourier basis and an algorithm based on the fast Fourier transform (FFT) can be used. However, abiding with traditional Fourier theory, the algorithm would assume periodicity of the signal which is not an assumption we can make in practice when considering wind power production data [HS13b]. Due to the reduced computational complexity, the fast algorithm is still considered. This algorithm is introduced in Section 3.4.

In contrast to the traditional CS problem, the dictionary in this case is infinite dimensional and thus the problem cannot be posed using a matrix transformation. However, the infinite number of functions in the dictionary can be defined as an infinite set. Let an index set for the functions in the set $\mathcal{D}$ be given as

$$I = \{k \in \mathbb{N} : a_k(t)\cos\big(\theta_k(t)\big) \in \mathcal{D}\}.$$

In this context, each index $k \in \mathbb{N}$ relates to a function $a_k(t)\cos\big(\theta_k(t)\big)$ in the dictionary $\mathcal{D}$. The dictionary $\mathcal{D}$ can then be expressed as

$$\mathcal{D} = \{a_k(t)\cos\big(\theta_k(t)\big)\}_{k\in I} = \{c_k(t)\}_{k\in I}$$

where $c_k(t) = a_k(t)\cos\big(\theta_k(t)\big)$. Assuming a signal $y(t)$ can be perfectly represented by the dictionary $\mathcal{D}$, then the signal $y(t)$ can be decomposed as

$$y(t) = \sum_{k\in I} \alpha_k c_k(t) \tag{3.4}$$

for $\alpha \in \{0,1\}$ and where $\sum_{k\in I} \mathbb{1}[\alpha_k \neq 0] \leq s$ for $s \in \mathbb{N}$, i.e. $y(t)$ is s-sparse with respect to $\mathcal{D}$. The goal is as mentioned to find the sparsest decomposition and this can be done by solving the following optimisation problem

$$\begin{aligned} &\text{Minimise} && s \\ &\text{Subject to} && y(t) = \sum_{k=1}^{s} a_k(t)\cos\big(\theta_k(t)\big), \\ &&& a_k(t)\cos\big(\theta_k(t)\big) \in \mathcal{D}, \text{ for } k = 1,\dots,s. \end{aligned} \tag{P}$$

In the presence of noise, the signal can be decomposed as

$$y(t) = \sum_{k\in I} \alpha_k c_k(t) + r(t)$$

where $r(t)$ is the residual which represents the noise. In this case, the problem is reformulated as follows

$$\begin{aligned} &\text{Minimise} && s \\ &\text{Subject to} && \|y(t) - \sum_{k=1}^{s} a_k(t)\cos\big(\theta_k(t)\big)\|_2 \leq \delta, \\ &&& a_k(t)\cos\big(\theta_k(t)\big) \in \mathcal{D}, \text{ for } k = 1,\dots,s \end{aligned} \tag{P$_\delta$}$$

where δ is dependent on the energy in the residual. The problems Eqs. (P) and (P$_\delta$) are non-linear L^0 minimisation problems, similarly to Eqs. (P0) and (P0$_\delta$), and are known to be NP-hard to solve [FR13, pp. 53-56]. Therefore, following the CS theory, the optimisation problem is relaxed. In this thesis, the L^1-regularised NMP method for solving the problem, introduced in [HS13b], is presented.

Using the NMP method, the first component is extracted by determining the function $a(t)\cos(\theta(t)) \in \mathcal{D}$ which matches $y(t)$ the best with respect to the L^2-norm. This is done iteratively, as to solve the problem by solving a series of sub-problems. At iteration k, these considerations imply the following optimisation problem

$$\begin{aligned} &\text{Minimise} && \|r_k(t) - a_k(t)\cos(\theta_k(t))\|_2^2 \\ &\text{Subject to} && a_k(t)\cos(\theta_k(t)) \in \mathcal{D} \end{aligned} \tag{$\mathrm{P_{NMP}}$}$$

where $r_k(t) = y(t) - \sum_{j=1}^{k-1} a_j(t)\cos\big(\theta_j(t)\big)$. This is referred to as an MP algorithm since at each iteration a search for the atom in the dictionary which matches the signal the best is made. However, the problem is non-linear since $\theta_k(t)$ is introduced in the objective function non-linearly which is the case as $\cos(\cdot)$ is a non-linear function. [HS13b]

3.3 NMP-EMD

In this section, the NMP-EMD algorithm to approximately solve Eq. ($\mathrm{P_{NMP}}$) is introduced. Initially, some considerations are needed to motivate the steps in the algorithm. This includes posing the optimisation problem in a way which results in a series of L^1-regularised least square problems while imposing the constraints $a_k(t)\cos(\theta_k(t)) \in \mathcal{D}$.

Firstly, the constraint that $a_k \in V(\theta_k;\lambda)$ is imposed by parameterising a_k directly in the $V(\theta_k;\lambda)$-space as

$$a_k(t) = \hat{a}_{k,0} + \sum_{\kappa=1}^{\lfloor \rho\lambda L_{\theta_k} \rfloor} \hat{a}_{k,\kappa}\cos\left(\frac{\kappa\theta(t)}{\rho L_{\theta_k}}\right) + \sum_{\kappa=1}^{\lfloor \rho\lambda L_{\theta_k} \rfloor} \hat{a}_{k,\lfloor \rho\lambda L_{\theta_k} \rfloor+\kappa}\sin\left(\frac{\kappa\theta(t)}{\rho L_{\theta_k}}\right) \tag{3.5}$$

where $L_{\theta_k} = \frac{\theta_k(T)-\theta_k(0)}{2\pi}$. This yields a representation of $a_k(t)$ in the $V(\theta_k;\lambda)$-space given as $\hat{\boldsymbol{a}}_k = \big(\hat{a}_{k,0},\ldots,\hat{a}_{k,2\lfloor \rho\lambda L_{\theta_k} \rfloor}\big)^T$. The minimisation is then done with respect to the parameter vector $\hat{\boldsymbol{a}}_k$. For notational convenience, Eq. (3.5) is also expressed as

$$a_k(t) = \sum_{\kappa=0}^{2\lfloor \rho\lambda L_{\theta_k} \rfloor} \hat{a}_{k,\kappa} v_\kappa(t) \tag{3.6}$$

where $v_\kappa(t)$ are the functions in the basis $V_b(\theta_k;\lambda)$ which has been defined in Eq. (3.3).

Secondly, an inner iteration is introduced. The algorithm inherently has an outer iteration where a component $a_k\cos\big(\theta_k(t)\big)$ is found in each iteration. For each outer iteration, a number of inner iterations are introduced not unlike the SP in the classical EMD. This inner iteration consists of steps to update IMF candidates. These steps iteratively update the amplitude function and the phase function, respectively. The candidate amplitude function and the candidate phase function in the kth outer

iteration and the lth inner iteration are denoted a_k^l and θ_k^l, respectively. In order to provide a method for these updates, the trigonometric identity

$$a_k^{l+1}\cos\left(\theta_k^l\right) + b_k^{l+1}\sin\left(\theta_k^l\right) = \mathcal{A}_k^{l+1}\cos\left(\theta_k^l - \phi_k^{l+1}\right) \tag{3.7}$$

where $\mathcal{A}_k^{l+1} = \sqrt{\left(a_k^{l+1}\right)^2 + \left(b_k^{l+1}\right)^2}$ and $\phi_k^{l+1} = \arctan\left(\frac{b_k^{l+1}}{a_k^{l+1}}\right)$ is used [Apo67, p. 334]. Consider then the optimisation problem

$$\begin{aligned}\underset{\hat{\boldsymbol{a}}_k^{l+1},\, \hat{\boldsymbol{b}}_k^{l+1}}{\text{Minimise}} \quad & \left\| r_k(t) - a_k^{l+1}(t)\cos\left(\theta_k^l(t)\right) - b_k^{l+1}(t)\sin\left(\theta_k^l(t)\right)\right\|_2^2 \\ & + \gamma\left(\left\|\hat{\boldsymbol{a}}_k^{l+1}\right\|_1 + \left\|\hat{\boldsymbol{b}}_k^{l+1}\right\|_1\right)\end{aligned} \tag{P2}$$

where $\gamma \geq 0$ is a regularisation parameter and $\hat{\boldsymbol{a}}_k^{l+1}, \hat{\boldsymbol{b}}_k^{l+1}$ are the representations of $a_k^{l+1}(t),\ b_k^{l+1}(t)$ in $V(\theta_k^l;\lambda)$, respectively. With the identity in Eq. (3.7) in mind, if the optimisation problem Eq. (P2) is solved given θ_k^l and Eq. (3.5) is used to determine a_k^{l+1} and b_k^{l+1}, then the update for the phase function is $\theta_k^{l+1} = \theta_k^l - \arctan\left(\frac{b_k^{l+1}}{a_k^{l+1}}\right)$. In the optimisation problem Eq. (P2), an L^1-regularisation term has been added since this tends to stabilise the least square problem using an over-complete Fourier basis and it also tends to produce a sparse decomposition. [HS13b]

Considering the constraint $a_k\cos(\theta_k) \in \mathcal{D}$ in Eq. ($\mathrm{P_{NMP}}$), the requirement $a_k \in V(\theta_k;\lambda)$ has been enforced, however the constraints $\omega_k = \theta_k' \geq 0$ and $\omega_k \in V(\theta_k;\lambda)$ still need to be imposed. In the implementation, this is done by updating the IF rather than the phase function. Hence, using Eq. (3.7) the update for the IF is derived by differentiation

$$\begin{aligned}\Delta\omega_k^{l+1} &= \left(\arctan\left(\frac{b_k^{l+1}}{a_k^{l+1}}\right)\right)' \\ &= \frac{\left(b_k^{l+1}\right)' a_k^{l+1} - b_k^{l+1}\left(a_k^{l+1}\right)'}{(a_k^{l+1})^2}\,\frac{1}{1+\left(\frac{b_k^{l+1}}{a_k^{l+1}}\right)^2} \\ &= \frac{\left(b_k^{l+1}\right)' a_k^{l+1} - b_k^{l+1}\left(a_k^{l+1}\right)'}{(a_k^{l+1})^2 + (b_k^{l+1})^2}.\end{aligned} \tag{3.8}$$

The update is then

$$\omega_k^{l+1} = \omega_k^l - \Delta\omega_k^{l+1}.$$

To enforce the criteria $\omega_k \in V(\theta_k;\lambda)$, a low-pass filter $\chi_{V(\theta_k^l;\lambda)}$ is used to limit the frequency domain support of $\Delta\omega_k^{l+1}$ to that of $V(\theta_k^l;\lambda)$. This motivates re-defining the IF update step as [HS13a; HS13c]

$$\Delta\omega_k^{l+1} = \chi_{V(\theta_k^l;\lambda)}\Big\{\frac{\left(b_k^{l+1}\right)' a_k^{l+1} - b_k^{l+1}\left(a_k^{l+1}\right)'}{(a_k^{l+1})^2 + (b_k^{l+1})^2}\Big\}.$$

Additionally, to enforce the criteria $\omega_k \geq 0$, a step size η_k^{l+1} is determined as

$$\eta_k^{l+1} = \max\left\{\mu \in [0,1] : \omega_k^l - \mu\Delta\omega_k^{l+1} \geq 0\right\} \tag{3.9}$$

and the IF is updated as [HS13b]

$$\omega_k^{l+1} = \omega_k^l - \eta_k^{l+1}\Delta\omega_k^{l+1}. \tag{3.10}$$

The update for the instantaneous phase can be recovered from the update for the IF by integration.

3.3.1 Discrete-Time Formulation

The optimisation problem in Eq. (P2) is an L^1-regularised least square problem. In the CS literature, this is also referred to as basis pursuit denoising (BPDN) [FR13, p. 18]. In the following, Eq. (P2) is formulated as a finite dimensional BPDN optimisation problem. Assume that the signal $y(t)$ is observed for a finite temporal interval $t \in [0,T]$ and consider an equidistant sampling of the signal in samples $t_i = (i-1)\Delta t$ for $i = 1,\dots,n$ where n is the number of samples and $\Delta t = \frac{T}{n-1}$ is the spacing between samples. The sampled signal is then collected in a vector $\boldsymbol{y} = \big(y(t_1),\dots,y(t_n)\big)^T$ and the procedure is repeated for all signals sampled in a discrete grid. Hence, we define the notation used in the kth outer iteration and lth inner iteration

$$\boldsymbol{r}_k = \begin{pmatrix} r_k(t_1) \\ \vdots \\ r_k(t_n) \end{pmatrix}, \quad \boldsymbol{a}_k^l = \begin{pmatrix} a_k^l(t_1) \\ \vdots \\ a_k^l(t_n) \end{pmatrix}, \quad \boldsymbol{b}_k^l = \begin{pmatrix} b_k^l(t_1) \\ \vdots \\ b_k^l(t_n) \end{pmatrix}, \quad \boldsymbol{\theta}_k^l = \begin{pmatrix} \theta_k^l(t_1) \\ \vdots \\ \theta_k^l(t_n) \end{pmatrix}.$$

Then by Eq. (3.6)

$$\begin{aligned} \boldsymbol{a}_k^l &= \sum_{\kappa=0}^{2\lfloor \rho\lambda L_{\theta_k^l} \rfloor} \hat{a}_{k,\kappa}^l \begin{pmatrix} v_\kappa(t_1) & \cdots & v_\kappa(t_n) \end{pmatrix}^T \\ &= \begin{pmatrix} v_0(t_1) & v_1(t_1) & \cdots & v_{2\lfloor \rho\lambda L_{\theta_k^l} \rfloor}(t_1) \\ v_0(t_2) & v_1(t_2) & \cdots & v_{2\lfloor \rho\lambda L_{\theta_k^l} \rfloor}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ v_0(t_n) & v_1(t_n) & \cdots & v_{2\lfloor \rho\lambda L_{\theta_k^l} \rfloor}(t_n) \end{pmatrix} \begin{pmatrix} \hat{a}_{k,0}^l \\ \hat{a}_{k,1}^l \\ \vdots \\ \hat{a}_{k,2\lfloor \rho\lambda L_{\theta_k^l} \rfloor}^l \end{pmatrix} \\ &= \boldsymbol{\mathcal{V}}_k^l \hat{\boldsymbol{a}}_k^l \end{aligned}$$

where $v_\kappa(t_i)$ for $i = 1,\dots,n$ are the samples of the basis functions. Equivalently, it follows that

$$\boldsymbol{b}_k^l = \boldsymbol{\mathcal{V}}_k^l \hat{\boldsymbol{b}}_k^l.$$

Finally, with the notation

$$\boldsymbol{x}_k^l = \begin{pmatrix} \hat{\boldsymbol{a}}_k^l \\ \hat{\boldsymbol{b}}_k^l \end{pmatrix}, \text{ and } \quad \boldsymbol{A}_k^l = \begin{pmatrix} \operatorname{diag}\big(\cos(\boldsymbol{\theta}_k^l)\big)\boldsymbol{\mathcal{V}}_k^l & \operatorname{diag}\big(\sin(\boldsymbol{\theta}_k^l)\big)\boldsymbol{\mathcal{V}}_k^l \end{pmatrix},$$

then Eq. (P2) can be formulated as [THS14]

$$\underset{\boldsymbol{z}_k^{l+1}}{\text{Minimise}} \qquad \frac{1}{2}\|\boldsymbol{r}_k - \boldsymbol{A}_k^l \boldsymbol{z}_k^{l+1}\|_2^2 + \gamma\|\boldsymbol{z}_k^{l+1}\|_1 \tag{P2}$$

with solution $\boldsymbol{x}_k^{l+1}$. This is an ℓ^1-regularised least square problem and methods for solving this convex optimisation problem are known [CDS98]. [HS13b]

3.4 FFT-NMP-EMD

In this section, the FFT-NMP-EMD algorithm is derived as a special case of the algorithm introduced in Section 3.3. This special case involves assuming periodicity of the observed signal, thus making the use of an over-complete dictionary obsolete. This implies that the over-completeness parameter $\rho = 1$ and the regularisation parameter $\gamma = 0$. In correspondence with Eq. (P2), the optimisation problem for the kth outer iteration and the lth inner iteration is in this case posed as

$$\begin{aligned} \underset{a_k^{l+1}, b_k^{l+1}}{\text{Minimise}} \quad & \|r_k(t) - a_k^{l+1}(t)\cos\left(\theta_k^l(t)\right) - b_k^{l+1}(t)\sin\left(\theta_k^l(t)\right)\|_2^2 \\ \text{Subject to} \quad & a_k^{l+1}(t), b_k^{l+1}(t) \in V(\theta_k^l; \lambda). \end{aligned} \tag{P2 Fast}$$

In the following, the continuous-time situation is considered to derive the solution to this optimisation problem. For this purpose, it is assumed that $r_k(t) \in L^2$. Subsequently, a discrete-time formulation is considered yielding an approximate solution to the problem. Herein, a description of how the algorithm can be applied in a practical setting is given.

Firstly, since the IF is non-negative, the phase function is monotonically increasing. This allows a reformulation of the optimisation problem Eq. (P2 Fast) by considering the functions in the θ-coordinate as

$$\begin{aligned} \underset{a_{k,\theta}^l, b_{k,\theta}^l}{\text{Minimise}} \quad & \|r_{k,\theta}(\theta_k^l) - a_{k,\theta}^{l+1}(\theta_k^l)\cos\left(\theta_k^l\right) - b_{k,\theta}^{l+1}(\theta_k^l)\sin\left(\theta_k^l\right)\|_2^2 \\ \text{Subject to} \quad & a_{k,\theta}^{l+1}(\theta_k^l), b_{k,\theta}^{l+1}(\theta_k^l) \in V(\theta_k^l; \lambda) \end{aligned} \tag{P2 Fast θ-domain}$$

where $r_{k,\theta}\big(\theta_k^l(t)\big) = r_k(t)$, $a_{k,\theta}^{l+1}\big(\theta_k^l(t)\big) = a_k^{l+1}(t)$, and $b_{k,\theta}^{l+1}\big(\theta_k^l(t)\big) = b_k^{l+1}(t)$. Temporarily, the constraints are disregarded and the minimum is found for the unconstrained optimisation problem. The solution is without error given as

$$\begin{aligned} \tilde{a}_{k,\theta}^{l+1}(\theta_k^l) &= r_{k,\theta}^{l+1}(\theta_k^l)\cos\left(\theta_k^l\right), \\ \tilde{b}_{k,\theta}^{l+1}(\theta_k^l) &= r_{k,\theta}(\theta_k^l)\sin\left(\theta_k^l\right). \end{aligned}$$

Using the fact that in the θ-coordinate $\cos\left(\theta_k^l\right)$, $\sin\left(\theta_k^l\right)$, and the functions in the basis $V_b(\theta_k^l; \lambda)$ are all simple Fourier modes, the constraint is applied to the solution. Using Plancherel's theorem, this can be done by applying a low-pass filter to $\tilde{a}_{k,\theta}^{l+1}$

and $\tilde{b}_{k,\theta}^{l+1}$ in the Fourier domain. For instance, the Fourier transform of $\tilde{a}_{k,\theta}^{l+1}$ can be constrained to be in the $V(\theta_k^l;\lambda)$-space by applying a low-pass filter $\chi_{V(\theta_k^l;\lambda)}$. In the frequency domain, this filter is applied as

$$\hat{a}_{k,\theta}^{l+1}(f) = \mathcal{F}_\theta\{\tilde{a}_{k,\theta}^{l+1}\}(f)\hat{h}_k^l(f) \tag{3.11}$$

where $\hat{h}_k^l$ is the frequency response of the filter $\chi_{V(\theta_k^l;\lambda)}$ and the Fourier transform in the θ-coordinate is defined as

$$\mathcal{F}_\theta\{\tilde{a}_{k,\theta}^{l+1}\}(f) = \int \tilde{a}_{k,\theta}^{l+1}(\theta_k^l)\exp\big(-j2\pi f\overline{\theta}_k^l\big)d\theta_k^l$$

for $\overline{\theta}_k^l(t) = \frac{\theta_k^l(t)-\theta_k^l(0)}{2\pi L_{\theta_k^l}}$ and $L_{\theta_k^l} = \frac{\theta_k^l(T)-\theta_k^l(0)}{2\pi}$. The function $\hat{a}_{k,\theta}^{l+1}$ is the function in $V(\theta_k^l;\lambda)$ which minimises the distance in L^2 to $\mathcal{F}_\theta\{\tilde{a}_{k,\theta}^{l+1}\}$. By Plancherel's theorem [Fol92, p. 222],

$$\begin{aligned}&\|r_{k,\theta}(\theta_k^l) - a_{k,\theta}^{l+1}(\theta_k^l)\cos\big(\theta_k^l\big) - b_{k,\theta}^{l+1}(\theta_k^l)\sin\big(\theta_k^l\big)\|_2^2\\ &\qquad = \|\mathcal{F}_\theta\big\{r_{k,\theta}(\theta_k^l) - a_{k,\theta}^{l+1}(\theta_k^l)\cos\big(\theta_k^l\big) - b_{k,\theta}^{l+1}(\theta_k^l)\sin\big(\theta_k^l\big)\big\}(f)\|_2^2.\end{aligned}$$

Exploiting this property, the optimisation problem Eq. (P2 Fast θ-domain) can instead be considered in the Fourier domain

$$\begin{aligned}&\underset{a_{k,\theta}^{l+1},b_{k+\theta}^{l+1}}{\text{Minimise}} \quad \|\mathcal{F}_\theta\big\{r_{k,\theta}(\theta_k^l)\big\} - \mathcal{F}_\theta\big\{a_{k,\theta}^{l+1}(\theta_k^l)\cos\big(\theta_k^l\big)\big\} - \mathcal{F}_\theta\big\{b_{k,\theta}^{l+1}(\theta_k^l)\sin\big(\theta_k^l\big)\big\}(f)\|_2^2\\ &\text{Subject to} \quad a_{k,\theta}^{l+1}(\theta_k^l), b_{k,\theta}^{l+1}(\theta_k^l) \in V(\theta_k^l;\lambda).\end{aligned}$$

By the definition of $\hat{h}_k^l$ and by Eq. (3.11), it follows that $\mathcal{F}_\theta^{-1}\{\hat{a}_{k,\theta}^{l+1}\}$ minimises Eq. (P2 Fast θ-domain). Similarly, $\mathcal{F}_\theta^{-1}\{\mathcal{F}_\theta\{\tilde{b}_{k,\theta}^{l+1}\}(f)\hat{h}_k^l(f)\}$ is the solution. Thereby, the solution to Eq. (P2 Fast) is given as

$$\begin{aligned}a_k^{l+1}(t) &= a_{k,\theta}^{l+1}(\theta_k^l(t)) = \mathcal{F}_\theta^{-1}\{\hat{a}_{k,\theta}^{l+1}(f)\}(\theta_k^l(t)),\\ b_k^{l+1}(t) &= b_{k,\theta}^{l+1}(\theta_k^l(t)) = \mathcal{F}_\theta^{-1}\{\hat{b}_{k,\theta}^{l+1}(f)\}(\theta_k^l(t)).\end{aligned}$$

Using the frequency shift property [Fol92, p. 214] of Fourier transforms of $r_{k,\theta}^{l+1}(\theta_k^l)\cos\big(\theta_k^l\big)$ and $r_{k,\theta}(\theta_k^l)\sin\big(\theta_k^l\big)$, this solution can alternatively be expressed as

$$a_k^{l+1}(t) = \mathcal{F}_\theta^{-1}\big\{\big(\hat{r}_{k,\theta}(f+L_{\theta_k^l}) + \hat{r}_{k,\theta}(f-L_{\theta_k^l})\big)\hat{h}_k^l(f)\big\}(\theta_k^l(t)), \tag{3.12}$$

$$b_k^{l+1}(t) = \mathcal{F}_\theta^{-1}\big\{j\big(\hat{r}_{k,\theta}(f+L_{\theta_k^l}) - \hat{r}_{k,\theta}(f-L_{\theta_k^l})\big)\hat{h}_k^l(f)\big\}(\theta_k^l(t)). \tag{3.13}$$

Thus, this provides a method of solving Eq. (P2 Fast) using the Fourier transform [HS13b]. In the following, the problem is posed in a discrete-time setting and an explicit algorithm employing the FFT is introduced.

3.4.1 Discrete-Time Formulation

Let $r_k(t_i)$ be an observed signal in discrete samples $t_i = (i-1)\Delta t$ for $i = 1, \dots, n$ where $\Delta t = \frac{T}{n-1}$ and let $\theta(t_i)$ for $i = 1, \dots, n$ be the sampled phase function. Define a normalised phase function as $\overline{\theta}_k^l(t_i) = \frac{\theta_k^l(t_i) - \theta_k^l(t_1)}{2\pi L_\theta}$ where $L_{\theta_k^l} = \frac{\theta_k^l(t_n) - \theta_k^l(t_1)}{2\pi}$. In line with Eq. (P2 Fast θ-domain), the signal is represented in the θ-coordinate. In order to apply the discrete Fourier transform and thereby the FFT, a uniform sampling of the signal in the θ-coordinate is needed. Let $\theta_{k,i}^l = 2\pi L_{\theta_k^l} t_i$ for $i = 1, \dots, n$ be the uniform mesh in the θ-coordinate, and define $r_{k,\theta}(\theta_{k,i}^l)$ as the ith sample in the interpolation of $r_{k,\theta}(\theta_k^l(t_i)) = r_k(t_i)$ to the uniform mesh in the θ-coordinate. Assuming n is an even number, compute the discrete Fourier transform of $r_{k,\theta}(\theta_{k,i}^l)$ as

$$\hat{r}_{k,\theta}(\kappa) = \sum_{i=1}^{n} r_{k,\theta}(\theta_{k,i}^l) \exp\big(-j2\pi\kappa\overline{\theta}_{k,i}^l\big)$$

for $\kappa = -\frac{n}{2} + 1, \dots, \frac{n}{2}$ where $\overline{\theta}_{k,i}^l = \frac{\theta_{k,i}^l - \theta_{k,1}^l}{2\pi L_{\theta_k^l}}$. Following the result in Eqs. (3.12) and (3.13), the discrete solution in the θ-coordinate is

$$\begin{aligned} a_{k,\theta}^{l+1}(\theta_{k,i}^l) &= \mathcal{F}_\theta^{-1}\big\{\big(\hat{r}_{k,\theta}(\kappa + \lfloor L_{\theta_k^l}\rfloor) + \hat{r}_{k,\theta}(\kappa - \lfloor L_{\theta_k^l}\rfloor)\big)\hat{h}_{k,\kappa}^l\big\}(\theta_{k,i}^l), \\ b_{k,\theta}^{l+1}(\theta_{k,i}^l) &= \mathcal{F}_\theta^{-1}\big\{j\big(\hat{r}_{k,\theta}(\kappa + \lfloor L_{\theta_k^l}\rfloor) - \hat{r}_{k,\theta}(\kappa - \lfloor L_{\theta_k^l}\rfloor)\big)\hat{h}_{k,\kappa}^l\big\}(\theta_{k,i}^l) \end{aligned}$$

for $i = 1, \dots, n$ where the low-pass filter is defined as

$$\hat{h}_{k,\kappa}^l = \begin{cases} 1, & |\kappa| < \lambda L_{\theta_k^l}, \\ 0, & \text{otherwise} \end{cases}$$

for $\kappa = -\frac{n}{2} + 1, \dots, \frac{n}{2}$ and the inverse discrete Fourier transform in the θ-coordinate is defined as

$$\mathcal{F}_\theta^{-1}\{\hat{r}_{k,\theta}\}(\theta_{k,i}^l) = \frac{1}{n} \sum_{\kappa=-\frac{n}{2}+1}^{\frac{n}{2}} \hat{r}_{k,\theta}(\kappa) \exp\big(j2\pi\kappa\overline{\theta}_{k,i}^l\big)$$

for $i = 1, \dots, n$. Finally, the solution in the time domain is the interpolation of $a_{k,\theta}^{l+1}(\theta_{k,i}^l)$ and $b_{k,\theta}^{l+1}(\theta_{k,i}^l)$ given in the uniform mesh in the θ-coordinate back to the physical grid $\theta_k^l(t_i)$, yielding the solution $a_k^{l+1}(t_i)$ and $b_k^{l+1}(t_i)$, respectively.

3.5 Practical Considerations

In this section, some practical considerations are given regarding efficiently solving Eq. (P2), choosing the over-completeness parameter ρ, the type of interpolation used in the FFT-NMP-EMD algorithm, specific design of the filter $\chi_{V(\theta_k^l;\lambda)}$, initialisation of the algorithms, stabilising computations, and determining the step size Eq. (3.9).

To conclude the section, an algorithm outlining the two methods introduced in this chapter is given.

The optimisation problem Eq. (P2) can be equivalently expressed as a perturbed linear program [CDS98, p. 51]

$$\begin{aligned} &\underset{\boldsymbol{\zeta}_k^{l+1},\, \boldsymbol{e}}{\text{Minimise}} && (\boldsymbol{h}_k^{l+1})^T \boldsymbol{\zeta}_k^{l+1} + \frac{1}{2}\|\boldsymbol{e}\|_2^2 \\ &\text{Subject to} && \boldsymbol{\Phi}_k^{l+1}\boldsymbol{\zeta}_k^{l+1} + \boldsymbol{e} = \boldsymbol{r}_k, \quad \boldsymbol{\zeta}_k^{l+1} \succcurlyeq 0 \end{aligned} \tag{P2 PLP}$$

where $\succcurlyeq$ denotes entry-wise inequality, $\boldsymbol{h}_k^{l+1} = \gamma \mathbf{1}_{4|V_b(\theta_k^l;\lambda)|}$, $\boldsymbol{\Phi}_k^{l+1} = \left(\boldsymbol{A}_k^{l+1}, -\boldsymbol{A}_k^{l+1}\right)$, $\boldsymbol{\zeta}_k^{l+1} = \left((\boldsymbol{u}_k^{l+1})^T, (\boldsymbol{v}_k^{l+1})^T\right)^T$, $\boldsymbol{e} \in \mathbb{R}^n$, and the solution is $\boldsymbol{x}_k^{l+1} = \boldsymbol{u}_k^{l+1} - \boldsymbol{v}_k^{l+1}$. This is a quadratic program with linear constraints, however it retains a structure similar to linear programming. Therefore, linear programming methods such as simplex type and interior-point type methods can be used to solve this optimisation problem [CDS98, p. 51]. In this thesis, an interior-point type method is used. Details regarding the method are outside the scope of this thesis and the implementation is based on using the `CVXOPT` library in Python [ADV22; Van10]. Regarding the regularisation parameter γ, this is fixed as 1 following the considerations of [HS13b].

For the NMP-EMD algorithm, a scheme for defining the over-completeness parameter ρ is employed. This is done to reduce the sensitivity of the algorithm to this parameter, particularly in situations where $\rho\lambda_j L_\theta$ is less than one since in this case $V_b(\theta;\lambda_j) = \{1\}$ contains no frequency modes. We choose to fix the cardinality of $V_b(\theta;\cdot)$, thereby defining the over-completeness parameter ρ at each iteration in order to comply with this criteria. To avoid the over-completeness parameter ρ becoming too small such that the adaptivity of the algorithm is lost, the lower bound for ρ is set as 2.

With the FFT-NMP-EMD algorithm, a step in the algorithm is to interpolate the phase function $\theta_k^l(t_i)$ to a uniform mesh in the θ-coordinate. Additionally, an interpolation is used to to map the solutions $a_{k,\theta}^{l+1}(\theta_{k,i}^l)$ and $b_{k,\theta}^{l+1}(\theta_{k,i}^l)$ given in the uniform mesh in the θ-coordinate back to the physical grid $\theta_k^l(t_i)$. Practically, these interpolations are done using cubic spline interpolation.

The filter $\chi_{V(\theta_k^l;\lambda)}$ can be constructed in a multitude of ways. One approach could be to define the frequency response of the filter as an ideal filter in the frequency domain, i.e.

$$\hat{h}_{k,\kappa}^l = \begin{cases} 1, & |\kappa| < \lambda L_{\theta_k^l}, \\ 0, & \text{otherwise.} \end{cases}$$

for $\kappa = -\frac{n}{2} + 1, \ldots, \frac{n}{2}$ assuming n is even. This type of filter has the disadvantage that a step function in the frequency domain causes ripple effects in the time domain as a consequence of Gibbs' phenomenon. Instead, we utilise a raised cosine filter

which has the frequency response [HS13b]

$$\hat{h}^l_{k,\kappa} = \begin{cases} \frac{1}{2} - \frac{1}{2}\cos\left((\kappa - \lambda L_{\theta^l_k})\frac{\pi}{\lambda L_{\theta^l_k}}\right), & |\kappa| < \lambda L_{\theta^l_k}, \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

for $\kappa = -\frac{n}{2} + 1, \ldots, \frac{n}{2}$ assuming n is even providing a smooth transition to zero near the boundary of the support of $\hat{h}^l_k$.

To initialise Eq. (3.10), an initial estimate ω^0_k of the IF ω_k is required. This initial estimate can be acquired with a multitude of different methods, e.g. the synchrosqueezed wavelet transform [DLW11]. However, following the approach in [HS13b], in this thesis the frequency is initialised by searching for a frequency which the high frequency content of the residual is centred around. Specifically, this is done using the HHT by computing

$$\omega^0_k = \frac{1}{n}\sum_{i=1}^{n} \omega^{\mathrm{EMD}}_k(t_i)$$

where ω^{EMD}_k is the IF derived using the EMD in conjunction with the analytic signal method using the Hilbert transform to determine the IF.

The type of iterative algorithm employed in this section is sensitive to the initial estimate of the IF. Empirically, it has been found that by defining a sequence of $V(\theta;\lambda_j)$-spaces, the algorithm tends to converge even from rough initial estimates. This sequence is designed as $V(\theta;\lambda_1) \subset V(\theta;\lambda_2) \subset \cdots \subset V(\theta;\lambda_L) = V(\theta;\lambda)$ and can be constructed by iteratively increasing the λ hyperparameter as the algorithm has converged within a given $V(\theta;\lambda_j)$-space. As such, an ordered sequence of λ_j parameters is defined as $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_L = \lambda$ from which the $V(\theta;\lambda_j)$-spaces can be constructed. Using this method, for the first iteration $j = 1$, the smoothness of a_k and ω_k is significantly restricted and as $j \to L$, the smoothness restriction is lessened. Following the choices of [HS13b], L is fixed as 20 and λ_1 is set to 0. [HS13b]

In a practical implementation, if the expression $(a^{l+1}_k)^2(t) + (b^{l+1}_k)^2(t)$ is close to zero for some t, then the division in Eq. (3.8) can become unstable. To alleviate this practical issue, a threshold $\xi > 0$ is fixed and if $(a^{l+1}_k)^2(t) + (b^{l+1}_k)^2(t) < \xi$ for some t, then the value of $\Delta\omega^{l+1}_k(t)$ is not used and instead the value is assigned based on an interpolation of $\Delta\omega^{l+1}_k(t)$ at other times where $(a^{l+1}_k)^2(t) + (b^{l+1}_k)^2(t) \geq \xi$. We have chosen $\xi = 0.1$ in the numerical experiments in this thesis. [HS13b]

To determine the maximum in Eq. (3.9), consider the inequality

$$\omega^l_k - \eta^{l+1}_k \Delta\omega^{l+1}_k > 0$$

for $\eta^{l+1}_k \in [0,1]$ and rearrange the inequality as

$$\eta^{l+1}_k < \frac{\omega^l_k}{\Delta\omega^{l+1}_k}.$$

Moreover, note that for points where $\Delta\omega_k^{l+1} < 0$ the step direction for the IF is away from zero. Let I denote the indices where $\Delta\omega_k^{l+1}(t_i)$ is greater than zero. With the previous discussion in mind, if $I = \emptyset$, then $\eta_k^{l+1} = 1$. Otherwise, the step size η_k^{l+1} is chosen as

$$\eta_k^{l+1} = \min\Big\{ \min_{i \in I} \Big\{ \frac{\omega_k^l(t_i)}{2\Delta\omega_k^{l+1}(t_i)} \Big\}, 1 \Big\}.$$

In this way, η_k^{l+1} is guaranteed to be in the interval $[0, 1]$ while also ensuring that $\omega_k^l - \eta_k^{l+1}\Delta\omega_k^{l+1}$ is positive.

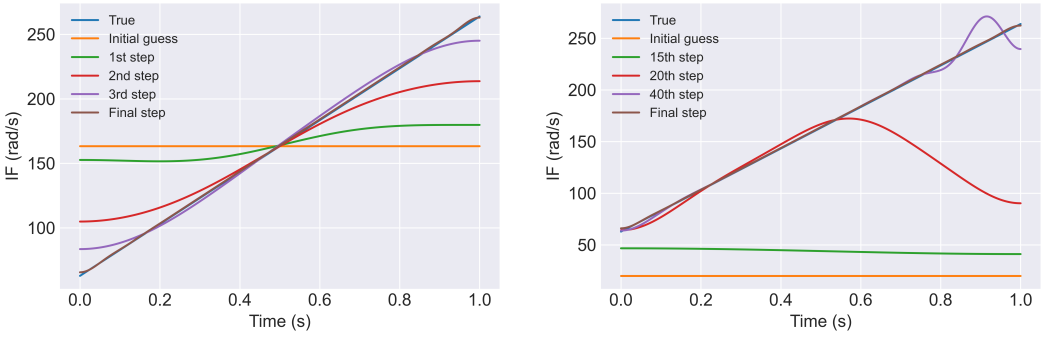
An algorithm outlining the NMP-EMD method is given in Algorithm 2. With the FFT-NMP-EMD method, the approach introduced in Section 3.4 replaces the step in Algorithm 2 where a_k^{l+1} and b_k^{l+1} are determined. The rest of the algorithm proceeds analogously.

3.6 Synthetic Example

To visualise the progression of the NMP-EMD algorithm, a synthetic example has been constructed in which a signal is defined as

$$y(t) = \cos\big(2\pi(10t + 16t^2)\big)$$

for $t \in [0, 1]$ with 300 equidistantly spaced samples. In Fig. 3.1, the updates of the IF when running the NMP-EMD method is visualised for a good initial guess and a poor initial guess. For this example $\gamma = 0$, $\lambda = 0.5$, and $\delta = 1\text{e}{-05}$. Moreover, for the good initial guess, the initial smoothness parameter, λ_1, is 0.1 and the length of the smoothness parameter sequence, L, is 10. On the other hand, for the poor initial guess, the initial smoothness parameter, λ_1, is 0 and the length of the smoothness parameter sequence, L, is 20. With both the good initial guess and the poor initial guess it is seen that the method converges to the true IF. This shows the benefit of the λ-sequence as this allows convergence in both cases. Additionally, it is noticed that a good initial guess can significantly reduce the number of required iterations.



(a) Good initial guess. **(b)** Poor initial guess.

Figure 3.1: Progress of IF during the NMP-EMD algorithm. Inspired by Fig. 18 in [HS13b].

Algorithm 2 NMP-EMD

Input: Signal $\boldsymbol{y}$, regularisation parameter γ, threshold parameter δ, smoothness parameters $\{\lambda_j\}_{j=1}^{L}$.

Initialise $k = 1$; $\boldsymbol{r}_k = \boldsymbol{y}$; IMFs $= \{\}$.
while $\|\boldsymbol{r}_k\|_2 > \delta$ **do**
  $l = 0$.
  Find ω_k^0.
  Determine θ_k^0 from ω_k^0 by summation.
  for all $j \in \{1, \dots, L\}$ **do**
    while $\|\theta_k^l - \theta_k^{l-1}\|_2 > \delta$ **do**
      For the $V(\theta_k^l; \lambda_j)$-space, solve the ℓ^1-regularised least square problem

$$\underset{\boldsymbol{z}_k^{l+1}}{\text{Minimise}} \qquad \frac{1}{2}\|\boldsymbol{r}_k - \boldsymbol{A}_k^l \boldsymbol{z}_k^{l+1}\|_2^2 + \gamma\|\boldsymbol{z}_k^{l+1}\|_1 \tag{P2}$$

      and let $\boldsymbol{x}_k^{l+1} = \big(\hat{\boldsymbol{a}}_k^{l+1}, \hat{\boldsymbol{b}}_k^{l+1}\big)^T$ be the solution.
      Update θ_k^l as

$$\theta_k^{l+1} = \theta_k^l - \eta_k^{l+1}\Delta\theta_k^{l+1} \tag{3.15}$$

      where $\Delta\theta_k^{l+1}(t_1) = 0$ and for $i = 2, \dots, n$

$$\Delta\theta_k^{l+1}(t_i) = \Delta\theta_k^{l+1}(t_{i-1}) + \frac{1}{f_s}\frac{\Delta\omega_k^{l+1}(t_{i-1}) + \Delta\omega_k^{l+1}(t_i)}{2},$$
$$\Delta\omega_k^{l+1} = \chi_{V(\theta_k^l;\lambda_j)}\Big\{\frac{\big(b_k^{l+1}\big)' a_k^{l+1} - b_k^{l+1}\big(a_k^{l+1}\big)'}{\big(a_k^{l+1}\big)^2 + \big(b_k^{l+1}\big)^2}\Big\}$$

      for the linear low-pass filter $\chi_{V(\theta_k^l;\lambda_j)}$ with frequency response as in Eq. (3.14), and where

$$\eta_k^{l+1} = \max\big\{\mu \in [0,1] : \omega_k^l - \mu\Delta\omega_k^{l+1} \geq 0\big\}.$$

      $l \leftarrow l + 1$.
    end while
  end for
  $a_k = \sqrt{\big(a_k^l\big)^2 + \big(b_k^l\big)^2}$.
  $\theta_k = \theta_k^l$.
  IMFs $\leftarrow$ IMFs $\cup \{a_k(t)\cos\big(\theta_k(t)\big)\}$.
  Update the residual

$$r_{k+1}(t) = y(t) - \sum_{i=1}^{k} a_i(t)\cos\big(\theta_i(t)\big).$$

  $k \leftarrow k + 1$.
end while
$r(t) = r_k(t)$.
Output: IMFs, $r(t)$.

3.7 Influence of Smoothness Parameter

In this section, the influence of the smoothness parameter is analysed using synthetic examples. This is done since the performance of the algorithm has been found to be particularly sensitive to the smoothness parameter λ.

Consider an example with two components of the same constant amplitude separated in frequency by a parameter f and where additionally the IF of the high frequency component is a linear function with slope α. The signal is defined as follows

$$y(t;\alpha,f) = \cos\left(20\pi t + \pi\alpha t^2\right) + \cos\left(2\pi f t\right)$$

for $t \in [0,T]$, $\alpha \leq 0$ and $f \in (0,10)$. The variable α both determines the bandwidth of the high frequency component and alters the separation between the two frequency components as a function of time.

Firstly, consider an experiment where $\alpha = 0$ is fixed and define a performance measure as

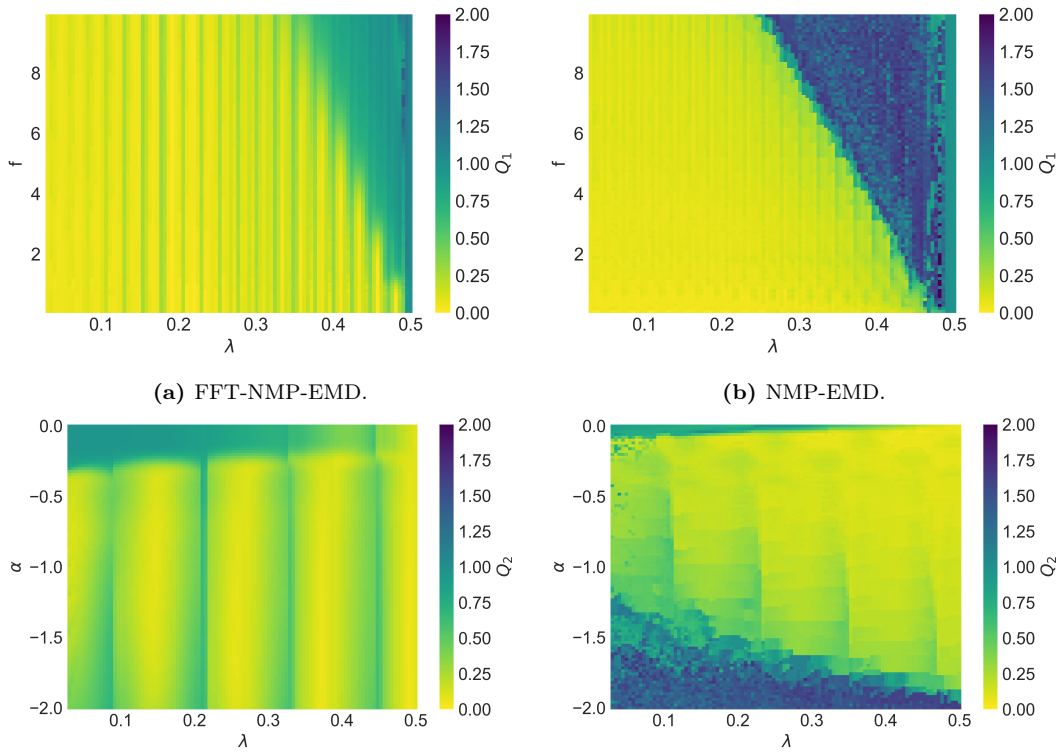
$$Q_1(\lambda, f; \hat{c}_1) = \frac{\|\hat{c}_1(t;\lambda,f) - \cos(20\pi t)\|_2}{\|\cos(2\pi f t)\|_2} \tag{3.16}$$

where $f \in (0,10)$, $\lambda \in (0,0.5]$, and $\hat{c}_1(t;\lambda,f)$ is the first IMF extracted with smoothness parameter λ and for relative frequency f. The results with the FFT-NMP-EMD and the NMP-EMD algorithms are shown in Figs. 3.2c and 3.2d. These figures show that as a general trend, components with constant frequency and constant amplitude are best found when the smoothness parameter is sufficiently low, and when this is the case even closely spaced components can be resolved.

Secondly, consider an experiment where $f = 5$ is fixed and define a performance measure as

$$Q_2(\lambda, \alpha; \hat{c}_1) = \frac{\|\hat{c}_1(t;\lambda,\alpha) - \cos\left(20\pi t + \pi\alpha t^2\right)\|_2}{\|\cos(10\pi t)\|_2} \tag{3.17}$$

where $\alpha \in [-2,0]$, $\lambda \in (0,0.5]$, and $\hat{c}_1(t;\lambda,\alpha)$ is the first IMF extracted with smoothness parameter λ and for slope α. The results are shown in Figs. 3.2c and 3.2d for the FFT-NMP-EMD and NMP-EMD algorithms and shows that the best choice of smoothness parameter λ is generally speaking $\lambda = 0.5$. Additionally, is seems that the FFT-NMP-EMD method outperforms the NMP-EMD method when α is close to -2.0.



(a) FFT-NMP-EMD.

(b) NMP-EMD.

(c) FFT-NMP-EMD.

(d) NMP-EMD.

Figure 3.2: Figs. 3.2a and 3.2b: Relation between relative frequency f and smoothness parameter λ evaluated according to Eq. (3.16). Figs. 3.2c and 3.2d: Relation between frequency slope α and smoothness parameter λ evaluated according to Eq. (3.17).

4. Uniqueness of Compressive Sensing Algorithm

In this chapter, theoretical results regarding the uniqueness of the decomposition resulting from the NMP-EMD and FFT-NMP-EMD methods introduced in Chapter 3 are proven. However, in order to prove the theoretical properties some assumptions have to be considered regarding the signals being decomposed. Therefore, the notion of scale separation is defined.

Definition 4.1 (Scale Separation)
A pair $\big(a(t), \theta(t)\big)$ is said to satisfy the scale separation property with a separation factor $\varepsilon > 0$ if $a(t) \in C^1(\mathbb{R})$ and $\theta(t) \in C^2(\mathbb{R})$ satisfy the following criteria

$$\inf_{t\in\mathbb{R}} \theta'(t) > 0, \quad \frac{\sup\limits_{t\in\mathbb{R}} \theta'(t)}{\inf\limits_{t\in\mathbb{R}} \theta'(t)} = M < \infty, \quad \left|\frac{a'(t)}{\theta'(t)}\right| \leq \varepsilon, \quad \left|\frac{\theta''(t)}{\big(\theta'(t)\big)^2}\right| \leq \varepsilon \tag{4.1}$$

for all $t \in \mathbb{R}$. The dictionary $\mathcal{D}_\varepsilon$ is defined as the set of functions on the form $y(t) = a(t)\cos(\theta(t))$ which fulfils Eq. (4.1) with separation factor ε. [LSH15]

A multi-component signal where each mono-component satisfies the scale separation property as in Definition 4.1 can be said to be well-separated if additional criteria are satisfied. Hence, the notion of well-separated signals is introduced.

Definition 4.2 (Well-Separated Signal)
Let $\mathcal{D}_\varepsilon$ be the dictionary from Definition 4.1 with separation factor ε. A signal $y : \mathbb{R} \to \mathbb{R}$ is said to be well-separated with separation factor ε and frequency ratio f_r if it can be expressed as

$$y(t) = \sum_{k=1}^{s} a_k(t)\cos(\theta_k(t)) + r(t)$$

where all $a_k(t)\cos(\theta_k(t)) \in \mathcal{D}_\varepsilon$, $|r(t)| \leq \varepsilon_0$ for all $t \in \mathbb{R}$ and for $\varepsilon_0 > 0$, and the phase functions satisfy

$$\theta_k'(t) \geq f_r \theta_{k-1}'(t)$$

for all $t \in \mathbb{R}$ and for $f_r > 1$, $f_r < N < \infty$ for some constant $N \in \mathbb{R}$. The dictionary $\mathcal{A}_{\varepsilon,f_r}$ is defined as the set of functions which are well-separated with scale separation factor ε and frequency ratio f_r. [LSH15]

Using $\mathcal{A}_{\varepsilon,f_r}$, a CS problem similar to Eq. (3.4) can be formulated where the dictionary $\mathcal{A}_{\varepsilon,f_r}$ is used in place of $\mathcal{D}$. For this dictionary, theoretical properties of the decomposition methods are shown.

Assume that the signal $y(t) \in \mathcal{A}_{\varepsilon,f_r}$. For this kind of signal, the existence of a decomposition is fulfilled since $y(t)$ has a representation

$$y(t) = \sum_{k=1}^{s} a_k(t)\cos(\theta_k(t)) + r(t). \tag{4.2}$$

Each of the feasible decompositions given by Eq. (4.2) gives an integer s and by collecting these integers in a set B, the solution to Eq. (P_δ) is given as $s_0 = \inf B$ where by the assumptions on $y(t)$, B is a non-empty set of positive integers. This gives the existence of a solution. Before presenting the main theorem of this chapter, a lemma is introduced.

Lemma 4.3 ([LSH15])
Let ψ be a wavelet function such that

$$I_1 = \int_{\mathbb{R}} |\psi(\tau)|d\tau < \infty, \quad I_2 = \int_{\mathbb{R}} |\tau\psi'(\tau)|d\tau < \infty, \quad I_3 = \int_{\mathbb{R}} |\tau^2\psi''(\tau)|d\tau < \infty.$$

Suppose $\big(a(t), \theta(t)\big)$ satisfies Definition 4.1 with separation factor ε. Then

$$\mathcal{W}\{ae^{-j\theta}\}(t,\omega) = \sqrt{\omega}a(t)e^{-j\theta(t)}\hat{\psi}\big(\omega\theta'(t)\big) + C\sqrt{\omega}\varepsilon \tag{4.3}$$

where $\mathcal{W}\{ae^{-j\theta}\}(t,\omega)$ denotes the wavelet transform of $ae^{-j\theta}$, $\hat{\psi}$ denotes the Fourier transform of ψ, and

$$C = (\sup_{t\in\mathbb{R}} |a(t)| + 4|a(t)| + 1)I_1 + (M + (M+1)|a(t)|)I_2 + M|a(t)|I_3$$

with $M = \frac{\sup_{t\in\mathbb{R}} \theta'(t)}{\inf_{t\in\mathbb{R}} \theta'(t)}$.

Proof.
For a proof see [LSH15]. ■

For readers unfamiliar with the wavelet transform, a definition can be found in Appendix A.2. The following theorem states that the decomposition is unique up to an error term depending on ε and ε_0. However, in order to simplify the notation assume $\varepsilon_0 = \varepsilon$.

Theorem 4.4 ([LSH15])
Let $y : [0,T] \to \mathbb{R}$ be well-separated with separation factor ε approaching 0 and frequency ratio f_r. Then $\{a_k, \theta_k\}_{k=1}^{s}$ is an optimal solution to the optimisation problem Eq. (P_δ) and it is unique up to the error ε, i.e. if $\{\tilde{a}_k, \tilde{\theta}_k\}_{k=1}^{\tilde{s}}$ is another optimal solution to Eq. (P_δ), then $\tilde{s} = s$ and

$$|a_k(t) - \tilde{a}_k(t)| \leq 2\sqrt{T}\varepsilon, \quad \frac{|\theta_k(t) - \tilde{\theta}_k(t)|}{\theta_k'(t)} \leq \frac{C'}{1 - \frac{\Delta}{2}}\varepsilon \tag{4.4}$$

for all $t \in [0,T]$ and for $k = 1, \ldots, s$ where $C' > 0$ is a positive constant and $0 < \Delta < \frac{\sqrt{f_r}-1}{\sqrt{f_r}+1}$.

Proof.
In order to prove this theorem, assume there are two decompositions of y such that

$$y(t) = \sum_{k=1}^{s} a_k(t)\cos(\theta_k(t)) + r(t) = \sum_{k=1}^{\tilde{s}} \tilde{a}_k(t)\cos\left(\tilde{\theta}_k(t)\right) + \tilde{r}(t)$$

where $|r(t)| \leq \varepsilon$ and $|\tilde{r}(t)| \leq \varepsilon$ for all t.

First it is proven that $s = \tilde{s}$. In order to use Lemma 4.3, a complex function is defined as

$$g(t) = \sum_{k=1}^{s} a_k(t)\exp(-j\theta_k(t)) + r(t) = \sum_{k=1}^{\tilde{s}} \tilde{a}_k(t)\exp\left(-j\tilde{\theta}_k(t)\right) + \tilde{r}(t).$$

Note that $y(t) = \mathrm{Re}\{g(t)\}$. Using Lemma 4.3

$$\begin{aligned}\frac{1}{\sqrt{\omega}}\mathcal{W}\{g\}(t,\omega) &= \sum_{k=1}^{s} a_k(t)e^{-j\theta_k(t)}\hat{\psi}(\omega\theta_k'(t)) + e(\varepsilon;t,\omega) \\ &= \sum_{k=1}^{\tilde{s}} \tilde{a}_k(t)e^{-j\tilde{\theta}_k(t)}\hat{\psi}(\omega\tilde{\theta}_k'(t)) + \tilde{e}(\varepsilon;t,\omega)\end{aligned} \tag{4.5}$$

where the terms $|e(\varepsilon;t,\omega)| = |sC\varepsilon + W\{r\}(t,\omega)| \leq (sC+\sqrt{T})\varepsilon$ and $|\tilde{e}(\varepsilon;t,\omega)| = |sC\varepsilon + W\{\tilde{r}\}(t,\omega)| \leq (sC+\sqrt{T})\varepsilon$. Note that the term $sC\varepsilon$ is derived using Lemma 4.3 and the term $\sqrt{T}\varepsilon$ is derived using the property that $|W\{r\}(t,\omega)| \leq \|r\|_2$ since then [Dau94, p. 24]

$$|W\{r\}(t,\omega)| \leq \|r\|_2 = \sqrt{\int_0^T |r(t)|^2 dt} \leq \sqrt{\int_0^T |\varepsilon|^2 dt} = \sqrt{T}\varepsilon$$

where it is assumed that $\|\psi\|_2 = 1$.

Now pick a wavelet function ψ which fulfils the following properties: The Fourier transform $\hat{\psi} \in C^2$ has support in $[1-\Delta, 1+\Delta]$ where $0 < \Delta < \frac{\sqrt{f_r}-1}{\sqrt{f_r}+1}$ and $\hat{\psi}(1) = 1$ is the maximum of $|\hat{\psi}|$. This wavelet is chosen as a fifth order B-spline after proper scaling and translation [LSH15]. Now fix $t = t_0 \in [0,T]$ and for any $l \in \{1,\dots,s\}$ let $\omega_{l,0} = \frac{1}{\theta_l'(t_0)}$. Using Eq. (4.5)

$$\begin{aligned}\frac{1}{\sqrt{\omega}}\mathcal{W}\{g\}(t_0,\omega_{l,0}) &= \sum_{k=1}^{s} a_k(t_0)e^{-j\theta_k(t_0)}\hat{\psi}\Big(\frac{\theta_k'(t_0)}{\theta_l'(t_0)}\Big) + e(\varepsilon;t_0,\omega_{l,0}) \\ &= \sum_{k=1}^{\tilde{s}} \tilde{a}_k(t_0)e^{-j\tilde{\theta}_k(t_0)}\hat{\psi}\Big(\frac{\tilde{\theta}_k'(t_0)}{\theta_l'(t_0)}\Big) + \tilde{e}(\varepsilon;t_0,\omega_{l,0}).\end{aligned} \tag{4.6}$$

Now consider $\frac{\theta_k'(t_0)}{\theta_l'(t_0)} \leq \frac{1}{f_r} < 1-\Delta$ and $\frac{\theta_k'(t_0)}{\theta_l'(t_0)} \geq f_r > \Delta+1$ for $k \neq l$ where the first inequalities are results of Definition 4.2 and the second inequality is a result of the fact that $f_r > 1$ and that the upper bound of Δ increases slower than f_r for $f_r > 1$ by Lemma A.9. Thus, by the assumption of the support of $\hat{\psi}$ only the term $k = l$ is non-zero in the first sum of Eq. (4.6)

$$a_l(t_0)e^{-j\theta_l(t_0)}\hat{\psi}(1) = \sum_{k=1}^{\tilde{s}} \tilde{a}_k(t_0)e^{-j\tilde{\theta}_k(t_0)}\hat{\psi}\Big(\frac{\tilde{\theta}_k'(t_0)}{\theta_l'(t_0)}\Big) + \tilde{e}(\varepsilon;t_0,\omega_{l,0}) - e(\varepsilon;t_0,\omega_{l,0}) \tag{4.7}$$

where $|\tilde{e}(\varepsilon;t_0,\omega_{l,0}) - e(\varepsilon;t_0,\omega_{l,0})| \leq 2\sqrt{T}\varepsilon$. In order for Eq. (4.7) to be true, there exists at least one $I(l,t_0) \in \{1,\dots,\tilde{s}\}$ such that

$$|\hat{\psi}\Big(\frac{\tilde{\theta}_{I(l,t_0)}'(t_0)}{\theta_l'(t_0)}\Big)| > 0$$

which means

$$1-\Delta < \frac{\tilde{\theta}_{I(l,t_0)}'(t_0)}{\theta_l'(t_0)} < 1+\Delta. \tag{4.8}$$

By the assumption that $y \in \mathcal{A}_{\varepsilon,f_r}$, for any $k \neq l$

$$\frac{\theta_l'(t_0)}{\theta_k'(t_0)} \geq f_r \quad \text{or} \quad \frac{\theta_l'(t_0)}{\theta_k'(t_0)} \leq \frac{1}{f_r}.$$

Using Lemma A.9 gives

$$\frac{\tilde{\theta}'_{I(l,t_0)}(t_0)}{\theta'_k(t_0)} = \frac{\tilde{\theta}'_{I(l,t_0)}(t_0)}{\theta'_l(t_0)} \frac{\theta'_l(t_0)}{\theta'_k(t_0)} \geq f_r(1-\Delta) > 1+\Delta$$

or

$$\frac{\tilde{\theta}'_{I(l,t_0)}(t_0)}{\theta'_k(t_0)} = \frac{\tilde{\theta}'_{I(l,t_0)}(t_0)}{\theta'_l(t_0)} \frac{\theta'_l(t_0)}{\theta'_k(t_0)} \leq \frac{1+\Delta}{f_r} < 1-\Delta.$$

Then for any $k \neq l$

$$|\hat{\psi}\Big(\frac{\tilde{\theta}'_{I(l,t_0)}(t_0)}{\theta'_k(t_0)}\Big)| = 0.$$

This implies $I(k,t_0) \neq I(l,t_0)$, $k \neq l$. Thus,

$$\tilde{s} \geq s.$$

However, $\{\tilde{a}_k, \tilde{\theta}_k\}_{k=1}^{s}$ is a solution to Eq. (P_δ), therefore $\tilde{s} \leq s$ and hence

$$\tilde{s} = s. \tag{4.9}$$

Now it is proven that $|a_k(t) - \tilde{a}_k(t)| \leq 2\sqrt{T}\varepsilon$. Eq. (4.9) implies that for any $t \in [0,T]$, $I(\cdot,t) : \{1,\ldots,s\} \to \{1,\ldots,s\}$ is a one-to-one mapping. As such, $I^{-1}(\cdot,t) : \{1,\ldots,s\} \to \{1,\ldots,s\}$ can be defined. The proof starts by studying the function $I^{-1}(k,\cdot) : [0,T] \to \{1,\ldots,s\}$ for $k = 1,\ldots,s$. Using the condition

$$\frac{\tilde{\theta}''}{\left(\tilde{\theta}'\right)^2} \leq \varepsilon$$

from Definition 4.1 and that the signal $y \in \mathcal{A}_{f_r,\varepsilon}$, then it is shown that $I^{-1}(k,\cdot)$ is constant over $[0,T]$, i.e.

$$I^{-1}(k,t) = I^{-1}(k,0)$$

for $t \in [0,T]$ and $k = 1,2,\ldots,s$. Suppose by contradiction that there exists $t_0 \in [0,T]$ such that $I^{-1}(k,t_0) \neq I^{-1}(k,0)$. Let $A = \{0 \leq t \leq t_0 : I^{-1}(k,t_0) = I^{-1}(k,0)\}$ and $\xi = \sup A$. Then for any $\eta > 0$ there exists $t_1, t_2 \in [0,T]$ such that

$$t_1 < \xi < t_2, \quad |t_2 - t_1| < \eta, \quad I^{-1}(k,0) = I^{-1}(k,t_1) \neq I^{-1}(k,t_2).$$

Now, define the notation

$$I^{-1}(k,t_1) = k_1, \quad I^{-1}(k,t_2) = k_2$$

for $k_1 \neq k_2$. Since $I(k,t)$ is a one-to-one mapping and by Eq. (4.8)

$$1-\Delta < \frac{\tilde{\theta}'_k(t_1)}{\theta'_{k_1}(t_1)} < 1+\Delta, \quad 1-\Delta < \frac{\tilde{\theta}'_k(t_2)}{\theta'_{k_2}(t_2)} < 1+\Delta. \tag{4.10}$$

Without loss of generality, assume that $\theta'_{k_2} > \theta'_{k_1}$. By Eq. (4.10)

$$\tilde{\theta}'_k(t_1) < (1+\Delta)\theta'_{k_1}(t_1), \quad \tilde{\theta}'_k(t_2) > (1-\Delta)\theta'_{k_2}(t_2).$$

Now, let $\eta \to 0$ such that $t_1, t_2 \to \xi$. Then,

$$\tilde{\theta}'_k(\xi) \leq (1+\Delta)\theta'_{k_1}(\xi), \quad \tilde{\theta}'_k(\xi) \geq (1-\Delta)\theta'_{k_2}(\xi).$$

Since $y \in \mathcal{A}_{\varepsilon,f_r}$,

$$\frac{\theta'_{k_1}(t)}{\theta'_{k_2}(t)} \leq \frac{1}{f_r}$$

and then

$$\tilde{\theta}'_k(\xi) \leq (1+\Delta)\theta'_{k_1}(\xi), \quad \tilde{\theta}'_k(\xi) \geq f_r(1-\Delta)\theta'_{k_1}(\xi) > (1+\Delta)\theta'_{k_1}(\xi)$$

which is a contradiction. Thus, $I^{-1}(k,\cdot)$ is a constant over $[0,T]$, and it can be assumed that

$$I^{-1}(k,t) = k$$

for $t \in [0,T]$ and $k = 1,\ldots,s$ which implies

$$1-\Delta < \frac{\tilde{\theta}'_k(t)}{\theta'_k(t)} < 1+\Delta.$$

Now for any $\theta \in C^1$ define the set

$$U_\theta = \{(t,\omega) \in \mathbb{R}^2 : |\hat{\psi}(\omega\theta'(t))| > \varepsilon\}.$$

Using the assumption that $y \in \mathcal{A}_{\varepsilon,f_r}$ and the choice of ψ, for any $k,l = 1,\ldots,s$, $k \neq l$ it follows that

$$\hat{\psi}\big(\omega\theta'_l(t)\big) = \hat{\psi}\big(\omega\tilde{\theta}'_l(t)\big) = 0, \quad \forall (t,\omega) \in U_{\theta_k}, \tag{4.11}$$

$$\hat{\psi}\big(\omega\theta'_l(t)\big) = \hat{\psi}\big(\omega\tilde{\theta}'_l(t)\big) = 0, \quad \forall (t,\omega) \in U_{\tilde{\theta}_k} \tag{4.12}$$

and by Lemma 4.3

$$\begin{aligned}\frac{1}{\sqrt{\omega}}\mathcal{W}\{g\}(t,\omega) &= \sum_{k=1}^{s} a_k(t)e^{-j\theta_k(t)}\hat{\psi}\big(\omega\theta'_k(t)\big) + e(\varepsilon;t,\omega) \\ &= \sum_{k=1}^{\tilde{s}} \tilde{a}_k(t)e^{-j\tilde{\theta}_k(t)}\hat{\psi}\big(\omega\tilde{\theta}'_k(t)\big) + \tilde{e}(\varepsilon;t,\omega). \end{aligned} \tag{4.13}$$

Using Eqs. (4.11) to (4.13)

$$|a_k(t)e^{-j\theta_k(t)}\hat{\psi}\big(\omega\theta'_k(t)\big) - \tilde{a}_k(t)e^{-j\tilde{\theta}_k(t)}\hat{\psi}\big(\omega\tilde{\theta}'_k(t)\big)| \leq 2\sqrt{T}\varepsilon$$

for all $(t,\omega) \in U_{\theta_k} \cup U_{\tilde{\theta}_k}$ which implies

$$\big||a_k(t)\hat{\psi}\big(\omega\theta_k'(t)\big)| - |\tilde{a}_k\hat{\psi}\big(\omega\tilde{\theta}_k'(t)\big)|\big| \leq 2\sqrt{T}\varepsilon. \tag{4.14}$$

Assume $a_k(t) > \tilde{a}_k(t)$ and choose $\omega = \frac{1}{\theta_k'(t)}$

$$\left|a_k(t)|\hat{\psi}(1)| - \tilde{a}_k(t)\left|\hat{\psi}\Big(\frac{\tilde{\theta}_k'(t)}{\theta_k'(t)}\Big)\right|\right| \leq 2\sqrt{T}\varepsilon.$$

Since $a_k(t) > \tilde{a}_k(t)$ and remembering $|\hat{\psi}(\xi)|$ achieves its maximum at $\xi = 1$,

$$0 \leq |\hat{\psi}(1)|(a_k(t) - \tilde{a}_k(t)) \leq a_k(t)|\hat{\psi}(1)| - \tilde{a}_k(t)\left|\hat{\psi}\Big(\frac{\tilde{\theta}_k'(t)}{\theta_k'(t)}\Big)\right| \leq 2\sqrt{T}\varepsilon.$$

This shows

$$a_k(t) - \tilde{a}_k(t) \leq \frac{2\sqrt{T}\varepsilon}{|\hat{\psi}(1)|} = 2\sqrt{T}\varepsilon.$$

By a similar argument, assuming $\tilde{a}_k(t) > a_k(t)$, it can be shown that

$$\tilde{a}_k(t) - a_k(t) \leq 2\sqrt{T}\varepsilon.$$

Combining these cases, the result follows

$$|a_k(t) - \tilde{a}_k(t)| \leq 2\sqrt{T}\varepsilon.$$

We end the proof here. For a proof of the assertion

$$\frac{|\theta_k'(t) - \tilde{\theta}_k'(t)|}{\theta_k'(t)} \leq \frac{C'}{1 - \frac{\Delta}{2}}\varepsilon$$

see [LSH15]. ■

Now a result regarding a variation of the Eq. ($\mathrm{P_{NMP}}$) problem is shown. However, first a lemma which is useful later is introduced.

Lemma 4.5 ([LSH15])
Let a_k, θ_k for $k = 1, \ldots, s$ be well-separated with frequency ratio f_r and separation factor ε. Furthermore, let $\big(a(t), \theta(t)\big)$ satisfy Definition 4.1 and let there exist an $\alpha \in [1, f_r)$ and $l \in \{1, \ldots, s\}$ such that

$$\alpha^{-1}\frac{d\theta_l(t)}{dt} \leq \frac{d\theta(t)}{dt} \leq \alpha\frac{d\theta_l(t)}{dt}$$

for $t \in [0,1]$. Then

$$\mu_{k,l} = \frac{|\langle a_k \cos(\theta_k), a_l \cos(\theta_l)\rangle|}{\|a_k \cos(\theta_k)\|_2 \|a_l \cos(\theta_l)\|_2} < 4\varepsilon\big(\frac{1}{2} - 3\varepsilon\big)^{-1}\big(1 + \frac{1}{(1 - f_r^{-|l-k|})^2}\big),$$
$$\mu_{k,l,\alpha} = \frac{|\langle a_k \cos(\theta_k), a \cos(\theta)\rangle|}{\|a_k \cos(\theta_k)\|_2 \|a \cos(\theta)\|_2} < 4\varepsilon\big(\frac{1}{2} - 3\varepsilon\big)^{-1}\big(1 + \frac{1}{(1 - \alpha f_r^{-|l-k|})^2}\big)$$

for any $k \in \{1, \ldots, s\}$, $k \neq l$.

Proof.
For a proof see [LSH15]. ■

Consider the following variation of Eq. ($\mathrm{P_{NMP}}$)

$$\begin{aligned} &\text{Minimise} && p(a,\theta) = \|y(t) - a(t)\cos(\theta(t))\|_2^2 \\ &\text{Subject to} && a(t)\cos(\theta(t)) \in \mathcal{D}_\varepsilon. \end{aligned} \tag{4.15}$$

A theorem which states under which conditions on $a(t)\cos(\theta(t))$ the solution to the optimisation problem Eq. (4.15) could provide an approximation to Eq. (P) is proven. The following theorem shows that for a periodic signal each IMF is a local minimiser of Eq. (4.15) under certain assumptions. Note that the theorem applies to signals with finite support in time. As such, without loss of generality assume that the signal has support in the time interval $[0,1]$. When working with signals of finite length end effects occur. However, this problem is avoided by assuming that the signal is periodic with a period of 1. However, this also means that for signals which are non-periodic the results shown only applies for the interior region away from the boundary.

Theorem 4.6 ([LSH15])
Let $y \in \mathcal{A}_{\varepsilon,f_r}$ and suppose there exists an $\alpha \in [1, f_r)$ and an $l \in \{1, \ldots, s\}$ such that

$$\alpha^{-1}\frac{d\theta_l(t)}{dt} \leq \frac{d\theta(t)}{dt} \leq \alpha\frac{d\theta_l(t)}{dt}$$

for $t \in [0,1]$. If

$$p(a,\theta) \leq p(a_l, \theta_l),$$

then

$$\frac{\|a\cos(\theta) - a_l\cos(\theta_l)\|_2}{\|a_l\cos(\theta)\|_2} < C''\varepsilon$$

for a positive constant $C'' > 0$.

Proof.
In the following proof, the explanatory variable t is omitted for ease of notation. Additionally, for convenience of notation, $|y|$ where $y : [0,1] \to \mathbb{R}$ is used to denote the $L^2([0,1])$-norm. By assumption,

$$\begin{aligned}
0 &\geq p(a,\theta) - p(a_l,\theta_l) \\
&= |y - a\cos(\theta)|^2 - |y - a_l\cos(\theta_l)|^2 \\
&= \Big|\sum_{k=1}^{s} a_k\cos(\theta_k) + r - a\cos(\theta)\Big|^2 - \Big|\sum_{k=1}^{s} a_k\cos(\theta_k) + r - a_l\cos(\theta_l)\Big|^2 \\
&= \Big|\sum_{k=1,k\neq l}^{s} a_k\cos(\theta_k) + r - a\cos(\theta) + a_l\cos(\theta_l)\Big|^2 - \Big|\sum_{k=1,k\neq l}^{s} a_k\cos(\theta_k) + r\Big|^2 \\
&= \int \Big(\big(\sum_{k=1,k\neq l}^{s} a_k\cos(\theta_k) + r - a\cos(\theta) + a_l\cos(\theta_l)\big)^2 \\
&\qquad - \big(\sum_{k=1,k\neq l}^{s} a_k\cos(\theta_k) + r\big)^2 \Big) dt.
\end{aligned}$$

Consider the integrand and compute the square

$$\begin{aligned}
0 \geq\ & \big(\sum_{k=1,k\neq l}^{s} a_k\cos(\theta_k)\big)^2 + r^2 + a^2\cos^2(\theta) + a_l^2\cos^2(\theta_l) + 2\sum_{k=1,k\neq l}^{s} a_k\cos(\theta_k)r \\
&- 2\sum_{k=1,k\neq l}^{s} a_k\cos(\theta_k)a\cos(\theta) + 2\sum_{k=l,k\neq l}^{s} a_k\cos(\theta_k)a_l\cos(\theta_l) \\
&- 2ra\cos(\theta) + 2ra_l\cos(\theta_l) - 2a\cos(\theta)a_l\cos(\theta_l) \\
&- \big(\sum_{k=1,k\neq l}^{s} a_k\cos(\theta_k)\big)^2 - 2\sum_{k=1,k\neq l}^{s} a_k\cos(\theta_k)r - r^2.
\end{aligned}$$

Cancelling terms and re-introducing the integral yield the inequality

$$0 \geq |a_l\cos(\theta_l) - a\cos(\theta)|^2 + 2\langle a_l\cos(\theta_l) - a\cos(\theta), \sum_{k=1,k\neq l}^{s} a_k\cos(\theta_k) + r\rangle. \quad (4.16)$$

Now focus on the inner product in the above inequality. Defining the terms

$$\mu_{k,l} = \frac{|\langle a_l\cos(\theta_l), a_k\cos(\theta_k)\rangle|}{|a_l\cos(\theta_l)||a_k\cos(\theta_k)|}, \quad \delta_1 = \sum_{k=1,k\neq l}^{s} \mu_{k,l}\frac{|a_k\cos(\theta_k)|}{|a_l\cos(\theta_l)|} + \frac{|r|}{|a_l\cos(\theta_l)|}.$$

Using these terms,

$$\begin{aligned}
&|\langle a_l \cos(\theta_l), \sum_{k=1,k\neq l}^{s} a_k \cos(\theta_k) + r\rangle| \\
&\qquad \leq \sum_{k\neq l,k=1}^{s} |\langle a_k \cos(\theta_k), a_l \cos(\theta_l)\rangle| + |r||a_l \cos(\theta_l)| \\
&\qquad = \sum_{k=1,k\neq l}^{s} \mu_{k,l} |a_l \cos(\theta_l)||a_k \cos(\theta_k)| + |r||a_l \cos(\theta_l)| \\
&\qquad = \delta_1 |a_l \cos(\theta_l)|^2. \qquad (4.17)
\end{aligned}$$

Similarly, defining

$$\mu_{k,l,\alpha} = \frac{|\langle a\cos(\theta), a_k \cos(\theta_k)\rangle|}{|a\cos(\theta)||a_k \cos(\theta_k)|}, \quad \delta_2 = \sum_{k=1,k\neq l}^{s} \mu_{k,l,\alpha} \frac{|a_k \cos(\theta_k)|}{|a_l \cos(\theta_l)|} + \frac{|r|}{|a_l \cos(\theta_l)|}$$

and then

$$\begin{aligned}
&|\langle a\cos(\theta), \sum_{k=1,k\neq l}^{s} a_k \cos(\theta_k) + r\rangle| \\
&\qquad \leq \sum_{k=1,k\neq l}^{s} |\langle a\cos(\theta), a_k \cos(\theta_k)\rangle| + |a\cos(\theta)||r| \\
&\qquad = \sum_{k=1,k\neq l}^{s} \mu_{k,l,\alpha} |a\cos(\theta)||a_k \cos(\theta_k)| + |a\cos(\theta)||r| \\
&\qquad = \delta_2 |a\cos(\theta)||a_l \cos(\theta_l)| \\
&\qquad \leq \delta_2 |a\cos(\theta) - a_l \cos(\theta_l)||a_l \cos(\theta_l)| + \delta_2 |a_l \cos(\theta_l)|^2 \qquad (4.18)
\end{aligned}$$

where the last inequality is a result of the triangle inequality $|a\cos(\theta)| \leq |a\cos(\theta) - a_l\cos(\theta_l)| + |a_l\cos(\theta_l)|$. Combining Eqs. (4.16) to (4.18), it follows that

$$\begin{aligned}
0 &\geq |a_l \cos(\theta_l) - a\cos(\theta)|^2 - 2\delta_2 |a\cos(\theta) - a_l\cos(\theta_l)||a_l\cos(\theta_l)| \\
&\quad - 2(\delta_1 + \delta_2)|a_l\cos(\theta_l)|^2. \\
&\geq \frac{|a_l\cos(\theta_l) - a\cos(\theta)|^2}{|a_l\cos(\theta_l)|^2} - 2\delta_2 \frac{|a\cos(\theta) - a_l\cos(\theta_l)|}{|a_l\cos(\theta_l)|} - 2(\delta_1 + \delta_2).
\end{aligned}$$

Now, this quadratic equation is solved for the positive solution, as the norm is positive, to obtain

$$\frac{|a\cos(\theta) - a_l\cos(\theta_l)|}{|a_l\cos(\theta_l)|} \leq \delta_2 + \sqrt{\delta_2^2 + 2(\delta_1 + \delta_2)}. \qquad (4.19)$$

Using Lemma 4.5, then

$$\mu_{k,l} < 4\varepsilon\big(\frac{1}{2} - 3\varepsilon\big)^{-1}\big(1 + \frac{1}{(1 - f_r^{-|l-k|})^2}\big),$$
$$\mu_{k,l,\alpha} < 4\varepsilon\big(\frac{1}{2} - 3\varepsilon\big)^{-1}\big(1 + \frac{1}{(1 - \alpha f_r^{-|l-k|})^2}\big).$$

From these inequalities and using $|r(t)| \leq \varepsilon$ for all t, it follows that

$$\delta_1 < \sum_{k=1,k\neq l}^{s} 4\varepsilon\big(\frac{1}{2} - 3\varepsilon\big)^{-1}\big(1 + \frac{1}{(1 - f_r^{-|l-k|})^2}\big)\frac{|a_k\cos(\theta_k)|}{|a_l\cos(\theta_l)|} + \frac{\varepsilon}{|a_l\cos(\theta_l)|}, \tag{4.20}$$

$$\delta_2 < \sum_{k=1,k\neq l}^{s} 4\varepsilon\big(\frac{1}{2} - 3\varepsilon\big)^{-1}\big(1 + \frac{1}{(1 - \alpha f_r^{-|l-k|})^2}\big)\frac{|a_k\cos(\theta_k)|}{|a_l\cos(\theta_l)|} + \frac{\varepsilon}{|a_l\cos(\theta_l)|}. \tag{4.21}$$

Thus, it follows from Eq. (4.19)

$$\frac{|a\cos(\theta) - a_l\cos(\theta_l)|}{|a_l\cos(\theta_l)|} < C''\varepsilon \tag{4.22}$$

for a positive constant $C'' > 0$ which can be expressed explicitly using Eqs. (4.19) to (4.21). Note that the positive constant is bounded since $0 < |a_k\cos(\theta_k)| < \infty$ by the assumption $y \in \mathcal{A}_{\varepsilon,f_r}$. [LSH15] ■

Therefore, the uniqueness of the Eq. (P0$_\delta$) problem has been shown up to an error bound given by $C''\varepsilon$ and it has been shown that the solution to Eq. (4.15) estimates the optimal solution. The factor ε is related to the scale separation factor and the energy in the residual. The positive constant C'' is influenced by the frequency ratio and the energy in the components $a_k\cos(\theta_k)$. Specifically, when extracting component l of a signal $y \in \mathcal{A}_{\varepsilon,f_r}$, the error bound decreases when the frequency ratio increases and additionally decreases as the energy in component l increases in comparison to the energy in components k for $k \neq l$. [LSH15]

5. Partial Differential Equation Based Adaptive Decomposition

In this chapter, a method to adaptively decompose a signal using PDEs is introduced. The method was first introduced in [WMV18] and is considered in this thesis based on the discussion in Chapter 2. The method is in this thesis dubbed PDE-EMD and is based on the heat equation. In the context of this thesis, only introductory results regarding PDEs are needed to introduce the method.

The chapter is organised by first introducing the basics of PDEs in the context of the heat equation in Section 5.1 and a finite difference scheme for numerically solving the heat equation in Section 5.2. Then, in Section 5.3, the PDE-EMD method is described. Finally, in Section 5.4, the PDE-EMD method is tested on a simulated example.

5.1 Partial Differential Equation Basics

A PDE, defined below, is an equation of two or more variables containing at least one partial derivative.

Definition 5.1 (Partial Differential Equation)
A PDE is an equation involving partial derivatives of an unknown function $u : S \to \mathbb{R}$ where S is an open subset of $\mathbb{R}^d$ for $d \geq 2$. [Jos13, p. 1]

The study of PDEs is usually not focused on arbitrary PDEs but instead on equations which naturally occur in various applications. As mentioned, the type of PDE which is used in this thesis is the heat equation, defined below, whose name originates from its usage in modelling diffusion processes such as heat.

Definition 5.2 (Heat Equation)
Let $u : S \times \mathbb{R}^+ \to \mathbb{R}$ and $u \in C^2$, then the heat equation is given as

$$\frac{\partial u(x_1, \ldots, x_d, t)}{\partial t} = \alpha\Big(\sum_{i=1}^{d} \frac{\partial^2 u(x_1, \ldots, x_d, t)}{\partial^2 x_i}\Big)$$

where $\alpha \in \mathbb{R}$, $t \in \mathbb{R}^+$ and $(x_1, x_2, \ldots, x_d) \in S \subset \mathbb{R}^d$. The variable t is called the time coordinate and $x_1, x_2, \ldots, x_d$ the spatial coordinates. [Jos13, p. 2] [Kre11, p. 558]

PDEs are often classified not only by their type but by their characteristics. Considering the heat equation, it is classified as a linear PDE since it only contains the partial derivatives of u linearly. Similarly, it can also be classified as a second order PDE as the highest order of occurring partial derivatives is second order.

The use of the heat equation for adaptive decompositions is motivated by its property regarding regularity or smoothness of its solution stated below.

Theorem 5.3 (Smoothness of the Solution)
Suppose, $u \in C^2$ is a solution to a heat equation in $S \times (0, T]$. Then, $u \in C^\infty$. [Eva10, p. 59]

Proof.
For a proof see [Eva10, pp. 59-61]. ■

5.1.1 Initial and Boundary Conditions

In general, PDEs have an infinite number of solutions and to determine a specific solution, some conditions need to be imposed. There are two types of conditions which are usually imposed on a PDE which are the initial conditions and the boundary conditions. An initial condition on a system is specified at some time t_0

$$u(x, t_0) = f(x)$$

for some function f and another initial condition can be added such as

$$\frac{\partial u(x, t_0)}{\partial t} = g(x)$$

for some function g. In general, this pattern can be continued adding conditions on the nth order derivative. For an nth order PDE in time, $n-1$ initial conditions are needed in order to obtain a particular solution. Therefore, with the heat equation, only one initial condition is needed. [Dem21, p. 32]

Another type of condition which can be imposed are the boundary conditions. Boundary conditions are imposed on the boundary of the spatial domain as defined below.

Definition 5.4 (Closure)
Let $S \subset \mathbb{R}^d$ be an open set and define an open ball with radius $r > 0$ centred at $\boldsymbol{z} \in \mathbb{R}^d$ as $B_r(z) = \{\boldsymbol{x} \in \mathbb{R}^d \mid \nu(\boldsymbol{x}, \boldsymbol{z}) < r\}$ for a metric $\nu : S \times S \to \mathbb{R}^+$. The closure of S is then defined as

$$\overline{S} = \{\boldsymbol{x} \in \mathbb{R}^d \mid B_r(\boldsymbol{x}) \cap S \neq \emptyset \text{ for all } r > 0\}.$$

Definition 5.5 (Boundary)
Let $S \subset \mathbb{R}^d$ be an open set and let $\overline{S}$ denote the closure of S. Then the boundary of S is given by

$$B_S = \overline{S} \backslash S.$$

In the simple case of one dimension, consider $x \in (0, K)$. Then the boundary conditions could be defined as

$$\begin{aligned} u(0,t) &= f(t), \\ u(K,t) &= g(t) \end{aligned} \tag{5.1}$$

for some functions f and g. In this thesis, two types of boundary conditions are tested. These are the Dirichlet boundary conditions and the Neumann boundary conditions, respectively, which have been tested since they are standard boundary conditions. The Dirichlet boundary conditions are conditions on u at the boundary and as such Eq. (5.1) are examples of Dirichlet boundary conditions. The Neumann boundary conditions are conditions on the normal derivative of u on the boundary, i.e. conditions on $\boldsymbol{n}\boldsymbol{\nabla}u$ where $\boldsymbol{n}$ is the normal-vector to the hyperplane which defines the boundary and $\boldsymbol{\nabla}$ is the vector gradient operator. [Jak19, pp. 19-21][Kre11, pp. 557-564]

Given a PDE and some initial and boundary conditions yield a so-called initial and boundary value problem. For applications, the notion of a well posed problem, defined below, is used to summarise some desirable features in relation to solving the problem.

Definition 5.6 (Well Posed Problem)
[Jak19, p. 21] A well posed problem is a problem which satisfies the following two criteria

1. Existence and uniqueness: There exists one and only one solution which satisfies all the criteria of the model.
2. Stability: The unique solution depends continuously on the data of the problem, i.e. small changes in data lead to small changes in the solution.

In the following, the existence and uniqueness of a solution to the heat equation is considered. The stability point is discussed in Section 5.2 when considering numerical solutions.

The uniqueness of the heat equation assuming continuity of the initial and boundary conditions is established below.

Theorem 5.7 (Uniqueness on Bounded Domains)
Let $S_T = S \times (0, T]$ denote the domain of the solution and define the boundary as $B_{S_T} = \overline{S}_T \backslash S_T$. Consider a heat equation with initial and boundary conditions $g \in C(B_{S_T})$. Then there exist at most one solution $u \in C_1^2(S_T) \cap C(\overline{S}_T)$ of the problem

$$\begin{cases} \frac{\partial u(x_1,\ldots,x_d,t)}{\partial t} = \alpha\Big(\sum_{i=1}^{d} \frac{\partial^2 u(x_1,\ldots,x_d,t)}{\partial^2 x_i}\Big) & \text{in } S_T \\ u = g & \text{on } B_{S_T}. \end{cases}$$

The notation $C_1^2(S_T)$ refers to functions which are once continuous differentiable in the temporal variable and twice continuous differentiable in the spatial variables. [Eva10, p. 57]

Proof.
For a proof see [Eva10, p. 57]. ■

This result guarantees the uniqueness of the solution in case there is a solution. The existence of a solution can also be proven although this is a more laborious exercise and requires more strict assumptions. In [Eva10, p. 388], a smooth solution is constructed by assuming smoothness of the initial and boundary conditions and assuming that the boundary conditions vanishes.

5.2 Numerical Solution of the Heat Equation

So far in this chapter, PDEs have been considered in a continuous-time domain operating on functions $u : S \to \mathbb{R}$. The solution to the PDEs are as such functions u given in the domain S. In the following, the heat equation in one spatial dimension is considered and the domain is defined as $S = [0, K] \times [0, T]$. Numerical solutions to PDEs yield solutions in a discrete set of points denoted $\mathcal{S}$. Assume for simplicity that the points in $\mathcal{S}$ are equidistantly spaced. As such, the set $\mathcal{S}$ defines a two-dimensional grid in which the PDE is solved. Let Δx and Δt be the spacing between points in the first axis and second axis, respectively. Then a discretised domain is defined in the spatial dimension as $\mathcal{S}_{\Delta x} = \{x_i : i = 1, \ldots, n\}$ where $x_i = (i-1)\Delta x$ for $i = 1, \ldots, n$ and in the temporal domain as $\mathcal{S}_{\Delta t} = \{t_j \mid j = 0, 1, \ldots, \mathcal{T}\}$ where $t_j = j\Delta t$, $\mathcal{T} = \frac{T}{\Delta t}$, and it is assumed that Δt divides T. The discrete domain in which the PDE is solved numerically is $\mathcal{S}_{\Delta x} \times \mathcal{S}_{\Delta t}$.

In the following section, a finite difference scheme for numerically solving the heat equation is introduced.

5.2.1 Finite Difference Scheme

Consider the heat equation

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2}.$$

The partial derivatives can be approximated by the following equations

$$\frac{\partial u(x,t)}{\partial t} = \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t} + R_1 \tag{5.2}$$

and

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{u(x-\Delta x, t) - 2u(x,t) + u(x+\Delta x, t)}{\Delta x^2} + R_2 \tag{5.3}$$

where $R_1 \leq \frac{\Delta t}{2} \| \frac{\partial^2 u}{\partial t^2} \|_\infty$ and $R_2 \leq \frac{\Delta x^2}{6} \| \frac{\partial^3 u}{\partial x^3} \|_\infty$. [OO18, pp. 96-99]

Now define $u_i^j = u(x_i, t_j)$ for $i = 1, \ldots, n$ and $j \geq 0$. The equations Eqs. (5.2) and (5.3) motivate the following update scheme

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{u_{i-1}^j - 2u_i^j + u_{i+1}^j}{\Delta x^2}$$

which in turn gives the update formula

$$u_i^{j+1} = \frac{\Delta t}{\Delta x^2}(u_{i-1}^j - 2u_i^j + u_{i+1}^j) + u_i^j. \tag{5.4}$$

Using Dirichlet boundary conditions given by $u_0^j = u_{n+1}^j = 0$ for $j \geq 0$ and initial conditions as $u_i^0 = y(x_i)$. With Eq. (5.4), the values of iteration $j+1$ can be obtained using only values from iteration j [TW, pp. 119-120].

Eq. (5.4) can be expressed with matrix-vector notation as

$$\boldsymbol{u}^{j+1} = (\boldsymbol{I} - \frac{\Delta t}{\Delta x^2}\boldsymbol{A}_{\mathrm{D0}})\boldsymbol{u}^j \tag{5.5}$$

for $j \geq 0$ where $\boldsymbol{u}^j = (u_1^j, u_2^j, \ldots, u_n^j)^T$ with initial condition $\boldsymbol{u}^0 = (y(x_1), y(x_2), \ldots, y(x_n))^T$ and

$$\boldsymbol{A}_{\mathrm{D0}} = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}. \tag{5.6}$$

Eq. (5.5) provides an explicit finite difference updating scheme for the heat equation. [TW, pp.123-126]

Stability of the update scheme depends on the step sizes Δt and Δx. Specifically, if the following inequality is satisfied, then the update scheme is stable [TW, pp. 129-130]

$$\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}. \tag{5.7}$$

5.3 PDE Based Empirical Mode Decomposition

In an effort to find an alternative to the cubic spline interpolation used in the EMD, [DLN05] introduced a PDE based method for determining the upper and lower envelope of a signal which in turn is used to extract the IMFs of a signal. Another method was introduced by [DAP13] where the PDE was used to find the IMFs directly from the signal. Finally, in [WMV18], a method was introduced in which a PDE is used to determine the local mean directly from the signal. After trying the methods of [DAP13] and [WMV18], it was found that the method of [WMV18] is less hyperparameter sensitive. Additionally, [WMV18] pointed out some deficiencies of [DAP13] as mentioned in Chapter 2.

Therefore, the method of [WMV18] is outlined in this section. In this method, the process of finding the local mean of the signal by use of envelopes is replaced by a direct computation of the local mean. Here the local mean is the solution to the heat equation

$$\begin{aligned} \frac{\partial u}{\partial t} &= \alpha \frac{\partial^2 u}{\partial x^2}, \\ u(x,0) &= y(x) \end{aligned} \tag{5.8}$$

for $\alpha > 0$ which has solution $u(x,T)$ for $T > 0$. Using this heat equation, the solution $m(x) = u(x,T)$ can be used as the local mean in the SP shown in Algorithm 1.

The primary motivation for the initial value problem in Eq. (5.8) is that the local mean $m(x)$ passes through the inflection points of the signal since the right side of

Eq. (5.8) reduces to 0 at inflection points. Additionally, by the theorem of regularity of the solution of the heat equation, cf. Theorem 5.3, the local mean is smooth.

In order to use the algorithm, the parameter α needs to be determined. The choice of α is important to ensure that the local mean is contained within the range of the signal amplitude. One way to ensure this is by choosing

$$\alpha \leq \frac{1}{\omega_k^2(t)}$$

where ω_k is the instantaneous angular frequency of the kth IMF. Thus, to ensure this inequality is fulfilled choose

$$\alpha = \frac{1}{\omega_{\max}^2} \tag{5.9}$$

where $\omega_{\max} = \max_{k,t}\{\omega_k(t)\}$.

5.3.1 PDE Based Sifting Algorithm

Consider the case where some signal $\boldsymbol{y}$ with entries y_i for $i = 1, 2, \ldots, n$ is observed and consider the PDE introduced in Eq. (5.8). Taking offset in the procedure of Section 5.2.1, an algorithm to solve this PDE is introduced and is given the name PDE-EMD. The solution to the PDE is initialised as the signal, i.e. $u_i^0 = y_i$ for $i = 1, 2, \ldots, n$. As in Section 5.2.1, the following update scheme is used

$$u_i^{j+1} = u_i^j + \frac{\alpha \Delta t}{\Delta x^2}(u_{i-1}^j - 2u_i^j + u_{i+1}^j)$$

for $j \geq 0$ and $i = 1, \ldots, n$ where Δt and Δx are step sizes in the time domain and spatial domain, respectively.

Using matrix-vector notation, this can be written as

$$\boldsymbol{u}^{j+1} = (\boldsymbol{I} - \frac{\alpha \Delta t}{\Delta x^2}\boldsymbol{A})\boldsymbol{u}^j \tag{5.10}$$

where $\boldsymbol{u}^j = (u_1^j, u_2^j, \ldots, u_n^j)^T$. Since the signal is observed in a finite observation window, boundary conditions should be introduced. If the Dirichlet boundary conditions $u_0^j = u_{n+1}^j = 0$ are used, the matrix $\boldsymbol{A}$ is the same as in Eq. (5.6). In practice, the structure of $\boldsymbol{A}$ depends on the choice of boundary conditions and some variations of $\boldsymbol{A}$ are introduced shortly, specifically for Dirichlet boundary conditions and Neumann boundary conditions.

The stability condition for the update scheme in Eq. (5.10) follows analogously from Eq. (5.7) with

$$\frac{\alpha \Delta t}{\Delta x^2} \leq \frac{1}{2} \tag{5.11}$$

ensuring stability.

Boundary Conditions

The previously shown $\boldsymbol{A}_{\text{D0}}$ matrix corresponds to the Dirichlet conditions with the boundaries given as $u_0^j = u_{n+1}^j = 0$ for $j = 1, \ldots, \mathcal{T}$. This type of boundary conditions is referred to as D0. With this type of boundary conditions, the local mean equals 0 on the boundary which through the SP results in IMFs which equal the input signal to the SP on the boundary.

Now consider the Dirichlet conditions where $u_1^j = y_1$ and $u_n^j = y_n$, then

$$\boldsymbol{A}_{\text{D1}} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix}. \tag{5.12}$$

These boundary conditions are referred to as D1. Using this type of boundary conditions results in a local mean which equals the signal on the boundary. This in turn implies that the IMFs equals zero on the boundary.

The Neumann boundary conditions are given by setting a criterion on the derivative on the boundary. One such example is

$$\frac{\partial u(x_1, t_j)}{\partial x} = 0, \quad \frac{\partial u(x_n, t_j)}{\partial x} = 0.$$

In practice, these derivatives are approximated by a central difference. As such, these Neumann boundary conditions are given as

$$\frac{u_0^j - u_2^j}{\Delta x} = 0 \Rightarrow u_0^j = u_2^j$$

and similarly

$$u_{n+1}^j = u_{n-1}^j.$$

This gives the following structure of $\boldsymbol{A}$

$$\boldsymbol{A}_{\text{N0}} = \begin{pmatrix} 2 & -2 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -2 & 2 \end{pmatrix}. \tag{5.13}$$

These boundary conditions are referred to as N0 and results in a local mean that extends as a horizontal line on the boundary.

Another type of Neumann conditions are given by

$$\frac{\partial u(x_1, t_j)}{\partial x} = \frac{\partial u(x_2, t_j)}{\partial x}, \quad \frac{\partial u(x_n, t_j)}{\partial x} = \frac{\partial u(x_{n-1}, t_j)}{\partial x}.$$

Using these conditions and approximating the derivatives by central differences give the following structure of $\boldsymbol{A}$

$$\boldsymbol{A}_{\mathrm{N1}} = \begin{pmatrix} -1 & 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \ddots & \vdots \\ 0 & -1 & 2 & -1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 & -1 \end{pmatrix}. \tag{5.14}$$

These boundary conditions are referred to as N1. This type of boundary conditions leads to a local mean which extends in a straight line at the boundary.

Practical Considerations

In order to execute the algorithm, the parameters T, α, Δt, and Δx must be determined. In practice, the value of Δx is determined by the sample rate since if the sampled signal includes n samples from a signal on the interval $[0, K]$, then

$$\Delta x = \frac{K}{n}.$$

In order to choose α, Eq. (5.9) is used. However, in order to do this, $\omega_{\max}$ must be approximated. This is done by determining the positions of the local minima of the signal $\boldsymbol{y}$ and then the maximum frequency is approximated as the reciprocal of the minimum distance between two local minima which is denoted $\hat{f}_{\max}$ and α is then given by

$$\alpha = \frac{1}{(2\pi \hat{f}_{\max})^2} = \frac{1}{\hat{\omega}_{\max}^2} \tag{5.15}$$

where $\hat{\omega}_{\max} = 2\pi \hat{f}_{\max}$. However, if two minima are not present in the current signal, we set $f_{\max} = 1$ which is chosen since in this case the signal must be of low frequency. Using α, then Δt is determined as

$$\Delta t = \frac{K^2}{4\alpha n^2} = \frac{\Delta x^2}{4\alpha} \tag{5.16}$$

which is chosen as it ensures stability of the update scheme, cf. Eq. (5.11), and it gives the local update formula

$$u_i^{j+1} = \frac{1}{2} u_i^j + \frac{1}{4}\left(u_{i+1}^j + u_{i-1}^j\right) \tag{5.17}$$

when inserted into Eq. (5.10).

The final hyperparameter T has to be chosen empirically. The choice of T determines the number of updates in time as the number of finite difference steps is

$$\mathcal{T} = \left\lfloor \frac{T}{\Delta t} \right\rfloor.$$

Setting a low value of T gives a low value of $\mathcal{T}$ which results in a lower computational complexity but a worse approximation of the local mean. The opposite is the case with a higher value of T except if $\boldsymbol{u}^j$ has converged in which case more updates do not yield a different local mean.

The update scheme in Eq. (5.17) continues until $j = \mathcal{T}$. Then the local mean is set as $\boldsymbol{m} = \boldsymbol{u}^{\mathcal{T}}$ from which the extracted IMF is computed as $\boldsymbol{c}_1 = \boldsymbol{y} - \boldsymbol{m}$ and the first residual is computed as $\boldsymbol{r}_1 = \boldsymbol{y} - \boldsymbol{c}_1$. The next IMF is then determined by using $\boldsymbol{r}_1$ as the initial condition to the finite difference scheme Eq. (5.10). This iterative method for extracting IMFs is continued until the residual is monotone or one of the following stopping criteria on the residual is met

$$\frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n} m_i}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} r_{k,i}}} < \tau_1, \qquad \max_{i=1,\dots,n} r_{k,i} - \min_{i=1,\dots,n} r_{k,i} < \tau_2, \qquad \sum_{i=1}^{n} |r_{k,i}| < \tau_3$$

for the kth residual obtained using the PDE-EMD and for 3 thresholds $\tau_1, \tau_2, \tau_3 > 0$.

The PDE-EMD method is summarised in Algorithm 3.

Algorithm 3 PDE-EMD

Input: Signal $\boldsymbol{y}$, boundary type, thresholds, T, sample rate f_s.
1: Initialise $\boldsymbol{u}^0 = \boldsymbol{y}$; $\boldsymbol{r}_1 = \boldsymbol{y}$; $\Delta x = \frac{1}{f_s}$; IMFs = $\{\}$; $j = 0$; $k = 1$.
2: Determine $\boldsymbol{A}$ based on the type of boundary conditions.
3: **while** a stopping criterion is not fulfilled **do**
4: Determine the positions of the minima of $\boldsymbol{r}_k$.
5: Find $\hat{\omega}_{\max}$.
6: Determine α using Eq. (5.15).
7: Compute Δt as Eq. (5.16).
8: Compute $\boldsymbol{u}^j = (\boldsymbol{I} - \frac{\alpha \Delta t}{\Delta x^2}\boldsymbol{A})\boldsymbol{u}^{j-1}$ for $j = 1, \dots, \mathcal{T}$.
9: $\boldsymbol{c}_k = \boldsymbol{r}_k - \boldsymbol{u}^{\mathcal{T}}$
10: IMFs = IMFs $\cup \{\boldsymbol{c}_k\}$
11: Update the residual $\boldsymbol{r}_{k+1} = \boldsymbol{y} - \sum_{i=1}^{k} \boldsymbol{c}_i$.
12: $k \leftarrow k + 1$.
13: **end while**
14: $\boldsymbol{r} = \boldsymbol{r}_k$.
Output: IMFs, $\boldsymbol{r}$.

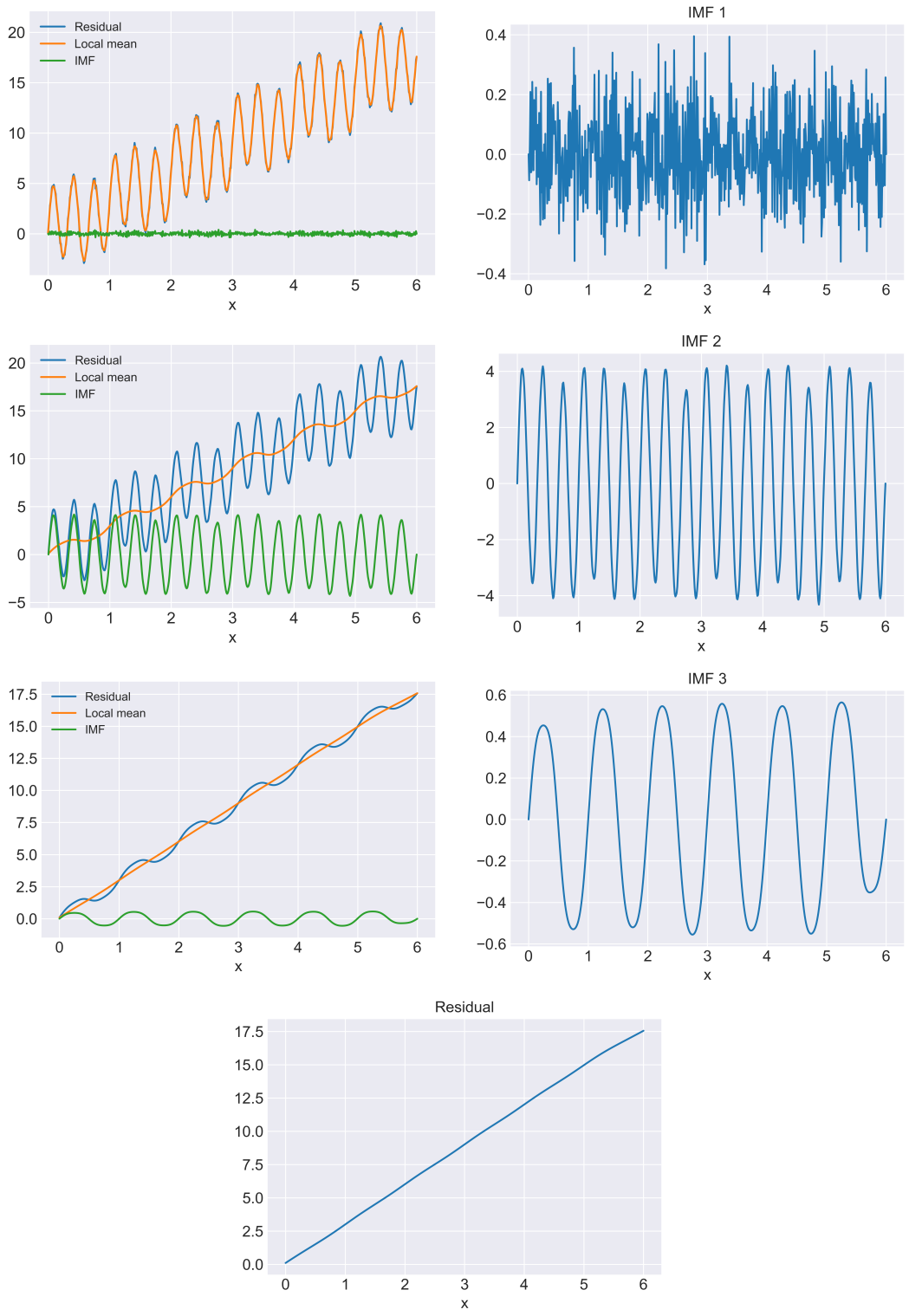


Figure 5.1: Depiction of the SP using the PDE-EMD. Example with a signal given by Eq. (5.18) and decomposition performed with $T = 6$. The procedure is depicted on the left and the resulting IMF on the right. Finally, the residual can be seen. The example has been made with D1 boundary conditions.

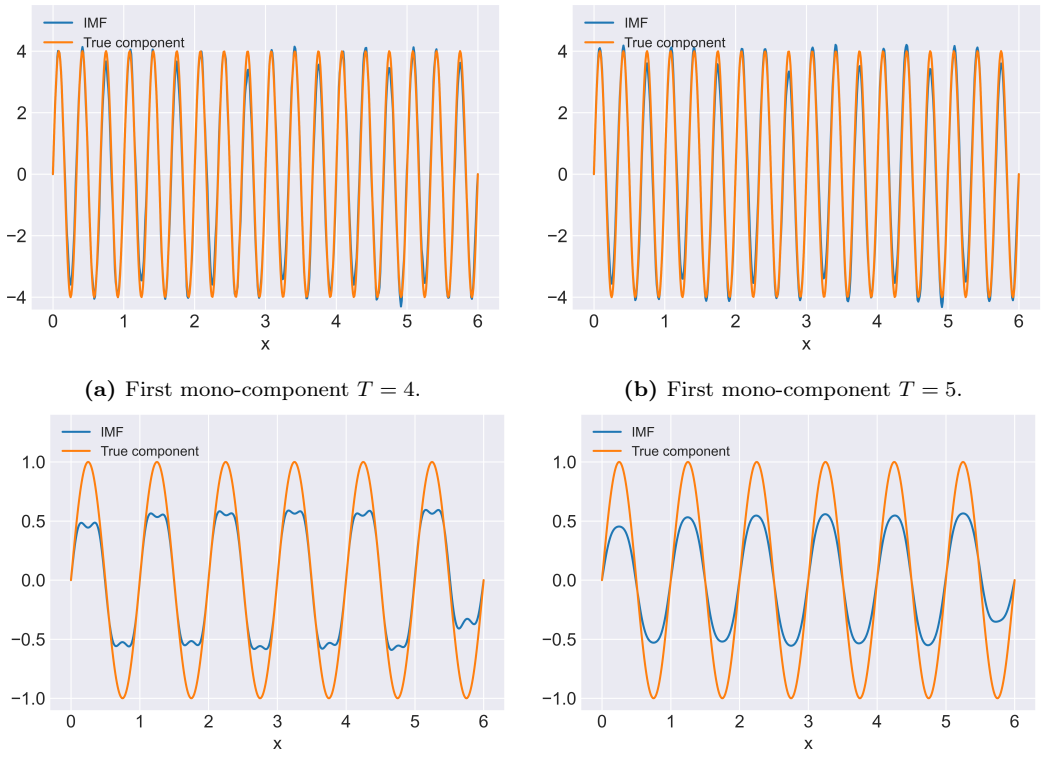
5.4 Synthetic Example

In this section, synthetic experiments with Algorithm 3 are performed. Specifically, the importance of the T parameter is tested on a simulated example.

Consider the following signal

$$y(x) = 4\sin(6\pi x) + \sin(2\pi x) + 3x + \varepsilon \tag{5.18}$$

where ε Gaussian white noise process with an SNR of 35 dB and with 576 equidistantly spaced datapoints for $x \in [0, 6]$. In Fig. 5.1, the SP of the PDE-EMD method is depicted. Here it is seen that a local mean is fitted by solving the PDE and then by subtracting the local mean from the signal, an IMF is extracted. Moreover, it is noticed that the procedure extracts all four components. However, both mono-components have a diminished amplitude and these parts of the signal are contained in the noise component found by the algorithm. Now the effect of T is considered.



(a) First mono-component $T = 4$.

(b) First mono-component $T = 5$.

(c) Second mono-component $T = 4$.

(d) Second mono-component $T = 5$.

Figure 5.2: An example showcasing the effect of T. The example has been made with the D1 boundary conditions.

The value of T should be chosen such that the finite difference scheme has converged, however setting T too high might result in an unnecessarily high amount of computations. In Fig. 5.2 the resulting decomposition for $T = 5$ and $T = 4$ can be seen

with the signal defined in Eq. (5.18). Looking at these figures, it is noticed that the PDE-EMD method seems to be able to find the first mono-component with these settings and the frequency of the second mono-component has also been captured precisely in the case $T = 5$ but with an artefact fluctuations around the maxima for $T = 4$. However, for the second mono-component the amplitude is attenuated for both values of T which in turn results in a residual with small fluctuations. The results for $T = 6$ and $T = 7$ are very similar to the results for $T = 5$ whereas for $T > 7$ the decomposition only returns a single component and the second component becomes part of the residual. Finally, setting $T < 4$ results in more than 2 mono-components, several of which have a frequency similar to the first mono-component but an attenuated amplitude. These tests suggest that $T = 5$ is a good choice for this specific example. However, an initial test for the choice of T should be made before this decomposition method is used.

Next the effect on the computational complexity as T gets larger is investigated. The effects of T is explored in relation to computation time and the number of extracted components. Looking at Fig. 5.3, the computation time for different choices of T as well as the amount of components which has been extracted is seen. It is noticed that increasing T does not necessarily increase the computation time. This is the case since the computation time not only depends on T but also the number of components and the α parameter for the components.

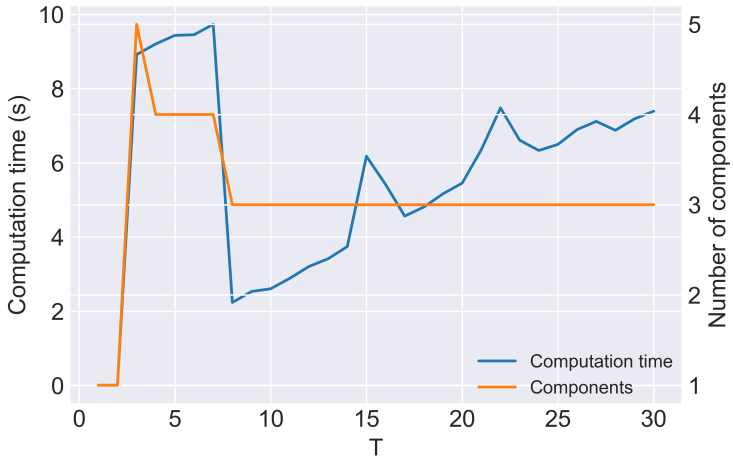


Figure 5.3: The effect of T on the computation time and the amount of components.

6. Experimental Setup

In this chapter, the setting in which the experiments are conducted is described. In Section 6.1, the data used for the experiments are introduced and in Section 6.2, the idea of decomposition based forecasting is introduced in a setup where the data is windowed. Then, in Section 6.3, performance measures for the adaptive decomposition methods are introduced and in Section 6.4, the deep neural network (DNN) forecast model used in this thesis is presented. Finally, in Section 6.5, the supervised learning setup which is used to train and test the forecasters is described.

6.1 Data Description

The data used in this thesis has been supplied by Energinet and consists of the wind power production data for each of the 21 sub-grids described in Section 1.2 given in 5 minute intervals in the time period from January 1st 2019 to October 5th 2021. Furthermore, the installed wind power production capacity for each sub-grid is available. We note that additional data such as weather forecasts has been supplied by Energinet, however due to differences in resolution and because the interpretation of the weather forecasts are unclear after the data has been decomposed, these are not be included in the forecasting models. Furthermore, we have previously seen good performance with an EMD based model which did not use weather predictions [AVK21, pp. 71-72].

For the experiments, we only consider a single sub-grid which is sub-grid 1 in DK1, see Fig. 1.2. The reasoning for this decision is to reduce the computation time and hyperparameter fitting time and additionally we deemed that only considering a single sub-grid is sufficient to answer the problem statement.

The wind power production data for sub-grid 1 in DK1 is denoted as the time series $\{P(t)\}_{t=1}^{n}$ where n is the number of datapoints. Note that the wind power production data has been collected using a data acquisition process in which sensor measurements from some wind turbines in sub-grid 1 in DK1 are used to approximate the actual wind power production. This process is referred to as SCADA-upscaling.

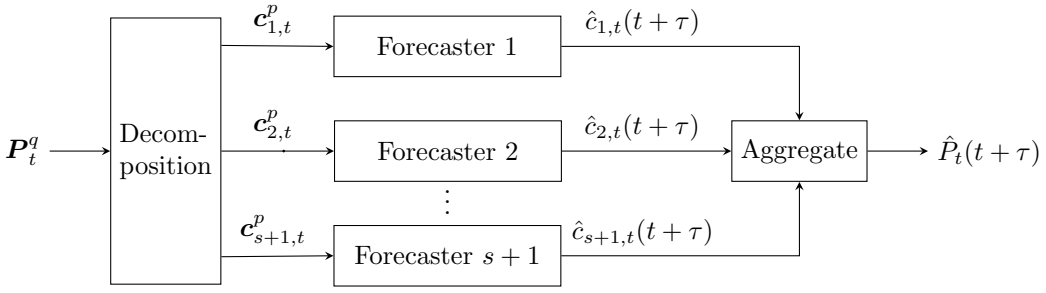


Figure 6.1: Block diagram of a decomposition based forecasting method.

6.2 Decomposition Based Forecasting

Decomposition based forecasting is motivated by a need to increase the predictability of data for forecasting. The purpose of the decomposition method as a pre-processing step is then to extract components from the signal which are easier to forecast than the signal itself. In this thesis, adaptive decomposition methods are considered and as such it is assumed that the wind power production can be expressed as

$$P(t) = \sum_{k=1}^{s} c_k(t) + r(t)$$

where c_k for $k = 1, \ldots, s$ are mono-components, r is called the residual, and we define $c_{s+1} = r(t)$ for ease of notation. For further details regarding adaptive decomposition methods and the concept of mono-components see Section 2.1. In general, the components resulting from the decomposition can vary depending on three aspects which are the decomposition method, the hyperparameters, and the data used to extract the components.

As mentioned in Section 1.3.2, the decomposition of the wind power production is computed for windows of length q, i.e. at time t the window of wind power data given by the datapoints $\boldsymbol{P}_t^q = \big(P(t), P(t-1), \ldots, P(t-q+1)\big)^T$ is decomposed into components $\boldsymbol{c}_{k,t}^q = \big(c_{k,t}(t), c_{k,t}(t-1), \ldots, c_{k,t}(t-q+1)\big)^T$ for $k = 1, \ldots, s+1$ such that

$$P(t) = \sum_{k=1}^{s+1} c_{k,m}(t)$$

for $m = t, \ldots, t+q-1$. Note that $\boldsymbol{P}_t^q$ is referred to as the wind power data for window t and $\boldsymbol{c}_{k,t}^q$ is the data of component k for window t. Using the wind power data in window t, a forecast for time $t+\tau$ is made for each of the components and the result is aggregated to obtain a forecast for the wind power production. The forecast is based on using the p most recent datapoints with $p \leq q$, i.e. $\boldsymbol{c}_{k,t}^p$.

The forecast of the components for time $t+\tau$ given time t is denoted $\{\hat{c}_{k,t}(t+\tau)\}_{t=q}^{n}$ for $k = 1, \ldots, s+1$ and the forecast is made as

$$\hat{c}_{k,t}(t+\tau) = f_k(\boldsymbol{c}_{k,t}^{p}; \phi_k)$$

where f_k are forecasters parameterised by ϕ_k. Hence, the forecast of the wind power production for time $t+\tau$ given time t is

$$\hat{P}_t(t+\tau) = \sum_{k=1}^{s+1} \hat{c}_{k,t}(t+\tau). \tag{6.1}$$

This is shown schematically in Fig. 6.1.

6.3 Performance Measures of Decompositions

In this section, performance measures for the adaptive decomposition methods are introduced. These measures are related to end effects, orthogonality, consistency of the decomposition when windowing data, and the uncertainty of the extracted components, respectively. While the actual forecasting performance is the seminal quantity for a forecasting application, the indirect performance measures introduced in this section are used to evaluate the success of a decomposition in Chapter 7 without having to train a forecaster.

6.3.1 End Effect Evaluation Index

In order to test whether or not the implemented methods alleviate end effects, the so-called end effect evaluation index (EEEI) is used. The EEEI quantifies the end effects in terms of the energy before and after decomposition. This is done based on the idea that a decomposition with end effects produces false mono-components which then results in an energy change before and after decomposition. In general, the root mean squared of the signal $\boldsymbol{P}_t^q$ for $t = 1, \ldots, n$ can be computed as

$$E_{\boldsymbol{P}_t^q} = \sqrt{\frac{1}{q}\sum_{j=0}^{q-1} P(t-j)^2}.$$

Now it is assumed that $\boldsymbol{P}_t^q$ has been decomposed into $s+1$ components $\boldsymbol{c}_{k,t}^q$ for $k = 1, \ldots, s+1$. Then for a decomposition made for a window of size q at time t the EEEI is computed as

$$\mathrm{EEEI}(t) = \frac{\left| \sqrt{\sum_{k=1}^{s+1} E_{\boldsymbol{c}_{k,t}^q}^2} - E_{\boldsymbol{P}_t^q} \right|}{E_{\boldsymbol{P}_t^q}} \tag{6.2}$$

where $E_{\boldsymbol{c}_{k,t}^q}$ is the root mean squared of the kth component for $k = 1, \ldots, s+1$. In general, a large EEEI is associated with a low decomposition precision and large end

effects, whereas EEEI = 0 means the end effects are minimal [HYX15]. The average EEEI across different windows is given as

$$\overline{\text{EEEI}} = \frac{1}{n-q+1}\sum_{t=q}^{n}\text{EEEI}(t)$$

thereby expressing how good the performance of the decomposition is in terms of EEEI on average.

6.3.2 Index of Orthogonality

Another performance measure is the index of orthogonality (IO). This performance measure is motivated by the fact that ideally the mono-components should be orthogonal to avoid redundancy. Thus, the inner product between two components should ideally be zero. However, since the decomposition is not perfect, this is not the case. The performance measures which can be used are the maximum index of orthogonality (MIO) and the average index of orthogonality (AIO), respectively. Before introducing the IO, the Pearson correlation coefficient (PCC) is introduced. The PCC for a vector $\boldsymbol{c}$ is computed as

$$\text{PCC}\{\boldsymbol{c}_j, \boldsymbol{c}_k\} = \frac{\langle \boldsymbol{c}_j, \boldsymbol{c}_k\rangle}{\|\boldsymbol{c}_j\|_2\|\boldsymbol{c}_k\|_2}.$$

The PCC returns a number between -1 and 1 where -1 is complete negative correlation, 0 is uncorrelated, and 1 is complete positive correlation. The IO is simply the absolute value of the PCC. Hence, for a decomposition made for a window of size q at time t, the MIO and AIO are given by

$$\text{MIO}(t) = \max_{k\neq j}|\text{PCC}\{\boldsymbol{c}_{j,t}^q, \boldsymbol{c}_{k,t}^q\}|, \tag{6.3}$$

$$\text{AIO}(t) = \frac{1}{s^2(s+1)/2}\sum_{j=1}^{s+1}\sum_{k=1}^{j-1}|\text{PCC}\{\boldsymbol{c}_{j,t}^q, \boldsymbol{c}_{k,t}^q\}|. \tag{6.4}$$

Both of these performance measures return a number in the interval $[0, 1]$. Ideally, both of these should be zero and an IO close to one indicates redundancy in the decomposition [HS14, p. 85]. When decomposing data in many windows, the sample mean of both the MIO and the AIO are defined as

$$\overline{\text{MIO}} = \frac{1}{n-q+1}\sum_{t=q}^{n}\text{MIO}(t), \quad \overline{\text{AIO}} = \frac{1}{n-q+1}\sum_{t=q}^{n}\text{AIO}(t).$$

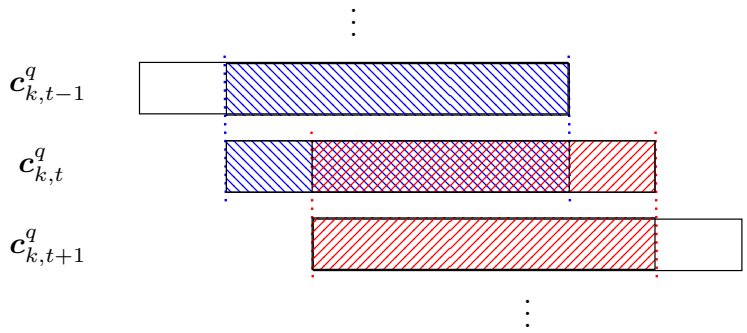


Figure 6.2: Block diagram illustrating the computation of the CPM for time shift $h = 1$.

6.3.3 Consistency

When windowing a dataset and then decomposing each window, there is no guarantee that the decompositions of adjacent windows are similar. In a forecasting application, however, this is a desirable property. Hence, we introduce a consistency performance measure (CPM) given as the following for component k and time shift h

$$\mathrm{CPM}_{k,h} = \frac{1}{n-q-2h+1} \sum_{t=q+h}^{n-h} \left(\frac{\|\boldsymbol{c}^q_{k,t}(t-h:t-q+1) - \boldsymbol{c}^q_{k,t-h}(t-h:t-q+1)\|_2}{\|\boldsymbol{c}^q_{k,t}(t-h:t-q+1)\|_2} + \frac{\|\boldsymbol{c}^q_{k,t+h}(t:t-q+1+h) - \boldsymbol{c}^q_{k,t}(t:t-q+1+h)\|_2}{\|\boldsymbol{c}^q_{k,t}(t:t-q+1+h)\|_2} \right) \tag{6.5}$$

for $k = 1, \ldots, s+1$ and $h = 0, \ldots, q-1$ where q is the window length and with the notation

$$\boldsymbol{c}^q_{k,t}(t-a:t-b) = \big(c_{k,t}(t-a), c_{k,t}(t-a-1), \ldots, c_{k,t}(t-b)\big)^T$$

for $a, b \in \{0, \ldots, q-1\}$ and $a < b$. The CPM quantifies how similar components at the same time-index resulting from different data windows are. This is illustrated in Fig. 6.2 where the colours indicate which parts of the adjacent components are compared. The CPM should ideally be zero and assuming that the amplitude scale of the extracted components for the different windows are similar, then a poor result is a value near 4. We define the average of $\mathrm{CPM}_{k,h}$ across the components as

$$\overline{\mathrm{CPM}}_h = \frac{1}{s+1} \sum_{k=1}^{s+1} \mathrm{CPM}_{k,h}.$$

6.3.4 Sample Entropy

Since the purpose of applying an adaptive decomposition method as a pre-processing step when forecasting is to increase the predictability, a quantitative performance measure designed for measuring the predictability is introduced. The term predictability refers to the existence of patterns within a time series and is related to the terms randomness, uncertainty, and complexity. The sample entropy (SampEN) is used to quantify the uncertainty of the time series. Intuitively, a low SampEN implies that a signal is repetitive and predictive, while a high SampEN indicates a low amount of repeated patterns and a high amount of randomness. [DBM19]

Consider a discrete time series $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ and let

$$\boldsymbol{y}_i^m = (y_i, y_{i+1}, \ldots, y_{i+m-1})^T$$

for $i \leq n - m + 1$ be a block of length m. The SampEN is defined as the negative logarithm of the empirical conditional probability that two blocks of length $m+1$ are similar given that two blocks of length m are similar. Whether two blocks are similar is determined by the Chebyshev distance defined as

$$\nu(\boldsymbol{y}_i^m, \boldsymbol{y}_j^m) = \max_{k=0,\ldots,m-1} |y_{i+k} - y_{j+k}|$$

for $i, j \leq n - m + 1$. The Chebyshev distance returns the largest difference between two points in the blocks $\boldsymbol{y}_i^m$ and $\boldsymbol{y}_j^m$. Two blocks are said to be similar if their Chebyshev distance is less than some tolerance $r > 0$.

The empirical probability that any two blocks of length m are similar within the tolerance r is computed as

$$B^m(\boldsymbol{y}; r) = \frac{1}{n-m} \sum_{i=1}^{n-m} \Big(\frac{1}{n-m-1} \sum_{\substack{j=1 \\ j \neq i}}^{n-m} \mathbb{1}[\nu(\boldsymbol{y}_j^m, \boldsymbol{y}_i^m) < r] \Big).$$

By construction, $B^{m+1}(\boldsymbol{y}; r) \leq B^m(\boldsymbol{y}; r)$ and the SampEN is defined as [DBM19, pp. 11-20]

$$\text{SampEn}(m; r)\{\boldsymbol{y}\} = -\log\Big(\frac{B^{m+1}(\boldsymbol{y}; r)}{B^m(\boldsymbol{y}; r)} \Big). \tag{6.6}$$

A popular choice for r is $r = 0.2\sigma$ where σ is the sample standard deviation of the time series and this choice is also used in this thesis. [RM00]

Considering component k in a decomposition when windowing the components in windows of length q, the average SampEN across all windows is defined as

$$\overline{\text{SampEN}}_k(m; r) = \frac{1}{n-q+1} \sum_{t=q}^{n} \text{SampEN}(m; r)\{\boldsymbol{c}_{k,t}^q\}.$$

6.4 Neural Network Forecaster

For the forecaster, a long short-term memory (LSTM) neural network is used which has been chosen due to its abilities for modelling temporal dependencies. For a description of the forward propagation equations of an LSTM layer see Appendix B.1, and for further insight regarding LSTM neural networks see [GBC16, pp. 397-400]. From this, a forecasting model is obtained for the wind power production for each of the decomposition methods.

A block diagram of the LSTM neural network can be seen in Fig. 6.3. This notation indicates the baseline LSTM model without using a decomposition method as a pre-processing step. With the decomposition based models using an LSTM forecaster, the input to the LSTM is $\boldsymbol{c}_{k,t}^{p}$ and the output is $\hat{c}_{k,t}(t+\tau)$.

Figure 6.3: Block diagram of the LSTM model.

In the following, the hyperparameters associated with the LSTM neural network used in this thesis are described. The LSTM neural networks take p samples as input in order to make a one hour forecast, i.e. 12 steps ahead forecast. The LSTM neural network has a number of hidden layers and a number of hidden units or cells in each hidden layer. During training, an optimiser which is a variation of stochastic gradient descent called root mean squared propagation is used to optimise the network [GBC16, pp. 299-300]. In this method, batches are sampled from the training data and is used to update the parameters of the neural network based on gradient information derived using back propagation [GBC16, pp. 197-211]. This is done by updating the parameters in the direction of the negative gradient with a step size called the learning rate. The size of the batches is called the batch size and choosing a larger batch size gives a more stable estimate of the gradient, whereas a smaller batch size can have a regularising effect which in turn can lead to better generalisation. However, a smaller batch size requires a smaller learning rate in order to maintain stable training leading to an increase in training time [GBC16, p. 272]. Typically, each available datapoint is used multiple times during training and the term epoch is used to refer to a complete pass through the training data. For each epoch that has been completed, the learning rate is updated by a multiplicative factor which is the so-called learning rate decay.

In terms of regularisation, we use both early stopping and dropout. Early stopping is a regularisation technique which works by letting the network train on a sub-training set until the loss on a validation set has not improved for a number of iterations referred to as the patience. When the validation loss has not improved for the patience number of iterations, the training is terminated and the total number of training iterations is saved. Then the network is retrained on the entire training set, i.e. the union of the sub-training set and the validation set, for as many iterations as was determined by the early stopping procedure [GBC16, pp. 239-245]. In practice, it

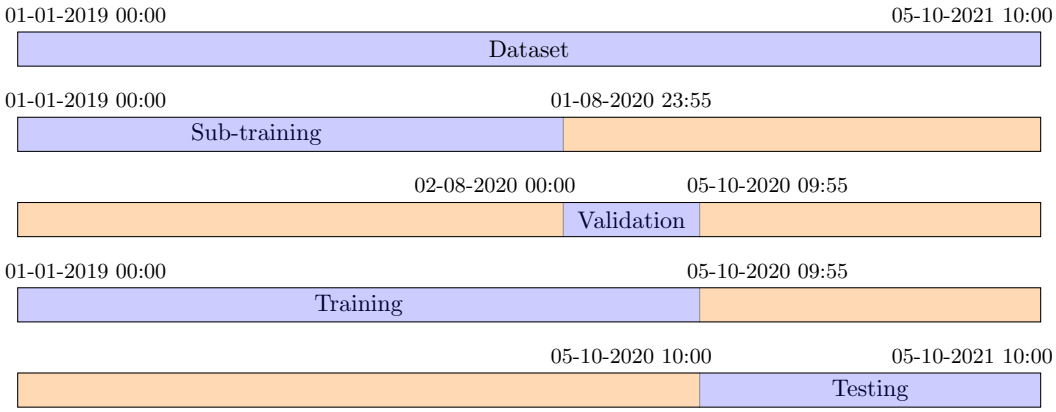


Figure 6.4: Partitioning of the dataset into the four datasets.

is computationally expensive to compute the validation loss each time the parameters of the network are updated and therefore the validation loss is only computed at an interval. Finally, as mentioned, we use dropout regularisation. Dropout works by excluding some of the cells in the hidden layers in a training iteration removing connections to and from this cell. The dropout rate refers to the likelihood that a cell is dropped in a training iteration and as such if the dropout rate is 0.1, it is expected that approximately 10% of the hidden layer cells are dropped in some training iteration. After the training iteration has finished, the dropped cells and their corresponding weights and biases are reintroduced into the model. [GBC16, pp. 251-261]

6.5 Supervised Learning Setup

As mentioned in the Section 6.1, the data which is used in this thesis is the wind power production data in the time period from January 1st 2019 at 00:00 to October 5th 2021 at 10:00 for sub-grid 1 in DK1. The dataset has been divided into two subsets, i.e. a training set and a test set. The training set consists of the wind power production data from the time period lasting from January 1st 2019 at 00:00 to October 5th 2020 at 09:55 and the test set consists of the data during the time period lasting from October 5th 2020 at 10:00 to October 5th 2021 at 10:00. The training set has been further divided into a sub-training set and a validation set in order to choose the hyperparameter setting before evaluating on the test set. The sub-training set consists of the data during the time period from January 1st 2019 at 00:00 to August 1st 2020 at 23:55 and the validation set consists of the data from August 2nd 2020 at 00:00 to October 5th 2020 at 09:55. The partitioning of the data is depicted in Fig. 6.4.

6.5.1 Training

The forecast of the wind power production is as seen in Eq. (6.1) expressed as the sum of the components. Therefore, a forecaster is trained for each of the components as illustrated in Fig. 1.3. To pose the training in a supervised learning setup, target values are necessary, however, when considering decomposition based forecasting models the assignment of target value is non-trivial. This is caused by the fact that the components extracted by adaptive decomposition methods depends on the data and as such when considering different parts of a dataset, the decomposition results differ. In [Qia+19], three methods of assigning targets used in the literature are outlined.

With the first method, the entire training data is decomposed jointly thereby giving access to target values. Then during testing, every time a new datapoint is observed a new decomposition is computed using all the data up to the current time. This method is costly in terms of memory and computation time. [SPC20]

With the second method, at each time t a window of the most recent observations are used to compute a decomposition. Then when forecasting time $t+\tau$ given time t, the target value is given as $c_{k,t+\tau+\xi}(t+\tau)$, i.e. the component value at time $t+\tau$ obtained from decomposing window $t+\tau+\xi$ where $\xi \in \mathbb{N}$. The inclusion of the ξ parameter is motivated by the possibility of end effects as $c_{k,t+\tau}(t+\tau)$ is at the end of a window and as such is expected to be subject to more end effects than $c_{k,t+\tau+\xi}(t+\tau)$ for $\xi > 0$.

A third method is to use the data itself directly as the target, however, this implies training the forecasters for all the components jointly which can be detrimental to successful training.

In this thesis, the second method is used to assign targets due to its computational and memory efficiency and due to its DNN training advantages. The method is depicted in Fig. 6.5. For training, the mean squared error (MSE) is used as the loss function which with this method is given by

$$\text{MSE} = \frac{1}{n_{\text{train}} - q - \tau - \xi + 1} \sum_{t=q}^{n_{\text{train}}-\tau-\xi} \left(c_{k,t+\tau+\xi}(t+\tau) - \hat{c}_{k,t}(t+\tau)\right)^2$$

where n_{train} is the number of datapoints in the training set, q is the window length, τ is the forecast horizon in samples, $\hat{c}_{k,t}(t+\tau)$ is the forecast at time $t+\tau$ given time t, $c_{k,t+\tau+\xi}(t+\tau)$ is the target value, and ξ is the window shift used to assign the target value.

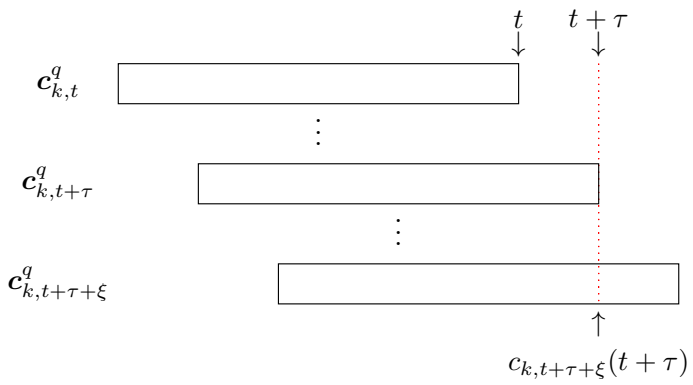


Figure 6.5: Block diagram depicting the choice of target value.

6.5.2 Testing

To evaluate the performance of the forecasting models, we use the normalised root mean squared error (NRMSE), the normalised mean absolute error (NMAE), and the normalised bias (NBIAS) given by

$$\text{NRMSE} = \sqrt{\frac{1}{n_{\text{test}} - q - \tau + 1} \sum_{t=n_{\text{train}}+1+q}^{n-\tau} \frac{(P(t+\tau) - \hat{P}_t(t+\tau))^2}{P^2_{\text{capacity}}}},$$

$$\text{NMAE} = \frac{1}{n_{\text{test}} - q - \tau + 1} \sum_{t=n_{\text{train}}+1+q}^{n-\tau} \frac{|P(t+\tau) - \hat{P}_t(t+\tau)|}{P_{\text{capacity}}},$$

$$\text{NBIAS} = \frac{1}{n_{\text{test}} - q - \tau + 1} \sum_{t=n_{\text{train}}+1+q}^{n-\tau} \frac{P(t+\tau) - \hat{P}_t(t+\tau)}{P_{\text{capacity}}}$$

where P_{capacity} is the installed capacity, n_{test} is the number of datapoints in the test set, $P(t+\tau)$ is the actual wind power production at time $t+\tau$, and $\hat{P}_t(t+\tau)$ is the forecast of the wind power production at time $t+\tau$ given time t. These performance measures are used since they have been highlighted by Energinet. The NRMSE and NMAE return numbers between 0 and 1, while NBIAS return numbers between -1 and 1. The NRMSE is more sensitive to outliers than the NMAE and the NMAE is intuitively understandable. Moreover, the NBIAS gives the bias of the forecaster. The use of normalisation has the advantage of resulting in numbers that are comparable to other datasets and providing a percentage error interpretation. Additionally, the 95th percentile of the normalised error is found. Empirically, this is defined as determining ζ such that the following equality holds

$$\mathcal{P}_{\text{data}}\left(\frac{|P(t+\tau) - \hat{P}_t(t+\tau)|}{P_{\text{capacity}}} \leq \zeta\right) = 0.95$$

where $\mathcal{P}_{\text{data}}$ is the empirical probability defined by the data. The 95th percentile determines the upper bound on the absolute error when infrequent peaks are ignored which gives an idea of the maximum absolute error that can be expected in most cases.

To compare the forecast ability for each individual component, a performance measure dubbed variance scaled mean squared error (VSMSE) is used

$$\text{VSMSE}_k = \frac{\frac{1}{n_{\text{test}}-q-\tau-\xi+1} \sum_{t=n_{\text{train}}+1+q}^{n-\tau-\xi} \left(c_{k,t+\tau+\xi}(t+\tau) - \hat{c}_{k,t+\tau+\xi}(t+\tau)\right)^2}{\overline{\sigma}^2_{c_{k,t+\tau+\xi}}} \tag{6.7}$$

where $\overline{\sigma}^2_{c_{k,t+\tau+\xi}}$ is the sample variance defined as

$$\overline{\sigma}^2_{c_{k,t+\tau+\xi}} = \frac{1}{n_{\text{test}}-\tau-\xi} \sum_{t=n_{\text{train}}+1}^{n-\tau-\xi} \left(c_{k,t+\tau+\xi}(t+\tau) - \frac{1}{n_{\text{test}}-\tau-\xi} \sum_{t=n_{\text{train}}+1}^{n-\tau-\xi} c_{k,t+\tau+\xi}(t+\tau)\right)^2.$$

If the forecast is made as the mean of the component, then the VSMSE equals 1. Therefore, the VSMSE yields a number between 0 and 1 where 0 means that the forecaster has flawless accuracy and 1 means that the forecaster has not learned any patterns allowing it to increase the forecast accuracy.

7. Numerical Experiments

In this chapter, numerical experiments with the decomposition methods and forecasting models introduced in this thesis are presented. The experiments are designed with the purpose of answering the problem statement given in Section 1.4. As such, the main objective is to test whether or not a forecast model using a decomposition method as a pre-processing step can perform better than a set of baseline models in an online setup. The baseline models considered are an autoregressive method and an LSTM neural network with no decomposition as pre-processing. A brief introduction to these methods is given in Appendix B. The baseline methods are compared to decomposition based forecasting models where each component of the decomposition is forecasted using an LSTM neural network. The decomposition based models are made using the adaptive decomposition methods introduced in this thesis, i.e. the EMD, FFT-NMP-EMD, NMP-EMD, and PDE-EMD algorithms, respectively. Additionally, to see the potential of decomposition based forecasting models, an offline decomposition of the entire dataset using the EMD is included.

Firstly, experiments with the decomposition methods are made in Section 7.1 and afterwards, in Section 7.2, numerical experiments with forecasting the wind power production are conducted.

7.1 Decomposition Experiments

To see the pros and cons of the different adaptive decomposition methods and to motivate some relevant design choices before proceeding to the forecasting application, experiments are conducted with the methods on both synthetic data and the wind power production data.

7.1.1 Decomposition of Simulated Data

To compare the decomposition methods in a controlled setting, a qualitative comparison of the decomposition methods is given for a specific synthetic example. Afterwards, the tone separation capability of the methods is tested.

Two-Component AM-FM Signal

Consider a two-component AM-FM signal

$$\begin{aligned} y(t) =& \Big(\frac{5}{4} + \frac{1}{4}\cos(\pi t + \pi)\Big)\cos\big(2\pi(7t + t^2)\big) \\ &+ \Big(1 + \frac{1}{2}\cos(3\pi t)\Big)\cos\bigg(8\pi t + \frac{1}{2}\sin(\pi t)\bigg) + \varepsilon(t) \end{aligned} \tag{7.1}$$

where $t \in [0, 2]$ and ε is a Gaussian white noise process with an SNR of 50 dB. In the following, this signal is decomposed using the EMD, PDE-EMD, FFT-NMP-EMD, and NMP-EMD. The maximum number of IMFs extracted with each method is $s = 2$. With regards to the NMP-EMD and FFT-NMP-EMD methods, the smoothness parameter is set as $\lambda = 0.3$ with frequency initialised as the mean of the IF of the IMFs obtained from the EMD. For the PDE-EMD, T is set to 3 and the N0 boundary conditions given in Eq. (5.13) are used.

In order to quantify the error of the decompositions, the following performance measure is used

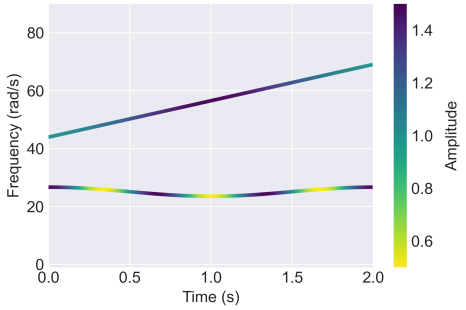
$$Q = \sum_{k=1}^{2} \frac{\|c_k - \tilde{c}_k\|_2}{\|c_k\|_2} \tag{7.2}$$

where c_1 is the high frequency component of y, c_2 is the low frequency component of y, and $\tilde{c}_1$, $\tilde{c}_2$ are the components extracted with a decomposition method. The value of Q should preferably be close to zero. [DSB19]

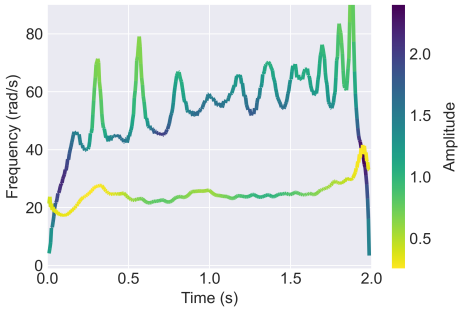
In Fig. 7.1, a qualitative comparison of the decomposition methods is given using the Hilbert-Huang spectrum. It is seen that the NMP-EMD method visually resembles the true spectrum the most. Moreover, the EMD and PDE-EMD methods have many fluctuations in the frequency domain. A quantitative comparison is given in Table 7.1 where it is seen that the NMP-EMD method has the best performance in terms of AIO, MIO, EEEI, and Q.

Decomposition	EMD	PDE-EMD	NMP-EMD	FFT-NMP-EMD
AIO	0.1304	0.6243	**0.04617**	0.09745
MIO	0.1788	0.7966	**0.09374**	0.1575
EEEI	0.03204	0.2491	**0.001576**	0.04360
Q	1.134	1.087	**0.7548**	0.9883

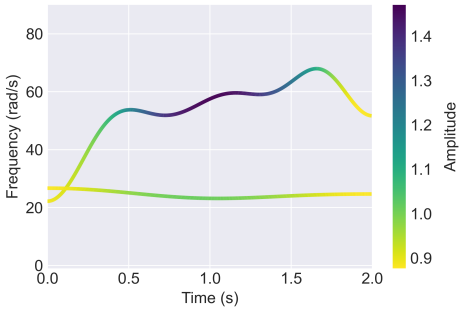
Table 7.1: Comparison of the EMD, PDE-EMD, NMP-EMD, and FFT-NMP-EMD methods in terms of the quantitative performance measures: AIO Eq. (6.4), MIO Eq. (6.3), EEEI Eq. (6.2), and Q Eq. (7.2).



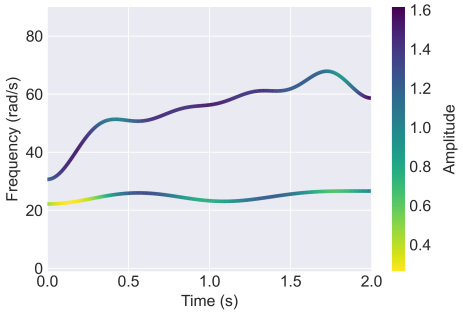
(a) True spectrum.



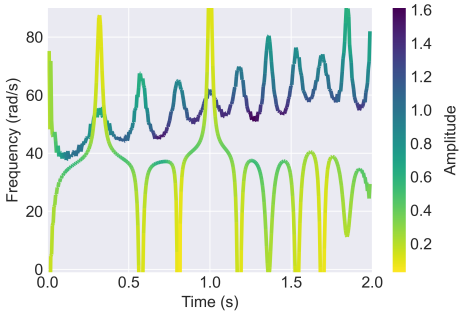
(b) EMD Hilbert-Huang spectrum.



(c) FFT-NMP-EMD Hilbert-Huang spectrum.



(d) NMP-EMD Hilbert-Huang spectrum.



(e) PDE-EMD Hilbert-Huang spectrum.

Figure 7.1: Comparison of the Hilbert-Huang spectrum of the signal defined by Eq. (7.1) for the different adaptive decomposition methods considered in this thesis.

Tone Separation Capability

In this section, the tone separation capabilities of the adaptive decomposition methods considered in this thesis are compared. This is done in correspondence with the method of [RF08]. The starting point is to consider the most general two-component signal with constant amplitudes, constant frequencies, and constant phase shifts given as

$$y(t) = a_1 \cos(2\pi f_1 t + \phi_1) + a_2 \cos(2\pi f_2 t + \phi_2)$$

for $t \in [0, K]$ where $K > 0$ is the length of the observation window, $a_1, a_2 > 0$ are the amplitudes, $f_1, f_2 > 0$ are the frequencies, and $\phi_1, \phi_2 \in [0, 2\pi]$ are phase shifts. This yields a total of 6 parameters. To simplify the analysis, it is assumed that the decomposition methods are only sensitive to the relative differences $a = \frac{a_1}{a_2}$ and $f = \frac{f_1}{f_2}$. Moreover, for simplicity of visualisation, we let $\phi_1 = \phi_2 = 0$. This reduces the parameters from 6 to 2. An example signal is then

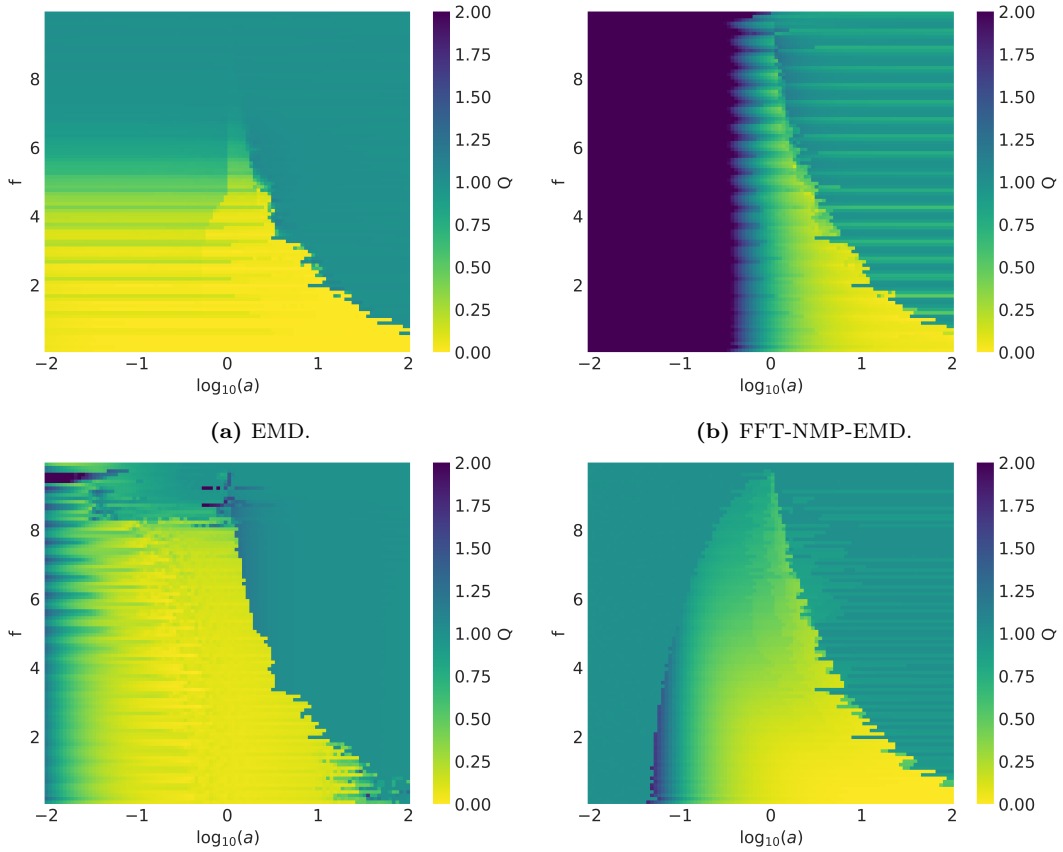
$$y(t; a, f) = \cos(20\pi t) + a \cos(2\pi f t) \tag{7.3}$$

which only depends on a where it is chosen that $a \in [0.01, 100]$ and f is chosen as $f \in (0, 10)$. Moreover, the length of the observation window is fixed as $K = 2$. The decomposition methods can be used in an attempt to separate the two components which $y(t)$ consists of. To quantitatively evaluate the performance of the separation, the following performance measure is used

$$Q(a, f; \tilde{c}_1) = \frac{\|\tilde{c}_1(t; a, f) - \cos(20\pi t)\|_2}{\|a \cos(2\pi f t)\|_2} \tag{7.4}$$

where $\tilde{c}_1(t)$ is the first IMF, i.e. the highest frequency component which has been extracted. When this performance measure is close to zero, the first IMF resembles the high frequency component of the signal and hence the method has successfully separated the two components. However, if the separation is poor, then the performance measure takes a value close to one due to the choice of denominator. [RF08]

In Fig. 7.2, the tone separation capabilities of the EMD, the FFT-NMP-EMD, the NMP-EMD, and the PDE-EMD methods are shown using the same hyperparameters for the decomposition methods as for the two-component AM-FM signal. Notably, the EMD can resolve the tones when $a \approx 0$ and $f < 5$. For the FFT-NMP-EMD method, it is noticed that it has a poor tone separation ability compared to the other methods for $a < 1$. The NMP-EMD performs well in a large frequency and amplitude area compared to the other methods, specifically, it is the only method with good tone separation capabilities for low amplitudes when $f > 5$. Finally, the PDE-EMD method, has a better capability to separate the tones when $0.1 < a < 10$ and $f > 5$ than what is achieved with the EMD.



(a) EMD.

(b) FFT-NMP-EMD.

(c) NMP-EMD.

(d) PDE-EMD.

Figure 7.2: Quantitative evaluation of tone separation using the performance measure defined in Eq. (7.4).

7.1.2 Decomposition of Wind Power Data

To apply the different decomposition methods effectively to the wind power production data, an analysis of the result of decomposing the wind power production data is made. In this context, design choices and selection of hyperparameters for the different decomposition methods are considered. This is done for the EMD, the FFT-NMP-EMD, the NMP-EMD, and the PDE-EMD methods. For all the methods, windows of length $q = 288$ are used due to computational complexity and memory complexity limitations. This is discussed further in the end of this section.

Empirical Mode Decomposition

The only hyperparameters related to the EMD are the stopping criteria and for this we employ the default settings used in the `EMD-signal` library in Python. However, some processing of the result of the EMD is done in order to increase the consistency of the EMD. This is done as described in Appendix C.1 and is referred to as the unification procedure. After this unification of the EMD result, the distribution of the number of components in the training data is as shown in Fig. 7.3. This result shows that the number of components changes depending on the window for which the EMD is computed. Moreover, typically 4 or 5 IMFs are present in the data, while the minimum number of IMFs in a window is 3.

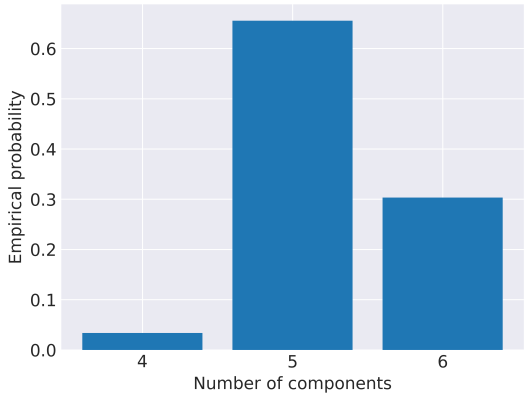


Figure 7.3: Bar chart displaying the distribution of the number of components extracted using the EMD.

FFT-NMP-EMD

When using the FFT-NMP-EMD algorithm, the sparsity is fixed based on the result with the EMD. This means that since the number of EMD IMFs is typically 4 or 5, the sparsity with the FFT-NMP-EMD method is initially fixed as $s = 5$. In Fig. 7.4, the average consistency performance measure $\overline{\text{CPM}}_h$ is given for the FFT-NMP-EMD

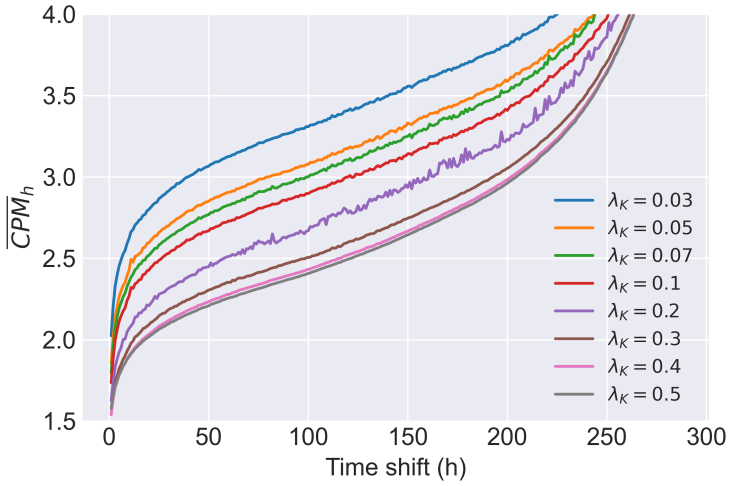


Figure 7.4: $\overline{\text{CPM}}_h$ of FFT-NMP-EMD for varying λ values.

with varying λ values on the training data. As a general trend, it is seen that a larger λ value gives smaller values of $\overline{\text{CPM}}_h$ and thereby a more consistent decomposition.

A quantitative comparison of the different decompositions according to the performance measures $\overline{\text{AIO}}$, $\overline{\text{MIO}}$, and $\overline{\text{CPM}}_{12}$ is given in Table 7.2. Note that $\overline{\text{CPM}}_{12}$ is considered since for one hour ahead forecasting $\tau = 12$. This table indicates that a smaller λ value is better in terms of the IO performance measures. The $\overline{\text{EEEI}}$ performance measure has also been considered, however no measurable difference has been seen for varying λ values. The choice of a λ value is then a trade-off between small IO and small CPM. As we suspect that the CPM is more relevant to the success of the forecasts, adhering to the trade-off, we choose $\lambda = 0.4$. This statement is analysed in further detail in Section 7.2.2.

λ	0.03	0.05	0.07	0.1	0.2	0.3	0.4	0.5
$\overline{\text{AIO}}$	**0.0268**	0.0277	0.0285	0.0302	0.0367	0.0446	0.0541	0.0650
$\overline{\text{MIO}}$	**0.1818**	0.1834	0.1843	0.1873	0.2019	0.2238	0.2497	0.2771
$\overline{\text{CPM}}_{12}$	2.677	2.476	2.402	2.315	2.101	1.991	1.935	**1.927**

Table 7.2: Comparison of decomposition performance measure with the FFT-NMP-EMD for varying λ values. Boldface indicates the best value for a given performance measure.

Due to the computational complexity of the NMP-EMD method, cf. Table 7.6, experimentation with hyperparameter settings has not been made for the NMP-EMD method. Therefore, $\lambda = 0.4$ is also used for the NMP-EMD method.

PDE-EMD

In this section, the results obtained from using the PDE-EMD are shown. As for the EMD, a unification procedure has been applied to the data, cf. Appendix C. After applying the unification procedure, the distribution of the number of components is as seen in Fig. 7.5.

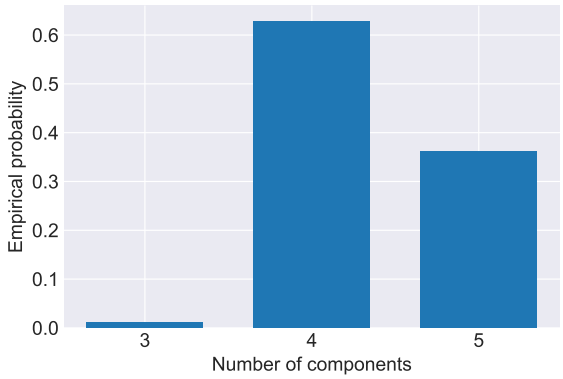


Figure 7.5: Distribution of the number of components extracted using the PDE-EMD after unification. Results found using N0 boundary conditions and $T = 5$.

In order to choose which type of boundary conditions to use, the performance in terms of the introduced performance measures is tested. This has been done using $T = 5$ and the results can be seen in Table 7.3. It is seen that the $\overline{\text{AIO}}$ and the $\overline{\text{MIO}}$ are very similar for all choices of boundary conditions. However, in line with expectations, changing the boundary conditions changes the $\overline{\text{EEEI}}$. Additionally, the choice of boundary conditions has a large influence on the $\overline{\text{CPM}}_{12}$. The lowest values of the $\overline{\text{EEEI}}$ and the $\overline{\text{CPM}}_{12}$ are observed when using the N0 boundary conditions and therefore the following tests are conducted using this choice of boundary conditions.

Boundary conditions	D0	D1	N0	N1
$\overline{\text{AIO}}$	0.2470	0.2189	**0.2045**	0.2703
$\overline{\text{MIO}}$	**0.5611**	0.5637	0.5635	0.6185
$\overline{\text{EEEI}}$	0.1450	0.6096	**0.0422**	0.2777
$\overline{\text{CPM}}_{12}$	1.456	0.5220	**0.3882**	1.487

Table 7.3: Comparison of decomposition performance measures for the implemented boundary conditions. Boldface indicates the best value for a given performance measure.

Next the effect of T is considered. The $\overline{\text{CPM}}_h$ has been plotted for different values of T which can be seen in Fig. 7.6. From this plot, it is noticed that a low value of T yields the best $\overline{\text{CPM}}$. Furthermore, the effect of T on $\overline{\text{EEEI}}$, $\overline{\text{AIO}}$, and $\overline{\text{MIO}}$ can be seen in Table 7.4. Looking at these measures, it is seen that lower values of T result in

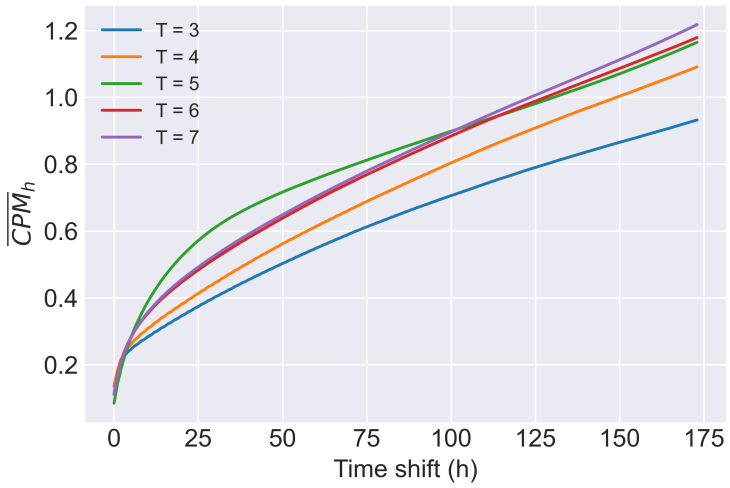


Figure 7.6: The $\overline{\text{CPM}}_h$ for different values of T using the N0 boundary conditions.

a worse IO, whereas the best $\overline{\text{AIO}}$ and $\overline{\text{MIO}}$ are observed when $T = 5$. When plotting the components for $T = 3$ and $T = 4$, it is noticed that the IMFs have low amounts of energy and that there are many fluctuations in the residual. This makes the residual more difficult to forecast and therefore these choices of T are not desirable. As such, we choose $T = 6$ as it performs well in terms of $\overline{\text{CPM}}_{12}$ and $\overline{\text{EEEI}}$.

T	3	4	5	6	7
$\overline{\text{AIO}}$	0.3701	0.3342	**0.2045**	0.2706	0.2544
$\overline{\text{MIO}}$	0.8721	0.8235	**0.5635**	0.7326	0.7018
$\overline{\text{EEEI}}$	**0.0059**	0.0093	0.0422	0.0258	0.0298
$\overline{\text{CPM}}_{12}$	**0.2839**	0.3087	0.3882	0.3537	0.3566

Table 7.4: The effect of T on $\overline{\text{EEEI}}$, $\overline{\text{AIO}}$, and $\overline{\text{MIO}}$. Experiment has been performed using N0 boundary conditions. Boldface indicates the best value for a given performance measure.

Comparison

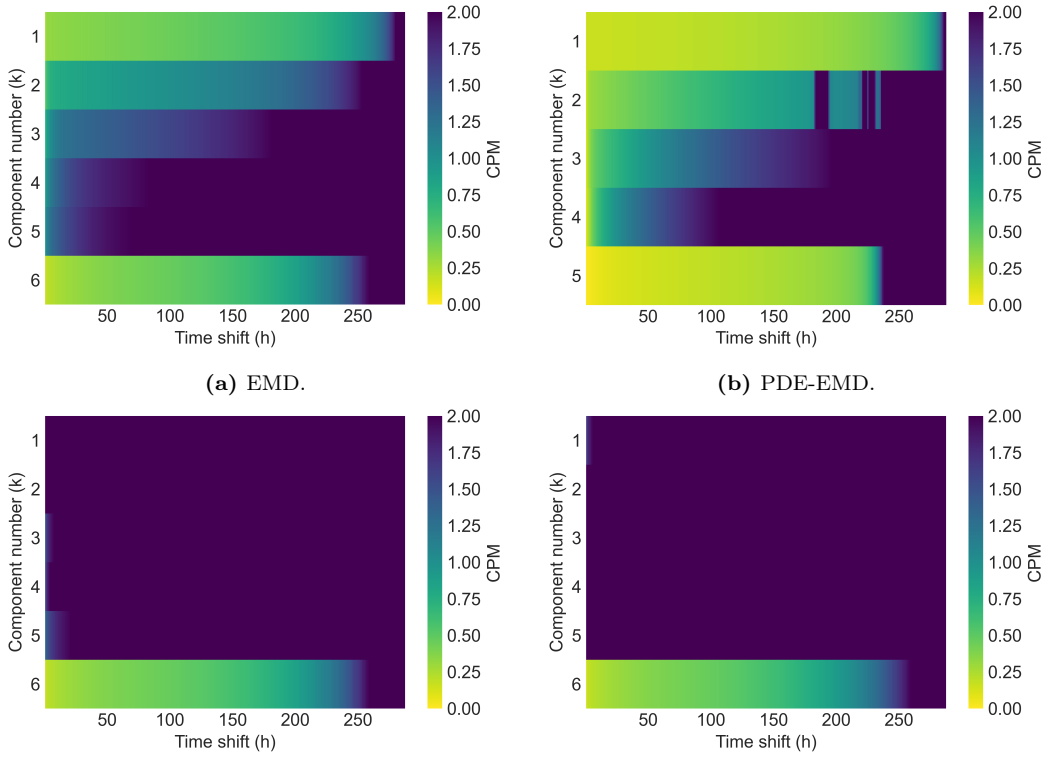
To conclude the numerical experiments with the decomposition methods, a comparison is made. In Table 7.5, a quantitative comparison in terms of the indirect performance measures of the decomposition results of the different adaptive decomposition methods is given. It is seen that in terms of $\overline{\text{AIO}}$, the FFT-NMP-EMD has the best performance, while in terms of $\overline{\text{MIO}}$ the EMD has the best performance. Additionally, the PDE-EMD has the smallest $\overline{\text{EEEI}}$ indicating a low amount of end effects and in terms of the consistency performance measure $\overline{\text{CPM}}_{12}$, the PDE-EMD also has the best performance. Finally, the NMP-EMD method has significantly

worse EEEI and $\overline{\text{CPM}}_{12}$ than the other methods. For this reason, the NMP-EMD method is not used for forecasting.

Decomposition	EMD	PDE-EMD	NMP-EMD	FFT-NMP-EMD
$\overline{\text{AIO}}$	0.0780	0.2706	0.2349	**0.05407**
$\overline{\text{MIO}}$	**0.1981**	0.7326	0.7048	0.2497
$\overline{\text{EEEI}}$	0.0428	**0.0258**	50.45	0.03876
$\overline{\text{CPM}}_{12}$	0.9018	**0.3537**	7.791	1.935

Table 7.5: Comparison of performance of the different adaptive decomposition methods in terms of the indirect performance measures $\overline{\text{AIO}}$, $\overline{\text{MIO}}$, $\overline{\text{EEEI}}$, and $\overline{\text{CPM}}_{12}$.

A more detailed overview of the CPM of the different methods is given in Fig. 7.7. From these plots, it is again seen that the PDE-EMD is best in terms of CPM and that the component with the worst CPM for this method is $k = 4$.



(a) EMD.

(b) PDE-EMD.

(c) FFT-NMP-EMD.

(d) NMP-EMD.

Figure 7.7: Colour plots of the CPM for the different decomposition methods.

An additional point of interest is the computational complexity of the methods. This is addressed in Table 7.6 in which the runtime of each method has been tested displaying that the EMD is the fastest algorithm, then follows the FFT-NMP-EMD, then the PDE-EMD, and lastly the NMP-EMD. As can be seen from the comparison of the complexity for decompositions with length $q = 288$ and $q = 576$, the PDE-EMD and NMP-EMD algorithms increase significantly in complexity for larger windows, whereas the EMD and FFT-NMP-EMD algorithms remain efficient.

Decomposition	EMD	PDE-EMD	NMP-EMD	FFT-NMP-EMD
$q = 288$	0.0314	0.371	10.2	0.294
$q = 576$	0.0539	17.3	38.4	0.364

Table 7.6: Timing the speed in seconds of the different decomposition methods averaged over 10 repetitions. Computations made on a 2000 MHz AMD EPYC Processor.

Concerning the value of q, we have chosen $q = 288$, i.e. a window length of 1 day. This has been chosen because it was feasible in terms of computational complexity and memory complexity for all the tested methods in accordance with Table 7.6 and since we expected a component relating to the fact that there is a diurnal pattern in wind speed [Bur+01, pp. 11-16]. We suspect that better performance could be achieved using a larger value of q as there would be more of the same samples in each window which in turn might make for a more consistent decomposition. Additionally, increasing the window length opens opportunities for extracting additional low frequency components. However, this comes at the cost of a higher computation time for the decomposition as seen in Table 7.6.

7.2 Forecasting Experiments

In this section, the application of decomposition based methods for forecasting is evaluated. This includes model selection in Section 7.2.1 as well as an analysis of the results in Section 7.2.2.

7.2.1 Model Selection

Before evaluating models for forecasting on the test data, a validation set has been used to fit hyperparameters. This has been done for all the forecasting methods used in this thesis.

Autoregressive Model

In order to choose the order p of the autoregressive baseline method, we have performed preliminary tests and based on these, we set p to 1. The AR(1) model is fitted using ordinary least square and from this, we obtain $\phi = 0.9979$. Hence, the obtained AR(1) model is almost equal to a persistence method. The validation NRMSE with this AR(1) model is 5.42 %.

LSTM

With the LSTM baseline model, many hyperparameter settings has been tested and the hyperparameters for which the best model has been obtained are $p = 3$, batch size of 32, initial learning rate of 0.001 with exponential learning rate decay of 0.7, dropout rate of 0.3, a total of 3 hidden layers with 128 hidden units in each layer. The validation NRMSE with the best hyperparameter setting is 5.34 %.

Offline EMD-LSTM

In the offline setup, we have tested the EMD-LSTM model for different hyperparameter settings and the setting yielding the best validation NRMSE is identical to the setting used for the baseline LSTM. In the offline setup, the EMD is applied to the entire dataset. When the EMD is applied to the entire dataset, it results in 20 components. During testing, it has been seen that the LSTM has difficulties forecasting the last component on the test set. By visual inspection of the residual of the EMD, i.e. the last component, seen in Fig. 7.8, it is seen that the residual only consist of a single oscillation and that the data in the training set does not resemble the data in the test set. Hence, the LSTM model is not able to learn the behaviour of the residual. Furthermore, it is noticed that the residual does not change a lot between samples. With this in mind, we choose to forecast the residual using a persistence model rather than an LSTM model. The validation loss in terms of NRMSE of this model is 3.03 %.

While the offline EMD-LSTM model is successful for forecasting, preliminary tests have revealed that the offline FFT-NMP-EMD-LSTM model cannot compete with the best observed validation NRMSE being 7.34 %. This poor performance is attributed to a higher SampEN observed for the offline FFT-NMP-EMD components in comparison to the offline EMD components and the offline FFT-NMP-EMD-LSTM model is therefore not considered further in this thesis.

EMD-LSTM

When using the EMD, the wind power data is decomposed into 6 components for each window where the IF descends as the component number increases. During preliminary tests, we have found that the first 2 components are unpredictable. This is seen since for a wide variety of training settings, the training loss and validation loss remain constant during training. Moreover, predicting zero in each point is

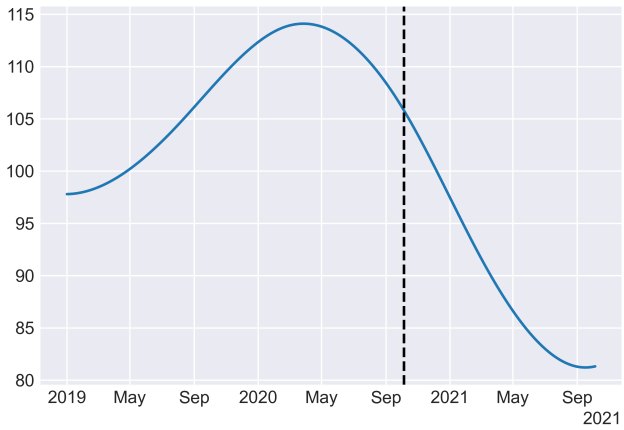


Figure 7.8: The residual of the offline EMD-LSTM where the vertical dotted line separates the training data from the test data.

equally good to implementing a neural network. Hence, we conclude that the first two components are noise.

An extensive hyperparameter search has been conducted. Among the important hyperparameters are the input size p and the target shift ξ as these hyperparameters change the input-output relation. The model with the lowest validation NRMSE uses $p = 3$ and $\xi = 0$ with a batch size of 32, an initial learning rate of 0.001 with exponential learning rate decay of 0.9, dropout rate of 0.3, and a total of 3 hidden layers with the number of hidden units given as 64, 64, 32 for layers 1, 2, 3, respectively. The validation NRMSE for this model is 6.70 % which is poor compared to the AR(1) and LSTM baselines. An overview of the performance with the model for each component is given in Table 7.7.

Component	1	2	3	4	5	6	Power
MSE	22.5	56.1	212	442	127	474	900

Table 7.7: Validation loss in terms of MSE for each component and of the wind power forecast using the EMD-LSTM.

Considering the poor performance observed in the preceding result, alternative representations of the EMD have been tested. This includes reducing the number of components to 3 or 4 by either collecting multiple components in the last IMF or in the residual. This decreases the CPM, however, it also decreases the predictability of the compounded component. The best result has been found by reducing the number of components to 3 by aggregating IMFs 3, 4, 5, and the residual thereby yielding a decomposition consisting of two high frequency noise components and one signal component. As such, this method reduces to a type of denoising using the EMD followed by forecasting. With this method, the NRMSE on the validation data is

5.77 % for the wind power forecast which is still worse than the AR(1) and LSTM baselines.

A different method for assigning targets has also been attempted in which the target is the actual wind power production and thereby the forecasters for all the components are trained jointly, as the loss is computed by combining the result from each of the forecasters. In this setup, the decomposition method is seen as a feature extraction method that increases the dimensionality of the input space. However, poor results have been observed with this method.

FFT-NMP-EMD-LSTM

Following the experiments with different hyperparameter settings for the EMD-LSTM, the hyperparameters for the forecasters with the FFT-NMP-EMD-LSTM model have been fitted. The model with the lowest validation NRMSE is obtained for $p = 18$, $\xi = 0$, a batch size of 32, an initial learning rate of 0.001 with exponential learning rate decay of 0.7, dropout rate of 0.3, a total of 4 hidden layers with the number of hidden units given as 512, 512, 512, 256 for layers 1, 2, 3, 4, respectively. The validation NRMSE for this model is 7.26 % which is worse than the AR(1), LSTM, and EMD-LSTM models.

PDE-EMD-LSTM

Using the PDE-EMD method, the wind power data has been decomposed into 5 components and for each of these components, an LSTM neural network has been trained. As for the EMD, the LSTM neural network cannot learn anything from the first two components extracted using the PDE-EMD. In order to determine the hyperparameters, a manual hyperparameter tuning has been made. This results in an LSTM neural network which consists of three layers with 128, 256 and 128 cells for layers 1, 2 and 3, respectively. The batch size has been chosen as 32 and the initial learning rate is 0.0007 with an exponential learning rate decay of 0.7. The dropout rate is set to 0.05 and we have chosen $\xi = 0$ and $p = 1$. This results in an NRMSE of 5.43 %. For these choices of hyperparameters, the validation loss in terms of MSE observed for the individual components is given in Table 7.8.

Component	1	2	3	4	5	Power
MSE	19.9	9.92	43.7	91.2	154	598

Table 7.8: Validation loss in terms of MSE for each component and of the wind power forecast using the PDE-EMD.

Notice that for all the tested adaptive decomposition methods, $\xi = 0$ has been found to be the best value. The reason the ξ parameter has been introduced is to avoid end effects, however, increasing ξ also has the downside of an increased CPM and as such there is a trade-off between these measures and the results have shown that it is more important to have a low CPM.

7.2.2 Test Data Results

Looking at Table 7.9, the performance of the forecasting models in terms of NRMSE, NMAE, and NBIAS can be seen. Additionally, the 95th percentile is given. It is noticed that the offline EMD-LSTM model is the best model by far on all the performance measures except NBIAS where the AR model performs the best. The good performance of this model has also been expected based on the results in [AVK21]. However, this model cannot be applied in an online setup and is therefore not as relevant in an actual forecasting scenario. Considering the models which can be

Model	NRMSE	NMAE	NBIAS	ζ
Offline EMD-LSTM	3.04	1.80	−0.200	6.56
AR	5.40	3.46	**0.108**	11.5
LSTM	**5.34**	**3.40**	−0.207	**11.5**
EMD-LSTM	6.82	4.61	0.335	14.7
FFT-NMP-EMD-LSTM	6.46	4.36	0.440	13.8
PDE-EMD-LSTM	5.39	3.46	0.225	11.7

Table 7.9: The NRMSE, NMAE, and NBIAS in percentage on the test data for one hour ahead forecasting with $P_{\text{capacity}} = 447.565$ MW. Additionally, the normalised 95th percentile in percentage can be seen. The best performance for a method which can be applied in an online setup is highlighted using boldface.

applied in an online setup, it is seen that the LSTM baseline model has the best performance in terms of NRMSE and NMAE. This result shows that the use of decomposition methods as a pre-processing tool in an online setup has not resulted in an improved model over the baseline. However, it is seen that the PDE-EMD-LSTM model is clearly the best of the three online decomposition based models, displaying significant improvements compared to the EMD-LSTM model. In comparison to the baselines, the PDE-EMD-LSTM has similar performance to the AR model and slightly worse performance than the LSTM model.

In Fig. 7.9, the performance of the LSTM, EMD-LSTM, PDE-EMD-LSTM, and offline EMD-LSTM models for each month in the test set can be seen. Again, it is clearly seen that the offline EMD-LSTM method performs the best. However, the offline EMD-LSTM method also has the largest fluctuations from month to month. For the other methods, the performance is fairly stable for all months with a small decrease in performance in November. Generally, the performance of the models are separated for all the months, i.e. the performance ranking of the models does not

depend on the time of year.

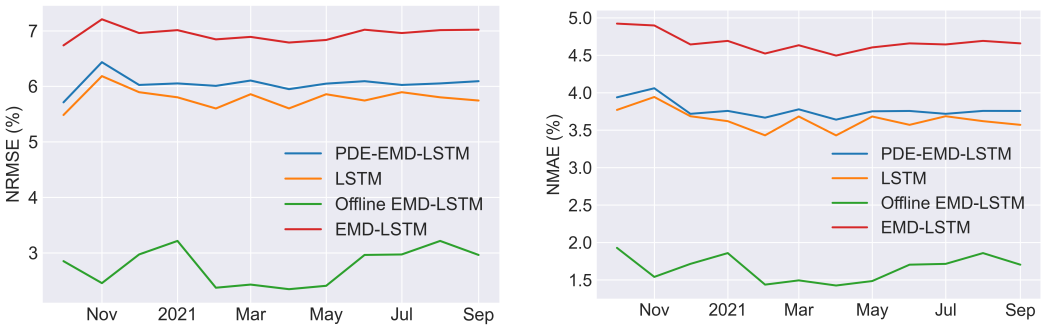


Figure 7.9: Performance in terms of NRMSE and NMAE in percentage for individual months during a year with the LSTM, EMD-LSTM, PDE-EMD-LSTM, and offline EMD-LSTM models.

The following analysis is focused on the PDE-EMD-LSTM and the offline EMD-LSTM as these methods have been shown to be the best decomposition based methods in an online and offline setup, respectively. In Fig. 7.10, a kernel density estimate (KDE) with a Gaussian kernel function [Bis06, pp. 122-124] on the errors of the PDE-EMD-LSTM, the offline EMD-LSTM, and the baseline LSTM models can be seen. It is noticed that the shape of the KDE for the LSTM and PDE-EMD-LSTM models are similar. Additionally, it is clearly seen that a better performance is obtained with the offline EMD-LSTM model as there is a higher concentration of low errors.

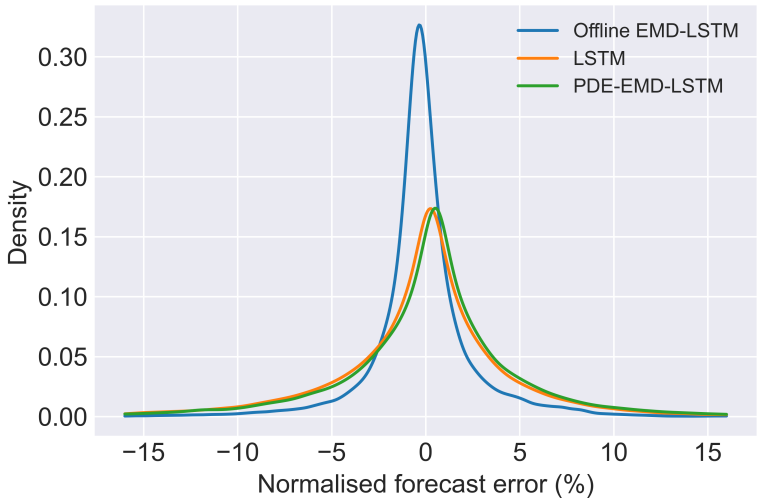


Figure 7.10: A KDE of the normalised errors in percent for the PDE-EMD-LSTM, the LSTM, and the offline EMD-LSTM.

To visualise the forecasting models, in Fig. 7.11, a forecast of the wind power for October 12th 2021 can be seen using the offline EMD-LSTM and the PDE-EMD-LSTM models. In this figure, it is seen that the offline EMD-LSTM forecast is smoother than both the production data and the PDE-EMD-LSTM forecast and is also more accurate than the PDE-EMD-LSTM forecast.

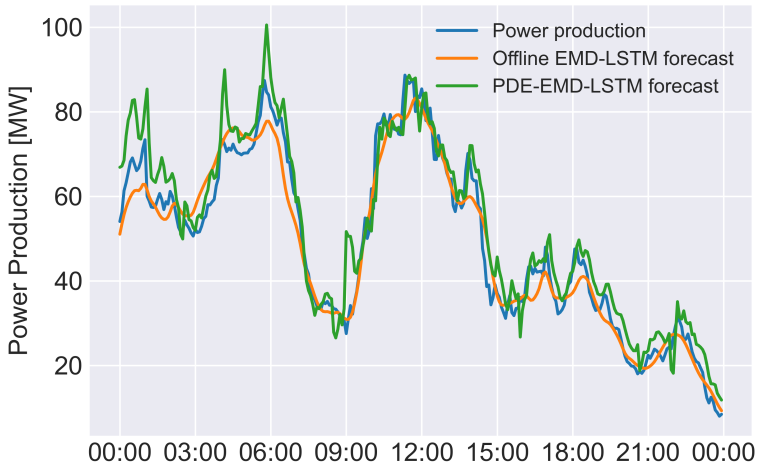


Figure 7.11: Forecast for October 12th 2021 for the offline EMD-LSTM and the PDE-EMD-LSTM models.

In the following, the decomposition based forecasting models are analysed in further detail to understand the successes and failures of the models. In relation to this, the indirect performance measures are related to the forecast performance with the models.

The effect of the SampEN and the CPM on the success of forecasting individual components have been investigated. The result can be seen in Fig. 7.12 in which the VSMSE_k, cf. Eq. (6.7), is plotted according to $\overline{\text{SampEN}}_k(2)$ and $\text{CPM}_{k,12}$. It is seen that the lowest VSMSE_k is achieved by the point in the lower left corner which is the PDE-EMD residual. Then follows the EMD residual which is the point to the right of the PDE-EMD residual. This component has a slightly lower $\overline{\text{SampEN}}_k(2)$ but a slightly higher $\text{CPM}_{k,12}$. Then comes the VSMSE_k for the power data itself using the baseline LSTM model which is the point at $\text{CPM}_{k,12} = 0$. This indicates that due to the higher $\overline{\text{SampEN}}_k(2)$ of the wind power compared to the residuals, the forecaster does not learn the patterns as effectively. Furthermore, a tendency to have VSMSE_k close to 1 when $\overline{\text{SampEN}}_k(2)$ is above 0.5 is seen and additionally by inspecting the points with $\overline{\text{SampEN}}_k(2)$ below 0.5 it is seen that a lower VSMSE_k tends to be achieved when $\text{CPM}_{k,12}$ is lower.

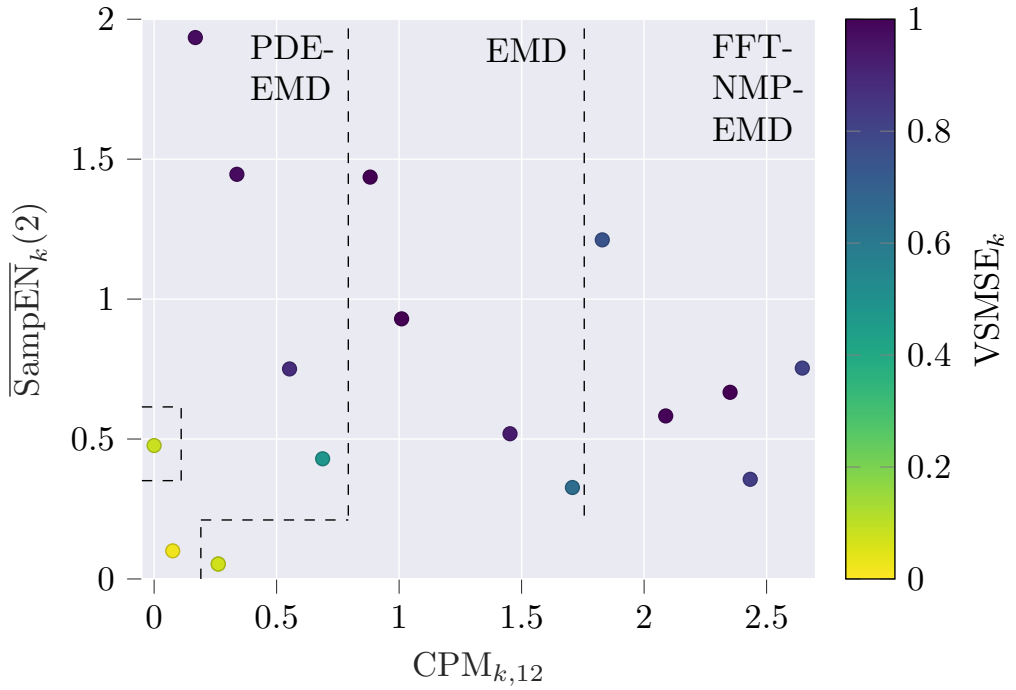


Figure 7.12: VSMSE_k for each component computed according to Eq. (6.7) for the EMD-LSTM, PDE-EMD-LSTM, and FFT-NMP-EMD-LSTM and plotted with respect to the sample entropy, $\overline{\text{SampEN}}_k(2)$, and the consistency performance measure, $\text{CPM}_{k,12}$. Additionally, the result with the LSTM without decomposition is included for comparison with $\text{CPM}_{k,12}$ set to 0. The dashed lines indicate which decomposition method each point belongs to and it is noted that the EMD and the FFT-NMP-EMD has the same last component. Moreover, note that with the EMD, $\text{CPM}_{5,12}$ is 59.8 and this point is therefore excluded from this figure.

As mentioned, the PDE-EMD-LSTM model has similar performance to the baselines, however, when analysing the performance of the PDE-EMD-LSTM model on the individual components of the decomposition in comparison to that of the EMD-LSTM model, encouraging results are seen. This is shown in Table 7.10 in which it is seen that the PDE-EMD-LSTM generally has significantly less MSE in the individual components. Specifically, we note that the sum of the MSE for the EMD-LSTM components is 1285 with an MSE on the power of 930, while for the PDE-EMD-LSTM the sum of the MSE for the individual components is merely 306 with an MSE on the power of 582. Hence, while the EMD-LSTM model decreases the MSE loss when aggregating the components, the opposite is observed for the PDE-EMD-LSTM.

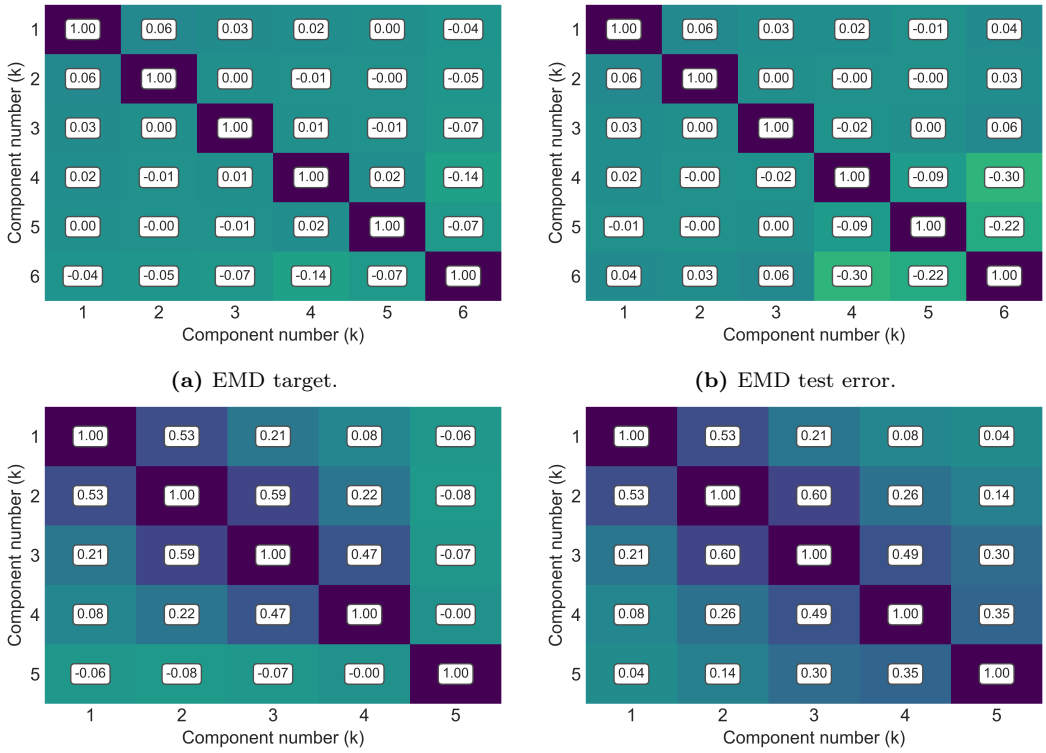
To analyse the error of the EMD-LSTM and PDE-EMD-LSTM further, the PCC for all pairs of components is considered for both the target, i.e. $c_{k,t+\tau}(t+\tau)$ and the forecast error, i.e. $\hat{c}_{k,t}(t+\tau) - c_{k,t+\tau}(t+\tau)$. By inspecting Figs. 7.13b and 7.13d, it is noted that significant positive correlations are present in the PDE-EMD-LSTM forecast errors particularly for neighbouring components, while for the EMD-LSTM

Component	1	2	3	4	5	6
EMD-LSTM	19.2	75.4	186	397	111	495
PDE-EMD-LSTM	15.6	10.3	42.1	93.7	144	

Table 7.10: Test loss in terms of MSE for each component.

model the correlations are closer to 0 with significant negative correlations between component 6 and components 4 and 5. In this situation, negative correlations are preferred to that of positive correlations since this means that the errors when aggregated tend to cancel each other out where the opposite is the case for positive correlations. Hence, the positive correlations in Fig. 7.13d explain why the sum of MSE for the individual components is much less than the MSE of the power forecast for the PDE-EMD-LSTM model. Additionally, by comparing Fig. 7.13c with Fig. 7.13d and comparing Fig. 7.13a with Fig. 7.13b, a tendency is observed which indicates that if the components are correlated in the targets, then the forecast errors of the components are also correlated. This can be related to the IO of the decompositions for which it has been seen in Table 7.5 that the PDE-EMD is significantly worse than the EMD on the wind power data in terms of both AIO and MIO. Thus, it seems that a poor performance of the PDE-EMD method in terms of AIO and MIO manifests itself in the power forecast error. However, while the correlation of the targets seems to be able to explain some of the correlations in the forecast error of the components, this is not a guarantee and significant differences can occur. For instance, this is the case for component 5 with components 3 and 4 with the PDE-EMD where unfortunately significant positive correlations are observed for the forecast error even though the correlations in the targets are approximately 0.

A KDE plot for each component for the PDE-EMD-LSTM and the EMD-LSTM can be seen in Fig. 7.14. Considering this plot, it is noticed that each component is right skewed. Additionally, it is seen that the residual with the PDE-EMD-LSTM model has a more concentrated shape than that of the residual with the EMD-LSTM model.



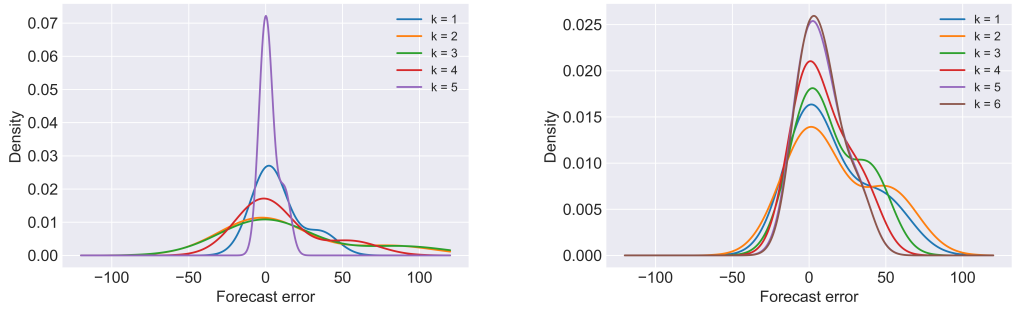
(a) EMD target.

(b) EMD test error.

(c) PDE-EMD target

(d) PDE-EMD test error.

Figure 7.13: PCC of decomposed EMD and PDE-EMD target as well as the forecast error of the components with the EMD-LSTM and PDE-EMD-LSTM models.



(a) PDE-EMD-LSTM.

(b) EMD-LSTM.

Figure 7.14: A KDE of the normalised errors for each component of the PDE-EMD-LSTM and the EMD-LSTM.

8. Conclusion

The aim of this thesis has been to study the effect of using adaptive decomposition methods as a pre-processing tool in forecasting of the Danish wind power production which we sought to do by answering the problem statement:

What is the effect of applying adaptive decomposition based models for online forecasting of wind power production compared to a purely deep neural network based model?

It has previously been seen that the predominant method for adaptive data analysis, i.e. the empirical mode decomposition (EMD), possesses a great potential for forecasting in an offline setting. However, it suffers from weaknesses such as mode mixing and end effects which might make the method unsuitable in an online setup. In order to answer the problem statement, we have investigated four different adaptive decomposition methods, namely the EMD, a partial differential equation based decomposition method dubbed PDE-EMD, and two compressive sensing based decomposition methods referred to as NMP-EMD and FFT-NMP-EMD, respectively. To introduce the theoretical aspects of the adaptive decomposition methods in consideration, theory regarding compressive sensing with time-frequency dictionaries and partial differential equations, specifically the heat equation, has been presented.

The decomposition methods have been analysed both in regards to performance of the decomposition itself as well as performance of the resulting forecasts of the wind power data. Through the analysis of the decompositions, it has been seen that the PDE-EMD is both least subject to end effects and obtains the most consistent decompositions among the four methods whereas the NMP-EMD has the worst performance. However, with regards to orthogonality of the resulting components, the PDE-EMD results in the largest index of orthogonality, whereas the EMD and FFT-NMP-EMD obtain the smallest index of orthogonality.

For the evaluation of the performance of the forecasting models of the wind power data, the models have been compared with an autoregressive model and a long short-term memory (LSTM) neural network as baselines. Due to its high computational complexity and poor decomposition results, the NMP-EMD has not been included for the forecasting part of the analysis. For the three remaining methods, an LSTM

neural network has been trained for each of the components. The forecasting results show that while the EMD-LSTM model in an offline setting obtain promising results, the performance is significantly degraded in an online setup. Specifically, the EMD-LSTM and FFT-NMP-EMD-LSTM models have worse performance in terms of normalised root mean squared error (NRMSE) and normalised mean absolute error (NMAE) than the baselines. The PDE-EMD-LSTM model in an online setup has performance rivalling that of the baselines in terms of NRMSE and NMAE, however the model does not manage to outperform the baselines. The good consistency of the decomposition as well as the low amount of end effects allow the PDE-EMD-LSTM model to have good performance in terms of mean squared error (MSE) on the individual components. However, when aggregating the results on the individual components, the forecast error is amplified due to positive correlations present in the forecast error of the components which can partly be attributed to less orthogonal components with the PDE-EMD than with the EMD.

To summarise, the performance of an adaptive decomposition based forecast model is degraded when applied in an online setup compared to an offline setup. This degradation can be alleviated by employing an adaptive decomposition method that can alleviate the weaknesses of the EMD, however, the adaptive decomposition methods considered in this thesis did not alleviate the degradation enough to outperform the baselines for forecasting.

9. Further Work

In this thesis, adaptive decomposition methods used as a pre-processing step in a one hour forecast has been investigated with a focus on an online implementation.

When making the decompositions for the online forecast, windows with a length of one day have been used. This has primarily been chosen based on considerations of computational complexity. However, we expect larger windows to yield a better overall result as we expect it would result in more consistent decompositions and allow the decomposition methods to find additional low frequency components. This would come at the cost of a higher computational complexity. However, the computational complexity would mostly be a problem when making the data windows for training and testing forecasters and it would still be possible to implement the trained models in an online setup.

When analysing the errors of the method using partial differential equations for adaptive data decomposition, it has been found that the forecast errors between the components are positively correlated and partly attribute this to a lack of orthogonality of the decomposition. This is an issue as it results in an amplification of the power forecast error when the forecast errors of the individual components are aggregated. However, seeing as the forecast for each component has been done separately, it is difficult to change in the setup used in this thesis. A different training procedure could be made for the forecaster where the LSTM neural network should learn to forecast the components simultaneously. In this setup, the forecaster should minimise the combined MSE which might incentivise it to minimise the correlations between the forecast errors.

Finally, other decomposition methods may have yielded better results and as seen in Chapter 2 there are still several methods which can be investigated. Based on the results of this thesis, it could be interesting to investigate the partial differential equation based methods further or look into new methods for decomposition. Furthermore, methods which make a combined decomposition may be investigated, i.e. instead of finding one component at a time, it might be beneficial to find all components concurrently. Doing this might allow the algorithm to ensure that the decomposition fulfils properties such as orthogonality.

A. Mathematical Preliminaries

In this appendix, preliminary results used in Chapters 2 to 4 are introduced. In Appendix A.1, the Hilbert transform alongside the Bedrosian identity is introduced. In Appendix A.2, some preliminary definitions regarding the wavelet transform and a result used in Chapter 4 are introduced.

A.1 Preliminaries for the Hilbert-Huang Transform

Firstly, for later convenience, the Fourier transform for integrable functions is defined.

Definition A.1 (Fourier Transform)
Let $y : \mathbb{R} \to \mathbb{C}$ be a function in L^1. Then its Fourier transform is given by

$$\mathcal{F}\{y\}(\omega) = \int_{-\infty}^{\infty} y(t) e^{-j\omega t} dt.$$

Furthermore, for $\mathcal{F}\{y\}(\omega) \in L^1$ the inverse Fourier transform is given by [Fol92, p. 213 & 218]

$$y(t) = \mathcal{F}^{-1}\{y\}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathcal{F}\{y\}(\omega) e^{j\omega t} dt.$$

The Fourier transform can be extended to L^2 as is stated by the following theorem.

Theorem A.2 (Plancherel's Theorem)
[Tes10, p. 187] The Fourier transform extends to a unitary operator $\mathcal{F} : L^2 \to L^2$.

Proof.
For a proof see [Tes10, p. 187]. ■

Secondly, an analytical signal is introduced as this type of signal is paramount in obtaining the instantaneous frequency using the Hilbert-Huang transform.

Definition A.3 (Analytical Signal)
[Smi07] Let $y : \mathbb{R} \to \mathbb{C}$ be a complex-valued function with Fourier transform $\mathcal{F}\{y\}(\omega)$. Then $y(t)$ is called an analytical signal if $\mathcal{F}\{y\}(\omega) = 0$ when $\omega < 0$.

A way to obtain analytical signals is by the use of the Hilbert transform. Specifically, given a signal $y : \mathbb{R} \to \mathbb{C}$, an analytical signal can be derived as $y_A(t) = y(t) + j\mathcal{H}\{y\}(t)$ where $\mathcal{H}\{y\}$ denotes the Hilbert transform of y. This representation does not change the distribution of the positive frequency content of the signal. The Hilbert transform is defined as follows.

Definition A.4 (Hilbert Transform)
[Joh12, pp. 1-2] The Hilbert transform $\mathcal{H}\{y\} : \mathbb{R} \to \mathbb{C}$ of a function $y : \mathbb{R} \to \mathbb{C}$ is defined for all t by

$$\mathcal{H}\{y\}(t) = y(t) * \frac{1}{\pi t} = \frac{1}{\pi} p.v. \int_{-\infty}^{\infty} \frac{y(\tau)}{t-\tau} d\tau$$

when the integral exists as a principle value and where the integral as a principal value is given by

$$p.v. \int_{-\infty}^{\infty} \frac{y(\tau)}{t-\tau} d\tau = \lim_{\varepsilon \to 0^+} \Big(\int_{-\infty}^{\tau-\varepsilon} \frac{y(\tau)}{t-\tau} d\tau + \int_{\tau+\varepsilon}^{\infty} \frac{y(\tau)}{t-\tau} d\tau \Big).$$

This section is ended by introducing the Bedrosian identity in its most general form.

Theorem A.5 (Bedrosian Identity)
[Bed63] Let $y : \mathbb{R} \to \mathbb{C}$ and $f : \mathbb{R} \to \mathbb{C}$ denote two functions in L^2. If

1. the Fourier transform $\mathcal{F}\{y\}(u)$ of $y(t)$ vanishes for $|u| > a$ and the Fourier transform $\mathcal{F}\{f\}(u)$ of $f(t)$ vanishes for $|u| < a$ where a is an arbitrary positive constant, or
2. $y(t)$ and $f(t)$ are analytic,

then the Hilbert transform of the product $y(t)$ and $f(t)$ is given by

$$\mathcal{H}\{y(x)f(x)\} = y(x)\mathcal{H}\{f(x)\}.$$

Proof.
For a proof see [Bed63]. ■

A.2 Preliminaries to the Uniqueness of the Compressive Sensing Solution

In this section, some results which are useful in Chapter 4 are presented. First, the wavelet function is defined.

Definition A.6 (Wavelet Function)
[Dau94, p. 24] A function $\psi \in L^2$ is called a wavelet function if it fulfils the following criterion

$$C_\psi = 2\pi \int_{\mathbb{R}} |\xi|^{-1} |\mathcal{F}\{\psi\}(\xi)|^2 d\xi < \infty. \tag{A.1}$$

This leads to the following criteria for $\psi \in L^1$.

Theorem A.7 (Wavelet Function in L^1)
[Dau94, p. 24] If $\psi \in L^1$ then Eq. (A.1) can only be satisfied if

$$\mathcal{F}\{\psi\}(0) = 0,$$

or

$$\int_R \psi(t) dt = 0.$$

Having defined a wavelet function, the wavelet transform can then be defined.

Definition A.8 (Wavelet Transform)
[Dau94, p. 3] The continuous-time wavelet transform of $y(t)$ is given by

$$\mathcal{W}\{y\}(\omega, t) = \frac{1}{\sqrt{|\omega|}} \int_{\mathbb{R}} \overline{\psi}(\frac{\tau - t}{\omega}) y(\tau) d\tau, \quad \omega \neq 0, t \in \mathbb{R}$$

where ω is a scale factor, ψ is the wavelet function defined in Definition A.6, and the $\overline{\psi}$ denotes complex conjugation of ψ.

Finally, a lemma which is used in the proof of Theorem 4.4 is presented. In the lemma, the rate at which a certain function increases is determined.

Lemma A.9
Let $y(x) = \frac{\sqrt{x}-1}{\sqrt{x}+1}$ and $x > 1$, then

$$\frac{dy(x)}{dx} = \frac{1}{\sqrt{x}(\sqrt{x}+1)^2} < 1.$$

Proof.
Using the quotient rule for differentiation, then

$$\begin{aligned}
\frac{d}{dx}\frac{\sqrt{x}-1}{\sqrt{x}+1} &= \frac{(\sqrt{x}-1)'(\sqrt{x}+1) - (\sqrt{x}-1)(\sqrt{x}+1)'}{(\sqrt{x}+1)^2} \\
&= \frac{\frac{1}{2\sqrt{x}}(\sqrt{x}+1) - (\sqrt{x}-1)\frac{1}{2\sqrt{x}}}{(\sqrt{x}+1)^2} \\
&= \frac{\sqrt{x}+1-\sqrt{x}+1}{2\sqrt{x}(\sqrt{x}+1)^2} \\
&= \frac{1}{\sqrt{x}(\sqrt{x}+1)^2}
\end{aligned}$$

and since $x > 1$, it follows that $\frac{1}{\sqrt{x}(\sqrt{x}+1)^2} < 1$. ■

B. Baselines

In this appendix, the theory of baselines used in the thesis are briefly introduced. In Appendix B.1, the forward propagation equations in a long short-term memory (LSTM) neural network are given and in Appendix B.2, an autoregressive (AR) equation is given.

B.1 Long Short-Term Memory Neural Network

LSTM neural networks are a type of gated recurrent neural network in which the usual hidden units of a neural network are replaced by so-called LSTM cells. In LSTMs, a gate structure is used in combination with an internal self loop. The gate structure plays an important role in how the information is passed through the network. LSTMs have three different gates, i.e. the forget, input, and output gate.

Consider an input time series $\boldsymbol{y}_t$ observed at $t = 1, 2, \ldots, n$, then at time t the hidden layer gives output $\boldsymbol{h}_t$ and the three gates can be expressed by the following equations

$$\begin{aligned}
\boldsymbol{f}_t &= \sigma(\boldsymbol{W}_{hf}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{yf}\boldsymbol{y}_t + \boldsymbol{b}_f), \\
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_{hi}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{yi}\boldsymbol{y}_t + \boldsymbol{b}_i), \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_{ho}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{yo}\boldsymbol{y}_t + \boldsymbol{b}_o)
\end{aligned}$$

where $\boldsymbol{f}_t$, $\boldsymbol{i}_t$, and $\boldsymbol{o}_t$ are the forget gate, input gate, and output gate, respectively, $\boldsymbol{W}_{**}$ and $\boldsymbol{b}_*$ denote the weights and biases, respectively, and σ denotes the sigmoid activation function. In addition to the gates, the LSTM also has a state which is updated through a self loop

$$\boldsymbol{s}_t = \boldsymbol{f}_t \odot \boldsymbol{s}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{g}_t$$

where $\odot$ denotes the Hadamard product and where $\boldsymbol{g}_t$ is given by

$$\boldsymbol{g}_t = \tanh(\boldsymbol{W}_{hg}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{hg} + \boldsymbol{W}_{yg}\boldsymbol{y}_t + \boldsymbol{b}_{yg}).$$

Afterwards, the hidden layer output of an LSTM cell is computed as

$$\boldsymbol{h}_t = \tanh(\boldsymbol{s}_t) \odot \boldsymbol{o}_t. \tag{B.1}$$

The architecture of an LSTM cell can be seen in Fig. B.1. [GBC16, pp. 397-400]

The gate structure can be interpreted as follows. The forget gate determines the information which needs to be discarded from the state. The input gate determines the information which should be stored in the state. Finally, the output gate determines what is outputted from the state to the hidden unit or if a cell should be shut down.

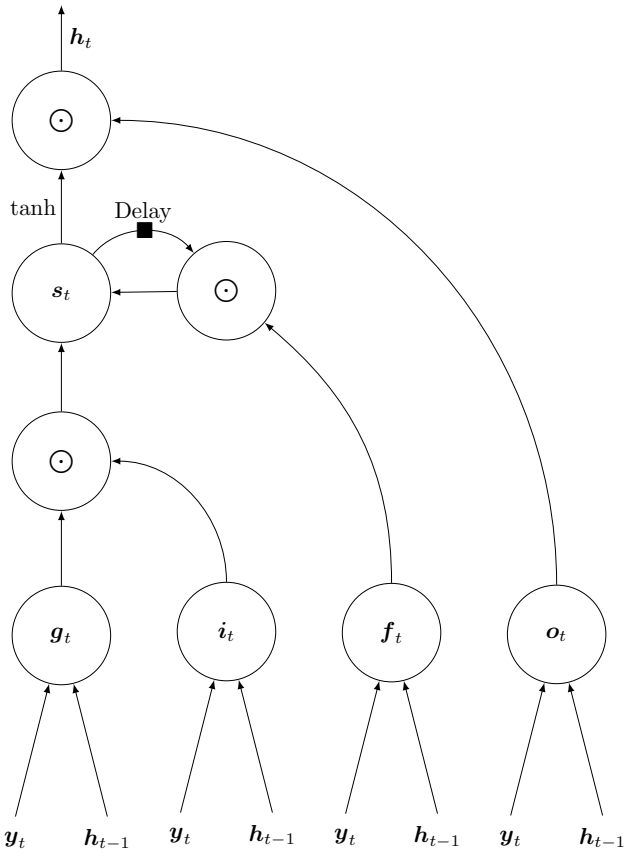


Figure B.1: A block diagram of an LSTM cell. The gate variables $\boldsymbol{g}_t$, $\boldsymbol{i}_t$, $\boldsymbol{f}_t$ and $\boldsymbol{o}_t$ internally apply the weights, biases, and activation functions seen in the forward propagation equations. Figure inspired by [GBC16, fig. 10.16].

B.2 Autoregressive Model

As mentioned in Chapter 7, an AR model is used as a baseline. An AR model of order p, AR(p) is given by [SS17, p. 76]

$$P(t) = \sum_{j=1}^{p} \phi_j P(t-j) + w(t)$$

where $P(t)$ is assumed to be stationary with zero mean, $w(t) \sim \mathcal{N}(0, \sigma_w^2)$, ϕ_j for $j = 1, \ldots, p$ are coefficients, and $\phi_p \neq 0$. In order to obtain a forecast for time $t + \tau$ at time t, the minimum mean squared error predictor is used [SS17, p. 109]

$$\hat{P}_t(t+\tau) = \sum_{j=1}^{p} \phi_j \hat{P}_t(t+\tau-j) \tag{B.2}$$

for $\tau > 0$ and where $\hat{P}_t(i) = P(i)$ for $1 \leq i \leq t$ and $\hat{P}_t(i) = 0$ for $i \leq 0$.

C. Unification Procedure

In this chapter, the unification procedure used in Chapter 7 to increase the consistency of the results from the empirical mode decomposition (EMD) and the method dubbed PDE-EMD is explained. This includes the motivation for the procedure and an outline of the procedure. The procedure is similar but not identical for the EMD and the PDE-EMD. Hence, the procedure for the EMD is described in Appendix C.1 and following in Appendix C.2, the procedure for the PDE-EMD is given.

C.1 Empirical Mode Decomposition

The unification is initially motivated by handling undesirable effects in the decomposition including mode mixing and the fact that the number of components when using the EMD on different windows of the data is not the same. Consider a window size $q = 288$, i.e. one day of wind power data. Then using the EMD, the distribution of the number of components found is illustrated by a bar chart in Fig. C.1a and a boxplot in Fig. C.1b. It is seen that most of the windows result in 5 or 6 components when using the EMD. The minimum amount of components found is 4 and a few instances occur where the number of components is 7, 8, or 9. An outlier is seen as for three windows 20 components are found when the maximum amount of components is fixed as 20. However, all the additional components with index above 6 contain a very small amount of energy.

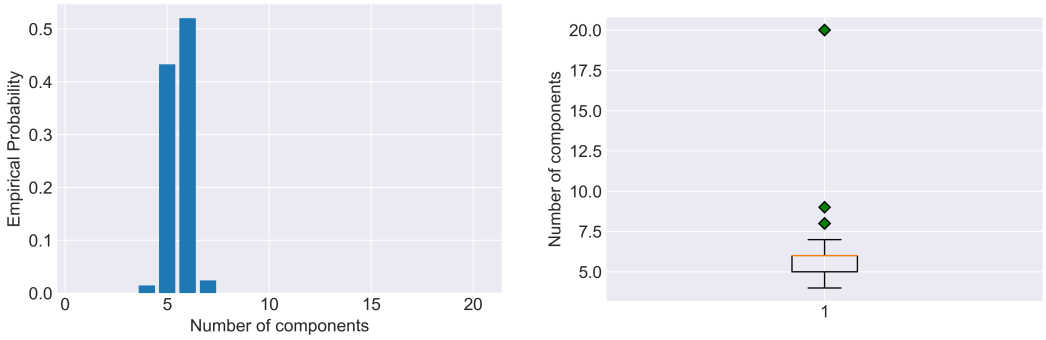
Based on the preceding discussion, the maximum number of intrinsic mode functions (IMFs) is set to 5. Now the unification procedure is introduced in order to handle situations where the number of IMFs found are less than s and to handle certain exceptions that can occur when applying the EMD. The heuristic rules applied in the following are based on observing the EMD decomposition on windows of the data while considering the stopping criteria used for the EMD:

1. Firstly, components with index above 5 are summed together to form the residual.

2. Secondly, the residual is checked for number of extrema and energy to make sure it is the trend. If it is not the trend, then the last IMF is joined with the residual to form the new residual. Additionally, if the last IMF has less than

4 extrema, then add this to the residual. Repeat this step until the residual is the trend.

3. Finally, if the number of components after this treatment is less than $s+1$, then the last component is assumed to be the residual and the missing components are assumed to be the last IMFs, i.e. the low frequency components.



(a) Bar chart.

(b) Boxplot.

Figure C.1: Distribution of the number of components extracted using the EMD.

After this unification of the EMD result, the distribution of the number of components is as shown in Fig. C.2.

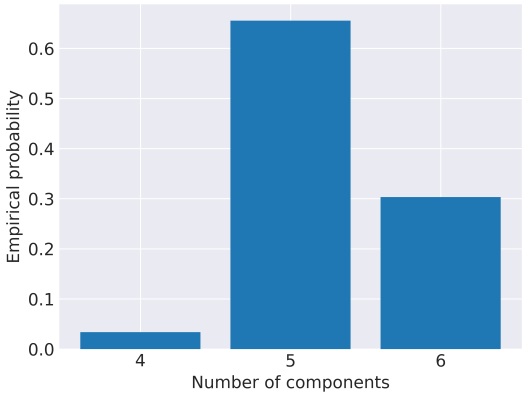
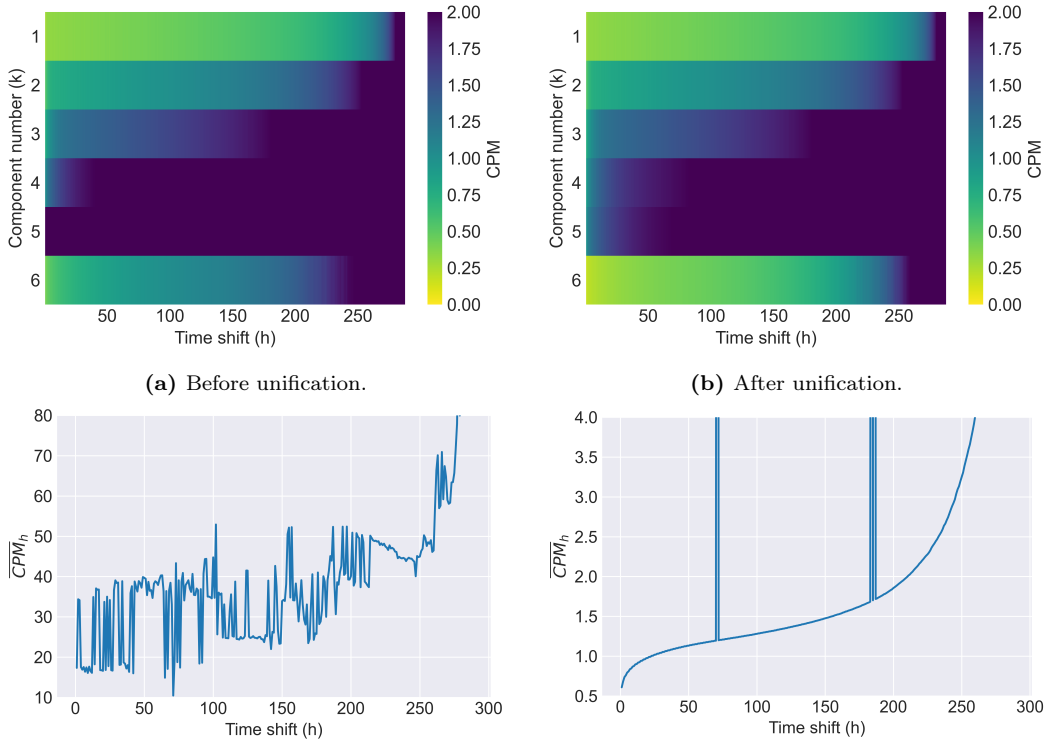


Figure C.2: Bar chart displaying the distribution of the number of components extracted using the EMD after the unification.

Comparing the average consistency performance measure (CPM) at time shift h, i.e. $\overline{\text{CPM}}_h$, defined in Section 6.3, for the EMD before and after the unification procedure, it is found that after unification $\overline{\text{CPM}}_1$ is 0.61, while before unification it is 17.3 which supports the use of the unification. The performance measures $\text{CPM}_{k,h}$ and $\overline{\text{CPM}}_h$ are plotted with respect to the time shift in Fig. C.3 both before and after unification. The gain of using the unification in terms of decreasing the CPM is clearly seen in these plots.



(a) Before unification.

(b) After unification.

(c) Before unification.

(d) After unification.

Figure C.3: The plots in the top row show colour plots of the CPM before and after unification. The bottom row shows $\overline{\text{CPM}}_h$ before and after unification.

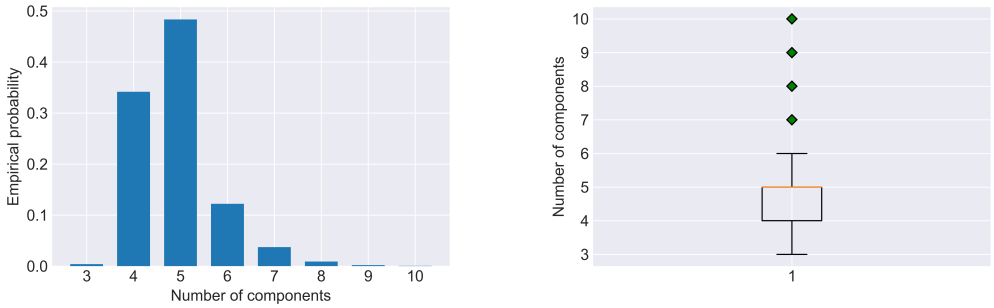
C.2 PDE-EMD

Seeing as the PDE-EMD method is similar to the EMD method in the sense that both methods are based on the SP, a similar analysis is made. As is the case for the EMD, the number of components obtained in each window is not constant. Thus, we start by defining some heuristic rules which limit the number of IMFs and make the IMFs in neighbouring windows more consistent. In Figs. C.4a and C.4b, a bar chart and a boxplot, respectively, of the number of components can be seen.

It is noticed that the number of components is concentrated below 6. Additionally, by inspecting individual IMFs we notice that multiple similar low frequency IMFs are extracted in each window. The number of low frequency IMFs which has been found has a large influence on how the resulting residual looks. Thus, for the sake of consistency the low frequency IMFs should be added to the residual. These two observations gives us the two heuristic rules which we apply to make a unification procedure of the decomposition which should result in a more consistent decomposition.

1. All components with index above 4 are added together to form the residual.
2. If an IMF has less than 4 extrema, it is added to the residual.

3. Finally, as with the EMD if the number of components after this treatment is less than 5, then the last component is assumed to be the residual and the missing components are assumed to be the last IMFs, i.e. the low frequency components.



(a) Bar chart. **(b)** Boxplot.

Figure C.4: Distribution of the number of components extracted using the PDE-EMD. Results found using N0 boundary conditions and $T = 5$.

After applying the unification procedure, the distribution of components is as seen in Fig. C.5. Notice that after unification the component before the 5th component has often become a part of the residual.

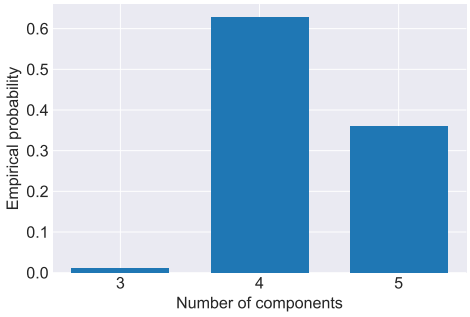
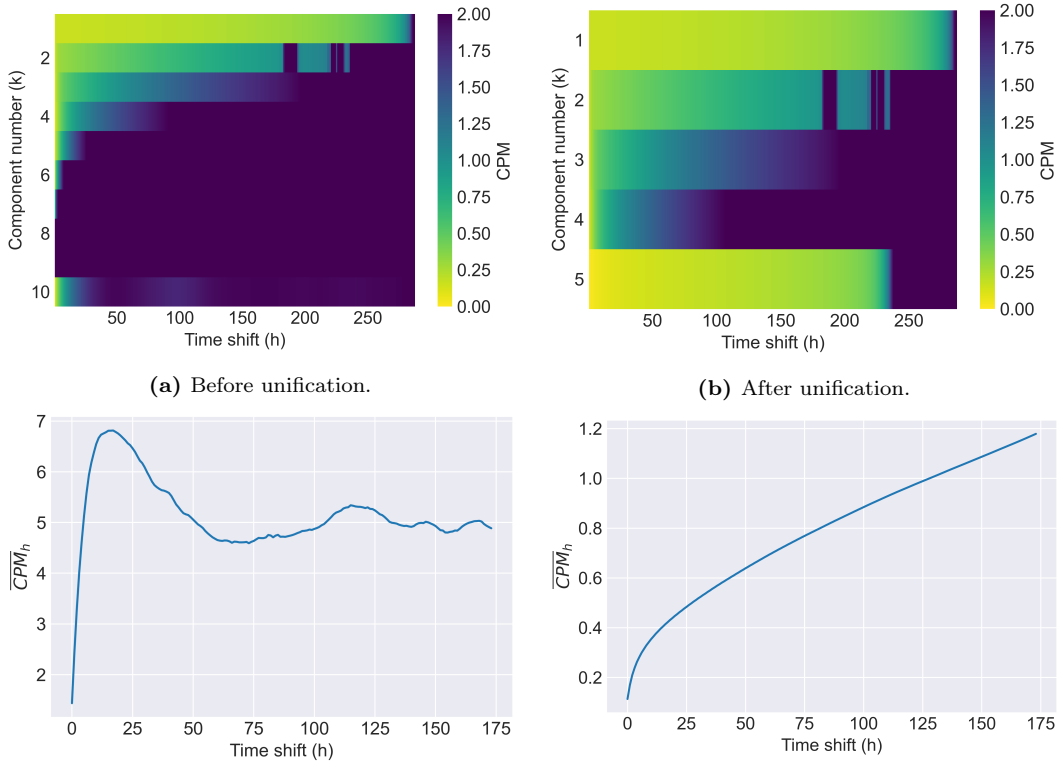


Figure C.5: Distribution of the number of components extracted using the PDE-EMD after unification. Results found using N0 boundary conditions and $T = 5$.

The CPM defined in Eq. (6.5) has been computed for decompositions before and after applying the heuristic rules and the result can be seen in Fig. C.6.



(a) Before unification.

(b) After unification.

(c) Before unification.

(d) After unification.

Figure C.6: The top row shows colour plots of the CPM before and after unification. The bottom row shows $\overline{\text{CPM}}_h$ before and after unification. Plots made for N0 boundary conditions and $T = 6$.

Bibliography

[ADV22] M. S. Andersen, J. Dahl, and L. Vandenberghe. "Quadratic programming." https://cvxopt.org/userguide/coneprog.html#quadratic-programming (visited: 15/03-2022). (2022).

[Apo67] T. M. Apostol, *Calculus*, 2nd ed. John Wiley & Sons, Inc., 1967, vol. 02, ISBN: 978-0-471-00007-5.

[AS17] D. Ambach and W. Schmid, "A new high-dimensional time series approach for wind speed, wind direction and air pressure forecasting," *Energy*, vol. 135, 833–850, 2017, ISSN: 0360-5442. DOI: 10.1016/j.energy.2017.06.137.

[AVK21] A. A. Andersen, M. V. Vejling, and M. S. Kaaber. "Forecasting wind power production." https://github.com/DeTreMusketerer/Wind_Power_Forecasting-P9/blob/main/P9___De_Tre_Musketerer.pdf (visited: 30/05-2022). (2021).

[Bed63] E. Bedrosian, "A product theorem for Hilbert transforms," *Proc. of the IEEE*, vol. 51, no. 5, pp. 868–869, 1963. DOI: 10.1109/PROC.1963.2308.

[Bis06] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer, 2006, ISBN: 0387310738. DOI: 10.5555/1162264.

[Bok+19] N. Bokde, A. E. Feijoo, D. Villanueva, and K. Kulat, "A review on hybrid empirical mode decomposition models for wind speed and wind power prediction," *Energies*, vol. 12, pp. 1–42, 2019. DOI: 10.3390/en12020254.

[BT07] T. G. Barbounis and I. B. Theocharis, "Locally recurrent neural networks for wind speed prediction using spatial correlation," *Information Sciences*, vol. 177, pp. 5775–5797, 2007.

[Bur+01] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Wind energy handbook*. J. Wiley, 2001, ISBN: 0471489972.

[CD14] C. Croonenbroeck and C. Dahl, "Accurate medium-term wind power forecasting in a censored classification framework," *Energy*, vol. 73, 221–232, 2014. DOI: 10.1016/j.energy.2014.06.013.

[CDS98] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, 1998. DOI: 10.1137/S1064827596304010.

[DAB09] E. H. S. Diop, R. Alexandre, and A.-O. Boudraa, "A PDE characterization of the intrinsic mode functions," in *Proc. 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2009, pp. 3429–3432. DOI: 10.1109/ICASSP.2009.4960362.

[DAB10] E. H. S. Diop, R Alexandre, and A. Boudraa, "Analysis of intrinsic mode functions: A PDE approach," *IEEE signal processing letters*, vol. 17, no. 4, pp. 398–401, 2010, ISSN: 1070-9908.

[DAP13] E. H. Diop, R. Alexandre, and V. Perrier, "A PDE based and interpolation-free framework for modeling the sifting process in a continuous domain," *Advances in Computational Mathematics*, vol. 38, 2013. DOI: 10.1007/s10444-011-9260-x.

[Dau94] I. Daubechies, *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, 1994, ISBN: 0898712742.

[DBM19] A. Delgado-Bonal and A. Marshak, "Approximate entropy and sample entropy: A comprehensive tutorial," *Entropy*, vol. 21, no. 6, pp. 541–, 2019.

[dBo78] C. de Boor, *A Practical Guide to Splines*. Springer Verlag, 1978, ISBN: 9780387953663.

[Dem21] L. Demanet, *Waves and imaging*, 2021. [Online]. Available: http://math.mit.edu/icg/resources/notes367.pdf.

[DLN05] E. Deléchelle, J. Lemoine, and O. Niang, "Empirical mode decomposition: An analytical approach for sifting process," *Signal Processing Letters, IEEE*, vol. 12, pp. 764 –767, 2005. DOI: 10.1109/LSP.2005.856878.

[DLW11] I. Daubechies, J. Lu, and H.-T. Wu, "Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 243–261, 2011, ISSN: 1063-5203. DOI: 10.1016/j.acha.2010.08.002.

[DPB21] S. Das, B. R. Prusty, and K. Bingi, "Review of adaptive decomposition-based data preprocessing for renewable generation rich power system applications," *Journal of Renewable and Sustainable Energy*, vol. 13, no. 6, p. 062 703, 2021. DOI: 10.1063/5.0070140.

[DSB19] E. H. Diop, K. Skretting, and A.-O. Boudraa, "Multicomponent AM-FM signal analysis based on sparse approximation," *IET Signal Processing*, vol. 14, 2019. DOI: 10.1049/iet-spr.2019.0110.

[Ene19] Energinet. "Introduktion til elmarkedet - kort introduktion til engros- og detailmarkedet." https://energinet.dk/-/media/41F2C6A30A834208B0225DB3132560C7.PDF (visited: 10/03-2022). (2019).

[Ene21] ——, "Strategi: Nye vinde." https://energinet.dk/-/media/17DDD9C8181E42EEBFC118E89E950D92.pdf (visited: 10/03-2022). (2021).

[Ene22a] ——, "Eltransmissionsnettet i dag." https://energinet.dk/El/Eltransmissionsnettet/Elnettet-i-dag (visited: 10/03-2022). (2022).

[Ene22b] ——, "Energinets opgaver." https://energinet.dk/Om-os/Opgaver (visited: 10/03-2022). (2022).

[Ene22c] ——, "Hvor kommer strømmen fra." https://energinet.dk/El/Gron-el/Deklarationer/Hvor-kommer-stroemmen-fra (visited: 10/03-2022). (2022).

[Ene22d] ——, "Roller og opgaver på elmarkedet." https://energinet.dk/El/Elmarkedet/Roller-paa-elmarkedet (visited: 10/03-2022). (2022).

[Eva10] L. C. Evans, *Partial differential equations*, 2nd ed., ser. Graduate studies in mathematics. American Mathematical Society, 2010, ISBN: 9780821849743.

[FO07] M. G. Frei and I. Osorio, "Intrinsic time-scale decomposition: Time–frequency–energy analysis and real-time filtering of non-stationary signals," *Proc. of the Royal Society. A, Mathematical, physical, and engineering sciences*, vol. 463, no. 2078, pp. 321–342, 2007, ISSN: 1364-5021.

[Fol92] G. B. Folland, *Fourier analysis and its applications*, ser. The Sally Series. American Mathematical Society, 1992, ISBN: 9780821847909.

[FR13] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. 2013, ISBN: 9780817649487 0817649484. DOI: 10.1007/978-0-8176-4948-7.

[Gab46] D. Gabor, "Theory of communication," *Journal of the Institution of Electrical Engineers*, pp. 429–457, 1946.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, ser. Adaptive computation and machine learning. The MIT Press, 2016, ISBN: 9780262035613.

[GLH15] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*, 1st ed., ser. Intelligent Systems Reference Library. Springer, 2015. DOI: 10.1007/978-3-319-10247-4.

[Guo+16] B. Guo, S. Peng, X. Hu, and P. Xu, "Complex-valued differential operator-based method for multi-component signal separation," *Signal Processing*, vol. 132, 2016. DOI: 10.1016/j.sigpro.2016.09.015.

[Han+20] S. Hanifi, X. Liu, Z. Lin, and S. Lotfian, "A critical review of wind power forecasting methods-past, present and future," *Energies*, vol. 13, 2020. DOI: 10.3390/en13153764.

[HHY17] J. Huang, C. Huang, and L. Yang, "Mono-frequency signals: Model and construction," *Digital Signal Processing*, vol. 69, 2017. DOI: 10.1016/j.dsp.2017.06.020.

[HK13] B. Huang and A. Kunoth, "An optimization based empirical mode decomposition scheme," *Journal of Computational and Applied Mathematics*, vol. 240, pp. 174–183, 2013, ISSN: 0377-0427. DOI: 10.1016/j.cam.2012.07.012.

[HPH13] X. Hu, S. Peng, and W.-L. Hwang, "Multicomponent AM-FM signal separation and demodulation with null space pursuit," *Signal, Image and Video Processing*, vol. 7, 2013. DOI: 10.1007/s11760-012-0354-9.

[HPH15] ——, "Adaptive integral operators for signal separation," *Signal Processing Letters, IEEE*, vol. 22, pp. 1383–1387, 2015. DOI: 10.1109/LSP.2014.2352340.

[HS11] T. Hou and Z. Shi, "Adaptive data analysis via sparse time-frequency representation.," *Advances in Adaptive Data Analysis*, vol. 3, pp. 1–28, 2011. DOI: 10.1142/S1793536911000647.

[HS13a] ——, "Sparse time-frequency representation of nonlinear and nonstationary data," *Science China Mathematics*, vol. 56, 2013. DOI: 10.1007/s11425-013-4733-7.

[HS13b] T. Y. Hou and Z. Shi, "Data-driven time–frequency analysis," *Applied and Computational Harmonic Analysis*, vol. 35, no. 2, pp. 284–308, 2013, ISSN: 1063-5203. DOI: 10.1016/j.acha.2012.10.001.

[HS13c] ——, *Sparse time-frequency decomposition by adaptive basis pursuit*, 2013.

[HS14] N. Huang and S. Shen, *Hilbert-Huang Transform and Its Applications*, 2nd ed. Interdisciplinary mathematical sciences, 2014, ISBN: 978-981-256-376-7. DOI: 10.1142/5862.

[HST13] T. Hou, Z. Shi, and P. Tavallali, "Convergence of a data-driven time-frequency analysis method," *Applied and Computational Harmonic Analysis*, vol. 37, 2013. DOI: 10.1016/j.acha.2013.12.004.

[Hua+17] C. Huang, L. Tan, Q. Zhang, and L. Yang, "Constructions of ϵ-mono-components and mathematical analysis on signal decomposition algorithm," *Applied Mathematics and Computation*, vol. 293, pp. 555–564, 2017, ISSN: 0096-3003. DOI: 10.1016/j.amc.2016.08.036.

[Hua+98] N. E. Huang *et al.*, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," vol. 454, pp. 903–995, 1998, ISSN: 1364-5021. DOI: 10.1098/rspa.1998.0193.

[HYX15] A. Hu, X. Yan, and L. Xiang, "A new wind turbine fault diagnosis method based on ensemble intrinsic time-scale decomposition and WPT-fractal dimension," *Renewable energy*, vol. 83, pp. 767–778, 2015, ISSN: 0960-1481.

[HYY15] C. Huang, L. Yang, and L. Yang, "ϵ-mono-component: Its characterization and construction," *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 234–243, 2015. DOI: 10.1109/TSP.2014.2370950.

[Jai16] P. Jain, *Wind Energy Engineering*, 2nd ed. McGraw-Hill Education, 2016, ISBN: 978-0-07-184384-3.

[Jak19] P. K. Jakobsen, *An introduction to partial differential equations*, 2019.

[Joh12] M. Johansson, "The Hilbert transform," M.S. thesis, Växjö University, 2012.

[Jos13] J. Jost, *Partial Differential Equations*, 3rd ed., ser. Graduate Texts in Mathematics. Springer New York, 2013, vol. 214, ISBN: 1461448085.

[Kre11] E. Kreyzig, *Advanced engineering mathematics*, 10th ed. Wiley-Blackwell, 2011, ISBN: 9780470646137.

[LC19] H. Liu and C. Chen, "Data processing strategies in wind energy forecasting models and applications: A comprehensive review," *Applied Energy*, vol. 249, pp. 392–408, 2019. DOI: 10.1016/j.apenergy.2019.04.188.

[LDB21] M.-D. Liu, L. Ding, and Y.-L. Bai, "Application of hybrid model based on empirical mode decomposition, novel recurrent neural networks and the ARIMA to wind speed prediction," *Energy Conversion and Management*, 2021. DOI: 10.1016/j.enconman.2021.113917.

[Li+18] H. Li, X. Qin, D. Zhao, J. Chen, and P. Wang, "An improved empirical mode decomposition method based on the cubic trigonometric B-spline interpolation algorithm," *Applied Mathematics and Computation*, vol. 332, no. C, pp. 406–419, 2018. DOI: 10.1016/j.amc.2018.02.039.

[Liu+10] H. Liu, H.-Q. Tian, C. Chen, and Y. fei Li, "A hybrid statistical method to predict wind speed and wind power," *Renewable Energy*, 2010. DOI: 10.1016/j.renene.2009.12.011.

[LSH15] C. Liu, Z. Shi, and T. Y. Hou, "On the uniqueness of sparse time-frequency representation of multiscale data," *Multiscale modeling & simulation*, vol. 13, no. 3, pp. 790–811, 2015, ISSN: 1540-3459.

[LSH17] C. Liu, Z. Shi, and T. Hou, "A two-level method for sparse time-frequency representation of multiscale data," *Science China Mathematics*, vol. 60, 2017. DOI: 10.1007/s11425-016-9088-9.

[Lyd+16] M. Lydia, S. Kumar, A. Selvakumar, and G. E. Prem Kumar, "Linear and non-linear autoregressive models for short-term wind speed forecasting," *Energy Conversion and Management*, vol. 112, pp. 115–124, 2016. DOI: 10.1016/j.enconman.2016.01.007.

[MNN05] H. Madsen, H. Nielsen, and T. Nielsen, "A tool for predicting wind power production of off-shore wind plants," in *Proc. of the Copenhagen Offshore Wind Conference & Exhibition*, 2005.

[MZ94] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, pp. 3397 –3415, 1994. DOI: 10.1109/78.258082.

[MZR14] P. Mandal, H. Zareipour, and W. Rosehart, "Forecasting aggregated wind power production of multiple wind farms using hybrid wavelet-PSO-NNs," *International Journal of Energy Research*, vol. 38, 2014. DOI: 10.1002/er.3171.

[Nie+07] H. Nielsen *et al.*, "Improvement and automation of tools for short term wind power forecasting," in *Proc. European Wind Energy Conference and Exhibition 2007*, vol. 1, EWEC, 2007.

[OO18] M. Oberguggenberger and A. Ostermann, *Analysis for Computer Scientists: Foundations, Methods, and Algorithms*, 2nd ed. Springer, 2018, ISBN: 978-3-319-91154-0.

[Pen+20] Z. Peng *et al.*, "A novel deep learning ensemble model with data denoising for short-term wind speed forecasting," *Energy Conversion and Management*, 2020. DOI: 10.1016/j.enconman.2020.112524.

[PH08] S. Peng and W.-L. Hwang, "Adaptive signal decomposition based on local narrow band signals," *IEEE Transactions on Signal Processing*, vol. 56, pp. 2669 –2676, 2008. DOI: 10.1109/TSP.2008.917360.

[PH10] ——, "Null space pursuit: An operator-based approach to adaptive signal separation," *IEEE Transactions on Signal Processing*, vol. 58, pp. 2475–2483, 2010. DOI: 10.1109/TSP.2010.2041606.

[PS+09] J. Palomares-Salas, J. J. de la Rosa, J. Ramiro-Leo, J. Melgar, A. Agüera-Pérez, and A. Moreno-Munoz, "ARIMA vs. neural networks for wind speed forecasting," in *Proc. Computational Intelligence for Measurement Systems and Applications*, IEEE, 2009, pp. 129–133, ISBN: 978-1-4244-3819-8. DOI: 10.1109/CIMSA.2009.5069932.

[Pus+17] S. Pushpendra, J. S. Dutt, P. R. Kumar, and S. Kaushik, "The Fourier decomposition method for nonlinear and non-stationary time series analysis," *Proc. of the Royal Society A*, 2017. DOI: 10.1098/rspa.2016.0871.

[Qia05] T. Qian, "Characterization of boundary values of functions in Hardy spaces with applications in signal analysis," *Journal of Integral Equations and Applications*, vol. 17, no. 2, pp. 159 –198, 2005. DOI: 10.1216/jiea/1181075323.

[Qia06] ——, "Mono-components for decomposition of signals," *Mathematical Methods in The Applied Sciences*, vol. 29, pp. 1187–1198, 2006.

[Qia+09] T. Qian, R. Wang, Y. Xu, and H. Zhang, "Orthonormal bases with nonlinear phases," *Advances in computational mathematics*, vol. 33, no. 1, pp. 75–95, 2009, ISSN: 1019-7168.

[Qia+19] Z. Qian, Y. Pei, H. Zareipour, and N. Chen, "A review and discussion of decomposition-based hybrid models for wind energy forecasting applications," *Applied Energy*, vol. 235, pp. 939–953, 2019. DOI: 10.1016/j.apenergy.2018.10.080.

[RF08] G. Rilling and P. Flandrin, "One or two frequencies? The empirical mode decomposition answers," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 85–95, 2008. DOI: 10.1109/TSP.2007.906771.

[RM00] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 278, no. 6, H2039–H2049, 2000. DOI: 10.1152/ajpheart.2000.278.6.H2039.

[SECB22] U. B. de Souza, J. P. L. Escola, and L. da Cunha Brito, "A survey on Hilbert-Huang transform: Evolution, challenges and solutions," *Digital Signal Processing*, vol. 120, 2022, ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2021.103292.

[SH07] G. Sideratos and N. Hatziargyriou, "An advanced statistical method for wind power forecasting," *IEEE Transactions on Power Systems*, vol. 22, pp. 258 –265, 2007. DOI: 10.1109/TPWRS.2006.889078.

[Sin+15] P. Singh, S. D. Joshi, R. K. Patney, and K. Saha, *The Hilbert spectrum and the energy preserving empirical mode decomposition*, 2015. arXiv: 1504.04104 [cs.IT].

[Sin16] V. Singh, "Application of artificial neural networks for predicting generated wind power," *International Journal of Advanced Computer Science and Applications*, vol. 7, 2016. DOI: 10.14569/IJACSA.2016.070336.

[Smi05] J. S. Smith, "The local mean decomposition and its application to EEG perception data," *Journal of The Royal Society Interface*, vol. 2, no. 5, pp. 443–454, 2005. DOI: 10.1098/rsif.2005.0058.

[Smi07] J. O. Smith III, *Mathematics of the discrete Fourier transform (DFT) with audio applications*, 2nd ed. Center for Computer Research in Music and Acoustics (CCRMA), 2007.

[SPC20] N. Safari, G. Price, and C. Chung, "Analysis of empirical mode decomposition-based load and renewable time series forecasting," in *Proc. 2020 IEEE Electric Power and Energy Conference*, 2020, pp. 1–6. DOI: 10.1109/EPEC48502.2020.9320072.

[SS17] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications - With R Examples*, 4th ed. Springer, 2017.

[SV13] T. Sivanagaraja and K. Veluvolu, "A hybrid approach for short-term forecasting of wind speed," *The Scientific World Journal*, vol. 2013, p. 548 370, 2013. DOI: 10.1155/2013/548370.

[Tes10] G. Teschl, *Topics in Real Analysis*, 1st ed. American Mathematical Society Providence, 2010.

[THS14] P. Tavallali, T. Y. Hou, and Z. Shi, "Extraction of intrawave signals using the sparse time-frequency representation method," *Multiscale Modeling & Simulation*, vol. 12, no. 4, pp. 1458–1493, 2014. DOI: 10.1137/140957767.

[Tou+21] J.-F. Toubeau, P.-D. Dapoz, J. Bottieau, A. Wautier, Z. D. Grève, and F. Vallée, "Recalibration of recurrent neural networks for short-term wind power forecasting," *Electric Power Systems Research*, 2021. DOI: 10.1016/j.epsr.2020.106639.

[TW] A. Tveito and R. Winther, *Introduction to Partial Differential Equations: A Computational Approach*, ser. Texts in Applied Mathematics. Springer Berlin Heidelberg, vol. 29, ISBN: 354022551X.

[Van10] L. Vandenberghe. "The CVXOPT linear and quadratic cone program solvers." https://www.seas.ucla.edu/~vandenbe/publications/coneprog.pdf (visited: 15/03-2022). (2010).

[WH09] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Advances in Data Science and Adaptive Data Analysis*, vol. 1, pp. 1–41, 2009.

[WMV18] H. Wang, R. Mann, and E. R. Vrscay, *A novel foward-PDE approach as an alternative to empirical mode decomposition*, 2018. arXiv: 1802.00835 [eess.SP].

[XHY19] W. Xu, H. Hu, and W. Yang, "Energy time series forecasting based on empirical mode decomposition and FRBF-AR model," *IEEE Access*, vol. 7, pp. 36 540–36 548, 2019. DOI: 10.1109/ACCESS.2019.2902510.

[Yan+14] L. Yang, Z. Yang, F. Zhou, and L. Yang, "A novel envelope model based on convex constrained optimization," *Digital Signal Processing*, vol. 29, 2014. DOI: 10.1016/j.dsp.2014.02.017.

[ZCA16] F. Ziel, C. Croonenbroeck, and D. Ambach, "Forecasting wind power – modeling periodic and non-linear effects under conditional heteroscedasticity," *Applied Energy*, vol. 177, 285–297, 2016, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2016.05.111.

[Zha+19] J. Zhang, J. Yan, D. Infield, Y. Liu, and F. sang Lien, "Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and gaussian mixture model," *Applied Energy*, vol. 241, pp. 229–244, 2019, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2019.03.044.

[Zha+20] Y. Zhang, Q. Ai, F. Xiao, R. Hao, and T. Lu, "Typical wind power scenario generation for multiple wind farms using conditional improved Wasserstein generative adversarial network," *International Journal of Electrical Power & Energy Systems*, vol. 114, p. 105 388, 2020, ISSN: 0142-0615. DOI: 10.1016/j.ijepes.2019.105388.

[Zhe+13] Z. Zheng, Y. Y. Chen, X. Zhou, M. Huo, B. Zhao, and M. Y. Guo, "Short-term wind power forecasting using empirical mode decomposition and RBFNN," *International Journal of Smart Grid and Clean Energy*, vol. 2, pp. 192–199, 2013.

[Zho+22] W. Zhou, Z. Feng, Y. Xu, X. Wang, and H. Lv, "Empirical Fourier decomposition: An accurate signal decomposition method for nonlinear and non-stationary time series analysis," *Mechanical Systems and Signal Processing*, vol. 163, p. 108 155, 2022, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2021.108155.