Machine Learning Applications for Thermal Manufacturing Processes

DISSERTATION

zur Erlangung des Doktorgrades
der Naturwissenschaften (Dr. rer. nat.)
der Fakultät für Physik
der Universität Regensburg

vorgelegt von
Peter Weiderer
aus Zwiesel

September 2019

Promotionsgesuch eingereicht am: 26. September 2019

Die Arbeit wurde angeleitet von: Prof. Dr. Elmar W. Lang

Contents

1 Introduction **1**

1.1 Background and Motivation 1

1.1.1 Thermal Manufacturing Process: Metal Casting 2

1.1.2 Blind Source Separation with Matrix Factorisation 4

1.1.3 Literature Review: Machine Learning and Data Mining in Manufacturing 5

1.2 Scope of this Thesis and Research Questions 8

1.3 Structure of this Thesis 8

2 Theory and Methods **11**

2.1 Nonnegative Matrix Factorisation 11

2.1.1 Unsupervised Learning and Matrix Factorisation 11

2.1.1.1 Low Rank Matrix Approximation 13

2.1.1.2 Singular Value Decomposition and Principal Component Analysis 14

2.1.1.3 Moore-Penrose-Inverse 16

2.1.2 The NMF Problem 16

2.1.2.1 Why Nonnegativity? 18

2.1.2.2 The Cost Function 19

2.1.2.3 Strategies for NMF 19

2.1.2.4 Initialisation Techniques 21

2.1.2.5 An SVD-based Initialisation 22

2.1.3 Alternating Least Squares and Hierarchical Alternating Least Squares 23

2.1.4 Alternating Least Squares Algorithm 24

2.1.5 Extensions to ALS 26

2.1.5.1 L_1- and L_2-regularisations 26

2.1.5.2 L_2-regularisation 27

2.1.5.3 L_1-regularisation 28

2.1.6 Hierarchical Alternating Least Squares 28
2.1.7 HALS with Regularisation 33
2.2 Linear Regression 33
2.2.1 Linear Regression with NMF Preprocessing 34
2.2.2 Semi-supervised Learning 38

3 Practical Part **41**
3.1 The Manufacturing Process: Gravity Mould Casting 41
3.1.1 Data Collection during the Casting Process 47
3.2 The NMF Approach for Time Aeries of Physical Quantities 48
3.2.1 Physics: The Heat Equation 48
3.2.2 Deriving the Matrix Decomposition 49
3.2.3 A Physics Inspired Initialisation Strategy for NMF 53
3.2.3.1 Initialisation for Temperature Time Curves 55
3.2.4 The NMF Model for Temperature Time Curves 56
3.2.4.1 Decomposition of Toy Data 59
3.3 Application to Real-World Datasets 61
3.3.1 Real-World Data from a Thermal Manufacturing Process 61
3.3.1.1 Dataset 1: A Simple Process 61
3.3.1.2 Dataset 2: A Complex Process 62
3.3.2 Results 63
3.3.2.1 Results for Dataset 1 63
3.3.2.2 Results for Dataset 2 70
3.3.2.3 Study on Convergence Speed 72
3.3.3 Remarks about the NMF-based Decomposition Approach 73
3.3.4 The Effect of Regularisation 74
3.4 An Application in Process Monitoring 81
3.4.1 A Data-driven Soft Sensor 81
3.4.2 Data Generation 82
3.4.2.1 Dependent Variable: Measurements of Layer Thickness . 82
3.4.2.2 Independent Variables: NMF Component Processes . . . 85
3.4.3 Model Training 86
3.4.4 Discussion and Interpretation 91
3.4.4.1 Application as a Monitoring System 92
3.4.4.2 Limitations of the Approach 94

4 Conclusion **97**
4.1 Discussion and Interpretation 97
4.1.1 Comments on the NMF-based Approach 97
4.1.1.1 Comparison with other Matrix Decomposition Techniques 98
4.1.1.2 Physically Inspired Machine Learning 99
4.1.1.3 Initialisation Strategies 100
4.1.2 Comments on the NMF-based Virtual Sensor 101
4.1.2.1 Comparison to other Approaches 101
4.1.3 Limitations 102
4.2 Summary of the Main Results 103
4.3 Outlook and Further Research 105

A Additional NMF Results **107**
A.1 Initialisations 108
A.2 Model Transfer 111
A.3 PCA and ICA Results 113

List of Figures **115**

References **121**

Chapter 1

Introduction

In this thesis I present the results of my investigation of possible applications of machine learning methods for thermal manufacturing processes. The main goal is the design of new approaches to extract information from sensory data generated during manufacturing processes. The central theme throughout this thesis is a technique called **Nonnegative Matrix Factorisation (NMF)** and its ability to decompose sensory data into physically meaningful components. The data used in this thesis was provided by the German car manufacturer BMW Group AG and the findings discussed in this thesis found immediate implementation in one of their production plants.

1.1 Background and Motivation

Industry 4.0 is the key word for a wide variety of recent developments in automation and data exchange in the manufacturing industry. The term includes innovations in cloud computing, internet of things as well as cognitive computing and is sometimes referred to as the fourth industrial revolution. One major key pillar in this ongoing trend are recent advances in the use of machine learning techniques in the production environment, which help in the early detection of defects and production failures, thus increasing productivity and quality (see [106, 67] for recent reviews).

The data used in this thesis was provided by BMW Group AG and stems from their metal foundry in Landshut, Germany. Metal casting is a thermal manufacturing process which is generally unstable and the quality of the products depends on many interacting variables, which turns the defect analysis into a challenging task as it includes a lot of trial and error approaches. Data analysis has always been part of the problem-solving strategy practised by engineers, but still remains tedious and often ineffective. With increasing complexity in manufacturing, the amount of generated process data to be analysed with regular statistical

methods soon becomes too large and more sophisticated techniques are needed. What is needed are tools to automatically extract the most important information from large datasets and this is where machine learning comes into play.

Machine learning is the general term for algorithms that can be used to teach a computer to perform a task without explicit programming. Instead, the computer is supposed to "learn" the necessary rules by detecting patterns and inference from data [9]. Machine learning as a field of study has a long history, with earliest publications during the 1960s [94], but limited computational processing power made most algorithms unviable for any usage in production. Due to the rising computational abilities of modern times, machine learning has now become an already widely used tool to analyse large and complex datasets and its applications in industry are now being widely discussed by researches and practitioners alike. However, the field is very broad and available methods, promising to solve specific problems, are diverse. This is why application studies are needed to provide a "handbook" of methods that have been proven to yield results that can be implemented in industrial applications. This is also the general motivation behind this thesis and, after having realised the difficulties in analysing industrial data, I started to specifically focus on feature learning algorithms like NMF. The reason for this is that sometimes not even domain knowledge is enough to evaluate the sensory data generated during manufacturing processes.

1.1.1 Thermal Manufacturing Process: Metal Casting

In metalworking and jewellery making, "casting" is a process in which a liquid metal is delivered into a cavity (it is usually delivered by a crucible) that contains a hollow form of the intended shape. This casting cavity is also called "mould" and in this thesis, I am going to switch between the two expressions. The metal is poured into the mould through a hollow channel, called a sprue. Both metal and mould are then cooled, and the metal part (the casting) is extracted. Casting is mostly used for making complex shapes that would be difficult or uneconomical to be fabricated by other methods [103, 104]. In Germany, a significant portion of the casting parts are produced for the automotive industry. Fig. 1.1 shows different car parts made out of aluminium and where they are placed inside of the car. In this thesis, I am mainly going to focus on one specific process called "gravity casting," which is shown in fig. 1.2. Gravity casting is a typical process to produce parts with rather complex geometry like cylinder heads or crankcases (see fig. 1.1). In gravity casting, the liquid metal is poured into a cavity from the top, i.e. the main driving force for the filling process is gravity.

The source of data used in this thesis is the sensory data collected during the continuous production of metal parts with a casting machine. The casting industry has a thousand years

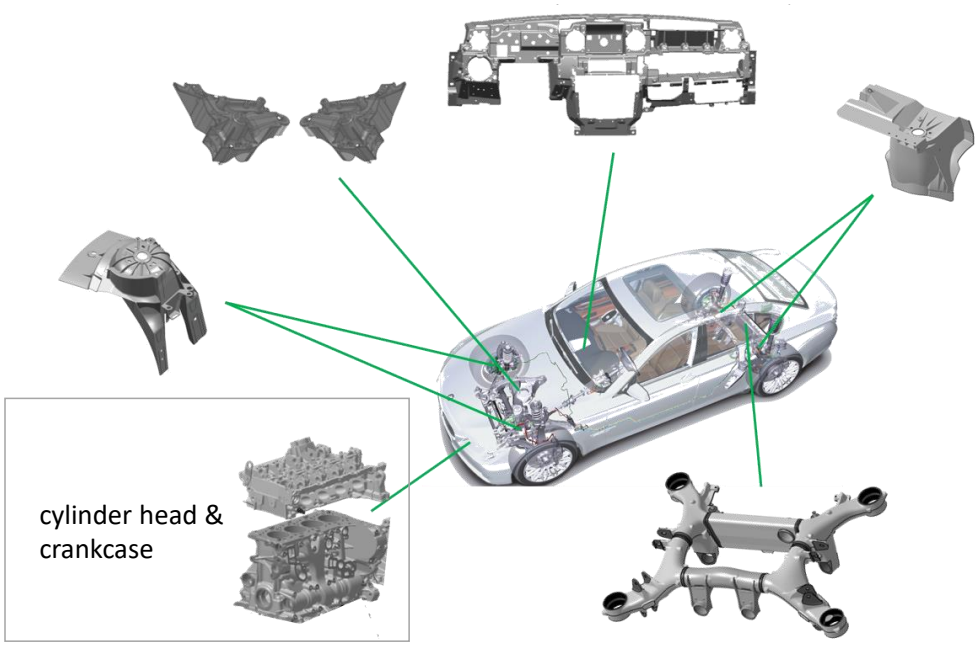

Fig. 1.1 Car parts produced by metal casting. The image has been provided by the BMW Group AG.

of history and its roots trace back back to the mining industry. This traditional background is one of the reasons why modern methods for process monitoring and data mining have not been employed to a significant extent in this domain compared to other industries like semiconductor manufacturing. From an economic view point, the main motivation behind the development of data mining solutions in the metal casting industry has always been the reduction of scrap rate and the optimisation of processes. Metal casting is a complex physical process in which a variety of defects can occur and the same defects can usually have multiple causes. Typical defect types are related to solidification issues like cracks, cold runs or porosities. Apart from that, also an incorrect filling can cause defects like air entrapment or surface defects. The variety of defects and their possible causes fills books and very detailed documentations can be found in related engineering literature. Because of that, there is a definite need for tools and solutions to speed up the process of root cause analysis, and data mining methods appear to be promising candidates due to their ability to extract complex information and interdependencies from large amounts of process data collected during series production.

Fig. 1.2 Picture taken during the filling of the mould during a gravity casting process with a modern casting machine.

1.1.2 Blind Source Separation with Matrix Factorisation

Forming a quantitative model of the observed data, by stating suitable assumptions about the data generation process, is the goal of Blind Source Separation techniques (BSS) [27]. The use of BSS in this thesis is motivated by the idea idea of individually extracting information about the many interacting physical processes during a casting process from single sensor signals. In a BSS model, each underlying influencing factor can be seen as one of K different component processes. There is a variety of models, which assume the data to be generated by a set of K unknown and hidden sources that overlap. Such models are commonly referred to as latent variable models. The definition of "latent variable" depends upon the assumptions made about the data generation process. With BSS techniques, this superposition can be modelled and sometimes reversed, which yields a new representation of the observed data. Widely used BSS techniques are the principal component analysis (PCA) and the independent component analysis (ICA), which respectively assume the underlying components to be mutually uncorrelated or statistically independent. An example to introduce the concept of BSS is given in fig. 1.3. During the casting of a metal part, a temperature sensor, which is embedded into the steel cavity, records a temperature time series. This sensor signal is the result of multiple different parameters and physical mechanisms. Of course, the rising temperature stems from the heat transfer from liquid metal into the steel cavity and the decreasing temperature marks the solidification of the metal part. Yet slight changes in the initial metal temperature, the temperature of the steel cavity or the environmental

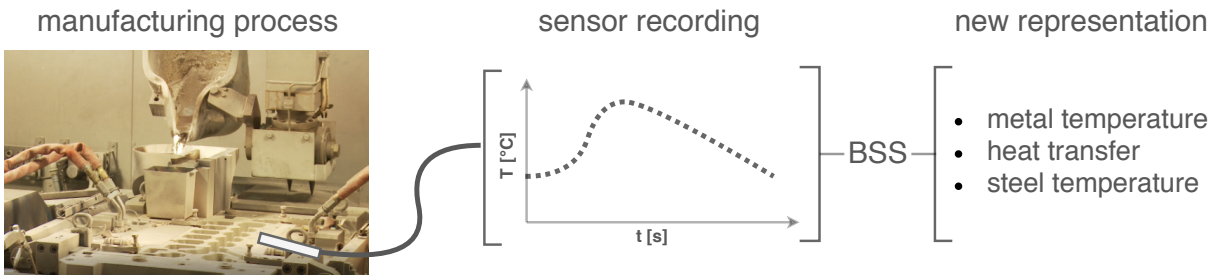

Fig. 1.3 Illustration of a standard blind source separation problem occuring in industrial manufacturing.

conditions all have a distinct effect on the shape of the sensor signal. If it is possible to model these distinct effects, the sensor signals can be represented in a new form, namely as the combination of different physical conditions present during the manufacturing process. This way, the information yield from single sensors that monitor the process is increased and the additional knowledge can be used by the process engineers. This combination of distinct effects, caused by process related mechanisms, is unravelled by a demixing system. If we model the superposition process with a linear model, the BSS model (i.e. the demixing system) schematically shown in fig. 1.3 can be expressed as a matrix factorisation

$$\mathbf{T} = \mathbf{W}\boldsymbol{\Theta}, \tag{1.1}$$

or written as a sum

$$T_{i*} = \sum_{i=1}^{K} W_{ik}\boldsymbol{\Theta}_{k*}. \tag{1.2}$$

If $\mathbf{T}$ contains the samples of the observed data as M-dimensional row vectors, they can be written as a linear combination of K components $\boldsymbol{\Theta}_{k*}$, using the coefficients W_{ik}, which can be interpreted as the contribution of a specific component within the respective observation.

1.1.3 Literature Review: Machine Learning and Data Mining in Manufacturing

As already mentioned, the variety of machine learning methods is quite extensive and the same goes for the applications of machine learning methods, which can be seen in a number of publications about the use of machine learning in manufacturing. Industrial processes typically generate a vast amount of machinery and sensory data, which first needs to be reduced into a manageable form to be evaluated and analysed by human experts. This dimensionality reduction can either be done manually, by designing features with prior

domain knowledge, or by the use of feature extraction algorithms. These techniques can be summarised with the term "data mining" or "automatic feature extraction." Automatic feature extraction reduces the time in analysing large data sets and can lead to previously unknown insights about processes if the right algorithm is chosen. Automatic feature extraction has already found various use cases in industrial applications.

One industry with a huge selection of data mining applications is semiconductor manufacturing. Here, the main research focus is also the analysis of failure causes and the prevention of possible defects and scrap. An important discussion about the non-triviality in choosing the right data preprocessing and the proper data mining method is given in [6] and the need for general guidelines is highlighted. Neural networks (NN) have repeatedly been studied for different application purposes in industry, because of their incorporated feature extraction ability and I am only going to deal with a fraction of the many publications about the application of neural networks in this domain. A main focus is the modelling of individual processes, real time equipment controlling, failure detection and the classification of process problems [64, 40].
Also standard statistical methods of nonparametric tests, like ANOVA, are used to discover root causes for defects or machine malfunctions, for example in [110, 4, 20].
Decision trees have always been popular, as they give interpretable results based on simple "Yes" and "No" rules. In [100], a classification and regression tree (CART) was used to analyse process data from semiconductor manufacturing and the obtained results have been compared to standard statistical methods. Compared to neural networks, decision trees do not transform the input data, but rather perform a feature selection and pick out the most relevant features by some statistical criteria. This way they are highly dependent on the correct choice of data preprocessing.
BSS techniques are also studied in this domain, and this is one reason which motivated the use of BSS in the course of this thesis. For example, independent component analysis (ICA) has been used specifically to model the generation of quality data stemming from a manufacturing process or to model process variations [98]. Projection Pursuit and mixture models have also been used to detect dependencies and irregularities in multivariate data sets [93]. PCA and ICA are also suitable algorithms for process monitoring or multivariate process monitoring (see [72, 42]), as they can be used to model latent features in process datasets. NMF, the method used in the practical work of this thesis, also has been used in this application domain to extract failure patterns from wafer test data [95].

Another industry, where data mining techniques are an established tool, is chemical engineering and manufacturing. This application domain is worth mentioning in this section as the generated process data in chemical manufacturing is similar to the process data

generated during metal casting (pressure and temperature time series, humidity etc.). Here, one of the main applications of machine learning methods is for process monitoring tasks, i.e. the monitoring of a desired quantity (for example a quality index). A common approach for this task is the design of "soft sensors" or "virtual sensors," which indirectly measure the desired quantity by modelling the process data and its variations. These approaches are typically based on the use of a matrix decomposition technique as a preprocessing step, like PCA [59, 44, 56]. More generally speaking, the design of data-driven soft sensors employs an automatic feature extraction and the fitting of a regression model.

In comparison to other fields of manufacturing, metal casting lacks wide-spread application of data mining techniques and the literature about studies in this domain yields only few results. A thorough study of different machine learning applications can be found in the thesis of Dörmann [31]. In his thesis, he outlines a general approach to integrate machine learning models or statistical models into the production process chain of casting processes. His work provides a descriptive study on existing machine learning methods and a concept work on how they can be applied in the metal casting industry. In contrast to his work, the focus of my thesis is going to be the study of a specific method and the design of an approach, which can readily be used in this manufacturing domain.
One goal for machine learning techniques in casting processes is defect prediction. Many casting defects can only be detected by cumbersome investigation methods like X-ray or computer tomography imaging. Those investigations are usually done hours after the actual casting process and the casting part has already gone through additional processing steps. If casting defects can be predicted in an early step, or even right after the casting of the part, these additional value chain steps can be saved and costs are reduced. This potential application is also discussed in [31]. The potential for saving costs will increase even further if processing steps are located across different plants. More recent publications about machine learning based control systems are for example [66] and [99].

To sum up, there is a wide variety of different data analysis and machine learning tools available, which can be applied to all kinds of data generated during manufacturing. Combined with domain knowledge, any mentioned algorithm can lead to an improvement in production and help in the diagnosis of defect causes. Yet every method has its limitations and can be more suitable for one application than the other. This is why methods need to be studied in different application domains to be able to correctly interpret and implement the obtained results.

1.2 Scope of this Thesis and Research Questions

This thesis was written with the intention of developing new analysis tools to aid engineers in their daily work to discover the root causes for various defects and to monitor the status of the processes and the quality of the produced parts. During the last three years, I have been working together with BMW Group in their plant in Landshut, Germany. I was given access to their databases, which store a tremendous amount of data collected during the production of motor parts like crankcases or cylinder heads. I decided to focus on the sensory data which is generated during the actual casting process and to develop methods to extract potentially useful information from it. Blind Source Separation methods proved to yield the remarkable ability to extract physically interpretable components. The main focus lies on a technique called **Nonnegative Matrix Factorisation (NMF)** and its ability to extract features from sensory data sets. This is why the main contribution of this thesis lies in the application of machine learning techniques to new domains and in investigating the results and their potential applications. Due to the close cooperation with the experts working in the plant, the explored methods have actually been developed to a degree, which makes them usable during running production. With these results, this thesis is at the forefront of machine learning applications in the casting industry.
Although the datasets used in the course of this thesis solely stem from metal casting processes, the developed principles are potentially applicable to other manufacturing processes.

1.3 Structure of this Thesis

This thesis is organised as follows: Chapter 2 introduces the theoretical background of the main algorithms used in the course of this thesis. After a general introduction to unsupervised learning with matrix factorisation techniques, NMF is introduced by providing a brief overview of multiple existing applications of the method. Afterwards, general concepts of implementational aspects concerning cost functions and optimisation strategies are discussed. The bulk of chapter 2 is dedicated to the derivation of the "Hierachical Alternating Least Squares" (HALS) algorithm, which is an efficient and flexible implementation to solve the NMF problem. This algorithm was used to estimate the decompositions of the sensory data in the practical works in chapter 3. An important research aspect of this thesis is the problem of properly initialising an NMF algorithm. The problem of non-convex optimisation and the necessity of a proper initialisation strategy is also discussed in chapter 2. Further, an initialisation strategy based on singular value decomposition (SVD) is derived. At the end of

chapter 2, I provide an introduction to linear regression and its combination with a feature extraction method like NMF.

The first part of chapter 3 gives an illustrative introduction to metal casting processes and the process which has provided the datasets. This part intentionally contains multiple photographs of the process and does not go into technical details because its main purpose is to provide the reader with an overall comprehension of metal casting processes.
Following this, I am going to present the interesting connection between the physical processes behind temperature time series and matrix decomposition techniques. This idea is introduced by first discussing the physical effects present during simple thermal processes and how they effect the temperature signal recorded by a thermocouple. In the end, a general form of the decomposition of a time series of any physical quantity is presented. Using this result, I am then going to present a novel initialisation strategy for any NMF decomposition of a time series of a physical quantity, which is based on designing initial guesses for the components by deriving the first order terms of the multivariate Taylor expansion of the physical quantity. The approach is tested and demonstrated by decomposing simulated toy datasets.

Having all the theoretical background set up, the outlined approach in chapter 3 is applied to real-world data sets collected from a gravity casting process. Two different datasets are used in this part and the obtained results and decompositions are discussed and interpreted. Each NMF component can be related to a process relevant quantity. In order to further study the application of the NMF-based approach, experiments with changing regularisation constraints are shown and discussed. At the end of chapter 3, the before mentioned combination of NMF with linear regression is applied to sensory data collected from another casting process. I am going to demonstrate how a "virtual sensor" for the release agent used in metal casting can be trained with this approach, a method which can be used to monitor the ongoing production.

In chapter 4, I am going to give a summary of the main findings of this thesis and discuss my results in the context of recent research areas. The thesis ends with an outlook about possible next steps of this work.

Chapter 2

Theory and Methods

2.1 Nonnegative Matrix Factorisation

NMF is one of the central algorithms used in the course of this thesis. In this section, I am going to provide a general introduction to this algorithm, starting from general concepts about matrix factorisation techniques and implementational issues, and afterwards I am going to present a detailed derivation of an efficient implementation of NMF.

2.1.1 Unsupervised Learning and Matrix Factorisation

Signal processing, data analysis and data mining are a pervasive topic throughout the manufacturing industry and engineering. Under the general topics of "Digitalisation", "Industry 4.0" and "Data Science", the extraction of interesting knowledge from raw datasets, measurements, observations and the understanding of complex data have become an important challenge and objective. In this thesis, I am going to focus on the fact that datasets generated by complex phenomena are usually the representation of the integrated result of several interrelated variables or a superposition of underlying latent components or factors. Thus, an important goal is to decompose these datasets, and separate them into components, to discover their structure and extract hidden information for further analysis.

Approximate low-rank matrix and tensor factorisations or decompositions are techniques that replace the original data by a lower dimensional approximate representation obtained via a matrix factorisation or decomposition [26]. In a multitude of applications, the signals or measurements are nonnegative quantities, or are sparse or smooth in nature. As it turns out, it is preferable to take these constraints into account to extract components or factors with physical meaning or reasonable interpretation and to avoid absurd results. In other

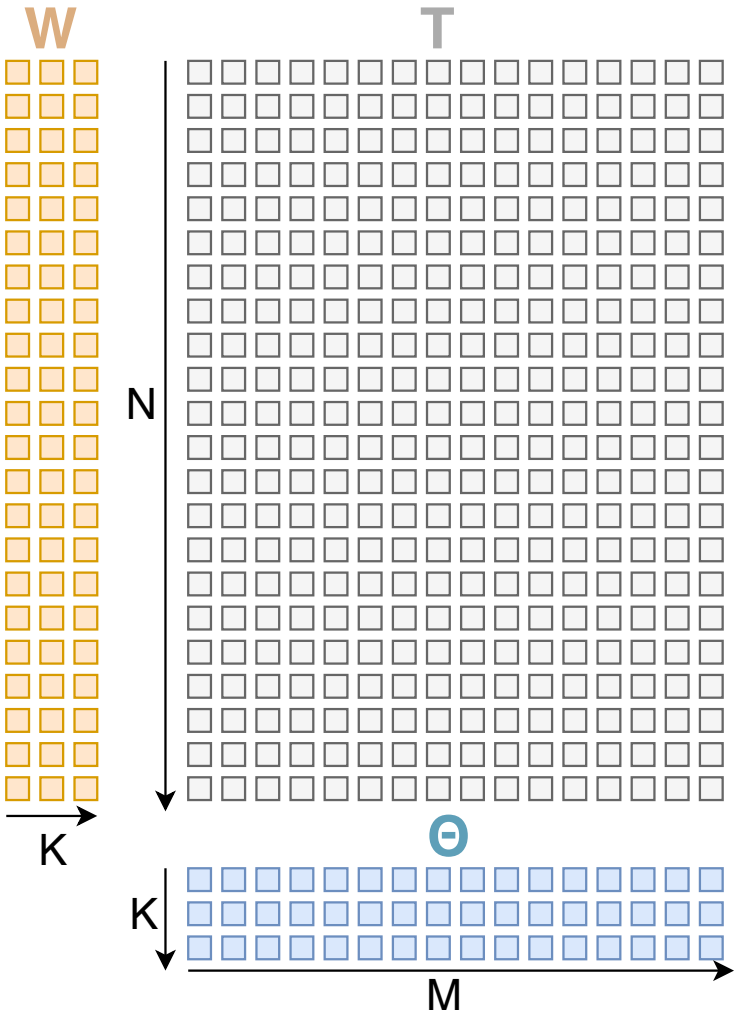

Fig. 2.1 Illustration of a linear dimension reduction via a factorisation model $\mathbf{T} \simeq (\mathbf{W}, \mathbf{\Theta})$. $\mathbf{W}$ contains the new coordinates of the input data in the reduced space and $\mathbf{\Theta}$ the latent components.

applications, the best sensor position or the actually important information might not be known beforehand, but has to be extracted from the gathered signals.

Nonnegative matrix factorisation is one of the main tools used for the work provided in the practical part of this thesis. The data collected during thermal manufacturing processes is characterised by unstable processes and changing environmental conditions, which makes any kind of data analysis or machine learning a challenging task. Approximate matrix factorisations offer effective tools to separate all these influencing factors and extract knowledge important for engineers to optimise the manufacturing process or to monitor the process. In the following sections, a general introduction to NMF is given and a detailed description of the main algorithms used in this thesis to solve the NMF problem is provided.

2.1.1.1 Low Rank Matrix Approximation

Suppose we have a dataset $\mathbf{T} \in \mathbb{R}^{N \times M}$, where N is the number of samples and M the number of features, i.e. the dimension of the dataset. Then a low rank matrix factorisation can be understood as any method that calculates an approximation of $\mathbf{T}$ with two factor matrices $\mathbf{W} \in \mathbb{R}^{N \times K}$ and $\mathbf{\Theta} \in \mathbb{R}^{K \times M}$:

$$\mathbf{T} \simeq \mathbf{W}\mathbf{\Theta} \tag{2.1}$$

Fig. 2.1 shows an illustration of the dimensions of the factor matrices and fig. 2.2 illustrates the decomposition in equation (2.1).

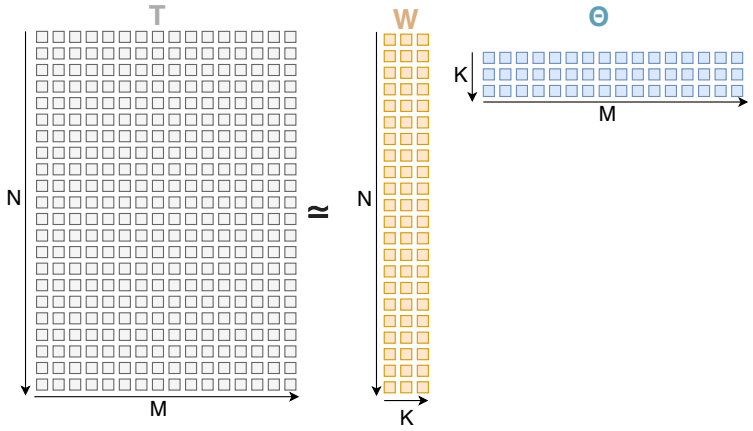

Fig. 2.2 Illustration of a linear dimension reduction via a factorisation model $\mathbf{T} \simeq (\mathbf{W}, \mathbf{\Theta})$. $\mathbf{W}$ contains the new coordinates of the input data in the reduced space and $\mathbf{\Theta}$ the latent components.

K is the number of dimensions in a new lower dimensional subspace of the original space. The general idea is that if such a low rank approximation is possible, then the most important

information about the dataset must be contained in this lower dimensional subspace. The columns of matrix **W** are the new coordinates in the lower dimensional subspace and can be used instead of the full feature vectors in **T**. The rows in **Θ** are the basis vectors of the subspace and should point into directions, which capture latent structures in the original cloud of high dimensional data points **T**. A low rank approximation of the form given in (2.1) can also be written as a sum of rank one matrices (see fig. 2.3). The terms in this expression are calculated as outer products of the columns in **W** and the rows in **Θ**. From this expression it becomes clear how the complexity or redundancy in the dataset **T** can be reduced by dropping terms on the right-hand side in fig. 2.3.

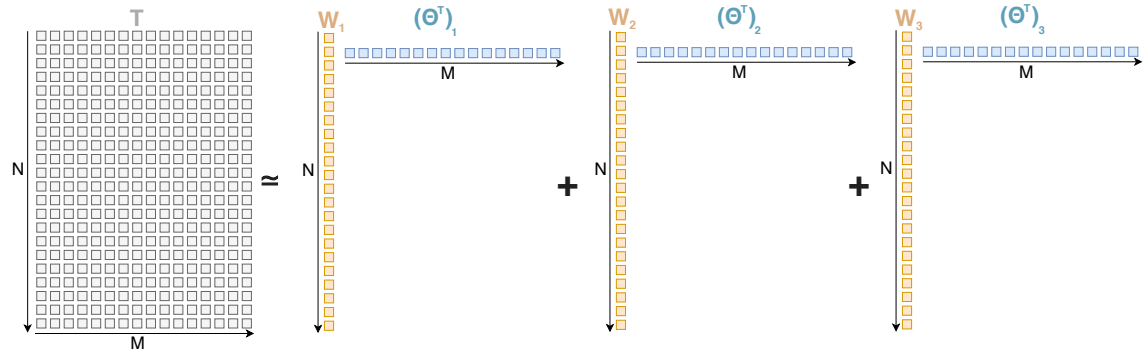

Fig. 2.3 Illustration of a rank one approximation via a factorisation model.

2.1.1.2 Singular Value Decomposition and Principal Component Analysis

One of the oldest matrix factorisation techniques is principal component analysis (PCA), which up to this day is still widely used in a variety of applications and can be considered one of the most popular linear dimensionality reduction techniques due to its simplicity [46]. PCA is a purely data-driven approach that constructs a low-dimensional representation of the data, which explains as much of the variance in the data as possible. In mathematical terms, PCA finds a new orthogonal basis of the data, which is oriented in such a way that the variance of the coordinates in the new basis is maximal. The usual scenario in which PCA is used, is when one can assume that there is a redundancy in the dataset under consideration. This redundancy should be reflected in terms of correlations, since the transformed data is uncorrelated after the application of PCA.

PCA is commonly performed with *Singular Value Decomposition* (SVD). Let **T** be our data matrix with dimension $N \times M$, where N is the number of samples and M is the number of variables. SVD is based on a theorem in linear algebra, which states that any real valued matrix **T** can be decomposed as

$$\mathbf{T} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{2.2}$$

where $\mathbf{U} \in \mathbb{R}^{N\times N}$, $\mathbf{\Sigma} \in \mathbb{R}^{N\times M}$ and $\mathbf{V}^T \in \mathbb{R}^{M\times M}$. In full matrix form, the SVD of $\mathbf{T}$ is as follows:

$$\mathbf{T} = \begin{pmatrix} u_{11} & & u_{N1} \\ & \ddots & \\ u_{1N} & & u_{NN} \end{pmatrix} \begin{pmatrix} \sigma_1 & & & 0 \\ & \ddots & & \\ & & \sigma_J & \\ & & & \ddots \\ 0 & & & 0 \end{pmatrix} \begin{pmatrix} v_{11} & & v_{M1} \\ & \ddots & \\ v_{1M} & & v_{MM} \end{pmatrix} \tag{2.3}$$

$\mathbf{\Sigma}$ is diagonal with only non-negative entries, which are the so called *singular values* $\mathbf{\Sigma}$. The usual convention is to order the columns of $\mathbf{\Sigma}$ from high to low with the highest singular value in the upper left of $\mathbf{\Sigma}$. Any matrix $\mathbf{T}$ will have as much singular values as its rank J. Both $\mathbf{U}$ and $\mathbf{V}^T$ are orthogonal, i.e.

$$\mathbf{U}\mathbf{U}^T = \mathbb{1}^{N\times N}, \quad \mathbf{V}\mathbf{V}^T = \mathbb{1}^{M\times M}. \tag{2.4}$$

This decomposition allows us to rewrite $\mathbf{T}\mathbf{T}^T$ as

$$\mathbf{T}\mathbf{T}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}\mathbf{V}^T\mathbf{\Sigma}\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T. \tag{2.5}$$

Here, we can see that $\mathbf{U}$ acts as an orthogonal basis transformation, which diagonalises $\mathbf{T}\mathbf{T}^T$, because $\mathbf{\Sigma}$ is diagonal and thus also $\mathbf{\Sigma}^2$. As $\mathbf{T}\mathbf{T}^T$ is proportional to the covariance matrix of the data matrix $\mathbf{T}$, $\mathbf{U}$ also diagonalises the covariance matrix, which results in uncorrelated variables with a transformation to the basis given by the columns of $\mathbf{U}$. From (2.2) it is possible to construct a low rank approximation of the data matrix $\mathbf{T}$ by first rewriting

$$\mathbf{T} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \widetilde{\mathbf{W}}\mathbf{V}^T, \tag{2.6}$$

with $\widetilde{\mathbf{W}} = \mathbf{U}\mathbf{\Sigma}$ and then just dropping the rows and columns except the ones with a high *singular values* σ_i:

$$\mathbf{T} \simeq \mathbf{W}\mathbf{\Theta}, \tag{2.7}$$

where $\mathbf{W}$ is made up of the first K columns of $\widetilde{\mathbf{W}} = \mathbf{U}\mathbf{\Sigma}$ and $\mathbf{\Theta}$ the frist K columns of $\mathbf{V}^T$. This low rank approximation is then called principal component analysis and the K rows in $\mathbf{\Theta}$ are called the first K principal components.

This low rank approximation comes with useful properties for real-world applications. First, the components are uncorrelated due to the derivation from the SVD. Second, they are naturally ordered by their variance. The first principal component always is the direction in

the dataset with maximum variance. Finally, it can be shown that PCA always yields the low rank approximation with the lowest overall mean squared error.
The SVD decomposition in (2.2) can also be written as a sum of outer products

$$\boldsymbol{T} = \sum_{j=1}^{J} \sigma_j \mathbf{u}_j \mathbf{v}_j^T = \sum_{j=1}^{J} \sigma_j \mathbf{C}^{(j)}, \tag{2.8}$$

where we have set $\mathbf{C}^{(j)} = \mathbf{u}_j \mathbf{v}_j^T$ and $\mathbf{u}_j$ and mfv_j are the column vectors of $\mathbf{U}$ and $\mathbf{V}$. From this expression we can see that by using the SVD-based low rank approximation, the terms in (2.8), that have a small contribution, are omitted and only the latent contributions are kept.

2.1.1.3 Moore-Penrose-Inverse

From the just obtained SVD decomposition we can derive a way to calculate the so-called pseudo-inverse of matrices of any dimension [32, 86]. A pseudo-inverse has some properties of a regular inverse matrix, but not necessarily all of them, and does not have to exist for for every arbitrary matrix. Let $\mathbf{A}$ be a matrix of size NxM and the SVD decomposition is as defined in the section before

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T. \tag{2.9}$$

A pseudo-inverse $\mathbf{A}^+$ of matrix $\mathbf{A}$ has to satisfy the condition $\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$. The pseudo-inverse can then be readily obtained by

$$\mathbf{A}^{-1} = \mathbf{V}\boldsymbol{\Sigma}^+\mathbf{U}^T, \tag{2.10}$$

where $\boldsymbol{\Sigma}^+$ is formed from $\boldsymbol{\Sigma}$ by taking the reciprocal of all the non-zero elements and then transposing the matrix.

2.1.2 The NMF Problem

As mentioned before, PCA can be described as a low rank approximation with the constraint that the data has to be projected onto orthogonal basis vectors that point in directions of maximum variance. Roughly speaking, all low rank approximation techniques mainly differ in the choice of constraints applied to the factor matrices. The constraint for NMF is to restrict the factor matrices to yield strictly additive decompositions. Let $\mathbf{T}$ be a data matrix with the same dimensions as before. $\mathbf{T}$ is now also strictly nonnegative, i.e. $T_{ij} \geq 0 \; \forall \; i, j$. We wish to find an approximation into two factor matrices like before, but with additional nonnegativity constraints:

$$\mathbf{T} \simeq \mathbf{W}\boldsymbol{\Theta}, \quad W_{ij} \geq 0, \Theta_{ij} \geq 0 \tag{2.11}$$

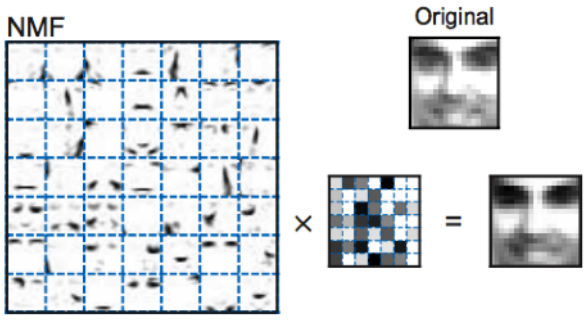

Fig. 2.4 Illustration of the parts-based reconstruction of faces with the components extracted with NMF. This reconstruction is just one example of the many applications, where NMF yields highly interpretable results. Illustration is taken from [74].

The dimensions of the factor matrices are $\mathbf{W} \in \mathbb{R}^{N \times K}$ and $\mathbf{\Theta} \in \mathbb{R}^{K \times M}$. K has to be a number with $K \leq min(N,M)$. The approximation in (2.11), together with the constraint, is called the NMF problem. As it turns out by enforcing the rather simple nonnegativity constraints, the NMF decomposition yields highly interpretable results in multiple application areas.

When tracking the history of NMF, we find many publications which mention the stated problem and give suggestions for solving it, but the first detailed introduction can be found in a publication of Paatero and Tapper in 1994 [85]. A few years later, the potential of NMF was widely recognised after a famous publication by Lee and Seung in 1999, where the algorithm they proposed was applied to images of human faces [74]. The astonishing result was that NMF is able to decompose the images into a parts-based representation with naturally emerging interpretable components. The images of human faces were decomposed into parts like eyes, nose and mouth, which corresponds to human visual perception. Fig. 2.4 shows the decomposition from their publication. Since then, NMF has been applied in many different areas:

- In image data processing, NMF decomposes images into parts-based representations, which can be used for face detection, handwritten digit recognition and general image classification problems [75, 13, 18, 70].
- In biomedical applications, NMF found application in microarray data analysis [19, 71, 39, 11].
- In text mining, NMF can extract topics of semantic features from collections of documents [97].
- In sound recognition and classification, NMF can be used to extract acoustic features, like instrument-specific patterns, to perform blind source separation [21, 101].

- NMF has an inherent clustering property and has found various applications in this domain [119, 97, 30].
- In spectral analysis, NMF can be used to decompose time-domain signals into different frequency components [89].
- NMF is a popular choice for blind source separation problems in multisensor systems [16].
- In industrial manufacturing applications, NMF is used to discover latent structure in process data [38, 120, 95].

The factorisation in (2.11) can be estimated with various optimisation strategies, but the general approach is the same for most algorithms. First, we have to define a goodness of fit measure for the approximation error in (2.11), which is going to be called $D(\mathbf{T}|\mathbf{W\Theta})$ in the following. Then, during an optimisation procedure, the two factor matrices are alternatingly going to be updated until a stopping criterion is reached. In the following, I intend to go through the general concept and the most common algorithms to solve the NMF problem. Afterwards, I am going to provide a detailed introduction to the algorithm which is used in the practical part of this thesis.

2.1.2.1 Why Nonnegativity?

In real-world applications many quantities are nonnegative by nature and hidden components have a physical meaning only when nonnegative. For a data-driven approach to yield components with a physical interpretation, nonnegativity is often desirable or necessary, for example in image processing and computer vision or spectral analysis. Many physical quantities are strictly nonnegative by definition, for example energy, intensities or frequency. Yet enforcing nonnegativity causes some of the explained variance of the approximate model to be lost, as there is a trade-off between interpretability and statistical fidelity. This problem is going to be be discussed later in this thesis by using the terms "overfitting" and "regularisation." Another advantage of nonnegativity is the fact that the decompositions are strictly additive, i.e. the extracted sources cannot cancel each other. This means that one source is either "present" or "not present" and the corresponding loadings in the factor matrix $\mathbf{W}$ measures the strength of the contribution.

The data analysed in this thesis are temperature time curves, which are also naturally nonnegative. Yet an additive decomposition of temperature-time curves does not have a straight forward physical interpretation because in general, it is not meaningful to sum up temperatures. In chapter 3, I am going to show the mathematical connection between an

additive decomposition of temperature curves and the underlying physical mechanisms of the corresponding thermal manufacturing process.

2.1.2.2 The Cost Function

The cost function quantifies the approximation error between $\mathbf{T}$ and its reconstruction with $\mathbf{W\Theta}$. A popular cost function is the squared Euclidean distance or quadratic error function [70, 76]:

$$D(\mathbf{T}|\mathbf{W\Theta}) = \|\mathbf{T} - \mathbf{W\Theta}\|_F^2 = \sum_{ij} \left((T_{ij} - [\mathbf{W\Theta}]_{ij}\right)^2. \tag{2.12}$$

The $\|\cdot\|_F$ is called the *Frobenius norm*. If we divide the Frobenius norm by the number of entries $M \cdot N$, it can be interpreted as a the mean reconstruction error of the reconstruction of $\mathbf{T}$ with $\mathbf{W\Theta}$. Another commonly used cost function is the generalised *Kullback-Leibler divergence* (KLD) [70]:

$$D(\mathbf{T}|\mathbf{W\Theta}) = \sum_{ij} \left(T_{ij} \log \frac{T_{ij}}{[\mathbf{W\Theta}]_{ij}} + [\mathbf{W\Theta}]_{ij} - T_{ij}\right) \tag{2.13}$$

The Kullback-Leibler divergence is originally a quantity used in probability theory to measure the difference between probability distributions. Algorithms based on the KLD can be interpreted as obtaining underlying probability distributions of latent factors. Apart from those two, there are many other possible choices for the cost function. Most of them are divergence measures like the KLD, for example Bregman divergences [28], α-divergences [24] and Itakura-Saito divergences [35]. The algorithm used in this thesis employs the Frobenius norm in (2.12).

2.1.2.3 Strategies for NMF

In this chapter, I am going to present the most commonly used algorithms for the NMF problem derived from the two cost functions (2.12) and (2.13) and address the question of initialisation and model order selection. The main difficulty in minimising the cost functions (2.12) and (2.13) lies in the fact that we have two matrices we do not know. $\mathbf{W}$ and $\mathbf{\Theta}$ must be found by an iterative procedure alternating between updating one matrix while keeping the other one fixed.
Lee and Seung introduced the following multiplicative update rules for the quadratic cost function (2.12)

$$\mathbf{\Theta}_{ij} \leftarrow \Theta_{kj} \frac{[\mathbf{W}^T\mathbf{T}]_{kj}}{[\mathbf{W}^T\mathbf{W\Theta}]_{ij}} \quad \text{and} \quad W_{ij} \leftarrow \mathbf{W}_{ik} \frac{[\mathbf{T\Theta}^T]_{ik}}{[\mathbf{W\Theta\Theta}^T]_{ik}} \tag{2.14}$$

and another set of update rules for the generalised KL-divergence D_{KL} (2.13)

$$\mathbf{\Theta}_{ij} \leftarrow \mathbf{\Theta}_{kj} \frac{\sum_i \mathbf{W}_{ik} T_{ij} / [\mathbf{W}\mathbf{\Theta}]_{ij}}{\sum_l \mathbf{W}_{lk}} \quad \text{and} \quad \mathbf{W}_{ij} \leftarrow \mathbf{W}_{ik} \frac{\sum_j \mathbf{\Theta}_{kj} T_{ij} / [\mathbf{W}\mathbf{\Theta}]_{ij}}{\sum_p \mathbf{\Theta}_{kp}}. \tag{2.15}$$

In [70], they gave mathematical proof that these multiplicative updates never increase their respective cost function.

How well an NMF algorithm converges depends on the specific algorithm and for most update rules, global convergence is hard to proof. Most NMF implementations test the convergence by measuring the decrease of the cost function between iterations and stop if the difference falls below some threshold value [51, 77, 63, 62]. Furthermore, the result of the decomposition depends on multiple factors one has to take into consideration. If the goal is to achieve the best nonnegative approximation, then other parameters are more important than if the goal is to extract interpretable decompositions. For a detailed comparison of the most common NMF algorithms see [128].
Overall, the following questions need to be addressed when solving the NMF problem:

- **Which cost function to choose?** See section 2.1.2.2.

- **How to initialise the matrices W and Θ?** The matrices $\mathbf{W}$ and $\mathbf{\Theta}$ are sometimes initialised with arbitrary nonnegative numbers, but there are various more sophisticated strategies for the initialisation. The goal is to find good starting point in the solution space, for example by doing k-means clustering or PCA [127, 2]. Since NMF algorithms are gradient based techniques, they suffer from getting stuck in local minima depending on the initialisation.

- **How to optimise the cost function?**
  The multiplicative update rules presented in this section are obviously not the only way to minimise the respective cost function. Several numerical strategies have been developed to solve the optimisation problem (some good survey papers are [105, 5, 22]). Multiplicative update rules and the *alternating least squares* (ALS) technique are the two most commonly used algorithms because they do not require additional parameters and thus are easy to implement [26, 69].
  One could also consider to place additional constraints on the matrices $\mathbf{W}$ and $\mathbf{\Theta}$. In sparse NMF, the matrix $\mathbf{\Theta}$ is assumed to be sparse, i.e. most of its entries are assumed to be zero [55]. It is also possible to relax the non-negativity constraints. This leads to NMF variants like semi-NMF and convex-NMF [30].

- **What is the best choice for K, i.e. the number of hidden components?**
  The model order K has to be chosen beforehand. As already mentioned, this is a difficult issue because there is no optimal decision criterion and in some cases, there does not even have to exist an ideal value for K. Information criteria like the "Bayes information criterion" (BIC) or the "Akaike information criterion" (AIC) can be useful to have a rough estimate for K [1, 96], but in practice it is often reasonable to try different values for K and decide by comparing the solutions. In some cases, there might be prior knowledge about the latent components or sources, which are present in the dataset. In such cases, the number of components can be chosen accordingly.

- **The stopping criterion.** NMF is estimated in an iterative optimisation procedure, so a simple stopping criterion would be a predefined number of iterations or a fixed running time. In literature, we can find different suggestions for stopping criteria. Brunet et. al suggest to measure the difference between recent iterations and set a threshold as a stopping criterion [11]. In the studies done in the course of this thesis, this stopping criterion was used. Other authors invoke stopping criteria from bound constrained optimisation [78]. The violation of the Karush-Kuhn-Tucker optimality conditions can also be used as a measure to construct a stopping criterion [63].

2.1.2.4 Initialisation Techniques

Without constraints, the NMF algorithm yields two indeterminancies. One is the non-uniqueness of the extracted components scaling. Each solution $\mathbf{W\Theta}$ can be transformed by multiplying with a matrix $\mathbf{B}$ and its inverse $\mathbf{WBB}^{-1}\mathbf{\Theta}$. The matrix $\mathbf{B}$ can be at least any non-negative monomial matrix, i.e. a permutation and scaling matrix. Secondly, the number of the extracted components is not determined automatically, but must be set to a fixed K beforehand. How to deal with the non-uniqueness remains an open question and there is no satisfactory solution yet for all cases [68, 53, 95]. Hence, different strategies exist to render an NMF solution unique. These strategies are closely related to the question of how to initialise an NMF decomposition. The initialisation has a large impact on the algorithm's performance and output. NMF algorithms are prone to get stuck in local minima due to the fact that the cost function is non-convex if one considers both arguments. This means that a random initialisation is almost never advisable, because the solutions will be different for each run and the number of iterations needed for a good fit will be larger. In practice it is recommended to run an algorithm several times using different random initialisations and pick the solution which offers the best approximation.

Canonically, a generally simple and effective way towards unique NMF solutions is to fix the initialisation of the factor matrices $\mathbf{\Theta}$ and $\mathbf{W}$. The nonnegative double singular value decomposition (NNDSVD) initialisation is an effective way to choose an initial set of components $\mathbf{\Theta}_{k*}$ [10]. This approach is based on the SVD, which has already been explained in section 2.1.1.2. The main idea behind using SVD to initialise NMF is the fact that it is mathematically provable that SVD yields the smallest approximation error (in (2.12)) compared to other matrix factorisation approaches.

In the practical part of the thesis, I am going to introduce an alternative to a canonical data-driven approach by incorporating prior knowledge and deliberately initialise the component temperature-time functions by physically motivated dependencies. This is done by connecting the heat equation with a matrix factorisation via the multivariate Taylor expansion. I am going to demonstrate that this way, we can achieve highly interpretable decompositions.

2.1.2.5 An SVD-based Initialisation

In this section, I am going to outline how the SVD is used to calculate an initial guess for the factor matrices $\mathbf{W}$ and $\mathbf{\Theta}$ in (2.11). A very detailed mathematical derivation can be found in the original publication by Boutsidis et. al [10]. The authors also show that this initialisation dramatically speeds up the convergence of NMF algorithms compared to plain random initialisations.

From section 2.1.1.2, we take the formulation of the SVD as a sum of J singular triplets:

$$\mathbf{T} = \sum_{j=1}^{J} \sigma_j \mathbf{u}_j \mathbf{v}_j^T = \sum_{j=1}^{J} \sigma_j \mathbf{C}^{(j)} \tag{2.16}$$

where we have set $\mathbf{C}^{(j)} = \mathbf{u}_j \mathbf{v}_j^T$. If we assume the matrix $\mathbf{T}$ to be nonnegative, the NNDSVD method uses a modification of this sum. We denote any vector $\mathbf{x}$ or matrix $\mathbf{X}$, which is projected into the positive quadrant, as $\mathbf{x}_+ \geq 0, \mathbf{X}_+ \geq 0$, where $\mathbf{x}_+, \mathbf{X}_+$ denotes a vector or matrix of the same size that has all negative components set to zero. In the same way, the projection into the negative quadrant is denoted $\mathbf{x}_- \geq 0, \mathbf{X}_- \geq 0$ and is defined by $\mathbf{X} = \mathbf{X}_+ - \mathbf{X}_-$, with a corresponding definition in case of a vector. Note that for both the negative and positive projection, we use a "greater than or equal" sign, because, if the original matrix $\mathbf{X}$ contains zero elements, the zeroes will be present in both projections. Using this definition we can write:

$$\sigma_j \mathbf{C}^{(j)} = \sigma_j (\mathbf{u}_{j+} - \mathbf{u}_{j-})(\mathbf{v}_{j+} - \mathbf{v}_{j-})^T, \tag{2.17}$$

$$= \sigma_j (\mathbf{u}_{j+}\mathbf{v}_{j+}^T + \mathbf{u}_{j-}\mathbf{v}_{j-}^T) - \sigma_j (\mathbf{u}_{j+}\mathbf{v}_{j-}^T + \mathbf{u}_{j-}\mathbf{v}_{j+}^T) \tag{2.18}$$

The positive and negative section can thus be written as:

$$\sigma_j \mathbf{C}_+^{(j)} = \sigma_j(\mathbf{u}_{j+}\mathbf{v}_{j+}^T + \mathbf{u}_{j-}\mathbf{v}_{j-}^T) \quad \text{and} \tag{2.19}$$

$$\sigma_j \mathbf{C}_-^{(j)} = \sigma_j(\mathbf{u}_{j+}\mathbf{v}_{j-}^T + \mathbf{u}_{j-}\mathbf{v}_{j+}^T) \tag{2.20}$$

We only take the positive section $\sigma_j \mathbf{C}_+^{(j)}$ and consider their SVD decomposition. It can be proven that they have $rank(\sigma_j \mathbf{C}_+^{(j)}) \leq 2$, hence that they possess at most only two non-zero singular values μ_{j+}, μ_{j-} and corresponding eigenvectors. The eigenvectors and eigenvalues can be readily obtained from (2.19). Let $\hat{\mathbf{u}}_{j\pm} = \frac{\mathbf{u}_{j\pm}}{\|\mathbf{u}_{j\pm}\|}$ and $\hat{\mathbf{v}}_{j\pm} = \frac{\mathbf{v}_{j\pm}}{\|\mathbf{v}_{j\pm}\|}$ be the normalised positive and negative sections of $\mathbf{u}$ and $\mathbf{v}$, then the SVD decomposition of $\sigma_j \mathbf{C}_+^{(j)}$ is given by:

$$\sigma_j \mathbf{C}_+^{(j)} = \mu_{j+}\hat{\mathbf{u}}_{j+}\hat{\mathbf{v}}_{j+}^T + \mu_{j-}\hat{\mathbf{u}}_{j-}\hat{\mathbf{v}}_{j-}^T, \tag{2.21}$$

$$\mu_{j+} = \left\|\mathbf{u}_{j+}\right\| \left\|\mathbf{v}_{j+}\right\| \sigma_j \quad \text{and} \tag{2.22}$$

$$\mu_{j-} = \left\|\mathbf{u}_{j-}\right\| \left\|\mathbf{v}_{j-}\right\| \sigma_j. \tag{2.23}$$

From there, we take the dominant singular triplet $(\mu_{j-}, \hat{\mathbf{u}}_{j-}, \hat{\mathbf{v}}_{j-}^T)$ or $(\mu_{j+}, \hat{\mathbf{u}}_{j+}, \hat{\mathbf{v}}_{j+}^T)$ to initialise the columns and rows of $\mathbf{W}$ and $\boldsymbol{\Theta}$:

$$\begin{aligned} \mathbf{W}_{:j} &= \sqrt{\mu_{j+}}\hat{\mathbf{u}}_{j+}, \\ \boldsymbol{\Theta}_{j:} &= \sqrt{\mu_{j+}}\hat{\mathbf{v}}_{j+}^T. \end{aligned} \tag{2.24}$$

If we wish to initialise an NMF run with $K \leq J$ components, we take the dominant $(\mu_{k\pm}, \mathbf{u}_{k\pm}, \mathbf{v}_{k\pm}^T)$ derived from the K leading $\sigma_k \mathbf{C}^{(k)}$.

In addition to the already mentioned performance improvements compared to random initialisation techniques, this initialisations also incorporates a natural order, similar to PCA, into the NMF decomposition. The initialisation is constructed by the dominant $(\mu_{k\pm}, \mathbf{u}_{k\pm}, \mathbf{v}_{k\pm}^T)$ and as such, the final NMF result is likely to keep this ordering, and the obtained components will be ordered according to their individual contributions in the reconstruction of $\mathbf{T}$. Another advantage is that NNDSVD is a purely data-driven approach as it is strictly mathematical and no prior knowledge about the data generation process is needed.

2.1.3 Alternating Least Squares and Hierarchical Alternating Least Squares

The algorithm used in the practical part of this thesis is based on the so called *Alternating Least Squares algorithm* (ALS) and is called *Hierachical Alternating Least Squares* (HALS)

[25]. HALS provides a fast and reliable solver for the NMF problem. Compared to the multiplicative update rules described in the last section, HALS is suitable for large scale datasets, which are commonly generated during industrial manufacturing. The first part of this section aims at explaining the standard ALS algorithm and modifications. As standard ALS still suffers from unstable convergence properties and the problem of suboptimal solutions, a detailed summary of the most common extensions is provided. Those extensions include additional constrains like sparsity or smoothness, which are implemented by adding suitable regularisation functions to the Frobenius norm cost function [125, 2, 52, 112]. After the derivation of the regularised ALS algorithm, I intend to go through the derivation of the the more efficient HALS algorithm, which was used for the practical applications in chapter three. The following sections are abstracted from the book of Cichocki et. al [26].

2.1.4 Alternating Least Squares Algorithm

ALS is the most used strategy for solving the NMF problem and is based on an alternating iteration of the factorisation matrices. The starting point is the known standard NMF model.

$$\mathbf{T} = \mathbf{W}\boldsymbol{\Theta}, \quad s.t.\ W_{ij} \geq 0 \text{ and } \Theta_{ij} \geq 0 \text{ with } \boldsymbol{\Theta} \in \mathbb{R}^{K \times M} \ \mathbf{W} \in \mathbb{R}^{N \times K} \tag{2.25}$$

As the name suggests, the cost function for this approach is the basic Euclidean distance function or Frobenius norm:

$$D(\mathbf{T} \parallel \mathbf{W}\Theta) = \frac{1}{2} \parallel \mathbf{T} - \mathbf{W}\Theta \parallel_F^2 \tag{2.26}$$

The ALS optimisation procedure now considers these cost functions as two separate optimisations or projections [23, 70].

$$W^{(k+1)} = argmin_W \parallel \mathbf{T} - \mathbf{W}\Theta^{(k)} \parallel^2 \quad s.t.\ \mathbf{W} \geq 0 \tag{2.27}$$

and

$$\Theta^{(k+1)} = argmin_\Theta \parallel \mathbf{T} - \mathbf{W}^{(k)}\Theta \parallel^2 \quad s.t.\ \Theta \geq 0 \tag{2.28}$$

This set of minimisation problems is now solved by exploiting a fixed point approach. For a solution in nonlinear programming to be optimal, it has to satisfy the already mentioned Karush-Kuhn-Tucker (KKT) optimality conditions [17]. The KKT conditions are an extension of Lagrange multipliers. From the KKT optimality condition it can be conducted that any stationary point $\boldsymbol{\Theta}^*$ and W^* of the cost functions (2.27) and (2.28) has to satisfy [49]:

$$\mathbf{W}^* \geq 0, \quad \boldsymbol{\Theta}^* \geq 0 \tag{2.29}$$

$$\nabla_W D(\mathbf{T} \parallel \mathbf{W}^*\mathbf{\Theta}^*) = \mathbf{W}^*\mathbf{\Theta}^*\mathbf{\Theta}^{*T} - \mathbf{T}\mathbf{\Theta}^{*T} \geq 0, \quad \mathbf{W} \odot \nabla_W D(\mathbf{T} \parallel \mathbf{W}^*\mathbf{\Theta}^*) = 0 \tag{2.30}$$

$$\nabla_\Theta D(\mathbf{T} \parallel \mathbf{W}^*\mathbf{\Theta}^*) = \mathbf{W}^{*T}\mathbf{W}^*\mathbf{\Theta}^* - \mathbf{W}^{*T}\mathbf{T} \geq 0, \quad \Theta \odot \nabla_\Theta D(\mathbf{T} \parallel \mathbf{W}^*\mathbf{\Theta}^*) = 0 \tag{2.31}$$

Here, $\odot$ is defined as component-wise multiplication.
Under the assumption of strictly positive entries, we estimate the stationary points by setting the gradients in (2.30) and (2.31) to zero:

$$\nabla_W D(\mathbf{T} \parallel \mathbf{W}^*\mathbf{\Theta}^*) = [\mathbf{W}^*\mathbf{\Theta}^*\mathbf{\Theta}^{*T} - \mathbf{T}\mathbf{\Theta}^{*T}]_+ = 0 \tag{2.32}$$

$$\nabla_\Theta D(\mathbf{T} \parallel \mathbf{W}^*\mathbf{\Theta}^*) = [\mathbf{W}^{*T}\mathbf{W}^*\mathbf{\Theta}^* - \mathbf{W}^{*T}\mathbf{T}]_+ = 0 \tag{2.33}$$

$[\cdot]_+$ is the projection onto the positive quadrant as defined in section 2.1.2.5. From there, we obtain the standard nonnegative ALS update equations:

$$W \longleftarrow [\mathbf{T}\mathbf{\Theta}^T(\Theta\Theta^T)^{-1}]_+ = [\mathbf{T}\mathbf{\Theta}^{-1}]_+ \tag{2.34}$$

$$\Theta \longleftarrow [(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{T}]_+ = [\mathbf{W}^{-1}\mathbf{T}]_+ \tag{2.35}$$

The pseudocode for standard ALS using these update rules is summarised in **Algorithm 1**. This basic algorithm is efficient and easy to implement, but suffers from a tendency to get stuck in local minima. In the following, some extensions for the standard ALS, aiming to avoid this problem, are going to be discussed in detail.

Algorithm 1: Standard ALS algorithm

```
1 function ALS (T, K);
  Input  : Nonnegative matrix T ∈ R_+^{N×M}: data matrix, K: rank of approximation
  Output : Nonnegative factorisation matrices W ∈ R_+^{N×K}
2 and Θ ∈ R_+^{K×M} such that the cost function (2.26) is minimised.;
3 begin
4     Initialise W and Θ;
5     repeat
6         Update W ⟵ [TΘ^{-1}]_+;
7         Update Θ ⟵ [W^{-1}T]_+;
8     until a stopping criterion is met;
9 end
```

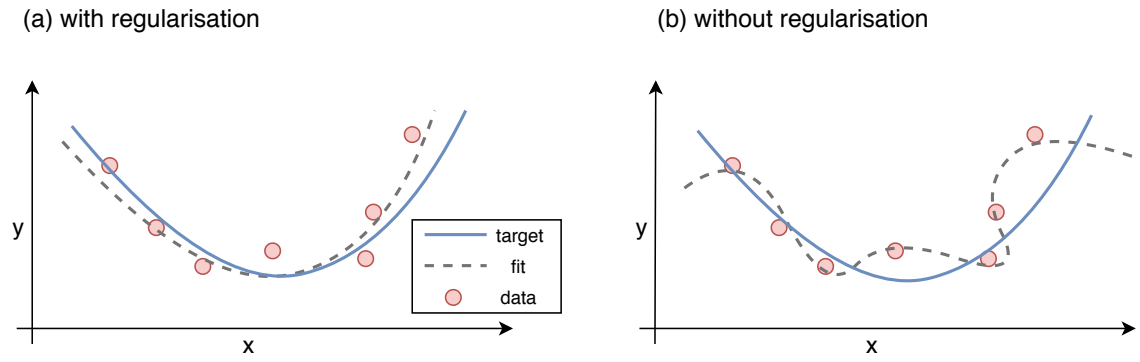

Fig. 2.5 The effect of regularisation on the obtained solution.

2.1.5 Extensions to ALS

2.1.5.1 L_1- and L_2-regularisations

Overfitting is a problem, which affects both supervised and unsupervised learning algorithms. A model which captures too much of the noise or outliers in the dataset, is said to be overfitting, because it does not learn general rules but instead random chance. Applying an overfitted model to new unseen data will show an increased error in its prediction or reconstructions. The chance of overfitting a model increases significantly if the number of dimensions in the dataset is larger or comparable to the number of samples available.

Regularisation is one strategy to avoid overfitting. Matrix factorisation techniques in particular are prone to overfitting, due to the naturally high complexity of the model (i.e. number of parameters). The number of parameters that have to be obtained from the optimisation in the standard NMF problem (see (2.25)) is

$$N \times K + K \times M, \tag{2.36}$$

which in most cases means more parameters than training points. Intuitively, the risk of overfitting should decrease if the number of training samples is increased, but in real-world applications, the number of samples is usually limited. Regularisation adds additional penalty terms to the cost function, which then favours solutions that do not contain extreme values or have a high local reconstruction error. This way, regularisation allows the training of complex models on a limited amount of data without severe overfitting.

The general cost function for NMF with regularisation using the Euclidean distance can be written in this form:

$$D_{Fr}(\mathbf{T} \parallel \mathbf{W\Theta}) = \frac{1}{2} \parallel \mathbf{T} - \mathbf{W\Theta} \parallel_F^2 + \gamma_W R_W(\mathbf{W}) + \gamma_\Theta R_\Theta(\mathbf{\Theta}) \;\; s.t.\, W_{ij} \geq 0,\, \Theta_{ij} \geq 0 \tag{2.37}$$

γ_W and γ_Θ are the respective regularisation parameters and the terms $R_W(\mathbf{W})$ and $R_\Theta(\mathbf{\Theta})$ are the regularisation penalties, which are chosen to enforce a certain desired property (smoothness, sparsity etc.). The regularisation parameters, which control the strength of the constraint, must either be chosen beforehand or calculated during the iteration steps. A common strategy is to choose the regularisation parameter γ with respect to the noise level $\mathbf{E} = \mathbf{T} - \hat{\mathbf{T}}$, which is unknown, but in some cases can be estimated beforehand. If the noise level is known, one can use the noise variance $\gamma = \sigma_E^2$. We can solve this optimisation problem in (2.37), the same as without regularisation, by setting the gradient to zero:

$$\frac{\partial}{\partial W_{ij}} D(\mathbf{T} \parallel \mathbf{W}^*\mathbf{\Theta}^*) = [\mathbf{W}^*\mathbf{\Theta}^*\mathbf{\Theta}^{*T} - \mathbf{T}\mathbf{\Theta}^{*T}]_{ij} + \gamma_W \frac{\partial R_W(\mathbf{W})}{\partial W_{ij}} = 0, \tag{2.38}$$

$$\frac{\partial}{\partial \Theta_{ij}} D(\mathbf{T} \parallel \mathbf{W}^*\mathbf{\Theta}^*) = [\mathbf{W}^{*T}\mathbf{W}^*\mathbf{\Theta}^* - \mathbf{W}^{*T}\mathbf{T}]_{ij} + \gamma_\Theta \frac{\partial R_\Theta(\mathbf{\Theta})}{\partial \Theta_{ij}} = 0, \tag{2.39}$$

Applying the half-rectifying projection, we derive the following update rules for the general case:

$$\mathbf{W} \leftarrow [(\mathbf{T}\mathbf{\Theta}^T - \gamma_W \frac{\partial R_W(\mathbf{W})}{\partial \mathbf{W}})(\mathbf{\Theta}\mathbf{\Theta}^T)^{-1}]_+ \quad \text{with} \quad \frac{\partial R_W(W)}{\partial \mathbf{W}} \in \mathbb{R}^{N\times K} \tag{2.40}$$

$$\mathbf{\Theta} \leftarrow [(\mathbf{W}^T\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{T} - \gamma_\Theta \frac{\partial R_\Theta(\mathbf{\Theta})}{\partial \mathbf{W}})]_+ \quad \text{with} \quad \frac{\partial R_\Theta(\mathbf{\Theta})}{\partial \mathbf{W}} \in \mathbb{R}^{K\times M} \tag{2.41}$$

In the following, the most used choices for the regularisation parameters $R_W(\mathbf{W})$ and $R_\Theta(\mathbf{\Theta})$ are shown and the way how they affect the final decomposition is discussed.

2.1.5.2 L_2-regularisation

If we want to impose smoothness constraints to the solution of the NMF optimisation, a standard approach is to use $R_W(\mathbf{W}) = \parallel \mathbf{W} \parallel_F^2$ and $R_\Theta(\mathbf{\Theta}) = \parallel \mathbf{\Theta} \parallel_F^2$. This choice of regularisation also bounds the obtained solution, because too large values will be suppressed during the optimisation. In a mathematical sense, smoothness in this context can be understood as small local variance in the obtained solution. Using the general derivation 2.40 and 2.41, we can now optimise the new cost function

$$D(\mathbf{T} \parallel \mathbf{W}\mathbf{\Theta}) = \frac{1}{2}(\parallel \mathbf{T} - \mathbf{W}\mathbf{\Theta} \parallel_F^2 + \gamma_W \parallel \mathbf{W} \parallel_F^2 + \gamma_\Theta \parallel \mathbf{\Theta} \parallel_F^2) \tag{2.42}$$

and obtain the following update rules:

$$\mathbf{W} \leftarrow [(\mathbf{T}\mathbf{\Theta}^T)^{-1}(\mathbf{\Theta}\mathbf{\Theta}^T + \gamma_W \mathbf{I})]_+ \quad \text{with } \mathbf{I} \in \mathbb{R}^{K\times K} \tag{2.43}$$

$$\mathbf{\Theta} \leftarrow [(\mathbf{W}^T\mathbf{W} + \gamma_\Theta \mathbf{I})(\mathbf{W}^T\mathbf{T})]_+ \quad \text{with } \mathbf{I} \in \mathbb{R}^{K\times K} \tag{2.44}$$

I is the identity matrix. This type of regularisation is called Tikhonov regularisation or L_2-regularisation.

2.1.5.3 L_1-regularisation

Another common regularisation is the so-called L_1-regularisation, where we use $R_{\mathbf{\Theta}}(\mathbf{\Theta}) = \| \mathbf{\Theta} \|_1$. Instead of the Frobenius norm, we use the l_1-norm, which is defined as $\| \mathbf{X} \| = \sum_{ij} |x_{ij}|$. The cost function with L_1-regularisation is:

$$D(\mathbf{T} \parallel \mathbf{W\Theta}) = \frac{1}{2}(\| \mathbf{T} - \mathbf{W\Theta} \|_F^2 + \gamma_W \| \mathbf{W} \|_1 + \gamma_\Theta \| \mathbf{\Theta} \|_1) \tag{2.45}$$

The ALS update rules now become:

$$\mathbf{W} \leftarrow [(\mathbf{T\Theta}^T - \gamma_W 1_{N\times K})(\mathbf{\Theta\Theta}^T)^{-1}]_+ \tag{2.46}$$

$$\mathbf{\Theta} \leftarrow [(\mathbf{W}^T\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{T} - \gamma_\Theta 1_{K\times M})]_+ \tag{2.47}$$

$1_{K\times M}$ and $1_{N\times K}$ are matrices where all entries are one.

L_1-regularisation is useful if a sparse nonnegative representation is a desired property of the solution. Sparse representations mostly contain zeros in the activation matrix and only a few non-zero elements. With the parameter γ_W and γ_Θ, the sparsity in either the activations or the components can be controlled. Sparsity is a desired property in many applications, because a sparse model definitely removes redundancy in the original dataset, if the new representation given by the model still has a high fidelity to the original data.

2.1.6 Hierarchical Alternating Least Squares

In the previous sections, the ALS algorithm was explained in detail. It is the foundation of the extension of the NMF implementation, which was used in the practical part of this thesis. The basic idea behind the *Hierachical Alternating Least Squares* (HALS) is that instead of one cost function, a set of local cost functions is used, which are sequentially minimised. This implementation of NMF yields advantages over other implementations, as it is shown to work well also for undercomplete cases (i.e. a system with less sensors than signals). The extensive practical applications of this algorithm show the validity and high performance of the HALS [26].

The first step in the derivation of the HALS update rules is to construct the set of cost functions from the standard cost function. In order to do so, we look at the rows and columns

of the factorisation matrices

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K], \tag{2.48}$$

with $\mathbf{w}_k$ being the column vectors of $\mathbf{W}$ and

$$\mathbf{\Theta}^T = \mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K], \tag{2.49}$$

with the $\mathbf{h}_k$'s being the row vectors of $\mathbf{\Theta}$ or the columns of $\mathbf{H}$. The already known Frobenius norm cost function can be rewritten like this:

$$D(T \parallel \mathbf{W\Theta}) = \parallel \mathbf{T} - \sum_{k=1}^{K} \mathbf{w}_k \mathbf{h}_k^T \parallel_F^2 = \parallel \mathbf{E} \parallel_F^2 \tag{2.50}$$

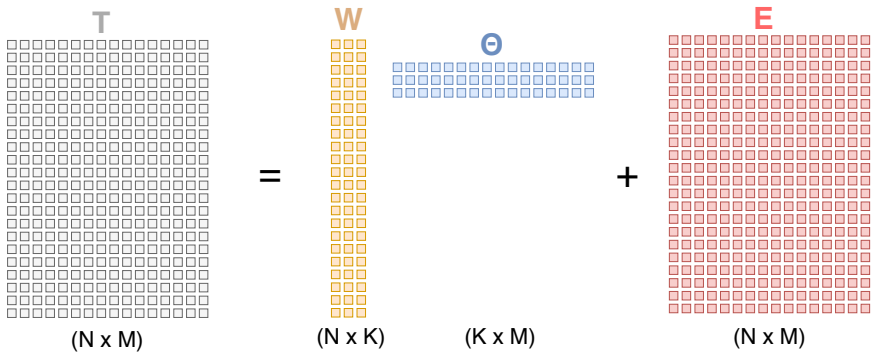

Fig. 2.6 Illustration of the definition of the error matrix. The error matrix $\mathbf{E}$ is the difference between original data and the current factorisation matrices.

Here, the factorisation is expressed as a sum over outer products between $\mathbf{w}_k$ and $\mathbf{h}_k$, which is a sum of rank 1 matrices. $\mathbf{E}$ is the error between the factorisation and original data matrix $\mathbf{T}$ (see fig. 2.6). With this representation, we define a set of residua

$$\mathbf{T}^{(j)} = \mathbf{T} - \sum_{k \neq j} \mathbf{w}_k \mathbf{h}_k^T = \mathbf{T} - \mathbf{W}\mathbf{H}^T + \mathbf{w}_j \mathbf{h}_j^T = \mathbf{E} + \mathbf{w}_j \mathbf{h}_j^T \tag{2.51}$$

and minimise the set of cost functions

$$D_{\mathbf{W}}^{(j)}(\mathbf{w}_j) = \frac{1}{2} \parallel \mathbf{T}^{(j)} - \mathbf{w}_j \mathbf{h}_j^T \parallel_F^2, \text{ for fixed } \mathbf{h}_j \tag{2.52}$$

$$D_{\mathbf{H}}^{(j)}(\mathbf{h}_j) = \frac{1}{2} \parallel \mathbf{T}^{(j)} - \mathbf{w}_j \mathbf{h}_j^T \parallel_F^2, \text{ for fixed } \mathbf{w}_j. \tag{2.53}$$

The definition of the residua matrices $\mathbf{T}^{(j)}$ is illustrated in fig. 2.7:

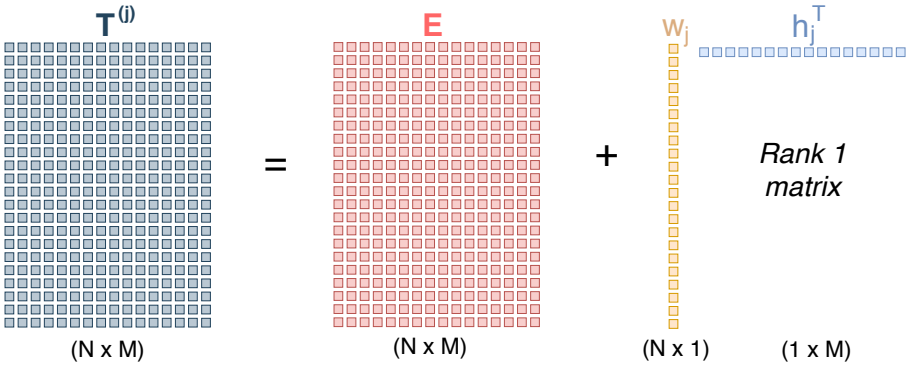

Fig. 2.7 Illustration of the definition of the residuum matrices. The residua matrices are calculated by adding a rank 1 matrix to the current error matrix.

The KKT optimality conditions for the set of cost functions

$$D_F^{(j)}(\mathbf{T}^{(j)} \parallel \mathbf{w}_j\mathbf{h}_j^T) = \frac{1}{2} \parallel \mathbf{T}^{(j)} - \mathbf{w}_j\mathbf{h}_j^T \parallel_F^2 \tag{2.54}$$

for $j = 1,\ldots,K$ subject to $\mathbf{w}_j \geq 0$ and $\mathbf{h}_j \geq 0$ are

$$\mathbf{w}_j \geq 0, \quad \mathbf{h}_j \geq 0, \tag{2.55}$$

$$\nabla_{\mathbf{w}_j} D_F^{(j)}(\mathbf{T}^{(j)} \parallel \mathbf{w}_j\mathbf{h}_j^T) \geq 0, \quad \nabla_{\mathbf{h}_j} D_F^{(j)}(\mathbf{T}^{(j)} \parallel \mathbf{w}_j\mathbf{h}_j^T) \geq 0, \tag{2.56}$$

$$\mathbf{w}_j \odot \nabla_{\mathbf{w}_j} D_F^{(j)}(\mathbf{T}^{(j)} \parallel \mathbf{w}_j\mathbf{h}_j^T)0, \quad \mathbf{h}_j \odot \nabla_{\mathbf{w}_j} D_F^{(j)}(\mathbf{T}^{(j)} \parallel \mathbf{w}_j\mathbf{h}_j^T) = 0. \tag{2.57}$$

The update rules are now derived as before by calculating the gradient with respect to $\mathbf{w}_j$ and $\mathbf{h}_j$:

$$\bigtriangledown_{\mathbf{w}_j} D_F^{(j)}(\mathbf{T}^{(j)} \parallel \mathbf{w}_j\mathbf{h}_j^T) = \frac{\partial D_F^{(j)}(\mathbf{T}^{(j)} \parallel \mathbf{w}_j\mathbf{h}_j^T)}{\partial \mathbf{w}_j} = \mathbf{w}_j\mathbf{h}_j^T h_j - \mathbf{T}^{(j)}\mathbf{h}_j, \tag{2.58}$$

$$\bigtriangledown_{\mathbf{h}_j} D_F^{(j)}(\mathbf{T}^{(j)} \parallel \mathbf{w}_j\mathbf{h}_j^T) = \frac{\partial D_F^{(j)}(\mathbf{T}^{(j)} \parallel \mathbf{w}_j\mathbf{h}_j^T)}{\partial \mathbf{h}_j} = \mathbf{w}_j^T\mathbf{w}_j^T\mathbf{h}_j - \mathbf{T}^{(j)T}\mathbf{w}_j. \tag{2.59}$$

Setting these gradients to zero and assuming strictly positive entries $\mathbf{w}_j$ and $\mathbf{h}_j$ for all j we obtain the HALS update rules without regularisation, which are also illustrated in fig. 2.8.

$$\mathbf{w}_j \leftarrow \frac{1}{\mathbf{h}_j^T\mathbf{h}_j}[\mathbf{T}^{(j)}\mathbf{h}_j]_+ = \frac{1}{\mathbf{h}_j^T\mathbf{h}_j}[\mathbf{T}^{(j)}\mathbf{h}_j]_+, \tag{2.60}$$

$$\mathbf{h}_j \leftarrow \frac{1}{\mathbf{w}_j^T\mathbf{w}_j}[\mathbf{T}^{(j)T}\mathbf{w}_j]_+ = \frac{1}{\mathbf{w}_j^T\mathbf{w}_j}[\mathbf{T}^{(j)T}\mathbf{w}_j]_+. \tag{2.61}$$

$[\,\cdot\,]_+$ is defined the same way as the positive section in section 2.1.2.5. In practical implementations of the HALS we do not set any emerging negative entries to zero but to a small positive constant $\varepsilon > 0$.

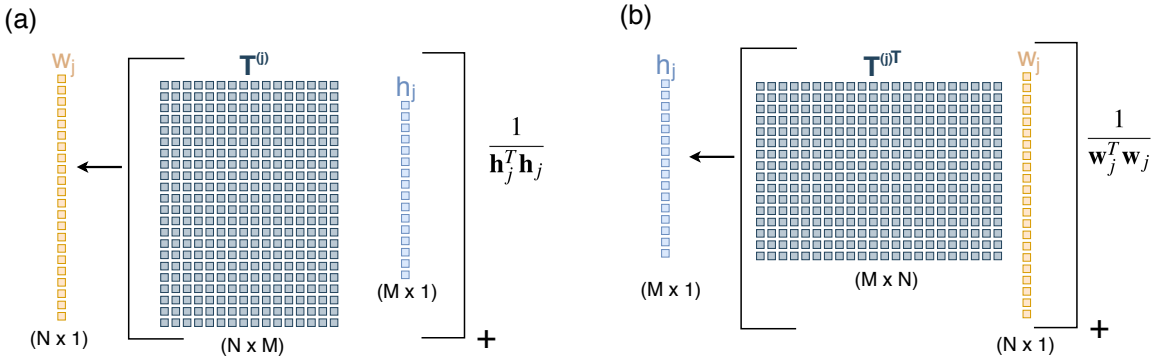

Fig. 2.8 (a) The update rule for $\mathbf{w}_j$. The following normalization step for $\mathbf{w}_j$ is not depicted here. (b) The update rule for $\mathbf{h}_j$.

In algorithm 2, the complete HALS procedure in its standard way is shown. Note that also a normalisation at each iteration step is performed on $\mathbf{w}_j$, which scales the column vectors to norm one (l_2-norm). The key difference between HALS and regular ALS is the consecutive optimisation of individual column vectors instead of the full matrices, which makes HALS more efficient and flexible than ALS. The iterative update of individual column vectors gives the opportunity to apply constrains only to specific components. Another possibility would be to keep certain components or coefficients constant during the training procedure. In ALS, such variations are not possible because in each iteration step, a full factor matrix is updated.

Algorithm 2: HALS algorithm

function HALS $(\mathbf{T}, K)$;
Input : Nonnegative matrix $\mathbf{T} \in \mathbb{R}_+^{N\times M}$: data matrix, K: rank of approximation
Output : Nonnegative factorisation matrices $\mathbf{W} \in \mathbb{R}_+^{N\times K}$ and $\mathbf{\Theta} \in \mathbb{R}_+^{K\times M}$ such that the cost function (2.26) is minimised.;
begin
  Initialise $\mathbf{W}$ and $\Theta = \mathbf{H}^T$;
  foreach $\mathbf{w}_j$ *in* $\mathbf{W}$ **do**
    $\mathbf{w}_j \leftarrow \mathbf{w}_j / \parallel \mathbf{w}_j \parallel_2$
  end
  $\mathbf{E} = \mathbf{T} - \mathbf{W}\mathbf{H}^T$;
  repeat
    for $j = 1, \ldots, K$ **do**
      $T^{(k)} \longleftarrow \mathbf{E} + \mathbf{w}_j \mathbf{h}_j^T$;
      $\mathbf{h}_j \leftarrow \frac{1}{\mathbf{w}_j^T \mathbf{w}_j} [\mathbf{T}^{(j)T} \mathbf{w}_j]_+$;
      $\mathbf{w}_j \leftarrow \frac{1}{\mathbf{h}_j^T \mathbf{h}_j} [\mathbf{T}^{(j)} \mathbf{h}_j]_+$;
      $\mathbf{w}_j \leftarrow \mathbf{w}_j / \parallel \mathbf{w}_j \parallel_2$;
      $\mathbf{E} \longleftarrow \mathbf{T}^{(k)} - \mathbf{w}_j \mathbf{h}_j^T$;
    end
  until *a stopping criterion is met*;
end

2.1.7 HALS with Regularisation

The set of cost functions for the HALS algorithm can also be extended with regularisation terms. Incorporating L_1- and L_2-regularisation for the matrix $\mathbf{\Theta}$ and $\mathbf{W}$ leads to the following set of cost functions:

$$D_F^{(j)}(\mathbf{T}^{(j)} \parallel \mathbf{w}_j\mathbf{h}_j^T) = \frac{1}{2} \parallel \mathbf{T}^{(j)} - \mathbf{w}_j\mathbf{h}_j^T \parallel_F^2 + \tag{2.62}$$

$$\gamma_{1,h} \parallel \mathbf{h}_j \parallel_1 + \frac{1}{2}\gamma_{2,h} \parallel \mathbf{h}_j \parallel_2^2 + \tag{2.63}$$

$$\gamma_{1,w} \parallel \mathbf{w}_j \parallel_1 + \frac{1}{2}\gamma_{2,w} \parallel \mathbf{w}_j \parallel_2^2 . \tag{2.64}$$

With the same calculations as in the previous sections, we can derive the following update rule for $\mathbf{h}_j$ (see equation (2.41)) [26]:

$$\mathbf{h}_j \leftarrow \left[\frac{1}{1+\gamma_{2,h}} \mathbf{I}([\mathbf{T}^{(j)}]^T \mathbf{w}_j - \gamma_{1,h}\mathbf{1}_{K\times 1}) \frac{1}{\mathbf{w}_j^T\mathbf{w}_j} \right]_+ \tag{2.65}$$

If we consider, that the cost function is symmetric then the update rule for $\mathbf{w}_j$ is

$$\mathbf{w}_j \leftarrow \left[\frac{1}{1+\gamma_{2,w}} \mathbf{I}(\mathbf{T}^{(j)}\mathbf{h}_j - \gamma_{2,h}\mathbf{1}_{K\times 1}) \frac{1}{\mathbf{h}_j^T\mathbf{h}_j} \right]_+ \tag{2.66}$$

With these update rules, the HALS procedure offers a flexible algorithm, which allows to apply regularisation constraints of any type to individual components or weights of the decomposition. In chapter 3, I am going to present a study about the effect of different combinations of regularisations on the decomposition results obtained from industrial datasets.

2.2 Linear Regression

So far in this thesis, I have mainly focused on unsupervised learning, which is also going to be in the center of the results of the practical work. In this short section, I am going to introduce one algorithm from the domain of supervised learning called linear regression. This technique is used in conjunction with NMF to design a so-called "virtual sensor," which is going to be presented in the results discussed in chapter 3 of this thesis.

The goal of regression is to predict the value of one or more continuous target variables y given the value of a K-dimensional vector of features $\mathbf{x}$. The following section is abstracted from [9]. Suppose we have multiple observations or samples $\mathbf{x_i}$, where $i = 1,\ldots,M$, and

a corresponding set of target values y_i. Our goal is to predict the value of the target y_{new} for new values of the input vector $\mathbf{x}_{new}$. In more general terms, we intend to find the probability distribution $p(t|\mathbf{x})$. If we have this predictive distribution, we have a measure for the uncertainty of the value t for each value of $\mathbf{x}$. As the name suggests, in linear regression, the goal is to estimate a linear mapping of y onto $\mathbf{x}$. Mathematically, we have to obtain a coefficient vector $\mathbf{b}$ and a bias term b_0, with which the following equation closely approximates the output $\mathbf{y}$:

$$y_n \simeq \mathbf{b}^T \mathbf{x}_n + b_0; \tag{2.67}$$

Just as it is the case in unsupervised learning, we need to define a suitably chosen cost function and an optimisation scheme. For linear regression, the Euclidean distance can be used as a cost function:

$$E_D(\mathbf{b}) = \frac{1}{2} \sum_{n=1}^{N} \{y_n - \mathbf{b}^T \mathbf{x}_n - b_0\}^2, \tag{2.68}$$

2.2.1 Linear Regression with NMF Preprocessing

Suppose we have a data set of multiple time series described by a data matrix $\mathbf{T}$, whose rows contain time series. $\mathbf{T_n}$ is a row vector of the n-th time series in $\mathbf{T}$. We now want to extend the standard linear regression to handle time series input. From a strictly mathematical viewpoint, there is no problem in using the full data matrix $\mathbf{T}$ as feature matrix and treat every time point $T_n(t)$ as an input feature for the regression model, but this approach has practical limitations. First, this will in most cases unnecessarily increase the input dimension compared to the information content of the whole time series. If only certain aspects of the time curve contain predictive information for the target variable y, the model complexity will be too large for the problem at hand. In such cases, we risk to overfit the model during the training process, i.e. the model maps noise features to the target y and will then perform poorly on new data. This fact is even more problematic if the set of target variables y is limited, which it usually is if y stems from real-world measurements. The standard approach is then to derive predefined features from the time series and use those as independent features for the regression model. This approach is advisable if combined with domain knowledge. In the practical part of this thesis, we deal with a dataset, where there is no domain knowledge about the most important features. So instead of predefining features, we use a feature learning algorithm like NMF. As explained before, NMF decomposes the data into latent factors that model the most dominant variations in the data. This way, the dominant information within the data matrix $\mathbf{T}$ should be captured while the input dimension is decreased. Furthermore, most of the random noise should be filtered after transforming the data into the NMF feature space. So the first step is

to train an NMF model on a data matrix $\mathbf{T}$ of sufficient size and keep the factorisation matrix $\boldsymbol{\Theta}$ (see fig. 2.9). $\boldsymbol{\Theta}$ can then be used to project new data onto the NMF components.

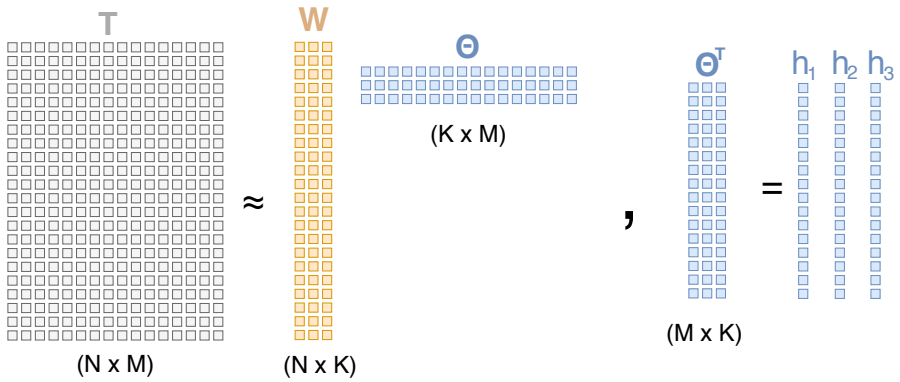

Fig. 2.9 An NMF model is trained and the fixed $\boldsymbol{\Theta}$ can be used to project new unseen data onto the components .

With NMF preprocessing, the cost function without the bias term for the linear regression model becomes

$$E_D(\mathbf{b}) = \frac{1}{2}\sum_{n=1}^{N}\{y_n - \mathbf{b}^T\hat{\boldsymbol{\Theta}}(\mathbf{T_n})\}^2, \tag{2.69}$$

where the $\hat{\boldsymbol{\Theta}}(\cdot)$ are transform functions which take the time series as input and return the corresponding value of the loadings estimated by the NMF model. Suppose $\mathbf{T_n}$ is a new, still unseen time curve:

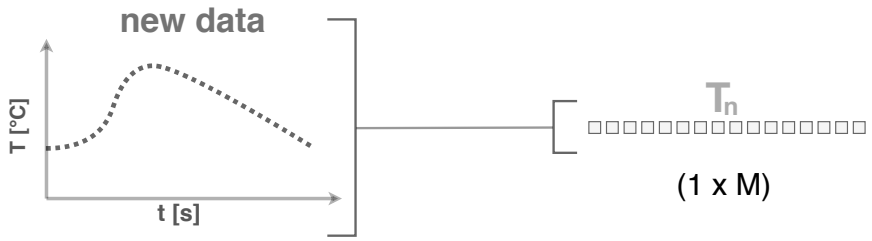

Fig. 2.10 Unseen data is represented as a new data vector.

Then

$$\hat{\boldsymbol{\Theta}}(\mathbf{T_n}) = (\hat{\Theta}_1(\mathbf{T_n}), \hat{\Theta}_2(\mathbf{T_n}), \ldots, \hat{\Theta}_K(\mathbf{T_n}))^T \tag{2.70}$$

is the projection of the time series vector $\mathbf{T_n}$ onto the component processes from the NMF model (with K components). $\hat{\boldsymbol{\Theta}}(\mathbf{T_j})$ can be described as repeatedly executing the optimisation procedure in (2.60) using the fixed $\boldsymbol{\Theta}$ until a stopping criterion is reached. The update

procedure is then executed only one-sided to estimate new weights $\mathbf{w}_j$. First the residua matrices are calculated using the fixed column vectors $\mathbf{h}_j$:

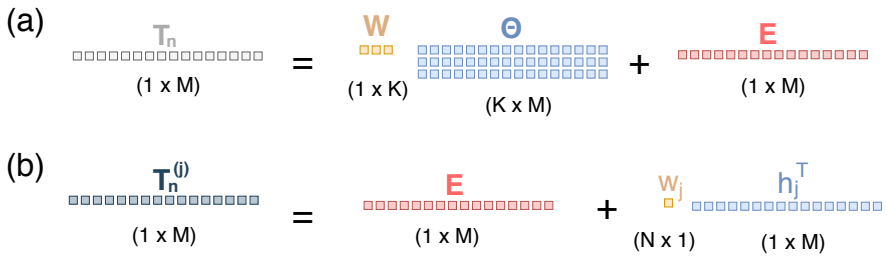

Fig. 2.11 (a) The approximation error. (b) The residuum matrix for a single value.

Afterwards, the update step for $\mathbf{w}_j$ is performed and repeated together with the estimation of the residuum matrices until a convergence criterion is reached.

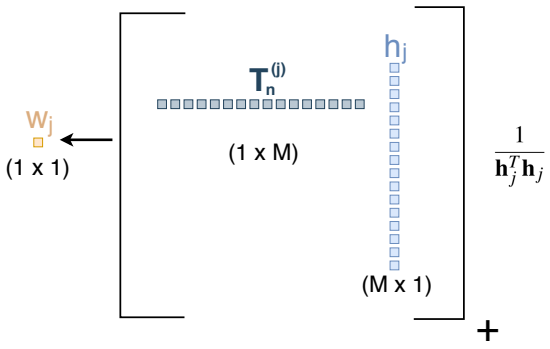

Fig. 2.12 The update step for a single value.

Similar to NMF, a regression model can be extended with regularisation terms .Using a regularisation term, the cost function becomes

$$E_D(\mathbf{b}) = \frac{1}{2}\sum_{n=1}^{N}\{y_n - \mathbf{b}^T\hat{\mathbf{\Theta}}(\mathbf{T_n})\}^2 + \frac{\lambda}{2}\sum_{j=1}^{K}|b_j|^q, \tag{2.71}$$

where q defines the type of regularisation and λ the strength of the regularisation. $q = 1$ (L_1-regularisation) is called lasso regression [109]. If we set $q = 2$ (L_2-regularisation), we perform ridge regression. Both regularisation types can be mixed and the parameter λ can be adjusted to the specific problem at hand. Lasso regression has the property that the coefficients in $\mathbf{b}^T$ are sparse, i.e. the b_j are driven to zero.

As mentioned before, regularisation allows to train the model with training datasets of limited size without the risk of severe over-fitting because the model complexity is limited

(the solution space for the coefficients is reduced), but the optimal value for λ has to be determined during the training phase. Solving (2.71), we first set the gradient with respect to $\mathbf{b}$ equal zero

$$\nabla_{\mathbf{b}} E_D(\mathbf{b}) = \sum_{n=1}^{N} \{y_n - \mathbf{b}^T \hat{\mathbf{\Theta}}(\mathbf{T_n})\} \hat{\mathbf{\Theta}}(\mathbf{T_n})^T = 0. \tag{2.72}$$

This leads to a simple algebraic solution for the least squares problem in (2.68):

$$\mathbf{b} = (\xi^T \xi)^{-1} \xi^T \mathbf{y}, \tag{2.73}$$

with $(\xi^T \xi)^{-1} \xi^T$ beeing the Moore-Penrose pseudo-inverse. The elemensts of $\xi \in \mathbb{R}^{M \times K}$ are defined as

$$\xi_{ij} = \hat{\Theta}_j(\mathbf{T_i}), \tag{2.74}$$

i.e. the projection of the i-th time curve onto the j-th NMF component. In summary, our combined NMF regression model includes the following training steps:

- step 1: Train an NMF model with K components on a training dataset $\mathbf{T}$;
- step 2: Use the projection onto NMF components as preprocessing function $\hat{\mathbf{\Theta}}(\cdot)$;
- step 3: Estimate the regression coefficients $\mathbf{b}$ by solving the least squares problem in (2.68).

The regression equation together with the $\hat{\mathbf{\Theta}}$ are then combined to construct the final model, which can be applied to any new time curve $\mathbf{T}_{new}$. From a pretrained NMF model, we construct the transformation functions $\hat{\mathbf{\Theta}}(\cdot)$. Then the regression equation is computed via linear least squares minimisation and the final model is then given by the combination of regression model and NMF model, which can be applied to any unseen time curve.
Usually, a regression model also contains a bias parameter b_0 that compensates for the difference between the averages (over the training set) of the target values and the weighted sum of the averages of the independent variables. The cost function with a bias parameter is

$$E_D(\mathbf{b}) = \frac{1}{2} \sum_{n=1}^{N} \{y_n - b_0 - \mathbf{b}^T \hat{\mathbf{\Theta}}(\mathbf{T_n})\}^2. \tag{2.75}$$

By setting the gradient with respect to the bias parameter to zero, we obtain

$$b_0 = \bar{y} - \sum_{j=1}^{K} b_j \bar{\hat{\Theta}}_j, \tag{2.76}$$

where

$$\bar{y} = \frac{1}{N} \sum_{k=1}^{N} y_k, \tag{2.77}$$

$$\bar{\hat{\Theta}}_j = \frac{1}{N} \sum_{k=1}^{N} \hat{\Theta}_j(\mathbf{T_n}) \tag{2.78}$$

2.2.2 Semi-supervised Learning

The combination of NMF, which is an unsupervised learning algorithm, and linear regularised regression, which is a supervised learning technique, can be seen as a so-called semi-supervised learning procedure. Semi-supervised learning in general is a class of machine learning tasks and techniques that make use of unlabelled data in a supervised learning setting [129]. Typically, the amount of labelled data is small compared to the amount of unlabelled data in this setting. In our case, we can exploit the fact, that we have a large unlabelled dataset for the training of the NMF model. The NMF model can be trained on a matrix $\mathbf{T}_{unlabelled} \in \mathbb{R}^{N_1 \times M}$, while the regression with the transformation function $\hat{\mathbf{\Theta}}(\mathbf{T_n})$ can be trained on a matrix $\mathbf{T}_{labelled} \in \mathbb{R}^{N_2 \times M}$, where $N_1 > N_2$. This way, the latent components, which are only identifiable from a huge amount of time series, can be used as input features for a supervised learning task, although only a few samples of labelled data are available.

In literature, many researchers report that the use of unlabelled data together with a small amount of labelled data can considerably improve the performance of regression or classification models (see the survey paper by Zhu for an overview [129]). In real-world applications, this scenario is not uncommon, for example if the acquisition of labelled data requires a human agent with expert knowledge (e. g. evaluating the content of images or classifying the results of experiments) or if the costs for a fully labelled dataset are too large (e.g. performing costly experiments). At the same time, the acquisition of unlabelled data might be relatively unproblematic. In an industrial manufacturing process, sensory data is generated en masse, but complicated measurements evaluating the machine status are costly and can usually only be performed while interrupting the series production. In such a scenario, a semi-supervised approach might be advisable.

Semi-supervised learning is often guaranteed to improve the model performance, if the underlying data generation process can be described as a mixture model [14, 15, 92, 36, 83, 3]. In such a scenario, the observed data $\mathbf{x}$ can be modelled as a combination of underlying latent components, i.e. probability distributions. The idea is to estimate a probabilistic model that generates the observed data. By sampling from this model, we are able to generate new data.

A mixture model with K components and given assumptions M takes the general form:

$$p(\mathbf{x}|M) = \sum_{k=1}^{K} p\left(\mathbf{x}|M_k, k\right) p\left(k|M_0\right) \tag{2.79}$$

On this basis the data can be generated by stochastically choosing one of the components under $p\left(k|M_0\right)$ and then drawing from $p\left(x|M_k, kk\right)$, where M_k are the component model assumptions and M_0 are assumptions about the mixture process. The model assumptions represent the model definition, i.e. parameters, model structure and prior information. $p(\mathbf{x}|M)$ quantifies the likelihood of the observed data if it is generated with the model assumptions M. The mixture model components can be identified if we have a large enough amount of data. If the mixture model assumption about the data generation is correct, the model performance can be improved, but if not, it is possible to effect the performance negatively, which was shown by Cozman et al. in [34]. This is why it is important to carefully construct the mixture model to reflect reality or to yield a sufficient approximation thereof.

Blind Source Separation techniques based on matrix factorisations (ICA, NMF) are a class of machine learning techniques that can be interpreted as techniques to extract underlying mixture models with certain mathematical properties. Nonnegative matrix factorisation in particular has a strong connection to well-known methods based on mixture model assumptions like probabilistic latent semantic analysis (PLSA) [41, 30]. According to [41], NMF can be interpreted in a similar manner like PLSA factors, which are probabilities. It has to be noted that most of the cited publications refer to semi-supervised classification tasks instead of a regression task. In general, the machine learning literature mainly focuses on semi-supervised classification tasks in this domain. Some algorithms naturally extend from classification to regression, so most of the empirical findings are similar and the theoretical background is transferable. The results reported in chapter 3 in this thesis contribute to the ongoing research in this domain.

Chapter 3

Practical Part

The previous chapter has provided the reader with an overview of the theoretical background of the the main algorithms used in this thesis. At the beginning of chapter 3, I am going to give an introduction to the manufacturing process in which the sensory data used in the experiments was recorded. This introduction is not intended to provide the reader with precise technical details about the manufacturing process, which can be found in related engineering literature. Instead, the goal is to create an intuitive understanding of the data generation process and this is why the following section contains multiple images taken during the ongoing production at different manufacturing steps. Afterwards, I am going to outline the construction of an NMF-based model, which allows for the extraction of highly interpretable information from sensory data that is recorded from a manufacturing processes. This model is formulated in a general way to make it applicable to different kinds of measurement data recorded as time series. After this, the model is applied to real-world data and its capabilities are presented. In the last part of chapter 3 the model is combined with the previously outlined regression approach and I am going to show how it is possible to design a virtual sensor which can be used to for in-line monitoring of a manufacturing process.

3.1 The Manufacturing Process: Gravity Mould Casting

The thermal manufacturing process, from which the datasets are taken, is called "gravity mould casting." As already mentioned, a thorough technical description of casting processes is beyond the scope of this thesis, instead this chapter includes multiple graphical illustrations and images to give the reader an overall understanding of the data generation process. Typically, motor parts are produced with gravity casting, which is a casting processes especially suited for the production of complex geometries. In the following, I will explain the main

concepts of a casting process by providing an illustrated guide through the production cycle of a cylinder head.

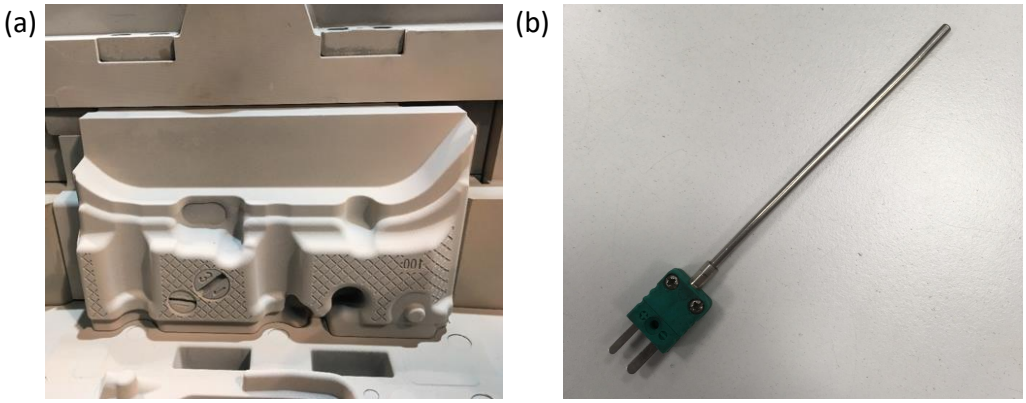

Fig. 3.1 (a) A look at the inside of the cavity. The white surface is due to a coating with a release agent. (b) A NiCr-Ni temperature sensor, which is placed inside the steel of the mould.

Gravity mould casting, sometimes referred to as permanent mould casting, is a repeatable casting process used for non-ferrous alloy parts, typically aluminium or copper based alloys. As the name suggests, the main force driving the filling process is gravity. The process can be summarised into three main stages:

1. The mould is preheated and prepared for the filling process. An important step, which is done before the casting process starts, is the preparation of the cavity's surface. Fig. 3.1 (a) shows an inside view of the cavity surface, that has been coated with a release agent (white color). Usually, this release agent is a mixture of Na_2O and SiO_2, which dries under the influence of CO_2. This coating ensures that the finished casting part can be removed from the cavity after solidification. The coating also has an isolation effect, i.e. it decreases the heat transfer number, which affects the temperature profile inside the cavity. Secondly, it smooths the roughness of the surface and thus influences the flow velocity of the liquid metal during the filling process. Both effects are critical for the production of parts with sufficient quality. Usually, the coating is applied manually with a spray pistol. Fig. 3.1 (b) shows a NiCr-Ni temperature sensor, which can be embedded in the cavity. From the backside of the cavity, holes are drilled that end 1-2

cm below the cavity surface and in which the temperature sensors are embedded. At this stage, sand cores are sometimes placed into the empty cavity to form the inner geometry of the metal part to be produced. In fig. 3.2, the empty cavity and the casting machine are shown. On the left side of the cavity, the "mould spure" can be seen, which has the purpose of "guiding" the liquid metal to flow into the cavity from the bottom up through a pipe system. Afterwards, the sand cores, which shape the geometry of the metal parts are placed into the cavity and the cavity is closed. Fig. 3.3 shows the casting machine after all preparations for the filling process are done.

Fig. 3.2 The empty cavity within the casting machine.

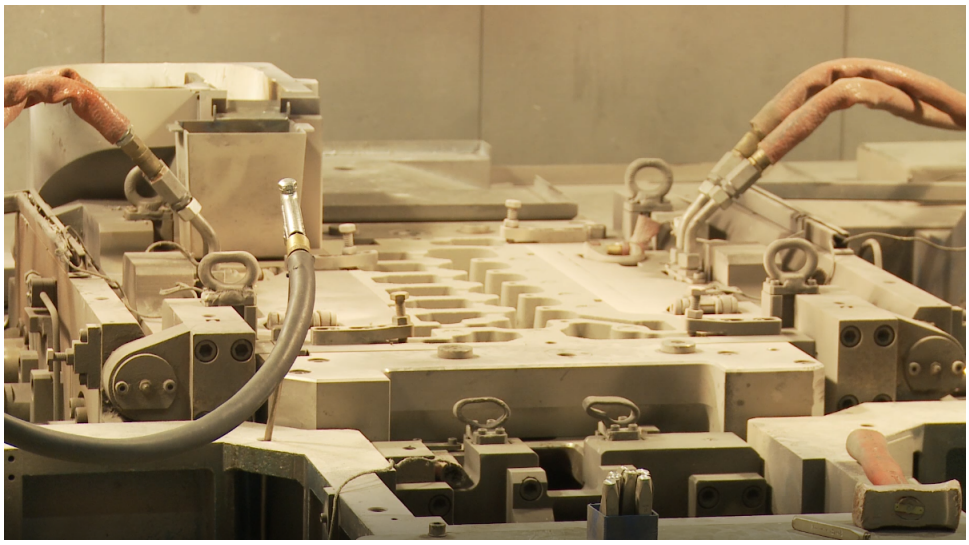

Fig. 3.3 The closed cavity with sand cores.

2. Now the casting ladle is moved over the mould sprue and tilted to start the filling process (see fig. 3.4). The liquid metal is now rising inside the cavity until it is completely filled. The molten metal flows into the cavity from below through the "mould sprue." This method reduces turbulence in the flow of the alloy, which might cause casting defects.

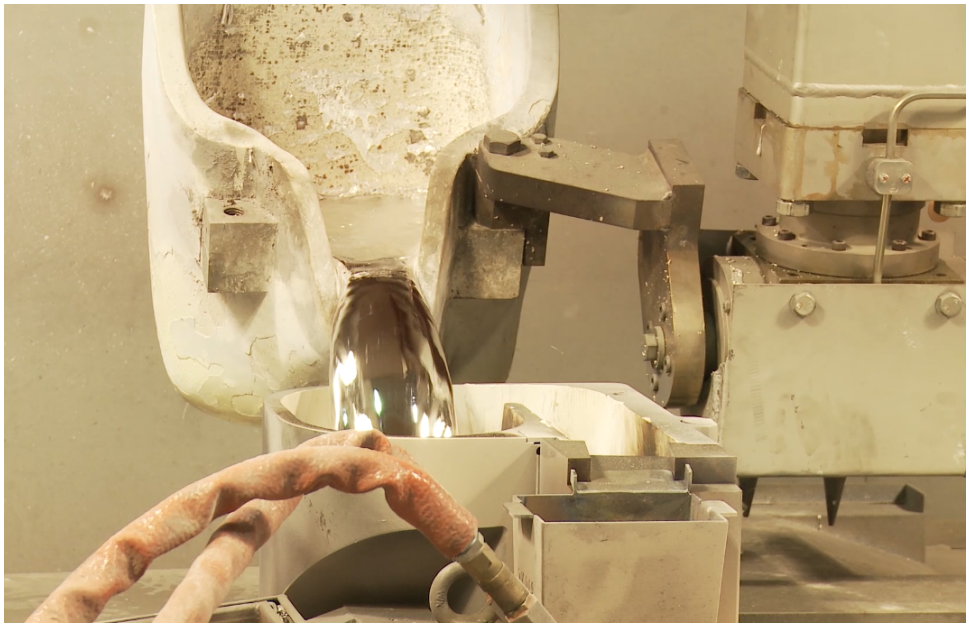

Fig. 3.4 The filling process (image provided by BMW AG).

3. This is the solidification phase. Usually, the solidification phase is guided by cooling or heating channels that run through the steel. Afterwards, the cavity is opened and the part is removed automatically. From fig. 3.5 to fig. 3.6, we can see how the metal is shrinking down due to solidification and during this critical time period, most of the casting defects will occur, if solidification is not controlled. The final step is the retrieval of the casting part (see fig. 3.7) and afterwards the removal of the sand cores that are still inside the casting part (fig. 3.8).

Fig. 3.5 The cavity filled with liquid metal (image provided by BMW AG).

Fig. 3.6 The metal shrinks during the solidification (image provided by BMW AG).

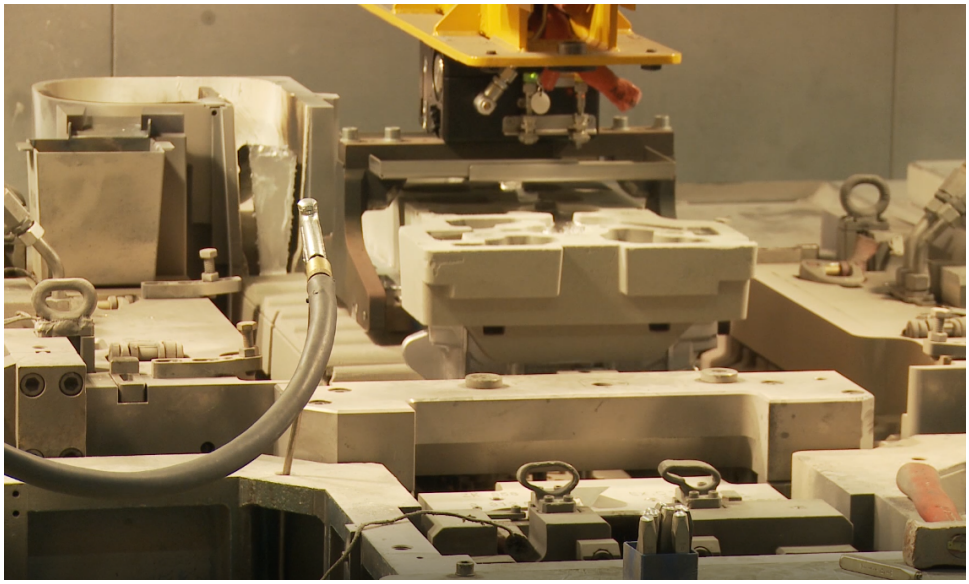

Fig. 3.7 The solid casting part is automatically removed from the cavity (image provided by BMW AG).

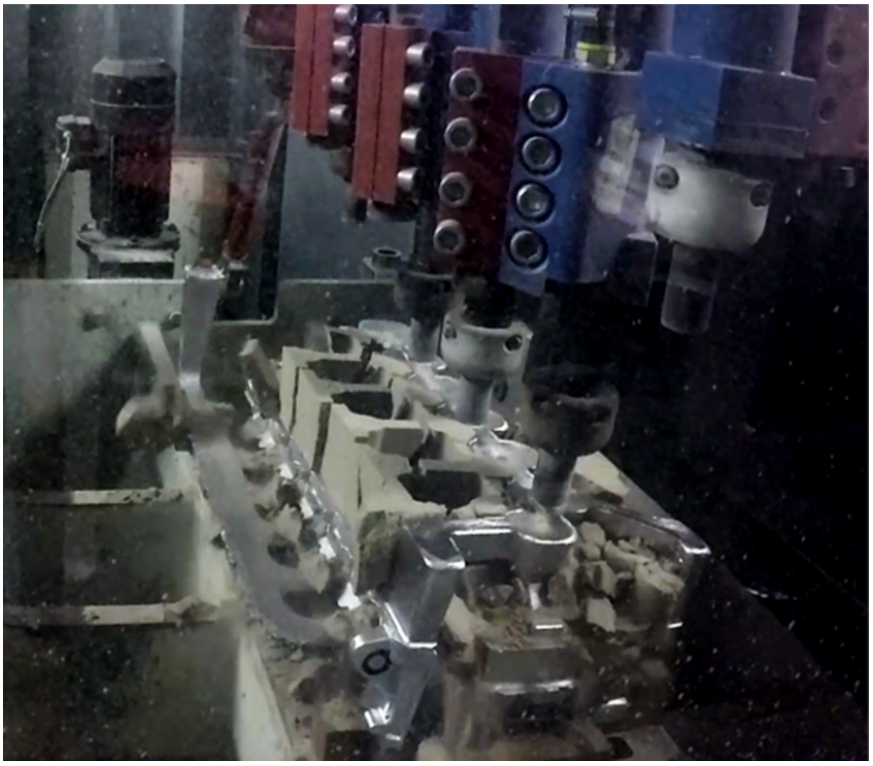

Fig. 3.8 Sand cores are removed with hammers (image provided by BMW AG).

3.1.1 Data Collection during the Casting Process

The whole casting process is monitored by multiple temperature sensors placed in different positions in the steel mould.

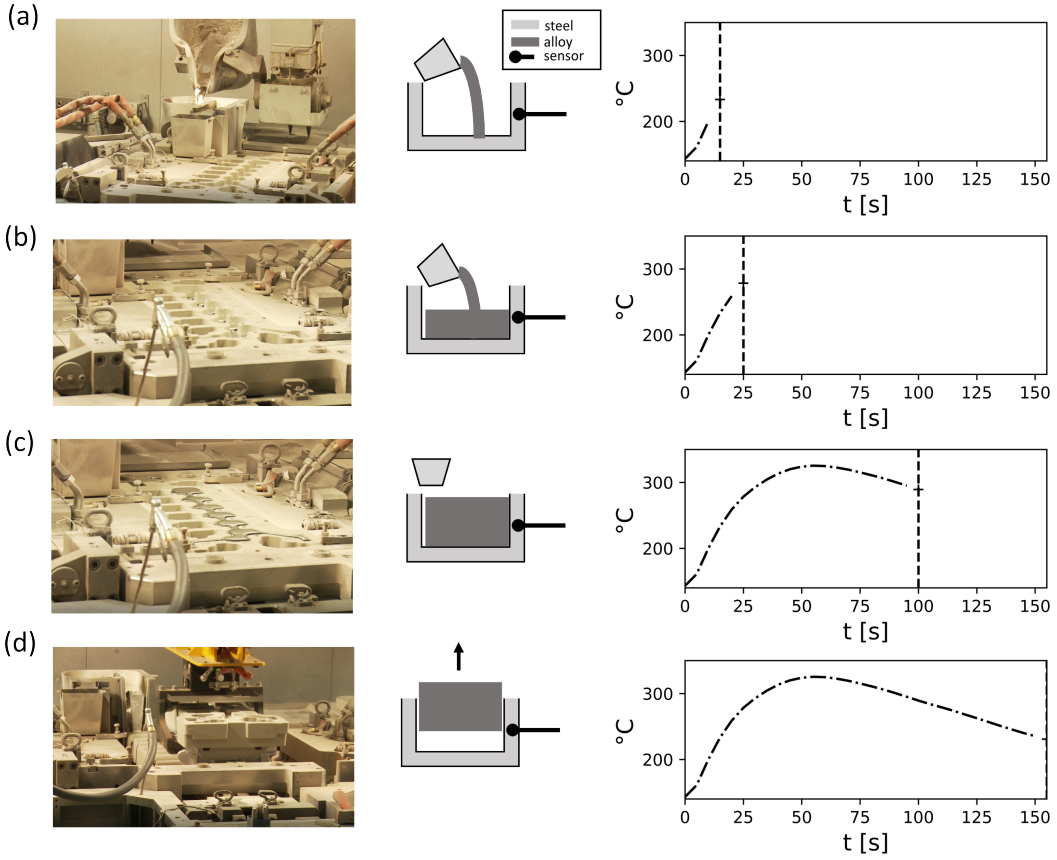

Fig. 3.9 Illustration of the data generation process. (a) Liquid metal is poured into the cavity. Until the metal reaches the sensor position, the sensor reflects the temperature of the steel. (b) The liquid metal reaches the position of the sensor. The temperature starts to rise. (c) Cavity is filled and metal cools down and solidifies. (d) Solid metal part is ejected and the temperature recording stops.

In this study, I focus on the sensor signals collected from one specific sensor during consecutive production of components (see fig. 3.1 (b) for an image of a typical Ni-NiCr sensor used to monitor the casting processes). These recordings resulted in a datasets $\boldsymbol{T} \in \mathbb{R}^{N \times M}$, where N is the number of production cycles, i. e. consecutive time series recorded at the sensor, and M the number of time points at which the sensor signal was sampled. Fig. 3.9 schematically illustrates how a typical temperature curve is generated during the casting process. When the liquid metal is poured in and reaches the sensor position, the temperature at the sensor starts to rise. The sensor then records a temperature - time

curve, whose shape is determined by the heat flux of the cooling metal, cooling channels and heating in the steel mould. The whole casting process takes roughly three minutes (depending on the specific part type) and the sensor stops recording when the cavity automatically opens after the solidification is finished.

3.2 The NMF Approach for Time Aeries of Physical Quantities

The focus of this thesis lies on analysing temperature time series with NMF, but the approach, which is outlined in this section is generally applicable to any time series of a physical quantity, if certain conditions are met. So in this section, the approach is motivated from the viewpoint of temperature time series at first, but the end result is going to be formulated in a general form.

3.2.1 Physics: The Heat Equation

Considering the flow of heat energy $Q = c_p \rho T$ inside the mould and the liquid metal filling, local temperature changes, as recorded by any specific sensor, are determined by the non-stationary heat equation, which represents a parabolic partial differential equation

$$\frac{\partial T(\boldsymbol{r},t)}{\partial t} - \lambda_T \nabla^2 T(\boldsymbol{r},t) = f(\boldsymbol{r},t),\ \boldsymbol{r} \in \Re^3, t > 0. \tag{3.1}$$

∇ denotes the nabla operator estimating the local gradient, $T(\boldsymbol{r},t)$ denotes the temperature measured at fixed sensor position $\boldsymbol{r}$ and at time points t, and $\lambda_T = \lambda_q/(\rho c_p)$ represents the thermal diffusivity, λ_q the thermal conductivity, ρ the mass density and c_p the specific heat at constant pressure [12]. For simplicity, here we consider units such that $\lambda_T = 1$. In our case, we also have external heat sources like heating and cooling channels, which can be modeled by a source function $f(\boldsymbol{r},t) \propto (\rho c_p)^{-1} \partial q(\boldsymbol{r},t)/\partial t$ where $q(\boldsymbol{r},t)$ denotes the heat flux density. The corresponding stationary and homogeneous situation is described by the Laplace equation $\boldsymbol{\Delta}_L T(\boldsymbol{r}) = 0$, the solution of which is composed of harmonic functions. Hereby, $\boldsymbol{\Delta}_L \equiv \nabla^2$ denotes the Laplace operator. However, the inhomogeneous stationary case is ruled by the Poisson equation $\boldsymbol{\Delta}_L T(\boldsymbol{r}) + f(\boldsymbol{r}) = 0$ and its solutions can be obtained with the help of Green's functions.

A special solution of the heat equation, called the fundamental solution, yields the heat kernel, which belongs to the family of exponential functions (a Gaussian):

$$T(\boldsymbol{r},t) = \frac{1}{\sqrt{2\pi\sigma^2(t)}} \exp\left(-\frac{\| \boldsymbol{r} \|^2}{2\sigma^2(t)}\right), \tag{3.2}$$

where σ^2 denotes the spatial variance and $\| .. \|$ the Euclidean norm.

The general solution to the non-stationary, inhomogeneous heat equation is given by a sum of contributions

$$T(\boldsymbol{r},t) = T^h(\boldsymbol{r},t) + T^s(\boldsymbol{r},t), \tag{3.3}$$

where $T^h(\boldsymbol{r},t)$ solves the homogeneous case, and $T^s(\boldsymbol{r},t)$ denotes the specific solution to the inhomogeneous problem in case of vanishing initial contributions from the external heat sources:

$$\begin{aligned} \frac{\partial T^s(\boldsymbol{r},t)}{\partial t} = \nabla^2 T^s(\boldsymbol{r},t) + f(\boldsymbol{r},t), \quad \boldsymbol{r} \in \Re^3, \quad t > 0 \\ \text{and} \quad T^s(\boldsymbol{r},t=0) = 0. \end{aligned} \tag{3.4}$$

Here, the source function $f(\boldsymbol{r},t)$ describes any heat source active during the process. So the resulting temperature at the sensor can be written as a sum of contributions, of which one captures the information about external sources in the process.

3.2.2 Deriving the Matrix Decomposition

Let $T(\boldsymbol{r}_0,t)$ be the solution to the heat equation given in (3.1). $T(\boldsymbol{r}_0,t)$ describes the sensor signal recorded by a sensor at fixed sensor position $\boldsymbol{r}_0$. With NMF, we wish to model the registered temperature - time curves $T(\boldsymbol{r}_0,t)$ at fixed sensor position $\boldsymbol{r}_0$ as a linear superposition of unknown, independent component processes.

In order to connect the solution of the heat equation with a matrix decomposition, let us look at the definition of the multivariate Taylor expansion

$$f(\boldsymbol{x}) = \sum_{|\alpha|\leq k} \frac{D^\alpha f(\boldsymbol{a})}{\alpha!}(\boldsymbol{x}-\boldsymbol{a})^\alpha + \sum_{|\beta|=k+1} R_\beta(\boldsymbol{x})(\boldsymbol{x}-\boldsymbol{a})^\beta \tag{3.5}$$

$$R_\beta(\boldsymbol{x}) = \frac{|\beta|}{\beta!}\int_0^1 (1-t)^{|\beta|-1} D^\beta f(\boldsymbol{a}+t(\boldsymbol{x}-\boldsymbol{a}))dt, \tag{3.6}$$

with $D^\alpha f = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1}\cdots\partial x_n^{\alpha_n}}$, $|\alpha| \leq k$ and the sum being constructed using the multi-index notation

$$|\alpha| = \alpha_1 + \cdots + \alpha_n, \quad \alpha! = \alpha_1!\cdots\alpha_n!, \quad \boldsymbol{x}^\alpha = x_1^{\alpha_1}\cdots x_n^{\alpha_n}. \tag{3.7}$$

In the derivation of $T(\boldsymbol{r}_0,t)$, multiple external factors, physical parameters or initial conditions like initial temperature or changing isolation effects have to be considered. So $T(\boldsymbol{r}_0,t)$ can be seen as a function which also depends on these quantities. For example, we could extend the arguments in the following form:

$$T(r_0,t,T_0,T_S,\alpha), \tag{3.8}$$

where T_0 is the initial temperature of the steel at the sensor position, T_S is the initial temperature of the liquid metal and α is the thermal diffusivity. Note that this is a simplification of the complex three dimensional scenario of a real world thermal process. With the definition of the multivariate Taylor expansion from above, $T(r_0,t,T_0,T_S,\alpha)$ can be approximated up to the first order as a matrix decomposition:

$$T(r_0,t,T_0,T_S,\alpha)|_{T_0=T_0^*,T_S=T_S^*,\alpha=\alpha^*} \simeq \tag{3.9}$$

$$T(r_0,t,T_0^*,T_S^*,\alpha^*)+ \tag{3.10}$$

$$\frac{\partial T}{\partial T_0}|_{T_0=T_0^*}(T_0-T_0^*)+ \tag{3.11}$$

$$\frac{\partial T}{\partial T_S}|_{T_S=T_S^*}(T_S-T_S^*)+ \tag{3.12}$$

$$\frac{\partial T}{\partial \alpha}|_{\alpha=\alpha^*}(\alpha-\alpha^*)= \tag{3.13}$$

$$\begin{bmatrix}1 & (T_0-T_0^*) & (T_S-T_S^*) & (\alpha-\alpha^*)\end{bmatrix} \times \begin{bmatrix} T(r_0,t,T_0^*,T_S^*,\alpha^*) \\ \frac{\partial T}{\partial T_0}|_{T_0=T_0^*} \\ \frac{\partial T}{\partial T_S}|_{T_S=T_S^*} \\ \frac{\partial T}{\partial \alpha}|_{\alpha=\alpha^*} \end{bmatrix}. \tag{3.14}$$

Since we want to extract and model contributions, which change during the running production, t and r_0 are kept constant. From this relation, we can expect that if the variations occurring during the ongoing production are small enough, their overall effect on the temperature signal should be approximately proportional to their corresponding partial derivative.

As an example, consider a simple physical system with a time-dependent temperature $T(t)$ that has the initial temperature $T(t=0)=T_0$. This system is embedded into an infinite space at constant temperate $T_S \geq T_0$. The temporal dynamics of this system are described by the one-dimensional heat transfer equation

$$\frac{\partial T(t)}{\partial t} = \alpha(T_S - T(t)). \tag{3.15}$$

Solving (3.15) yields a simple exponential heating process:

$$T(t) = T_S + (T_0 - T_S)e^{-\alpha t} \tag{3.16}$$

If we take the partial derivatives $\frac{\partial T}{\partial T_0}, \frac{\partial T}{\partial T_S}$ and $\frac{\partial T}{\partial \alpha}$ from (3.16), we obtain the following expressions:

$$\frac{\partial T}{\partial T_S} = 1 - e^{-\alpha t}, \tag{3.17}$$

$$\frac{\partial T}{\partial T_0} = e^{-\alpha t}, \tag{3.18}$$

$$\frac{\partial T}{\partial \alpha} = (T_S - T_0)te^{-\alpha t}. \tag{3.19}$$

If we suppose that, during the ongoing production, these three parameters vary between different manufacturing processes, then also the recorded temperature time curve will vary. If these process-related variations are small enough (i.e. non-linearities are negligible), we can model their distinct effect on the signal response with the exponential expressions given in (3.17), (3.18) and (3.19). Let us further look at the following intervals:

$$T_0 \in [T_{0,min}, T_{0,max}], \tag{3.20}$$

$$T_S \in [T_{S,min}, T_{S,max}], \tag{3.21}$$

$$\alpha \in [\alpha_{min}, \alpha_{max}], \tag{3.22}$$

which are the smallest intervals, which contain all values that T_0, T_S and α take on during the considered period of time of running production. If we calculate the Taylor expansion at $\alpha^* = \alpha_{min}$, $T_0* = T_{0,min}$ and $T_S* = T_{S,min}$, the bracket terms $(\alpha - \alpha^*), (T_0 - T_0^*)$ and $(T_S - T_S^*)$ in (3.11),(3.12) and (3.13) will be strictly nonnegative. By enforcing nonnegativity on the weight matrix $\mathbf{W}$, the NMF solution should be guided to extract the evaluation of the partial derivatives at the lower bounds of the intervals given in (3.20), (3.21) and (3.22), i.e. the NMF decomposition estimates the first order Taylor approximation at α_{min}, $T_{0,min}$ and $T_{S,min}$. Fig. 3.10 illustrates schematically how the processes described by the partial derivatives (3.11) to (3.13) will affect the shape of the measured temperature signal. Fig. 3.10 (a) shows the effect if the thermal diffusivity α is changed, which will result in a faster rising temperature but also a faster cooling effect. The dashed signal is calculated with a lower value for α. The change in curvature shape is plotted in the middle column, which is described by the partial derivative shown in the right column. Fig. 3.10 (b) and (c) respectively, illustrate

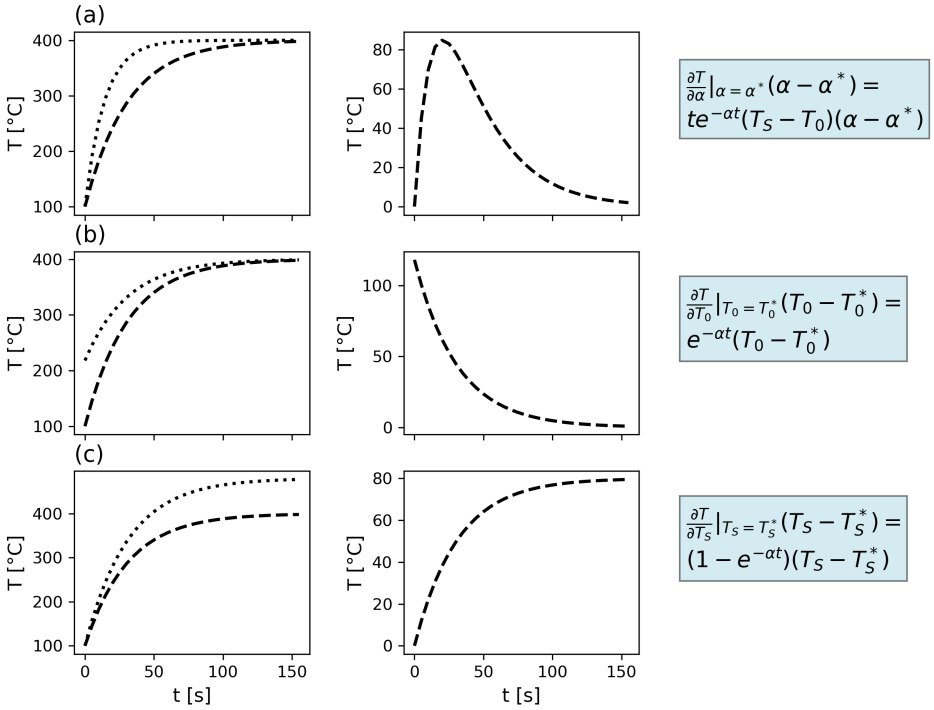

Fig. 3.10 Illustration of how different contributing factors change the shape of the recorded sensor signal. The left column shows the initial curve (dashed line) and the curve after one parameter was changed (dotted line). The middle column shows the shape of the change and the right column the corresponding partial derivative from the Taylor expansion. For (a), the parameter α was increased. For (b), the initial temperature T_0 and for (c), the environmental temperature T_S were increased.

the effect of a changing initial temperature at the sensor position and a changing environment temperature T_S.

In general, this interpretation makes NMF a promising candidate in many other problem settings. If the error or the first order Taylor approximation is small, any physical quantity $Q(r_0, t, p_1, p_2, \ldots, p_Z)$ can be decomposed with a similar interpretation. Here, p_n ($n = 1, \ldots, Z$) are physical parameters which describe the quantity Q. Writing the first order Taylor expansion at $p_{n,min}$ ($n = 1, \ldots, Z$) in matrix form, with the minimal values defined in the

same way as in (3.20) to (3.22), we obtain:

$$Q(r_0,t,p_1^*,\ldots,p_Z^*) \simeq Q(r_0,t,p_{1,min},\ldots,p_{Z,min}) + \sum_{n=1}^{Z} \frac{\partial Q}{\partial p_n}\bigg|_{p_n=p_{n,min}} (p_n^* - p_{n,min}) = \quad (3.23)$$

$$\begin{bmatrix} 1 & (p_1^* - p_{1,min}) & \ldots & (p_Z^* - p_{Z,min}) \end{bmatrix} \times \begin{bmatrix} Q(r_0,t,p_{1,min},\ldots,p_{Z,min}) \\ \frac{\partial Q}{\partial p_1}\Big|_{p_1=p_{1,min}} \\ \vdots \\ \frac{\partial Q}{\partial p_Z}\Big|_{p_K=p_{Z,min}} \end{bmatrix} = \quad (3.24)$$

$$\mathbf{W}(p_1^*,\ldots,p_Z^*)\mathbf{\Theta}(p_{1,min},\ldots,p_{Z,min}) \quad (3.25)$$

Expression (3.25) can be seen as the reconstruction of one single row in an NMF decomposition with $K = Z+1$ components. Due to the construction of $\mathbf{\Theta}(p_{1,min},\ldots,p_{K,min})$ with $p_{1,min},\ldots,p_{Z,min}$, it is clear that the component processes are the same for all temperature signals in one dataset. $\mathbf{W}(p_1^*,\ldots,p_Z^*)$ will be different for every recorded sensor signal, because it depends on the varying $p_1^*,\ldots,p_Z^*$.
By the aid of the Taylor expansion, it is also possible to naturally describe the limitations of the NMF-based approach. If the non-linear terms in the Taylor expansion have a significant contribution, the linear NMF model will likely construct mixture components, which lack interpretability.

3.2.3 A Physics Inspired Initialisation Strategy for NMF

The connection between NMF and underlying physical processes through the Taylor expansion given in (3.25) can be used to design a new kind of initialisation strategy. From (3.25), we expect NMF to estimate an approximation of the first order Taylor expansion, which shall yield highly interpretable component processes. Since the NMF result strongly depends on the algorithm's initialisation, this motivates the following approach:

1. Let $Q(t,r)$ be the physical quantity that is measured with a sensor during a manufacturing process over a fixed amount of time and with a fixed time step Δt. The manufacturing process is repeated N times and all recordings of Q are saved and

ordered chronologically in a data matrix

$$\mathbf{Q} = (Q_{nm}),\ Q_{nm} = Q_n\big(t_m = (m-1)\Delta t\big) \tag{3.26}$$

$$\text{with } n = 1,\ldots,N \text{ and } m = 1,\ldots,M. \tag{3.27}$$

Here, $Q_n(t)$ is the sensor recording from the n-th manufacturing process performed. Each signal starts at $t = 0$ and ends at $(M-1)\Delta t = t_{end}$.

2. Estimate a physical model for $Q(t,r) = Q(t,r,p_1,\ldots,p_Z)$, with $p_1,\ldots,p_Z$ being physical parameters. The latter describe the physical mechanisms which generate the measured quantity and can also vary during the consecutive production.

3. Derive or approximate the partial derivatives $\frac{\partial Q(t,r,p_1,\ldots,p_Z)}{\partial p_k}$ and construct the initial matrix $\mathbf{\Theta}_{init}$ in the following way:

$$\mathbf{\Theta}_{init} = \begin{pmatrix} \overline{Q(0)} & \overline{Q(\Delta t)} & \ldots & \overline{Q((M-1)\Delta t)} \\ \frac{\partial Q(0)}{\partial p_1} & \frac{\partial Q(\Delta t)}{\partial p_1} & \cdots & \frac{\partial Q\left((M-1)\Delta t\right)}{\partial p_1} \\ \vdots & \vdots & \ldots & \vdots \\ \frac{\partial Q(0)}{\partial p_Z} & \frac{\partial Q(\Delta t)}{\partial p_Z} & \cdots & \frac{\partial Q\left((M-1)\Delta t\right)}{\partial p_Z} \end{pmatrix} \tag{3.28}$$

Here, $\overline{Q(t)}$ is defined as the mean of all the signals in the dataset $\mathbf{Q}$, i.e.

$$\overline{Q(t)} = \frac{1}{N}\sum_{n=1}^{N} Q_n(t). \tag{3.29}$$

Afterwards, the rows in $\mathbf{\Theta}_{init}$ should be normalised. In the course of this thesis, the rows were always scaled to have L_1-norm length one, i.e. $\sum_{m=1}^{M} |(\Theta_{init})_{im}|_1 = 1$ for $i = 1,\ldots,(Z+1)$.

4. The initial weight matrix $\mathbf{W}_{init}$ is estimated with the Moore-Penrose pseudo-inverse (see section 2.1.1.3):

$$\mathbf{W}_{init} = \mathbf{\Theta}_{init}^{-1} \cdot \boldsymbol{Q} \tag{3.30}$$

The idea behind this approach is to initialise the decomposition close to a highly interpretable result. If an extracted component resembles one of the curves that we obtain from the partial derivatives, this might indicate that it indeed yields information about a specific physical quantity which varies during running production. The reason to use the mean of all recorded sensor signals is based on the fact that in (3.25), the first row is given by the evaluation of

$Q(t,r) = Q(t,r,p_1,\ldots,p_Z)$ at the minimum values of the parameters $p_1,\ldots,p_Z$, which in reality should not deviate significantly from the mean. This approach further tackles some of the main inherent difficulties of any NMF implementation. First, the non-uniqueness problem is relaxed by having a fixed initialisation. Second, there is the determination of the number of components to extract. In the framework of this approach, the number of components is estimated as the number of terms in the first order Taylor expansion, that are expected to make a significant contribution to the overall signal, which at least sets an upper limit to the number of components. An additional advantage is that this approach makes it possible to use domain knowledge to guide the NMF decomposition to extract the desired solutions.

3.2.3.1 Initialisation for Temperature Time Curves

The outlined approach in this section is demonstrated with temperature recordings from different casting processes. For this sake, two different initialisation matrices are mainly used in the following chapters. The physical model used to construct the initialisation is the simple heating process from (3.16). This process relates to the scenario of metal casting, where we mainly have a strong heating-up process due to the heat flow from the liquid metal. This model does not include the cooling phase during solidification, but, as it is going to be shown in the experiments, using this simple model, it is already possible to extract interpretable and usable decompositions from the temperature curves. Depending on the placement of the specific sensor within the steel, the effect of the cooling phase might not be noticeable during the time of one casting process. Let $\hat{T}(t)$ be the equation which describes our process:

$$\hat{T}(t) = T_S + (T_0 - T_S)e^{-\alpha t} \tag{3.31}$$

With this equation, we can define the row vector $(\hat{\mathbf{T}})_{m+1} = \hat{T}(m\Delta t)$ $(m = 0,(M-1))$ which has M elements. $\mathbf{T} \in \mathbb{R}^{N\times M}$ is the data matrix, which contains the recorded sensor signals as defined before. Note that in the three-dimensional heat equation, the thermal diffusivity, which describes a material property, is used, but in the real world process, the material is not isotropic and we have to consider a heat transfer through a system made up of different materials. This is why in this case α is the heat transfer coefficient. Following the initialisation strategy outlined in section 3.2.3, we design two different initialisation matrices,

which are going to be used to initialise the decomposition of real-world data:

$$\boldsymbol{\Theta}_{init,1} = \begin{pmatrix} \overline{\mathbf{T}} \\ \frac{\partial}{\partial T_0}\hat{\mathbf{T}} \\ \frac{\partial}{\partial \alpha}\hat{\mathbf{T}} \end{pmatrix}, \qquad \boldsymbol{\Theta}_{init,2} = \begin{pmatrix} \overline{\mathbf{T}} \\ \frac{\partial}{\partial T_0}\hat{\mathbf{T}} \\ \frac{\partial}{\partial \alpha}\hat{\mathbf{T}} \\ \frac{\partial}{\partial T_S}\hat{\mathbf{T}} \end{pmatrix} \tag{3.32}$$

Here, $\overline{\mathbf{T}}$ is calculated with $\mathbf{T}$ using the definition provided in (3.29). The parameter T_0 can be directly related to an important process quantity, which is the temperature of the steel cavity at the sensor position. The effective thermal diffusivity α can be related to the heat transfer between the alloy and the steel at the sensor position. The thermal conductivity between alloy and steel can vary due to effects like a different layer of release agent or gap formation between the solidifying part and the steel cavity. T_S is defined as ambient temperature in the simplified physical model in (3.31), which does not directly relate to any process parameter in the real casting process, because the temperature of the liquid metal is not constant. One can suspect, that it can be associated with the initial temperature of the liquid metal. In the experiments section, $\Theta_{init,1}$ and $\Theta_{init,2}$ are used to initialise the decompositions and the results are compared.

3.2.4 The NMF Model for Temperature Time Curves

To sum up, the here outlined approach yields a way to approximate the first order Taylor expansion for a physical quantity which is measured during a manufacturing process over a certain time period. The approach exploits the fact that we have a large number of sensor recordings, which are typically generated during series production. With the multivariate Taylor expansion, it is possible to connect the measured time curve to a matrix decomposition, as is shown in (3.25). The Taylor expansion is estimated in such a way that the coefficients of the expansion terms are all nonnegative. This then motivates the use of NMF to approximate the decomposition (right-hand side of (3.25)) of the time curves. As NMF does not necessarily have to estimate a solution which approximates the desired Taylor expansion terms, the initialisation of NMF can be constructed using partial derivatives derived from a simplified physical model of the process. This way, the optimisation procedure can be guided to estimate solutions which lie close to the first order terms in (3.25) and as such are highly interpretable.

Fig. 3.11 illustrates the NMF model for the decomposition of temperature datasets:

$$\mathbf{T} = \mathbf{W}\boldsymbol{\Theta} \tag{3.33}$$

The data matrix $\mathbf{T}$ is constructed as explained in the previous chapters by stacking the temperature curves in chronological order. This way the weights in $\mathbf{W}$ will also be ordered chronologically, which will then help us in detecting time dependencies in the individual contribution of the component processes.

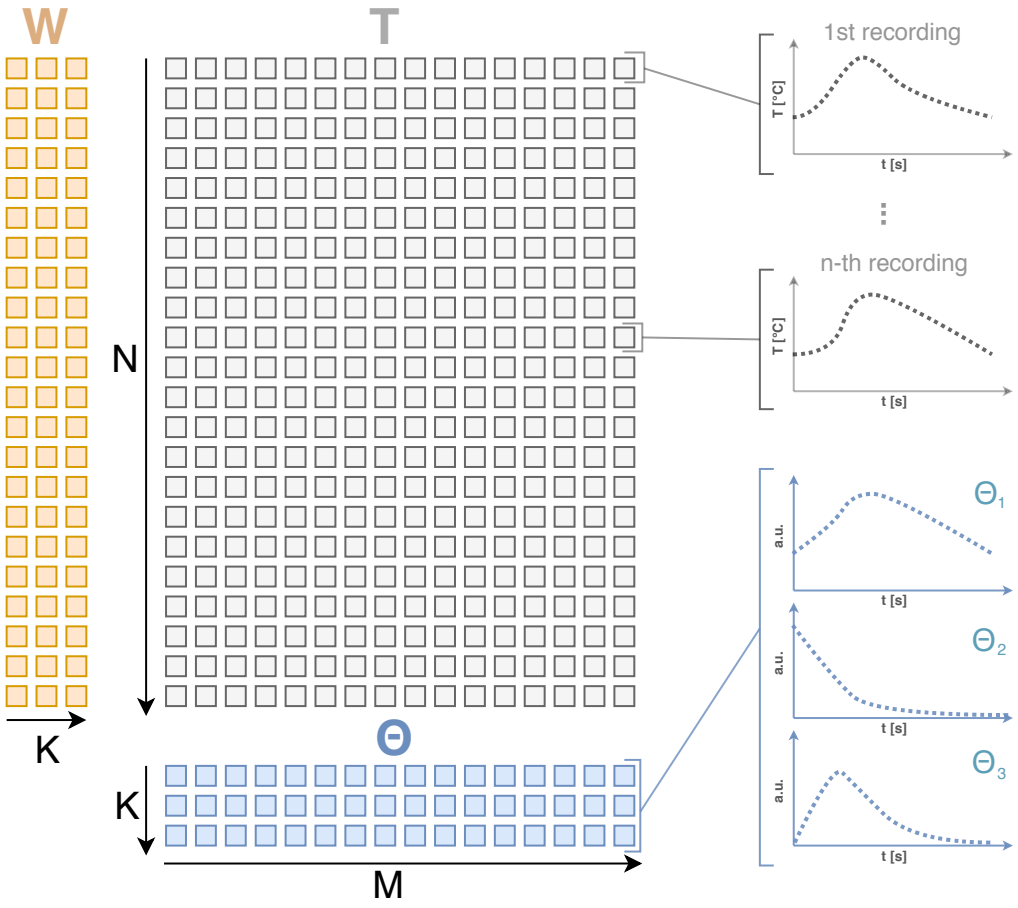

Fig. 3.11 Illustration of the matrix factorisation model $\mathbf{T} \simeq (\mathbf{W}, \boldsymbol{\Theta})$. $\mathbf{T}$ contains the recordings of a specific sensor from consecutive production of parts. The rows of $\mathbf{T}$ are ordered chronologically. $\boldsymbol{\Theta}$ contains the basis functions and W the activations of the corresponding basis functions for a specific time series.

In fig. 3.11, the data matrix $\mathbf{T}$ is decomposed into $K = 3$ component processes, which are contained in the rows of $\boldsymbol{\Theta}$. If we initialise the NMF optimisation with $\Theta_{init,1}$ in (3.32), we expect the extracted component processes to resemble the curves shown in fig. 3.13. From this resemblance, it becomes possible to classify the origin of the extracted component

processes. For example, $\mathbf{\Theta}_3$ in fig. 3.11 probably describes a process-related variation in the thermal diffusivity of the heat flow towards the sensor. Of course, this classification by resemblance always needs to be verified by expert knowledge or experiments, but in the following chapters, it is going to be shown that this visual interpretation approach yields practical results.

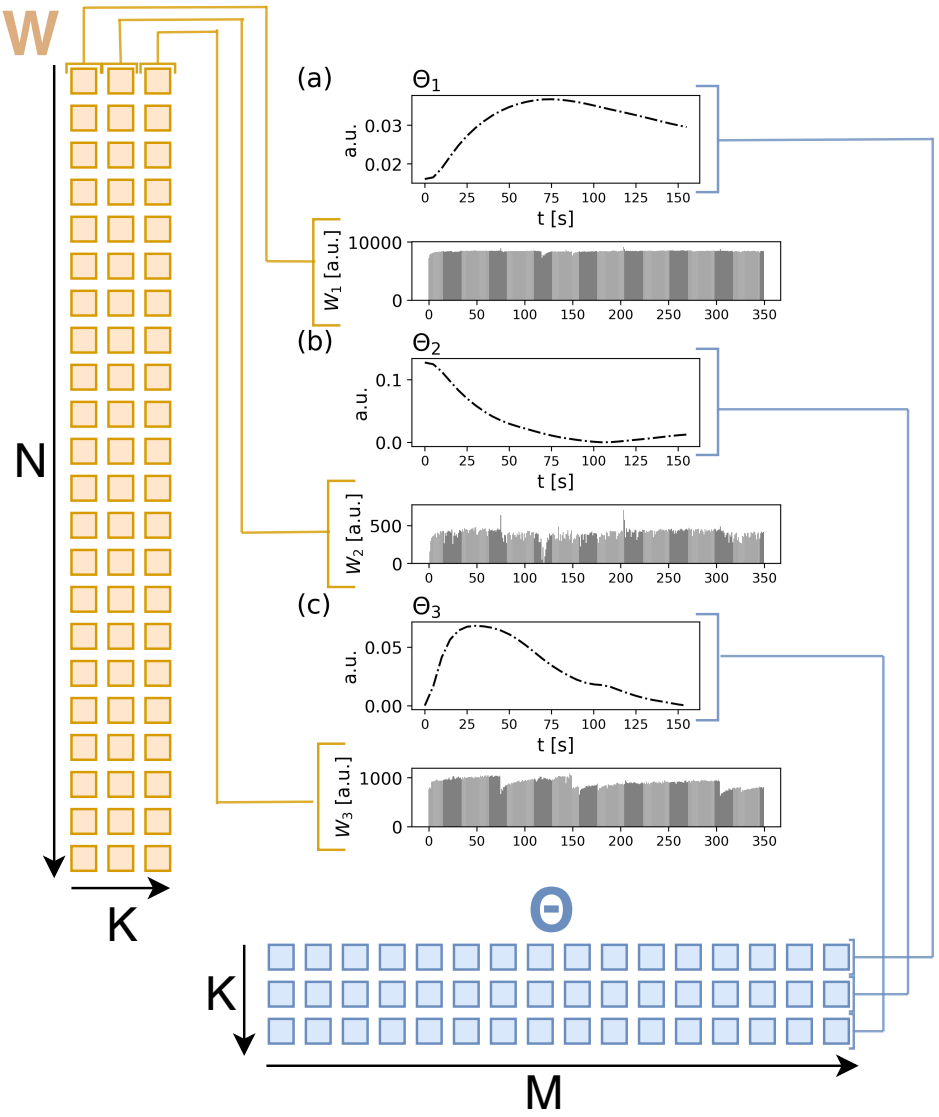

Fig. 3.12 Depiction of how the results of the matrix factorisation model $\mathbf{T} \simeq (\mathbf{W}, \mathbf{\Theta})$ are presented in this thesis.

In the following section, multiple decomposition results are going to be presented. Fig. 3.12 gives a description of how these results are going to be plotted. The extracted component

processes are contained in the rows of matrix $\mathbf{\Theta}$ and are referred to as $\mathbf{\Theta}_i$. They are plotted as time curves. Below them, the corresponding weights are plotted, which are referred to as $\mathbf{W}_i$. The x-axis is the row number in $\mathbf{W}$ or the *process number* of the n-th process. In the example in fig. 3.12, there would be $N = 350$ processes. M is the number of time steps of the recordings, i.e. $M = \frac{155}{\Delta t}$ in fig. 3.12, where Δt is the time step.

3.2.4.1 Decomposition of Toy Data

In order to demonstrate the ability of the approach outlined in this chapter and also to investigate the problem of mixture components, the outlined approach is applied to toy datasets obtained from simulations.
The data set contains 350 temperature time curves, which were generated by adding variation terms in (3.16):

$$T_k(t) = T_S + ((T_0 + T_i(k)) - T_S)e^{-(\alpha+\alpha(k))t},\ k = 1,\ldots,350 \tag{3.34}$$

$T_i(k)$ was varied with a triangle wave function and $\alpha(k)$ was varied with a sinusoidal wave function with different amplitudes and both were shifted vertically to be strictly nonnegative. For $T_i(k)$, the amplitude of the triangle wave was set to 0.1, 1 and 10. The amplitude for the sinusoidal wave $\alpha(k)$ was varied from 0.001, 0.1, 0.1. Fig. 3.13 shows $T_i(k)$ and $\alpha(k)$ with the amplitudes set $T_{0,A} = 0.1$ and $\alpha_A = 0.001$. If the NMF decomposition is able to model the variation within the dataset, it will yield component processes which resemble the triangular and the sinusoidal shape. Since only the initial temperature and the thermal diffusivity are varied, the first order Taylor expansion contains three different terms, so a $K = 3$ NMF decomposition is calculated. The results for the three simulation runs are shown in fig. 3.14. Fig. 3.14 (a)- (c) shows the result obtained from the low variation dataset. Here, the extracted component processes almost exactly match the partial derivatives

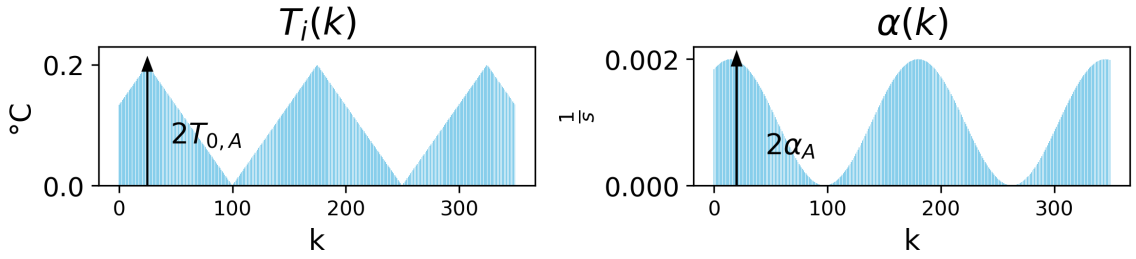

Fig. 3.13 Illustration of the simulated toy data set. The left shows the artificial variation of the initial temperature with a triangular wave function. The right-hand side shows the sinusoidal variation of the thermal diffusivity.

in (3.18) and (3.19). Also the weights $\mathbf{W_2}$ and $\mathbf{W_3}$ closely resemble the triangular and sinusoidal wave in fig. 3.13. If the magnitude of the variation is increased, we can see how NMF starts to extract mixture components (3.14 (d)-(f)). In (f), the weights $\mathbf{W_3}$ are already distorted, although the triangular shape is still visible. The shape of the component processes has slightly changed. The results of the simulation with the highest variation are shown in (g)-(i). Here, the component process $\mathbf{\Theta_2}$ and its corresponding weights $\mathbf{W_2}$ (fig. 3.14 (h)) do not resemble (3.18) or the sinusoidal wave function. The simulated variation is large enough for the linear approximation to not hold anymore and the output gets distorted by non-linearities and mixture components. The same effect can be seen in fig. 3.14 (i).

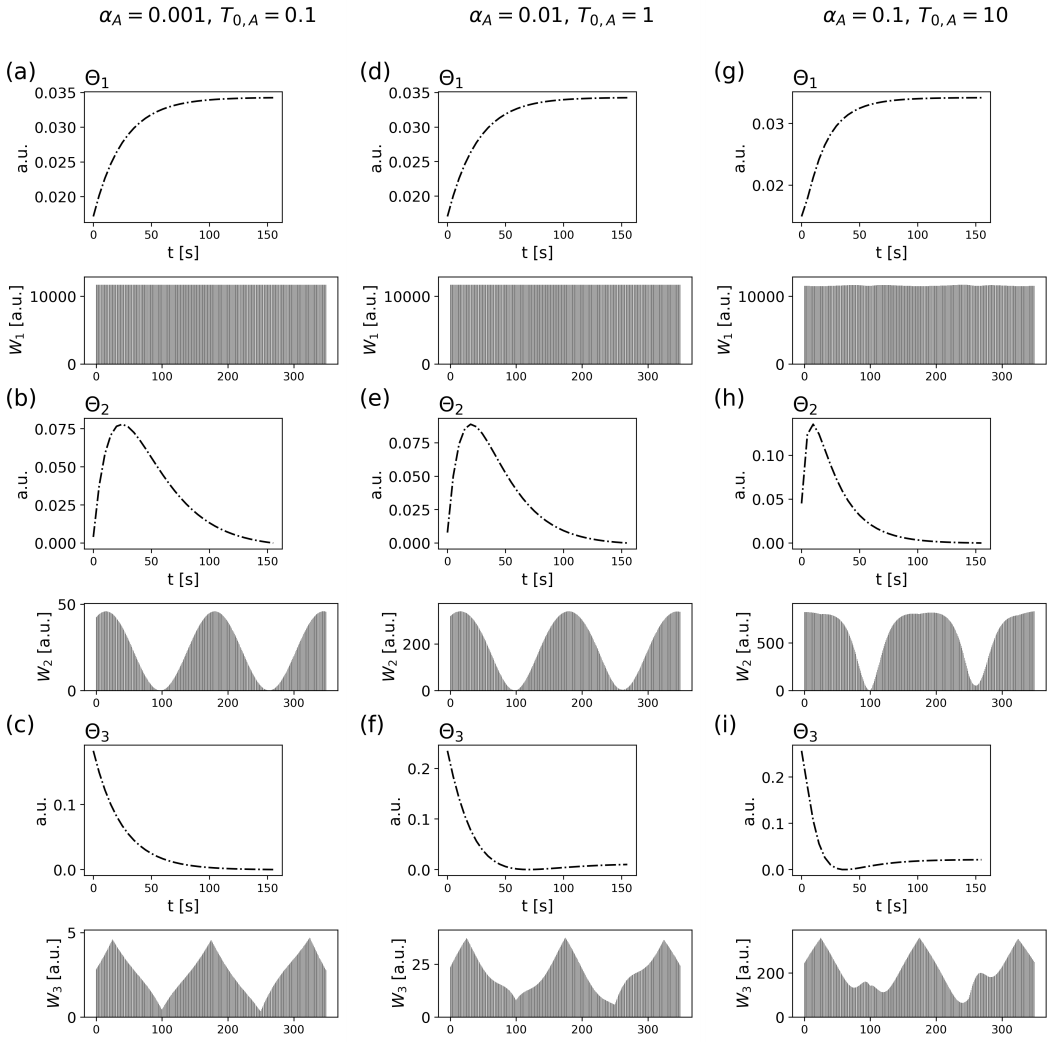

Fig. 3.14 Results with simulated data. Each column shows the decomposition results, obtained from a simulated dataset with different magnitude of inherent variations.

From this toy data example, we can see that the NMF decomposition can indeed estimate an approximation of the first order Taylor expansion, if the inherent variations in the dataset are small enough. This sets an expected natural limit to the outlined NMF-based approach, because NMF yields a linear decomposition into latent component processes and is unable to model non-linearities. If there are strong non-linear variations present in the dataset, the extracted component processes will lack the interpretability given by the connection to the first order Taylor expansion shown in (3.25). Real-world manufacturing processes are typically kept at a stable point and change only marginally, i.e. the assumptions about non-linearities is valid in many cases.

3.3 Application to Real-World Datasets

In this section, the outlined approach from the sections before is applied to data collected during the series production of a gravity casting process. The gravity casting process and the data generation process are described in section 3.1. The NMF algorithm used is the HALS algorithm, which is derived in section 2.1.6 in the theoretical part of this thesis. To initialise NMF, two different procedures are used, one being the data-driven NNDSVD approach explained in section 2.1.2.5 and the other being the physics motivated knowledge-based approach explained in 3.2.3, which aims at the estimation of the first order Taylor expansion. In the following, two different datasets are analysed, which stem from temperature measurements at two different positions from the production of a cylinder head. The process works almost exactly as the one shown in section 3.1. The actual casting machine cannot be depicted here, because this is confidential information of the partner company.

3.3.1 Real-World Data from a Thermal Manufacturing Process

3.3.1.1 Dataset 1: A Simple Process

The first dataset is generated by a sensor that is positioned in such a way that it mainly records the heat flow from the liquid metal and the cooling-down process during solidification. Fig. 3.15 (a) shows three example curves from the dataset. As can be seen, the temperature-time curve seemingly reflects a rather simple heating-up and cooling-down process. The sensor is positioned in such a way that it is affected by a cooling channel that is turned on at around 60 seconds and not turned off afterwards. Each curve records a time period of 155 seconds with a five seconds time step, i.e. each curve consists of 32 time points. Until roughly 60 seconds, the temperature is rising. At around this time point, the filling process is finished. Afterwards, the sensor measures a gradual decrease in temperature.This dataset is made up

of temperature curves from 350 consecutive processes, i.e the dimension of the data matrix is $\mathbf{T_1} \in \mathbb{R}^{350\times32}$.

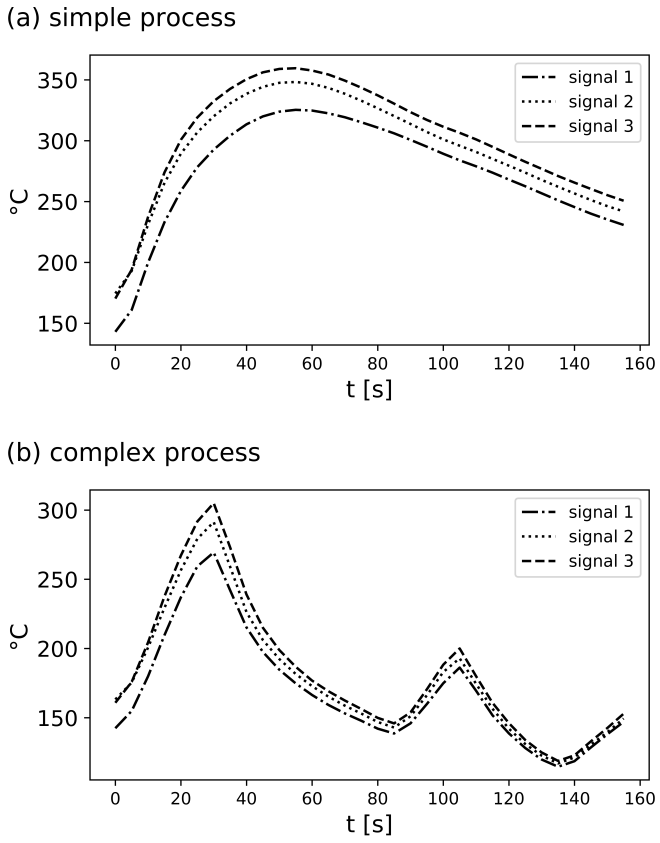

Fig. 3.15 Examples of three different temperature recordings from the two datasets. (a) A simple process with a heating and a cooling phase. (b) A more complex process with additional cooling effects.

3.3.1.2 Dataset 2: A Complex Process

The second dataset stems from the same processes but from a sensor at a different position in the cavity. The data matrix has the same dimensions and is referred to as $\mathbf{T_2} \in \mathbb{R}^{350\times32}$. In fig. 3.15 (b), three example curves from $\mathbf{T_2}$ are shown. As can bee seen, the time series is more complex that the ones shown in (a) because there is an additional cooling process, which starts at around 30 seconds. The reason for this is that the sensor position is close to a cooling channel, which is switched on at this time point and is then switched off from around 70 to 110 seconds. At 140 seconds, it is switched off again, which is why the temperature starts to rise again. Compared to the curves shown in (a), there are additional external contributions.

This dataset is chosen, because there it is possible to show, how the NMF decomposition deals with external sources acting on the temperature signal.

3.3.2 Results

3.3.2.1 Results for Dataset 1

$\mathbf{T}_1$ is decomposed with the knowledge-based initialisation using $\mathbf{\Theta}_{init,1}$ and $\mathbf{\Theta}_{init,2}$ from (3.32) for a respective decomposition with $K = 3$ and $K = 4$. Since the NMF output is not unique in terms of scaling, the rows of the resulting matrix $\mathbf{\Theta}$ are divided by their respective L_1-norm and the corresponding columns in $\mathbf{W}$ are multiplied by the same value. The goal is to render the extracted weights comparable in their individual contribution to the signals. The HALS algorithm, described in the theoretical part, was stopped after the change of reconstruction error during iteration steps was below 10^{-8}.
The resulting decomposition is shown in fig. 3.16 and an explanation on how to read the image is given in fig. 3.12. The left-hand side (a)-(b) shows the $K = 3$ decomposition initialised with $\mathbf{\Theta}_{init,1}$ and the right-hand side (d)-(g) the $K = 4$ decomposition initialised with $\mathbf{\Theta}_{init,2}$. First thing to notice is that $\mathbf{\Theta}_1$ makes the strongest contribution in the reconstruction of the original signal, as can be seen in $\mathbf{W}_1$ in fig. 3.16 (a) and (d) in both results. This goes along with the interpretation of estimating a first order Taylor expansion. Due to the initialisation, it can be expected that the first component process relates to the first term in (3.25) and the corresponding coefficient is just 1, i.e. the coefficient is constant. The weights $\mathbf{W}_1$ appear to be almost constant at a high value compared to the other $\mathbf{W}_k$ in both decomposition results. This further indicates that the first order Taylor expansion is appropriate, because if the first order term is already large compared to the linear terms, then the non-linearities should be negligible. The component $\mathbf{\Theta}_1$ can also be seen as sort of "base line" to which all other components are added up.

K=3 The $K = 3$ decomposition in fig. 3.16 (a)-(c) was initialised with the partial derivatives $\frac{\partial}{\partial T_0}T$ and $\frac{\partial}{\partial \alpha}T$ and, indeed, $\mathbf{\Theta}_2$ in (b) and $\mathbf{\Theta}_3$ in (c) resemble the curves shown in fig. 3.10. This suggests that the components extracted by NMF contain information about the physical quantities in their corresponding weights because of the connection between the NMF weights and the physical parameters shown in (3.25).

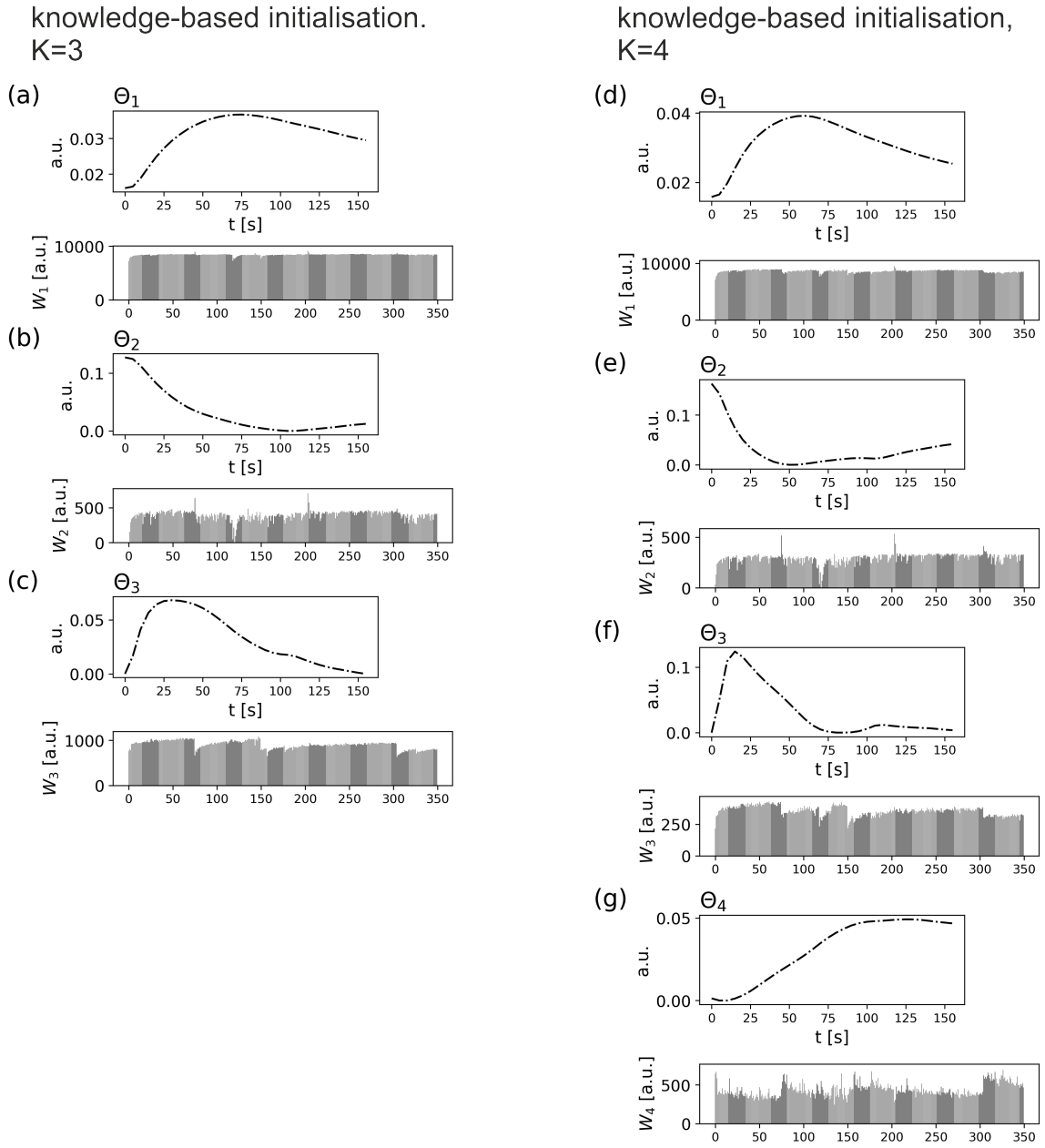

Fig. 3.16 Dataset 1: The NMF decomposition with knowledge-based initialisation with $K = 3$ and $K = 4$ components. The extracted component processes resemble the shape of the partial derivatives we obtain from the Taylor expansion.

$\boldsymbol{\Theta}_2$ in fig. 3.16 (b) should yield information about the initial temperature of the steel mould at the beginning of the casting process due to its initialisation with $\frac{\partial}{\partial T_0} T$. This assertion is strengthened by the two significant peaks in $\mathbf{W}_2$, which happen right after a long interruption during the manufacturing process. In such cases, the cavity is be pre-heated with flame burners to prepare the cavity for the casting process and to speed up the ramp-up time.

This then results in a higher than usual temperature at the cavity's surface.
The most interesting result in this decomposition is the obtained component process $\mathbf{\Theta}_3$ in fig. 3.16 (c), which can be related to a changing heat transfer coefficient during the series production. A variation in the heat transfer coefficient, i.e. a change in the rate of heat flow and thus consequently a different solidification process, is a critical parameter for the final product quality. Component $\mathbf{\Theta}_3$ is obtained by initialising with $\frac{\partial}{\partial \alpha} T$ and the corresponding weights $\mathbf{W}_3$ are therefore interrelated with the heat transfer coefficient of the system alloy, release agent and steel mould. This relationship is schematically illustrated in fig. 3.17. Typically, the release agent layer decreases the heat transfer coefficient, i.e. it has an isolation effect and lowers the rate of heat transfer. If the thickness d_1 during one process is smaller than during another process with d_2, then the rate of heat flow $\dot{Q}_1 > \dot{Q}_2$ and consequently the recorded sensor signals will differ in their shape. The difference in their curvature should be approximately describable by the partial derivative $\frac{\partial}{\partial \alpha} T$, if the change in the heat transfer coefficient is small enough.

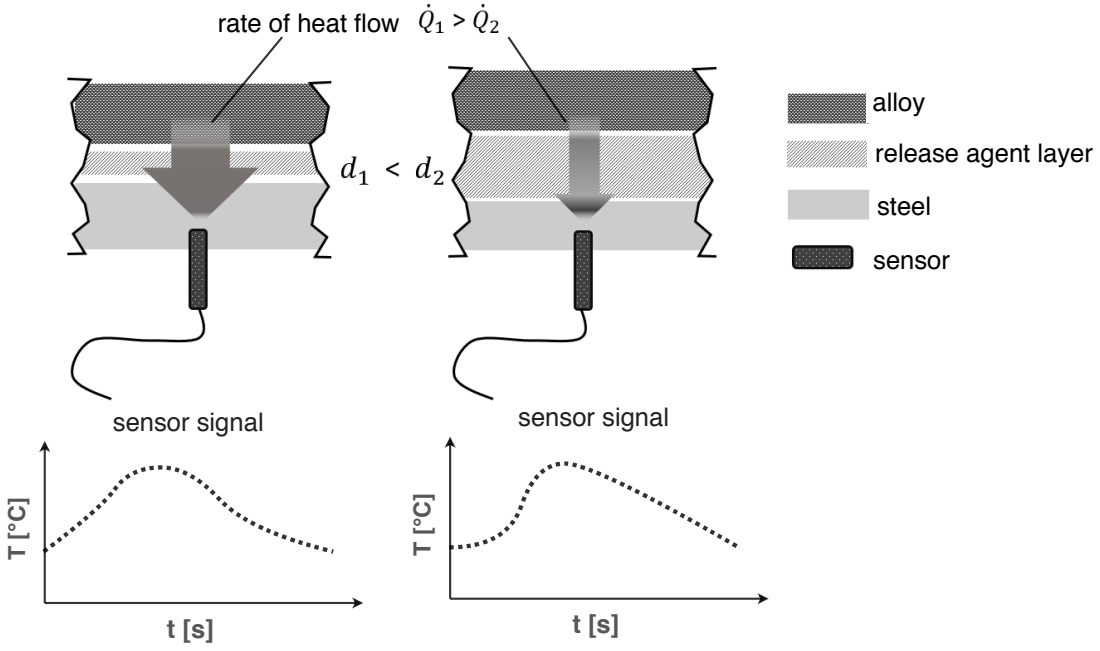

Fig. 3.17 Schematic illustration of how a different layer of release agent influences the temperature signal. A thin layer (left) causes a higher rate of heat flow than a thicker layer (right). The shape of the recorded sensor signal changes in a specific way depending on the layer thickness.

Fig. 3.18 shows how the release agent is applied in between the casting processes. This reapplication is done regularly, because the layer is gradually removed during consecutive production, which will result in the production of defective parts if the layer becomes too thin. The component process $\mathbf{\Theta}_3$ in 3.16 (c) captures these effects in its corresponding weights $\mathbf{W}_3$. Fig. 3.19 shows the same $\mathbf{\Theta}_3$ (from Fig. 3.16) in a larger image. Here, the additionally added black arrows mark the events at which a coating is manually reapplied in a similar fashion as shown in fig. 3.18. These events result in a sudden drop in the weights, i.e. a sudden decrease of the heat transfer coefficient due to the increased layer thickness. Furthermore, in between these sudden drops, the weights $\mathbf{\Theta}_3$ gradually increase, which comes from the gradual removal of the release agent layer and the consequent increase of the heat transfer coefficient α. If we observe how $\mathbf{\Theta}_3$ affects the sensor signal, namely that a higher contribution results in a faster rising temperature during the filling phase and also a faster decrease during the solidification phase, the relation to the heat transfer coefficient becomes clear. The ability to extract this information from temperature recordings offers a variety of potential applications, which is going to be discussed at the end of the third chapter.

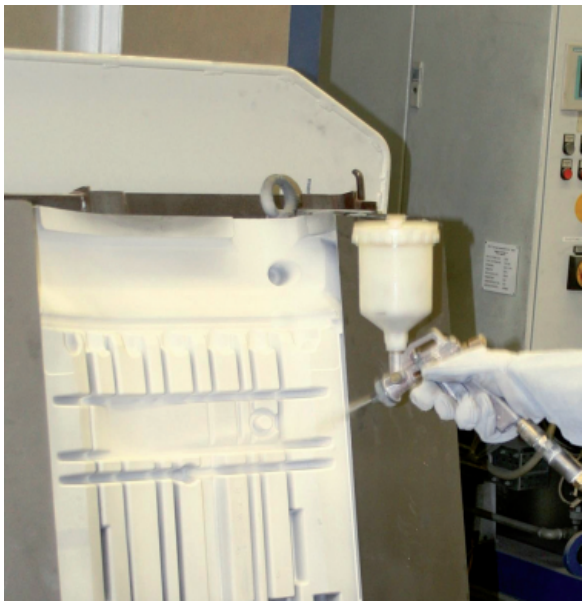

Fig. 3.18 The image shows how the release agent is applied to a casting cavity. During the running production the layer has to be reapplied regularly (image is taken from [33]).

K=4 The $K = 4$ decomposition in fig. 3.16 (d)-(g) is obtained with $\mathbf{\Theta}_{init,2}$ from (3.32) and thus has one more component which is initialised with $\frac{\partial}{\partial T_S}T$. In comparison with the result on the left-hand side in fig. 3.16, we can see that the first three components are similar in their shape and that also the weights have a similar structure. $\mathbf{W}_1$ and $\mathbf{W}_2$ in (d) and (e)

are comparable in their magnitude, but $\mathbf{W}_3$ in (f) has lower weights than $\mathbf{W}_3$ in (c). The reason for this is that (c) was split into (f) and (g), which might indicate that with $K = 3$, the decomposition estimates mixture components to some degree. Also $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$ have a slightly different curvature. $\mathbf{\Theta}_4$ resembles the function for $\frac{\partial}{\partial T_S}T$ in fig. 3.10, which is a simple exponential function. $\mathbf{W}_4$ has sudden steps in its weights similar to $\mathbf{W}_3$. The steps sometimes coincide with the steps in $\mathbf{W}_3$, which has process-related reasons. The oven, from which the liquid metal is delivered to the machine, is refilled after a certain number of production cycles. The time period of the oven refilling is often used to reapply the release agent. The oven temperature is slightly lower than the temperature of the recently melted metal which is delivered to the oven.

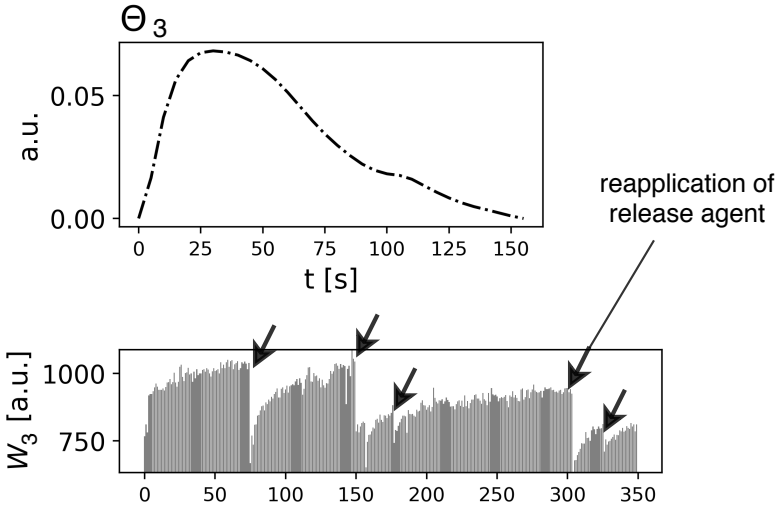

Fig. 3.19 The component process $\mathbf{\Theta}_3$. The arrows mark the event at which the release agent is manually reapplied.

K=4 For comparison, the dataset from the simple process $\mathbf{T_1}$ is also decomposed with the data-driven NNDSVD initialisation algorithm based on the SVD. The results are depicted in the same format in fig. 3.20, with the K=3 decomposition shown in (a)-(c) and the K=4 decomposition shown in (d)-(g).
Similar to the knowledge-based initialisation, the first components $\mathbf{\Theta}_1$ resemble the mean of all the curves in the dataset and also the corresponding weights are almost constant at a large value compared to the other components. The reason is that the first eigenvector estimated by the SVD is also the one with the largest eigenvalue and as such makes the highest contribution in the reconstruction of the original dataset. Using the interpretation given by the relation in (3.25), the highest contribution comes from the first term in the expansion. So

this result stems from an inherent property of the SVD. The rest of the components lacks the interpretable nature of the ones shown in fig. 3.16. While $\mathbf{\Theta}_2$ resembles the exponential decay function, its weights $\mathbf{W}_2$ show stepwise jumps, which are also present in $\mathbf{W}_3$. $\mathbf{\Theta}_3$ appears similar to $\mathbf{\Theta}_2$ in its curvature, yet the weights $\mathbf{\Theta}_3$ appear to be reversed. Since the steps in these components seem to correlate and cannot be explained by process-related reasons, it is likely that no specific physical quantities are captured in these component processes. In fig. 3.20 (d)-(g), the K=4 decomposition is shown. The component process $\mathbf{\Theta}_2$ remains almost unchanged. $\mathbf{\Theta}_3$ in (c) is now split up into $\mathbf{\Theta}_3$ in (f) and $\mathbf{\Theta}_4$ in (g). The stepwise nature is now more dominant in $\mathbf{W}_3$ and the curvature of $\mathbf{\Theta}_3$ seems to resemble an inverse form of the one obtained with the knowledge-based initialisation in fig. 3.16 (c). However, it is not clear how to interpret each individual component as mostly thy have similar curvatures and similar features in their corresponding weights.

This result shows that the NNDSVD initialised result does not yield the same level of interpretability as the knowledge-based initialised NMF decomposition. This is expected as the NNDSVD is a purely data-driven technique derived from linear algebra and as such no physical information can be incorporated in this initialisation strategy. Still, the inherent feature of the SVD to place the dominant component in first position is a useful property, which can be exploited in a situation where the physical model is more complicated to discover than the inherent structures of the recorded dataset. Another useful property of this approach is that the obtained components are ordered according to the magnitude of their contribution, as can be seen in fig. 3.20. Here, the components are ordered in such a way that mostly $(\mathbf{W}_i)_j > (\mathbf{W}_{i+1})_j$, where $(\mathbf{W}_i)_j$ is one entry in the weights $\mathbf{W}_i$. With this property, it can be possible to filter out very small contributions or discover strong inherent structures.

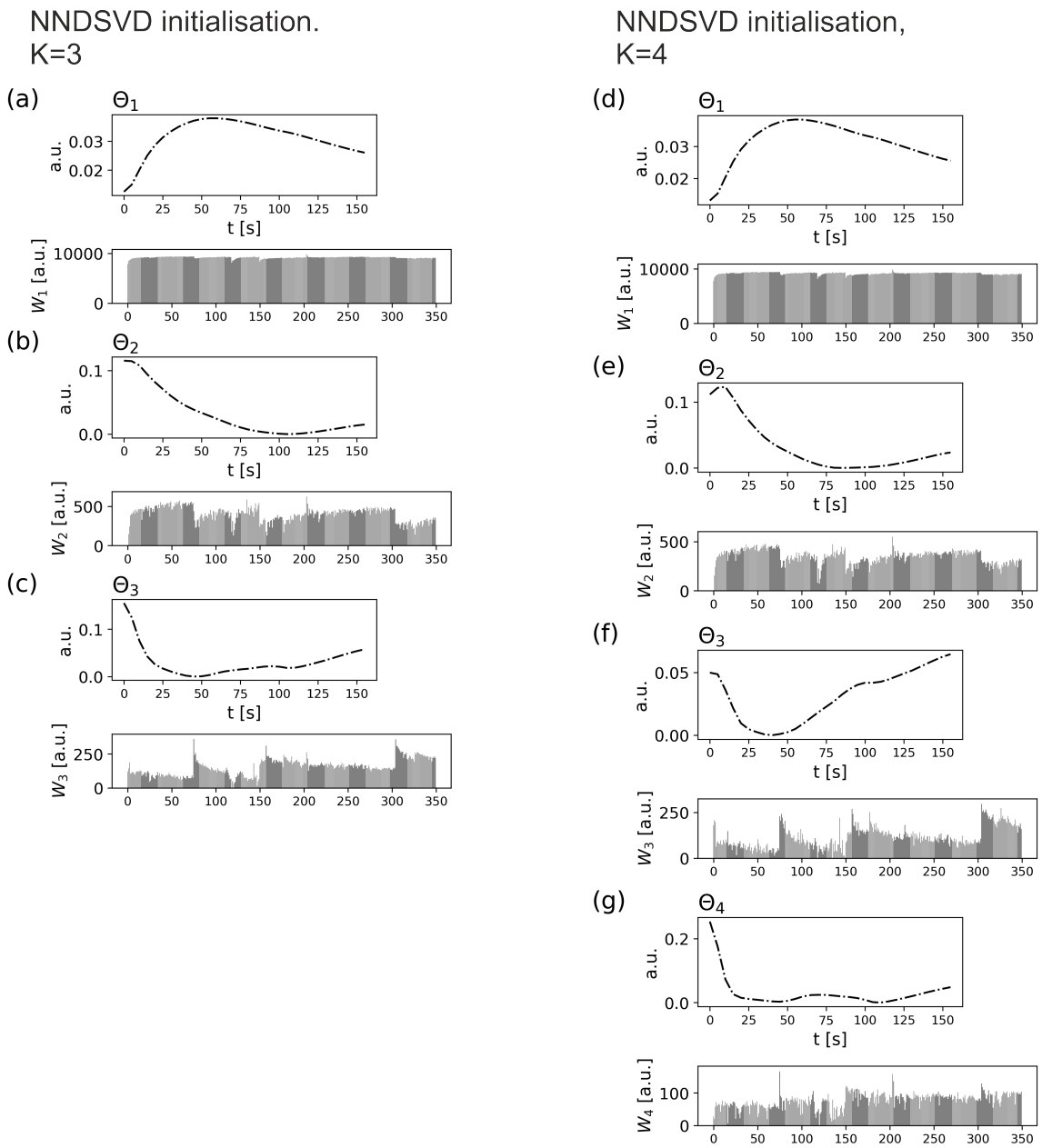

Fig. 3.20 Dataset 1: The NMF decomposition with NNDSVD initialisation with $K = 3$ and $K = 4$ components. The extracted component processes lack interpretability.

3.3.2.2 Results for Dataset 2

The knowledge-based decomposition using $\mathbf{\Theta}_{init,1}$ and $\mathbf{\Theta}_{init,2}$ from (3.32) is repeated in this section for the more complex process shown in fig. 3.15 (b). $\mathbf{T}_2$ is decomposed respectively with this initialisation in $K = 3$ and $K = 4$ components processes. The normalisation procedure and the stopping criteria is the same as for the $\mathbf{T}_1$ decomposition. An NNDSVD-based decomposition of $\mathbf{T}_2$ is shown and discussed in the appendix of this thesis (see fig. A.3 in the appendix).

Decomposing $\mathbf{T}_2$ with $\mathbf{\Theta}_{init,1}$ yields the component processes shown in fig. 3.21 (a)-(c) and decomposing with $\mathbf{\Theta}_{init,1}$ yields the component processes shown in (d)-(g). As before, the first component $\mathbf{\Theta}_1$ in (a) and (d) resembles the general curvature of the signals in $\mathbf{T}_2$ and can be interpreted as the first constant term in the Taylor expansion. Also the corresponding weights $\mathbf{W}_1$ are nearly constant at large values compared to the other components.

In the $K = 3$ decomposition in fig. 3.21 (a)-(c) the component processes $\mathbf{\Theta}_2$ and $\mathbf{\Theta}_3$ roughly resemble the functions for $\frac{\partial}{\partial T_0}T$ and $\frac{\partial}{\partial \alpha}T$ shown in fig. 3.10. In $\mathbf{\Theta}_3$, the second peak at around 100 seconds is still present. The significant peaks in $\mathbf{W}_2$, which come from the production interruptions and the step-wise curvature in $\mathbf{W}_3$ stemming from the reapplication and gradual removal of the release agent, are also present. As already mentioned, this dataset was recorded from the same processes but with a sensor at different position, thus the interpretation and reasons for the features in $\mathbf{W}_2$ and $\mathbf{W}_3$ are the same as for the decompositions in the previous section.

Studying the results of the decomposition initialised with $\mathbf{\Theta}_{init,2}$, in fig. 3.21 (d)-(g), it can be seen that the component processes $\mathbf{\Theta}_2$ and $\mathbf{\Theta}_3$ resemble the partial derivatives $\frac{\partial}{\partial T_0}T$ and $\frac{\partial}{\partial \alpha}T$ even more and they appear to be more smooth in their curvature. The second peak in $\mathbf{\Theta}_3$ is only slightly present. The component process $\mathbf{\Theta}_4$ in (g) can be roughly approximated with the exponential function shown in fig. 3.10 (d). Yet the weights are similar to ones obtained in fig. 3.16 (g) and the interpretation is the same as these recordings were taken during the same production cycle, i.e. the temperature of the same liquid metal was measured with this sensor.

In summary, even for this more complex process with an additional cooling effect, which is switched on and off during a production cycle, the proposed NMF-based approach can extract physically interpretable results. The reason for this lies in the fact, that the first NMF component apparently "absorbs" most of the additional cooling channels effect on the temperature signal. Note that for the initialisation, the very simple physical model given in (3.15) is still employed, which is a strong oversimplification of all the ongoing physical mechanisms during a metal casting process.

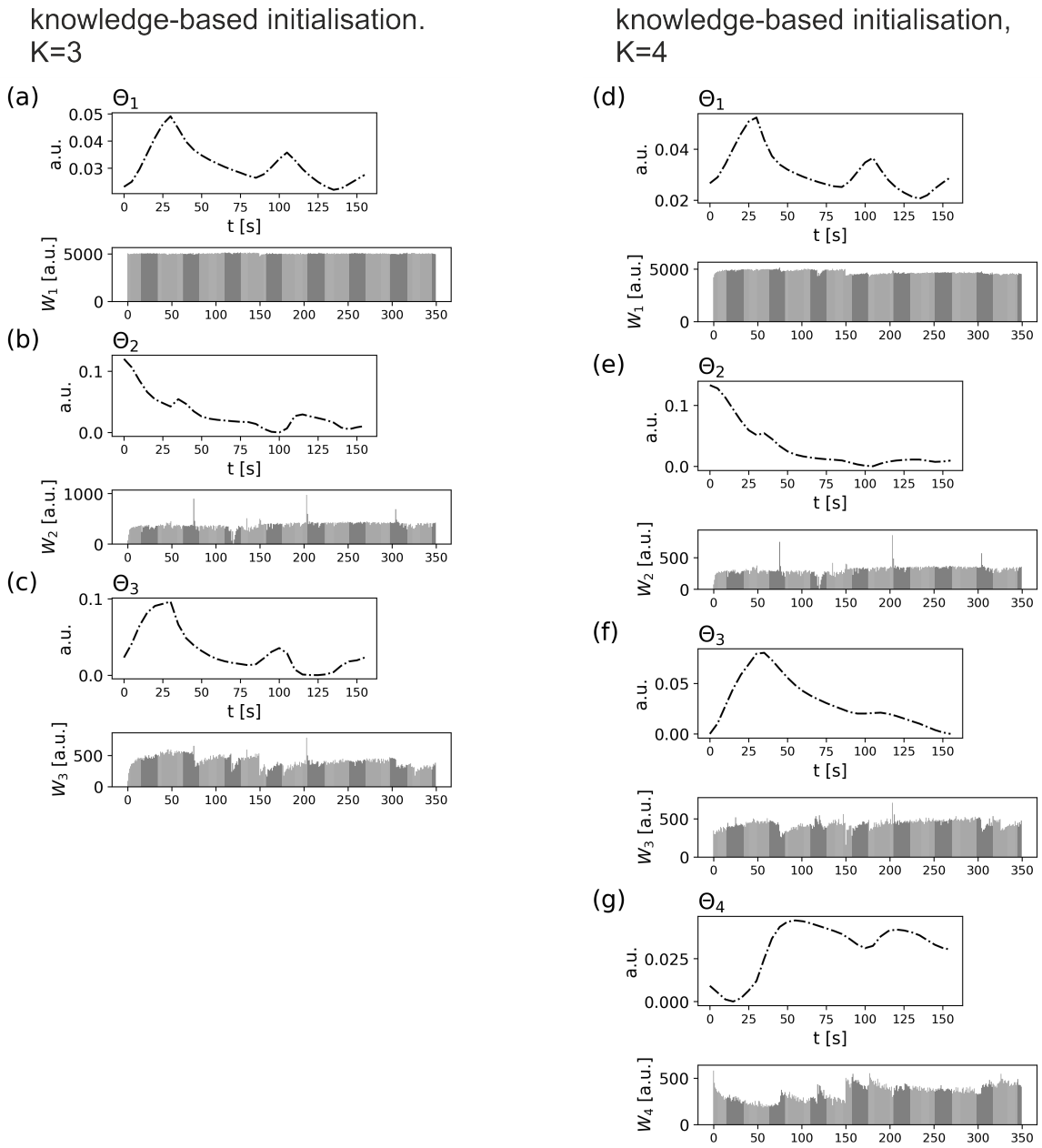

Fig. 3.21 Dataset 2: The NMF decomposition with knowledge-based initialisation with $K = 3$ and $K = 4$ components. With $K = 4$, the extracted component processes resemble the shape of the partial derivatives one obtains from the Taylor expansion.

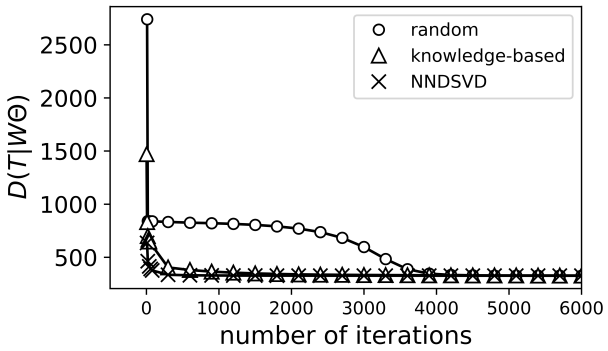

Fig. 3.22 Comparison of convergence speed with different initialisation strategies. The reconstruction error is the value of the cost function after each iteration step. Knowledge-based initialised NMF and NNDSVD initialised NMF converge after a few iterations. Random initialisation needs significantly more iteration steps to achieve a similar error.

3.3.2.3 Study on Convergence Speed

Algorithmic optimisations are not in the main focus of this thesis because the studies are mainly about the applicability of the approach to real-world data and the potential uses of the techniques. Yet a comparison of convergence speed of the NMF algorithm initialised with different initialisation techniques is shown here as this is an area where future research could be directed at. Fig. 3.22 shows the reconstruction error (the cost function in 2.12) plotted over the number of iterations for NMF decompositions initialised with NNDSVD, the proposed knowledge-based initialisation and a plain random initialisation. The dataset used to generate this plot is the dataset from the simple process discussed in section 3.3.1.1. First thing to notice is that, as expected, the randomly initialised NMF run is stuck searching in solution space for roughly 3000 iterations until it starts to converge. Combined with the fact that the results are different ever ytime, this approach is almost never advisable. The NNDSVD and knowledge-based approach converge much faster in less than 1000 iterations. The NNDSVD converges slightly faster than the knowledge-based approach, which is expected as the SVD is designed to yield the best low-rank approximation possible. Both approaches have the advantage that the end result is fixed, so multiple runs of the algorithm are not necessary. NNDSVD is purely data-driven, which has advantages in situations where no prior knowledge about the data exists. Instead, the knowledge-based approach allows for the incorporation of prior knowledge or expert domain knowledge.

3.3.3 Remarks about the NMF-based Decomposition Approach

So far, it has been demonstrated that this NMF-based approach is useful in analysing temperature profiles measured by a temperature sensor and generated by a thermal manufacturing process. An arrangement of multiple time series in a data matrix can be decomposed into physically meaningful features, which can be associated with physical thermal quantities that vary during the production process. This could be proven by relating the extracted component processes to specific process-related events in close collaboration with the process experts from the manufacturing company. In the experiments performed for this thesis, the ability of NMF to extract information about different physical quantities from a temperature time series is shown, but with the generalisation of the approach given in (3.25), this methodology could potentially be applicable to other kinds of time series. If the process-related variations are small enough to approximate the signals with a linear Taylor expansion, the decomposition into two matrices of the dataset can be possible. Furthermore, the connection between NMF and physical quantities through the expansion in (3.25) has to the best of my knowledge not been noticed yet by the research community and opens up the possibility for more extensive research both in the algorithmic domain or the application domain. Further, it has been shown that a simplified physical model as initialisation is enough to steer the optimisation towards the desired solution.

The work in this thesis then joins in with recent studies presenting the promise in the idea of combining machine learning techniques with physical knowledge [84, 113, 91]. The general approach is to incorporate structured information into a learning algorithm, which results in amplifying the information content of the data that the algorithm sees, enabling it to quickly steer itself towards a physically meaningful and interpretable solution and to generalise well even when only a few training examples are available.

The initialisation chosen in (3.10) is an empirical result obtained during the experiments. Which partial derivatives and how many components to choose, remains a question of further research. Also the parametrisation of the initialisation was found empirically, i.e. the scaling and coefficients of the exponential functions. From the results shown so far, it appears like a simple physical model can be enough to construct a suitable NMF initialisation, but this does not necessarily have to be the case in other application domains. The process analysed in this thesis allows the derivation of a simplified analytical solution from the heat equation. In more complicated scenarios, one might have to use sophisticated simulation tools to derive functions for the NMF initialisation. While this would definitely be more time-consuming, this might be a potential route to connect the use of simulation tools with machine learning methods.

3.3.4 The Effect of Regularisation

In this section, I am going to study the effect of regularisation on the NMF result. I am going to use the dataset described in section 3.3.1.1 to discuss the effect of L_1- and L_2-type regularisation with different strengths in different combinations. If both regularisation types are equally applied to both factorisation matrices the NMF cost function with regularisation extension is as follows:

$$\frac{1}{2}||\mathbf{T}-\mathbf{\Theta W}||_2^2+\gamma_1||\mathbf{W}||_1+\gamma_1||\mathbf{\Theta}||_1+\gamma_2||\mathbf{W}||_2^2+\gamma_2||\mathbf{\Theta}||_2^2 \tag{3.35}$$

γ_1 and γ_2 respectively control the magnitude of L_1- and L_2-type regularisation. Here, the values for γ_1 and γ_2 are set to 0,1,10 and 50 and the data matrix $\mathbf{T}$ is decomposed. Note that for the experiments with L_1-type regularisation, γ_2 is set to zero for all values and vice versa to observe the effect of each regularisation type individually. It is also possible to mix the two types.

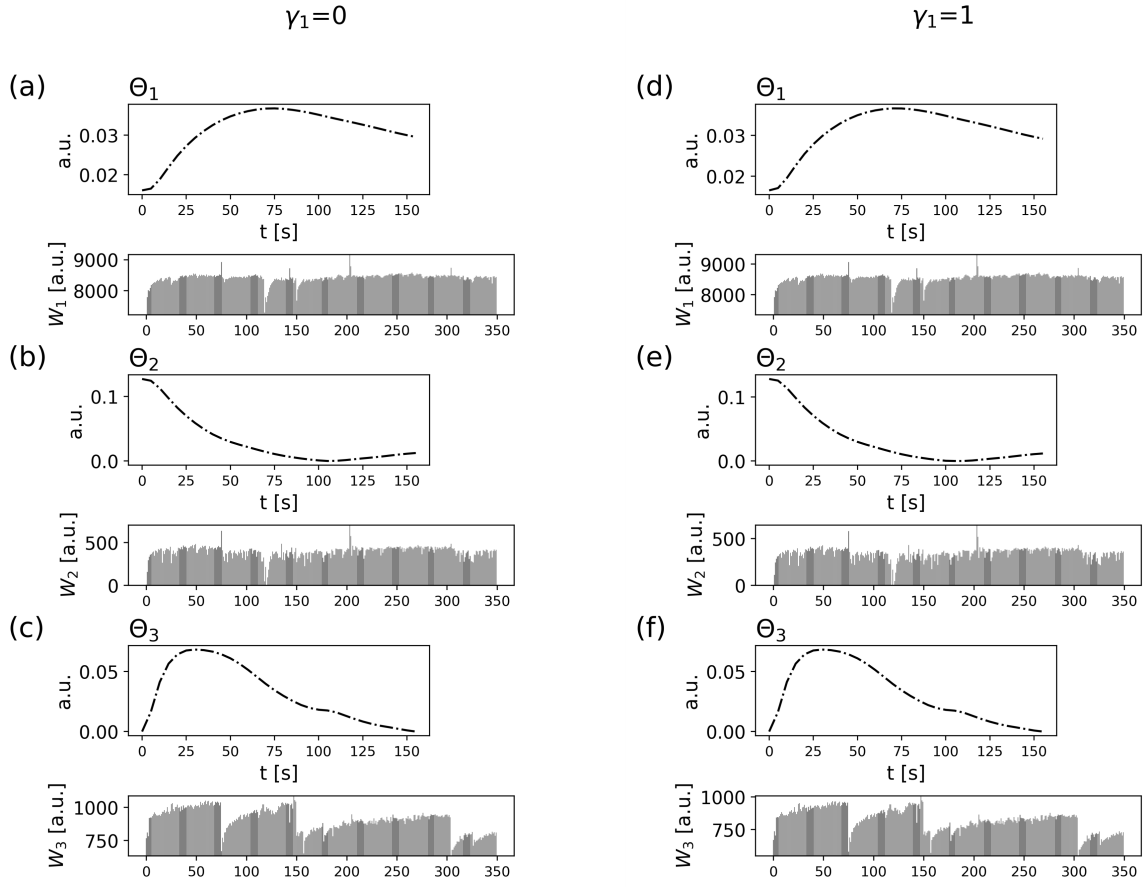

Fig. 3.23 L_1-regularisation with γ_1=0 and γ_1=1 for both matrices.

The main goal of regularisation is to reduce the solution space for an optimisation problem to avoid too large values or to decrease the influence of outliers and random noise if the model is well conditioned. In most publications about the application of NMF, regularisation is recommended to obtain interpretable and stable solutions and to avoid overfitting the training data. In the previous chapter I have already demonstrated that NMF with HALS optimisation is able to extract interpretable and stable solutions from process data collected from a casting process. In all of the experiments performed during the course of this thesis, regularisation was not needed to achieve stable solutions, yet the effect of the most common regularisation types L_1- and L_2-regularisation in this domain is of theoretical interest.

Fig. 3.23 shows the decomposition into $K = 3$ components without regularisation ($\gamma_1 = 0$) on the left-hand side and a decomposition with $\gamma_1 = 1$ on the right-hand side. A L_1-type regularisation with this magnitude does not seem to have any impact at all on the decomposition result. The shape and ordering of the extracted component processes and the weights in $\mathbf{W}$ are almost identical. Also the strong structure extracted in $\mathbf{W_3}$ is still present.

By increasing the magnitude of the γ_1 to 10, the decomposition result changes. The left-hand side of fig. 3.24 (a-c) shows the decomposition for $\gamma_1 = 10$. First thing to notice is that the order of the components has changed. $\mathbf{\Theta_3}$ from the $\gamma_1 = 1$ decomposition is now $\mathbf{\Theta_2}$. The component that resembled an exponential decay has vanished and $\mathbf{\Theta_3}$ instead shows a component that resembles an exponential rise. $\mathbf{\Theta_2}$ still shows the periodic structure but in a slightly attenuated form. As already mentioned, the introduction of a regularisation term changes the solution space of the NMF model. With L_1-type regularisation, the component process, which resembled an exponential decay seems to be less dominant in the solution space than in the unconstrained solution space. As the HALS algorithm tends to order the extracted components by their individual magnitude of contribution, the L_1-type regularisation also causes a reordering of the components.

In fig. 3.24 (d-f), the decomposition with $\gamma_1 = 50$ is shown. It can be seen that the shape of the component processes from $\gamma_1 = 10$ to $\gamma_1 = 50$ changed only slightly. The magnitude of the coefficients $\mathbf{W_1}$ and $\mathbf{W_2}$ has not changed, but $\mathbf{W_3}$ shows a typical result obtained by applying L_1-type regularisation, called sparseness. As already explained in the theoretical chapter of this thesis, sparseness is a property which describes a data vector or matrix, which mostly contains entries with value zero. In many NMF applications, sparseness is a desired property, because it can lead to highly interpretable components if it the unknown sources are sparse in their activations. In our case, the emerging sparseness property might be artificial as the same component process can already be extracted with a lower value for the regularisation parameter γ_1.

Compared to L_1-type regularisation, the L_2-norm constrained cost function should invoke

smoothness and bounded solutions. Fig. 3.25 shows the decomposition into $K = 3$ components without regularisation ($\gamma_2 = 0$) on the left-hand side (a-c) and a decomposition with $\gamma_2 = 1$ and the right-hand side (d-f). As before with $\gamma = 1$, the resulting decomposition extracts similar component processes $\boldsymbol{\Theta}$ and coefficients $\mathbf{W}$.

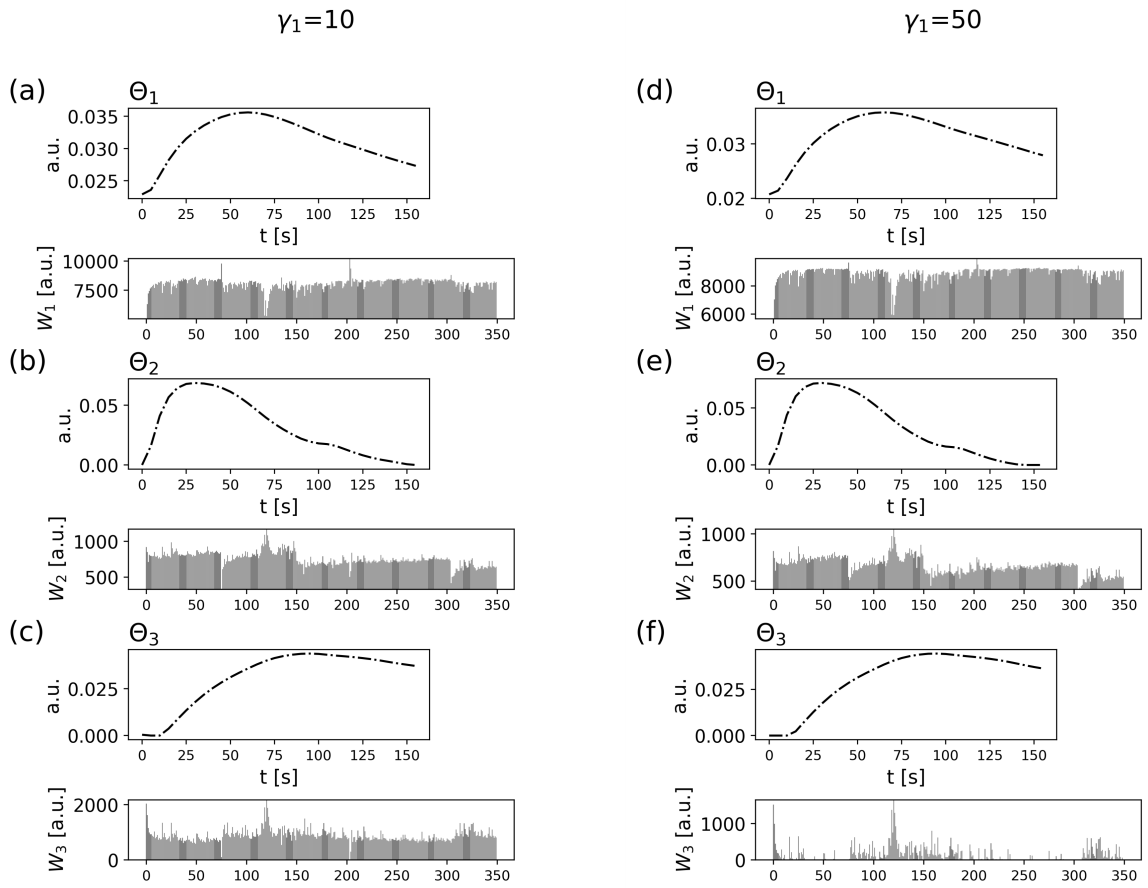

Fig. 3.24 L_1-regularisation with γ_1=10 and γ_1=50 for both matrices.

In fig. 3.26 (a-c), γ_2 is set to 10 and the effects of the regularisation are now visible. $\boldsymbol{\Theta_2}$ has changed its shape and now does not resemble the exponential decay function. Additionally, the weights $\mathbf{W_2}$ have increased. Similarly, $\boldsymbol{\Theta_3}$ now appears lifted upwards at 125 to 155 seconds and also its corresponding weights $\mathbf{W_3}$ are increased compared to $\gamma_2 = 0$ or $\gamma_2 = 1$. $\mathbf{W_1}$ now shows more structure and the scale of its corresponding weights has decreased and is now closer to the scale of $\mathbf{W_2}$ and $\mathbf{W_3}$. This equalising property in scales is typical for a L_2-type regularisation. Fig. 3.25 (d-f) shows the decomposition with $\gamma_2 = 50$. The component processes $\boldsymbol{\Theta_2}$ and $\boldsymbol{\Theta_3}$ have only slightly changed, with $\boldsymbol{\Theta_2}$ being lifted up at around 25 to 50 seconds and $\boldsymbol{\Theta_3}$ at around 125 to 155 seconds. It can be seen that the effect of the L_2-type regularisation does not differ in a significant way compared to $\gamma_2 = 10$.

One application domain of NMF is topic modelling [107]. The goal in topic modelling is to retrieve a clustering of the input data by similarity. If a matrix factorisation technique is used, the algorithm should retrieve components with sparse activations and the components should be interpretable, i.e. the solution space should be bounded. The component processes in our case are also time series and enforcing sparsity onto them via an L_1-type regularisation would reduce the interpretability, because they would be forced to be zero-valued if the regularisation parameter is large enough. L_2-type regularisation instead bounds the solution and tends to result in smooth components, which is a desired property for time curves that ought to describe a physical process.

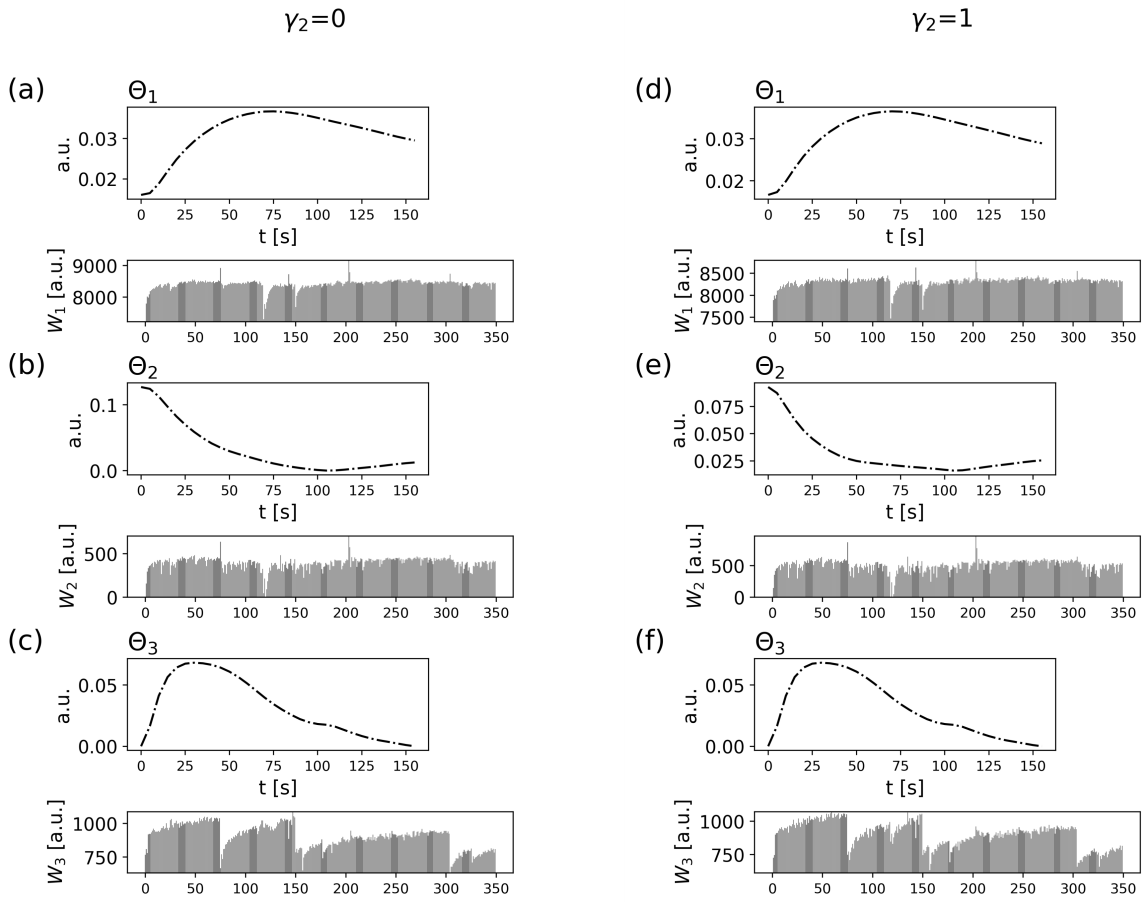

Fig. 3.25 L_2-regularisation with γ_2=0 and γ_2=1 for both matrices.

Combining L_2-regularisation for $\mathbf{\Theta}$ and L_1-regularisation for $\mathbf{W}$, we change the notation for the regularisation terms in the cost function like this:

$$\frac{1}{2}||\mathbf{T}-\mathbf{\Theta W}||_2^2+\gamma_{W,1}||\mathbf{W}||_1+\gamma_{\Theta,2}||\mathbf{\Theta}||_2^2 \tag{3.36}$$

As before, the regularisation parameters are are set to $\gamma_{W,1}, \gamma_{\Theta,2} = 0, 1, 10$ and 50 and the results are respectively shown in fig. 3.27 and 3.28. In fig. 3.27 (a-c) the results without regularisation can be compared with the results obtained by setting the regularisation parameters to 1 and it can be seen that neither the extracted component processes nor the corresponding weights have changed. This result strengthens the assertion that for this kind of data, a slight regularisation constraint has no effect on the decomposition.

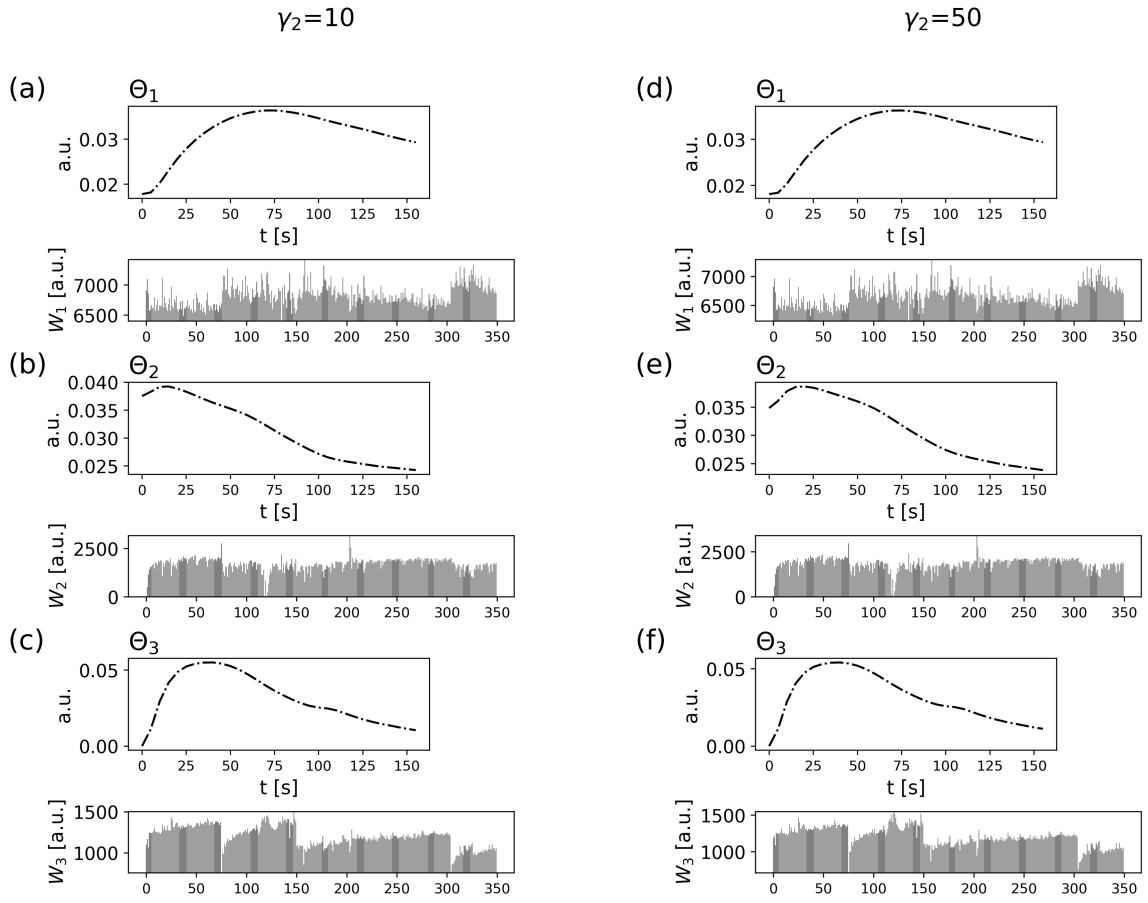

Fig. 3.26 L_2-regularisation with γ_2=10 and γ_2=50 for both matrices.

Fig. 3.28 (a-c) shows the $K = 3$ decomposition with $\gamma_{W,1}, \gamma_{W,1} = 10$. The obtained component processes are similar to fig. 3.24 (a-c), where both matrices are subjected to L_1-regularisation. Again, the ordering of the components has changed according to their magnitude of contribution and instead of the component process, which resembles an exponential decay, we obtain the one in fig. 3.28 (c), which resembles a typical heating process. Furthermore, it can be seen that W_3 shows the expected sparsity property. By further increasing the regularisation, the weights in W_3 are even more sparse, as can be seen in fig. 3.28 (c). Comparing the results in fig. 3.24 with the ones shown in fig. 3.28, it appears as if it

makes no difference to apply L_1-regularisation onto both matrices or only on $\mathbf{W}$ because the obtained results are almost similar, with the only difference being that the sparsity property appears more distinct already with $\gamma_{W,1} = 10$.

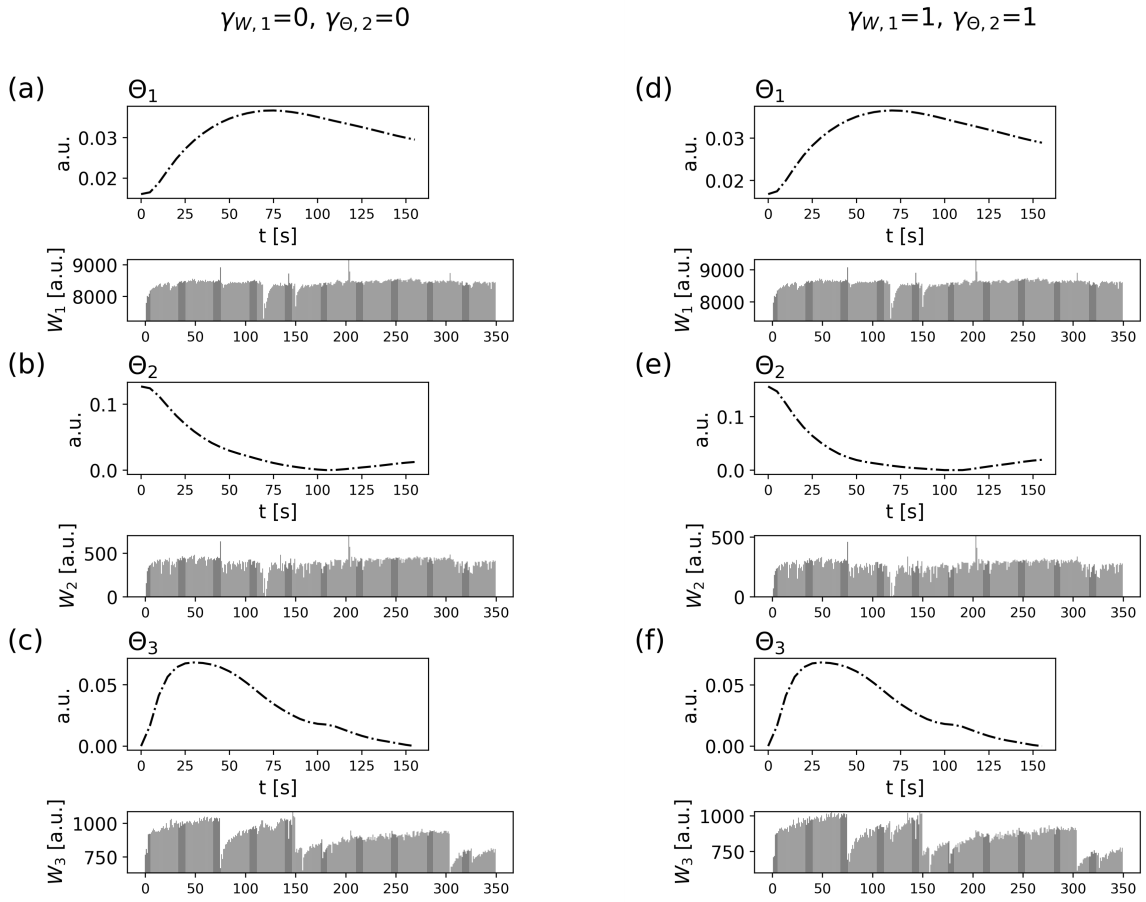

Fig. 3.27 L_1-regularisation for the weights ($\gamma_{W,1} = 0, 1$) and L_2-regularisation for the component processes ($\gamma_{\Theta,2} = 0, 1$).

In summary, it can be said that by adding regularisation terms to the cost function, the resulting component processes obtained from the decomposition can significantly change. One interesting takeaway from this study is that due to regularisation, the emerging component processes can drift away from the knowledge-based initialisation towards unexpected component processes. As can be seen in the case of L_1-regularisation, those new components appear to be also similar to functions emerging typically from thermal processes. One problem that remains, is the fact that the regularisation parameter γ as to be chosen manually. Usually one chooses the regularisation parameter by employing cross-validation or checking the result for overfitting. Yet in our case the scenario is more complex, because

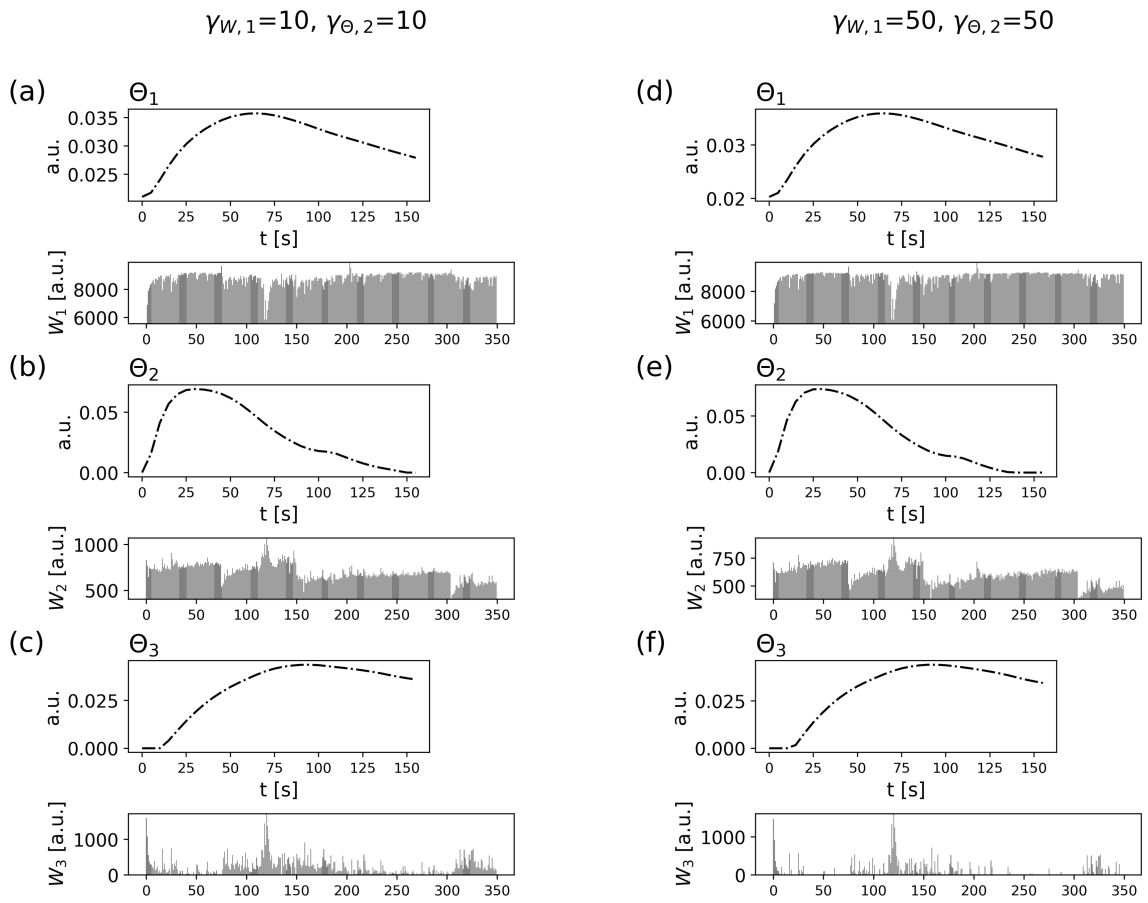

Fig. 3.28 L_1-regularisation for the weights ($\gamma_{W,1} = 10, 50$) and L_2-regularisation for the component processes $\gamma_{\Theta,2} = 10, 50$).

we are explicitly looking for interpretable components, which is not possible to describe in mathematical terms. Further, the properties obtained by regularisation (i.e. sparsity or smoothness) might not relate to any real ongoing hidden mechanism in the process, and thus by enforcing these properties we might produce artificial components without any physical meaning. The possibility to include regularisation terms into the cost function makes the NMF algorithms very flexible and adaptable to different problems, which is why further experiments to study the effects of regularisation beyond the standard penalty terms (L_1- and L_2-regularisation) can be a possible topic for further research.

3.4 An Application in Process Monitoring

3.4.1 A Data-driven Soft Sensor

Soft sensors are systems that consist of a software part ("soft") and an information delivery part (like hardware sensors). A soft sensor generates a new hardware-sensor-like signal, which can be used for analysing or monitoring different processes and employed in many application domains (see [102] for a recent review paper). In literature, the same concept is sometimes referred to as inferential sensor, virtual on-line analyser or observer-based sensor. Basically, a soft sensor indirectly measures a desired quantity by employing one or more other quantities. Such a system can be derived if the process is sufficiently described by a first principle model (FPM), which is subject to process knowledge and the experience of experts. This strategy has disadvantages as the acquisition of expert knowledge is difficult or time-consuming, especially for modern complex manufacturing processes. Due to the high amount of process data continuously generated during industrial manufacturing, the rendering of a data-driven soft sensor becomes a viable option.

In this section, the design of such a soft sensor is shown, which uses the NMF model from section 3.2.4. Component processes extracted by NMF allow for a representation of process data, which can readily be combined with a desired output quantity. The output quantity in this application is the amount of applied release agent on the steel cavities surface, which has a large influence on the casting product quality (see fig. 3.1). This is motivated by the fact that to this point there is no in-line measurement system for the release agent which does not disturb the process or prolong the process cycle time. The approach employs the combined NMF regression method described in section 2.2.1. The use of semi-supervised regression for the design of soft sensors in this application domain is rather new and to the best of my knowledge no related publications exist.

In other domains like chemical manufacturing, semi-supervised regression appears to be a new trend as many recent publications can be found. In chemical engineering, it is relatively easy to obtain input variables like temperature or pressure, but output variables might be much more difficult to obtain. In literature, the most used methods to implement soft sensors are principal component regression [59, 44, 56], partial least squares (PLS) [126, 123, 37, 82], artificial neural networks [73, 48, 7], kernel-based methods [79, 43, 58, 122, 118] or Bayesian methods [60, 57, 121, 61]. In this sense, NMF as basis for soft sensor design is also a new application for this algorithm.

In the following, the approach outlined in section () is used and I will explain the data generation process and the preprocessing of the input data for the model. Afterwards, I will

explain the training procedure used to train the virtual sensor model and discuss the results and show how the results can be implemented in running production.

3.4.2 Data Generation

3.4.2.1 Dependent Variable: Measurements of Layer Thickness

The dependent variable for the training are measurements **y** of the thickness of the release agent's layer (see fig. 3.18) taken during running production. For this sake, a magnetic induction thickness measurement device was used to measure the release agent's layer thickness at different positions. The measurement device uses a low frequency magnetic field generated by an excitation current. The strength of the magnetic field corresponds to the distance between the probe and the base material. By measuring the magnetic field with a measurement coil, the obtained measurement signal can be converted into a value for the coating thickness via a characteristic output function (i.e. the functional correlation between the probe signal and the coating thickness).
After one cylinder head is produced, there is a certain time period (roughly 30 seconds), where the cavity is empty and measurements can be taken. It is important to mention that these measurements are not taken in a controlled environment. Firstly, it is a highly dangerous procedure because the surface is still at high temperatures (the cavity surface still reaches 300 $^{\circ}C$) while one is reaching out for the measurement points. Secondly, the measurements have to be taken within a few seconds to not disturb the ongoing production. A picture showing the measurement procedure can be seen in fig. 3.29. The person taking the measurements is holding the magnetic induction probe in his left hand, while bending over the open cavity. In the experiments done for this thesis, a certain measurement position was chosen and five measurements were performed at the designated position during the ongoing production. The positions were selected according to the expert knowledge of the responsible process engineers. Due to the before mentioned circumstances present during production, thickness measurements in this domain come along with multiple disruptive effects, which cause a certain level of noise in the target variable:

- accidently tilting the measurement device;
- measurements not taken exactly at the designated measurement spot;
- measurements at the end of the time period become more difficult due to the heat;
- the magnetic measurement device heats up.

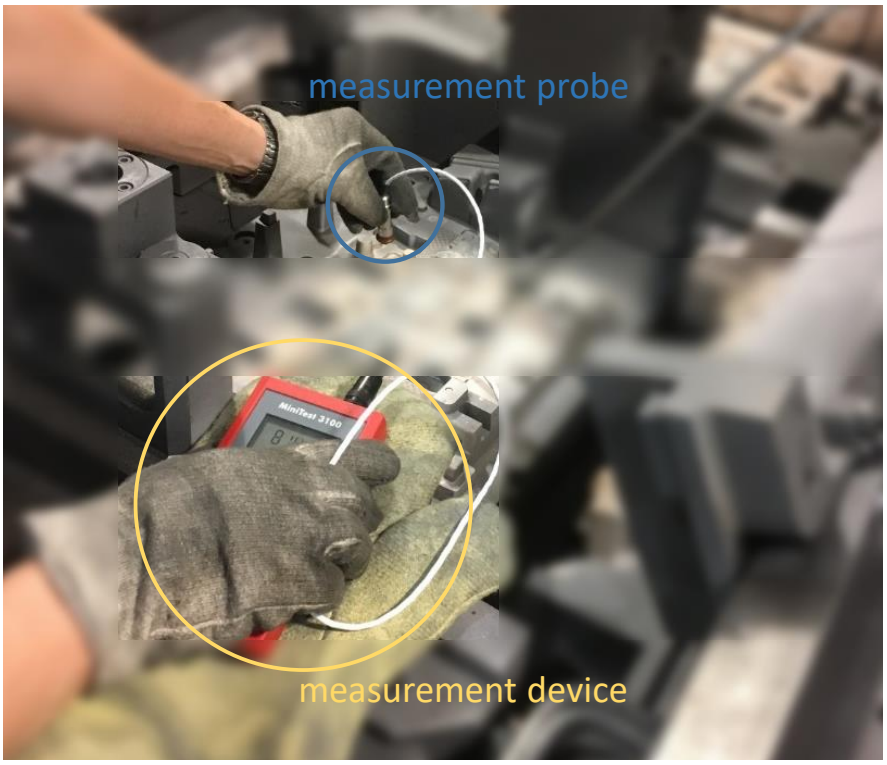

Fig. 3.29 Image taken during one measurement session. During a 30-second time window, multiple measurements at different positions were taken. The image is blurred to hide any confidential information.

Multiple measurements at different positions were taken to test the capability of the combined NMF regression approach described in section 2.2.1. In this section, I focus on one measurement dataset taken at a specific position, which lies close to its respective temperature sensor position. As mentioned, thickness measurements are affected by multiple noise-adding effects. For this reason, the measurement was repeated five times at the same position and then averaged. This way the effect of outliers to due incorrect measurements is less severe. par Table 3.1 shows the 47 measurements taken at this specific position. In the right-most column, the standard deviation is calculated for each measurement series. The maximum standard deviation for measurement section is 12.57 μm (series 15). This is due to an outlier which might be caused by accidentally tilting the probe. In series 19, one of the measurements is missing and could not be repeated because the next casting process was about to start.

Nr.	m_1	m_2	m_3	m_4	m_5	$\overline{m}$	σ
1	40.0	43.6	46.6	45.2	40.0	43.08	2.69
2	42.2	40.8	52.6	50.4	47.0	46.60	4.55
3	35.2	37.4	35.6	30.8	32.2	34.24	2.40
4	42.8	41.2	34.0	32.8	35.6	37.28	3.99
5	35.8	40.2	39.2	38.0	41.4	38.92	1.92
6	41.6	37.8	50.8	42.8	43.6	43.32	4.24
7	31.6	30.4	30.2	30.4	34.4	31.40	1.58
8	39.4	44.6	39.2	43.2	31.0	39.48	4.73
9	33.8	27.6	27.2	25.0	28.0	28.32	2.93
10	29.0	37.0	28.6	38.0	27.4	32.00	4.53
11	26.0	25.6	32.4	27.8	21.8	26.72	3.45
12	25.6	24.2	31.6	27.4	28.6	27.48	2.55
13	27.8	32.6	29.6	26.4	28.8	29.04	2.08
14	27.4	22.6	28.2	28.0	37.8	28.80	4.95
15	26.4	24.2	25.8	57.6	29.4	32.68	12.57
16	34.6	25.6	29.4	30.0	24.2	28.76	3.66
17	30.4	29.0	37.6	33.0	36.8	33.36	3.40
18	39.2	33.0	39.4	36.4	31.4	35.88	3.23
19	29.4	28.2	31.4	28.6	NaN	29.40	1.23
20	34.0	23.2	29.4	31.4	26.6	28.92	3.75
21	30.6	33.6	29.0	33.6	28.0	30.96	2.31
22	30.2	26.2	25.2	25.4	29.8	27.36	2.19
23	27.2	24.4	33.2	26.2	27.8	27.76	2.95
24	35.0	31.8	28.6	34.6	29.2	31.84	2.65
25	28.2	33.6	28.4	35.0	30.8	31.20	2.73
26	29.0	36.2	31.0	31.0	32.8	32.00	2.42
27	55.0	66.8	56.8	58.4	56.0	58.60	4.25
28	72.0	69.0	53.2	68.6	59.0	64.36	7.09
29	46.0	40.6	52.2	54.0	52.0	48.96	4.98
30	53.2	50.6	58.8	53.4	55.8	54.36	2.76
31	53.2	49.0	53.8	52.4	61.6	54.00	4.15
32	46.4	49.2	50.0	42.0	48.4	47.20	2.86
33	63.0	58.0	51.0	51.2	52.2	55.08	4.72
34	69.0	54.0	53.2	49.2	57.4	56.56	6.74
35	45.6	41.6	42.4	43.2	49.4	44.44	2.82
36	48.2	58.0	62.2	65.0	60.6	58.80	5.77
37	52.0	58.6	58.0	39.0	55.6	52.64	7.20
38	51.8	58.0	56.0	44.0	62.4	54.44	6.24
39	55.2	61.2	68.6	51.6	49.0	57.12	7.05
40	54.2	57.0	60.2	57.6	53.2	56.44	2.50
41	48.0	44.2	54.8	56.8	39.0	48.56	6.60
42	57.6	39.8	42.4	50.6	55.8	49.24	7.08
43	56.2	63.0	54.8	76.8	65.0	63.16	7.85
44	65.8	48.0	52.8	53.0	56.0	55.12	5.92
45	62.6	70.8	79.6	70.8	65.2	69.80	5.85
46	78.4	70.0	59.4	63.8	63.0	66.92	6.68
47	66.0	57.0	73.0	58.8	58.2	62.60	6.08

Table 3.1 The measurement data taken during running production. The right columns show the respective mean and standard deviation.

3.4.2.2 Independent Variables: NMF Component Processes

To each thickness measurement, we assign the temperature recording that was generated during the casting process before the thickness measurement was taken. In fig. 3.30, this procedure is illustrated.

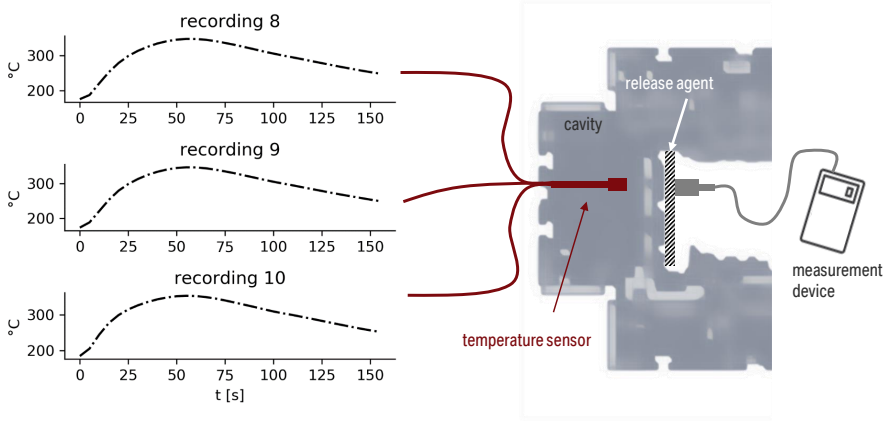

Fig. 3.30 Illustration of the measurement procedure.

A sensor in the steel cavity records a temperature curve during filling and solidification. When the process is done and the part is automatically removed from the cavity, the layer thickness is measured on the cavity's surface in an area close to the sensor position. This temperature curve and consecutive thickness measurement form a pair, which is used to train the model. For example, in fig. 3.30, the recordings 8, 9 and 10 are shown, which are respectively assigned to the measurement series 8, 9 and 10 in table 3.1.

Overall, the full dataset for the regression model training consists of 47 time curves with 32 time points, i.e. a matrix $\mathbf{T}_\mu \in \mathbb{R}^{47\times 32}$ and a target vector $\mathbf{y} \in \mathbb{R}^{47\times 1}$ containing the measured thickness values. Additionally, for the NMF model, another time series matrix $\mathbf{T}_{NMF} \in \mathbb{R}^{500\times 32}$ with 500 time curves recorded with the same sensor during running production is used. In fig. 3.31 on the left-hand side, these 500 temperature recordings are plotted together. In summary, my approach relies on a data set, which consists of a large number of unlabled time series and a smaller data set, which consists of time series with assigned measurement values.

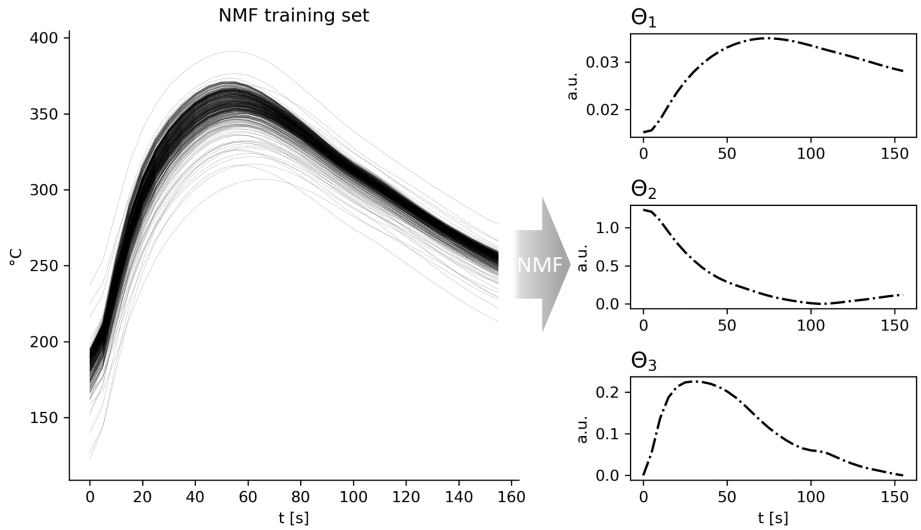

Fig. 3.31 Left: The training set for the NMF model. Right: The extracted component processes used for preprocessing.

3.4.3 Model Training

The model training employs the combined NMF and regression procedure described in section 2.2.1. The NMF model needs to be pretrained with a dataset with a larger sample-size than $\mathbf{T}_\mu$ because the latent factors cannot be modelled with only 47 time curves. If the sample-size is similar to the number of variables, the NMF decomposition will likely overfit the dataset. The temperature recordings are taken during the ongoing production and are readily available in the company's process-data database. Due to the high number of casting parts produced, the available number of temperature curves is more than 10^5, but the NMF result with this data becomes already stable with only a subset of 500 curves.

For the NMF model, an initialisation approach has to be chosen. This choice can be seen as another model parameter and any of the before mentioned initialisation techniques can be tried. Also the number of components K needs to be chosen. In this example, the knowledge-based initialisation strategy is used and the number of components is set to $K = 3$ components.

The right-hand side of fig. 3.31 shows the extracted latent component processes obtained from the NMF decomposition. With the NMF model trained with $\mathbf{T}_{NMF}$ the transformation functions $\hat{\boldsymbol{\Theta}}$ defined in (2.70) are constructed and the time series matrix $\mathbf{T}_\mu$ is transformed

into the reduced dimension space

$$\hat{\mathbf{\Theta}}(\mathbf{T}_\mu) = \mathbf{W}_\mu, \tag{3.37}$$

where $\mathbf{W}_\mu \in \mathbb{R}^{47\times 3}$ contains the corresponding loadings for each time series in the rows of $\mathbf{T}_\mu$. These loadings are now used as independent variables for the regression model and the regression model coefficients b_1, b_2, b_3 and b_0 are estimated by the optimisation procedure described in section 2.2.1.

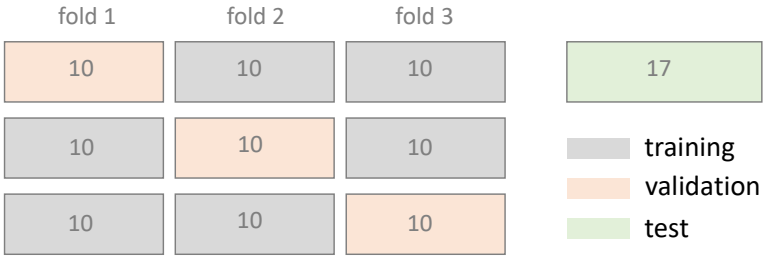

Fig. 3.32 Illustration of the training process with 3-fold cross-validation and the split into training, validation and test set. The sample number in each set is written in the boxes.

To evaluate the model error and performance, the whole dataset is split into two parts, one training set and one test set. The training data set contains 30 randomly sampled rows from the matrix $\mathbf{T}_\mu$ and their corresponding measured thickness value. The remaining 17 rows are used as test set, i.e. the regression equation is applied to their respective NMF coefficients and the output compared to their assigned measurement value.

	b_1	b_2	b_3		b_0		MSE
1	-0.03	0.01	-0.13	1	295.88	1	49.68
2	-0.02	0.01	-0.11	2	298.62	2	49.29
3	-0.02	0.03	-0.14	3	404.70	2	52.77
4	-0.03	0.15	-0.15	4	241.96	4	50.63
5	-0.03	0.00	-0.12	5	214.02	5	43.50
6	-0.01	-0.01	-0.09	6	273.87	6	46.23

Table 3.2 The regression equation coefficients, the bias variable and the mean squared error for six different training sessions.

This way, the model performance on new unseen data can be estimated. Furthermore, this split into training and test set is performed six times, with randomly sampled training and

test data sets. The reason for this is the small number of measurements, which might lead to unfavourable splits by random chance and thus to an overestimation or underestimation of the models performance.

In order to estimate the regularisation parameter λ, a procedure called n-fold cross-validation is employed. There, the training set is split in n subsets (folds) and the model is trained with changing λ with $(n-1)$ folds and is tested against the remaining fold (validation fold). This procedure is repeated n times, whereby each one of the n folds acts as a validation fold once. At the end, the model with λ value which has performed best on average on all n folds is chosen. In this case, a 3-fold cross-validation was used. Fig. 3.32 illustrates the training process and the split into training, validation and test set.

As an evaluation metric for the model performance, the mean squared error between the model output on the test data set and the corresponding measurement value is used. The mean squared error (MSE) is defined as

$$\mathrm{MSE} = \frac{1}{Z}\sum_{i=1}^{n}\left(y_{pred,i} - y_{real,i}\right)^2, \tag{3.38}$$

where Z is the number of samples in the test set or the validation set. Here $y_{pred,i}$ is the predicted value of the model for the i-th entry in the test set and $y_{real,i}$ is its actual measured value.

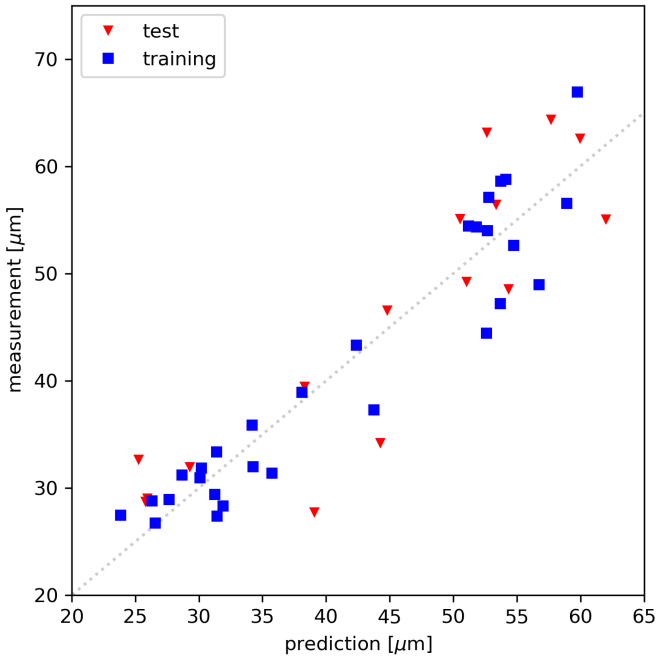

Fig. 3.33 Performance of the regression model.

The results are shown in fig. 3.33. There, the predicted thickness (x-axis) is plotted against the measured value (y-axis). The dotted line is the diagonal, i.e. the closer the points are to the dotted line, the smaller the prediction error is. Since the sample size is rather small, the here outlined training procedure was repeated multiple times and the results are plotted in the same fashion in fig. 3.34. The reason for this is that in the case of a small sample size a random test and training split can result in random effects that look like dependencies or correlations, which are not actually there. By repeating the procedure, we are making sure that the obtained result is not just occurring by chance. In the figure, both the predictions on the training set (blue squares) and the predictions on the test set (red triangles) are shown for six different runs with randomly sampled training and test sets. Corresponding regression coefficients, bias parameter and MSE are summarised in table 3.2.

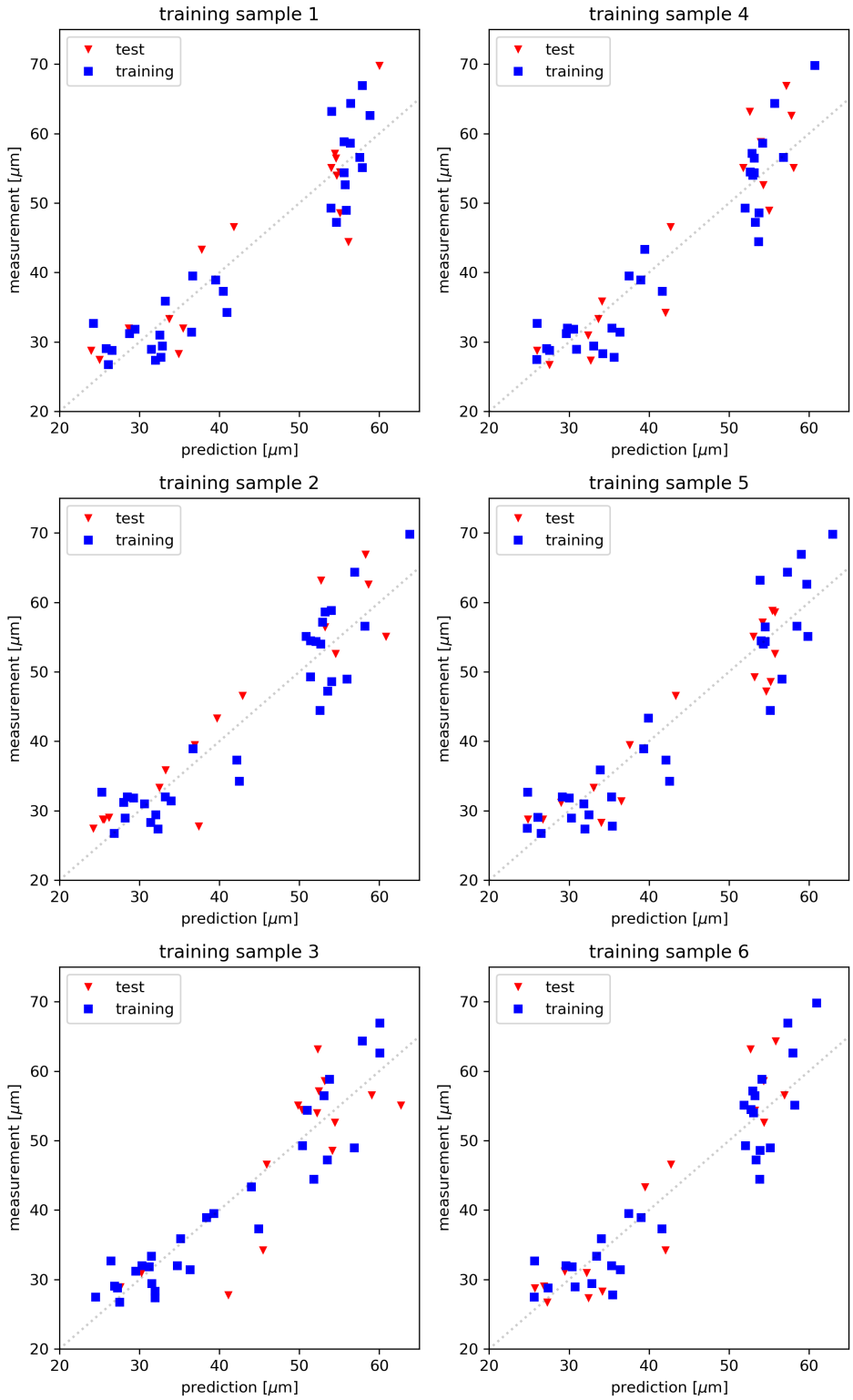

Fig. 3.34 Performance of the regression model. Six different training sessions.

3.4.4 Discussion and Interpretation

From table 3.2, we can see that the regression mainly focuses on the third component to make its predictions, which goes along with the hypothesis from the section before and proves that NMF can indeed extract physically meaningful sources from the process data. The coefficients b_1,b_2 and b_3 have a straight forward interpretation, because they can be directly related to one of the components from the NMF model (i.e. $\mathbf{\Theta}_1$, $\mathbf{\Theta}_2$ and $\mathbf{\Theta}_3$). This way the magnitude of the coefficient tells us how strongly a specific NMF component effects the output of the regression model. Since we can also physically interpret the individual components, this results helps us in understanding and evaluating the regression equation. In this case the most important factor used to predict the layer thickness is $\mathbf{\Theta}_3$, which can be associated with the heat transfer coefficient of the system, and thus makes physical sense.

In fig. 3.34, we can see that a linear model is able to generate predictions close to the real measurements. It seems that there are two clusters one for higher values and one for lower measured values. The reason for this is based on the fact, that the measurements were mainly taken within two measurement sessions on different days. The workers operating the casting machine regularly switch and each one might have a different judgement about how to apply the release agent. The error for the predictions in the range around 60 μm is overall larger in all six runs. This might be due to the fact that the calibration of the measurement device was not adjusted in an optimal way for this range. For the calibration, reference plates are used, which are coated with a fixed layer thickness. For our measurements, only a 25 μm, a 110 μm and a 200 μm reference plate were available.

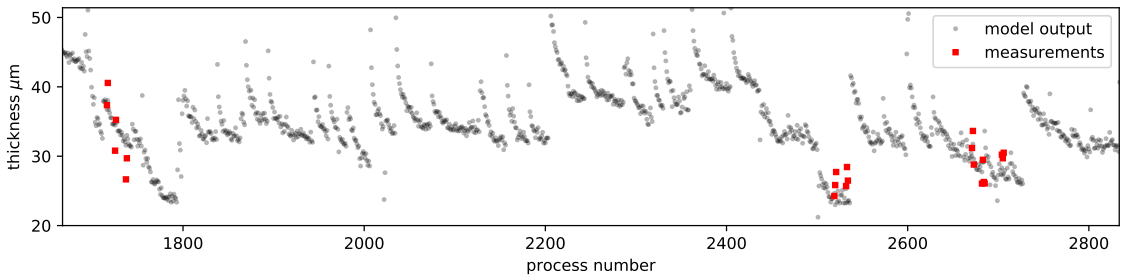

Fig. 3.35 The output of the virtual sensor model during running series production. The red dots mark control measurements. It is important to note, that this illustration is not to be taken as a representation of the real thermal processes.

Evaluating the results, we can see that the prediction error is close to the uncertainty of the measurements. In this sense, the soft sensor designed with NMF preprocessing is a viable substitute for the magnetic induction measurement device. Furthermore, the requirements for measurement precision are not high, as due to process related reasons it is not possible

to control the amount of applied release agent with a sub $\pm 10 \mu m$ precision, as the release agent is applied in a manual process based on an employees' visual judgement. Therefore, useful information for process experts in most situations is the knowledge about the time of reapplication of the release agent and if the layer is thinner or thicker than a comparative condition at a different time during the production.

3.4.4.1 Application as a Monitoring System

The results reported so far already offer the possibility to be used in real-world applications. In the course of my thesis, I implemented an in-line measurement system to monitor the release agent applied to the casting cavity's surface.

Fig. 3.36 Image of the alerting system implemented on top of the control panel of the casting machine. The monitor shows the status of the cavity's surface. The status is estimated using a software which employs the results from the NMF-based approach outlined in this chapter.

The output of this system can be seen in fig. 3.35. There, the sudden jumps can be identified as the moments when an employee applied a new layer of release agent and the

gradual decrease in-between shows the gradual removal of the layer. The red dots mark test measurements, which were performed to test the validity of the system. This behaviour of the release agent layer during the continuous production is known, but up until now it could not be monitored as an additional process parameter. With this approach, many new process optimisations become possible. For example, as can be seen in fig. 3.35, the layer thickness varies significantly during on-going production (the time period in fig. 3.35 shows roughly two and a half days of continuous production). One scenario is that, until a major cleaning procedure, the reapplication of the release agents leads to a stepwise increase in layer thickness. If the layer is too thick, i.e. the heat transfer coefficient is too small, the possibility of casting defects might increase. Similarly, if the layer is too thin, the solidification process might be too fast, which also might favour certain defect types (e.g. porosities).

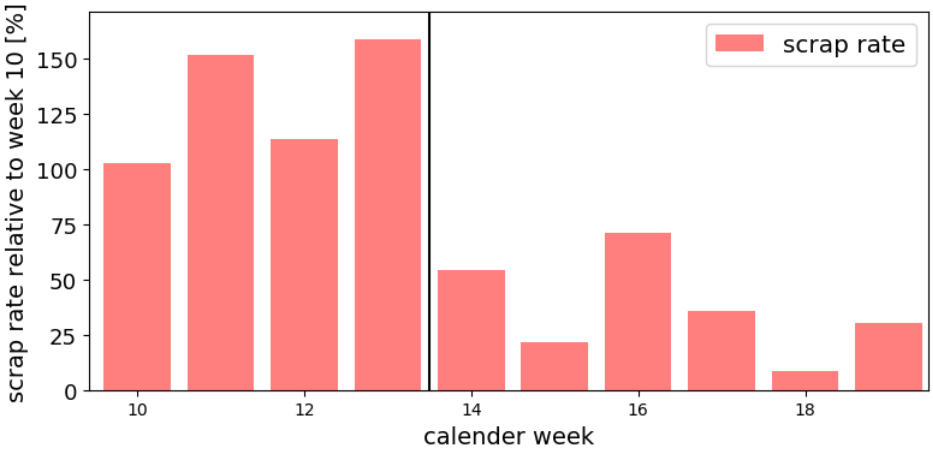

Fig. 3.37 The plot shows the scrap rate per week over a period of multiple weeks. The black line marks the time, when the monitoring system was implemented at the production site. The scrap rate is given as relative to the scrap rate in week 10 (week 10 is 100), because the actual scrap rate numbers are confidential information.

Due to this high potential of cost saving, an alert system was implemented in the BMW plant in Landshut (Germany) and tested during production. In this experiment, a significant reduction of certain defect types could be achieved, which lead to the management's decision of a full implementation for all related processes. This highlights the practical applicability of the results reported in this thesis. Fig. 3.36 shows the alert system on site attached to the control panel of a casting machine. The monitor shows the actual status of the cavity's surface, which is calculated by using the NMF-based approach outlined in this chapter. If the quality of the release agent layer is insufficient, the alert system presents a warning message and the machine operator can immediately react. This way, the production of scrap parts could be significantly reduced because up until now the quality of the release agent has

only been checked visually with human experience, which carries the risk of misjudgements. Fig. 3.37 shows how the scrap rate changed after the alerting system was implemented. At the beginning of week 14, the system was ready and the scrap rate was significantly lower. On average, the scrap rate could be reduced by roughly 63%, which shows the significant potential for cost saving. The results from this chapter, which are used in the software of the alerting system, lead to a patent application [116].

The proposed approach exploits the fact that in industrial manufacturing, standard process data like temperature recordings are generated in a large amount and can be analysed with pattern extraction algorithms like NMF, which rely on a certain amount of data to perform properly. Repeating patterns can only be extracted if there is a large enough sample size available. After the component processes have been identified, they can be used in conjunction with a limited amount of measurements (in the range of $\sim 10^1$). The same approach might be applicable to other industrial processes.

3.4.4.2 Limitations of the Approach

Data-driven soft sensors are, due to the fact that they are based on measurements from the real processing plant, closer to the actual process conditions than first-principle models, which are commonly based on ideal descriptions of steady-state processes. Yet there can also be drawbacks if one tries to model processes solely based on data. One problem is that in reality, processes gradually develop during their operation time and sometimes sudden or abrupt changes occur. In our case, the casting machine has a certain life time due to wear down effects. Additionally, there are wear down effects in some components like cooling channels or heatings. Effects like these will gradually result in a deterioration of prediction accuracy, because the generated process data will also gradually change, while the NMF model is based on the distribution of an earlier generated dataset. In such cases, a common counter measure is to regularly re-train the model to adjust the coefficients to the new conditions.

A second problem is the fact that the approaches for the design of data-driven soft sensors are usually based on algorithms which model the steady-state conditions of a continuous process. This means that the model is not able to deal with transient states like start-up processes or process interruptions. In metal casting, during the start-up phase, the machine and the steel of the cavity are usually much colder and the temperature distribution is different, i.e. the NMF reconstruction has a larger error on data from these time periods. After a metal part has been produced, some part of the thermal energy dissipates into the casting machine and some part of this energy will is then lost during the time until a new casting process starts. At some point after a few castings, the casting machine will reach an equilibrium state and in this steady-state process, the casting quality is the best. Since most of our process data

is generated while the machine is operating in this equilibrium state, NMF also models the main variations that are present during these process states.

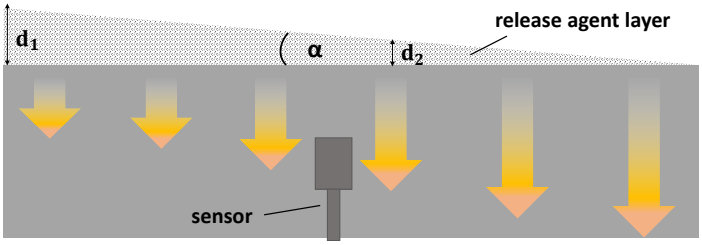

Fig. 3.38 Schematic illustrations of the designed sensor limitations. Here, the layer has a higher thickness on the left-hand side. The orange arrows represent the magnitude of the heat flow, which will be larger if the layer is thinner.

Another event during the process, which the model cannot account for, is when the sensor position changes due to maintenance work or other reasons. In such a scenario, the model has to be retrained with data collected after the change in sensor position. The layer of release agent also is not uniform. Due to the manual application process, in some areas, the layer thickness is larger than in other areas. Furthermore, the layer is removed non-uniformly after multiple processes. One reason for this is that the casting part shrinks during solidification and might shrink on certain areas of the cavity. In these areas, the layer is worn down faster than in others. As the sensor is placed within the steel of the cavity (0.5 cm to 3 cm from the surface), the measurement is affected by the heat flow coming from an area of the surface. If the layer thickness is not distributed equally in this area, the NMF-based model cannot capture such effects. In fig. 3.38, this is illustrated schematically. Here, the layer thickness is larger on the left-hand side (d_1) than the right-hand side (d_2) because the layer is titled by an angle α. We would expect the heat flow to be lower on the left-hand side due to the isolation effect of the layer. As the sensor is placed in the middle, it is going to record a temperature signal which is influenced by the heat flow from these different areas. Our model is based on the recordings of a single sensor and cannot recognize different non-uniform thickness distributions.

Chapter 4

Conclusion

4.1 Discussion and Interpretation

In this section, I am going to discuss the results that have been reported so far in this thesis. I also intend to give comments on additional methods and algorithms that could either have been also studied in the course of this thesis or share similarities with the chosen methods. Afterwards, I am going to discuss limitations of the approaches outlined in the previous chapter and then end the thesis with a summary of the main findings of the thesis and an outlook on possible next steps.

4.1.1 Comments on the NMF-based Approach

The NMF-based approach outlined in section 3.2 shows that the idea of analysing temperature profiles (measured during a manufacturing process) with matrix decomposition techniques, is a practical and viable way to extract additional knowledge from sensor signals. This is demonstrated in the application to real-world data in section 3.3. More precisely, the application shows that it is possible to extract features that can be associated with physical thermal quantities that vary during the manufacturing process and the feature extraction is based on a linear decomposition of the form $\mathbf{T} = \mathbf{W}\boldsymbol{\Theta}$. The NMF decomposition is known for its ability to extract interpretable results and there are many reports in literature from various application domains (see section 2.1.2), but this specific application area and also the decomposition of temperature time series in this form has not been reported yet in literature.

The combination of multivariate Taylor expansion and matrix decomposition techniques discussed in section 3.2.2 allows for a natural interpretation of the decomposition results obtained from any time series dataset and to the best of my knowledge have not yet been

reported in literature. Most NMF applications base the interpretabilty on the nonnegativity quantity that typically directly relates to the measured quantity (like spectral or image data).

In contrast to this, temperature is a quantity that is not additive and therefore this straight forward interpretation is not possible. However, if we consider the component processes extracted with NMF to relate to the first order terms in a multivariate Taylor expansion, this directly gives an interpretation to each component, because each Taylor term is obtained by a specific partial derivative. By enforcing nonnegativity on the weight matrix $\mathbf{W}$, the decomposition is guided towards a solution which describes the processes with components that are derived from the lowest hidden parameter values that are present in the dataset (see section 3.2.2 and equation (3.20)-(3.22)). However, the nonnegativity property does not have to hold for the component matrix $\mathbf{\Theta}$. In the application shown in this thesis, the physical model used for the initialisation is strictly based on nonnegative functions, so the nonnegativity constrain does not disturb the space of possible solutions in an unwanted way. For other processes and other quantities, this might not be the case. The partial derivatives used to initialise the component matrix can also be negative and then one should relax the nonnegativity constraint and only enforce it for the coefficient matrix $\mathbf{W}$:

$$\mathbf{T} \simeq \mathbf{W}\mathbf{\Theta} \text{ with } \mathbf{W}_{ij} \geq 0 \text{ and } \mathbf{\Theta}_{ij} \in \mathbb{R}. \tag{4.1}$$

This relaxation is called semi-NMF and implementations of algorithms that solve (4.1) can be found in [45]. Also the HALS algorithm outlined in section 2.1.6 can easily be extended to solve the semi-NMF problem by removing the positive projection in the update rule for $\mathbf{W}$ in (2.60). With this relaxation, the approach outlined in this thesis offers the potential to be applicable in many other application domains.

4.1.1.1 Comparison with other Matrix Decomposition Techniques

The other mentioned well-known matrix decomposition techniques like PCA and ICA cannot be used in a similar fashion because both impose mathematical constraints that have no physical meaning. PCA is purely data-driven and has an algebraic foundation that does not require any initialisation, i.e. there is no way to incorporate prior knowledge into the algorithm. ICA typically has to be initialised, but most algorithms require the input data to be whitened, i.e. transformed to uncorrelated coordinates with unit variance, which acts as an additional mathematical constraint because there is the possibility of information loss during this process. PCA enforces the extracted projections to be uncorrelated and ICA enforces statistically independent projections. Both constrains do not have a direct physical

interpretation and, indeed, if the component processes that occur during the manufacturing process do not follow these specific properties, then both techniques will fail to capture these components. In the appendix, I have included a decomposition of the dataset from section 3.3, both with PCA and ICA. There is a variant of ICA called "nonnegative ICA," which imposes nonnegativity constrains similar to NMF [88, 87], but the algorithm still optimises the projections to be statistically independent and as such is still not applicable in the same way as NMF. There is a similar variant of PCA called "nonnegative PCA" [124].

There is a variety of techniques that have a mathematical connection to NMF or are under certain conditions equivalent to NMF. Although in this thesis, I have not performed any comparison with these techniques, because this is not the primary goal of this work, they potentially might yield similar results as NMF. Some types of NMF implementations are an instance of a general probabilistic model called "multinomial PCA." If the cost function is the Kullback-Leibler divergence (see (2.13)), then NMF becomes equivalent to a technique called "probabilistic latent semantic analysis" (PLSA) [41]. So we could consider to use the implementations of these algorithms instead. The already mentioned semi-NMF can also be seen as a clustering algorithm, namely a relaxed form of the well known "k-means" algorithm, which could also act as an alternative to NMF in this application domain. Another clustering algorithm with strong mathematical connections to NMF is "spectral clustering" [29]. So investigating clustering algorithms to decompose datasets similar to the ones analysed in this thesis might be an interesting route for further research.
The k-means algorithm can be formulated as an expectation maximization (EM) algorithm that solves the problem of estimating a gaussian mixture model [108]. Also the before mentioned PLSA is formulated as an EM algorithm [41]. This strong connection of the NMF problem with the problem of estimating mixture models with EM algorithms offers the possibility to replace the problem of solving the NMF problem with the estimation of probabilistic mixture models. This could be advantageous because there are various extensions and techniques for the design of EM algorithms. In the course of this thesis, I did no further investigations on this connection because algorithmic improvements were not my main focus.

4.1.1.2 Physically Inspired Machine Learning

The work in this thesis joins in with recent studies presenting the promise in the idea of combining machine learning techniques with physical knowledge [84, 113, 91]. The general approach is to incorporate structured information into a learning algorithm, which results in amplifying the information content of the data that the algorithm extracts, enabling it to quickly steer itself towards a physically meaningful and interpretable solution. Furthermore,

the obtained models generalise well even when only a few training examples are available. Since the proposed initialisation technique is based on modelling a signal from underlying differential equations, there are some similarities with recent approaches to "learn" solutions to nonlinear partial differential equations with deep learning techniques [90]. Another approach based on neural networks called "precision learning" has been reported in the last years [81]. In this publication, the author incorporates physical dependencies into the learning procedure of the network. In general, one main problem with approaches based on neural networks is that they lack the interpretability that comes with NMF.

4.1.1.3 Initialisation Strategies

As already mentioned, the question of how to initialise NMF is still an open question and up until now an, optimal strategy that is viable in every application area does not exist [68, 53, 95]. In section 2.1.2.5, I introduced an initialisation strategy based on the singular value decomposition and, as was demonstrated in section 3.3.2, it is possible to extract to some degree similar component processes as with NMF. So the suggestion here would be to use this data-driven initialisation strategy as a head-start if no prior knowledge is available and if there is no physical model of the process at hand. The idea is to initialise NMF with any strong inherent structure that is present in the dataset. Another data-driven approach would be to initialise NMF with k-means clustering. Here, the initial component vectors are chosen to be the cluster centroids and the initial weights are the corresponding cluster indices. NMF can be seen as a general version of k-means clustering with orthogonality constraints [29].

Typically, any NMF initialisation strategy aims at providing a starting point for the optimisation that lies close to the global optimum of the solution space of the optimisation problem. In this sense, the goal for which the approach outlined in section 3.2 was designed differs from typical NMF applications. Instead of looking for the global optimum of the cost function, the interesting point is rather a local minimum, which approximates the time series in a physically interpretable way (i.e. as a Taylor expansion). So the results reported in this thesis can be seen as a side effect of the inherent property of NMF implementations to get stuck in local minima. Instead of exploiting this property of NMF algorithm, a more sophisticated approach might be to incorporate the underlying physical model into the NMF cost function as a constraint or as multiple constraints. This way, it might be possible to enforce the convergence towards the desired solution. How such constraint terms have to be designed is a question for further research and has not been studied in this thesis.

4.1.2 Comments on the NMF-based Virtual Sensor

The outlined method to extract component processes with NMF in combination with a linear regression to predict a target variable shown in section 3.4.1 has a huge practical applicability. This is demonstrated by the fact that an implementation of the results has already been tested in the series production of the partner company and significant results could be achieved (scrap rate reduction of a specific defect by roughly 63%). The main advantage is that there are no additional requirements for this application except standard temperature sensors that are already typically embedded into the cavity. If the production starts and the sensory data is recorded and saved in a database, then after one week there will be enough data to train the NMF model. Overall, the amount of expenditure to set up the monitoring system is rather low. The highest expenditure is the acquisition of the measurements of the release agent layer thickness.

4.1.2.1 Comparison to other Approaches

In general, the outlined approach follows the same principle as the already mentioned application of virtual sensors in other industrial domains. First, the input dimension of the independent data is reduced by a data-driven modelling technique and then a regression model is trained with the features from the reduced space to predict the desired target variable. These approaches are referred to as "black-box" models in literature because the actual dependencies learned by the dimension reduction model are not of interest. There are also the so-called "first principle models," which are based on hard-coded rules (no learning algorithms). A general comparison of the approaches discussed in this thesis with first principle models is not provided here, because, since the models are hard coded, one would need to evaluate specific application areas, which is beyond the scope of this thesis.

Most applications use PCA as a dimension reduction technique [114, 50]. PCA can be described as a matrix decomposition technique similar to NMF but with different constraints. If the goal is solely a low regression error, then PCA is a viable alternative for NMF in our setting. One advantage of NMF is that with the knowledge-based initialised NMF, one captures information about hidden parameters in individual components, whereas with PCA this information might be spread across multiple components. Furthermore, from the regression coefficients it is possible to measure the individual contributions of the input variables. The interpretability of the NMF components thus allows to interpret the interdependencies of the whole regression model in physical terms. This can be advantageous if the model quality deteriorates over time (for example due to changing environmental

conditions) and the reason for the deterioration needs to be found. A technique that shares similarities with principal component regression is "partial least squares" (PLS), which is based on calculating principal components from both dependent and independent variables. PLS is also used for the design of indirect measurement devices [65, 80]. Similar to PCA, the method is also purely data-driven and does not allow to incorporate knowledge about underlying physical processes.

Artificial neural networks (ANN) have an incorporated dimension reduction which is learned during an optimisation procedure of the networks weights and as such they have found application in similar problem settings [47, 8]. Neural networks transform the input data multiple times with nonlinear functions, depending on the network depth and architecture. This is why the actual physical dependencies are lost or hard to extract from the network weights. In contrast to NMF, neural networks are able to learn complex nonlinear interdependencies in the data, which theoretically makes them applicable to a wide range of problems. The downside is that neural networks require a larger amount of data for their training compared to other methods. In casting processes, there is only be a limited number of measurements available due to the costly and dangerous measurement procedure. This drawback makes ANNs impractical for monitoring the release agent layer in casting processes.

4.1.3 Limitations

In section 3.4.4, I have already already discussed limitations of the use of NMF and linear regression to design a virtual sensor for process monitoring. In this section, I am going to extend this discussion with more general remarks about the limitations of NMF as an analysis tool for temperature time curves.

The NMF-based analysis approach comes with the general limitations of any NMF decomposition. The latter is limited to linear processes and will surely fail, if strong nonlinearities are involved in the underlying physical processes. In section 3.2.2, the connection between NMF and the linear Taylor expansion is outlined and the Taylor expansion can also only be viable if the higher order terms are assumed to be small compared to the linear terms. Thermal processes are generally difficult to control and as thus non-linearities might occur due to process related variations. In practical implementations, it would be helpful to have a metric to test, if the NMF approximation of the time curves is a reasonable representation of the data. Such a metric could be based on the approximation error or one could employ simulated data of the process. As already mentioned, there are nonlinear dimension reduction techniques that can learn latent structure from datasets, but these techniques offer no

straightforward way to interpret the output. This is why such techniques are not used in a similar fashion as NMF.

Additionally, any NMF has an inner degree of freedom, which needs to be determined independently, either via some model order selection or simply by trial and error. The initialisation strategy in section 3.2.3 offers a way to determine the number of components K by considering a simplified physical model of the underlying mechanisms. Yet also this approach does not offer a way to determine a fixed amount of components beforehand because each initialisation is at first just a guess about the expected component processes and does not have to match the real case. An extension to the outlined approach in section 3.2.3 could be the inclusion of a model order selection step. A short discussion about model order selection can be found in section 2.1.2.

Another limitation of the NMF-based approach is that each obtained model is applicable only for data that stems from the specific sensor used to generate the training data. This means that for a different sensor position, a new model needs to be trained. This is presented in the appendix (see appendix A.2). Furthermore, the output of the model (i.e. the weight matrix $\mathbf{W}$) is not comparable between the different models because of the mentioned indeterminancies of NMF (see section 2.1.2). If it is not possible to easily compare different model outputs and to reuse the same model for a different sensor, then this will limit the use of the approach in practical implementations because an individual NMF models needs to be trained for every new sensor. If the underlying hidden parameters follow the same distribution for different sensor positions, then one way to make the results more comparable can be to shift and scale the $\mathbf{W}_i$ to align with a reference distribution.

In the theoretical part of this thesis, I mentioned that the NMF cost function is non-convex in both arguments, which causes the tendency of the algorithm to get stuck in local minima. With an intelligent initialisation, it is possible to guide the optimisation towards a desired solution, as has been shown in the results section of this thesis. In general, this approach can still fail and there is no guarantee that NMF converges in a local minimum that reflects any physically interpretable results, which is another limitation of the approach.

4.2 Summary of the Main Results

In this thesis, I demonstrated an NMF-based approach to analyse temperature profiles generated by a thermal manufacturing process. An arrangement of multiple time series in a data matrix can be decomposed into physically meaningful features, which can be associated with ongoing physical phenomena during the production process. This decomposition can be guided by a knowledge-based initialisation strategy, linking the NMF model to hidden

physical parameters. The approach is motivated and demonstrated by its application to real-world data sets. The extracted features can be used for process monitoring and defect diagnosis and further analysis. The NMF-based feature extraction can also be used as a first preprocessing step for the training of a regression model, which can be used as a virtual sensor to measure important quantities during the ongoing production. Placing sensors to cover all the different aspects and interactions during a manufacturing process is a challenging task. Thus, the possibility to extract multiple sources from a single sensors signal is very appealing.
The main findings and results of this thesis are:

- Combining the NMF decomposition with the multivariate Taylor expansion to interpret the decomposition of time curves allows to extract information about otherwise hidden physical parameters. One important parameter in casting processes is the heat transfer coefficient and with the approach outlined in this thesis, it becomes possible to extract information about this quantity from temperature measurements. This approach exploits the fact that these hidden parameters are not constant during the ongoing production and have a certain process related variation. To the best of my knowledge, the connection between NMF and physical quantities via the Taylor expansion has not yet been noticed in literature and offers the potential to transfer my approach to other manufacturing processes and time series of other physical quantities (e.g. pressure). These results are the basis of a patent application [115].

- The initialisation strategy based on physically motivated initial component processes is similar to other applications where prior knowledge about the process is exploited. Yet there is to the best of my knowledge no publication in which partial derivatives of simplified physical models are used as initial guesses for NMF. The results reported in this thesis motivate a further investigation of this approach.

- The potential of NMF as a preprocessing step to train regression models that can be used as indirect measurement systems in manufacturing has not yet been recognized in literature. Also the specific application presented in this thesis is completely new in this domain and up to this date, no similar measurement system for the release agent in casting processes exists. This indirect measurement system is described in another submitted patent application [116].

- The results reported in chapter 3.2 have been submitted to "Journal of Manufacturing Systems" and have been accepted for publication. A preprint can be found in [117].

- The results reported in chapter 3 have been implemented in the form of a process monitoring tool at production site of a casting process. With this tool, it was possible to achieve a significant reduction of the scrap rate, i.e. an immediate financial benefit for the company. This practical applicability led to the management decision to implement this monitoring tool in other casting processes.
- The results in chapter 3.2 and 3.4 have been presented to process experts at "63. Österreichische Gießereitagung."

4.3 Outlook and Further Research

In the course of this thesis, I have considered the signals of only one specific sensor embedded in a casting machine, yet there actually are multiple temperature sensors at different positions that record a time curve during a manufacturing process. A natural extension of the analysis method outlined in this thesis would be the use of tensor factorisation techniques. Instead of a two-dimensional matrix one could design a three dimensional data tensor that contains signals from multiple sensors. There are different tensor factorisation algorithms available. The most common types are PARAFAC [54] and the Tucker decomposition [111]. Tensor decomposition can also be formulated with nonnegative constrains like NMF and an extensive summary of available algorithms can be found in the book of Cicocki et al. [26]. A higher order decomposition that includes different sensor positions into underlying components might extract hidden information that cannot be extracted from data structured as two-dimensional matrices.

As already mentioned, in the main part of this thesis, the results reported so far offer the potential to be applicable for other manufacturing processes and time curves of different physical quantities. In casting processes, typically also pressure is measured as a time curve during the manufacturing process. In consecutive studies, the analysis of this kind of data could be a possible first step. Furthermore, NMF can act as a preprocessing step to train a regression model to act as an indirect sensor. Instead of the release agent, one could try to predict other important quantities that are relevant for the product quality or the process in general

Finally, an interesting topic for further research would be to investigate approaches to incorporate known physical models about the process directly into the NMF algorithm. This could either be done by incorporating the knowledge into the optimisation procedure or by extending the NMF cost function with suitable designed constraints.

Appendix A

Additional NMF Results

A.1 Initialisations

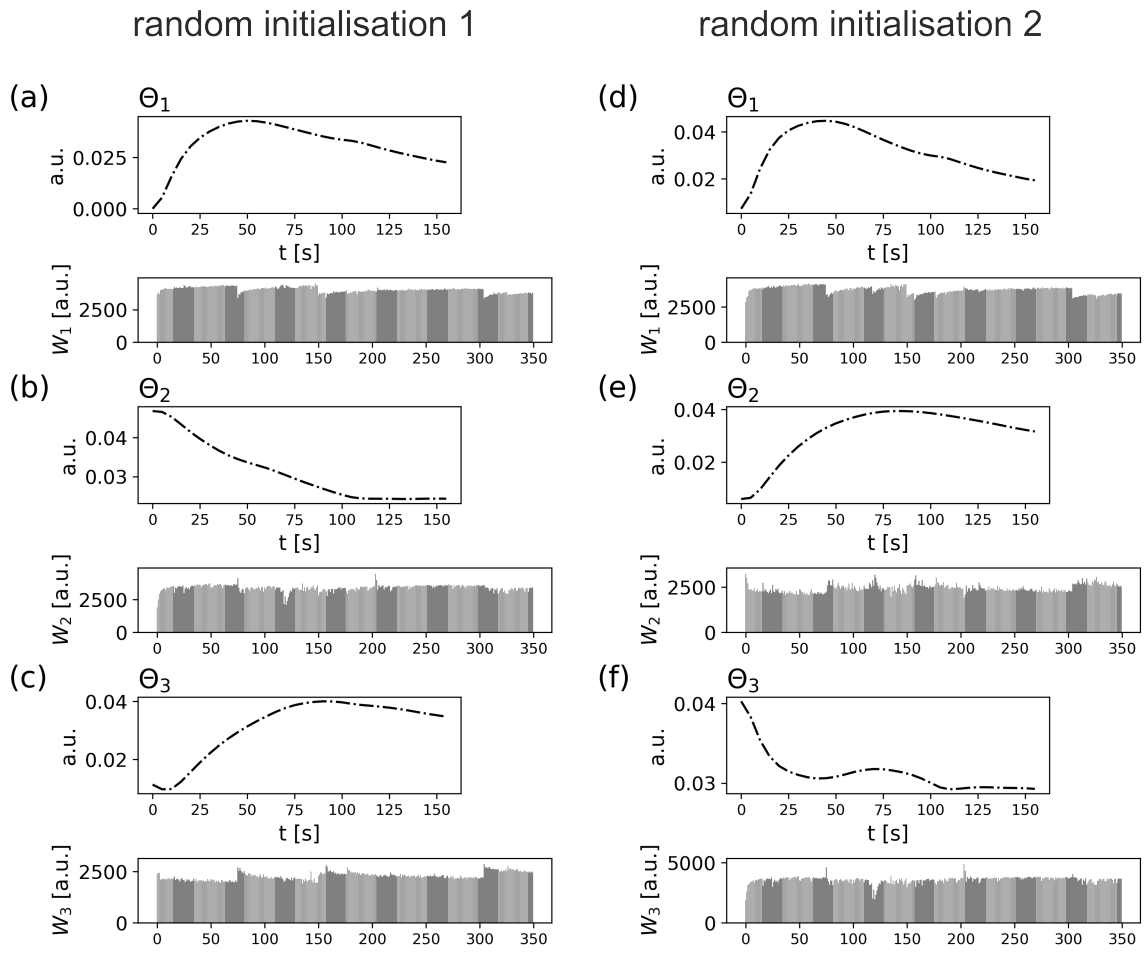

Fig. A.1 Dataset 1: The NMF decomposition with random initialisation with $K = 3$ components. The extracted component processes lack interpretability.

Fig. A.1 shows two NMF decompositions with $K = 3$ components. Both results were obtained by randomly initialising the starting values $\mathbf{W}_{init}$ and $\mathbf{\Theta}_{init}$. As already mentioned, the Frobenius norm cost function is non-convex and so the NMF optimisation procedure is prone to get stuck in a local minima. If the algorithm is randomly initialised the results will differ significantly between different runs. If one is looking for interpretable results, this is a problem because there is no clear way to say which result is the one keep. The left-hand side and the right-hand side in fig. A.1 both contain component processes which appear to resemble exponential functions that might originate from underlying physical processes, but there is no decision criteria to decide which one is the best representation of the data. If one uses the knowledge-based initialisation the results are easier to relate to possible other physical quantities.

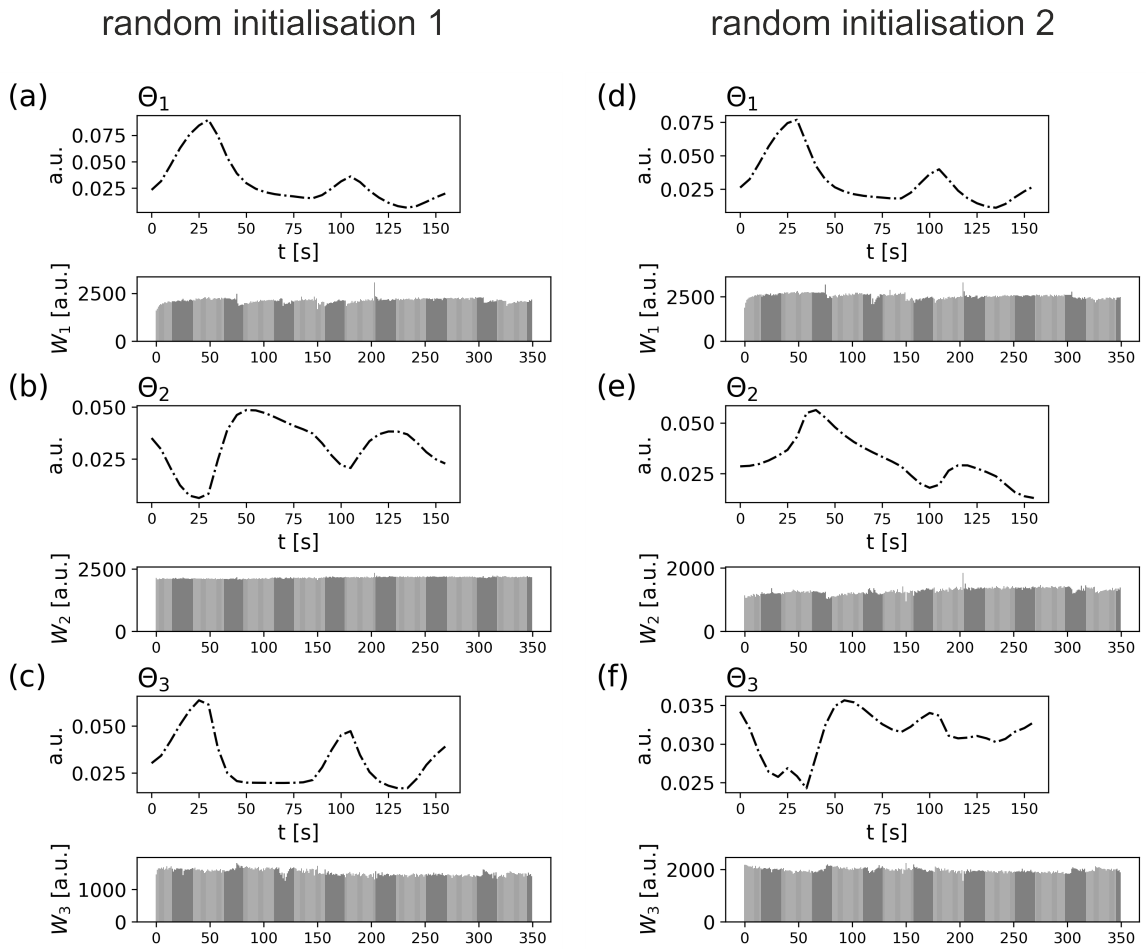

Fig. A.2 Dataset 2: The NMF decomposition with random initialisation with $K = 3$ components. The extracted component processes lack interpretability.

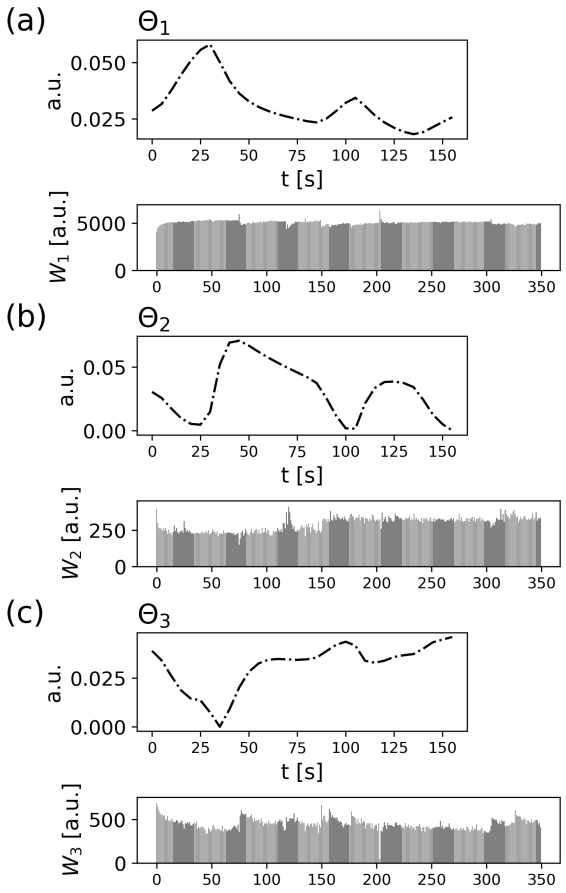

Fig. A.3 Dataset 2: The NMF decomposition with NNDSVD initialisation with $K = 3$ components.

Still one decision criteria which is typically used if the NMF is randomly initialised, is the final value of the cost function, i.e. the best fit is chosen to be the best solution. Yet this method does not have any physical motivation and solely evaluates the result from an optimisation point of view. Both solutions shown in fig. A.1 have a very low approximation error below ± 1.5. A "best fit" solution obtained after multiple runs of the NMF algorithm can only slightly improve the approximation error and there is no reason for the solution with the smallest cost function error to have any relation with the time series decomposition in (3.25). Fig. A.2 also shows two NMF decompositions with random initialised factor matrices, but for the dataset $\mathbf{T}_2$. Again the components obtained differ significantly if we compare the left-hand and right-hand side of fig. A.2. Furthermore, the components lack any kind of interpretability as they cannot be related to any of the partial derivatives discussed in chapter 3. This shows how useful the outlined knowledge-based initialisation strategy is, since only this way it is possible to extract interpretable components from dataset $\mathbf{T}_2$.

A.2 Model Transfer

As already mentioned in section 4.1.3, the trained NMF models are not transferable if the sensor position is different. This is demonstrated by using the model trained in section 3.3.1.1 with the data from the simple process (see fig. 3.15 (a)) to extract weights from a dataset collected from a sensor at a different position. The new sensor is positioned in proximity to the sensor discussed in section 3.3.1.1. In fig. A.4 three example curves from this sensor are shown. Due to the proximity, the curves are similar to the ones shown in fig. 3.15 (a). Fig. A.5 shows the resulting weights obtained by applying the model to the new data. Note that the data was collected during the same manufacturing processes as the one discussed in section 3.3.1.1, i.e. the same events as discussed in this section are affecting the data. $\mathbf{\Theta}_1$,$\mathbf{\Theta}_2$ and $\mathbf{\Theta}_3$ in fig. A.5 are the same as in fig. 3.16 (a)-(c), because only the weights are calculated. As can be seen, $\mathbf{W}_1$ is still the dominant coefficient, which is to be expected because it as the largest impact on the decrease of reconstruction error. The main difference is $\mathbf{W}_3$, which has significantly lower values than the in fig. 3.3.1.1 and also goes to zero for some processes. This means that the output of the coefficients cannot be compared by applying the same model to the data generated from sensors at different positions. If NMF is to be used as a preprocessing step in a data processing pipeline then this system is limited to data that stems from a sensor at one specific position.

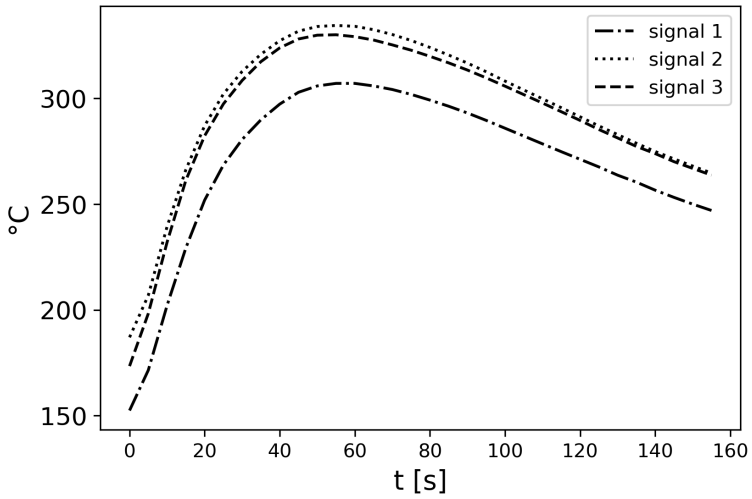

Fig. A.4 Example signals taken from a sensor positioned closely to the one discussed in chapter 3.

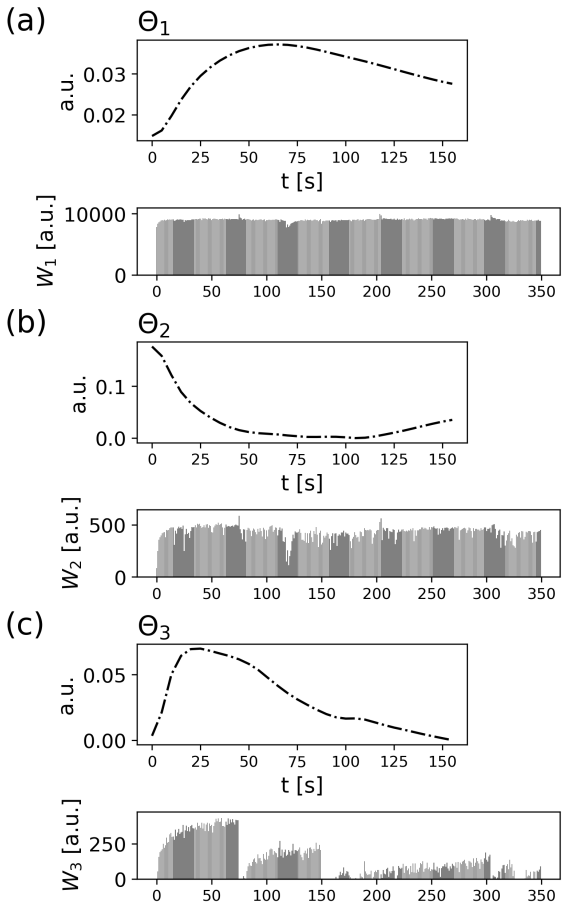

Fig. A.5 Results of the application of the model discussed in chapter 3 on data from a different sensor.

A.3 PCA and ICA Results

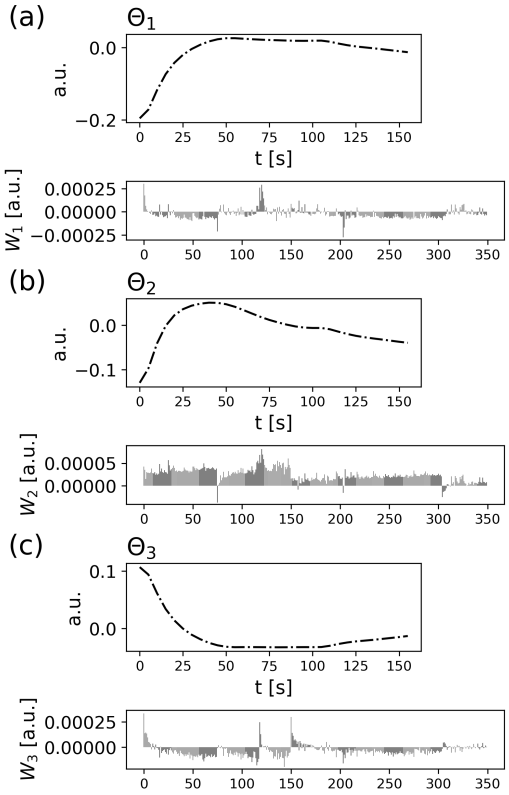

Fig. A.6 PCA decomposition with $K = 3$ components.

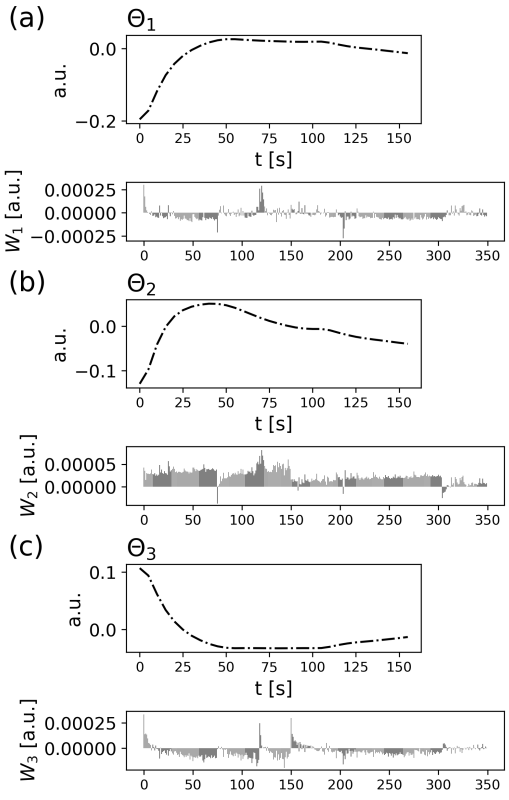

Fig. A.7 ICA decomposition with $K = 3$ components.

List of Figures

1.1 Car parts produced by metal casting. The image has been provided by the BMW Group AG. 3
1.2 Picture taken during the filling of the mould during a gravity casting process with a modern casting machine. 4
1.3 Illustration of a standard blind source separation problem occuring in industrial manufacturing. 5

2.1 Illustration of a linear dimension reduction via a factorisation model $\mathbf{T} \simeq (\mathbf{W}, \mathbf{\Theta})$. $\mathbf{W}$ contains the new coordinates of the input data in the reduced space and $\mathbf{\Theta}$ the latent components. 12
2.2 Illustration of a linear dimension reduction via a factorisation model $\mathbf{T} \simeq (\mathbf{W}, \mathbf{\Theta})$. $\mathbf{W}$ contains the new coordinates of the input data in the reduced space and $\mathbf{\Theta}$ the latent components. 13
2.3 Illustration of a rank one approximation via a factorisation model. 14
2.4 Illustration of the parts-based reconstruction of faces with the components extracted with NMF. This reconstruction is just one example of the many applications, where NMF yields highly interpretable results. Illustration is taken from [74]. 17
2.5 The effect of regularisation on the obtained solution. 26
2.6 Illustration of the definition of the error matrix. The error matrix $\mathbf{E}$ is the difference between original data and the current factorisation matrices. . . . 29
2.7 Illustration of the definition of the residuum matrices. The residua matrices are calculated by adding a rank 1 matrix to the current error matrix. 30
2.8 (a) The update rule for $\mathbf{w}_j$. The following normalization step for $\mathbf{w}_j$ is not depicted here. (b) The update rule for $\mathbf{h}_j$. 31
2.9 An NMF model is trained and the fixed $\mathbf{\Theta}$ can be used to project new unseen data onto the components 35
2.10 Unseen data is represented as a new data vector. 35

2.11 (a) The approximation error. (b) The residuum matrix for a single value. . . 36
2.12 The update step for a single value. 36

3.1 (a) A look at the inside of the cavity. The white surface is due to a coating with a release agent. (b) A NiCr-Ni temperature sensor, which is placed inside the steel of the mould. 42
3.2 The empty cavity within the casting machine. 43
3.3 The closed cavity with sand cores. 44
3.4 The filling process (image provided by BMW AG). 44
3.5 The cavity filled with liquid metal (image provided by BMW AG). 45
3.6 The metal shrinks during the solidification (image provided by BMW AG). 45
3.7 The solid casting part is automatically removed from the cavity (image provided by BMW AG). 46
3.8 Sand cores are removed with hammers (image provided by BMW AG). . . 46
3.9 Illustration of the data generation process. (a) Liquid metal is poured into the cavity. Until the metal reaches the sensor position, the sensor reflects the temperature of the steel. (b) The liquid metal reaches the position of the sensor. The temperature starts to rise. (c) Cavity is filled and metal cools down and solidifies. (d) Solid metal part is ejected and the temperature recording stops. 47
3.10 Illustration of how different contributing factors change the shape of the recorded sensor signal. The left column shows the initial curve (dashed line) and the curve after one parameter was changed (dotted line). The middle column shows the shape of the change and the right column the corresponding partial derivative from the Taylor expansion. For (a), the parameter α was increased. For (b), the initial temperature T_0 and for (c), the environmental temperature T_S were increased. 52
3.11 Illustration of the matrix factorisation model $\mathbf{T} \simeq (\mathbf{W}, \mathbf{\Theta})$. $\mathbf{T}$ contains the recordings of a specific sensor from consecutive production of parts. The rows of $\mathbf{T}$ are ordered chronologically. $\mathbf{\Theta}$ contains the basis functions and W the activations of the corresponding basis functions for a specific time series. 57
3.12 Depiction of how the results of the matrix factorisation model $\mathbf{T} \simeq (\mathbf{W}, \mathbf{\Theta})$ are presented in this thesis. 58
3.13 Illustration of the simulated toy data set. The left shows the artificial variation of the initial temperature with a triangular wave function. The right-hand side shows the sinusoidal variation of the thermal diffusivity. 59

3.14 Results with simulated data. Each column shows the decomposition results, obtained from a simulated dataset with different magnitude of inherent variations. 60
3.15 Examples of three different temperature recordings from the two datasets. (a) A simple process with a heating and a cooling phase. (b) A more complex process with additional cooling effects. 62
3.16 Dataset 1: The NMF decomposition with knowledge-based initialisation with $K = 3$ and $K = 4$ components. The extracted component processes resemble the shape of the partial derivatives we obtain from the Taylor expansion. . . 64
3.17 Schematic illustration of how a different layer of release agent influences the temperature signal. A thin layer (left) causes a higher rate of heat flow than a thicker layer (right). The shape of the recorded sensor signal changes in a specific way depending on the layer thickness. 65
3.18 The image shows how the release agent is applied to a casting cavity. During the running production the layer has to be reapplied regularly (image is taken from [33]). 66
3.19 The component process $\mathbf{\Theta}_3$. The arrows mark the event at which the release agent is manually reapplied. 67
3.20 Dataset 1: The NMF decomposition with NNDSVD initialisation with $K = 3$ and $K = 4$ components. The extracted component processes lack interpretability. 69
3.21 Dataset 2: The NMF decomposition with knowledge-based initialisation with $K = 3$ and $K = 4$ components. With $K = 4$, the extracted component processes resemble the shape of the partial derivatives one obtains from the Taylor expansion. 71
3.22 Comparison of convergence speed with different initialisation strategies. The reconstruction error is the value of the cost function after each iteration step. Knowledge-based initialised NMF and NNDSVD initialised NMF converge after a few iterations. Random initialisation needs significantly more iteration steps to achieve a similar error. 72
3.23 L_1-regularisation with γ_1=0 and γ_1=1 for both matrices. 74
3.24 L_1-regularisation with γ_1=10 and γ_1=50 for both matrices. 76
3.25 L_2-regularisation with γ_2=0 and γ_2=1 for both matrices. 77
3.26 L_2-regularisation with γ_2=10 and γ_2=50 for both matrices. 78
3.27 L_1-regularisation for the weights ($\gamma_{W,1} = 0, 1$) and L_2-regularisation for the component processes ($\gamma_{\Theta,2} = 0, 1$). 79

3.28 L_1-regularisation for the weights ($\gamma_{W,1} = 10, 50$) and L_2-regularisation for the component processes $\gamma_{\Theta,2} = 10, 50$). 80
3.29 Image taken during one measurement session. During a 30-second time window, multiple measurements at different positions were taken. The image is blurred to hide any confidential information. 83
3.30 Illustration of the measurement procedure. 85
3.31 Left: The training set for the NMF model. Right: The extracted component processes used for preprocessing. 86
3.32 Illustration of the training process with 3-fold cross-validation and the split into training, validation and test set. The sample number in each set is written in the boxes. 87
3.33 Performance of the regression model. 88
3.34 Performance of the regression model. Six different training sessions. 90
3.35 The output of the virtual sensor model during running series production. The red dots mark control measurements. It is important to note, that this illustration is not to be taken as a representation of the real thermal processes. 91
3.36 Image of the alerting system implemented on top of the control panel of the casting machine. The monitor shows the status of the cavity's surface. The status is estimated using a software which employs the results from the NMF-based approach outlined in this chapter. 92
3.37 The plot shows the scrap rate per week over a period of multiple weeks. The black line marks the time, when the monitoring system was implemented at the production site. The scrap rate is given as relative to the scrap rate in week 10 (week 10 is 100), because the actual scrap rate numbers are confidential information. 93
3.38 Schematic illustrations of the designed sensor limitations. Here, the layer has a higher thickness on the left-hand side. The orange arrows represent the magnitude of the heat flow, which will be larger if the layer is thinner. . . . 95

A.1 Dataset 1: The NMF decomposition with random initialisation with $K = 3$ components. The extracted component processes lack interpretability. . . . 108
A.2 Dataset 2: The NMF decomposition with random initialisation with $K = 3$ components. The extracted component processes lack interpretability. . . . 109
A.3 Dataset 2: The NMF decomposition with NNDSVD initialisation with $K = 3$ components. 110
A.4 Example signals taken from a sensor positioned closely to the one discussed in chapter 3. 111

A.5 Results of the application of the model discussed in chapter 3 on data from a different sensor. 112
A.6 PCA decomposition with $K = 3$ components. 113
A.7 ICA decomposition with $K = 3$ components. 114

References

[1] Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *International Symposium on Information Theory*, 2:267–281.

[2] Albright, R., Cox, J., Duling, D., Langville, A., and Meyer, C. (2014). Algorithms, initializations, and convergence for the nonnegative matrix factorization. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[3] Baluja, S. (1999). Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. *Advances in Neural Information Processing Systems 11*, pages 854–860.

[4] Bergeret, F. and Le Gall, C. (2003). Yield improvement using statistical analysis of process dates. *IEEE Transactions on Semiconductor Manufacturing*, 16(3):535–542.

[5] Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., and Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(1):155–173.

[6] Bertino, E., Catania, B., and Caglio, E. (1999). Applying data mining techniques to wafer manufacturing. *Principles of Data Mining and Knowledge Discovery*, 1704:41–50.

[7] Bhattacharya, S., Pal, K., and Pal, S. K. (2012). Multi-sensor based prediction of metal deposition in pulsed gas metal arc welding using various soft computing models. *Applied Soft Computing*, 12(1):498–505.

[8] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA.

[9] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[10] Boutsidis, C. and Gallopoulos, E. (2008). Svd based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4):1350–1362.

[11] Brunet, J.-P., Tamayo, P., Golub, T. R., and Mesirov, J. P. (2004). Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences of the United States of America*, 101(12):4164–4169.

[12] Buchdahl, H. A. (1960). The concepts of classical thermodynamics. *American Journal of Physics*, 28(3):196–201.

[13] Buchsbaum, G. and Bloch, O. (2002). Color categories revealed by non-negative matrix factorization of Munsell color spectra. *Vision Research*, 42(5):559–563.

[14] Castelli, V. and Cover, T. M. (1995). On the exponential value of labeled samples. *Pattern Recognition Letters*, 16(1):105–111.

[15] Castelli, V. and Cover, T. M. (1996). The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42(6):2102–2117.

[16] Chan, T.-H., Ma, W.-K., Chi, C.-Y., and Wang, Y. (2008). A convex analysis framework for blind separation of non-negative sources. *IEEE Transactions on Signal Processing*, 56(10):5120–5134.

[17] Chen, D. and Plemmons, R. J. (2009). Nonnegativity constraints in numerical analysis. *łdotson the Birth of Numerical Analysis*, pages 1–32.

[18] Chen, X., Gu, L., Li, S. Z., and Zhang, H.-J. (2001). Learning representative local features for face detection. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–1126 – I–1131 vol.1.

[19] Chen, Z. C. Z., Cichocki, A., Rutkowski, T., and Zhe Chen (2006). Constrained non-Negative Matrix Factorization Method for EEG Analysis in Early Detection of Alzheimer Disease. *IEEE International Conference on Acoustics Speech and Signal Processing*, 5(4):V–893–V–896.

[20] Chien, C.-F., Wang, W.-C., and Cheng, J.-C. (2007). Data mining for yield enhancement in semiconductor manufacturing and an empirical study. *Expert Systems with Applications*, 33(1):192–198.

[21] Cho, Y. C. and Choi, S. (2005). Nonnegative features of spectro-temporal sounds for classification. *Pattern Recognition Letters*, 26(9):1327–1336.

[22] Chu, M. and Plemmons, R. (2008). Nonnegative matrix factorization and applications. *Bulletin of the International Linear Algebra Society*, 34:1–5.

[23] Cichocki, A. and Amari, S.-i. (2002). *Adaptive Blind Signal and Image Processing*. John Wiley & Sons, Ltd, Chichester, UK.

[24] Cichocki, A., Lee, H., Kim, Y.-D., and Choi, S. (2008). Non-negative matrix factorization with α-divergence. *Pattern Recognition Letters*, 29(9):1433–1440.

[25] Cichocki, A., Zdunek, R., and Amari, S.-i. (2006). *Csiszár's Divergences for Non-negative Matrix Factorization: Family of New Algorithms*, volume 3889 of *Lecture notes in computer science*. Springer, Heidelberg, Germany, Berlin, Heidelberg.

[26] Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S. I. (2009). *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. John Wiley and Sons, Hoboken, New Jersey, US.

[27] Comon, P. and Jutten, C., editors (2010). *Handbook of Blind Source Separation, 1st edition*. Academic Press, Oxford.

[28] Dhillon, I. S. and Sra, S. (2005). Generalized Nonnegative Matrix Approximations with Bregman Divergences. *Advances in neural information processing systems*, 19:283–290.

[29] Ding, C., He, X., Simon, H., and Jin, R. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. *Proceedings of the 2005 SIAM International Conference on Data Mining*.

[30] Ding, C., Tao Li, and Jordan, M. (2010). Convex and Semi-Nonnegative Matrix Factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55.

[31] Dörmann Osuna, H. (2009). *Thesis: Ansatz für ein prozessintegriertes Qualitätsregelungssystem für nicht stabile Prozesse*. TU Ilmenau Universitätsbibliothek.

[32] Dresden, A. (1920). The fourteenth western meeting of the american mathematical society. *Bull. Amer. Math. Soc.*, 26(9):385–396.

[33] Dycote (2007). *Dycote Kokillenschlichten Handbuch*. Borken, Germany.

[34] Fabio, C., Ira, C., Marcelo, C., and Escola, P. (2003). Semi-supervised learning of mixture models. *ICML 20th International Conference on Machine Learning*, pages 99–106.

[35] Févotte, C., Bertin, N., and Durrieu, J.-L. (2009). Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis. *Neural computation*, 21(3):793–830.

[36] Fujino, A., Ueda, N., and Saito, K. (2005). A hybrid generative/discriminative approach to semi-supervised classifier design. *Proceedings of the 20th National Conference on Artificial Intelligence*, 2:764–769.

[37] Galicia, H. J., Peter He, Q., and Wang, J. (2012). Comparison of the performance of a reduced-order dynamic pls soft sensor with different updating schemes for digester control. *Control Engineering Practice*, 20(8):747–760.

[38] Gao, B., Zhang, H., Woo, W. L., Tian, G. Y., Bai, L., and Yin, A. (2014). Smooth nonnegative matrix factorization for defect detection using microwave nondestructive testing and evaluation. *IEEE Transactions on Instrumentation and Measurement*, 63(4):923–934.

[39] Gao, Y. and Church, G. (2005). Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, 21(21):3970–3975.

[40] Gardner, R. M., Bieker, J., and Elwell, S. (2000). Solving tough semiconductor manufacturing problems using data mining. *2000 IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, pages 46–55.

[41] Gaussier, E. and Goutte, C. (2005). Relation between plsa and nmf and implications. *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 601–602.

[42] Ge, Z. and Song, Z. (2007). Process monitoring based on independent component analysis−principal component analysis (ica−pca) and similarity factors. *Industrial & Engineering Chemistry Research*, 46(7):2054–2063.

[43] Ge, Z. and Song, Z. (2010). Nonlinear soft sensor development based on relevance vector machine. *Industrial & Engineering Chemistry Research*, 49(18):8685–8693.

[44] Ge, Z. and Song, Z. (2011). Semisupervised bayesian method for soft sensor modeling with unlabeled data samples. *AIChE Journal*, 57(8):2109–2119.

[45] Gillis, N. and Kumar, A. (2015). Exact and heuristic algorithms for semi-nonnegative matrix factorization. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1404–1424.

[46] Golub, G. H. and van Loan, C. F. (1996). *Matrix computations 3rd edition*. Johns Hopkins studies in the mathematical sciences. Johns Hopkins University Press.

[47] Gonzaga, J., Meleiro, L., Kiang, C., and Filho, R. M. (2009a). Ann-based soft-sensor for real-time process monitoring and control of an industrial polymerization process. *Computers and Chemical Engineering*, 33(1):43–49.

[48] Gonzaga, J., Meleiro, L., Kiang, C., and Maciel Filho, R. (2009b). Ann-based soft-sensor for real-time process monitoring and control of an industrial polymerization process. *Computers and Chemical Engineering*, 33(1):43–49.

[49] Gonzalez, E. and Zhang, Y. (2005). Accelerating the lee-seung algorithm for non-negative matrix factorization. *Department computational and applied mathematics, Rice University, Houston, TX, Technical Report TR-05-02*, pages 1–13.

[50] Gonzalez, G. D. (1999). Soft sensors for processing plants. *Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials*, 1:59–69.

[51] Grippo, L. and Sciandrone, M. (2000). On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Operations Research Letters*, 26(3):127–136.

[52] Hancewicz, T. M. and Wang, J.-H. (2005). Discriminant image resolution: a novel multivariate image analysis method utilizing a spatial classification constraint in addition to bilinear nonnegativity. *Chemometrics and Intelligent Laboratory Systems*, 77(1-2):18–31.

[53] Hans, L., Mads, C., Mark D., P., Lars Kai, H., and Søren Holdt, J. (2008). Theorems on positive data: On the uniqueness of nmf. *Computational Intelligence and Neuroscience*, 2008:788 – 791.

[54] Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84.

[55] Hoyer, P. O. (2004). Non-negative Matrix Factorization with Sparseness Constraints. *The Journal of Machine Learning Research*, 5:1457–1469.

[56] Huang, S.-M. and Yang, J.-F. (2012). Improved principal component regression for face recognition under illumination variations. *IEEE Signal Processing Letters*, 19(4):179–182.

[57] Jin, X., Wang, S., Huang, B., and Forbes, F. (2012). Multiple model based lpv soft sensor development with irregular/missing process output measurement. *Control Engineering Practice*, 20(2):165–172.

[58] Kaneko, H. and Funatsu, K. (2011). Development of soft sensor models based on time difference of process variables with accounting for nonlinear relationship. *Industrial & Engineering Chemistry Research*, 50(18):10643–10651.

[59] Keithley, R. B., Heien, M. L., and Wightman, R. M. (2009). Multivariate concentration determination using principal component regression with residual analysis. *Trends in analytical chemistry : TRAC*, 28(9):1127–1136.

[60] Khatibisepehr, S. and Huang, B. (2008). Dealing with irregular data in soft sensors: Bayesian method and comparative study. *Industrial & Engineering Chemistry Research*, 47(22):8713–8723.

[61] Khatibisepehr, S., Huang, B., Xu, F., and Espejo, A. (2012). A bayesian approach to design of adaptive multi-model inferential sensors with application in oil sand industry. *Journal of Process Control*, 22(10):1913–1929.

[62] Kim, H. and Park, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502.

[63] Kim, H. and Park, H. (2008). Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730.

[64] Koppenhoefer, B., Wuerthner, S., Ludwig, L., Rosenstiel, W., Kuge, H.-H., Hummel, M., and Federl, P. (1997). Analysis of electrical test data using a neural network approach. *IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop ASMC 97 Proceedings*, pages 37–41.

[65] Kourti, T. (2002). Process analysis and abnormal situation detection: from theory to practice. *IEEE Control Systems Magazine*, 22(5):10–25.

[66] Kozłowski, J., Jakimiuk, M., and Rogalewicz, M. (2019). *Analysis and Control of High-Pressure Die-Casting Process Parameters with Use of Data Mining Tools*. Springer International Publishing, Cham.

[67] Lade, P., Ghosh, R., and Srinivasan, S. (2017). Manufacturing analytics and industrial internet of things. *IEEE Intelligent Systems*, 32(3):74–79.

[68] Laurberg, H. (2007). Uniqueness of non-negative matrix factorization. In *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*, pages 44–48. IEEE.

[69] Lawson, C. L. and Hanson, R. J. (1974). *Solving Least Squares Problems*, volume 53. SIAM Classics in Applied Mathematics.

[70] Lee, D. and Seung, H. (2001). Algorithms for non-negative matrix factorization. *Advances in neural information processing*, (1):556–562.

[71] Lee, H., Kim, Y.-D., Cichocki, A., and Choi, S. (2007). Nonnegative tensor factorization for continuous EEG classification. *International journal of neural systems*, 17(4):305–17.

[72] Lee, J.-M., Qin, S. J., and Lee, I.-B. (2006). Fault detection and diagnosis based on modified independent component analysis. *AIChE Journal*, 52(10):3501–3514.

[73] Lee, M. W., Joung, J. Y., Lee, D. S., Park, J. M., and Woo, S. H. (2005). Application of a moving-window-adaptive neural network to the modeling of a full-scale anaerobic filter process. *Industrial & Engineering Chemistry Research*, 44(11):3973–3982.

[74] Lee, T. W., Girolami, M., and Sejnowski, T. J. (1999). Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural Computation*, 11:417–441.

[75] Li, S., Hou, X. W. H. X. W., Zhang, H. J. Z. H. J., and Cheng, Q. S. C. Q. S. (2001). Learning spatially localized, parts-based representation. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1(C):1–6.

[76] Li, S., Li, C., Lo Kwok-Tung, and Chen, G. (2008). Cryptanalyzing an encryption scheme based on blind source separation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(4):1055–1063.

[77] Lin, C. J. (2007a). On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 18(6):1589–1596.

[78] Lin, C. J. (2007b). On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 18(6):1589–1596.

[79] Liu, Y., Hu, N., Wang, H., and Li, P. (2009). Soft chemical analyzer development using adaptive least-squares support vector regression with selective pruning and variable moving window size. *Industrial & Engineering Chemistry Research*, 48(12):5731–5741.

[80] lldiko E. Frank and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135.

[81] Maier, A. K., Schebesch, F., Syben, C., Würfl, T., Steidl, S., Choi, J. H., and Fahrig, R. (2017). Precision learning: Towards use of known operators in neural networks. *24th International Conference on Pattern Recognition*, pages 183–188.

[82] Ni, W., Tan, S. K., Ng, W. J., and Brown, S. D. (2012). Localized, adaptive recursive partial least squares regression for dynamic system modeling. *Industrial & Engineering Chemistry Research*, 51(23):8025–8039.

[83] Nigam, K. and Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93.

[84] Owhadi, H. (2015). Bayesian numerical homogenization. *Multiscale Modeling & Simulation*, 13(3):812–828.

[85] Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.

[86] Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413.

[87] Plumbley, M. D. (2003). Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3):534–543.

[88] Plumbley, M. D. (2005). Geometrical methods for non-negative ica: Manifolds, lie groups and toral subalgebras. *Neurocomputing*, 67:161–197. Geometrical Methods in Neural Networks and Learning.

[89] Qin, B., Hu, C., and Huang, S. (2016). Target/background classification regularized nonnegative matrix factorization for fluorescence unmixing. *IEEE Transactions on Instrumentation and Measurement*, 65(4):874–889.

[90] Raissi, M. (2018). Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19(1):932–955.

[91] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017). Machine learning of linear differential equations using gaussian processes. *Journal of Computational Physics*, 348:683–693.

[92] Ratsaby, J. and Venkatesh, S. S. (1995). Learning from a mixture of labeled and unlabeled examples with parametric side information. In Maass, W., editor, *Proceedings of the eighth annual conference on Computational learning theory - COLT '95*, pages 412–417, New York, New York, USA. ACM Press.

[93] Rohatsch, T., Poppel, G., and Werner, H. (2006). Projection pursuit for analyzing data from semiconductor environments. *IEEE Transactions on Semiconductor Manufacturing*, 19(1):87–94.

[94] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3:210–229.

[95] Schachtner, R., Poppel, G., and Lang, E. W. (2010). A nonnegative blind source separation model for binary test data. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(7):1439–1448.

[96] Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

[97] Shahnaz, F., Berry, M. W., Pauca, V. P., and Plemmons, R. J. (2006). Document clustering using nonnegative matrix factorization. *Information Processing and Management*, 42(2):373–386.

[98] Shannon, T. T. and McNames, J. (2007). Ica based disturbance specific control charts. *2007 IEEE International Conference on Information Reuse and Integration*, pages 251–256.

[99] Sheta, A. F., Rausch, P., and Al-Afeef, A. (2012). A monitoring and control framework for lost foam casting manufacturing processes using genetic programming. *International Journal of Bio-Inspired Computation*, 4:111–118.

[100] Skinner, K. R., Montgomery, D. C., Runger, G. C., Fowler, J. W., McCarville, D. R., Rhoads, T. R., and Stanley, J. D. (2002). Multivariate statistical methods for modeling and analysis of wafer probe test data. *IEEE Transactions on Semiconductor Manufacturing*, 15(4):523–530.

[101] Smaragdis, P. and Brown, J. (2003). Non-negative matrix factorization for polyphonic music transcription. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*, pages 177–180.

[102] Souza, F. A., Araújo, R., and Mendes, J. (2016). Review of soft sensor methods for regression applications. *Chemometrics and Intelligent Laboratory Systems*, 152:69–79.

[103] Spur, G., Neugebauer, R., and Hoffmann, H. (2012). *Handbuch Umformen*. Carl Hanser Verlag GmbH & Company KG, Mü nchen, Germany.

[104] Spur, G. and Stöferle, T. (1985). *Handbuch der Fertigungstechnik. 2,3. Umformen und Zerteilen*. Number Bd. 2. Carl Hanser Verlag GmbH & Company KG, Mü nchen, Germany.

[105] Sra, S. and Dhillon, I. S. (2006). Nonnegative Matrix Approximation : Algorithms and Applications. *SciencesNew York*, pages 1–36.

[106] Tao, F., Qi, Q., Liu, A., and Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*, 48:157–169. Special Issue on Smart Manufacturing.

[107] Tesfamariam M. Abuhay, Sergey V. Kovalchuk, Klavdiya Bochenina, Gali-Ketema Mbogo, Alexander A. Visheratin, George Kampis, Valeria V. Krzhizhanovskaya, and Michael H. Lees (2018). Analysis of publication activity of computational science society in 2001–2017 using topic modelling and graph theory. *Journal of Computational Science*, 26:193–204.

[108] Teukolsky, W., Vetterling, W., and Flannery, B. (2007). *Gaussian Mixture Models and k-Means Clustering*. Cambridge University Press, New York, USA.

[109] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

[110] Tobin, K. W., Karnowski, T. P., Gleason, S. S., Jensen, D., and Lakhani, F. (1999). Using historical wafermap data for automated yield analysis. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, 17(4):1369–1376.

[111] Tucker, L. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

[112] Wang, J.-H., Hopke, P. K., Hancewicz, T. M., and Zhang, S. L. (2003). Application of modified alternating least squares regression to spectroscopic image analysis. *Analytica Chimica Acta*, 476(1):93–109.

[113] Wang, J.-X., Wu, J., and Xiao, H. (2017). Physics informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data. *Physical Review Fluids*, 2:1–22.

[114] Warne, K., Prasad, G., Rezvani, S., and Maguire, L. (2004). Statistical and computational intelligence techniques for inferential model development: a comparative evaluation and a novel proposition for fusion. *Engineering Applications of Artificial Intelligence*, 17(8):871 – 885.

[115] Weiderer, P., Lang, H., and Hopfensberger, P. (2017). Verfahren zum ʻuberwachen von Gussformen sowie Vorrichtung. DE Patent Application 102018211653.9.

[116] Weiderer, P., Lang, H., and Hopfensberger, P. (2019). Verfahren zum Herstellen eines Gussbauteils sowie Gussbauteil. DE Patent Application 102019100606.6.

[117] Weiderer, P., Tomé, A. M., and Lang, E. W. (2019). Decomposing Temperature Time Series with Non-Negative Matrix Factorization. *arXiv e-prints*.

[118] Wibowo, A. and Desa, M. I. (2012). Kernel based regression and genetic algorithms for estimating cutting conditions of surface roughness in end milling machining process. *Expert Systems with Applications*, 39(14):11634–11641.

[119] Xu, W., Liu, X., and Gong, Y. (2003). Document clustering based on non-negative matrix factorization. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - SIGIR '03*, page 267.

[120] Yang, J., Chen, Y., and Zhang, L. (2017). An efficient approach for fault detection, isolation, and data recovery of self-validating multifunctional sensors. *IEEE Transactions on Instrumentation and Measurement*, 66(3):543–558.

[121] Yu, J. (2012a). A bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses. *Computers and Chemical Engineering*, 41:134–144.

[122] Yu, J. (2012b). Multiway gaussian mixture model based adaptive kernel partial least squares regression method for soft sensor estimation and reliable quality prediction of nonlinear multiphase batch processes. *Industrial & Engineering Chemistry Research*, 51(40):13227–13237.

[123] Yu, J. (2012c). Online quality prediction of nonlinear and non-gaussian chemical processes with shifting dynamics using finite mixture model based gaussian process regression approach. *Chemical Engineering Science*, 82:22–30.

[124] Zass, R. and Shashua, A. (2007). Nonnegative sparse pca. *MIT Press,Advances in Neural Information Processing Systems 19*, pages 1561–1568.

[125] Zdunek, R. and Cichocki, A. (2007). Nonnegative matrix factorization with constrained second-order optimization. *Signal Processing*, 87(8):1904–1916.

[126] Zhang, Y. and Zhang, Y. (2009). Complex process monitoring using modified partial least squares method of independent component regression. *Chemometrics and Intelligent Laboratory Systems*, 98(2):143–148.

[127] Zheng, Z., Yang, J., and Zhu, Y. (2007). Initialization enhancer for non-negative matrix factorization. *Engineering Applications of Artificial Intelligence*, 20(1):101–110.

[128] Zhou, G., CICHOCKI, A., Zhao, Q., and Xie, S. (2014). Nonnegative matrix and tensor factorizations: An algorithmic perspective. *IEEE Signal Processing Magazine*, 31(3):54–65.

[129] Zhu, J., Ge, Z., and Song, Z. (2015). Robust supervised probabilistic principal component analysis model for soft sensing of key process variables. *Chemical Engineering Science*, 122:573–584.

Acknowledgements

An dieser Stelle möchte ich mich bei einigen Menschen bedanken, die mich während den letzten drei Jahren unterstützt und begleitet haben.

Großer Dank gebührt meinem Betreuer der Doktorarbeit, Prof. Elmar W. Lang, ohne dessen ausgezeichnete fachliche Betreuung und die ständige Erreichbarkeit bei allen Problemen, diese Arbeit nicht möglich gewesen wäre.

Ebenso will ich mich bei allen Mitgliedern der AG Lang für das angenehme Arbeitsklima und die vielen fruchtbare Diskussionen der letzten Jahre bedanken. Dank gebührt auch Prof. Ana Maria Tomé, deren fachlicher Rat bei meiner Arbeit und den Publikationen eine große Hilfe war.

Desweiteren bedanke ich mich bei den vielen Kollegen der Firma BMW in Landshut mit denen ich in den letzten drei Jahren zusammen gearbeitet habe und die mich unterstützt haben. Dabei bedanke ich mich natürlich bei meinem Betreuer von der Firma BMW, Hubert Lang, dessen Engagement und wertvollen Ratschläge mir während meiner Zeit in Landshut vieles erst ermöglicht haben. Bei meinem Abteilungsleiter, Jean-Marc Ségaud, bedanke ich mich für die Unterstützung bei vielen organisatorischen Problemen, die vielen hilfreichen Anregungen und natürlich auch für die großen Freiheiten, die es mir ermöglichten viele eigene Ideen zu verfolgen.

Der größte Dank an dieser Stelle geht jedoch an meine Eltern, auf deren Hilfe und Unterstützung in sämtlichen Lebenslagen Verlass ist und ohne die mein Werdegang nicht möglich gewesen wäre.