Anomaly detection for building systems using low-rank multilinear model identification

Vorgelegt im Promotionsausschuss der
HafenCity Universität Hamburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

Dissertation

von
Leona Schnelle

aus
Neumünster

2026

Impressum

Anomaly detection for building systems using low-rank multilinear model identification
Leona Schnelle

1. Gutachter : Prof. Dr.-Ing. Harald Sternberg
2. Gutachter : Prof. Dr.-Ing. habil. Gerwald Lichtenberg

Zusätzlicher Professor der Prüfungskommission: Prof. Dr.-Ing. Youness Dehbi
Vorsitzender der Prüfungskommission: Prof. Dr.-Ing. Ingo Weidlich

HCU-Fachgebiet: Hydrographie und Geodäsie

Tag der mündlichen Prüfung: 25.02.2025
Veröffentlicht im Januar 2026

HafenCity Universität Hamburg
Henning-Voscherau-Platz 1
20457 Hamburg

DOI: 10.34712/142.79

Kurzzusammenfassung

Die Motivation für diese Arbeit ergab sich aus der Tatsache, dass der Energieverbrauch in Gebäuden im Vergleich zu anderen Sektoren hoch ist und Gebäude aufgrund verschiedener Fehler oft nicht optimal betrieben werden. Gerade in Nichtwohngebäuden, die mengenmäßig nur einen geringen Anteil am gesamten Gebäudebestand ausmachen, ist der Anteil am Endenergieverbrauch überproportional hoch. Durch die zunehmende Digitalisierung stehen zwar Messdaten zur Verfügung, diese werden aber noch nicht standardmäßig mit automatisierten Methoden genutzt, um Verbesserungspotenziale im Gebäudebetrieb zu identifizieren.

Ziel dieser Dissertation ist es daher, ein Verfahren zur Anomalieerkennung mit hohem Automatisierungsgrad zu entwickeln. Ein modellbasierter Ansatz mit Parameteridentifikation von multilinearen zeitinvarianten Black-Box Modellen soll Veränderungen in der Gebäudedynamik durch Klassifizierung der Parameterräume in Sollverhalten und auftretende Anomalien detektieren. Die multilineare Modellstruktur ist in der Lage, die Dynamik von thermischen Energiesystemen abzubilden, deren physikalische Gleichungen multilineare Terme enthalten, und ist daher für die Gebäudemodellierung geeignet. Darüber hinaus ermöglicht die multilineare Struktur der Modelle die Abbildung in mehrdimensionalen Datenstrukturen, den Tensoren. Dies ermöglicht den Einsatz von Tensordekompositionsverfahren, die durch Faktorisierung der Modelle zu einer signifikanten Reduktion des Speicherbedarfs führen.

Der entwickelte Algorithmus überwindet die komplexe White-Box Modellierung einzelner Nichtwohngebäude durch die Verwendung von multilinearen Black-Box Modellen, die aus Messdaten geschätzt werden, und ist somit auf andere Gebäude übertragbar, deren Messdaten in Zeitreihen vorliegen. Durch Normalisierungsverfahren der Modellparameter, wird eine eindeutige Darstellung der Modelle erreicht, die eine Auswertung der Modellparameter zur Anomalieerkennung ermöglicht. Weiterhin wird durch die Normalisierung der Modellparameter eine Interpretierbarkeit erreicht, indem der Einfluss eines Signals auf den aktuellen Zustand des Modells direkt an dem Wert eines Parameters abgelesen werden kann. Die Skalierbarkeit auf große Gebäude ist durch die Reduktion der Modelle durch die Tensordekompositionsverfahren in Verbindung mit der Normalisierung gegeben, da diese zu einer effizienten Speicherung und Berechnung führt.

Als Methoden zur Parameteridentifikation werden die Schätzung der Modellparameter in voller Tensordarstellung mit der direkten Schätzung der dekomponierten normalisierten Parameter verglichen. Die Schätzung der vollen Tensoren hat den Vorteil, dass in dieser Darstellung die Kostenfunktion in den Parametern linear ist und somit effiziente Standardmethoden der linearen Algebra verwendet werden können. Allerdings ist dieser Weg nicht skalierbar, da jedes zusätzliche Signal zu einem exponentiellen Anstieg der Modellparameter führt, was die Anwendung auf Gebäudedaten unmöglich macht. Der direkte Weg zur Identifikation der reduzierten Modellparameter in dekomponierter und normalisierter Form, wie sie später zur Anomaliedetektion verwendet werden, weist ein nichtlineares Optimierungsproblem auf. Ein neuer iterativer Optimierer, der speziell für die faktorisierte multilineare Modellstruktur entwickelt wurde und effiziente Least-Squares-Methoden verwendet, liefert gute Ergebnisse sowohl in Bezug auf die benötigte Rechenzeit als auch auf den erreichten Fehler zwischen Modell und Messdaten.

Zur Anomalieerkennung werden die mit aktuellen Messdaten identifizierten Modellparameter mit Hilfe von Klassifizierungsmethoden in nominales und anomales Verhalten getrennt. Hierzu sind klassifizierte Trainingsdaten erforderlich, mit Hilfe derer der Klassifizierer trainiert wird. Das Einstellen der Empfindlichkeit des Klassifizierungsmodells ist durch zwei freie Parameter möglich, wobei weniger unerkannte Anomalien mit mehr Fehlalarmen einhergehen. Der Algorithmus zur Erkennung von Anomalien wird mit verschiedenen Datensätzen von Systemen zur Heizung, Lüftung und Klimatisierung evaluiert. Hierbei zeigen sich gute Ergebnisse für Anomalien, wie geöffnete Fenster, festhängende Lüftungsklappen und zugesetzte Ventile.

Abstract

The motivation for this work arose from the fact that the energy consumption in buildings is high compared to other sectors, and that buildings are often not operated optimally due to various deficiencies. In particular, the share of final energy consumption is disproportionately high in non-residential buildings, which represent only a small part of the total building stock. Although measurement data are available due to increasing digitalization, it is not yet used as a standard with automated methods to identify potential for improvement in the building operation. The aim of this thesis is therefore to develop a method for anomaly detection with a high degree of automation. A model-based approach with parameter identification of multilinear time-invariant black-box models is used to detect changes in building dynamics by classifying the parameter spaces into nominal behavior and anomalies. The multilinear model structure is capable of representing the dynamics of thermal energy systems whose physical equations contain multilinear terms and is therefore suitable for building modeling. In addition, the multilinear structure of the models allows them to be mapped into multidimensional data structures called tensors. This enables the use of tensor decomposition methods, which lead to a significant reduction in memory requirements by factorization.

The developed algorithm overcomes the complex white-box modeling of individual non-residential buildings by using multilinear black-box models estimated from measured data and is therefore transferable to other buildings for which measured time-series data are available. By normalizing the model parameters, a unique representation of the models is achieved, allowing the model parameters to be evaluated for anomaly detection. In addition, the normalization of the model parameters provides interpretability in that the influence of a signal on the current state of the model can be read directly from the value of a parameter. Scalability to large buildings is achieved by reducing the models using tensor decomposition methods in conjunction with normalization, resulting in efficient storage and computation.

For parameter identification, the estimation of the model parameters in the full tensor representation is compared to the direct estimation of the decomposed normalized parameters. The estimation of full tensors has the advantage that in this representation the cost function with respect to the parameters is linear and thus efficient standard linear algebra methods

can be used. However, this method is not scalable, as each additional signal leads to an exponential increase in the model parameters, making it impossible to apply to building data. The direct way to identify the reduced model parameters in decomposed and normalized form, as used later for anomaly detection, presents a nonlinear optimization problem. A new iterative optimizer, specifically developed for the factorized multilinear model structure and using efficient least squares methods, provides good results both in terms of computational time and error between model and measured data.

For anomaly detection, the model parameters identified with current measurement data are separated into nominal and anomalous behavior using classification methods. This requires classified training data, which is used to train the classifier. Two free parameters can be used to tune the classification model between undetected anomalies and false alarms. The anomaly detection algorithm is evaluated with different data sets from HVAC systems. This shows good results for anomalies such as open windows, stuck dampers, and blocked valves.

Danksagung

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Harald Sternberg für die Übernahme der Betreuung meines Promotionsverfahrens. Die wiederkehrenden Gespräche brachten immer wieder neue wichtige Anregungen und haben dadurch entscheidend zur Weiterentwicklung dieser Arbeit beigetragen. Ebenso danke ich Herrn Prof. Dr.-Ing. habil. Gerwald Lichtenberg für die Betreuung an der HAW. Der stetige Austausch führte zu wertvollen Diskussionen, die wichtige fachliche Erkenntnisse lieferten. Zusätzlich trugen die häufigen, anregenden Gespräche dazu bei, die Motivation über die gesamte Dauer der Promotion aufrechtzuerhalten.

Darüber hinaus gilt ein ganz besonderer Dank der HAW/ILES-Gruppe. Die wöchentlichen Treffen eröffneten neue Perspektiven und bereicherten meine Arbeit insbesondere durch Ansätze angrenzender Anwendungsgebiete. Besonders hervorheben möchte ich Georg, Carlos und Thorsten, die mir in der intensiven Endphase stets fachlich beratend zur Verfügung standen. Ebenso wichtig war für mich Marah, die zusätzlich zur fachlichen Unterstützung auch eine wichtige Freundin geworden ist.

Bedanken möchte ich mich ebenfalls beim Physik-Team für die schöne Arbeitsatmosphäre im Büro, die den wissenschaftlichen Alltag positiv geprägt hat.

Abschließend möchte ich mich ganz besonders bei meiner Familie, meinen Eltern, Geschwistern sowie meinen Kindern und meinem Mann für ihre bedingungslose Liebe und Unterstützung danken. Ohne ihren Rückhalt wäre die Realisierung dieser Arbeit nicht möglich gewesen.

Contents

List of Figures

List of Tables

List of Abbreviations

AHU	air handling unit
ALS	alternating least squares
BMS	building management system
BUDO	Buildings Unified Data point naming schema for Operation management
CP	Canonical Polyadic
CPN	normalized CP decomposed
HVAC	Heating, Ventilation and Air Conditioning
LS-SVM	Least Squares Support Vector Machines
MTI	multilinear time-invariant
NLS	natural language processing
OA	outdoor air
RA	return air
rbf	radial basis function
S/N	signal-to-noise ratio
VAV	variable air volume

Chapter 1

Introduction

1-1 Motivation

The final energy demand for heat generation accounted for almost 50 % of Germany's total final energy demand in 2022 and is therefore a significant sector in the topic of climate protection by energy conservation. Over 35 % of the heating demand was consumed for space heating and hot water production in buildings and for lighting. The German government has set targets to reduce the energy demand of buildings and to switch to sustainable energy sources, in order to achieve a climate-neutral building stock by 2050, see BMWK (2022); Bürger et al. (2020).

The existing energy saving potential in the area of energy supply to buildings could be better exploited by monitoring building energy systems. There are already many modern buildings that collect and store large amounts of data. However, these are rarely used to optimize the energy performance of the buildings. The goal of this research is to develop a new method for anomaly detection in building energy systems, which detects anomalies in the ongoing operation of the buildings based on available measurement data in order to identify optimization potential in the energy supply. The following key questions arise:

- How can the structure of a mathematical building model be generated automatically on the basis of measured data?
- How can the parameters of this structured model be determined?
- How can anomalies in building operation be detected with the help of a model identified in this way?

For this purpose, the mathematical model structure of the building is generated in the form of parameter sets. If anomalies occur during operation, the continuously generated parameter sets will deviate from those of error-free operation and will be detected. The very time-consuming and costly manual modeling is not necessary due to this generic approach of anomaly detection from measured data, which is a major advantage of this method.

Within the framework of the EU project *mySmartLife*, which is currently being carried out in Hamburg, research is being conducted into solutions for smart cities with a focus on the digitalization of energy supply, Pajares (2024). In the course of this, a platform for storing and evaluating large amounts of data from energy systems is planned. Due to the increasing digitalization of the building sector and the associated storage in data centers, large amounts of data (Big Data) are accumulating on servers, which can only be used meaningfully by automated evaluations. The BMBF-funded third-party project *SONDE* addresses the potential for energy savings through supervision and optimization of buildings during operation, (Schnelle and Lichtenberg, 2024). Through the project partners, access to the building control system (BMS) of buildings is possible, so that large amounts of data are available for the evaluation of developed methods of this thesis. In addition, it is possible to artificially generate data sets under predefined conditions using test rooms equipped with measurement technology.

1-2 State of research

In the following section, an overview of previous research in the topics relevant to this PhD thesis on tensors and decomposition methods, multilinear systems, and anomaly and fault detection is given. These findings are used, linked and extended within this PhD thesis.

Buildings can be considered as thermal energy systems whose dynamic behavior is described by thermal power balances, in which polynomials (e.g., of temperatures and volume flows) occur. This dynamic behavior is nonlinear and can therefore only be represented with sufficient accuracy by linear models in the vicinity of the operating point. Therefore, linear models are not suitable as global models. State space models describe the dynamic behavior of systems with the help of differential or difference equations. The class of multilinear state-space models extends the linear ones in that in the first order differential equations, in addition to linear terms, multiplications of states and inputs (e.g. temperatures and volume flows) are also allowed, (Pangalos et al., 2015). This structure fits the described heat flow balance of heating systems. In Pangalos et al. (2013) it was shown that thermal energy systems can be classified in the class of multilinear time-invariant (MTI) models. As shown in Figure 1-1, a model class is available which allows further structures than the linear model class, but cannot be arbitrarily complex as nonlinear systems can be. The parameters of the multilinear models can be represented as tensors, i.e. multidimensional arrays, due to their multilinear structure. In order to easily account for the individual differences of buildings in modeling, component-based approaches to model heating systems were developed, (Pangalos, 2016).

The number of parameters of MTI models increases exponentially with each additional signal (input, state), so that models of large systems with many signals can no longer be meaningfully represented with full rank of the underlying tensor. Approximation methods are necessary to reduce the number of parameters. Tensor decomposition methods are ideal for this purpose, since they preserve the multilinear structure with a reduced number of parameters and an adjustable accuracy. In Kruppa (2018), it is investigated how tensor decomposition methods can be applied to MTI models and revealed their benefits on controller design. Here, it was shown that applying tensor decomposition methods to MTI models led to a reduction in memory requirements. In addition, a Matlab library *MTI Toolbox* was

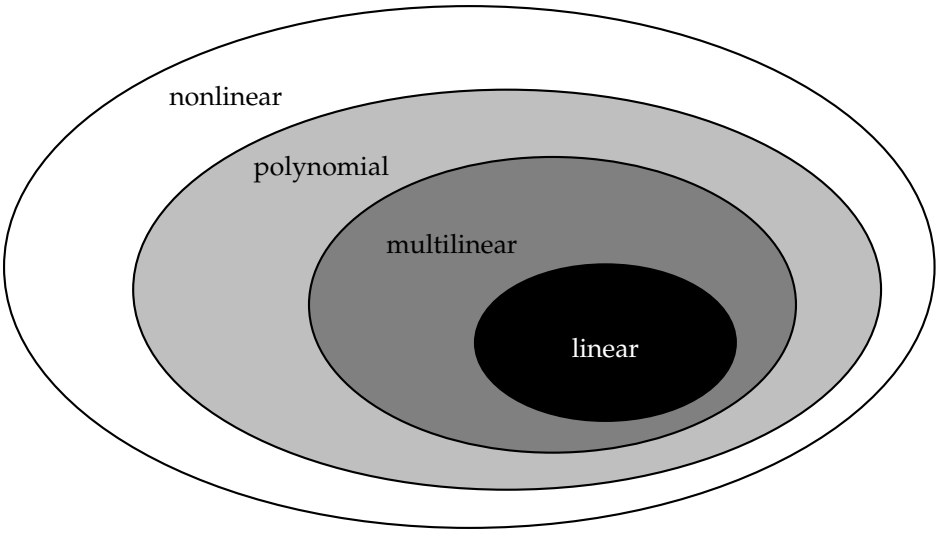

Figure 1-1: Presentation of the different model classes, see Kruppa (2018)

developed to provide functions for model building and simulation of MTI models in tensor decomposition, (Kruppa, 2017b).

In recent years, tensors and tensor decomposition have become an increasing focus of research. In Kolda and Bader (2009), Canonical Polyadic decomposition and Tucker decomposition were introduced and studied in various mathematical applications. Numerical algorithms for using tensor decompositions have been developed and made available in Matlab libraries. Other tensor decomposition methods such as Tensor Train Decomposition and Hierarchical Tucker Decomposition have been presented and studied in Oseledets (2013) and Grasedyk et al. (2013).

Anomaly and fault detection methods are widely used today. In Isermann (1997) different methods for parameter-based and non-parameter-based fault detection are summarized, which can be applied to linear and nonlinear models. Especially in the automotive industry, reliable methods for automated fault detection have existed for a long time. There is also a trend toward increased use of model-based fault diagnosis algorithms in the building sector. For example, in Sewe (2018) fault detection in building systems using parity equations was investigated. In recent years, the development of neural networks, especially for pattern recognition, has increased in many areas of research and application. Neural networks for anomaly detection in building systems have been discussed in Borda et al. (2023) and Himeur et al. (2023), but the challenge of interpretability and explainability of the decision generated by artificial intelligence is a common issue, (Carabantes, 2020; De Bruijn et al., 2022) with which many research papers are concerned, but is hardly found for anomaly detection in building systems yet, (Gugliermetti et al., 2024).

First applications of fault diagnosis with tensor decomposition methods were performed with model-based approaches in Müller-Eping (2020) using qualitative models. Qualitative models can be used for fault detection, but they also have disadvantages, e.g. a high memory requirement, Müller-Eping (2020). Signal-based methods with tensorized monitoring data and model-based methods for fault diagnosis using multi-linear models and linear subspace-based parameter identification of building energy systems for fault detection have been

compared in Sewe et al. (2019).

Based on the existing research on CP decomposition of MTI models and their application to building energy systems, the resulting advantages will be exploited. In particular, the reduction in memory and concomitant reduction in computational effort through the application of tensor decomposition methods is essential for large building systems. Likewise, the assignment to the class of multilinear systems allows a sufficiently accurate representation without too much complexity. In contrast to previous work, the focus shall be on the development of algorithms for fault detection of building energy systems and their automated application to different buildings without complex modeling of the components. Black-box models without structural information will be developed through parameter identification methods and these will be made available through anomaly detection algorithms to optimize buildings.

1-3 Research questions

The scientific interest is focused on the development of new tensor-based methods for fault and anomaly detection of building energy systems based on measurement data, which could later be used to optimize the energy demand of buildings. In order to ensure the broad applicability of such methods, their suitability for practical use should already be taken into account during their development. As little prior information about the building data as possible should be required, as the manual input of this data is always costly. This brings the class of so-called black-box identification methods into focus, which, for the reasons mentioned above, will be supplemented in this thesis by another method for multilinear modeling.

For this purpose, MTI models in decomposed tensor representation are to be used. The memory demand reduction shown in other works with accompanying reduction of computational requirements by using decomposed models is very attractive in the case of large data sets. When modeling complex building energy systems as a non-decomposed MTI model, the number of model parameters in full tensor structures increases exponentially with each added state or input.

The anomaly detection methods developed in this work shall be based on parameter identification methods of black-box models. Black-box modeling refers to model building without structural information of the buildings. Using given input signals and measured values of the output signals, the dynamics of the system is identified by parameter identification. For this purpose, the output variables of the applied models are compared with the measured values from the buildings and the residuals are minimized by an optimization algorithm. The parameter identification methods will be applied to multilinear structures in decomposed tensor form to produce a nominal parameter set in Canonical Polyadic (CP) decomposition. For anomaly detection, it is to be tested whether deviations from the nominal state can be detected using the decomposed parameter sets. If it is possible to successfully apply this method for anomaly detection of the available demo buildings, it can be transferred to any other buildings without much effort due to the choice of black-box modeling, without structural information of the buildings. This is a major advantage over other methods that require elaborate modeling of individual components.

Since there are no studies on the optimization problem of parameter identification of reduced MTI models for anomaly detection, it must first be investigated which class of mathematical optimization problems this parameter identification problem belongs to. For example, if the optimization problem of parameter identification of MTI models were convex under certain conditions, global minima could be found numerically reliably and easily, and existing very effective methods as in Boyd and Vandenberghe (2009) could be used for the solution. This in turn would bring advantages by reducing computation time and memory requirements. If this is not the case, an optimizer must be found that can handle the multilinearity of the models and is also computationally efficient.

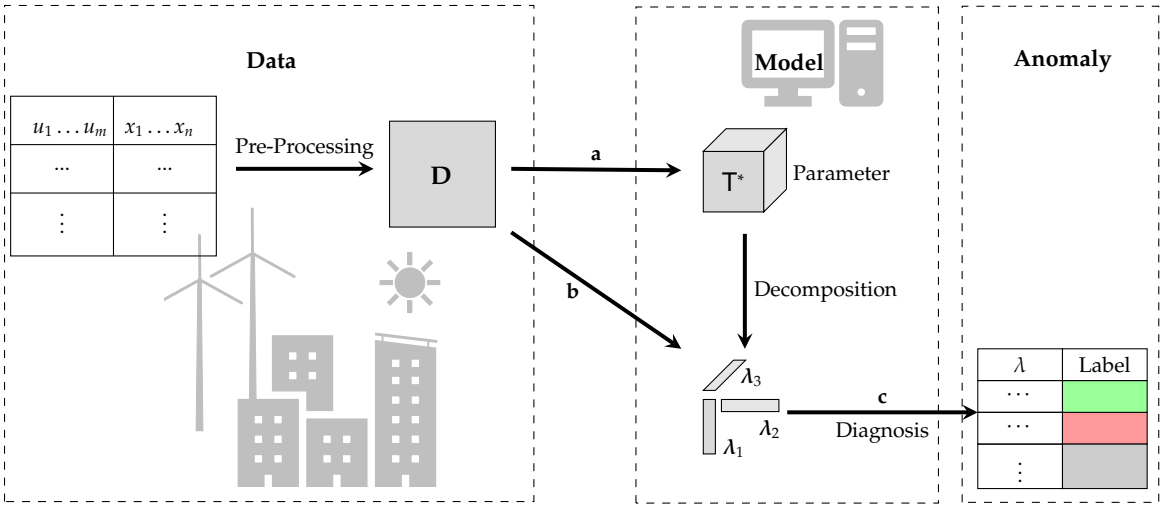

Figure 1-2: Anomaly detection with multilinear parameter identification from data, own illustration

Figure 1-2 illustrates two principal approaches to anomaly detection through parameter identification of MTI models, starting from original measurement data stored in a data matrix **D**. In path a, a full-rank MTI model T in tensor representation is estimated from the measurement data and subsequently decomposed using CP decomposition. Path b represents a direct approach. It uses the data matrix to estimate the decomposed parameters of the MTI model directly, expressed as factor vectors λ with a predefined rank-r. Path c describes the process of computing a diagnostic decision in building operation by analyzing the decomposed parameters.

The knowledge objectives justified in this section can be formulated as the following research questions:

a) How can an MTI model T be computed from the data matrix **D**?

b) How can a decomposed model (λ) be computed directly from the data matrix **D**?

c) How can the parameters of the decomposed model be implemented and applied for automated anomaly detection in building energy systems?

1-4 Outline

The thesis is structured as follows: In Chapter 2 fundamentals are given. There are introductions to methodological rudiments as anomaly detection methods and existing parameter identification methods followed by an introduction to tensors, tensor decomposition and the MTI models in this tensor representation. At the end of Chapter 2, information about modern buildings and their data collection structure is given, before a simple mathematical model for buildings is derived. In Chapter 3 a new reduced normalized modeling approach is developed from polynomials and adapted to MTI models. The normalized MTI models are given in full representation and in a structured sparse representation, which allows a further reduction of the storage amount. Finally, methods for transformation such as scaling and discretization are given. In Chapter 4 different approaches for parameter identification of the new normalized MTI model framework are derived. Here, a new specialized alternating least squares algorithm for parameter identification of normalized MTI models is developed and tested. In Chapter 5, an algorithm for anomaly detection using the model parameters as residuals and evaluating them with classification techniques is derived and tested on a sample data set of supervised data. In the following application chapter Chapter 6 the developed algorithm for automatic anomaly detection is evaluated with five different data sets of clustered simulation and real measurement data with anomalies in the HVAC system of buildings followed by an unsupervised data set of an office building. The summary and outlook of this thesis are given in Chapter 7.

Chapter 2

Fundamentals

In the rapidly evolving field of building management, the ability to detect anomalies from measurement data has become a critical component of efficient and safe operations. This chapter provides a comprehensive overview of the diverse landscape of anomaly detection methods in building systems, spanning traditional manual approaches, rule-based systems, and sophisticated automated techniques. Among automated techniques, a distinction exists between theory-based methods and data-driven approaches. The focus here is primarily on data-driven approaches that generate models from empirical data, as this should also form the basis for the algorithm proposed later for recognizing anomalies. These models necessitate robust parameter identification techniques. To establish a solid foundation for understanding these techniques, existing parameter identification methods will be introduced, providing essential context for data-driven anomaly detection. Subsequently, attention will shift to a specific model class utilized later in this work: the multilinear time-invariant (MTI) model. The basic principles of MTI are introduced, which sets the stage for understanding the idea of how to use these models for effective anomaly detection in building systems.

2-1 Methodological rudiments

In this section, the methodological foundations necessary for the development of an anomaly detection algorithm for building systems using multilinear parameter identification are presented. Therefore, the current state of anomaly detection in building systems is presented first, followed by existing parameter identification methods. At the end of the section, tensors and decomposition methods are introduced for the subsequent model approach with MTI models. This knowledge serves as a basis for further concepts and applications, which are dealt with in the following chapters.

2-1-1 Anomaly detection

Fault and anomaly detection for diagnostics are broad fields in engineering. However, in the past the main focus was on safety-related applications, such as in the automotive industry,

in aircraft operations and in safety-related process engineering. Nowadays, more attention is being paid to energy-related issues, which means that diagnosis during operation is also becoming increasingly important in building automation. Particularly in the design phase of buildings, a great emphasis is placed on energy efficiency. However, the planned values for energy efficiency in modern office buildings are not achieved during operation, as the building is not operated at the optimum level, (David et al., 2017). This is where anomaly detection becomes important in order to detect and avoid problems in building operation.

What are anomalies?

A definition of the term 'anomaly' can be found in Chandola et al. (2009). There an anomaly is described as any pattern in the data that does not fall within a previously defined range of normal behavior. It is important for anomaly detection that the deviating behavior is of interest to the analyst. Therefore, anomaly detection is distinguished from noise detection. Noise refers to deviations in the measured data that interfere with the further processing and analysis of the data, but do not cause a real anomaly or error in the system and are therefore not of interest in anomaly detection. Various challenges arise in anomaly detection, which mainly result from the fact that a normal behavior of the system must first be defined, in order to be able to recognize the anomaly later as a deviation from normal behavior. Normal behavior can lie in a wide range, and the boundaries between normal and abnormal behavior cannot be precisely separated. In addition, the definition of normal behavior varies greatly from application to application, so that algorithms developed to detect anomalies cannot be directly transferred to other applications. Furthermore, the availability of data labeled as normal and abnormal is often difficult, which makes training and validation of the algorithms difficult. It can also happen that noise in the measured values is similar to an anomaly when analyzing the data, which can lead to false alarms. Depending on the availability of training data, different anomaly detection modes can be operated. Supervised anomaly detection can be performed if labeled data is available for both the nominal and abnormal building operation. If labeled data is available only for the normal behavior, but not for the abnormal behavior, the technique is called semi-supervised. For the third case, where the analyst is faced with a lack of any labeled data, unsupervised techniques are needed, (Chandola et al., 2009).

Anomaly detection points to abnormal behavior, which can be a technical fault somewhere in the building, similar to fault detection or other undesirable behavior. Anomalies can also occur due to manual intervention by users in the operation of the building. This is different from fault isolation, where the location of the fault is isolated. Fault identification goes even further by determining the size of the fault, (Isermann, 1994). Anomaly detection does not point directly to the faulty component, but is mostly easier and faster. It helps to find symptoms that can be used for further diagnosis or investigation, (Shi and O'Brien, 2019).

The different approaches to anomaly and fault detection can be found in Figure 2-1, showing methods listed in Neumann et al. (2011), Rehault et al. (2015) and Isermann (2005). As shown, they can be divided into signal-based and model-based techniques. Most of the techniques find application in fault detection and anomaly detection problems for different engineering domains. They are described in the following section, with application examples for anomaly detection in building systems.

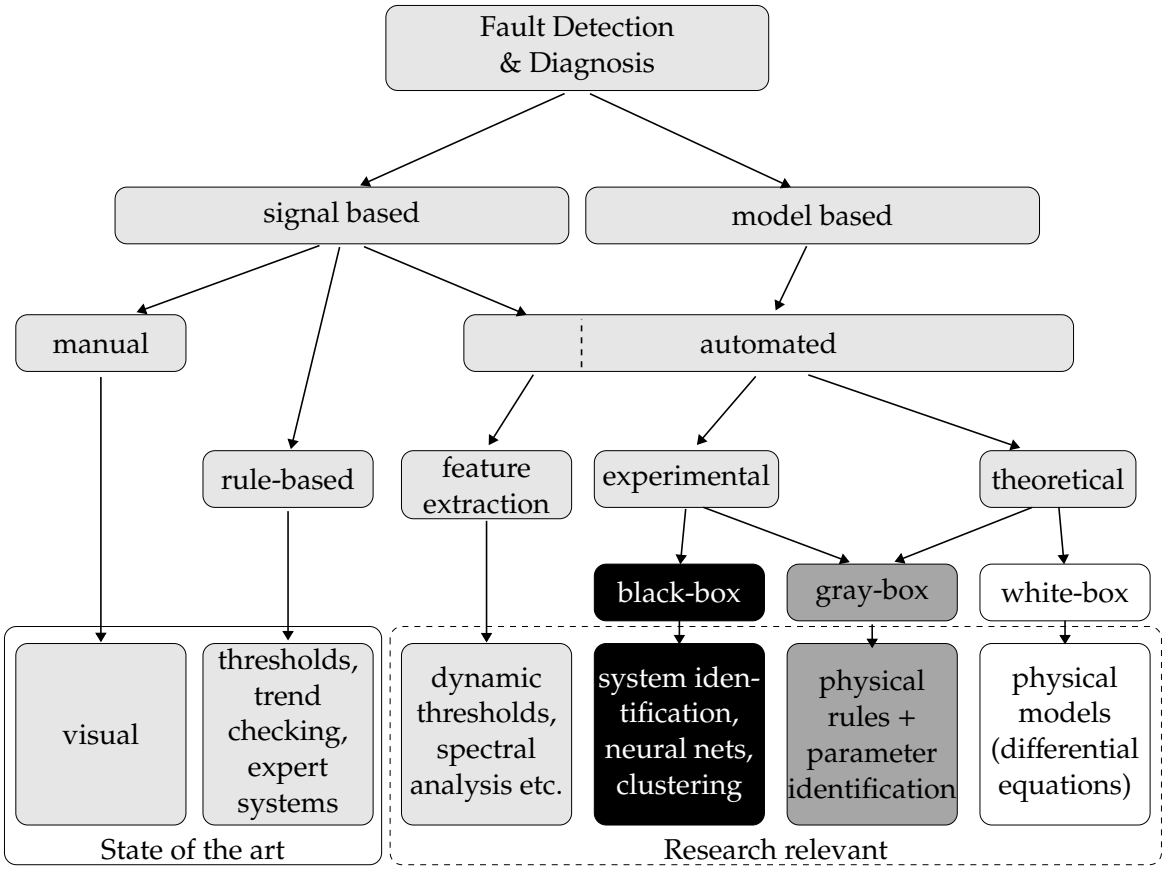

Figure 2-1: Anomaly and fault detection methods, adapted from Müller-Eping (2020)

Signal-based methods

The primary distinction is that model-based approaches rely on input-output models of the system, while signal-based techniques utilize measurement data directly for diagnostic decisions. In signal-based methods, the symptoms of occurring faults are reflected in the measurement data, allowing fault decisions to be made based on these symptoms and prior knowledge of the fault-free system (Gao et al., 2015). In the field of building automation, the state of the art in modern buildings with monitoring capabilities primarily comprises the first two categories: manual and rule-based anomaly detection. These techniques can be named as simple signal-based approaches. The manual methods involve visual monitoring of measured signals through graphical representation and rule-based use simple rules. Some examples which are described in detail in Müller-Eping (2020), are summarized below.

The collected measurement data can be plotted over time using *time series plots*. By visually examining these time series, expert knowledge can be used to identify significant deviations of the measured variable from the expected values. Typically, these evaluations are suitable for smaller measurement series and a few data points, as they quickly become confusing with large data sets. Possible errors that can be detected are those that have a direct effect on the values of a measured variable. These include implausible measurements due to sticking or failed sensors, or atypical measurement ranges due to missing sensor calibration.

Unlike simple time series plots, *carpet plots* can be used over long periods of time. The strength of this type of plot lies in its ability to visualize patterns in the measured variable under investigation. In building monitoring applications, carpet plots can be used to display temperatures as a function of date and time, for example. Temperatures are typically displayed as a third dimension with a color legend. This results in patterns that can be quickly recognized and can give an indication of the thermal comfort of a room over the year.

Another visual method is the visualization in so-called *scatter plots*. Here, the measured values are not plotted over time, but against each other. This results in point clouds that can be analyzed manually. If, for example, the outdoor temperature is plotted against the required heating power, the result is a scatter plot that would probably show that higher heating power is required at lower outdoor temperatures. With these scatter plots and expert knowledge, bands of typical behavior can be generated so that abnormal behavior outside these defined bands can be detected, (Müller-Eping, 2020).

In Müller-Eping (2020), the advantages and disadvantages of the methods described for manual and visual monitoring for error detection are mentioned. The advantage is that the methods are easy to use, as implementation does not require extensive programming knowledge. Additionally, the evaluation of the plots is intuitive, but it requires a high level of system knowledge. Furthermore, for more complex buildings with many data points, the methods are time-consuming and therefore difficult to implement for visually analyzing all data. One limitation of visual evaluation is that errors affecting system dynamics and that cannot be directly read from individual values or data curves may go unrecognized. These include errors that change the time constants of the buildings. An example of such an error is a changing slope of the cool-down curve, in response to certain input signals.

In addition to the manual methods, as shown in Figure 2-1, there are also automated methods for anomaly and fault detection in addition to the purely manual, visual methods. Among the methods that can be assigned to the state of the art, however, are only the rule-based methods. These include pure *limit checking* and expert systems. Limit checking is a commonly used method where thresholds are defined inside controllers to define a range for each data point, within which the measured values should move in the nominal case. An error is triggered automatically above and below these limit values. For critical failures, limit checking performs well, e.g. for fire alarms or carbon monoxide alarms, but has difficulties in other areas, (Shi and O'Brien, 2019). As an example, an outdoor temperature sensor is listed in Müller-Eping (2020), whose values typically only move within plausible limits for a specific location. If these deviate significantly, it can be assumed that there is an error within the temperature sensor. It is clear from this example that simple limit values cannot always be applied, as there can be large differences in the nominal ranges over the seasons, making it difficult to detect an actual error. Similar to limit checking, trend checking is easy to implement, as it uses the first derivative of the measurement to define nominal trends for the measurement data. For many applications, simple rules for limit value monitoring and trend checking are not sufficient, so these are supplemented by so-called expert systems. This is done by implementing conditions in a form that is known as "if-then-else" formulations. As an example of this, we can consider the temperature within a boiler. Limit values for the temperature must be set differently in heating mode than in night setback or summer operation. Rules are required here that take into account the operating states (heating mode, cooling mode, etc.) and the times of use.

More advanced methods are not common in the field of building systems and can therefore be regarded as research-relevant methods. These include model-based methods, such as white-box and black-box modeling, and advanced signal-based methods. Signal-based methods are available for the time domain and for the frequency domain. In the time domain common methods are to investigate the features of the signals by analyzing the mean, standard deviation, trends, peaks, slope, magnitude or phase of the data, while in the frequency domain a common practice is to use spectrum analysis, such as discrete Fourier transform. Both approaches are already known and used for real-time applications in power converters, mechanical components of systems or asynchronous motors as a comprehensive review by Gao et al. (2015) shows, but are, so far, less common in building technology.

Model-based methods

This section focuses on model-based methods, as a basic overview of model-based methods for diagnosis and fault detection is necessary to understand the anomaly detection approach developed later in this thesis. For model based fault or anomaly detection a dynamic input-output model of the system or process is needed, which represents the system behavior. If models are available, Gao et al. (2015) gives different approaches for the model based fault detection. One is a deterministic method, where the constructed model is simulated in parallel with the real system and receives the same input. The output of the system is then compared with the output of the model and the residual signal r is built from the output deviations. The diagnosis decision is then made on the basis of these residuals. Similar to this approach, observer based fault detection uses an observer, who is running in parallel to the system. The error between the model and the system output is fed back to the observer. The observer can be designed, such that the residual signal for fault detection is more sensitive to faults than to measurement noise and disturbances, (Frank and Wunnenberg, 1989; Isermann, 2006; Zhang et al., 2016). Besides deterministic fault detection, statistical fault diagnosis methods are developed, where a Kalman filter is constructed similar to the observers. Here the corresponding residual signal is investigated with statistical methods, such as generalized likelihood, χ^2 testing and hypothesis test to check the likelihood or probability of an occurred fault, (Gao et al., 2015).

Another approach was first proposed by Baskiotis et al. (1979) and is based on system identification. The idea is that the model parameters of the nominal behavior differ from those of the faulty behavior. Therefore, the model structure must first be known as well as the parameters of the nominal behavior. The actual parameters are then estimated online during the operation of the system. The difference between the actual obtained parameters, and the nominal system parameters is then used as residuals for fault diagnosis. This method is reviewed in Isermann (2005); Simani et al. (2013) and is especially straightforward, if the model parameters are directly related to physical coefficients.

Other fault diagnosis methods that have been described in recent years for building applications are summarized in Rogers et al. (2019) and Kim and Katipamula (2018). The generation of residuals with parity equations should also be added, which provides a simple option for fault detection, especially for linear state-space models. It was described in Isermann (2006) and used in Sewe (2018) in an application to fault detection in buildings. Qualitative fault detection in building systems was applied in Müller-Eping (2020) using stochastic automata,

dealing with the evaluation of discrete probability distributions of quantized signals. What all model-based fault diagnosis methods have in common is that they first require a model of the occupied process or system. As shown in Figure 2-1, the model-based methods can be divided into white-box, gray-box and black-box methods. This distinction is made on the basis of the different procedures for creating the models.

White-box methods for fault detection use, for example, linear or nonlinear state-space models of the system, which are based on physical equations that describe the behavior of the system. These equations are typically differential or difference equations and form an equation system where the parameters represent physical parameters such as material properties like density, heat capacity, resistance or dynamic coefficients like time constants. In general, a white-box representation

$$\dot{\mathbf{x}} = \boldsymbol{f}(\mathbf{x}, \mathbf{u}), \tag{2-1}$$

$$\mathbf{y} = \boldsymbol{g}(\mathbf{x}, \mathbf{u}), \tag{2-2}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{2-3}$$

describes the system dynamics through the state transition vector function $\boldsymbol{f}$. In this way, the behavior of the state derivative $\dot{x}$ can be expressed as a function of the input vector $\mathbf{u}$ and the state vector $\mathbf{x}$. The output vector function $\boldsymbol{g}$ describes the behavior of the output vector $\mathbf{y}$ depending on the state vector and the input vector. All signals are time-dependent, (Lunze, 2014). A linear system in state-space representation therefore has a linear state and output equation with

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{2-4}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \tag{2-5}$$

$$\mathbf{x}(0) = \mathbf{x}_0, \tag{2-6}$$

where parameter matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ contain the model dynamics.

A more general representation can be achieved by multilinear models, which are introduced in Lichtenberg (2011) and described in detail in subsubsection 2-1-4. Multilinear dynamics can be given, e.g., by

$$\dot{\mathbf{x}} = f_0 + f_1 x_1 + f_2 x_2 + f_3 x_1 x_2, \tag{2-7}$$

$$\mathbf{y} = g_1 x_1 + g_2 x_1 u, \tag{2-8}$$

$$\mathbf{x}(0) = \mathbf{x}_0, \tag{2-9}$$

for a second order MTI model with one input.

To create a white-box model of a building is therefore a tedious process, that requires a lot of knowledge about the construction of the building. This includes the geometry of the building envelope, as well as detailed information about the components and the plant technology, which is used.

This is where the problems of white-box modeling become apparent: even for simple systems, complex analyses of the theoretical background may be necessary. Nevertheless, pure white-box models are often not precise enough, as not all processes that take place within a real system are known. In addition, the creation of theoretical white-box models for complex systems can be very time-consuming and therefore expensive.

Black-Box Models, on the other hand, do not require any information about the building or the system components in general, but they do require detailed experimental data in the form of historical input and output data of the system operation. The name "black-box" refers to the fact that the internal structure of the system is unknown or not of interest. The dynamic behavior is identified by system identification methods using the input and output data. A black-box model is therefore always an approximation of the original input-output behavior, generated with an assumption on the model structure, Isermann and Münchhof (2011). The identified parameters are generally not mapped to real system properties or physical coefficients. This leads to the problem, that the model parameters of black-box models cannot be interpreted, so that the results of black-box methods are not comprehensible to users. In recent years, this problem has increasingly become the focus of research, Carabantes (2020); De Bruijn et al. (2022). Apart from this drawback, black-box models have significant advantages over white-box models. One important advantage is that the same identification algorithms can be applied to different input-output data. Thus, if a suitable algorithm is available, the effort and time required to generate black-box models is low. Depending on the application, the accuracy and thus the size of the model can be adjusted. Black-box models are often more accurate when it comes to identifying the input-output behavior as accurately as possible, Isermann and Münchhof (2011). However, the availability of the individual system's data is required, so only models for operating systems can be generated. There are several approaches for system identification methods. Machine learning algorithms, such as neural networks, are often used to create black-box models. Other approaches, such as subspace identification methods, are based on system theory. The different methods of parameter identification are discussed in more detail in the following section.

In addition to pure white-box and black-box models, there are many gradations called gray-box models. Light gray-box models for example, contain the physical law behind the models, from which the differential equations can be derived, but use measurable signals in addition to estimate unknown parameters. Dark gray-box models include the main physical rules for the system, but the model structure is unknown, and the measured signals are used to identify the structure and the parameters, (Isermann and Münchhof, 2011).

2-1-2 Parameter identification

Black-box and gray-box modeling uses available signals from the input and output data of the systems to be represented by a model. The parameters of these models are initially unknown and must be estimated using signal-based methods for system identification. In system identification, there are a variety of methods for parameter identification that have specific advantages and areas of application. Some well known techniques for dynamical systems are summarized below. A fundamental distinction of the introduced parameter identification methods in four categories for direct *parameter estimation* methods, *iterative optimization* methods, *subspace-based* methods and *neural networks* is made.

Parameter estimation

In direct parameter estimation for linear and nonlinear static models, the straightforward and well known least squares method based on linear regression is available, if the models

are linear in the parameters, and was formulated already by Gauss in 1821 and 1823. The aim is to fit a model best to the N observations of measurements. Later the linear regression methods were adapted to use it for parameter estimation of dynamic processes with difference equations, which are linear in the parameters, (Kalman, 1958).

In the following the least squares method for linear discrete-time models is introduced, according to Isermann and Münchhof (2011). Given is the linear time-invariant difference equation

$$y_u(k) + a_1 y_u(k-1) + \ldots + a_m y_u(k-m) = b_1 u(k-d-1) + \ldots + b_m u(k-d-m), \tag{2-10}$$

where y_u is the model output, u is the input signal, k is the actual time step, m is the model order and d is the dead time of the system, which describes how many time steps the input $u(k-d)$ needs to affect the output $y(k)$. The model parameters are given with $a_1, \ldots, b_m$. If the measured output $y(k)$ are used instead of the model output $y_u(k)$, rearranging (2-10) gives

$$\begin{aligned} y(k) + \hat{a}_1(k-1)y(k-1) + \ldots + \hat{a}_m(k-1)y_u(k-m) - \\ \hat{b}_1(k-1)u(k-d-1) - \ldots - \hat{b}_m(k-1)u(k-d-m) = e(k), \end{aligned} \tag{2-11}$$

where $e(k)$ is an equation error at time k. An error occurs due to the use of the measured output instead of the real output and the assumed parameters $\hat{a}_1, \ldots, \hat{b}_m$ instead of real parameters. This can be interpreted as the model one-step prediction $\hat{y}(k|k-1)$ of the system output $y(k)$ from the previous available measurement $y(k-1)$ and can be therefore expressed as

$$\begin{aligned} \hat{y}(k|k-1) = -\hat{a}_1(k-1)y(k-1) - \ldots - \hat{a}_m(k-1)y_u(k-m) + \\ \hat{b}_1(k-1)u(k-d-1) + \ldots + \hat{b}_m(k-1)u(k-d-m). \end{aligned} \tag{2-12}$$

By setting up a data vector

$$\boldsymbol{\psi}^{\mathrm{T}}(k) = \begin{pmatrix} -y(k-1) & \ldots & -y(k-m) | u(k-d-1) & \ldots & u(k-d-m) \end{pmatrix} \tag{2-13}$$

and a parameter vector

$$\hat{\boldsymbol{\theta}}^{\mathrm{T}}(k) = \begin{pmatrix} \hat{a}_1 & \ldots & \hat{a}_m | \hat{b}_1 & \ldots & \hat{b}_m \end{pmatrix}, \tag{2-14}$$

as is usual in least squares methods, (2-12) can be expressed by

$$\hat{y}(k|k-1) = \boldsymbol{\psi}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1). \tag{2-15}$$

The notation $\mathbf{V}^{\mathrm{T}}$ is used for the transpose of $\mathbf{V}$. It follows from (2-11) and (2-1-2), that the equation error is

$$e(k) = y(k) - \boldsymbol{\psi}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1) \tag{2-16}$$

and a system of $N+1$ equations can be set up for time steps $k = m+d, m+d+1, \ldots, m+d+N$. To identify the $2m$ parameters from the equation system, at least the same number of equations is required, which means that $N \geq 2m-1$. The equation system can then be stacked together in matrix form

$$\mathbf{y}^{\mathrm{T}}(m+d+N) = \boldsymbol{\psi}(m+d+N)\hat{\boldsymbol{\theta}}(m+d+N-1) + \mathbf{e}(m+d+N) \tag{2-17}$$

The sampled data for y and u are stacked in the output vector $\mathbf{y}^{\mathrm{T}} \in \mathbb{R}^{N+1}$ and the data matrix $\boldsymbol{\psi}(m+d+N) \in \mathbb{R}^{(N+1)\times 2m}$, see Isermann and Münchhof (2011), to construct an error vector $\mathbf{e}^{\mathrm{T}} \in \mathbb{R}^{N+1}$. To identify the parameters in $\boldsymbol{\theta}$ the quadratic cost function

$$J = \mathbf{e}^{\mathrm{T}}(m+d+N)\mathbf{e}(m+d+N) \tag{2-18}$$

needs to be minimized, which can be done by setting the first order derivative

$$\left.\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\theta}}\right| = -2\boldsymbol{\psi}^{\mathrm{T}}(\mathbf{y} - \boldsymbol{\psi}\boldsymbol{\theta}) = \mathbf{0} \tag{2-19}$$

to zero and the parameter vector

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{\psi}^{\mathrm{T}}\boldsymbol{\psi}\right)^{-1}\boldsymbol{\psi}^{\mathrm{T}}\mathbf{y} \tag{2-20}$$

can be calculated using the inverse of the matrix $\boldsymbol{\psi}^{\mathrm{T}}\boldsymbol{\psi}$ and the vector $\boldsymbol{\psi}^{\mathrm{T}}\mathbf{y}$, as explained in detail in Åström and Eykhoff (1971).

For a more general method, where the values of $\mathbf{e}(k)$ need to be weighted by different weights, the cost function (2-18) changes to

$$J = \mathbf{e}^{\mathrm{T}}\mathbf{Q}\mathbf{e}(m+d+N), \tag{2-21}$$

where $\mathbf{Q}$ is a symmetric positive-definite weighting matrix, (Isermann and Münchhof, 2011). The parameter vector is then obtained with

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{\psi}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\psi}\right)^{-1}\boldsymbol{\psi}^{\mathrm{T}}\mathbf{Q}\mathbf{y}. \tag{2-22}$$

Linear Regression in general is a basic statistical method for modeling linear relationships between variables. This method, described by Groß (2003), is easy to implement and widely used in MATLAB, Python (Scikit-learn) and Excel. It is often used in data analysis and predictive modeling, for example to predict energy consumption in buildings based on historical data, (Alshibani, 2020). On the basis of this, there are many extended methods, such as the least squares method for dynamic systems described above. Based on this method, the *Instrumental Variables (IV)* (Huffel and Vandewalle, 1989; Söderström and Mahata, 2002; Guo and Small, 2016) and *Total Least Squares (TLS)* (Huffel and Vandewalle, 1989) and (Kommenda et al., 2020) were developed and implemented in Fortran77,(Van Hueffel, 1988).

Finally, there is the *Structured Full-Rank MTI* method, which is used for multilinear system identification. This method was used for a gray-box approach for multilinear full-rank models in the application for building systems, (Sridharan et al., 2020).

Iterative optimization

Iterative optimization methods are parameter identification methods, which are available for nonlinear systems and attempt to fit the model output to the measured data by minimizing a defined cost function in a sequence of iterations, (Isermann and Münchhof, 2011). As a cost function any even function can be chosen. A typical cost function is the weighted squared error sum

$$J(\boldsymbol{\theta}, \mathbf{e}) = \mathbf{e}'\mathbf{Q}\mathbf{e} \tag{2-23}$$

where $\mathbf{e}$ is a vector of the error between the measurement and the model output and $\boldsymbol{\theta}$ is the set of model parameters, which are estimated to minimize the cost function. With $\mathbf{Q}$ a weighting of the values of $\mathbf{e}$ is possible. In nonlinear iterative optimization processes it is also possible to determine additional constraints. This can be stability criteria of the resulting model or boundaries for the parameter space. The general formulation of the nonlinear optimization problem is then given by

$$\begin{aligned} &\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \\ \text{s.t. } &\boldsymbol{g}(\mathbf{x}) \leq \mathbf{0} \\ &\boldsymbol{h}(\mathbf{x}) = \mathbf{0}. \end{aligned} \tag{2-24}$$

This notes that the cost function $J(\boldsymbol{\theta})$, which is called the *objective function* should be minimized by adapting the parameter vector $\boldsymbol{\theta}$ and subject to (s.t.) the inequality and equality constraints, defined in $\boldsymbol{g}(\boldsymbol{\theta})$ and $h(\boldsymbol{\theta})$. An initial guess of the parameter vector $\boldsymbol{\theta}$ is needed as a starting point. Depending on the formulation of the model, this can be derived from data sheets or from previous identifications. If the parameters are completely unknown, it is also possible to select a random initial value.

Many optimization methods work iteratively and the next investigated parameter vector can be described by

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \alpha \mathbf{s}(k). \tag{2-25}$$

Here the starting point is the so called *minimizing sequence* for $k = 0, 1, \ldots$. The *search vector* $\mathbf{s}(k)$ indicates the direction in which the algorithm progresses, while the parameter α determines the step size in this direction. The aim is to reach the optimal point $\mathbf{x}^*$.

To terminate the iterative process some criteria are needed. One is e.g. set by the condition that an optimum of the unconstrained function should be reached, which is fulfilled when the gradient

$$\nabla J(\boldsymbol{\theta}) = \left(\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right) = \mathbf{0} \tag{2-26}$$

equals a vector of zeros. To make sure that at that point a minimum (at least a local) of the cost function is reached the Hessian matrix

$$\nabla^2 J(\boldsymbol{\theta}) = \left(\frac{\partial^2 J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^\mathrm{T} \partial \boldsymbol{\theta}}\right) = \begin{pmatrix} \frac{\partial^2 J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^\mathrm{T} \partial \boldsymbol{\theta}} & \cdots & \frac{\partial^2 J(\boldsymbol{\theta})}{\partial \theta_1^\mathrm{T} \partial \theta_1} \\ \vdots & & \vdots \\ \frac{\partial^2 J(\boldsymbol{\theta})}{\partial \theta_1^\mathrm{T} \partial \theta_p} & \cdots & \frac{\partial^2 J(\boldsymbol{\theta})}{\partial \theta_p^\mathrm{T} \partial \theta_p} \end{pmatrix} \tag{2-27}$$

is positive definite. In constrained optimization the two conditions may not be reached exactly. Therefore, *termination criteria*, such as a minimum update step or a minimum improvement of the cost function can be added as

$$\|\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)\| \leq \epsilon_\theta \qquad J(\boldsymbol{\theta}(k)) - J(\boldsymbol{\theta}(k+1)) \leq \epsilon_J. \tag{2-28}$$

There are many different algorithms for iterative optimization, including one-dimensional ones for functions that depend on only one variable, such as the point estimation algorithm or

the golden section search. For problems where the objective function J depends on a vector of design variables, multidimensional algorithms are available, including the Newton-Raphson algorithm as well as the Gaussian Newton and Levenberg-Marquardt algorithms, all listed in Isermann and Münchhof (2011). The interior-point methods from Nocedal and Wright (2006) are particularly suitable for constrained optimization problems, especially with inequality constraints. The Prediction Error Method (PEM) is also used here, which minimizes the prediction errors by solving a nonlinear least squares problem to estimate the model parameters, Ljung (2002). Implementations are available in MATLAB and Python. Besides the iterative optimization methods, which bring some disadvantages in high computational cost and duration, subspace methods are introduced next, which are especially for the state and parameter identification of linear state space models available.

Subspace methods

A widely used numerical method is the *Numerical Subspace State Space System Identification (N4SID)* for estimating the parameter matrices of linear state space models. This method was described in Van Overschee and De Moor (1993), Jansson and Wahlberg (1996) and Van Overschee and De Moor (1996) and is based on subspace methods. Subspace methods are used to identify state-space models when only the input and output data are available. The exclusive use of robust linear algebra tools, such as SVD, QR decomposition, matrix projection and subspace angles, is an advantage over iterative optimization methods. The identified models are not unique as state measurements are not required and are estimated together with the order of the model during the identification process. The decomposition of the Hankel matrix from input and output data is used in N4SID to extract the features of the state-space model. Implementations of this method are available in various programming languages, such as MATLAB and the System Identification Package for Python (SIPPY). In Sewe (2018) it is shown how the N4SID can be used to estimate the model parameters of multilinear building models. Multi-linear means a combination of several linear models and thus differs fundamentally from multilinear models, which allow multiplications of states and input signals.

Another similar method is *Multivariable Output Error State Space (MOESP)*, which is also used for the subspace identification of linear state space models. This method was described by Jamaludin et al. (2013). MOESP is particularly useful for modeling multivariable systems, such as those found in many areas of engineering, where multiple input and output variables need to be considered simultaneously. Contrary to N4SID, MOESP uses orthogonal subspace projection in the Hankel matrices, which results in better performance of the MOESP algorithm, especially with noisy data.

The *Canonical Variate Analysis (CVA)* is another subspace method for fitting state space models suitable for linear state-space models and nonlinear models, (Larimore, 1990). CVA is available in MATLAB and Python (Scikit-learn) and offers a robust way to analyze systems with multiple variables. Besides the introduced subspace methods for state and parameter estimation, other state estimation methods from control theory like the Kalman filter from Kalman (1960) are available, but not discussed in detail in this section.

Neural Networks

In contrast to the introduced methods from systems theory, a wide field of machine learning techniques, such as neural networks, has been developed very rapidly in recent years and used in many applications for parameter identification and fault diagnosis, see (Borda et al., 2023). Neural networks should function similarly to biological neuronal connections and learn behavioral patterns. They are based on a mathematical description of a neuron, (Isermann and Münchhof, 2011). The interconnection of individual neurons creates neural networks. The neurons are linked in such a way that several layers with parallel neurons are created: the input layer, the hidden layers and the output layer. Binary, discrete and continuous signals are possible in neural networks. In system identification tasks, these are usually continuous, while binary signals are mainly used for classification models. The neural networks are trained with input and output signals. The identification problem is also solved by setting up and minimizing a cost function. However, due to the nonlinearity of the mostly quadratic cost functions of multi-layer networks, the direct methods of parameter estimation with least-squares methods cannot be applied. Instead, nonlinear methods of iterative optimization must be used. Neural networks based on radial basis function (rbf) can simplify the cost-intensive optimization to some extent through a linear cost function in the parameters, as shown in subsubsection 2-1-2 on the Least Squares Support Vector Machines (LS-SVM) classification. Neural networks are capable of mapping complex nonlinearities. However, one disadvantage of the black-box structure is the lack of interpretability of the optimization results, which is important for many engineering applications. In recent years, a large field of research has opened up that deals with the interpretability of neural networks, (De Bruijn et al., 2022; Himeur et al., 2023).

The system identification methods for dynamic models described in this section are based on deterministic parameters and signals. Other methods with stochastic models, such as the Markov estimation and maximum likelihood methods Young (2015), which are based on probability distributions of the parameters, also often use iterative optimization methods to minimize their cost function, but are not discussed in detail here.

Cluster algorithms

In anomaly detection cluster algorithms play an important role, e.g., to cluster the identified parameters in a class corresponding to nominal data and a class associated with data, where an anomaly occurred. There are many clustering methods available to cluster data in different classes, (Xu and Tian, 2015). Binary clustering methods, such as support vector machines (SVM) classification, see Steinwart and Christmann (2008), cluster the data in two classes and are therefore applicable for supervised anomaly detection. A specific subclass of the support vector machines is the Least Squares Support Vector Machines, described in Suykens et al. (2002). Its advantage is the applicability to large clustering problems, because the quadratic cost function from standard SVM becomes a linear least squares problem in the LS-SVM and is therefore faster in computation.

The support vector classifier determines a class value

$$y(x) = \operatorname{sgn}\left[\sum_{k=1}^{N} \alpha_k y_k K(x, x_k) + b\right] \in \{-1, 1\}. \tag{2-29}$$

for each value x, by using a training data set $\{x_k, y_k\}_k = 1^N$. The parameters α and b, can be determined in the LS-SVM by solving a linear system of equations

$$\begin{pmatrix} 0 & \mathbf{Y}^{\mathrm{T}} \\ \mathbf{Y} & \mathbf{\Omega} + \gamma^{-1}\mathbf{I} \end{pmatrix} \begin{pmatrix} b \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{1} \end{pmatrix}, \tag{2-30}$$

where

$$\Omega_{k,l} = y_k y_l \varphi(x_k)^{\mathrm{T}} \varphi(x_l) = y_k y_l K(x_k, x_l) \tag{2-31}$$

contains the kernel function and $\mathbf{Y} = \{y_1, y_2, ..., y_N\}$, $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_N\}$ and $\mathbf{1} = \{1, 1, ..., 1\}$.

Various kernels, such as linear, polynomial, radial-basis-functions (rbf) and others can be used. The choice of kernel function depends on the data. With the rbf kernels used here

$$K(x_k, x_l) = e^{\frac{-\|x_k - x_l\|^2}{\sigma^2}} \tag{2-32}$$

and with (2-31), there are two parameters, σ and γ, which influence the classification results and can therefore be used to tune the classifier model.

In the case of semi-supervised anomaly detection, a specific adaptation of the support vector machine, the one-class support vector machine, can be used, (Li et al., 2003). In this case, the classifier is trained with nominal data, the boundaries are calculated based on the available data set. For each new observation, the classifier examines whether the new data is within that boundary and therefore belongs to the class. Otherwise, the data are outliers.

The model-based anomaly detection methods introduced so far are based on models. How to identify the parameters of these models is explained in this section as well as a possibility for classification. A modeling approach with MTI models, which will be used in this work, will be introduced in the following. First, the multidimensional structures, the tensors, have to be introduced.

2-1-3 Tensors and decomposition

In this section an introduction to definitions for tensors and tensor decompositions derived in Kolda and Bader (2009) is given, since they are needed for the tensor representation of the MTI models, introduced in the next section.

Tensors

Definition 2-1.1. An n-th order tensor is an n-way array

$$\mathsf{F} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}, \tag{2-33}$$

with elements $x(i_1, i_2, \ldots, i_n)$ indexed by $i_j \in \{1, 2, \ldots, I_j\}$ for $j = 1, \ldots, n$, as illustrated on the left side of Figure 2-2.

Definitions for mathematical operations with tensors are given, e.g., in Cichocki et al. (2009) for the contracted product $\langle \mathsf{F} \,|\, \mathsf{G} \rangle$ of two tensors. An adaptation to Canonical Polyadic tensors is given in Kruppa (2018).

Definition 2-1.2. The contracted product of two tensors $\mathsf{F} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times I_{N+1} \times \cdots \times I_{N+M}}$ and $\mathsf{G} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is given element-wise by

$$z(k_1, \ldots, k_m) = \langle \mathsf{F} \,|\, \mathsf{G} \rangle_{1,\ldots N;1,\ldots,N} (k_1, \ldots, k_m) \tag{2-34}$$
$$= \sum_{i_1=1}^{I_1} \cdots \sum_{i_n=1}^{I_n} f(i_1, \ldots, i_n, k_1, \ldots, k_m) g(i_1, \ldots, i_n)$$

with $k_i \in \{1, 2, \ldots, I_{N+i}\}, i = 1, \ldots, M$.

Decomposition

An exponentially increasing number of elements with the order of the tensor, makes full tensors not applicable for computations such as in simulations. This property would lead to unacceptably high computational time and storage requirements. To avoid this, tensor decomposition methods, such as Canonical Polyadic (CP), are available, which leads to a large reduction of complexity and memory requirements, (Kolda and Bader, 2009). The suitability of CP decomposition for MTI models was demonstrated in Kruppa (2017a). Therefore, this decomposition was used for further reduction of the MTI models in Jöres et al. (2022) and is also used here.

Definition 2-1.3. A CP decomposed tensor

$$\mathsf{X} = \sum_{k=1}^{r} \lambda(k) \cdot \mathbf{X}_1(:, k) \circ \ldots \circ \mathbf{X}_n(:, k) \in \mathbb{R}^{I_1 \times \cdots \times I_n} \tag{2-35}$$

is given by the sum of rank-1 tensors. The outer products, denoted by $\circ$, of the column vectors $\mathbf{X}_i(:, k)$ of the factor matrices $\mathbf{X}_i \in \mathbb{R}^{I_i \times r}$ result in the rank-1 tensors. The vector $\boldsymbol{\lambda}$ contains weighting factors.

Remark: In this thesis, the notation

$$\mathsf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n] \cdot \boldsymbol{\lambda} \tag{2-36}$$

from Kruppa (2018) adopted from Domanov and De Lathauwer (2013) is used for CP tensors, which joins all factor matrices and connects them with a weight vector.

2-1-4 Multilinear models

Because of the multidimensional structure of tensors, they can be used for the representation of MTI models, which are introduced next.

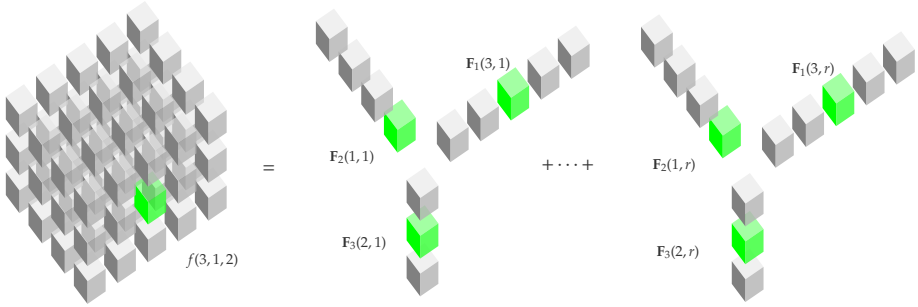

Figure 2-2: CP decomposition of 3-dimensional tensor, see Lichtenberg et al. (2022)

Multilinear time-invariant (MTI) models

Multilinear time-invariant (MTI) models are proposed as a subclass of nonlinear models in Lichtenberg (2011). Their characteristics are the multilinear transition functions. They were used for various applications, e.g. for controller design in heating systems in Kruppa (2018); Lautenschlager (2019) or for applications with hybrid systems inPangalos et al. (2015). In the following description, outputs are not considered, but states are assumed to be measurable outputs. In situations, where this is not the case, MTI models can be extended to include an output function, see Pangalos et al. (2015). The characteristic of MTI models is, that the right-hand side of the transition function allows all combinations of states and inputs in addition to linear terms.

Definition 2-1.4. A discrete-time MTI model is given by

$$\mathbf{x}(k+1) = \langle\, \mathsf{F} \mid \mathsf{M}(\mathbf{x}(k), \mathbf{u}(k)) \,\rangle \tag{2-37}$$

where the monomial tensor

$$\mathsf{M}(\mathbf{x}(k), \mathbf{u}(k)) = \begin{pmatrix} 1 \\ u_m(k) \end{pmatrix} \circ \cdots \circ \begin{pmatrix} 1 \\ u_1(k) \end{pmatrix} \circ \begin{pmatrix} 1 \\ x_n(k) \end{pmatrix} \circ \cdots \circ \begin{pmatrix} 1 \\ x_1(k) \end{pmatrix} \tag{2-38}$$

with dimension $\mathbb{R}^{\times^{(n+m)}2}$ has a rank of one. It is computed by the outer product of vectors containing the elements x_i, $i = 1, \ldots, n$ of the state vector $\mathbf{x} \in \mathbb{R}^n$ and the elements u_j, $j = 1, \ldots, m$ of the input vector $\mathbf{u} \in \mathbb{R}^m$ in descending order.

All parameters of the state transition equation are stored in the transition tensor $\mathsf{F} \in \mathbb{R}^{2n+m}$.

The notation $\mathbb{R}^{\times^{(n+m)}2} = \mathbb{R}^{\overbrace{2 \times 2 \times \cdots \times 2}^{\text{(n+m) times}}}$ is used for the dimensions of tensors.

Definition 2-1.5. A continuous-time MTI model is given by

$$\dot{\mathbf{x}} = \langle\, \mathsf{F} \mid \mathsf{M}(\mathbf{x}, \mathbf{u}) \,\rangle\,, \tag{2-39}$$

which describes the transition function of the first derivative of the state $\dot{\mathbf{x}}$ with the same structure for the monomial and parameter tensor, as in the previous definition.

Example 1. An example second order discrete-time MTI in full tensor representation has the monomial tensor

$$\mathsf{M}(\mathbf{x}(k)) = \begin{pmatrix} 1 & x_1(k) \\ x_2(k) & x_1(k)x_2(k) \end{pmatrix}$$

and the transition tensor

$$\mathsf{F}(:,:,1) = \begin{pmatrix} f_{1,1,1} & f_{1,2,1} \\ f_{2,1,1} & f_{2,2,1} \end{pmatrix}$$
$$\mathsf{F}(:,:,2) = \begin{pmatrix} f_{1,1,2} & f_{1,2,2} \\ f_{2,1,2} & f_{2,2,2} \end{pmatrix}$$

With (2-37) and (2-34) the multilinear discrete time state space system

$$\mathbf{x}_1(k+1) = f_{1,1,1} + f_{1,2,1}x_1(k) + f_{2,1,1}x_2(k) + f_{2,2,1}x_1(k)x_2(k)$$
$$\mathbf{x}_2(k+1) = f_{1,1,2} + f_{1,2,2}x_1(k) + f_{2,1,2}x_2(k) + f_{2,2,2}x_1(k)x_2(k)$$

is given.

CP decomposed multilinear models

The full tensor representation of the MTI model has the disadvantage, that with every state and input the number of tensor elements in the monomial and transition tensor is increasing exponentially, which makes the models not applicable for large systems. Therefore, the CP decomposition can be applied to MTI models to achieve a more efficient tensor representation.

Definition 2-1.6. A discrete-time CP decomposed MTI model is given by 2-37 with the monomial tensor

$$\mathsf{M}(\mathbf{x},\mathbf{u}) = \left[\begin{pmatrix} 1 \\ x_1(k) \end{pmatrix}, \cdots, \begin{pmatrix} 1 \\ x_n(k) \end{pmatrix}, \begin{pmatrix} 1 \\ u_1(k) \end{pmatrix}, \cdots, \begin{pmatrix} 1 \\ u_m(k) \end{pmatrix}\right] \tag{2-40}$$

is a CP decomposed rank-1 tensor of dimension $\mathbb{R}^{\times^{(n+m)}2}$ with the factor vectors of the elements x_i, $i = 1,\ldots,n$ of the state vector $\mathbf{x} \in \mathbb{R}^n$ and the elements u_j, $j = 1,\ldots,m$ of the input vector $\mathbf{u} \in \mathbb{R}^m$.

The state transition tensor

$$\mathsf{F} = [\mathbf{F}_{x_1}, \cdots, \mathbf{F}_{x_n}, \mathbf{F}_{u_1}, \cdots, \mathbf{F}_{u_m}, \mathbf{F}_\Phi] \cdot \boldsymbol{\lambda} \in \mathbb{R}^{n\times^{(n+m)}2} \tag{2-41}$$

can be represented as a CP tensor of rank r. All parameters are composed in F, and it is assumed that they describe the whole system's dynamics. In the CP representation, the model parameters are expressed as factor matrices. For each state and input the corresponding parameters are represented by matrices $\mathbf{F}_{x_1} \cdots \mathbf{F}_{u_m} \in \mathbb{R}^{2\times r}$. In addition, the matrix $\mathbf{F}_\Phi \in \mathbb{R}^{n\times r}$ contains the coefficients that determine how each term is assigned to the n state equations. With the representation of the parameter tensor as CP tensor, it is possible to compute the

right-hand side of the state equation by vector and matrix multiplication of the factors and the monomial vectors

$$\mathbf{x}(k+1) = \mathbf{F}_\Phi \left(\boldsymbol{\lambda}_F \circledast \left(\mathbf{F}_{u_m}^{\mathrm{T}} \begin{pmatrix} 1 \\ u_m(k) \end{pmatrix} \right) \circledast \cdots \circledast \left(\mathbf{F}_{u_1}^{\mathrm{T}} \begin{pmatrix} 1 \\ u_1(k) \end{pmatrix} \right) \circledast \right. \\ \left. \circledast \left(\mathbf{F}_{x_n}^{\mathrm{T}} \begin{pmatrix} 1 \\ x_n(k) \end{pmatrix} \right) \circledast \cdots \circledast \left(\mathbf{F}_{x_1}^{\mathrm{T}} \begin{pmatrix} 1 \\ x_1(k) \end{pmatrix} \right) \right) . \tag{2-42}$$

The symbol $\circledast$ denotes the so-called Hadamard product for element-wise matrix multiplication, (Kruppa, 2017a).

2-2 Building systems

2-2-1 Requirements and technologies of non-residential new buildings

As mentioned in the introduction, buildings are one of the main energy consumers in Europe. In the EU, buildings account for about 40 % of total energy consumption and 36 % of CO_2 emissions. In Germany, building-related final energy consumption accounts for about 35 % of total energy consumption. Non-residential buildings account for 34 % of total building energy use, with space heating being the largest contributor, followed by lighting and air conditioning, (Becker et al., 2023). For this reason, improving the energy efficiency of buildings is a high priority in European policy. By 2050, the entire building stock in the EU is to be climate-neutral. From 2030, all new buildings must be climate-neutral, and for public buildings, this requirement applies from 2028. The average energy consumption in the building sector is to be reduced by at least 16 % by 2030 and by at least 22 % by 2035. The 16 % of the worst-performing non-residential buildings must be renovated by 2033. Heating systems that run on fossil fuels are to be replaced by 2040. From 2025, the installation of pure gas or oil heating systems will no longer be financially supported; instead, the installation of solar systems will be mandatory, where technically and economically feasible. The new directive aims at a detailed strategy for the decarbonization of the building stock. These goals are part of the EU's 'Fit for 55' climate package, which aims to reduce net greenhouse gas emissions by at least 55 % by 2030 compared to 1990 levels, (Europäisches Parlament and Rat der Europäischen Union, 2024).

With the increasing modernization of buildings and the use of renewable energy sources and modern heating and cooling technology, it is essential that the various components communicate with each other to optimize building operation in terms of energy consumption and comfort. Estimates suggest that significant energy losses in buildings result from inefficient operation and system failures, such that the calculated energy demand values are not fulfilled. The deviation is defined as the Energy Performance Gap, (Gram-Hanssen and Georg, 2018). Common problems include incorrectly adjusted heating and cooling systems, inefficient ventilation control, lighting and appliances left on when not in use, and lack of system maintenance.

Modern non-residential buildings use a variety of data sources, including energy consumption data for heating, cooling, lighting, etc., indoor temperatures and air quality, occupancy data, and outdoor climate data. Internet of Things (IoT) Solutions play an important role

in modern buildings to guarantee the collection of data and the communication of different components, as depicted in Figure 2-3. Modern buildings use a variety of technologies for data collection and transmission, such as MQTT (Message Queuing Telemetry Transport), WLAN, Bluetooth, LoRaWAN, and 5G. A comprehensive approach to data collection, analysis, and use in buildings should address the integration of different sensors and systems, the use of appropriate network technologies, scalable cloud solutions or local databases for data storage, algorithms for pattern recognition and optimization, intuitive dashboards for building managers and users, automated system optimization, and ensuring cybersecurity and compliance with privacy policies, (Al Dakheel et al., 2020; Daissaoui et al., 2020).

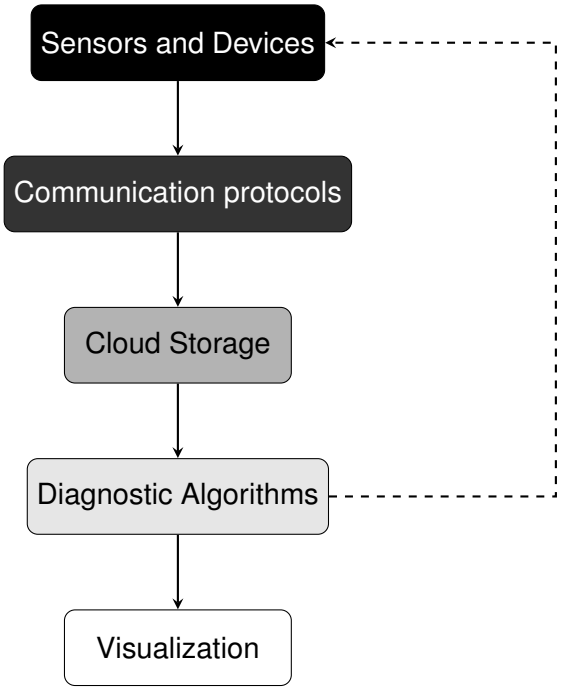

Figure 2-3: Data flow in smart buildings

To make use of the data provided by sensors and devices in diagnostic algorithms, a standardized naming of data points is helpful. In practice, however, a unified naming scheme is often missing, as different actors use their own conventions. This complicates automated evaluation and increases the effort required for external analysis algorithms.

The Buildings Unified Data point naming schema for Operation management (BUDO) scheme, introduced by (Stinner et al., 2018), provides a standardized naming convention. It uses a hierarchical structure consisting of five categories: system, subsystem, position/medium, type, and input/output function. These categories are separated by underscores and dots, and the structure is complemented by an optional free-text section for organization-specific information.

In Figure 2-4 an example name string from a real building is used to demonstrate, how it can be transferred in a standardized scheme, like the BUDO scheme. To use all collected building data for model based diagnostic algorithms, models of the building are needed, as explained in the next section.

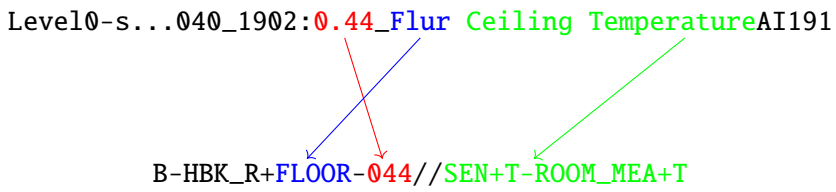

Figure 2-4: Natural language Structuring of data point names

2-2-2 Modeling of building systems

The thermal behavior of a building can be modeled with the law of energy conservation

$$\sum Q = 0, \tag{2-43}$$

with thermal energy

$$Q = c\rho VT \tag{2-44}$$

modeled by a constant value c for the heat capacity, density ρ, the volume V and temperature T of the medium. By considering the time derivative, the heat flow rate

$$\dot{Q} = c\rho\dot{V}T + c\rho V\dot{T} \tag{2-45}$$

results from 2-44 and can be simplified for an air flow with $\dot{T} = 0$ to

$$\dot{Q} = c\rho\dot{V}T. \tag{2-46}$$

The change of heat, which is stored inside the system, follows to

$$\dot{Q} = c\rho V\dot{T} \tag{2-47}$$

for systems with constant volume, as buildings, (Pangalos, 2016). If we assume a simple one zone building approach with one room the change of heat inside the room

$$\dot{Q}_{room} = \dot{Q}_{supply} - \dot{Q}_{return} - \dot{Q}_{loss} \tag{2-48}$$

results from the incoming heat flow from the heating system $\dot{Q}_{supply}$ minus the outgoing heat flow $\dot{Q}_{return}$ and the heat flow caused by losses

$$\dot{Q}_{loss} = UA\Delta T = UA(T_{room} - T_{out}) \tag{2-49}$$

through the wall with cross-section A, heat transmission coefficient U, room temperature T_{room} and ambient temperature T_{out}. The heating is provided through the ventilation system, where air enters with a constant temperature T_{sup} and volume flow rate $\dot{V}_{air}$, and exits with a return temperature T_{return}. The ventilation dynamics are modeled as a first order system with respect to the volume flow rate

$$\ddot{V}_{air} = -\frac{1}{\tau}\dot{V}_{air} + \frac{\hat{V}_{air}}{\tau}\alpha, \tag{2-50}$$

where $\alpha \in [0,1]$ is a control signal for the ventilation system with time constant τ and maximum volume flow rate $\hat{V}_{air}$.

From these differential equations a second order multilinear state space system

$$\begin{aligned} \dot{x}_1 &= p_1(p_2 x_2 - x_1 x_2) + p_3(u_1 - x_1) \\ \dot{x}_2 &= p_4 x_2 + p_5 u_2 \end{aligned} \tag{2-51}$$

can be derived, with state variables $x_1 = T_{return} = T_{room}$ and $x_2 = \dot{V}_{air}$. The inputs are given by $u_1 = T_{out}, u_2 = \alpha$ and the parameters are given by

$$\begin{aligned} p_1 &= \frac{1}{V_{room}}, & p_4 &= \frac{-1}{\tau}, \\ p_2 &= T_{sup}, & p_5 &= \hat{V}_{air}. \\ p_3 &= \frac{UA}{c\rho V_{room}}, \end{aligned}$$

The output is assumed to equal to the states.

Remark: The first state equation (2-51) of this simplified room model is not in the class of linear state space models as it consists of a multiplication of the two states and can be therefore modeled within the multilinear class.

Example 2. The continuous time multilinear state space model from (2-51) is parametrized with suitable values for a standard room by

$$\dot{\mathbf{x}} = \begin{pmatrix} \frac{1}{100}(20x_2 - x_1 x_2) + \left(\frac{0.2 \cdot 49}{0.33 \cdot 100}\right)(u_1 - x_1) \\ \frac{-1}{2} x_2 + \frac{525}{2} u_2 \end{pmatrix} \tag{2-52}$$

with initial state $\mathbf{x}0 = \begin{pmatrix} 20 & 0 \end{pmatrix}^{\mathrm{T}}$ and real outside temperature values for Germany in January. By simulating the shown multilinear state space model for 4 days the input output behavior can be plotted in Figure 2-5. Dependent on the control signal, the volume flow reaches 100 %, 20 % and 50 % of the maximum flow rate, while the room temperature hovers around 20 °C.

This example multilinear system will be used for showing the different steps from modeling, over identification to anomaly detection in this thesis.

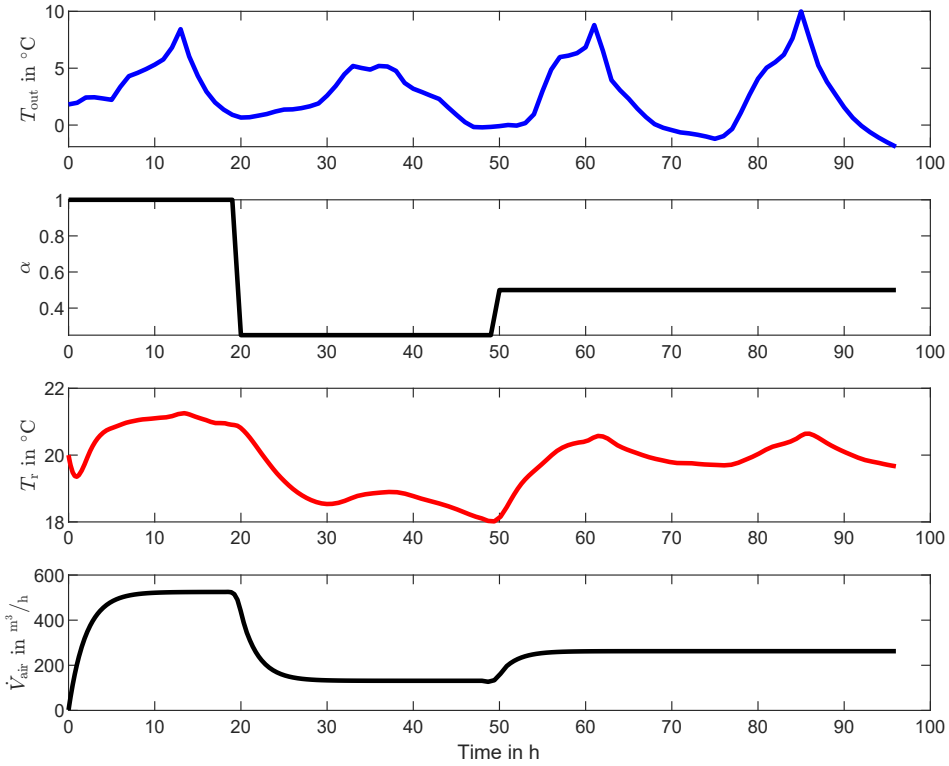

Figure 2-5: Simulation of differential equations for room model

Chapter 3

Modeling

In this chapter, a modeling approach is developed that uses the proven structure of multilinear models, but transforms it into a new, reduced representation. For this purpose, multilinear polynomials are introduced and normalization forms for the parameter spaces of the multilinear models are developed on the basis of these polynomials. This serves to obtain an efficient, unambiguous, and interpretable representation of the models, which is particularly necessary for diagnostic applications. A sparse representation of the normalized models indicates their dynamic structure, which can be generated using pre-structuring methods. In addition, transformations of the models are provided in the normalized representation.

3-1 Normalization

3-1-1 Normalized multilinear polynomials

This section presents a discussion of multilinear functions in normalized representations of factored polynomials, as introduced in reference Jöres et al. (2022). This normalization of multilinear polynomials can later be transferred to the multilinear time-invariant (MTI) models.

A multilinear function in the representation of a factored polynomial

$$f(\mathbf{x}) = \sum_{k=1}^{r} \prod_{i=1}^{n} (f_{1_{i,k}} + f_{x_{i,k}} x_i), \tag{3-1}$$

where x_i are the variables of the function and $f_{1_{i,k}}, f_{x_{i,k}} \in \mathbb{R}$ and $f_{x_{i,k}} \in \mathbb{R}\backslash\{0\}$ are the parameters. With $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, r$ the function has an overall number of $2rn$ model parameters, where the integer value r describes the number of summands.

Remark: The general factored polynomial representation is not unique, as shown in the example

$$(2 + 3x_1)(2 + 4x_2) = (4 + 6x_1)(1 + 2x_2).$$

A visualization of the factors of the polynomial (3-1) in a 2-D Cartesian coordinate system is possible by creating a vector $(f_1\ f_x)^T$ of the parameters, as depicted in Figure 3-1 with an example vector. The first component f_1 is oriented along the 1-axis, while the second component f_{x_i} is oriented along the x-axis.

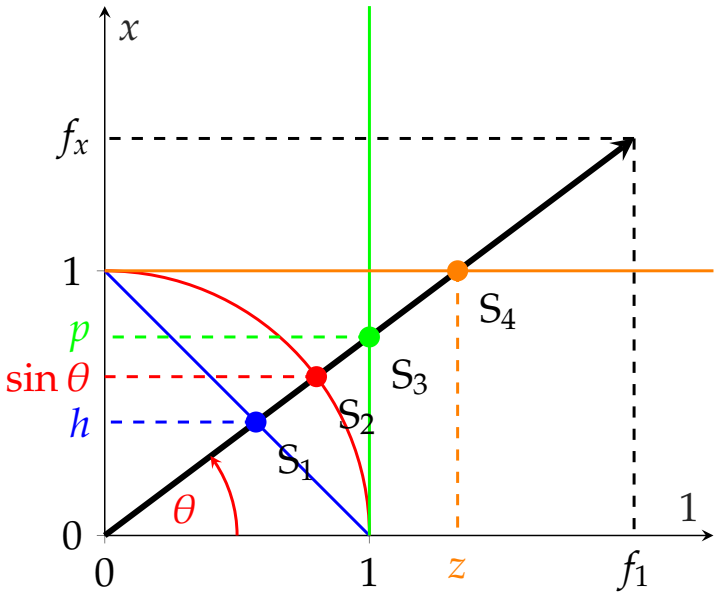

Figure 3-1: Normalized factor representation, (Jöres et al., 2022)

The intersections S_1 to S_4 indicate the lengths of the new vectors resulting from the application of various normalization techniques, which are explained below. The normalization techniques presented convert the general factored polynomial into unique representations, as illustrated in the following example.

Example 3. The example vector from Figure 3-1

$$1\left(2+\frac{3}{2}x\right) = 2.5\left(\cos(0.6435)+\sin(0.6435)x\right) = 3.5\left(\frac{4}{7}+\frac{3}{7}x\right) = \frac{3}{2}\left(\frac{4}{3}+x\right) = 2\left(1+\frac{3}{4}x\right)$$

can be normalized to the different norms, accordingly to Figure 3-1.

Euclidean norm

By applying the Euclidean 2-norm condition, i.e., the length of the vector must be one, the new vector is restricted by the unit circle and points in the same direction as the original vector in Figure 3-1 to the intersection point S_2. The intersection points of the 2-norm normalized vector are obtained by applying the trigonometric functions via $\sin\theta$ and $\cos\theta$. The 2-norm normalized polynomial

$$f(\mathbf{x}) = \sum_{k=1}^{r} \lambda_{\theta,k}\left(\prod_{i=1}^{n}(\cos\theta_{i,k}+\sin\theta_{i,k}x_i)\right), \tag{3-2}$$

with parameter

$$\theta_{i,k} = \text{atan2}\left(f_{x_{i,k}}, f_{1_{i,k}}\right) \tag{3-3}$$

and length factor

$$\lambda_{\theta,k} = \prod_{i=1}^{n} \sqrt{f_{1_{i,k}}^2 + f_{x_{i,k}}^2}, \tag{3-4}$$

is unique and has $r(n+1)$ parameters, which is a large reduction compared to the number of parameters in (3-1). The function atan2 is the four-quadrant inverse tangent and returns the angle between the first and the second argument in radians.

Absolute norm

A 1-norm representation is obtained with a similar procedure. Here, the intersection S_1 of the parameter vector with the blue line defines the new parameter vector in the 1-norm normalized polynomial

$$f(\mathbf{x}) = \sum_{k=1}^{r} \lambda_{h,k} \left(\prod_{i=1}^{n} (1 - |h_{i,k}| + h_{i,k} x_i) \right). \tag{3-5}$$

The 1-axis intercept

$$h_{i,k} = \text{sign}(f_{1_{i,k}}) \frac{f_{x_{i,k}}}{|f_{1_{i,k}}| + |f_{x_{i,k}}|} \tag{3-6}$$

and the new length factor

$$\lambda_{h,k} = \prod_{i=1}^{n} \text{sign}(f_{1_{i,k}}) \left(|f_{1_{i,k}}| + |f_{x_{i,k}}| \right), \tag{3-7}$$

specify the new parameters of this representation. The function

$$\text{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{3-8}$$

takes into account the sign of the first vector component, which would be lost through the absolute value in the normalization.

For completeness, two further possibilities of normalizing multilinear polynomials are shown here, as introduced in Jöres et al. (2022).

Zero representation

The next one results from the limitation of the vertical component to the value one, which is shown by the orange line and the intersection point S_4, and leads to the polynomial

$$f(\mathbf{x}) = \sum_{k=1}^{r} \lambda_{z,k} \left(\prod_{i=1}^{n} (z_{i,k} + x_i) \right), \tag{3-9}$$

with the parameters

$$z_{i,k} = \frac{f_{1_{i,k}}}{f_{x_{i,k}}} \quad \text{and} \quad \lambda_{z,k} = \prod_{i=1}^{n} f_{x_{i,k}}. \tag{3-10}$$

In this representation, the zeros of each factor of the polynomial are easily found through the connection $x_{i,k} = -z_{i,k}$.

Sparse representation

Another so-called *sparse* representation is based on the time-constant form of transfer function and is given by the last intersection S_3 with the green line in Figure 3-1. It fixes the 1-axis coordinate to one, which leads to the representation

$$f(\mathbf{x}) = \sum_{k=1}^{r} \lambda_k \left(\prod_{i=1}^{n} (1 + p_{i,k} x_i) \right), \tag{3-11}$$

with the parameters

$$p_{i,k} = \frac{f_{x_{i,k}}}{f_{1_{i,k}}} \quad \text{and} \quad \lambda_k = \prod_{i=1}^{n} f_{1_{i,k}}, \tag{3-12}$$

for all $f_{1_{j,k}} \neq 0$. For the special case of $f_{1_{i,k}} = 0$, the factor vector is parallel to the x_i-axis resulting in $p_{i,k} \to \infty$.

In this case, the polynomial factors without a constant part are not treated as in (3-12), but a new index j is introduced for them and (3-11) changes to

$$f(\mathbf{x}) = \sum_{k=1}^{r} \lambda_k \prod_{j} x_j \prod_{i} (1 + p_{i,k} x_i) \,, \tag{3-13}$$

with the adjusted factors $\lambda_k = \prod_i f_{1_{i,k}} \prod_j f_{x_{j,k}}$.

3-1-2 Normalized MTI models

In this work, the parameters of multilinear models shall be used for further purposes as for diagnosis. Therefore, a unique representation is required which is not given in the general Canonical Polyadic (CP) MTI representation as the next example shows.

Example 4. Consider the multilinear rank-1 state equation

$$x_1(k+1) = 2 + 2x_1(k) + x_2(k) + x_1(k)x_2(k) \tag{3-14}$$

which is not unique in the CP representation from Definition 2-1.6 as it can be represented with the CP parameter tensor

$$\mathsf{F} = \left[\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\1\end{pmatrix}, \begin{pmatrix}1\end{pmatrix}\right] = \left[\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1\\0.5\end{pmatrix}, \begin{pmatrix}2\end{pmatrix}\right] = \left[\begin{pmatrix}0.5\\0.5\end{pmatrix}, \begin{pmatrix}1\\0.5\end{pmatrix}, \begin{pmatrix}4\end{pmatrix}\right], \tag{3-15}$$

which gives with (2-42) the state equation as factored polynomial

$$x_1(k+1) = 1(1+1x_1(k))(2+x_2(k)) = 2(1+1x_1(k))(1+0.5x_2(k)) = 4(0.5+0.5x_1(k))(1+0.5x_2(k)).$$

To overcome this ambiguity, normalization methods as proposed in subsection 3-1-1 for multilinear functions and introduced in Lichtenberg et al. (2022) for implicit MTI models and used in Schnelle et al. (2022); Jöres et al. (2022) for explicit MTI models with low rank can help. In the following, all equations are derived in discrete-time representation. As the structure of the right-hand sides of the transition function is similar for continuous-time MTI, a transmission to continuous-time MTI is possible.

Definition 3-1.1. A normalized CP decomposed (CPN) model is given by (2-37)

$$\mathbf{x}(k+1) = \left\langle \widetilde{\mathsf{F}} \mid \mathsf{M}(\mathbf{x}, \mathbf{u}) \right\rangle, \tag{3-16}$$

with (2-40) and transition tensor

$$\widetilde{\mathsf{F}} = \left[\widetilde{\mathbf{F}}_{z_1}, \widetilde{\mathbf{F}}_{z_2}, \ldots, \widetilde{\mathbf{F}}_{z_{n+m}}, \widetilde{\mathbf{F}}_{\Phi}\right], \tag{3-17}$$

where $\mathbf{z} = (x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m)^{\mathrm{T}} \in \mathbb{R}^{(n+m)}$ holds all states and inputs in a vector and has l-normalized columns $k = 1, \ldots, r$

$$\|\widetilde{\mathbf{F}}_i(:,k)\|_l = 1, \tag{3-18}$$

for all $i = 1, \ldots, (n+m)$ factor matrices and $\widetilde{\mathbf{F}}_{\Phi}$ in its last factor matrix.

The Absolute-value norm (1-norm) and Euclidean norm (2-norm) are used in (3-18) for CPN models by

$$\|\widetilde{\mathbf{F}}_i(:,k)\|_1 = \sum_{j=1}^{2} |\widetilde{\mathbf{F}}_i(j,k)| = |\widetilde{\mathbf{F}}_i(1,k)| + |\widetilde{\mathbf{F}}_i(2,k)| = 1\,, \tag{3-19}$$

and

$$\|\widetilde{\mathbf{F}}_i(:,k)\|_2 = \sqrt{\sum_{j=1}^{2} \widetilde{\mathbf{F}}_i(j,k)^2} = \sqrt{\widetilde{\mathbf{F}}_i(1,k)^2 + \widetilde{\mathbf{F}}_i(2,k)^2} = 1\ . \tag{3-20}$$

by a column-wise normalization of the $i = 1, 2, \ldots, n+m$ factor matrices $\mathbf{F}_i$.

The computation of (2-42) shows that the right-hand sides of CP decomposed MTI from Definition 2-1.6 consist of polynomials like (3-1). Therefore, it follows that computing the next state of an MTI model can be done in the form of normalized factored polynomials (3-2) to (3-9) in the different norm representations. This can be derived from the corresponding normalized parameter tensors of (3-17) as shown for the 1-norm and 2-norm in the following.

The transition tensor of a 2-norm CPN model (3-16) within the tensor framework is then given by (3-17) with the angles (3-3) as

$$\begin{aligned}\tilde{\mathsf{F}}_{|_2} = \Bigg[&\begin{pmatrix}\cos\theta_{1,1} & \ldots & \cos\theta_{1,r}\\ \sin\theta_{1,1} & \ldots & \sin\theta_{1,r}\end{pmatrix}, \begin{pmatrix}\cos\theta_{2,1} & \ldots\\ \sin\theta_{2,1} & \ldots\end{pmatrix}\\ &\begin{pmatrix}\cos\theta_{2,r}\\ \sin\theta_{2,r}\end{pmatrix}, \cdots, \begin{pmatrix}\cos\theta_{n+m,1} & \ldots & \cos\theta_{n+m,r}\\ \sin\theta_{n+m,1} & \ldots & \sin\theta_{n+m,r}\end{pmatrix}, \tilde{\mathbf{F}}_\Phi\Bigg]\end{aligned} \tag{3-21}$$

In 1-norm the transition tensor is built by applying (3-6)

$$\begin{aligned}\tilde{\mathsf{F}}_{|_1} = \Bigg[&\begin{pmatrix}1-|h_{1,1}| & \ldots & 1-|h_{1,r}|\\ h_{1,1} & \ldots & h_{1,r}\end{pmatrix}, \begin{pmatrix}1-|h_{2,1}| & \ldots\\ h_{2,1} & \ldots\end{pmatrix}\\ &\begin{pmatrix}1-|h_{2,r}|\\ h_{2,r}\end{pmatrix}, \cdots, \begin{pmatrix}1-|h_{n+m,1}| & \ldots & 1-|h_{n+m,r}|\\ h_{n+m,1} & \ldots & h_{n+m,r}\end{pmatrix}, \tilde{\mathbf{F}}_\Phi, \Bigg]\end{aligned} \tag{3-22}$$

where $h_{i,j} = \widetilde{\mathbf{F}}_i(2,j)$ with $i = 1, ..., n+m$ and $j = 1, ..., r$ is obtained from (3-6) and represents the entire tensor due to the norm condition. All columns are normalized except the columns of

$$\widetilde{\mathbf{F}}_\Phi(:,j) = \lambda_k \mathbf{F}_\Phi(:,j) \tag{3-23}$$

with the original length stored in the column vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_r)^{\mathrm{T}}$ from the specific norm (3-7) or (3-4).

This representation shows that each factor matrix $\widetilde{\mathbf{F}}_i \in \mathbb{R}^{2\times r}$ from $i = 1, \ldots, n+m$ of the parameter tensor $\widetilde{\mathsf{F}}_{|}$ of an CPN model can be represented by a single parameter vector

$$\widetilde{\mathbf{F}}_{|_1 i} = \mathbf{h}_i = \left(h_{i,1}, h_{i,2}, \ldots, h_{i,r}\right) \in \mathbb{R}^{1\times r} \tag{3-24}$$

or

$$\widetilde{\mathbf{F}}_{|_2 i} = \boldsymbol{\theta}_i = \left(\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,r}\right) \in \mathbb{R}^{1\times r} \tag{3-25}$$

from which the missing element can be reconstructed using the norm condition (3-19) or (3-20), as proposed in Lichtenberg et al. (2022). With (2-42), the next state can be computed from the normalized factor matrices from $\tilde{\mathsf{F}}_{|_1}$ or $\tilde{\mathsf{F}}_{|_2}$.

Remark: To build all multilinear models in normalized representation only the right half of the coordinate system is needed, if the non-normalized factor $\mathbf{F}_\phi$ allows also negative values. Expressing the parameter vectors from the left half plane is in fact possible by those of the right half plane by

$$\phi\left(-f_1 + f_x x\right) = -\phi\left(f_1 - f_x x\right)$$

Therefore, the norm condition (3-18), where per definition only positive norms are allowed is valid, which concerns the $\widetilde{\mathbf{F}}_i(1,j) = 1 - \left|h_{i,j}\right|$ parameters in (3-22). The original sign of the $\mathbf{F}_i(1,j)$ parameter is considered with the sign function (3-8) inside the new ϕ value. For the angle representation in Euclidean norm, this means, that positive angles $0 \leq \theta \leq \pi$ are sufficient to represent all possible MTI through

$$\begin{aligned}\phi\left(\cos(-\theta) + \sin(-\theta)x\right) &= \phi\left(\cos(\theta) - \sin(\theta)x\right)\\ &= \phi\left(-\cos(\pi-\theta) - \sin(\pi-\theta)x\right)\\ &= -\phi\left(\cos(\pi-\theta) + sin(\pi-\theta)x\right).\end{aligned}$$

Example 5. The multilinear state equation from Example 2 can be represented in CP decomposed multilinear models from the expanded state equation as described in Kruppa (2018) by

$$\begin{array}{lcl}
\dot{x}_1 & = & -0.0705x_1 + 0.0629x_2 - 0.0029x_1x_2 + 0.0705u_1\\
\dot{x}_2 & = & -0.5x_2 + 262u_2\\
\mathbf{F}_{x_1} = \mathbf{F}_{T_r} & = & \begin{pmatrix}0 & 1 & 0 & 1 & 1 & 1\\ 1 & 0 & 1 & 0 & 0 & 0\end{pmatrix}\\
\mathbf{F}_{x_2} = \mathbf{F}_{\dot{V}} & = & \begin{pmatrix}1 & 0 & 0 & 1 & 0 & 1\\ 0 & 1 & 1 & 0 & 1 & 0\end{pmatrix}\\
\mathbf{F}_{u_1} = \mathbf{F}_{T_{out}} & = & \begin{pmatrix}1 & 1 & 1 & 0 & 1 & 1\\ 0 & 0 & 0 & 1 & 0 & 0\end{pmatrix}\\
\mathbf{F}_{u_2} = \mathbf{F}_{\alpha} & = & \begin{pmatrix}1 & 1 & 1 & 1 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{pmatrix}\\
\mathbf{F}_{\phi} & = & \begin{pmatrix}1 & 0 & 1 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 1\end{pmatrix}\\
\boldsymbol{\lambda} & = & (-0.0705 \quad 0.0629 \quad -0.0029 \quad 0.0705 \quad -0.5 \quad 262)
\end{array}$$

Using (3-6) and (3-7) in (3-23), the right-hand side of the CP represented state equation can be reduced to a 1-norm normalized representation (3-16). With (3-5) and (3-24) the same model is represented by

$$\begin{array}{lcl}
\dot{x}_1 & = & -0.0705(1-1+1x_1) + 0.0629(1-1+1x_2) - 0.0029(1-1+1x_1)(1-1+1x_2) + 0.0705(1-1+1u_1)\\
\dot{x}_2 & = & -0.5(1-1+1x_2) + 262(1-1+1u_2)\\
\mathbf{h}_{T_r} & = & (1 \quad 0 \quad 1 \quad 0 \quad 0)\\
\mathbf{h}_{\dot{V}} & = & (0 \quad 1 \quad 1 \quad 0 \quad 0)\\
\mathbf{h}_{T_{out}} & = & (0 \quad 0 \quad 0 \quad 1 \quad 0)\\
\mathbf{h}_{\alpha} & = & (0 \quad 0 \quad 0 \quad 0 \quad 1)\\
\widetilde{\mathbf{F}}_{\phi} & = & \begin{pmatrix}-0.0705 & 0.0629 & -0.0029 & 0.0705 & 0\\ 0 & -0.5 & 0 & 0 & 262\end{pmatrix}
\end{array} \tag{3-26}$$

Remark: By including $\boldsymbol{\lambda}$ in the matrix $\widetilde{\mathbf{F}}_\Phi$ and storing only the needed second row of all parameter matrices $\widetilde{\mathbf{F}}_i$ in one matrix $\widetilde{\mathbf{F}}_{|_l} = (\mathbf{h}_1, \ldots, \mathbf{h}_{n+m})^{\mathrm{T}}$, the normalization step saves almost half of the parameters needed to represent exactly the same model. It is then also possible to reduce the rank of the model by combining the column two and five of the CP matrices, because they both select the same term x_2 for both state equations.

As the implementation of a CP tensor is proposed in Bader et al. (2019) and used in Kruppa (2017b) as a MATLAB struct, the CPN representation shows advantages in storage demand and simulation time, because it reduces the number of parameters and uses standard numeric matrices. Table 3-1 shows this comparison for the model from Example 5.

Table 3-1: Performance of CPN format

	CP MTI	**CPN MTI**
Order	2	2
Inputs	2	2
Number of parameters	66	30
Storage demand	824 Bytes	240 Bytes
Simulation time	0.2625 s	0.0089 s

The described normalization of the MTI model transforms it into a unique representation and also leads to a reduction in the number of parameters. Nevertheless, this depends on the rank of the models. For minimal representations, a limitation of the rank is proposed below.

Limited rank CPN models

A limitation of the rank of the CP transition tensor F, creates a subclass of the MTI models, which is presented next, Schnelle et al. (2022).

Definition 3-1.2. A Rank-1 CPN MTI model is given by (3-16), with a rank-1 transition tensor

$$\widetilde{\mathsf{F}} = [\tilde{\mathbf{f}}_{x_1}, \cdots, \tilde{\mathbf{f}}_{x_n}, \tilde{\mathbf{f}}_{u_1}, \cdots, \tilde{\mathbf{f}}_{u_m}, \tilde{\mathbf{f}}_\Phi,] \cdot \lambda \,, \tag{3-27}$$

consisting of the vector

$$\tilde{\mathbf{f}}_\Phi = (\phi_1 \ \phi_2 \ \cdots \ \phi_n)^{\mathrm{T}} \in \mathbb{R}^n \,, \tag{3-28}$$

and the factor vectors $\tilde{\mathbf{f}}_{u_j} \in \mathbb{R}^2$ and $\tilde{\mathbf{f}}_{x_i} \in \mathbb{R}^2$, which are normalized to one with any norm $\|\tilde{\mathbf{f}}_{u_j}\| = \|\tilde{\mathbf{f}}_{x_i}\| = 1$.

Remark: Due to the norm condition, all information is given by one value per factor, therefore it is possible to store all parameters in one vector $\tilde{\mathbf{f}} = \left(\theta_1, \theta_2, ..., \theta_{n+m}\right)^{\mathrm{T}} \equiv \left(h_1, ..., h_{n+m}\right)^{\mathrm{T}} \in \mathbb{R}^{n+m}$, with (3-21) or (3-22). The second element can be reconstructed, if needed by the norm condition. In this minimal representation the total number of parameters is $2n+m$, as the transition tensor can be substituted by $n+m$ angles θ_i in the Euclidean norm or elements h_i in Absolute norm of the factors $\tilde{\mathbf{f}}_{u_j}$ and $\tilde{\mathbf{f}}_{x_i}$ and n elements of $\tilde{\mathbf{f}}_\Phi$.

Example 6. The 1-norm normalized CP transition tensor of a second order MTI with one input

$$\widetilde{\mathsf{F}} = \left[\begin{pmatrix}1/4\\3/4\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1/2\\1/2\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}\right] \cdot 8$$

gives with the monomial

$$\mathsf{M}(\mathbf{x}, u) = \begin{pmatrix}1\\x_1\end{pmatrix} \circ \begin{pmatrix}1\\x_2\end{pmatrix} \circ \begin{pmatrix}1\\u\end{pmatrix}$$

and (2-42)

$$\begin{pmatrix}x_1(k+1)\\x_2(k+1)\end{pmatrix} = \begin{pmatrix}1\\2\end{pmatrix}\left(8 \circledast \left(\begin{pmatrix}1/4 & 3/4\end{pmatrix}\begin{pmatrix}1\\x_1(k)\end{pmatrix}\right) \circledast \right.$$
$$\left. \circledast \left(\begin{pmatrix}0 & 1\end{pmatrix}\begin{pmatrix}1\\x_2(k)\end{pmatrix}\right) \circledast \left(\begin{pmatrix}1/2 & 1/2\end{pmatrix}\begin{pmatrix}1\\u(k)\end{pmatrix}\right)\right),$$

which leads to the state space model

$$\begin{aligned} x_1(k+1) &= x_2(k) + u(k)x_2(k) + 3x_1(k)x_2(k) + 3u(k)x_1(k)x_2(k) \\ x_2(k) &= 2x_2(k) + 2u(k)x_2(k) + 6x_1(k)x_2(k) + 6u(k)x_1(k)x_2(k) \end{aligned}$$

Remark: With some abuse of notation, an alternative minimal representation of the CPN is given by the 2-norm with its angles θ_i from (3-3) and ϕ from (3-4) by

$$\left[\tan_2^{-1}(3/4, 1/4), \tan_2^{-1}(1, 0), \tan_2^{-1}(1/2, 1/2), \begin{pmatrix}4.47\\8.94\end{pmatrix}\right].$$

It becomes clear that no different structures are possible in the right-hand sides of the state space models when the rank is limited to 1. The rank-1 limitation reduces the complexity but also has the disadvantage that all n states can only differ by a constant factor, which restricts the dynamics. For states with differing dynamic characteristics the diagonal normalized rank-n MTI model from Schnelle et al. (2022) is introduced in the following.

Definition 3-1.3. A Diagonal Rank-n CPN MTI model is given by (3-16) and (3-17) and has a diagonal matrix

$$\widetilde{\mathbf{F}}_\Phi = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} \in \mathbb{R}^{n \times n} \tag{3-29}$$

and normalized factor matrices $\widetilde{\mathbf{F}}_{u_i} \in \mathbb{R}^{2\times n}$ and $\widetilde{\mathbf{F}}_{x_i} \in \mathbb{R}^{2\times n}$.

Remark: Similar to the rank-1 models, diagonal rank-n models have a fixed number of parameters that fully describe the dynamics of the model. The factor matrices $\widetilde{\mathbf{F}}_i$ contain $n(n+m)$ parameters and the matrix $\widetilde{\mathbf{F}}_\Phi$ contains n diagonal elements λ_i.

Example 7. A second order diagonal normalized rank-n MTI model with state equation

$$\mathbf{x}(k+1) = \begin{pmatrix} 2x_2(k) + 2u(k)x_2(k) + 2x_1(k)x_2(k) + x_1(k)x_2(k)u(k) \\ x_1(k)u(k) + x_1(k)x_2(k)u(k) \end{pmatrix}$$

can be represented in CP tensor framework as

$$\widetilde{\mathsf{F}}_{|1} = \left[\begin{pmatrix} 2/3 & 0 \\ 1/3 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0.5 \\ 1 & 0.5 \end{pmatrix}, \begin{pmatrix} 0.5 & 0 \\ 0.5 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \right],$$

which leads to the next state equations

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} 6 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} (1 - 1/3 + 1/3x_1)\,(x_2)\,(1 - 1/2 + 1/2u) \\ (x_1)\,(1 - 1/2 + 1/2x_2)\,(u) \end{pmatrix}.$$

The following section will introduce a sparse representation for MTI models, which is a suitable representation especially for large systems.

Normalized rank-*r* sparse representation

Since large systems typically have a number of states that depend on only a few other signals, rank-r sparse MTI models are a suitable representation from Schnelle et al. (2023).

Definition 3-1.4. A Normalized rank-r sparse MTI model is given by (2-37) and (3-17) with the sparse matrix

$$\mathbf{\Theta} = \begin{pmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,r} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,r} \\ \vdots & \vdots & \cdots & \vdots \\ \theta_{m+n,1} & \theta_{m+n,2} & \cdots & \theta_{m+n,r} \end{pmatrix} \in \mathbb{R}^{(n+m)\times r}, \tag{3-30}$$

including the nonzero elements $\theta_{i,j}$ in the 2-norm representation, with $j = 1, 2, .., r$ and $i = 1, 2, ...n + m$, which indicate an influence of a certain signal j (state or input) on the next state and the $\widetilde{\mathbf{F}}_{\phi} \in \mathbb{R}^{n\times r}$. A similar parameter matrix is given for the 1-norm with

$$\mathbf{H} = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,r} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,r} \\ \vdots & \vdots & \cdots & \vdots \\ h_{m+n,1} & h_{m+n,2} & \cdots & h_{m+n,r} \end{pmatrix} \in \mathbb{R}^{(n+m)\times r}. \tag{3-31}$$

Proposition 1. *If the parameter tensor $\widetilde{\mathsf{F}}$ of an MTI model is in CP representation and normalized, the computation of the contracted product in the state equation*

$$\mathbf{x}(k+1) = \left\langle \widetilde{\mathsf{F}} \,|\, \mathsf{M}(\mathbf{x}(k), \mathbf{u}(k)) \right\rangle, \tag{3-32}$$

is possible in the matrix form of a polynomial representation

$$\mathbf{x}(k+1) = \widetilde{\mathbf{F}}_{\phi} \begin{pmatrix} \prod_{i=1}^{n+m} (\cos\theta_{i,1} + \sin\theta_{i,1} \cdot z_i(k)) \\ \prod_{i=1}^{n+m} (\cos\theta_{i,2} + \sin\theta_{i,2} \cdot z_i(k)) \\ \vdots \\ \prod_{i=1}^{n+m} (\cos\theta_{i,r} + \sin\theta_{i,r} \cdot z_i(k)) \end{pmatrix} \tag{3-33}$$

for the 2-norm or according to the same principle, but with (3-5) *for the 1-norm representation with* (3-31). *The vector* $\mathbf{z} = (\mathbf{x}\ \ \mathbf{u})^T \in \mathbb{R}^{n+m}$ *comprises all state and input signals.*

Proof. With

$$\widetilde{\mathsf{F}} = \left[\widetilde{\mathbf{F}}_{z_1}, \widetilde{\mathbf{F}}_{z_2}, \ldots, \widetilde{\mathbf{F}}_{z_{n+m}}, \widetilde{\mathbf{F}}_{\Phi} \right] , \tag{3-34}$$

it follows from (2-42) that

$$\mathbf{x}(k+1) = \left\langle \widetilde{\mathsf{F}} \mid \mathsf{M}(\mathbf{x}(k), \mathbf{u}(k)) \right\rangle \tag{3-35}$$

$$= \widetilde{\mathbf{F}}_{\phi} \left(\left(\widetilde{\mathbf{F}}_{z_1}^{\mathrm{T}} \begin{pmatrix} 1 \\ z_1 \end{pmatrix} \right) \circledast \cdots \circledast \left(\widetilde{\mathbf{F}}_{z_{n+m}}^{\mathrm{T}} \begin{pmatrix} 1 \\ z_{n+m} \end{pmatrix} \right) \right). \tag{3-36}$$

Inserting the elements of the CP parameter factor matrices gives

$$\mathbf{x}(k+1) = \widetilde{\mathbf{F}}_{\phi} \left(\left(\begin{pmatrix} \widetilde{\mathbf{F}}_{z_1}(1,1) & \widetilde{\mathbf{F}}_{z_1}(2,1) \\ \widetilde{\mathbf{F}}_{z_1}(1,2) & \widetilde{\mathbf{F}}_{z_1}(2,2) \\ \vdots & \vdots \\ \widetilde{\mathbf{F}}_{z_1}(1,r) & \widetilde{\mathbf{F}}_{z_1}(2,r) \end{pmatrix} \begin{pmatrix} 1 \\ z_1 \end{pmatrix} \right) \circledast \cdots \circledast \left(\begin{pmatrix} \widetilde{\mathbf{F}}_{z_{n+m}}(1,1) & \widetilde{\mathbf{F}}_{z_{n+m}}(2,1) \\ \widetilde{\mathbf{F}}_{z_{n+m}}(1,1) & \widetilde{\mathbf{F}}_{z_{n+m}}(2,1) \\ \vdots & \vdots \\ \widetilde{\mathbf{F}}_{z_{n+m}}(1,1) & \widetilde{\mathbf{F}}_{z_{n+m}}(2,1) \end{pmatrix} \begin{pmatrix} 1 \\ z_{n+m} \end{pmatrix} \right) \right) \tag{3-37}$$

$$= \widetilde{\mathbf{F}}_{\phi} \left(\begin{pmatrix} \widetilde{\mathbf{F}}_{z_1}(1,1) \cdot 1 + \widetilde{\mathbf{F}}_{z_1}(2,1) z_1 \\ \widetilde{\mathbf{F}}_{z_1}(1,2) \cdot 1 + \widetilde{\mathbf{F}}_{z_1}(2,2) z_1 \\ \vdots \\ \widetilde{\mathbf{F}}_{z_1}(1,r) \cdot 1 + \widetilde{\mathbf{F}}_{z_1}(2,r) z_1 \end{pmatrix} \circledast \cdots \circledast \begin{pmatrix} \widetilde{\mathbf{F}}_{z_{n+m}}(1,1) \cdot 1 + \widetilde{\mathbf{F}}_{z_{n+m}}(2,1) z_{n+m} \\ \widetilde{\mathbf{F}}_{z_{n+m}}(1,2) \cdot 1 + \widetilde{\mathbf{F}}_{z_{n+m}}(2,2) z_{n+m} \\ \vdots \\ \widetilde{\mathbf{F}}_{z_{n+m}}(1,1) \cdot 1 + \widetilde{\mathbf{F}}_{z_{n+m}}(2,1) z_{n+m} \end{pmatrix} \right) \tag{3-38}$$

With the norm condition (3-20), (3-21) can be inserted to

$$\mathbf{x}(k+1) = \widetilde{\mathbf{F}}_{\phi} \left(\begin{pmatrix} \cos\theta_{1,1} \cdot 1 + \sin\theta_{1,1} z_1 \\ \cos\theta_{1,2} \cdot 1 + \sin\theta_{1,2} z_1 \\ \vdots \\ \cos\theta_{1,r} \cdot 1 + \sin\theta_{1,r} z_1 \end{pmatrix} \circledast \cdots \circledast \begin{pmatrix} \cos\theta_{n+m,1} \cdot 1 + \sin\theta_{n+m,1} z_{n+m} \\ \cos\theta_{n+m,2} \cdot 1 + \sin\theta_{n+m,2} z_{n+m} \\ \vdots \\ \cos\theta_{n+m,r} \cdot 1 + \sin\theta_{n+m,r} z_{n+m} \end{pmatrix} \right). \tag{3-39}$$

Since the Hadamard product $\circledast$ denotes element-wise multiplication, (3-39) simplifies to (4-10). □

Remark: Proposition 1 is shown for the Euclidean (2-norm) CPN. Obviously, a similar representation is also possible for the absolute norm (1-norm) using (3-19). Here the elements of the factor matrices are given by (3-22).

The following example from Schnelle et al. (2023) shows the usage of sparse MTI models.

Example 8. The second order model with three inputs

$$x_1(k+1) = 1 + x_1(k) + 2x_2(k) + 2x_1(k)x_2(k)$$
$$x_2(k+1) = 2u_2(k)u_3(k) + 2u_1(k)u_2(k)u_3(k)$$

is considered, which can be expressed in normalized rank-2 representation using the full parameter matrix

$$\mathbf{\Theta} = \begin{pmatrix} \arctan_2(1,1) & 0 \\ \arctan_2(2,1) & 0 \\ 0 & \arctan_2(1,1) \\ 0 & \arctan_2(2,0) \\ 0 & \arctan_2(1,0) \end{pmatrix}$$

and

$$\widetilde{\mathbf{F}}_\phi = \begin{pmatrix} \sqrt{1^2+1^2}\,\sqrt{2^2+1^2} & 0 \\ 0 & \sqrt{1^2+1^2}\,\sqrt{2^2+0^2}\,\sqrt{1^2+0^2} \end{pmatrix}. \tag{3-40}$$

The zeros in $\mathbf{\Theta}$ are structurally significant for the model. Only the non-zero elements indicate an existing influence of one signal on the next states. Known zero entries thus decrease the number of parameters. This is important for a parameter identification problem. An approach on pre-defining the model structure is introduced next.

3-1-3 Structured MTI models

The sparse representation from the previous section shows, that the parameters of CPN models can be stored efficiently and that the zeros in the sparse matrix are of structural importance. This is even more important if we consider big systems with many data points. By assuming an influence of every signal on each state huge black-box models would be the result. This would be difficult for parameter identification and further use of the models, e.g. for diagnostic purposes, due to large memory requirements and long computation times.

Automated data point mapping is desirable in modeling for many applications, but it requires knowledge of data point types and their relationships, which is often lacking (Wang et al., 2018). A pre-structuring approach is proposed in the next section to reduce the number of data points considered in modeling by identifying or neglecting possible dependencies of data points on the examined states, (Schnelle et al., 2023).

Natural language processing dependency matrix

natural language processing (NLS) can be used to support the modeling of building systems by analyzing the names of monitoring data points. If the data points follow a standardized naming convention, such as the Buildings Unified Data point naming schema for Operation management (BUDO) scheme or similar approaches (see Stinner et al. (2018)), specific components of the names can be used to group related data points. However, in real-world applications, naming is often inconsistent, as different installers use their own conventions. In such cases, natural language processing methods, as discussed by Hirschberg and Manning (2015) and applied by García (2022), can help identify meaningful words and patterns in the names. This enables automatic grouping of data points and supports the modeling of sparse CPN models.

The notation($z_i \rightarrow z_j$) is used in the following for an existing influence from one signal z_i on another signal z_j.

A possible procedure for model structuring with NLS is proposed in Algorithm 1.

Algorithm 1 Natural Language Pre-Structuring (NLS), Schnelle et al. (2023)

1. Sorting data points in states and inputs
2. Building clusters with possibly dependent data points
3. Saving the structure in a Boolean sparse adjacency matrix $\mathbf{G}_s \in \mathbb{R}^{n\times(n+m)}$, with

$$\mathbf{G}_s = (g_{s_{ji}}) \text{ with } \begin{cases} g_{s_{ji}} = 1 & \text{for} \quad z_i \rightarrow z_j \\ g_{s_{ji}} = 0 & \text{else} \end{cases} \tag{3-41}$$

4. Building sub models for each state
5. Combining sub models in sparse representation using (3-30)

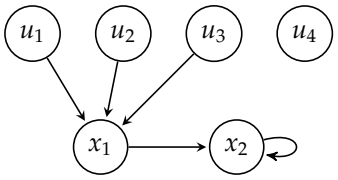

Figure 3-2: Signal dependency graph

Remark: It is essential to distinguish between states and other signals (inputs, set points, disturbances). In good naming schemes the type is detectable by NLS, in other cases expert knowledge is needed.

Example 9. Using the signal dependency graph of Figure 3-2, which determines the influence of signals x_1 to u_4 to the states x_1 and x_2, the model structure can be transferred in a sparse adjacency matrix

$$\mathbf{G}_s = \left(\begin{array}{c|cccccc} & x_1(k) & x_2(k) & u_1(k) & u_2(k) & u_3(k) & u_4(k) \\ \hline x_1(k+1) & 0 & 0 & * & * & * & 0 \\ x_2(k+1) & * & * & 0 & 0 & 0 & 0 \end{array}\right) \in \mathbb{R}^{n\times(n+m)}.$$

In this matrix, the Boolean `TRUE` marks all non-zero entries (denoted by $*$) indicating an existing influence of the signal on the next state. `FALSE` entries (0) are structural zeros, representing no dependency and can be fixed for a subsequent parameter identification of the CPN parameters.

In this context the Boolean matrix decomposition method described in Wicker et al. (2019) can be utilized to decompose $\mathbf{G}_s$. The decomposition gives the structure of $\mathbf{F}_\phi$ and $\mathbf{\Theta}^{\mathrm{T}}$ for the sparse representation from Definition 3-1.4. An approximation in the diagonal rank-n framework Definition 3-1.3 is achieved, as shown in the following by an example:

Example 10. The Boolean decomposition of the adjacency matrix of Example 9 is given with some abuse of notation given by

$$\mathbf{G}_s = \begin{pmatrix} * & 0 \\ 0 & * \end{pmatrix} \begin{pmatrix} 0 & 0 & * & * & * & 0 \\ * & * & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The nonzero entries are denoted with $*$. Only these need to be identified by parameter identification.

Alternative structuring methods are required for cases where NLS cannot be applied. If white-box models are available, Schnelle et al. (2023) proposes exporting the model structure for black-box CPN models from the Jacobian of a white-box model.

In the previous section the meaning of zero parameters in the adjacency matrix as well as in the parameter matrix of CPN models became clear. In order to generate further interpretation of the parameters for later diagnostic applications, these are examined in the following section.

3-1-4 Investigation of normalized parameters

The normalized representation of the multilinear models provides unique parameters, as discussed in subsection 3-1-2, which is essential for using these model parameters for diagnosis.

The uniqueness of the parameters of a polynomial for arbitrary state variable values is achieved by the normalization in subsection 3-1-1. This section discusses how normalization influences the uniqueness and interpretability of an MTI model.

Figure 3-3 shows how the function value $f(x) = 1 - |h_x| + h_x x$ of one factor of the multilinear polynomial behaves depending on the state variables x and on the parameters h_x in 1-norm representation. With regard to the uniqueness of the parameters, it can be seen that the

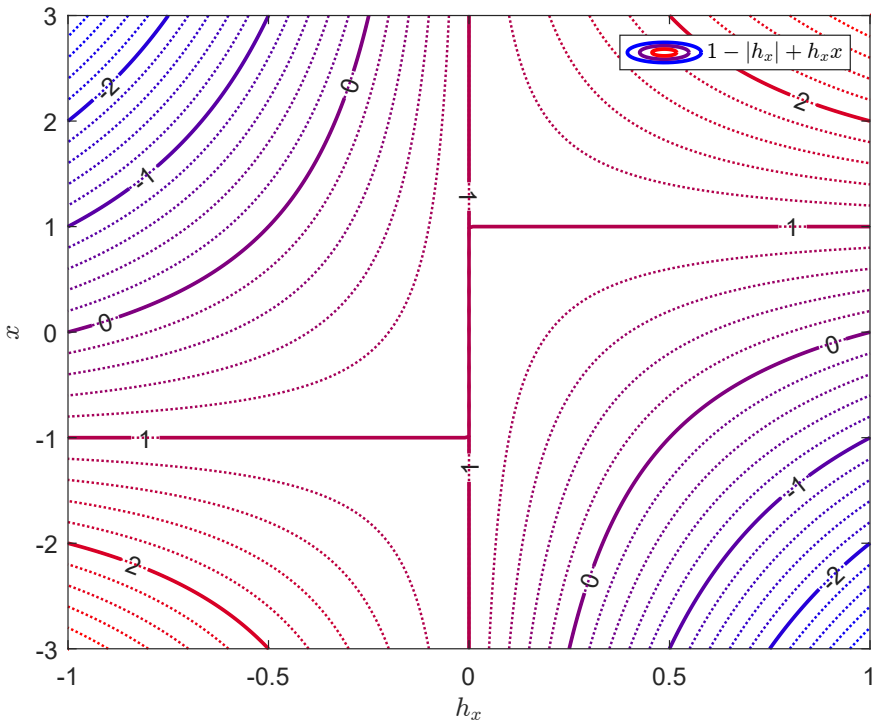

Figure 3-3: Contour function value of 1-norm factor

contour lines $0 \leq f(x) \leq 1$ are reached with one positive parameter as well as one negative parameter for some variable values in the range $-1 \leq x \leq 1$. The 1-contour line is a special case, as it is reached for state values $x = 1$ independent of the parameter's positive value and for state values $x = -1$ independent of the parameter when negative. These restrictions in uniqueness are only valid for constant variable values, as the following small example shows

$$f(x) = \left(1 - \left|\frac{1}{2}\right| + \frac{1}{2}x\right) = \left(1 - \left|-\frac{1}{3}\right| - \frac{1}{3}x\right) = \frac{3}{5} \qquad \text{for } x = \frac{1}{5}$$

$$f(x) = \left(1 - \left|\frac{1}{2}\right| + \frac{1}{2}x\right) \neq \left(1 - \left|-\frac{1}{3}\right| - \frac{1}{3}x\right) \qquad \text{for } x = \frac{1}{5} + \epsilon$$

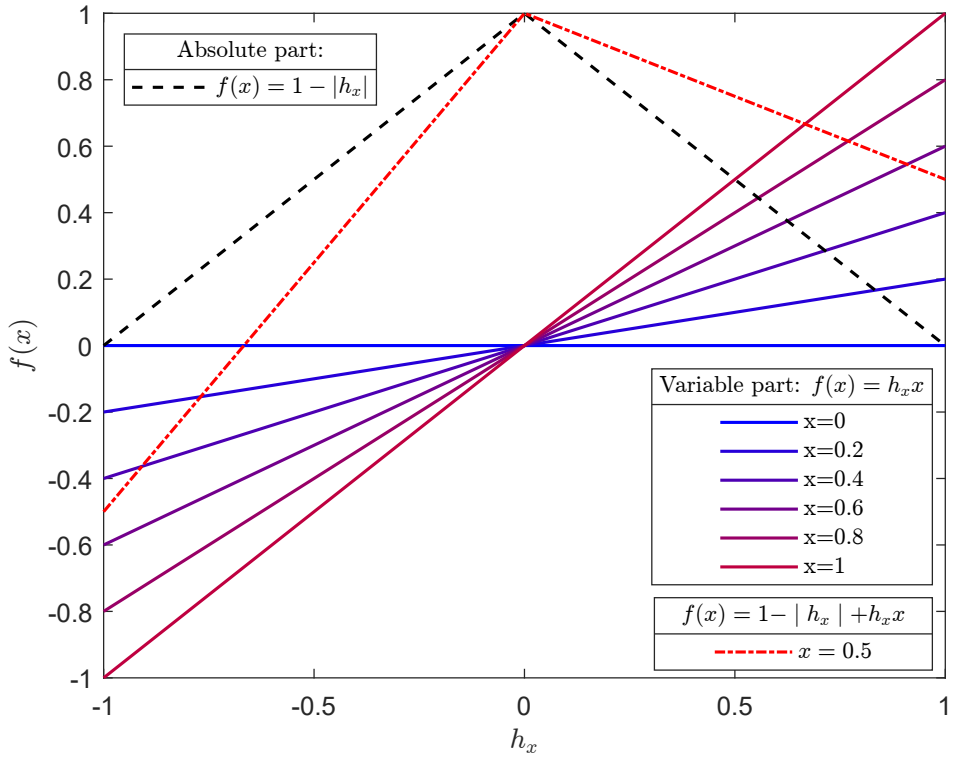

Figure 3-4: Function value of 1-norm factor

with $\epsilon \in \mathbb{R}\backslash\{0\}$. As the same function value is reached with one negative and one positive parameter, a constraint for non-negative parameter estimation can be considered later.

From the direct multiplication of the parameter with the variable in (3-5), interpretability of the parameters is assumed, which is discussed in the following. Figure 3-4 shows that the function value of one factor of the polynomial depends on a constant part, namely $1 - |h_x|$ and a variable part $h_x x$. In the norm-1 representation, the maximum value for $-1 \leq x \leq 1$ is reached when $1 - |h_x| + h_x x = 1$ (see Figure 3-3). For parameters $h_x \to 0$, the function value (example red dashed line in Figure 3-4) has a large proportion of the absolute part (black dashed line), while the variable part tends towards zero. Since one is the neutral element in multiplication, the entire state equation will not be influenced by the state variable in the factor when $h_x = 0$. In the direction of parameters with large magnitude, the influence of the variable part on the function value increases, while the absolute part decreases. The slope of the variable part depends on the value of the state variable. In general, the sum of the absolute and the variable part decreases and therefore differs more from one with increasing magnitude of parameters, as the example red dashed line for $x = 0.5$ shows.

It follows that the parameters of a 1-norm normalized MTI are interpretable: The function value of one polynomial factor is influenced by the variable x inside the factor depending on the magnitude of the parameter h_x. Applying this knowledge to state tracking in an MTI model in normalized polynomial representation allows the following statement: The influence of a signal (input variable or state variable) on the next input value increases with increasing magnitude of the parameter.

For the 2-norm representation, the contour lines in Figure 3-5 show the function value of one factor $f(x) = (\cos(\theta_x) + \sin(\theta_x)x)$ of the polynomial (3-2). The splitting in the absolute part $\cos(\theta_x)$ and variable part $\sin(\theta_x)x$ is illustrated in Figure 3-6. The general shape is similar to that of the 1-norm representation. The "1-line" contour in Figure 3-5 is also reached at

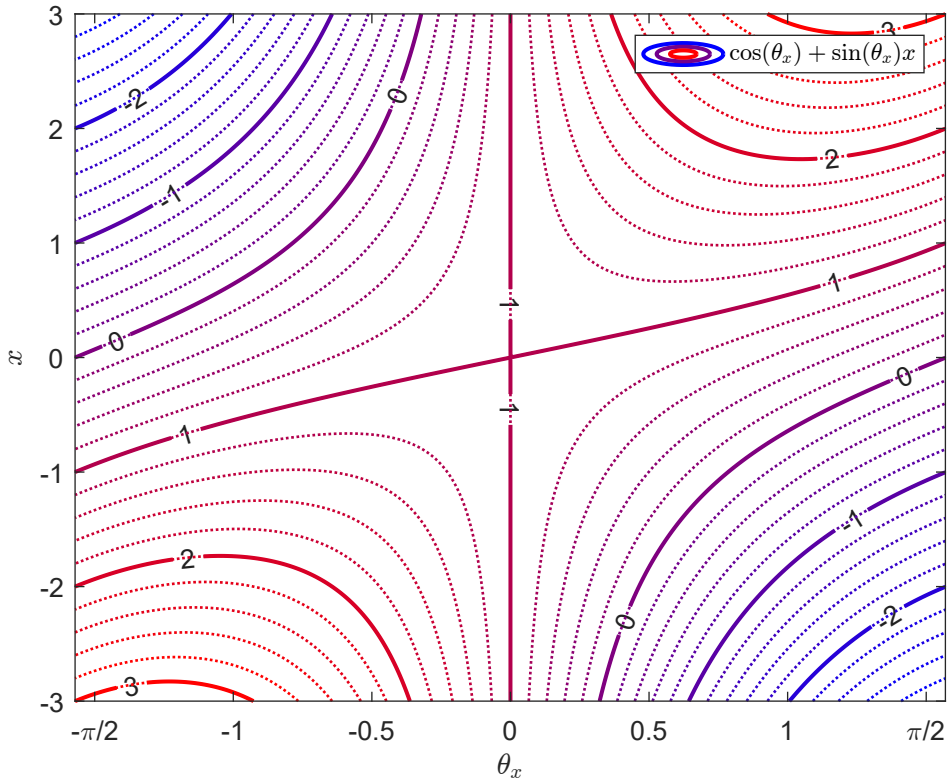

Figure 3-5: Contour function value of 2-norm factor

parameter value $\theta_x = 0$ for all possible state values. In direction of the horizontal axis, the "1-line" contour is not parallel to the horizontal axis as in the 1-norm representation. Therefore, in 2-norm representation there is no variable value x, where the function value is independent of the parameter θ_x. Similar to the case in 1-norm representation, the representation is not unique for single constant values of the variable x, because two parameter values result in the same function value. For function values with contour lines $0 \leq f(x) < 1$ the ambiguity is given with a pair of one negative and one positive parameter. For function values with contour lines $1 < f(x) < 2$ the ambiguity occurs for parameter pairs, which are either both positive or both negative and for contour lines with function values $f(x) \geq 2$ no ambiguity occurs.

To look at the interpretability of the parameters in 2-norm representation, Figure 3-6 gives the example line for $x = 0.5$ to show how it is composed by the absolute and the variable part. It is noticeable, that the variable part $f(x) = \sin(\theta_x)x = 0$ for parameter $\theta_x = 0$. The curves for the variable part are increasing depending on the variable value x with increasing parameters θ_x. The absolute part (dashed black line) is decreasing with increasing magnitude of θ_x. In direction of negative parameters both parts are decreasing. The sum of these two is the function value, which is denoted for $x = 0.5$ with the red dashed line. Interestingly, the

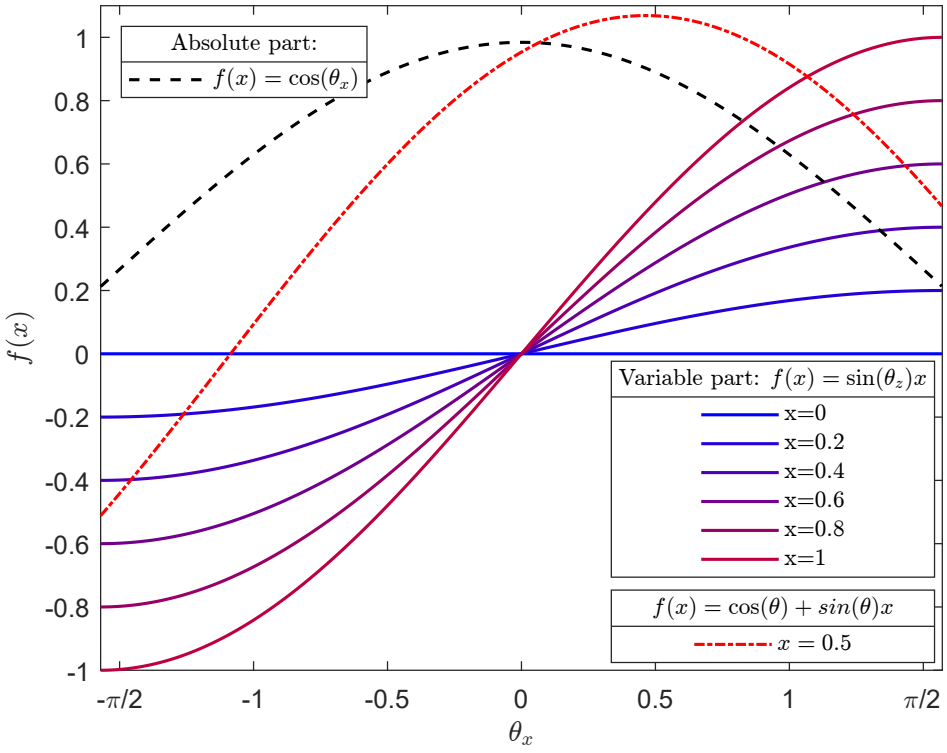

Figure 3-6: Function value of 2-norm factor

value one, which is the neutral element in multiplication, is reached twice for two positive parameters. The first time, the curve hits the one is for zero parameter, where the variable part has no influence and the second time is with parameter $\theta_x = 0.92$, where the one is achieved with a similarly high proportion of the absolute and the variable part. It should be noted that this ambiguity only applies to variables that remain constant, e.g. for $x = 0.5$.

The uniqueness and interpretability generally discussed in this section is shown below using an example.

Example 11. Consider the model of the example heated room from subsection 2-2-2 in CPN representation from Example 5. To investigate the interpretability of CPN parameters, the values of the parameter vector $\mathbf{h}_{T_{out}}$ are varied by a scaling factor a. In Figure 3-7, it is shown how the signals are influenced by the varying parameter. It is obvious that only the first state (return temperature) depends on the varying parameter. It can be clearly seen, that the parameter defines the dependency of the first state (T_r) on the first input T_{out}, which is the outside temperature. If the parameter is zero ($a = 0$), the return temperature does not depend on the outside temperature, which becomes visible because it is not following the shape of the outside temperature signal at all. With an increase of the corresponding parameter, the shape of the state signal follows more and more the shape of the outside temperature signal. This occurs due to the increasing variable part of the factor, which contains the outside temperature signal in the first state equation (3-26). On the other hand, smaller parameters lead to higher absolute parts. This affects the polynomial term and increases the dependency on the other signals of the term, which is in this example the second state ($\dot{V}_{air}$).

The example shows the interpretability of the CPN parameters, which is later useful for anomaly detection applications.

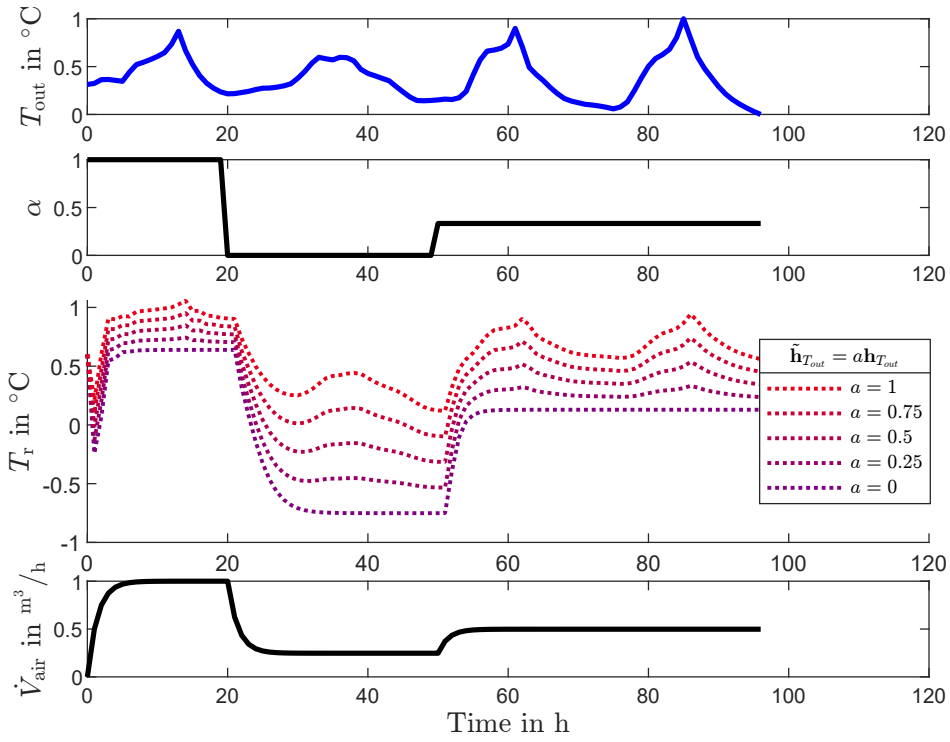

Figure 3-7: Example for interpretability of parameters

Remark: The ambiguity for constant variables is important for the identifiability of MTI models in CPN format. If the identified parameters are later used for diagnosis, uniqueness is important. Therefore, it should be considered not to use signals with low excitation and constant signals during the parameter identification procedure for the anomaly detection algorithm proposed later.

Remark: The uniqueness in the parameters for the polynomial factors including excited signals, can be transferred to the diagonal rank-*n* representation of the CPN MTI models. Rank-1 models are not suitable for systems with several states. An investigation of higher rank models is not carried out here. It is obvious that it is possible to add and subtract equal proportions in the rank-*r* data, which can lead to ambiguity.

As indicated, MTI models can be in continuous-time representation, which is needed when they are derived from differential equations. In most cases, discrete-time representations are needed when measurements are used for identification. The following section describes how to convert the continuous representation into a discrete time representation.

3-2 Transformation

In this section transformation of MTI models is proposed in CPN representation. First, it is shown, how a continuous-time CPN MTI can be transformed into a discrete-time representation. The second part of this section gives a possibility to scale a CPN MTI, such that the states stay within a given bound.

3-2-1 Discretization of normalized MTI models

The description of physical systems can be done with differential equations, which results in continuous-time state-space models as shown in subsection 2-2-2. In continuous-time models, the signals are given at every time $t \in \mathbb{R}$. If real data should be used for simulation, it is usually not available at every time, but the input signals and output measurements are collected by digital computers at discrete-time steps with a given sampling time. There are several methods available to approximate continuous-time differential equation systems with discrete-time difference equations to use in discrete-time state space models. The Euler forward method is applied to CP decomposed MTI models in Kruppa (2018) and is used here to develop a procedure for the CPN MTI models.

The continuous-time MTI model

$$\dot{\mathbf{x}}(t) = \left\langle \widetilde{\mathsf{F}}_{\mathsf{c}} \mid \mathsf{M}(\mathbf{x}(t), \mathbf{u}(t)) \right\rangle \tag{3-42}$$

with $\widetilde{\mathsf{F}}_{\mathsf{c}} \in \mathbb{R}^{n \times (n+m)2}$, holds all transition parameters of the continuous-time model and should be converted in a discrete-time representation

$$\mathbf{x}(k+1) = \left\langle \widetilde{\mathsf{F}}_{\mathsf{d}} \mid \mathsf{M}(\mathbf{x}(k), \mathbf{u}(k)) \right\rangle \tag{3-43}$$

with the converted parameter tensor $\widetilde{\mathsf{F}}_{\mathsf{d}} \in \mathbb{R}^{n \times (n+m)2}$.

The time derivative of the continuous state at a specific sample time can be approximated by the difference equation

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}(k+1) - \mathbf{x}(k)}{T_s}. \tag{3-44}$$

Therefore, the next state at time step $k+1$ is computed with

$$\mathbf{x}(k+1) \approx T_s \dot{\mathbf{x}}(kT_s) + \mathbf{x}(k). \tag{3-45}$$

To translate this into the normalized CP representation, the addition of the state is done by

$$\mathbf{x}(k+1) = T_s \left\langle \widetilde{\mathsf{F}}_{\mathsf{c}} \mid \mathsf{M}(\mathbf{x}(k), \mathbf{u}(k)) \right\rangle + \left\langle \widetilde{\mathsf{F}}_{\mathsf{x}} \mid \mathsf{M}(\mathbf{x}(k)) \right\rangle, \tag{3-46}$$

where the parameter tensor $\widetilde{\mathsf{F}}_{\mathsf{x}} \in \mathbb{R}^{n \times n2}$ adds the terms of the state from the monomial tensor $\mathsf{M}(\mathbf{x}(k)) \in \mathbb{R}^{n2}$ of the states.

Remember that in the normalized CPN representation the parameters can be stored in a reduced way inside matrices of all vectors of the second components of the normalized CP factors $\mathbf{F}_{x_1}, \mathbf{F}_{x_2}, ..., \mathbf{F}_{x_n}, \mathbf{F}_{u_1}, \mathbf{F}_{u_2}, ..., \mathbf{F}_{u_m}$ and an unnormalized matrix $\mathbf{\Phi} = \lambda \widetilde{\mathbf{F}}_{\phi}$. This leads to the following proposition of the discretization of 1-norm normalized continuous-time MTI models directly with the normalized factor matrices $\mathbf{H} \in \mathbb{R}^{(n+m) \times r}$.

Proposition 2. *The transition parameter matrices of the discrete-time MTI model*

$$\mathbf{H}_d = \left(\begin{array}{c|c} \mathbf{H}_c & \mathbf{H}_x \\ \hline & \mathbf{0} \end{array}\right) \tag{3-47}$$

and

$$\mathbf{\Phi}_d = \begin{pmatrix} T_s\mathbf{\Phi}_c & \mathbf{H}_x \end{pmatrix} \tag{3-48}$$

can be built by the original continuous-time factor matrix stacked together with a new factor matrix $\mathbf{H_x} = \mathbf{I}^n$ *for the selection of the states and* $\mathbf{0} \in \{0\}^{m\times n}$ *is a filling zero matrix.*

Proof. With the computation of the CPN MTI in the polynomial matrix form from Proposition 1, with

$$\mathbf{H}_d = \left(\begin{array}{c|c} \mathbf{H}_c & \begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{matrix} \\ \hline & \begin{matrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{matrix} \end{array}\right) \tag{3-49}$$

and

$$\mathbf{\Phi}_d = \left(\begin{array}{c|c} T_s\mathbf{\Phi}_c & \begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{matrix} \end{array}\right), \tag{3-50}$$

the discrete-time state equation in 1-norm representation using (4-10) is given by

$$\mathbf{x}(k+1) = \sum_{j=1}^{r} \phi_{d(:,j)} \prod_{i=1}^{n+m} (1 - |h_{d(i,j)}| + h_{d(i,j)} z_i). \tag{3-51}$$

By splitting the summation to

$$\mathbf{x}(k+1) = \sum_{j=1}^{r_c} \phi_{d(:,j)} \prod_{i=1}^{n+m} (1 - |h_{d(i,j)}| + h_{d(i,j)} z_i) + \sum_{l=r_c+1}^{r_c+1+n} \phi_{d(:,l)} \prod_{q=1}^{n} (1 - |h_{d(q,l)}| + h_{d(q,l)} z_q), \tag{3-52}$$

gives with r_c equals the rank of $\mathbf{H}_d$ the same function as in (3-45).

□

Remark: The discretization is shown in 1-norm representation here, for 2-norm the diagonal ones in (3-49) have to be replaced with $\frac{\pi}{2}$, as $\sin\frac{\pi}{2} = 1$, to get the discretized CPN in Euclidean

norm.
Remark: The concatenation of the matrices does not lead to a minimal representation of the discretized MTI model. If there is already a column with a single one in the first n rows of the continuous-time normalized factor matrix $\mathbf{F}_c$, no further identical columns need to be added. Both columns can be combined by keeping only one in $\mathbf{F}_d$ and adding the values of both columns in $\mathbf{\Phi}_d$.

Example 12. The continuous-time example from Example 2 can be approximated using a discrete-time representation by

$$x_1(k+1) = T_s\left(-0.0705x_1(k) + 0.0629x_2(k) - 0.0029x_1(k)x_2(k) + 0.0705u_1(k)\right) + x_1(k)$$
$$x_2(k+1) = T_s\left(-0.5x_2 + 262u_2\right) + x_2(k)$$
$$\mathbf{H} = \left(\begin{array}{ccccc|cc} 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right)$$
$$\mathbf{\Phi} = \left(\begin{array}{ccccccc} T_s & -0.0705 & 0.0629 & -0.0029 & 0.0705 & 0 & 1 & 0 \\ & 0 & -0.5 & 0 & 0 & 262 & 0 & 1 \end{array}\right)$$

With $T_s = 1$ h and the concatenation of the identical columns in $\mathbf{F}$, the rank of the model can be reduced to

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$
$$\mathbf{\Phi} = \begin{pmatrix} 0.9295 & 0.0629 & -0.0029 & 0.0705 & 0 \\ 0 & 0.5 & 0 & 0 & 262 \end{pmatrix}$$

The discretized model is simulated with a sampling time of one hour. The result of the approximation of the states is plotted in Figure 3-8, where the difference between the continuous-time model and discrete-time model is especially visible during periods of rapid changes in short time, but approximates the main dynamics of the system.

3-2-2 Scaling of normalized MTI models

Data collected from different physical signals with different units typically have different operating ranges, and their values can vary greatly. To avoid numerical problems, e.g. in parameter identification, the signals should be scaled to a defined interval, such as $\in [l, u]$. To work with the scaled signals while preserving the same dynamical behavior, the models can be transformed by linear state and input transformation with

$$z_i = a_i \underline{z_i} + b_i \tag{3-53}$$

and

$$\underline{\dot{z}_i} = a_i^{-1}(\dot{z}_i - b_i), \tag{3-54}$$

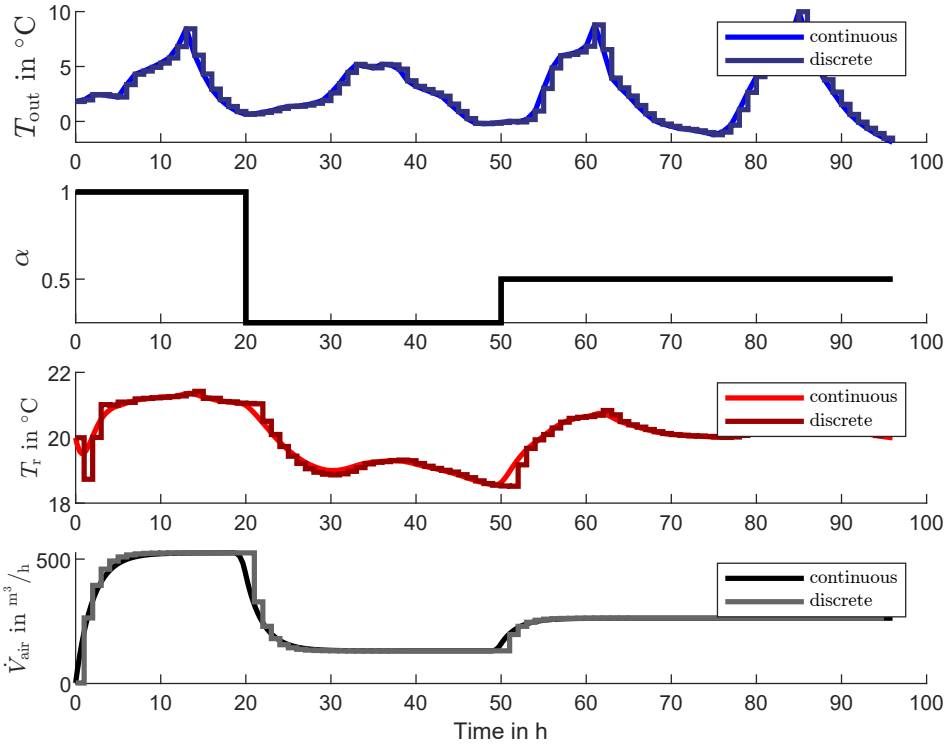

Figure 3-8: Discretized simulation of room example

with

$$a_i = \frac{z_{i,l} - z_{i,u}}{\underline{z}_{i,l} - \underline{z}_{i,u}} \tag{3-55}$$

$$b_i = z_{i,l} - \underline{z}_{i,l} a_i \tag{3-56}$$

where $z_i \in [z_{i,l}, z_{i,u}]$ is the original signal within the original operation range and $\underline{z}_i \in [\underline{z}_{i,l}, \underline{z}_{i,u}]$ is the scaled signal within the new interval. The index l and u stands for the lower bound and the upper bound and $i = 1, ..., n + m$. An approach to scale the MTI system in tensor representation is proposed in Kruppa (2018) and is used here for the adaptation of the normalized CPN representation.

Proposition 3. *A transformed MTI system in CP representation using* (2-37) *is given by*

$$\dot{\underline{\mathbf{x}}} = \left\langle \underline{\mathsf{F}} \,\middle|\, \mathsf{M}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \right\rangle . \tag{3-57}$$

In CP representation, the scaled parameter tensor

$$\underline{\mathsf{F}} = \left[\underline{\mathbf{F}}_{z_1}, \underline{\mathbf{F}}_{z_2}, ..., \underline{\mathbf{F}}_{z_{n+m}}, \underline{\mathbf{\Phi}} \right] \tag{3-58}$$

with factor matrices

$$\underline{\mathbf{F}}_{z_i} = \begin{pmatrix} 1 & b_i \\ 0 & a_i \end{pmatrix} \begin{pmatrix} \mathbf{F}_{z_i} & 1 \\ & 0 \end{pmatrix} \tag{3-59}$$

and

$$\underline{\mathbf{\Phi}} = \underset{i=1,\dots,n}{\operatorname{diag}}(a_i^{-1})\begin{pmatrix}\mathbf{\Phi} & -\mathbf{b}\end{pmatrix}, \tag{3-60}$$

where $\mathbf{b} = (b_1, b_2, ..., b_{n+m})^T$ *and* $\mathbf{F}_{z_i}$ *are the factor matrices of the original system. Normalizing* $\underline{\mathsf{F}}$ *using* (3-19) *or* (3-20) *yields the normalized scaled factor matrix* $\widetilde{\underline{\mathbf{F}}}$ *and* $\underline{\mathbf{\Phi}}$.

Proof. To scale the normalized MTI system

$$\dot{\mathbf{x}} = \sum_{j=1}^{r} \mathbf{\Phi}(:, j) \prod_{i=1}^{n+m} (\mathbf{F}_i(1, j) + \mathbf{F}_i(2, j) z_i) \tag{3-61}$$

with the scaling factors $\mathbf{a} = (a_1, a_2, ..., a_{n+m})$ and $\mathbf{b}$ to the scaled MTI system

$$\dot{\underline{\mathbf{x}}} = \sum_{j=1}^{r} \underline{\mathbf{\Phi}}(:, j) \prod_{i=1}^{n+m} \left(\underline{\mathbf{F}}_i(1, j) + \underline{\mathbf{F}}_i(2, j)\underline{z}_i\right), \tag{3-62}$$

where $\underline{\mathbf{F}}_i(2, j)$ is the second element in the j-th column of the normalized and scaled i-th factor matrix and $\underline{\mathbf{\Phi}}_{:,j}$ is the j-th column of the scaled $\underline{\mathbf{\Phi}}$-matrix, (3-53) and (3-60) are inserted into (3-62) to

$$\dot{\underline{\mathbf{x}}} = \underset{i=1,\dots,n}{\operatorname{diag}}(a_i^{-1})\left(\sum_{j=1}^{r} \mathbf{\Phi}(:, j) \prod_{i=1}^{n+m} \left(\underline{\mathbf{F}}_i(1, j) + \underline{\mathbf{F}}_i(2, j)(a_i\underline{z}_i + b_i)\right) - \mathbf{b}\right). \tag{3-63}$$

By multiplying out the inner brackets and rewriting it follows

$$\dot{\underline{\mathbf{x}}} = \sum_{j=1}^{r} \underset{i=1,\dots,n}{\operatorname{diag}}(a_i^{-1})\mathbf{\Phi}(:, j) \prod_{i=1}^{n+m} \left(\mathbf{F}_i(1, j) + \mathbf{F}_i(2, j)b_i + \mathbf{F}_i(2, j)a_i\underline{z}_i\right) - \underset{i=1,\dots,n}{\operatorname{diag}}(a_i^{-1})\mathbf{b}. \tag{3-64}$$

The scaled factor matrices are built by (3-59) to

$$\underline{\mathbf{F}}_i = \begin{pmatrix}1 & b_i\\ 0 & a_i\end{pmatrix}\begin{pmatrix}\mathbf{F}_{z_i} & 1\\ & 0\end{pmatrix} = \begin{pmatrix}\mathbf{F}_i(1,1) + \mathbf{F}_i(2,1)b_i & \dots & \mathbf{F}_{n+m}(1,r) + \mathbf{F}_i(2,r)b_i & 1\\ \mathbf{F}_i(2,1)a_i & \dots & \mathbf{F}_i(2,r)a_i & 0\end{pmatrix} \tag{3-65}$$

and

$$\underline{\mathbf{\Phi}} = \underset{i=1,\dots,n}{\operatorname{diag}}(a_i^{-1})\begin{pmatrix}\mathbf{\Phi} & -\mathbf{b}\end{pmatrix}. \tag{3-66}$$

By inserting $\underline{\mathbf{F}}_i$ and $\underline{\mathbf{\Phi}}$ into (3-62), (3-57) is fulfilled and the MTI system is scaled within the given bounds. □

The new scaled CP factor matrices can be normalized with (3-19) or (3-20).

Example 13. The discretized MTI model from Example 12 with $z_i \in [\min(z_i), \max(z_i)]$ is transformed into a scaled MTI model where $\underline{z}_i \in [0, 1]$. The simulation results of the scaled MTI model are shown in Figure 3-9. The dynamical behavior is the same as in the original model, but the signals are staying within the new boundaries.

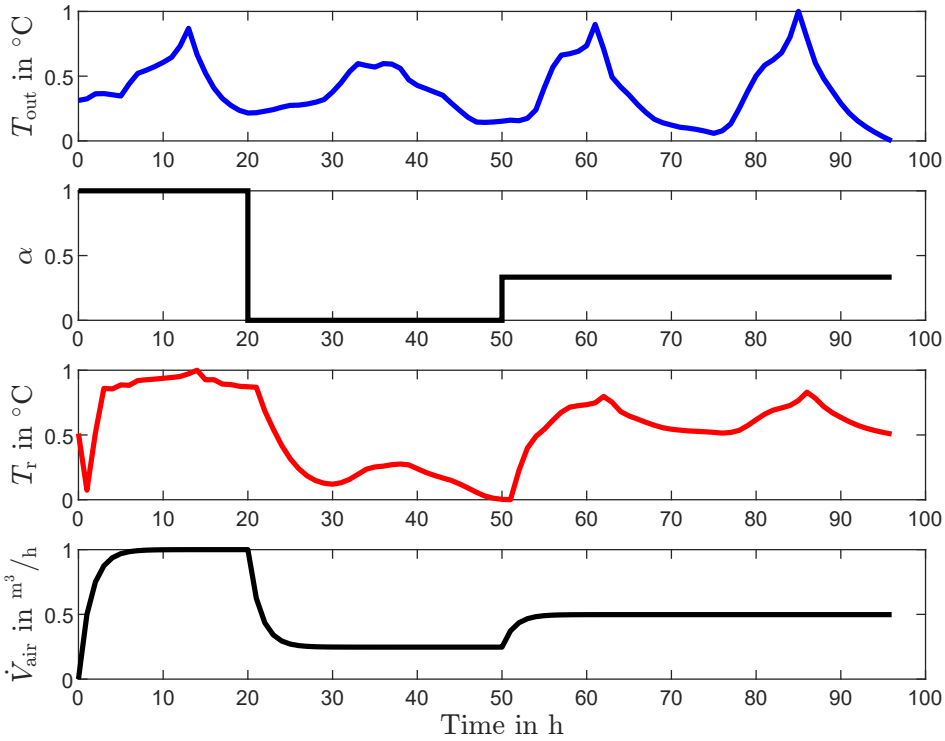

Figure 3-9: Scaling of MTI model

Chapter 4

Black-box parameter identification

Building white-box models of multilinear systems is trivial, if the differential equations and their parameters are known, as shown in Example 2. With less prior knowledge about the system, gray-box models need to be identified using multilinear parameter identification algorithms, which are rarely found in literature yet, (Sridharan et al., 2020). Efficient black-box algorithms for multilinear state space model identification, as they are available for linear systems, e.g. with the `n4sid` methods from Van Overschee and De Moor (1993) are in research state, (Sridharan et al., 2020). As shown in the previous chapter, the multilinear time-invariant (MTI) models can be represented in a full tensor representation or in a decomposed tensor representation, which leads to a normalized polynomial description in a further reduced form. A multilinear system identification method for Tensor Train matrices is proposed in Batselier et al. (2018), which cannot be adapted to the Canonical Polyadic (CP) decomposition used in this thesis. Thus, a new parameter identification approach for multilinear models is developed in this chapter. In the following, two MTI model representations are used to develop and investigate a parameter identification procedure for both the full tensor representation and the decomposed normalized representation.

4-1 Parameter identification for MTI models

In the black-box parameter identification approach chosen here, all parameters and structural information of the model are unknown. The only constraint is the model class of normalized discrete-time MTI models. Historical or experimental data, such as input signals and state measurements, are assumed to be available.

The general procedure of linear or nonlinear parameter identification from monitoring data is given in Isermann and Münchhof (2011). To identify the model parameters that are mapped in the parameter tensor F of an MTI in any representation, the procedure illustrated in Figure 4-1 is used, (Schnelle et al., 2022). In this case, the state variables $\tilde{\mathbf{x}}(k)$ are the variables measured at discrete times $k = 0, 1, \ldots, N$ in the building, while $\mathbf{x}(k)$ represents the calculated

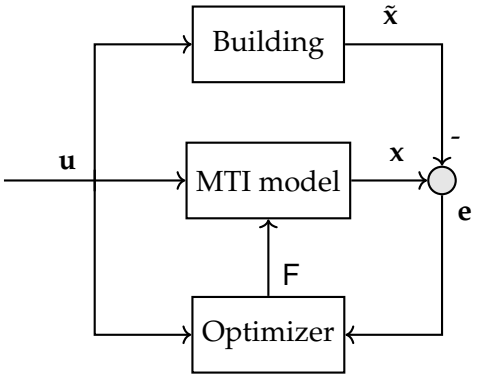

Figure 4-1: Parameter identification for MTI models

state variables of the normalized CP decomposed (CPN) MTI model from Definition 3-1.1. An error vector

$$\mathbf{e}(k) = \mathbf{x}(k) - \tilde{\mathbf{x}}(k) \tag{4-1}$$

is computed at each time step k as the deviation between the measured states and the model in order to obtain an error sum

$$E = \sum_{k=1}^{N} \mathbf{e}'(k)\mathbf{Q}\ \mathbf{e}(k)\ , \tag{4-2}$$

from the weighted error squares. The matrix $\mathbf{Q}$ can be chosen as the identity matrix in the case of uniform weighting, which is suitable if all state and input data is scaled within an interval $[0, \ldots, 1]$, as described in subsection 3-2-2.

The input $\mathbf{u}$ to the plant is also used as input for the MTI model, while the task is to minimize the error $\mathbf{e}$ between the measured state vector $\tilde{\mathbf{x}}$ and the model state $\mathbf{x}$ to estimate the transition tensor F of the low-rank MTI model, as shown in Problem 4-1.1.

Problem 4-1.1. Fixed rank MTI parameter identification, (Schnelle et al., 2022)

Given:

- State measurements $\tilde{\mathbf{x}}(k)$ for $k = 0, 1, \ldots N$
- Input variables $\mathbf{u}(k)$ for $k = 0, 1, \ldots N-1$
- rank r

Find:

- fixed-rank r parameter tensor F represented as
- normalized CP factors $\widetilde{\mathbf{F}}_i$ and weighting $\mathbf{\Phi}$

to minimize the squared error sum (4-2) by

$$\min_{\widetilde{\mathbf{F}}_i, \mathbf{\Phi}} \quad E. \tag{4-3}$$

The result is a parameter set of a normalized CP decomposed MTI model that captures the dynamics of the nominal building behavior with unique and interpretable parameters, which are essential for anomaly detection, see subsection 3-1-4. For solving the optimization problem different methods are discussed in the next sections.

4-2 Estimation of full tensor MTI models from data

This section investigates how the parameters of an MTI model in full tensor representation can be identified from the building data. To convert the full parameter tensor into the desired form of a CPN MTI for further use in anomaly detection, it must then be decomposed and normalized. At the end of the section, the optimization problem and its scalability are discussed.

4-2-1 Solving system of multilinear equations

The parameter identification of the parameters of a multilinear model in full tensor representation from Definition 2-1.4 is straightforward. As the resulting state equations are linear in the parameters, standard methods from linear algebra can be used. A multilinear state space model

$$\mathbf{x}(k+1) = \langle\, \mathsf{F} \mid \mathsf{M}(\mathbf{x}(k), \mathbf{u}(k)) \,\rangle\,,$$

from (2-37) contains constants, the states and inputs and all multilinear combinations of states and inputs multiplied each by one parameter of the parameter tensor F. This ends up in a system of n multilinear polynomials with 2^{n+m} summands and an overall number of $n2^{n+m}$ parameters. To identify the parameters of the model a sequence of at least $n2^{n+m}$ time steps k is needed, to build an overdetermined system of equations. The Least Squares method (2-12) to (2-22) is used to minimize the error between the model and the measurement data.

By unfolding the tensor F to a matrix $\mathbf{F}$ of dimension $\mathbb{R}^{n\times 2^{n+m}}$ the linear optimization problem can be written as

$$\min_{\mathbf{F}} \|\mathbf{X} - \mathbf{M}\mathbf{F}^{\mathrm{T}}\|\,, \tag{4-4}$$

where

$$\mathbf{X} = \begin{pmatrix} \tilde{x}_1(1) & \tilde{x}_2(1) & \dots & \tilde{x}_n(1) \\ \tilde{x}_1(2) & \tilde{x}_2(2) & \dots & \tilde{x}_n(2) \\ \vdots & \vdots & \dots & \vdots \\ \tilde{x}_1(N) & \tilde{x}_2(N) & \dots & \tilde{x}_n(N) \end{pmatrix} \in \mathbb{R}^{N\times n} \tag{4-5}$$

contains all N next state measurements and

$$\mathbf{M} = \begin{pmatrix} \mathbf{m}(\mathbf{x}(0), \mathbf{u}(0)) \\ \mathbf{m}(\mathbf{x}(1), \mathbf{u}(1)) \\ \vdots \\ \mathbf{m}(\mathbf{x}(N-1), \mathbf{u}(N-1)) \end{pmatrix} \in \mathbb{R}^{N \times 2^{n+m}} \tag{4-6}$$

is a matrix of all elements of the monomial matrix of the MTI for the time steps $k = 1, ..., N$ with

$$\mathbf{m}(\mathbf{u}(k), \mathbf{x}(k)) = \begin{pmatrix} 1 \\ u_m(k) \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ u_1(k) \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ x_n(k) \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ x_1(k) \end{pmatrix} \in \mathbb{R}^{2^{n+m}}, \tag{4-7}$$

where the Kronecker product $\otimes$ gives all monomial elements for one time step in a vector, (Lichtenberg et al., 2022). To solve for the parameters in $\mathbf{F}$, standard methods from linear regression, see Groß (2003) can be used to rewrite (4-4) in the form of (2-22) and solve

$$\mathbf{F} = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1}\mathbf{M}^{\mathrm{T}}\mathbf{X}. \tag{4-8}$$

To illustrate the procedure of full tensor MTI parameter identification a second order MTI system is used in the example given next.

Example 14. The state equation of a second order multilinear model

$$\mathbf{x}(k+1) = \begin{pmatrix} f_{1,0} + f_{1,1}x_1(k) + f_{1,2}x_2(k) + f_{1,3}x_1(k)x_2(k) \\ f_{2,0} + f_{2,1}x_1(k) + f_{2,2}x_2(k) + f_{2,3}x_1(k)x_2(k) \end{pmatrix}$$

has $2 \cdot 2^2$ parameters $f_{i,j}$ with $i = 1, 2$ and $j = 0, \ldots, 3$. To solve the linear equation for the parameters, the state measurement for the last $N >= 8$ time steps are used to build the system of linear equations

$$\begin{pmatrix} x_1(k+1) & x_2(k+1) \\ x_1(k+2) & x_2(k+2) \\ \vdots & \vdots \\ x_1(k+N) & x_2(k+N) \end{pmatrix} - \begin{pmatrix} 1 & x_1(k) & x_2(k) & x_1x_2(k) \\ 1 & x_1(k+1) & x_2(k+1) & x_1x_2(k+1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_1(k+N-1) & x_2(k+N-1) & x_1x_2(k+N-1) \end{pmatrix} \begin{pmatrix} f_{1,0} & f_{2,0} \\ f_{1,1} & f_{2,1} \\ f_{1,2} & f_{2,2} \\ f_{1,3} & f_{2,3} \end{pmatrix} = 0$$

As specified in Problem 4-1.1, the MTI parameters are required in a normalized decomposed representation. In order to use them later for anomaly detection in real buildings, a memory-reduced form and the uniqueness and interpretability of the CPN parameters are required. Therefore, the identified parameters from the full tensor format are decomposed with the *cp_als* from Kolda and Bader (2009), which performs an alternating least squares optimization to decompose a full tensor into a CP tensor with fixed rank. The normalization is done with (3-6) and (3-4) to gain the representation from (3-16).

4-2-2 Investigation of optimization problem

To investigate the MTI identification algorithm described above, the simulation data from the original differential equations of the room from Example 2 is used. The identification of a full tensor MTI model is performed with the method from subsection 4-2-1 using (4-4) to (4-8) in MATLAB. In order to minimize numerical errors, which can occur when the magnitude of different signals differs greatly, the data of the input signals and the state variables are scaled in the interval $0 \leq \mathbf{x} \leq 1$ and $0 \leq \mathbf{u} \leq 1$ first with (3-53).

As shown in Figure 4-2, the identified full tensor MTI model tracks the states, i.e. the room temperature T_r (sub figure 3) and the volume flow signal of the air unit $\dot{V}_{air}$ (sub figure 4) without visible deviations. As described before, an MTI model in CP decomposed and normalized form is required for the further use in anomaly detection, therefore, the full tensor MTI model is decomposed with *cp_als* algorithm, see Bader et al. (2019), to a rank $r = 6$ representation and normalized in norm-1 representation with the procedure described in subsection 3-1-2. The result of simulation data of the identified CPN MTI model in comparison to the original data is shown in Figure 4-3. The model represents the state values (T_r and $\dot{V}_{air}$) still very well. Small deviations between the data (solid line) and the model (dashed line) are mainly visible between 30 and 40 hours of simulation.

Where the full tensor identification by simply solving a system of linear equations is straightforward, the tensor decomposition with `cp_als` is a complex optimization problem, (Kolda and Bader, 2009) To investigate the results of the cost function of the solutions, the identification is repeated 100 times. The mean cost and standard deviation of the squared error (4-4) are depicted in Table 4-1 for the full tensor solution and the *cp_als* solution. The distribution of the *cp_als* solution is illustrated in Figure 4-4. It is notable that the full tensor solution achieves the same error for across 100 observations, with almost zero standard deviation, while the solution of the normalized CPN MTI model shows a distribution of error results. The shape of the distribution is exponentially decreasing with the highest number of observations in the smallest error interval of 0 to 0.1. In addition, individual outliers can be found for larger error sums, up to 2.

Method	**Error mean**	**Standard deviation**
Full tensor	6.3841e-15	7.9284e-31
CP tensor	0.1201	0.2018

Table 4-1: Error metrics for tensor identification methods

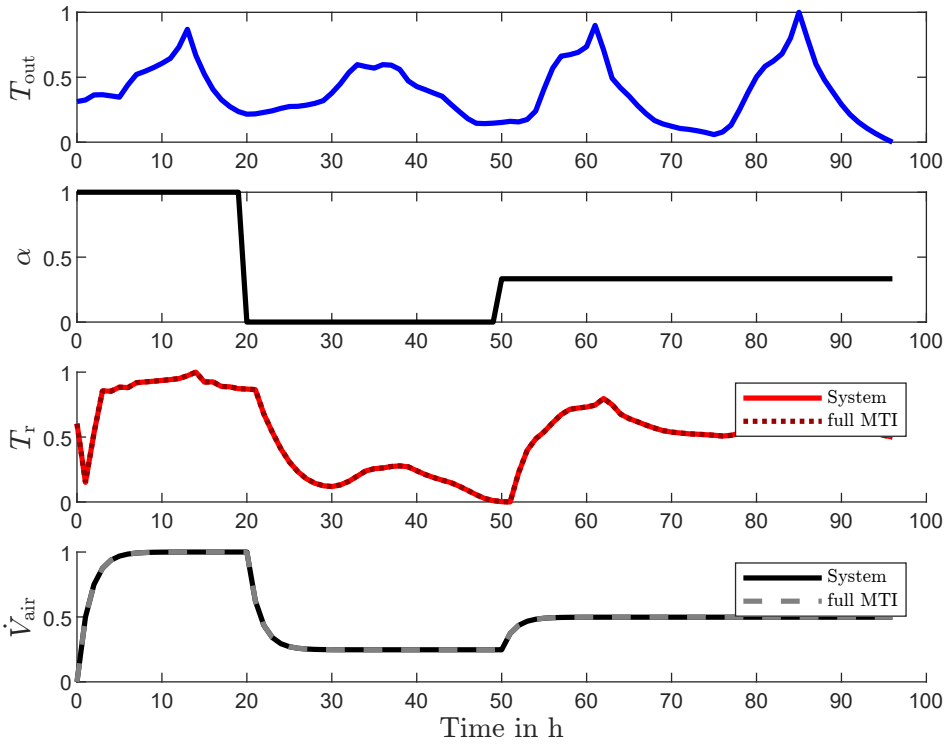

Figure 4-2: Full tensor MTI identification

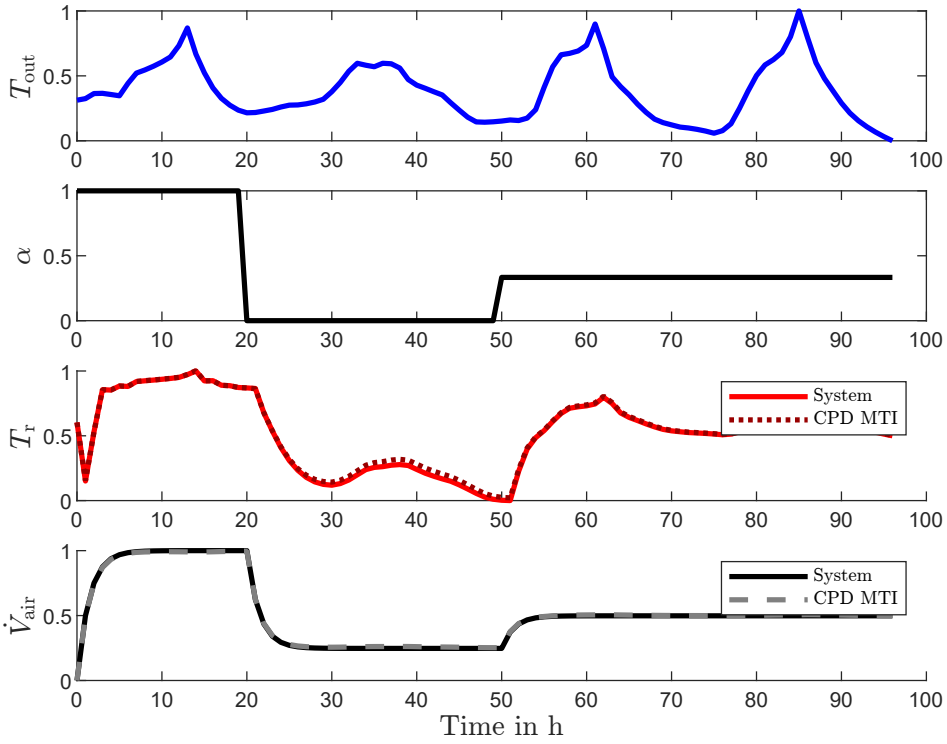

Figure 4-3: Full tensor to norm-1 CPN MTI identification

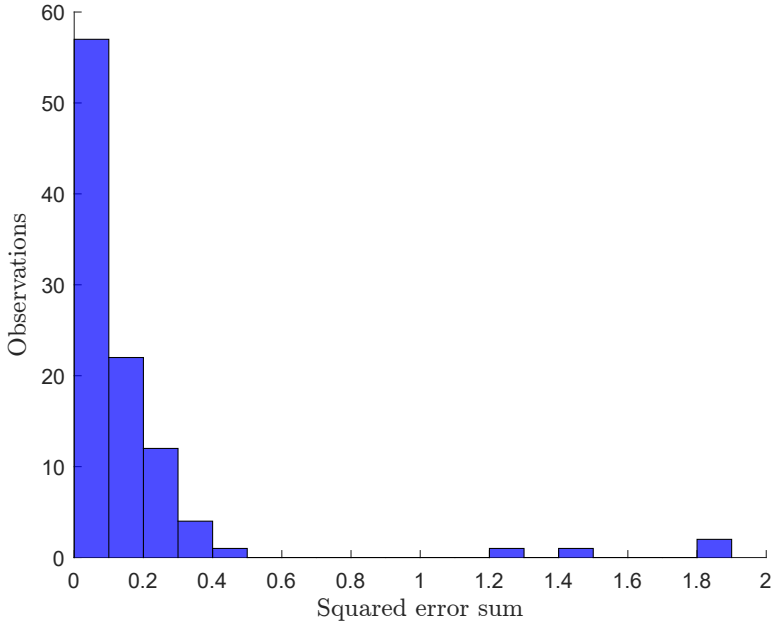

Figure 4-4: Cost function distribution for full tensor to norm-1 CPN MTI identification

4-2-3 Scalability

To use the prescribed parameter identification procedure for anomaly detection in building systems, it needs to be scalable for big buildings. The full tensor identification is, on one hand, easy to compute but has the disadvantage that the number of parameters grows exponentially, with every state and input, as demonstrated in Figure 4-5.

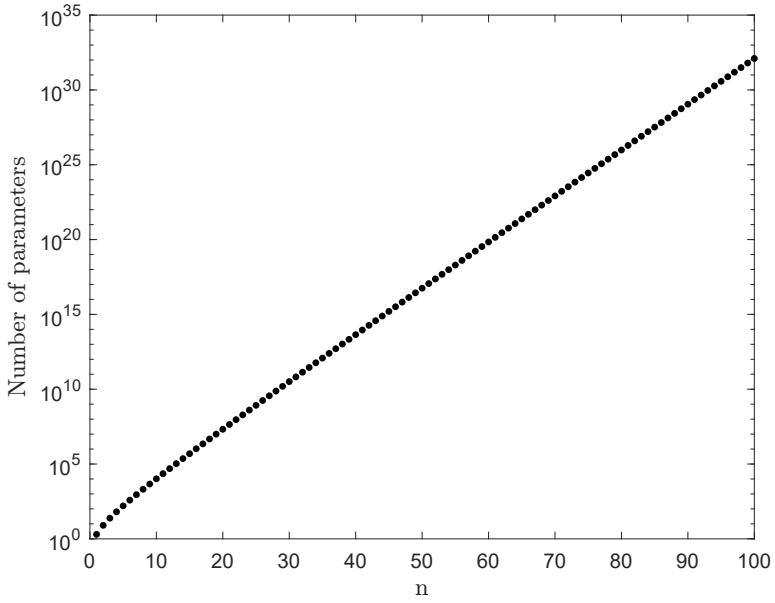

Figure 4-5: Scalability of full parameter tensor

This results in big models, which lead to high memory demand and computational costs, thus restrict the method to smaller systems. A second limitation is that the system of equations needs to be overdetermined to get unique solutions for all parameters. Therefore, the amount of data needs to be large enough to fulfill the criteria of overdetermination. With growing number of states and inputs, the hours of collected data increases also exponentially. For the identification of a fifth order model with 160 parameters, at least 2.7 days of data with a one-minute sampling interval are needed, and the identification of a 15th order model would require already 342 days of sampling data and would have 491520 parameters, see Table 4-2. A standard laptop with 16 GB RAM would run out of memory by trying to create a full tensor MTI model with 27 states or inputs. Therefore, full tensor identification is not suitable for large models and decomposed methods need to be developed as shown in the following section.

Table 4-2: Full tensor optimization

Order	Parameter	Bytes	Data in hours	Data in days	Data in years
2	8	64	0.133	0.00556	1.52e-05
5	160	1280	2.67	0.111	0.000304
10	10200	81900	171	7.11	0.0195
15	4.92e+05	3.93e+06	8190	341	0.935
27	3.62e+09	2.9e+10	6.04e+07	2.52e+06	6890
50	5.63e+16	4.5e+17	9.38e+14	3.91e+13	1.07e+11
100	1.27e+32	1.01e+33	2.11e+30	8.8e+28	2.41e+26

4-3 Estimation of decomposed MTI models from data

The identification of MTI parameters in the full tensor representation achieves good results for small models but is not applicable to larger systems. Therefore, a parameter identification method is developed that optimizes the parameters of the reduced normalized CP MTI (CPN) model directly from the data. First, a standard nonlinear algorithm is used and then compared with a newly developed specialized alternating least squares algorithm.

4-3-1 Solution and investigation with standard algorithms

The direct identification of the parameters of the decomposed normalized model is the desired option due to the exponential growth in the number of parameters of the non-decomposed MTI approach, shown in Figure 4-5. First, the optimization is performed with standard nonlinear optimizers in MATLAB to study the optimization problem. To identify the parameters of the normalized MTI model (3-16) with the method described in Problem 4-1.1, various nonlinear optimization methods are available. To identify the parameters in normalized representation, a nonlinear optimization method with bounded parameter values is required. MATLAB provides the constrained optimization function *fmincon* for nonlinear optimization in the *Optimization Toolbox*, (MathWorks, 2023b). The optimization problem from Problem 4-1.1 can be rewritten as

$$\min_{h_i,\mathbf{\Phi}} E \text{ such that } lb \leq h_i \leq ub, \tag{4-9}$$

with $lb = 0, ub = 1$, to ensure a solution with 1-norm normalized model parameters. To compute the error $\mathbf{e}$ from (4-1), the model states

$$\mathbf{x}(k+1) = \mathbf{\Phi} \begin{pmatrix} \prod_{i=1}^{n+m}(1 - |h_{i,1}| + h_{i,1}z_i(k)) \\ \prod_{i=1}^{n+m}(1 - |h_{i,2}| + h_{i,2}z_i(k)) \\ \vdots \\ \prod_{i=1}^{n+m}(1 - |h_{i,r}| + h_{i,r}z_i(k)) \end{pmatrix} \tag{4-10}$$

are generated according to Proposition 1. Most of the options of the algorithm are set to the MATLAB default parameters, see Table 4-3.

Table 4-3: Parameter table for *fmincon* algorithm

Parameter	**Value**
Algorithm	Interior point
Max Function Evaluations	3000
Objective Limit	-1.0000e+20
Optimality Tolerance	1.0000e-06
Step Tolerance	1.0000e-10
Maximum Iterations	3000
Constraint Tolerance	1.0000e-06

If the optimization problem is convex, convex optimization methods could be used. This fact is examined below:

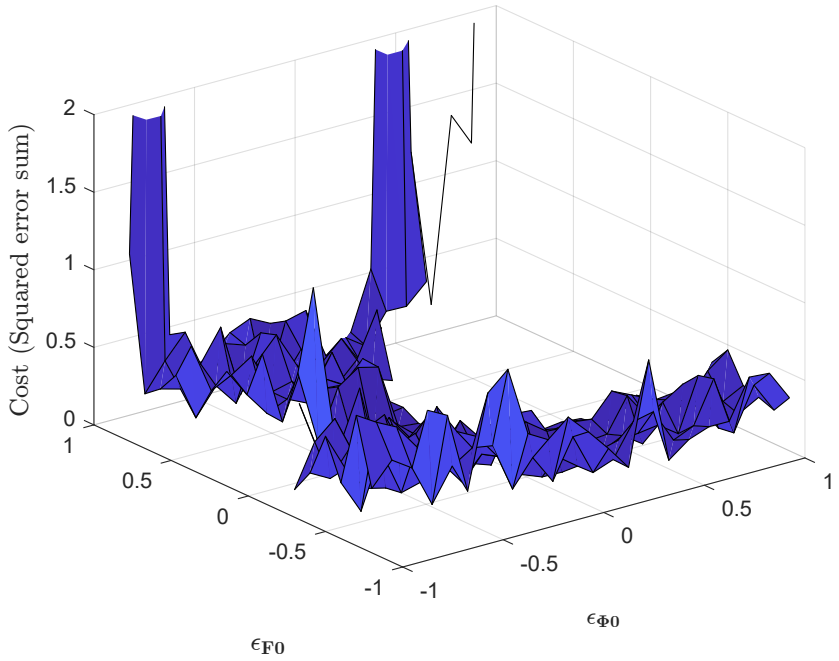

Figure 4-6: Cost function value E with varying initial parameters

Assumption 1. The optimization problem (4-3) is convex.

To check the assumption, the necessary condition for convex functions, that all local minima coincide with the global minimum, is tested experimentally. For this purpose, the optimization problem from Problem 4-1.1, using the nonlinear MATLAB *fmincon* optimizer, is initialized with varying initial parameters

$$\begin{aligned} \mathbf{H0} &= \mathbf{H} + \epsilon_{\mathbf{H0}} \mathbb{1}_{(n+m)\times R} \\ \mathbf{\Phi 0} &= \mathbf{\Phi} + \epsilon_{\mathbf{\Phi}} \mathbb{1}_{n\times R}, \end{aligned} \tag{4-11}$$

where $\mathbb{1}_{i\times j}$ is a matrix of ones with dimension $(i \times j)$ and $\epsilon_{\mathbf{F0}} = \epsilon_{\mathbf{\Phi}} = -1, -0.9, ..., 0, 0.1, ..., 1$ varies with each new run of the algorithm. The matrices $\mathbf{H}$ and $\mathbf{\Phi}$ are the exact parameters of the 1-norm CPN MTI model derived from the multilinear function to fit the data. For the identification, the data from Example 2 is used to identify a rank-6 MTI model. The costs achieved from (4-3) are plotted in Figure 4-6 against ϵ.

Contradiction: The objective function results in different values when the optimization is initialized with different initial conditions. Thus, local minima do not coincide with the global minimum and the Assumption 1 is contradicted. It is apparent that the optimization problem for all CPN MTI models is not convex. This optimization problem has a global minimum, which is reached when the initial parameter set matches the parameter set of the multilinear system derived from differential equations.

4-3-2 Alternating least squares algorithm

The connection between the multilinear model class and tensors enables the use of the full range of tensor algebra, including decomposition and normalization techniques, which can significantly reduce computational costs and improve efficiency. Parameter identification with the described nonlinear standard optimizers is inefficient due to the lack of adaptation to the multilinear models. For this reason, an efficient algorithm is developed in this thesis and used in Schnelle and Lichtenberg (2024) specifically for the specific structure of a normalized CP decomposed MTI models. The alternating least squares algorithm is already widely used for the factorization of matrices and has been adapted for the decomposition of CP tensors, see Kolda and Bader (2009), and to system identification of MIMO Volterra systems in Batselier et al. (2017). In the following, an alternating least squares (ALS) algorithm for normalized MTI models is proposed. For better comprehension, the algorithm is first described with a rank-1 limitation, and later adapted to rank-n models.

Rank-1 alternating least squares algorithm

A rank-1 multilinear state equation is represented as the factorized polynomial

$$\mathbf{x}(k+1) = (f_{i,1} + f_{i,2} z_i(k)) \prod_{j=J(1)}^{J(n+m-1)} (1 - |f_j| + f_j z_j(k)) \mathbf{\Phi}, \tag{4-12}$$

where the second part of the equation is normalized as shown in subsection 3-1-2 and $f_{i,1}$ and $f_{i,2}$ are free parameters of a decomposed, non-normalized multilinear model from (2-41). The index i addresses one single dimension out of the $n+m$ dimensions of the parameter tensor

$$\mathsf{F} = \left[\begin{pmatrix} 1 - |f_1| \\ f_1 \end{pmatrix}, \begin{pmatrix} 1 - |f_2| \\ f_2 \end{pmatrix} \cdots, \begin{pmatrix} f_{i,1} \\ f_{i,2} \end{pmatrix}, \begin{pmatrix} 1 - |f_{i+1}| \\ f_{i+1} \end{pmatrix}, \cdots, \begin{pmatrix} 1 - |f_{n+m}| \\ f_{n+m} \end{pmatrix}, \begin{pmatrix} \Phi_1 \\ \vdots \\ \Phi_n \end{pmatrix} \right] \tag{4-13}$$

in CP representation from (2-41). The index $j \in J = (1, 2, ...i-1, i+1, ..., n+m)$ includes all other fixed parameters, except for the free parameters with the current index i. By fixing all but one dimension, the multilinear state equation can be expressed as a linear equation

$$\mathbf{x}(k+1) = \left(\prod_{j=J(1)}^{J(n+m-1)} (1 - |f_j| + f_j z_j(k)) \Phi \quad \prod_{j=J(1)}^{J(n+m-1)} (1 - |f_j| + f_j z_j(k)) \Phi z_i(k) \right) \mathbf{V} \tag{4-14}$$

with the vector of free parameters

$$\mathbf{V} = \begin{pmatrix} f_{i,1} \\ f_{i,2} \end{pmatrix} \in \mathbb{R}^{2 \times 1}. \tag{4-15}$$

Defining a vector of all next state measurements

$$\mathbf{X} = \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \\ x_1(k+2) \\ x_2(k+2) \\ \vdots \\ x_n(k+2) \\ \vdots \\ x_1(k+N) \\ x_2(k+N) \\ \vdots \\ x_n(k+N) \end{pmatrix} \in \mathbb{R}^{(nN)\times 1}, \tag{4-16}$$

for time steps $k = 1, 2, \ldots, N$ leads to a linear overdetermined system of equations

$$\mathbf{X} = \mathbf{U}\mathbf{V} \tag{4-17}$$

with

$$\mathbf{U} = \begin{pmatrix} \prod_{j=J(1)}^{J(n+m-1)} \left(1 - \left|f_j\right| + f_j z_j(k)\right) \Phi & \prod_{j=J(1)}^{J(n+m-1)} \left(1 - \left|f_j\right| + f_j z_j(k)\right) \Phi z_i(k) \\ \prod_{j=J(1)}^{J(n+m-1)} \left(1 - \left|f_j\right| + f_j z_j(k+1)\right) \Phi & \prod_{j=J(1)}^{J(n+m-1)} \left(1 - \left|f_j\right| + f_j z_j(k+1)\right) \Phi z_i(k+1) \\ \vdots & \vdots \\ \prod_{j=J(1)}^{J(n+m-1)} \left(1 - \left|f_j\right| + f_j z_j(k+N)\right) \Phi & \prod_{j=J(1)}^{J(n+m-1)} \left(1 - \left|f_j\right| + f_j z_j(k+N)\right) \Phi z_i(k+N) \end{pmatrix}, \tag{4-18}$$

which is a matrix of dimension $\mathbf{U} \in \mathbb{R}^{nN\times 2}$ and contains the constants given by the multilinear polynomial of fixed parameters and states. This results in a separation of the variable $\mathbf{V}$ from the constants $\mathbf{U}$.

The number of time steps $N >= 2r$ must be chosen sufficiently large to obtain an overdetermined system with $k = 1, 2, ..., N$ linear equations

$$\mathbf{x}(k+1) = \mathbf{U}(k,:)\mathbf{V} + \mathbf{e}(k), \tag{4-19}$$

where the error $\mathbf{e}(k)$ represents the difference between the measured state and the function value of the one-step prediction.

The optimization problem consists of minimizing the quadratic error sum

$$E = \mathbf{e}^{\mathrm{T}}\mathbf{e} = ||\mathbf{X} - \mathbf{U}\mathbf{V}||^2 \tag{4-20}$$

and can be solved using the standard linear least squares method (2-22) by solving

$$\mathbf{V} = (\mathbf{U}^{\mathrm{T}}\mathbf{U})^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{X}. \tag{4-21}$$

To apply the idea of alternating least squares to the parameter identification problem for the normalized parameters of a discrete time MTI, the index i of the free parameter dimension alternates between 1 and $n+m$. The optimization problem 4-20 is solved for the parameters of the free dimension and then its index changes to the next dimension as shown in Algorithm 2.

At the end of an iteration, which involves the successive solution of $n+m$ systems of equations, the algorithm switches to the $n+m+1$-th dimension to solve the parameters in $\mathbf{\Phi}$. The matrix

$$\mathbf{U}_\phi = \begin{pmatrix} \prod_{j=1}^{n+m}(1 - \left|f_j\right| + f_j z_j(k)) \\ \vdots \\ \prod_{j=1}^{n+m}(1 - \left|f_j\right| + f_j z_j(k+N)) \end{pmatrix} \tag{4-22}$$

then contains the constants from the fixed parameters of all other $n+m$ dimensions of F and the error

$$E = \|\chi - \mathbf{U}_\Phi \mathbf{\Phi}^{\mathrm{T}}\|_F^2 \tag{4-23}$$

with

$$\chi = \begin{pmatrix} \mathbf{x}^{\mathrm{T}}(k) \\ \mathbf{x}^{\mathrm{T}}(k+1) \\ \vdots \\ \mathbf{x}^{\mathrm{T}}(k+N) \end{pmatrix} \in \mathbb{R}^{N \times n}, \tag{4-24}$$

with $\mathbf{x}^{\mathrm{T}}(k) = (x_1 \ldots, x_n)$ is minimized according to (4-21). The $\|\cdot\|_F$ denotes the Frobenius norm for element-wise least squares: $\|Y\|_F = \sqrt{\sum_{i,j} y_{i,j}^2}$. This is important because the state matrix χ has n columns. The minimization of E is performed by solving n separate estimation problems $\mathbf{\Phi}_{1\ldots n} = \mathbf{U}_\Phi^+ \mathbf{X}_{1\ldots n}$ and is again done with the pseudo inverse $\mathbf{U}_\Phi^+ = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}$, (Groß, 2003).

Algorithm 2 CPN-ALS Algorithm

```
procedure CPN-ALS(Z,R)
  initialize F
  repeat
    for i = 1..., n + m do
      V = F_i
      min_V ||X − UV||
      normalize columns of V
      update Φ, F
    end for
    min_Φ ||χ − U_Φ Φ^T||_F
    return F
  until maximum iterations exhausted or no improvement
```

Remark: For a restriction to non-negative parameters, the optimization problem from Algorithm 2 changes to

$$\min_V \|\mathbf{X} - \mathbf{U}\mathbf{V}\|, \text{ where } v_l \geq 0$$

with $l = 1, 2$ and $\mathbf{V} = (v_1 \;\; v_2)^{\mathrm{T}}$. This can be solved with non-negative least squares, (Lawson and Hanson, 1995), and is available in MATLAB.

Rank-*n* alternating least squares algorithm

The ALS algorithm explained in the previous section for rank-1 MTI models can be extended to any rank-n MTI models in normalized CP representation. Expression (4-12) generally yields a rank-r representation

$$\mathbf{x}(k+1) = \sum_{r=1}^{R} \left((\mathbf{F}_i(1,r) + \mathbf{F}_i(2,r) z_i(k)) \prod_{j=J(1)}^{J(n+m-1)} \left(1 - \left|\mathbf{F}_j(2,r)\right| + \mathbf{F}_j(2,r) z_j(k)\right) \mathbf{\Phi}(:,r) \right), \tag{4-25}$$

where the factor matrices $\mathbf{F}_i$ with $i = 1, \ldots, n+m$ and $\mathbf{F}_j$ with $j \in J = \{1, 2, \ldots, n+m\} \backslash \{i\}$ have the dimensions $\mathbb{R}^{2 \times R}$ and are normalized to the 1-norm and $\mathbf{\Phi} \in \mathbb{R}^{n \times R}$.

The optimization problem is similar to the rank-1 approach by minimizing (4-20), where the vector

$$\mathbf{V} = \begin{pmatrix} \mathbf{F}_i(1,1) \\ \mathbf{F}_i(2,1) \\ \mathbf{F}_i(1,2) \\ \mathbf{F}_i(2,2) \\ \vdots \\ \mathbf{F}_i(1,R) \\ \mathbf{F}_i(2,R) \end{pmatrix} \in \mathbb{R}^{2R \times 1} \tag{4-26}$$

contains all $2 \cdot R$ free parameters of the $i-$th factor matrix. The matrix of fixed constants

$$\mathbf{U} = \begin{pmatrix} \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,1)\right| + \mathbf{F}_j(2,1) z_j(k)) \mathbf{\Phi}(:,1) & \cdots & \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,1)\right| + \mathbf{F}_j(2,1) z_j(k+N)) \mathbf{\Phi}(:,1) \\ \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,1)\right| + \mathbf{F}_j(2,1) z_j(k)) \mathbf{\Phi}(:,1) z_i(k) & \cdots & \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,1)\right| + \mathbf{F}_j(2,1) z_j(k+N)) \mathbf{\Phi}(:,1) z_i(k+N) \\ \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,2)\right| + \mathbf{F}_j(2,2) z_j(k)) \mathbf{\Phi}(:,2) & \cdots & \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,2)\right| + \mathbf{F}_j(2,2) z_j(k+N)) \mathbf{\Phi}(:,2) \\ \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,2)\right| + \mathbf{F}_j(2,2) z_j(k)) \mathbf{\Phi}(:,2) z_i(k) & \cdots & \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,2)\right| + \mathbf{F}_j(2,2) z_j(k+N)) \mathbf{\Phi}(:,2) z_i(k+N) \\ \vdots & \cdots & \vdots \\ \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,R)\right| + \mathbf{F}_j(2,R) z_j(k)) \mathbf{\Phi}(:,R) & \cdots & \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,R)\right| + \mathbf{F}_j(2,R) z_j(k+N)) \mathbf{\Phi}(:,R) \\ \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,R)\right| + \mathbf{F}_j(2,R) z_j(k)) \mathbf{\Phi}(:,R) z_i(k) & \cdots & \prod_{j=J(1)}^{J(n+m-1)} (1 - \left|\mathbf{F}_j(2,R)\right| + \mathbf{F}_j(2,R) z_j(k+N)) \mathbf{\Phi}(:,R) z_i(k+N) \end{pmatrix} \tag{4-27}$$

expands to the dimension $\mathbb{R}^{nN \times 2R}$. The optimization problem from Algorithm 2 can be solved with (4-21).

Example 15. The parameters of a second order multilinear state equation system

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} \mathbf{\Phi}(1)(\mathbf{F}_1(1) + \mathbf{F}_1(2) x_1(k))(\mathbf{F}_2(1) + \mathbf{F}_2(2) x_2(k)) \\ \mathbf{\Phi}(2)(\mathbf{F}_1(1) + \mathbf{F}_1(2) x_1(k))(\mathbf{F}_2(1) + \mathbf{F}_2(2) x_2(k)) \end{pmatrix} \tag{4-28}$$

can be expressed in a rank-1 CP tensor

$$\mathsf{F} = \left[\begin{pmatrix} \mathbf{F}_1(1) \\ \mathbf{F}_1(2) \end{pmatrix}, \begin{pmatrix} \mathbf{F}_2(1) \\ \mathbf{F}_2(2) \end{pmatrix}, \begin{pmatrix} \mathbf{\Phi}(1) \\ \mathbf{\Phi}(2) \end{pmatrix} \right]. \tag{4-29}$$

Using the CPN-ALS algorithm to identify the parameters, the following steps are performed:

1. Initialize $\mathbf{F}_2$ and $\mathbf{\Phi}$ with random numbers
2. Solve for $\mathbf{F}_1$ by rearranging (4-28) to $\mathbf{UV}$ with

$$\mathbf{U} = \mathbf{\Phi}\Big((\mathbf{F}_2(1) + \mathbf{F}_2(2)x_2(k)) \quad x_1(k)(\mathbf{F}_2(1) + \mathbf{F}_2(2)x_2(k))\Big),$$
$$\mathbf{V} = \begin{pmatrix}\mathbf{F}_1(1)\\ \mathbf{F}_1(2)\end{pmatrix}$$

3. Solve system of linear equations for time steps ($k = 1 \dots N$) with (4-20) and update F
4. Solve for $\mathbf{F}_2$ by rearranging (4-28) to $\mathbf{UV}$ with

$$\mathbf{U} = \mathbf{\Phi}\Big((\mathbf{F}_1(1) + \mathbf{F}_1(2)x_1(k)) \quad x_2(k)(\mathbf{F}_1(1) + \mathbf{F}_1(2)x_1(k))\Big),$$
$$\mathbf{V} = \begin{pmatrix}\mathbf{F}_2(1)\\ \mathbf{F}_2(2)\end{pmatrix}$$

5. Solve linear equation system for time steps ($k = 1 \dots N$) with (4-20) and update F
6. Solve for $\mathbf{\Phi}$ by rearranging (4-28) to

$$\mathbf{U}_\Phi = \Big((\mathbf{F}_1(1) + \mathbf{F}_1(2)x_1(k))(\mathbf{F}_2(1) + \mathbf{F}_2(2)x_2(k))\Big)$$

and $\mathbf{\Phi}$ and solve (4-23)

7. Repeat steps 2. to 6. until convergence

4-3-3 Evaluation and comparison of Identification Methods

To compare the standard nonlinear MATLAB optimization algorithm with the new ALS algorithm for parameter identification, which is adapted to the structure of the normalized CPN MTI models, the identification time and the values of the cost function are compared with each other. In order to determine a distribution and an average value of these two variables, both algorithms are executed 100 times again using the data from Example 2. The default values from MATLAB are used as option parameters, as defined in Table 4-3.

The CPN-ALS algorithm is initialized with randomly generated parameters. The standard MATLAB algorithm is initialized with the *Genetic-Algorithm* from the Global Optimization toolbox, (MathWorks, 2023a). Random initialization is not possible here because the algorithm terminates if the cost function is undefined. This happens when the model becomes unstable with the random parameters and exceeds the maximum limit for double-precision values (1.8×10^{308}). The results are shown in Figure 4-7. It can be seen that the cost function and identification time values for the CPN-ALS and for the nonlinear *fmincon* algorithm scatter between minimum and maximum values, due to non-convexity. The CPN-ALS algorithm achieves mean cost values of 0.14, which is half of the mean cost values of the *fmincon*

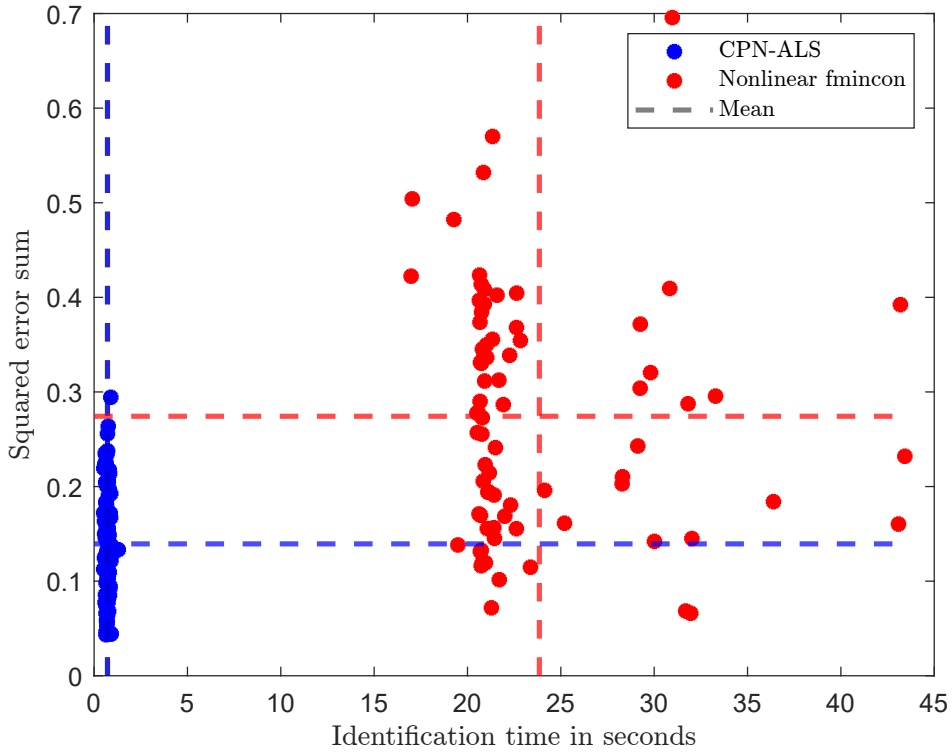

Figure 4-7: Comparison of identification result over identification time

algorithm (0.28). The difference in the identification time is even clearer with a mean identification time of 0.72 seconds for the CPN-ALS algorithm compared to 23.85 seconds for the standard algorithm.

Although the initialization for the CPN-ALS algorithm is done randomly, the cost function distribution in Figure 4-8 shows low variability. Most observations reached a cost function value in the bin of $0.1 < E <= 0.2$ and are therefore only marginally worse than the cost function distribution of the full tensor solution in combination with the `cp_als` algorithm in Figure 4-4. The simulation of the state values with the identified model from the CPN-ALS algorithm is shown in Figure 4-9 and follows the original data very well. The CPN-ALS algorithm is implemented in the MTI-Toolbox for MATLAB, see Lichtenberg et al. (2024).

Based on the promising results of investigation and evaluation of the new CPN-ALS algorithm in this chapter, it is considered useful for identifying normalized CPN MTI models and is therefore used later for the anomaly detection in building systems with real data.

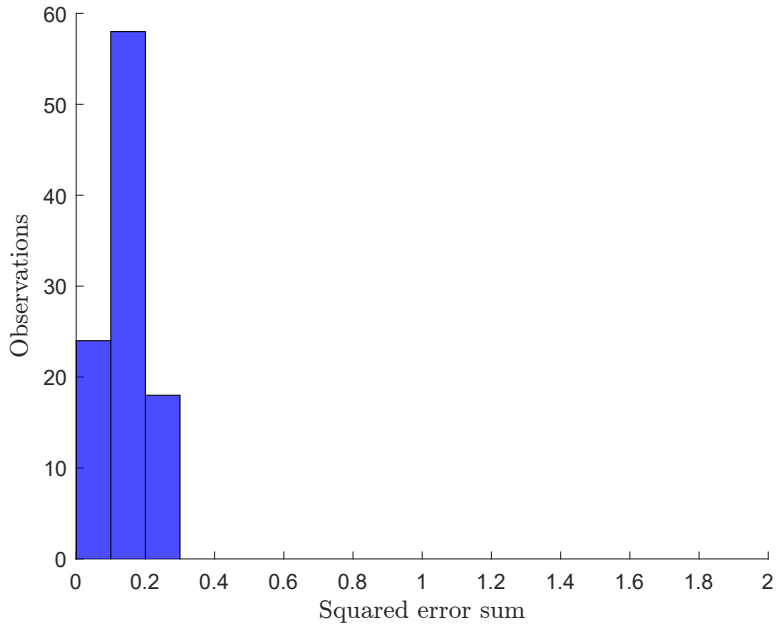

Figure 4-8: Cost function distribution CPN-ALS identification

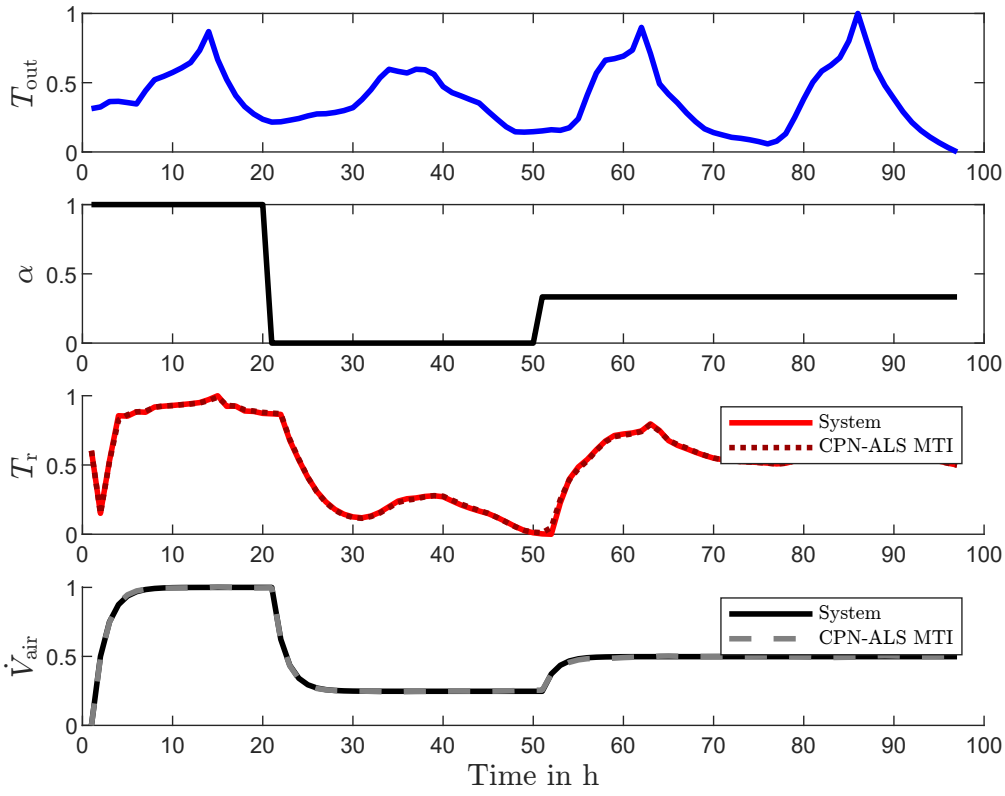

Figure 4-9: Identification result for CPN-ALS identification

Chapter 5

Anomaly detection

As described earlier, the parameters of the normalized multilinear time-invariant (MTI) models are intended to be used as normalized CP decomposed (CPN) parameters for the automatic detection of anomalies in building operation. The identification of these parameters from building data was developed in the previous chapter. In the following, a method is developed and evaluated to detect time intervals in which building behavior deviates from nominal operation, based on the identified parameters. For this purpose, residuals are first generated, which are then used in a second evaluation step to produce the output of the anomaly detection algorithm.

5-1 Residual generation

5-1-1 Residuals from CPN Parameters

The detection of anomalies should be based on the generation and evaluation of residuals. To achieve this, residuals must first be computed over the entire time series of building measurements under analysis. Residuals are generally defined as deviations from expected values. In this context, the residuals are derived directly from the CPN parameters of the identified MTI. To this end, two methods for generating residuals from CPN parameters are developed below.

Sequential residual generation

In sequential residual generation, the CPN factors of the parameter tensor $\hat{\mathsf{F}}(k)$ of an MTI model are generated for each sequence

$$(k), (k+1), \ldots, (k+N) \tag{5-1}$$

with N input and state measurements of the time interval

$$(k-N),(k-N+1),\ldots,(k-1). \tag{5-2}$$

The algorithm then waits until the next sequence of measurements contains N time steps of new data before identifying the parameter tensor $\hat{\mathsf{F}}(k+1)$ at $k = N, 2N, 3N, \ldots$. The sequence length N must be sufficiently large to allow for the identification of a total of $r(n+m)$ parameters.

This length is therefore dependent on the selected identification algorithm. In order to guarantee an overdetermination of the equation system, it must comprise at least $> 2^{n+m}$ time steps for the full tensor identification, while theoretically only at least $r(n+m)$ time steps are necessary for the alternating least squares (ALS) algorithm. However, it should be noted that the dynamics of the system must be reflected in the chosen sequence in order to be able to identify unique parameters that generate meaningful residuals. For noisy data, it is recommended to choose $N \gg r(n+m)$.

Each parameter set identified in this way represents the current system behavior of the single sequence of data used. Residuals are generated by comparing these different parameter sets obtained from the different sequences. With sequential residual generation, a new residual can only be generated, and a diagnostic decision about the system behavior can only be made if enough new data is available for the entire length of the new sequence. In an online or real-time application, this means that an anomaly, even if it occurs right at the beginning of the new sequence, can only be detected after N new data points have been collected. In order to be able to make a diagnostic decision about the current system behavior at any point in time, the generation of moving horizon residuals is described below.

Moving horizon residual generation

The moving horizon residual generation for anomaly detection in building systems is developed in Schnelle et al. (2022). The approach takes up the idea of moving horizon estimation (MHE), as used in state estimators, e.g. in control engineering Ganesh et al. (2021) and anomaly detection in buildings Awawdeh et al. (2024). For this purpose, a window is placed over the measured values, which has a fixed number of time steps and moves forward by one time step after each new time step to obtain the next CPN factors of MTI parameter tensors

$$\hat{\mathsf{F}}(k), \hat{\mathsf{F}}(k+1), \ldots, \hat{\mathsf{F}}(k+N)\ \forall k \in \mathbb{R}^{+} \tag{5-3}$$

of the state and input measurements of time steps from (5-2). With each new time step, the optimization problem is solved again to determine the current model parameters. The moving time horizon $N \gg n+m$ should again be chosen large enough to identify meaningful parameters. In order to use the identified parameters for anomaly detection, the difference between the parameter tensor F of the nominal system behavior and the parameter tensor $\hat{\mathsf{F}}$ of the currently identified system behavior at each time step gives a residual. Solving a new optimization problem at each new time step incurs significant computational effort. Furthermore, a large number of residuals is generated and must be evaluated. Since the focus of this work is on reducing the number of parameters for efficient computation and memory savings, a restriction to low-rank models for residual generation is investigated next.

5-1-2 Low-rank limitation

The uniqueness of the CPN parameters has been discussed in terms of rank-1 MTI models and is therefore also given for each of the n state equations of the diagonal rank-n CPN MTI model of Definition 3-1.3. The exact representation of most MTI models is not low-rank. Nevertheless, rank-1 MTI models may be able to approximate the main dynamics of building systems with sufficient accuracy to use their rank-1 parameters for anomaly detection. Therefore, the advantage of small model size and the important property of uniqueness for anomaly detection can be used here. To demonstrate how the dynamics of energy flow systems, such as those found in buildings, can be approximated by a rank-1 MTI model, the example from Example 2 is used. As shown in Example 5, the exact representation in the discretized representation is rank-6. To identify rank-1 CPN parameters for the two state equations from simulation data, Algorithm 2 is used. In the diagonal rank-n MTI representation from Definition 3-1.3, the state-space model is expressed with

$$\begin{pmatrix} x_{T_r}(k+1) \\ x_{\dot{V}_{air}}(k+1) \end{pmatrix} = \begin{pmatrix} \phi_1 & 0 \\ 0 & \phi_2 \end{pmatrix} \begin{pmatrix} \prod_{i=1}^{n+m}(1 - |\mathbf{H}_{i,1}| + \mathbf{H}_{i,1}\mathbf{z}_i) \\ \prod_{i=1}^{n+m}(1 - |\mathbf{H}_{i,2}| + \mathbf{H}_{i,2}\mathbf{z}_i) \end{pmatrix} \tag{5-4}$$

with the vector of states and inputs $\mathbf{z} = \begin{pmatrix} T_{room} & \dot{V}_{air} & T_{out} & \alpha \end{pmatrix}^{\mathrm{T}}$ and the parameter matrix

$$\mathbf{H}^{\mathrm{T}} = \begin{pmatrix} h_{T_{room},1} & h_{\dot{V}_{air},1} & h_{T_{out},1} & h_{\alpha,1} \\ h_{T_{room},2} & h_{\dot{V}_{air},2} & h_{T_{out},2} & h_{\alpha,2} \end{pmatrix}. \tag{5-5}$$

The parameters are identified using the simulation data of the discretized and scaled rank-6 model from (12). The result of the rank-1 approximation for the first state, T_r, is shown in Figure 5-1. The main dynamics of the system are approximated by the model, but the error between the simulated state and the measurement is clearly higher than in the rank-6 approximation of Example 5.

To evaluate whether the rank-1 parameters can reflect differences in dynamics despite the visible error between the model and the data, a parameter in the model equation from Example 2 that affects the dynamics of the system is altered. Changing the heat transfer coefficient U, affects the losses through the wall. The value of the heat transfer coefficient is changed from the original $U = 0.2$ to $U \in [0.2, 0.4, 0.6, 0.8]$ in four successive sequences. For each sequence, the CPN parameters of the diagonal rank-1 model are identified in norm-1 representation using the ALS Algorithm 2. To avoid random identification results, which can occur due to random initialization and due to hitting local minima in the non-convex optimization problems, each identification of each sequence is performed 50 times. As an example, the distribution of the parameter $h_{T_{out}}$ of the first state equation for $x_{T_{room}}$ is shown in Figure 5-2.

It can be seen that with the rank-1 constraint, the distribution of the parameters is barely scattered, despite the non-convex optimization problem of CP decomposed MTI models. The parameter is determined 45 times within a bin of width 0.05. In a few cases, the parameters are randomly distributed over values between 0.15 and 0.6. Furthermore, it can be clearly seen that the parameter values vary depending on the heat transfer coefficient. The higher the heat transfer coefficient, the larger the parameter associated with the input of the external temperature signal. This results in the interpretability of the parameter values derived from

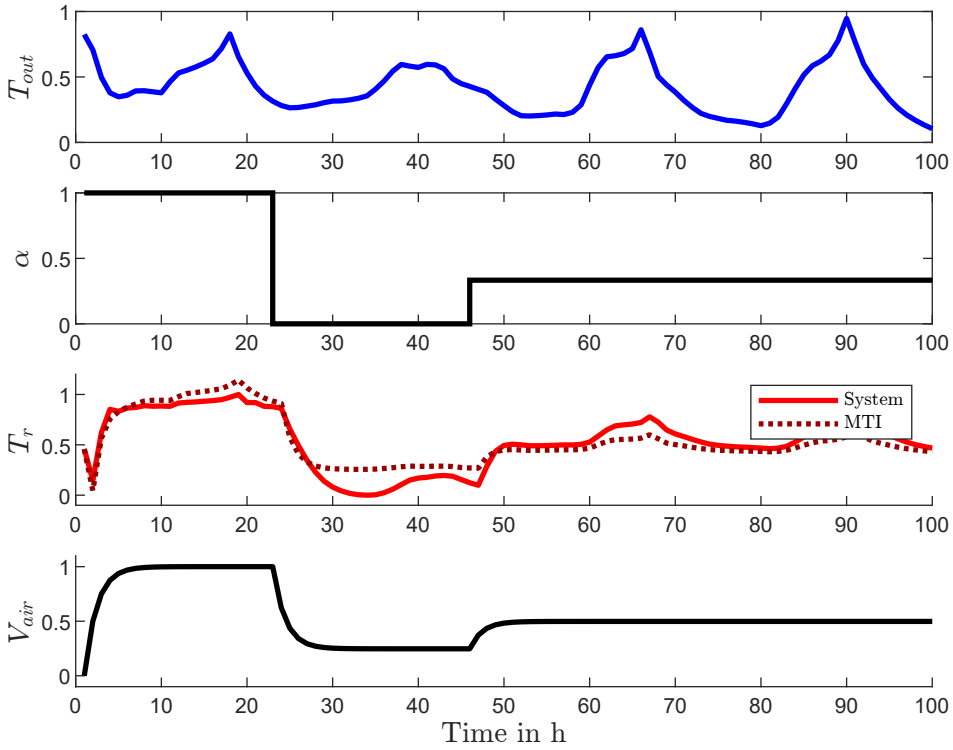

Figure 5-1: Rank-1 approximation of room example

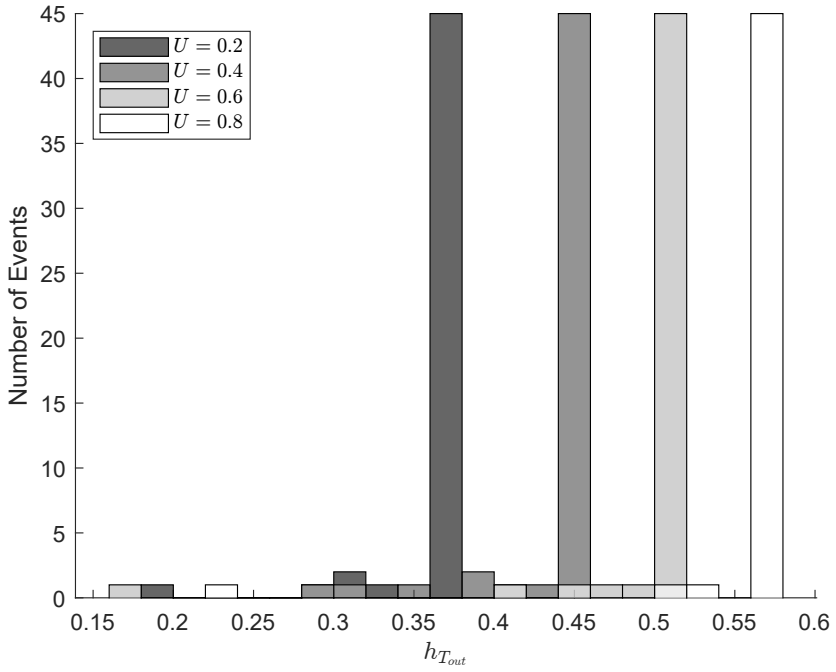

Figure 5-2: Rank-1 parameter $h_{T_{out}}$ histogram for heat transmission coefficient U

the model identification. The heat transfer coefficient affects the losses through the exterior

wall, which of course also depend on the outside temperature. If the heat transfer coefficient is higher, the thermal permeability of the wall is also higher and the external temperature has a greater influence on the internal temperature than at low heat transfer coefficients. This result again demonstrate the interpretability of the parameters, which play an important role in anomaly detection. If the influence of a parameter on the state changes noticeably from one sequence to the next, there is an anomaly, and the changing parameter provides an indication of the part of the system where the anomaly has an effect.

In Figure 5-3 it is shown how the changed heat transfer coefficient affects the other rank-1 parameters of the first state equation. The mean values of the 50-times parameter identification of the simulation data sequences with heat transmission coefficients between $U = 0$ and $U = 1$ are shown. It should be emphasized that the slope of the parameter $h_{T_{out}}$ already shown is steepest as the heat transmission coefficient increases. The parameter h_{T_r}, which belongs to the room temperature state, and the parameter h_α, which belongs to the control signal of the supply air, do not change. The influence of the supply air flow, which is expressed by the parameter $h_{\dot{V}_{air}}$, increases slightly.

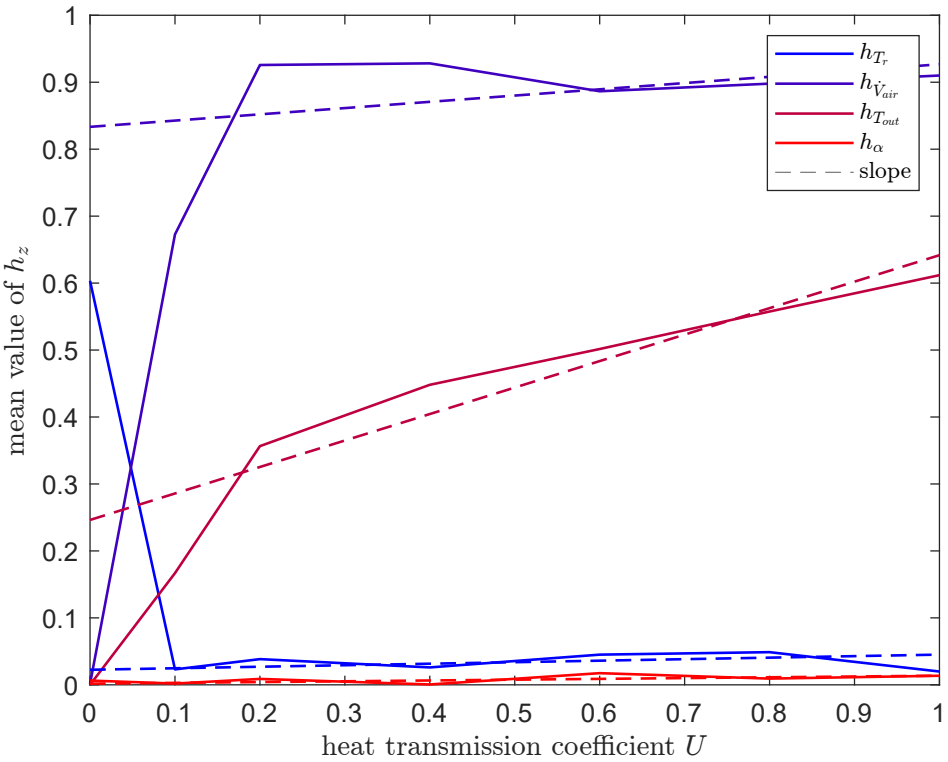

Figure 5-3: Influence of heat transmission coefficient U on rank-1 parameter identification

5-1-3 Robustness of low-rank residuals

In this section, we will examine how robust the residual generation by the method presented in subsubsection 5-1-1 and subsubsection 5-1-1 is against white noise. For this purpose, the data of the example model from Example 2 is used in discrete norm-1 CPN representation from Example 12 to generate multiple sequences of state data. The signal-to-noise ratio (S/N)

is defined as the ratio between the signal to the noise in Decibel and can be used as a metric for noisy signals. The first data sequence remains unchanged, while a white noise with a S/N = 40 dB to S/N = 18 dB of the noiseless state signals is applied to the noisy sequences. The parameters of each sequence are then estimated using sequential parameter identification of the diagonal rank-1 CPN model from (5-4). To receive a distribution of results, the parameter identification is again performed 50 times for each sequence.

Figure 5-4 shows histograms of the parameter identification results for the noise-free and noisy signals, illustrating how the distribution of the 50 identification runs changes with the signal-to-noise ratio.

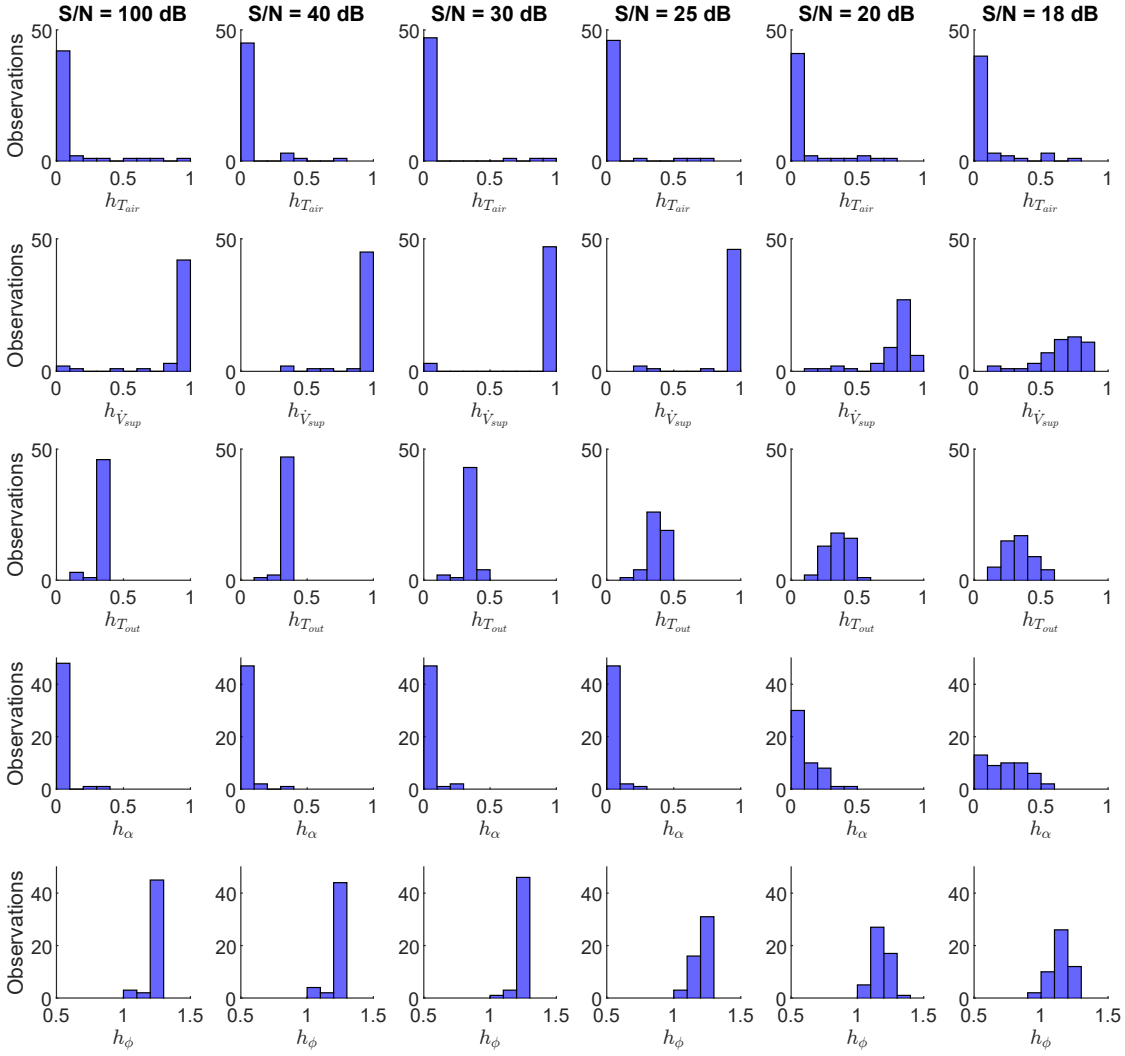

Figure 5-4: Rank-n parameter estimation depending on noise

It can be seen that noise up to a signal-to-noise ratio S/N = 30 dB has no effect on the results of the parameter values. In addition, the noise is barely noticeable in the location of the parameter distribution. At S/N = 25 dB, the noise is visible in the parameter distribution of $h_{T_{out}}$ and h_{ϕ}, but the majority of all identified parameters are still in the same bin as without

noise. The scatter increases at a S/N = 20 dB for the other parameters as well, although almost all observations still lie within an interval of 0.3. At S/N = 18 dB, the frequency distributions flatten out. To provide intuition about the magnitude of noise at S/N = 100dB and S/N = 20dB, the noisy state signals are depicted in Figure 5-5. The findings of this section

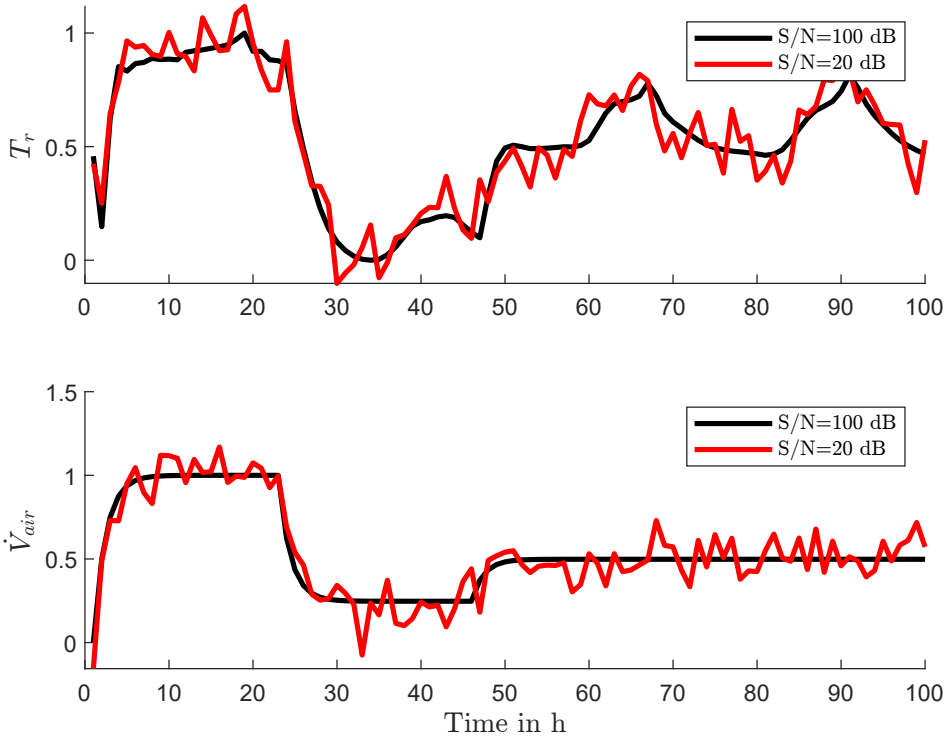

Figure 5-5: State signals in case of noise

on the generation of residuals using the CPN MTI parameters can be summarized as follows:

- The dynamics of simple multilinear systems seem to be sufficiently represented in diagonal rank-n approximation to use their parameters for anomaly detection.
- A change of a parameter in the exact state equation confirms the idea of interpretability, since the influence of the outside temperature on the room temperature can be read directly from the corresponding parameter.
- Identification of CPN parameters remains robust against white noise in state measurements. Noticeable effects only occur in the presence of very noisy signals.

The results from the investigations in this section confirm the idea that the low-rank CPN MTI parameters are suitable for the use in anomaly detection. To evaluate residuals from the actual parameters an evaluation approach, using classification algorithms, is described next.

5-2 Residual evaluation

In order to evaluate the classification result, a confusion matrix is introduced, which contains with

- True positive anomalies (TP)
- False negative anomalies (FN)
- True negative nominals (TN)
- False positive nominals (FP)

the four states. The overall accuracy is given with

$$PA = 100\frac{TP + TN}{TP + FN + TN + FP}, \tag{5-6}$$

in Starovoitov and Golub (2020)

The identification of parameters for residual generation in section 5-1 produces multidimensional clouds of CPN parameter sets from MTI models, which approximate the dynamic behavior of the system at different points in time. The residuals are derived by evaluating the values of various parameter sets within these clouds. To interpret the large number of parameters, classification methods can be applied. Binary classifiers, such as Least Squares Support Vector Machines (LS-SVM) from Suykens et al. (2002) and two-class k-means clustering from Hartigan and Wong (1979), introduced in subsubsection 2-1-2, are particularly suitable for distinguishing between normal and abnormal system behavior.

The LS-SVM classifier requires supervised training data and typically assigns a class label y to each parameter set $\mathbf{x}$. The classification depends on the choice of a kernel function. Various kernels, such as linear, polynomial, radial-basis-functions (rbf) and others, can be used. The appropriate kernel function depends on the data. The radial basis function (rbf) kernels used in this study, in combination with the LS-SVM from (2-31) have two free parameters, σ^2 and γ, which influence the classification results. These parameters can function as tuning parameters to adjust the sensitivity of the classification model. In the following analysis, the influence of these two tuning parameters on the anomaly detection result is investigated using an example data set containing nominal and anomalous data. The data is provided by the University of Applied Sciences in Augsburg and was generated inside the SONDE Project, (Schnelle and Lichtenberg, 2024). The data set contains simulation data of a seminar room over one year. The simulated anomaly is an open window of different intensity and duration, which occurs several times over the year.

The data set contains two state variables: room temperature and radiator return temperature. Additionally, it includes three input data signals: radiator supply temperature, radiator flow rate, and outdoor temperature. Each signal is sampled with a 15-minute sampling interval. The data is used to identify the CPN parameter of a 2-norm normalized rank-1 MTI from Definition 3-1.2 using the parameter identification procedure described in section 4-3, with the nonlinear parameter identification algorithm from subsection 4-3-1. To obtain parameter sets for each day and each night of the year, the sequential parameter identification as outlined

in subsubsection 5-1-1 is used with a horizon N of 12 hours. From the identified parameter sets, four intervals, evenly distributed over the year, are selected for training. The training data set covers 10 % of the test data set for the year as marked by the blue shadowed parts in Figure 5-7. The LS-SVM classifier was trained using the training data set by mapping the parameter sets $\boldsymbol{\theta}$ of the training period with a class value y (minus one for anomalous behavior or plus one for nominal behavior). The trained LS-SVM model was then fed with the parameter sets for the entire year and generated a classification result for each day and each night of the year.

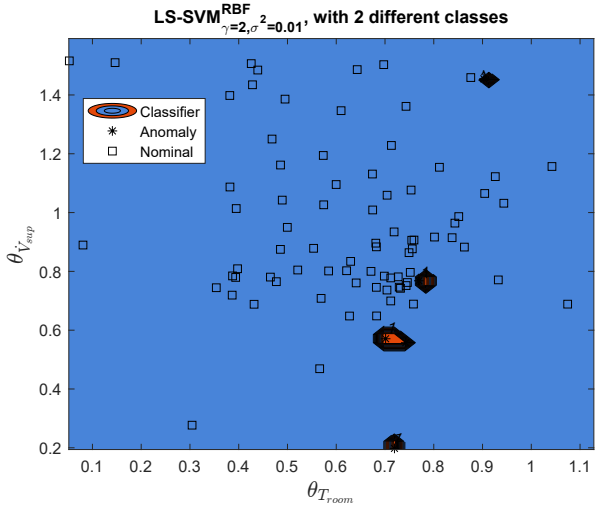

Figure 5-6: SVM classification $\gamma = 2$ and $\sigma^2 = 0.01$

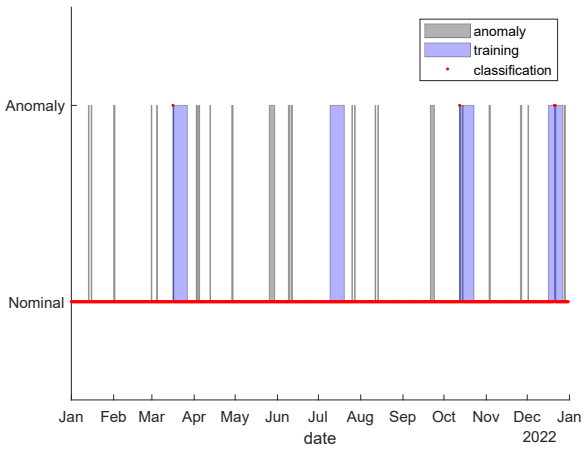

Figure 5-7: Classification result $\gamma = 2$ and $\sigma^2 = 0.01$

Table 5-1: Confusion matrix $\sigma^2 = 0.01$ and $\gamma = 1$

		Classified state	
		Positive (P)	Negative (N)
Real state	Positive (P)	5/46 (TP)	41/46 (FN)
	Negative (N)	0/683 (FP)	683/683 (TN)

Three different combinations of the parameter pairs σ^2 and γ were implemented to investigate the effects of these two parameters on the classification and anomaly detection result. In the first case, the classification and the anomaly detection are investigated for values of $\sigma^2 = 0.01$ and $\gamma = 2$. The classification result of the training data is shown in Figure 5-6 in a two-dimensional projection of the five-dimensional parameter space. This parameter space includes the parameters related to the room temperature (the first state) and the heating volume flow (an input to the system). The two-dimensional representation cannot accurately represent the exact position of the class boundaries within the parameter space, but

provides insight into the effect of the two tuning parameters. It can be observed that the very small value for σ^2, in combination with an average value for γ, leads to the formation of a narrow class boundary around each individual training point labeled as anomalous. The high weighting of the correct classification of each training point is determined by the low value of σ^2. The corresponding anomaly detection result for the entire year shows the effects of these narrow class boundaries: only the anomalies within the training set are recognized as such. Presenting the results in a confusion matrix supports the evaluation. This provides information about the four possible categorizations of a classification result. These include the true positive (TP) and true negative (TN) events for correctly classified anomalies and nominals as well as the false positive (FP) for nominals incorrectly classified as anomalies and the false negative (FN) for anomalies incorrectly classified as nominals. The first pair of σ^2 and γ values results in the confusion matrix from Table 5-1, which reports five true positives, 41 false negatives, zero false positives and 683 true negatives. This outcome indicates overfitting of the SVM to the training set, resulting in poor performance when applied to new data.

In the second case, where $\sigma^2 = 0.1$ and $\gamma = 20$, the strong influence of each training point overlaps with the effect of the large γ, which affects the smoothness of the class boundaries. The result shown in Figure 5-8, is again a region of the anomaly class around the anomalous training points, but the boundaries are less narrow than in the first case. As a result of this classification, it can be seen in Figure 5-9 that, although more anomalies over the entire year were classified correctly (18, see Table 5-2), there are also many false alarms (40) because nominal events were misclassified as anomalies.

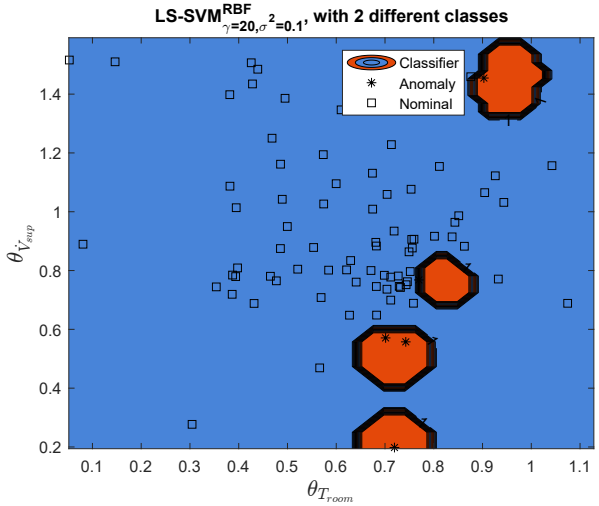

Figure 5-8: SVM classification $\gamma = 20$ and $\sigma^2 = 0.1$

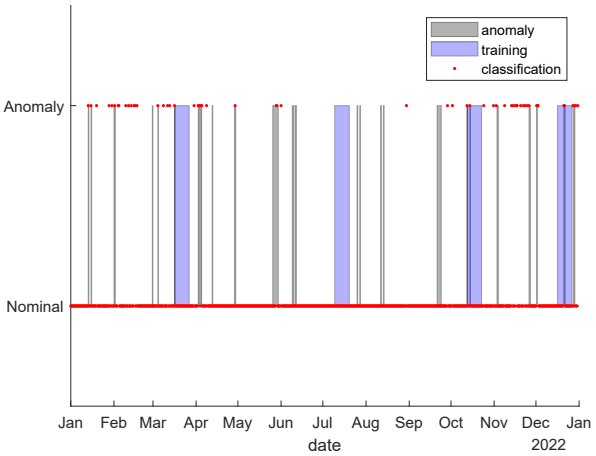

Figure 5-9: Classification result $\gamma = 20$ and $\sigma^2 = 0.1$

The third pair of values, $\sigma^2 = 5$ and $\gamma = 1.5$, shows a different pattern for the two-dimensional classification in Figure 5-10. The weighting of the distance of each individual training parameter to the support vector of the correct class is small, due to the larger value of σ^2. The result of this classification is, that the LS-SVM is not fitted to all training points. Some training parameters for anomalous behavior lie further out and in between other nominal parameters and are therefore evaluated as outliers. The anomaly class appears in the lower-right

Table 5-2: Confusion matrix $\sigma^2 = 0.1$ and $\gamma = 20$

		Classified state	
		Positive (P)	Negative (N)
Real state	Positive (P)	18/46 (TP)	28/46 (FN)
	Negative (N)	40/683 (FP)	643/683 (TN)

region and does not include all anomalous training parameters. The result generated by the trained LS-SVM using the full-year data is illustrated in Figure 5-11. While some anomalies were detected, not all training parameters within the training periods are classified correctly. This is a result of underfitting. There are four correctly classified anomalies over the year, see Table 5-3 and three nominals are misclassified.

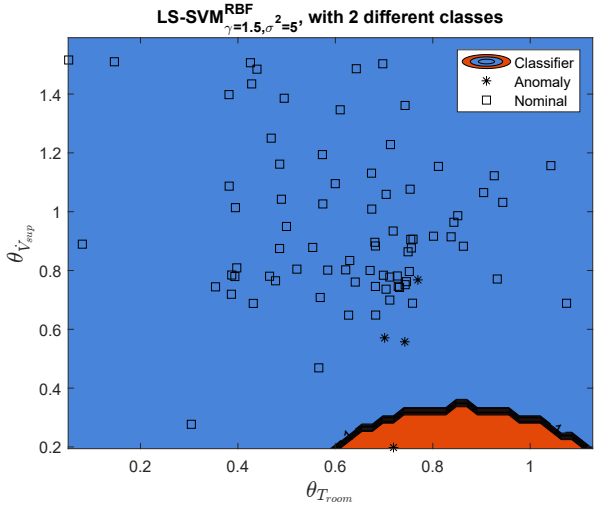

Figure 5-10: SVM classification $\gamma = 5$ and $\sigma^2 = 1.5$

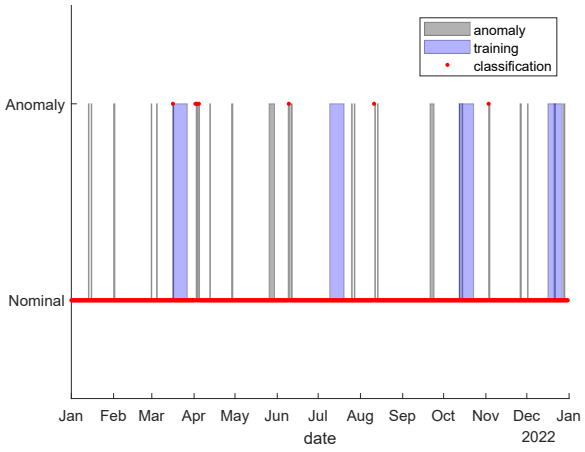

Figure 5-11: Classification result $\gamma = 5$ and $\sigma^2 = 1.5$

Table 5-3: Confusion matrix $\sigma^2 = 5$ and $\gamma = 1.5$

		Classified state	
		Positive (P)	Negative (N)
Real state	Positive (P)	4/46 (TP)	42/46 (FN)
	Negative (N)	3/683 (FP)	680/683 (TN)

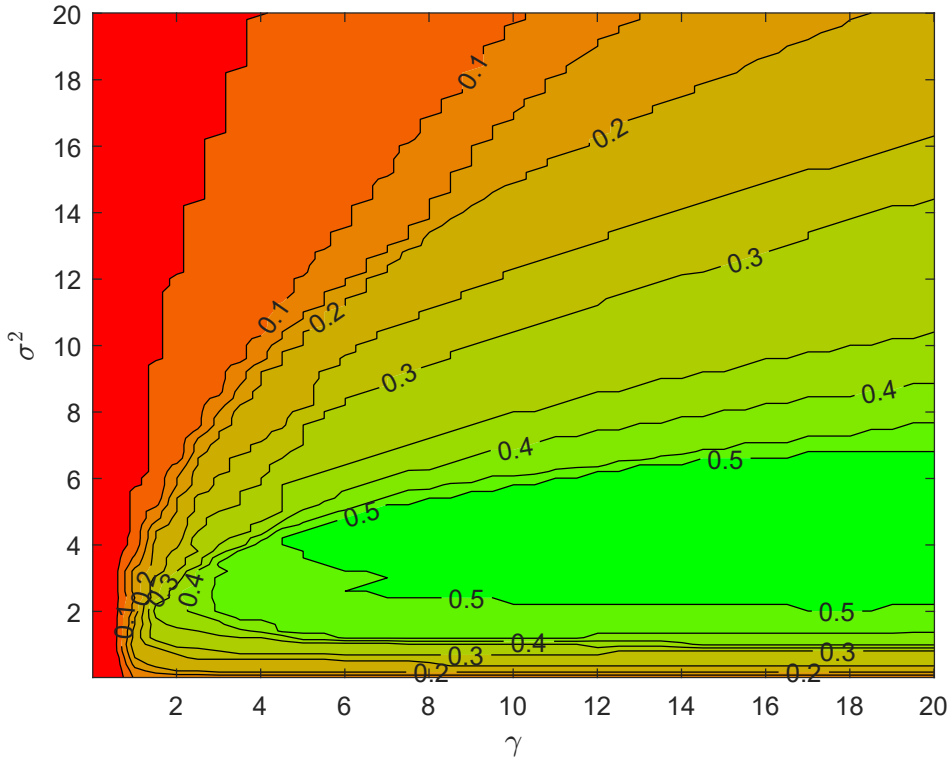

Figure 5-12: True Positive (γ, σ^2)

The three pairs of example values show the influence of the tuning parameters in the LS-SVM. It is evident that σ^2 has an inversely proportional effect on the weighting of the distances of the training points to the support vectors, while increasing γ values increases the smoothness of the class boundaries. The combination of the two free parameters is therefore a trade-off between smoothness and accuracy, leading to an anomaly detection result with a trade-off between false alarms and undetected anomalies.

To thoroughly examine the impact of the two tuning parameters, a study was conducted across wider ranges $0.01 \leq \sigma^2 \leq 20$ and $0.1 \leq \gamma \leq 20$. The results are visualized in four diagrams (Figure 5-12, Figure 5-13, Figure 5-14, Figure 5-15), depicting the percentages of the four states from the confusion matrices as a result of the five-dimensional classification problem.

The analysis reveals two primary trends. The first can be observed in the range of small values [$\sigma^2 < 4$, $\gamma < 5$]. The rate of correctly detected anomalies (true positive rate) increases along the diagonal as both γ and σ^2 increase, and remains high for further increases in γ. In a similar pattern, the misclassified anomalies, known as false negatives, decrease. The second trend shows that as γ and σ^2 decrease, the rate of correctly detected nominal values (true negative rate) improves in a similar pattern to the decrease of misclassified nominals (false positive rate). The optimal classification aims to maximize both true positive and true negative events. However, these variables are inversely related to each other, which represents a conflict of objectives. Enhancing anomaly detection (true positives) leads to more false alarms (false positives), while reducing false alarms reduces anomaly detection performance.

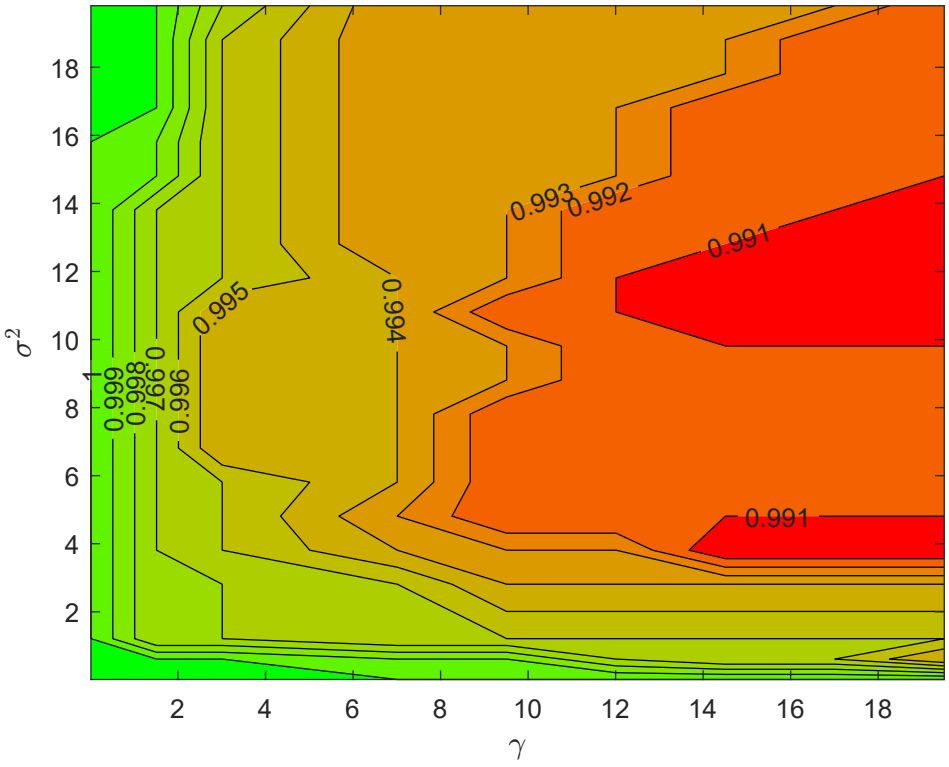

Figure 5-13: True Negative (γ, σ^2)

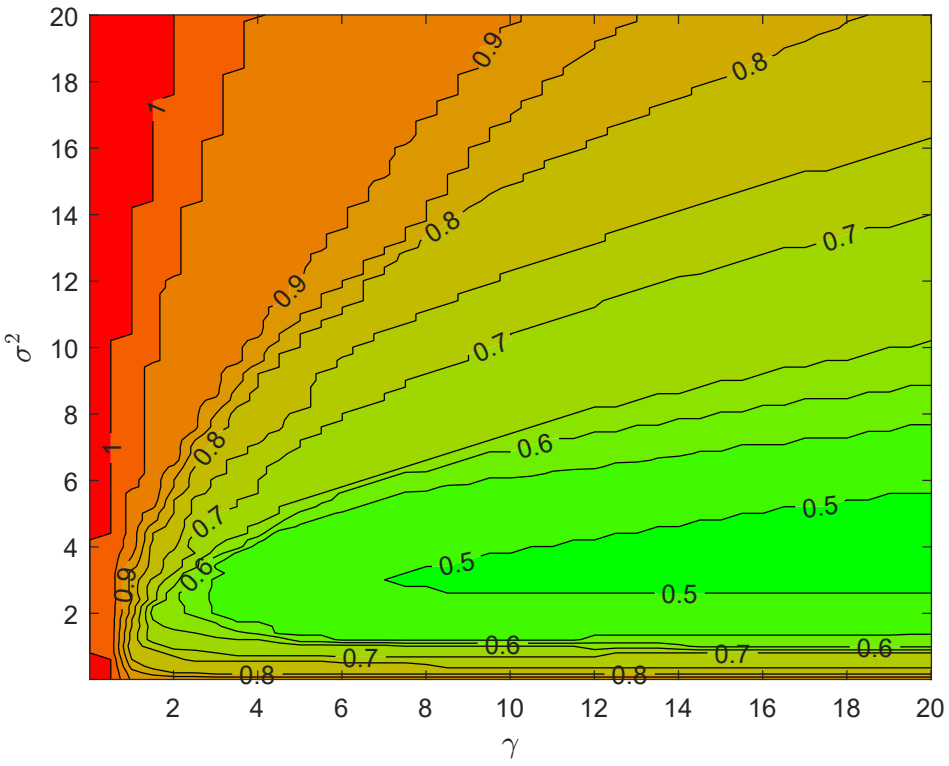

Figure 5-14: False Negative (γ, σ^2)

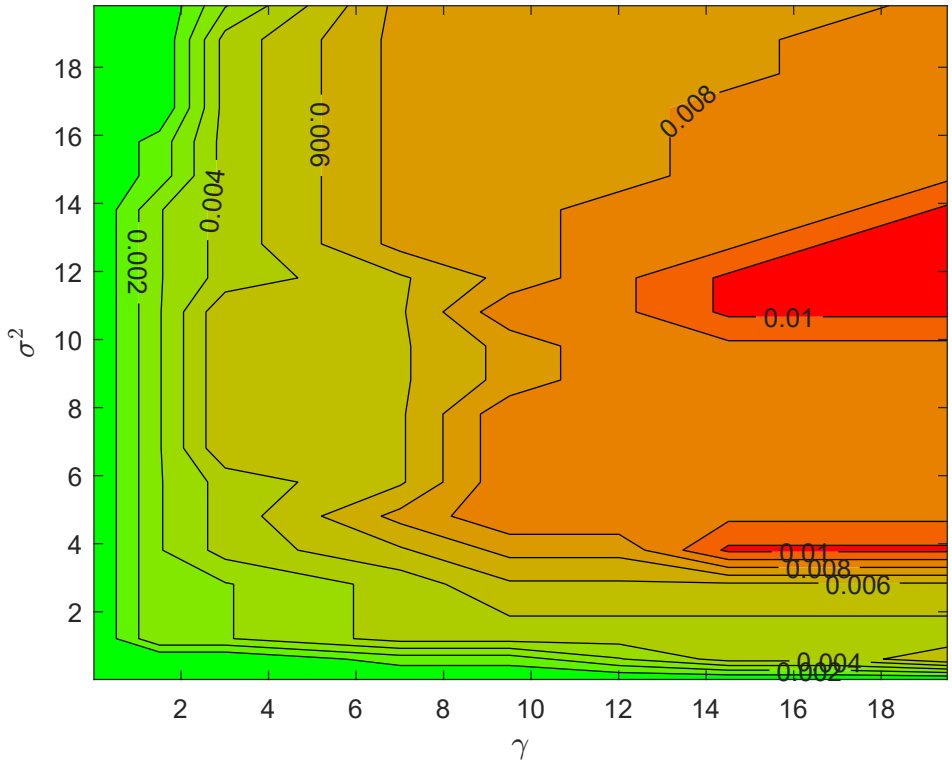

Figure 5-15: False Positive (γ, σ^2)

The optimal overall classification accuracy lies at an intermediate point, balancing these competing factors. This study identifies the optimal values as $\gamma = 5$ and $\sigma^2 = 2.8$, representing a compromise between effective anomaly detection and minimizing false alarms. The classification model achieves its best performance with 23 correctly identified anomalies and 681 correctly classified nominal events, as detailed in Table 5-4. According to the accuracy presented in (5-6), the total number of correctly classified events is 704, resulting in an overall accuracy of 96.6 %. This high accuracy rate reflects the model's robustness in distinguishing between anomalous and nominal events, with only a small fraction of misclassifications in both categories.

Table 5-4: Confusion matrix $\sigma^2 = 2.8$ and $\gamma = 5$

		Classified state	
		Positive (P)	Negative (N)
Real state	Positive (P)	23/46 (TP)	23/46 (FN)
	Negative (N)	2/683 (FP)	681/683 (TN)

The residuals generated as described in section 5-1 must be reliable in order to decide

whether a current system behavior is nominal or anomalous and whether an alarm should be triggered or not. This can only be the case if the previous identification problem was able to determine unique parameters of the normalized MTI. The actual impact of errors in building data and whether these can be detected using the described method through residuals of the parameters is evaluated in the following chapter using representative errors in building data.

Chapter 6

Application

In this chapter the developed automated anomaly detection algorithm is evaluated by applying it to three different types of data. The main difference between the data are the distinction between supervised and unsupervised data sets. The latter poses the difficulty that there is no information available on whether the system was in nominal operation or whether anomalies occurred. Hence, there is no training data for the classification model. In the first section of this chapter, supervised data sets containing nominal data and a single type of anomaly are used. The second section uses standard data sets available from various research institutes that contain multiple supervised failures. Finally, it is discussed whether the algorithm evaluated in this way could also be applied to unsupervised data. This question is examined using real measurement data from a large office building.

6-1 Application to supervised data sets - single anomaly

In this section, the developed parameter identification and anomaly detection algorithm is tested with two different sets of building monitoring data. Both data sets consist of supervised data with a single type of anomaly. The first data set contains real measurement data from a test office building, while the second data set contains simulation data of a digital twin of a seminar room.

6-1-1 Test office building

The following data set contains real measured data of a test office room in a laboratory that is used for different test cases for operation and monitoring tests in buildings and for multilinear time-invariant (MTI) modeling and anomaly detection in Schnelle et al. (2022). This is described in this section. The room is equipped with a heating and cooling unit implemented as heating and cooling panels, a ventilation system and internal loads representing technical equipment and people. The people are represented by dummies with internal heating units and the possibility of autonomous movements. The test office building is shown in

Figure 6-1: Test office room as demonstrator, (Schnelle et al., 2022)

Figure 6-1. Sensors are installed in and around the test office space to collect data such as air temperatures, water temperatures, volume flow signals, and occupancy sensor data to identify dummy movement. Control signals and set points are also collected.

Five monitored data points are used to estimate a black-box model of the nominal system behavior, including the supply temperature T_{sup} and control signal $\dot{V}_{air}$ of the heating panels and the volume flow of the ventilation system C_{sup}. These are used as inputs for the subsequent model building procedure. It is assumed that the states are measurable and consist of ΔT_{room}, which is the difference between the inside temperature and ambient temperature around the room as well as the return temperature of the heating panel, T_{ret}. A second order MTI model is built as the block diagram in Figure 6-2 shows.

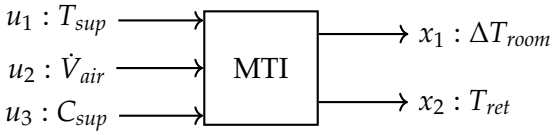

Figure 6-2: Block diagram of the test office room, (Schnelle et al., 2022)

The selection of the data points thus leads to a model with $n = 2$ states and $m = 3$ inputs, which is modeled with (3-16). The monomial tensor

$$\mathsf{M}(\mathbf{x}, \mathbf{u}) = \left[\begin{pmatrix} 1 \\ \Delta T_{room} \end{pmatrix}, \begin{pmatrix} 1 \\ T_{ret} \end{pmatrix}, \begin{pmatrix} 1 \\ C_{sup} \end{pmatrix}, \begin{pmatrix} 1 \\ \dot{V}_{air} \end{pmatrix}, \begin{pmatrix} 1 \\ T_{sup} \end{pmatrix}\right] \in \mathbb{R}^{\times^5 2}$$

consists of the variables while the transition tensor $\mathsf{F} \in \mathbb{R}^{\times^6 2}$ merges all parameters. In Euclidean normalized rank $r = 1$ decomposition, the normalized CP decomposed (CPN) representation has $(2n + m)r = 7$ parameters, therein the five angles θ_i and two weighting values of $\widetilde{\mathbf{F}}_\phi$.

To prevent numerical problems, all data are normalized within the interval $[0, 1]$, which

additionally allows the choice of the identity matrix as the weight $\mathbf{Q}$ for the cost function evaluation (4-2) in the optimization problem (4-3) for parameter identification.

The second step is to identify a nominal rank-1 MTI model from Definition 3-1.2 from data using Problem 4-1.1 and the nonlinear standard optimization algorithm, described in subsection 4-3-1. The comparison of the measured and simulated state data in Figure 6-3 shows that the identified model is able to represent the behavior of the room.

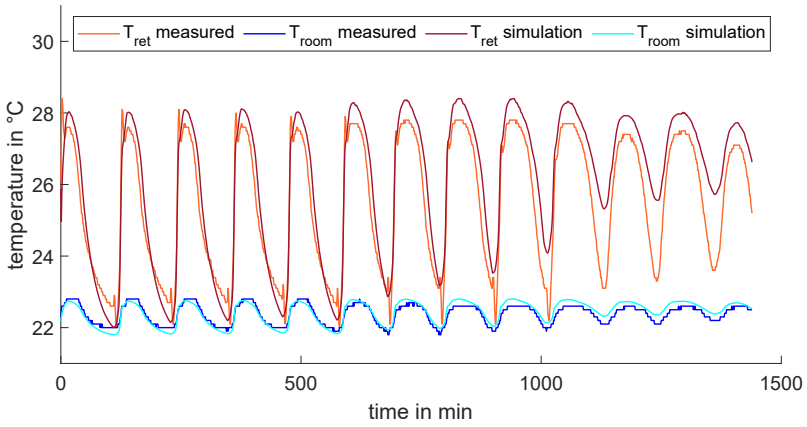

Figure 6-3: Measured and simulated states of the test room, (Schnelle et al., 2022)

To test the anomaly detection algorithm with this real data, an open door allowing cold air to enter from the outside was implemented as an anomaly. The resulting data set contains 600 minutes of data sampled at one-minute intervals, as shown in Figure 6-4.

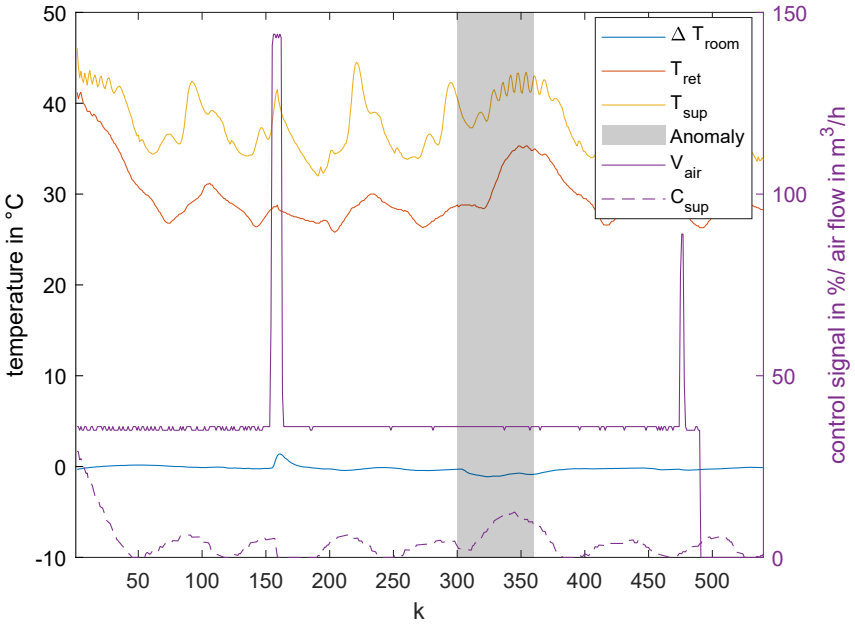

Figure 6-4: Input and state measurement of test office, (Schnelle et al., 2022)

The gray shaded interval $k = 300, \ldots, 360$ is where the anomaly occurred.

With a moving horizon of 30 minutes, current parameter tensors $\widetilde{\mathsf{F}}$ in the form of the normalized angles θ_i and the n weighting elements λ_i of $\widetilde{\mathbf{F}}_\Phi$ of an n-diagonal MTI model from Definition 3-1.3 are estimated using the method proposed in subsubsection 5-1-1.

The subsequent residual evaluation is performed with a Least Squares Support Vector Machines (LS-SVM) with radial basis function (rbf) kernel of the *LS-SVMlab* Matlab toolbox by De Brabanter et al. (2010). The classifier tries to find a hyperplane to separate the nominal parameters from the parameters of the anomalous behavior. For this purpose, the parameter angles $\theta_i, 1$, which describe the first state equation are entered into the classification model.

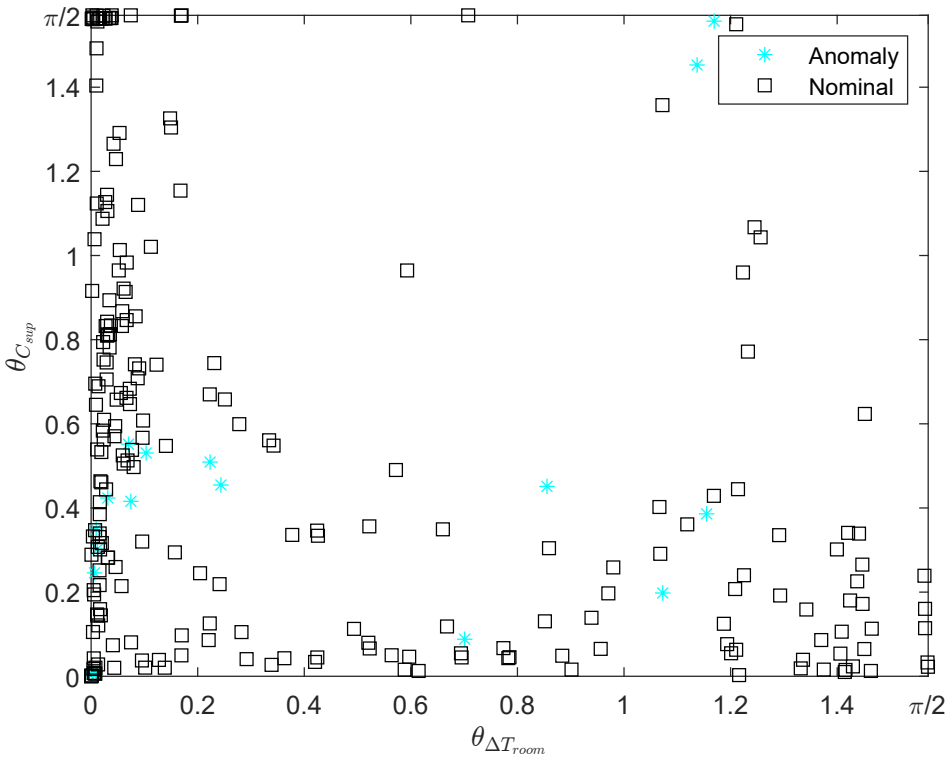

Figure 6-5: Rank-1 MTI Anomaly Detection (λ <= 0.7), (Schnelle et al., 2022)

The two-dimensional plots in Figure 6-5, Figure 6-6 and Figure 6-7 plot the parameters $\theta_{\Delta T_{room}}$ corresponding to the first state against the parameters $\theta_{C_{sup}}$ corresponding to the heating control valve input. Since the interval of anomalous behavior is known, these parameters are marked with cyan stars, while the nominal parameters are marked with black squares. In this data example the LS-SVM classification model is not tuned using the free parameters. Instead, the aim here is to investigate how the weighting factor λ_i in the factor $\widetilde{\mathbf{F}}_\phi$ affects the classification result and how this can be used to tune the classification.

In general, it can be seen that the length factor is crucial for the result of anomaly detection, since no separation of anomalous and nominal behavior is possible for very small values of $\lambda \leq 0.7$ in Figure 6-5. Similarly, the LS-SVM classifier cannot draw a clear class boundary between anomalous and nominal regions when considering the parameter point clouds of the parameter sets with remarkably large weighting factors $\lambda \geq 1.6$ in Figure 6-7. For

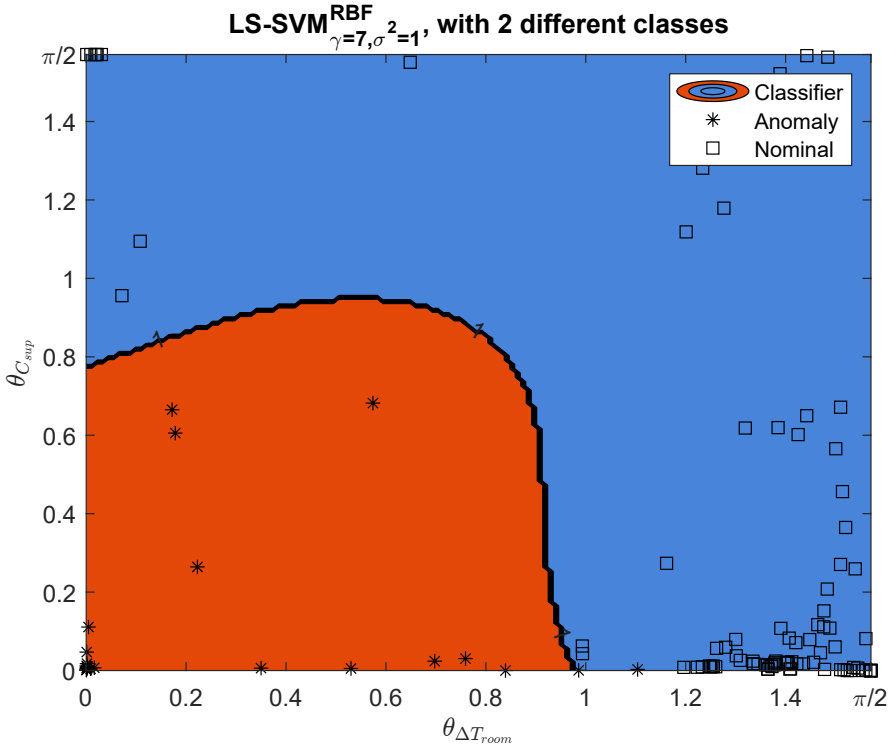

Figure 6-6: Rank-1 MTI Anomaly Detection ($0.7 < \lambda < 1.6$), (Schnelle et al., 2022)

the remaining parameter sets with $0.7 < \lambda < 1.6$, a clear separation of the two classes is possible. Almost all anomalous parameter sets are correctly separated here. No parameters are misclassified as anomalies, which is a requirement for anomaly detection algorithms in HVAC systems.

It follows that the length λ plays an instrumental role in anomaly detection. The identification of reasonable parameters, which are useful for diagnostic purposes, is rendered impossible when the values of λ are exceedingly low. Low weighting parameters in MTI models indicate that the changes in the measurements are too minor. It should be noted that persistent excitation is also relevant for parameter identification of multilinear models. Conversely, very high values of λ can be mapped to events with high rates of change of the measurements, such as outliers or other rare events.

The conclusion is that the parameter λ also serves as a tuning parameter. Due to the previous restriction of the weighting factor, models that are not meaningful for diagnosis because of the data quality are excluded from the classification. In order to avoid parameters being wrongly classified into the anomaly class when the system excitation is too low, as in Figure 6-5, a diagnostic result is omitted for these time steps. Triggering false alarms should be avoided. This also applies to uncertain events that indicate a measurement outlier.

By further investigation, the magnitude of the parameter values in Figure 6-6 allows an interpretation. The magnitudes in the nominal class are either high along the vertical $\theta_{C_{sup}}$ axis or along the horizontal $\theta_{\Delta T_{room}}$ axis or show high values in both directions. This indicates a high dependency of the next state either on the control signal of the heating valve or on the previous state of the room temperature difference. In the anomaly class, the parameters

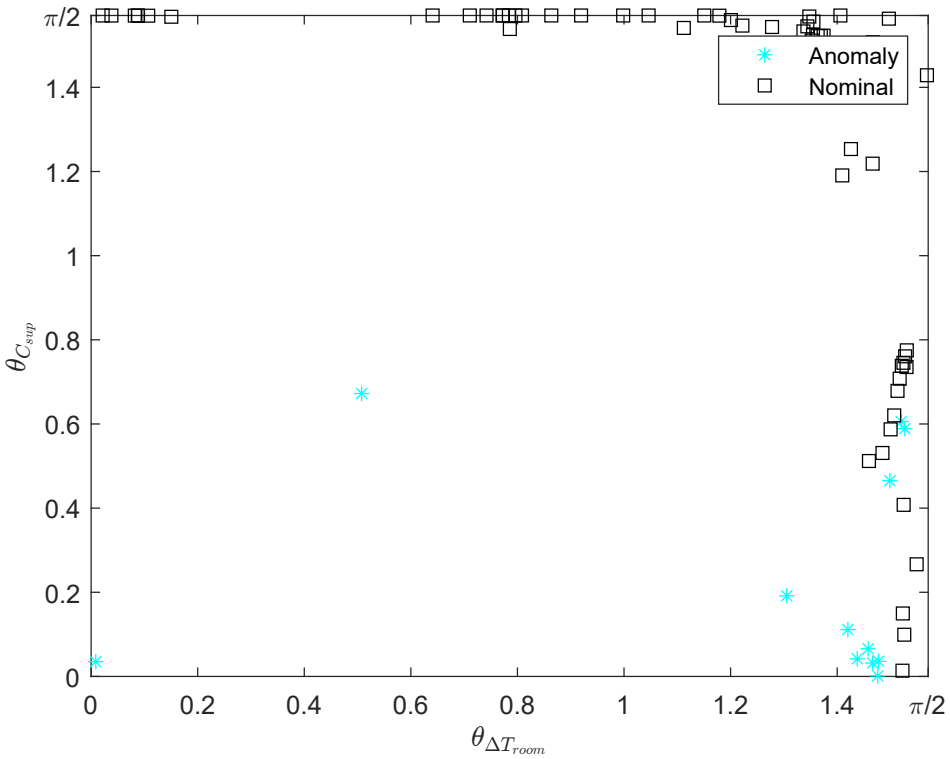

Figure 6-7: Rank-1 MTI Anomaly Detection (λ >= 1.6), (Schnelle et al., 2022)

show different magnitudes. Here the influence of the cool air entering the room is dominant and neglects the influence of the C_{sup} and ΔT_{room}, whose parameters lie at zero angles for a large part of the points. The direct interpretation of the results by the magnitude of the parameter angles is possible due to the CPN representation of the MTI.

The temporal representation of the classified events over the time step k in Figure 6-8 shows

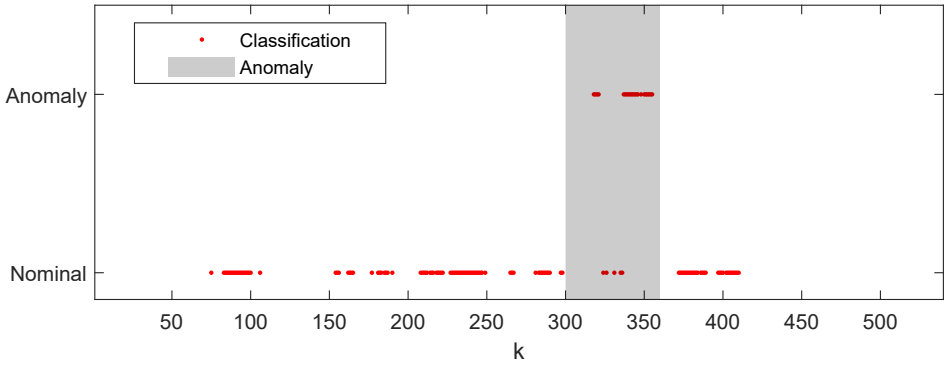

Figure 6-8: Detection of anomaly or nominal behavior, (Schnelle et al., 2022)

that no false alarms are triggered by the algorithm. The open door anomaly is detected after 15 minutes due to the moving horizon. For inertial systems such as buildings, the time difference between anomaly occurrence and detection is acceptable, (Schnelle et al., 2022).

6-1-2 Seminar Room

The second example data set used to evaluate the presented anomaly detection algorithm for supervised data with a single type of anomaly consists of simulation data from a digital twin of a seminar room at the University of Applied Sciences in Augsburg, Germany, (Heinrich et al., 2022). Similar to the previous data set, it provides three input signals (T_{supply}, $\dot{V}_{supply}$ and $T_{outside}$) and two state signals (T_{room}, T_{return}), resulting in an MTI model structure similar to the block diagram shown in Figure 6-2. The performance of the anomaly detection algorithm using this data set was presented in Schnelle et al. (2023).

A rank-1 approximation of a CPN MTI model from Definition 3-1.2 is identified with the Problem 4-1.1 with nonlinear optimizer, as in subsection 4-3-1. Unlike the previous case, the parameter identification algorithm uses sequentially sampled data sets, as described in subsubsection 5-1-1. The time horizon window is set to one day and the sampling time of the simulation data is 15 minutes, giving a set of $2n + m$ parameters per simulated day. An

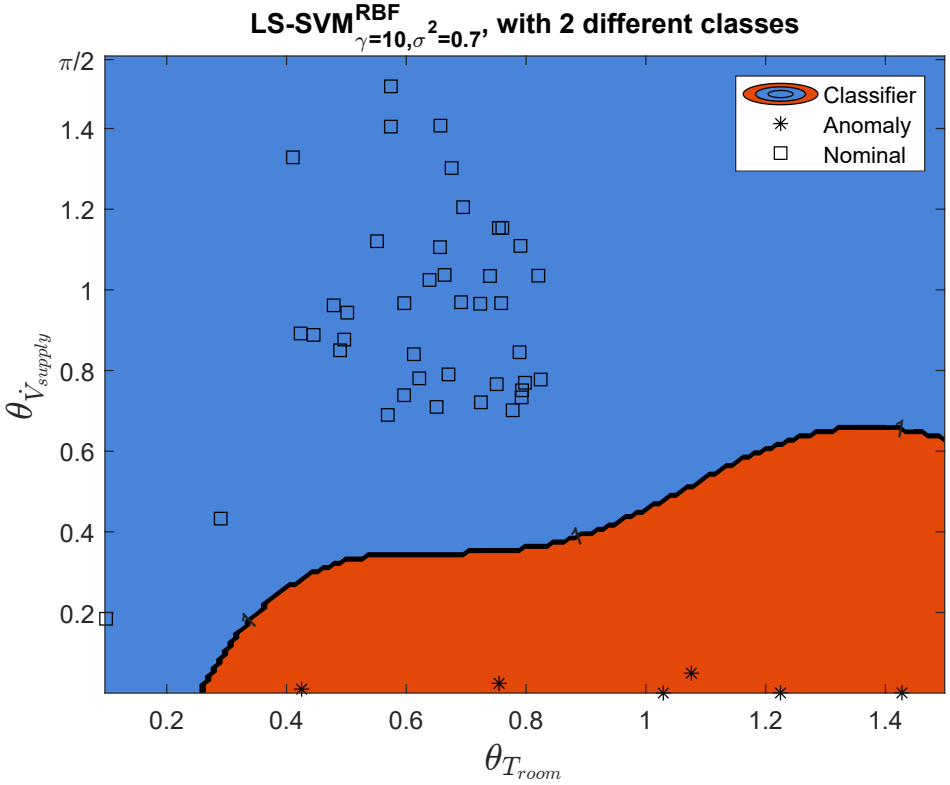

Figure 6-9: Parameter Classification, (Schnelle et al., 2023)

identification procedure optimizes the CPN MTI model from data using the Euclidean norm representation. The nonlinear identification method from subsection 4-3-1 is again applied to identify a normalized rank-1 MTI from data using Problem 4-1.1. The LS-SVM classifier is trained with the parameter results $\theta_{i,1}$ of the first seven days, containing six nominal and one anomaly day. Subsequently, the rest of the daily parameter sets of the 1.5 months of data, including six further unknown anomalies, is fed into the trained classification model. A two-dimensional view of the seven-dimensional classification problem is given over the

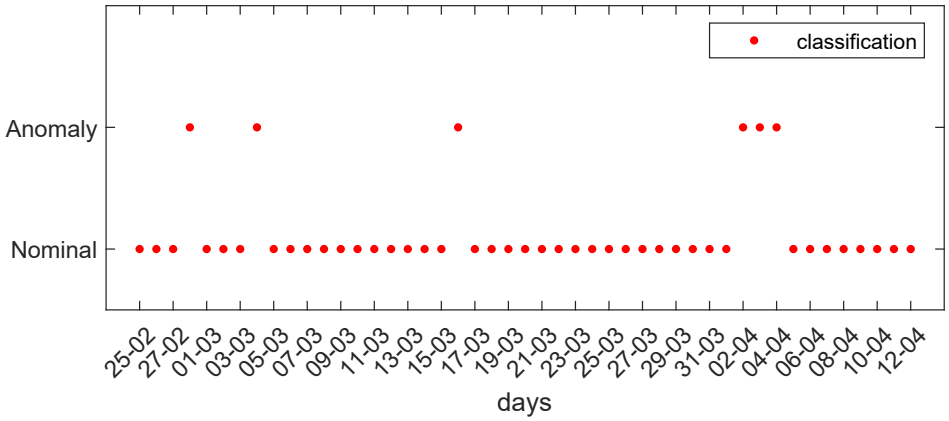

Figure 6-10: Results Anomaly Detection, (Schnelle et al., 2023)

dimension $\theta_{\dot{V}_{supply}}$ and $\theta_{T_{room}}$ in Figure 6-9.

A similar result is obtained as in the previous data set. In the class of anomaly behavior, the angle parameters of $\theta_{\dot{V}_{supply}}$ are small, indicating that the supply air volume flow has little influence on the room temperature state in case of a window opening anomaly, which can be explained by the heat loss through the window.

In total, six of the seven implemented anomalies are detected, as depicted in Figure 6-10. Only one, on April 12th, was not detected by the algorithm. This can be explained by the high outside temperature in Southern Germany on that day, which was close to the indoor temperature and therefore had no effect on the room temperature even when the window was open.

6-2 Application to standard data sets - several anomalies

The following standard data sets are provided by the Lawrence Berkeley National Laboratory. They were produced to evaluate and validate fault detection and diagnostic methods for Heating, Ventilation and Air Conditioning (HVAC) systems in buildings, (Granderson and Lin, 2019).

6-2-1 HVAC system with faulty valves and ducts

System and data

The fault data set was produced by Drexel University and provides simulation data of a small commercial building in Iowa. The building has three air handling units (AHUs), each serving four zones of the building. One of the air handling unit (AHU)s is used as a test system. The system contains two fans to supply and exhaust air to and from the zones. The supply air flow is generated from an outdoor air flow and a recirculation air flow is generated from the return air flow. A preheater is installed for the outdoor air flow. The mixed flow of recirculated and fresh air is either heated by a heating coil or cooled by a cooling coil, both controlled by control valves. Ducts are installed to carry the air to and from the conditioned zones. The return air flow is divided into an exhaust flow and a recirculation flow. The exhaust and fresh air flows are controlled by dampers. The data set includes 17 measurement signals of temperature and pressure measurements and set points, control signals for valve and damper positions and fan speed. An additional signal indicates the occupied mode status and the fault ground truth indicates the presence of faults.

The system is controlled in two different modes. One is the occupied day mode and the other is the unoccupied mode. In the occupied mode, the temperature of the conditioned zones is maintained within an interval 70-72°F. A minimum fresh air flow is set to 40 %. The cooling and heating valves are controlled depending on the outdoor temperature and to maintain the zone temperature set points. The supply fan is equipped with a static pressure control to maintain a static pressure. In the unoccupied night mode, 100 % of the return air is recirculated and the chiller and heater valves are completely closed. The data set contains simulation data for error-free operation and for scenarios with several anomalies in the system with different error intensities. The faults are simulated for stuck and leaking heating and cooling coils and stuck outside air dampers, by manually overwriting the control signals. All faults last for an entire day. The data are not continuous but contains 25 non-consecutive days over a two year period, including 13 anomalies.

Anomaly detection evaluation

The evaluation of the anomaly detection algorithm with the given data set is done with the sequential parameter identification from subsubsection 5-1-1 for residual generation and the residual evaluation with LS-SVM, described in section 5-2. The data is causally ordered in inputs and states, with the mixed air temperature and the return air temperature defined as states. The model structure is generated by using linear correlation analysis. All data points with a correlation of $0.1 \leq c < 1$ to the states and with a correlation $c < 1$ to each other are

Data Point Name	Description	Unit/Type
Supply Air Temperature	Measured AHU supply air temperature	°F
Outdoor Air Temperature	Measured AHU outdoor air temperature	°F
Mixed Air Temperature	Measured AHU mixed air temperature	°F
Return Air Temperature	Measured AHU return air temperature	°F
Supply Air Fan Status	AHU supply air fan status	boolean
Return Air Fan Status	AHU return air fan status	boolean
Supply Air Fan Speed Control Signal	AHU supply air fan speed	range 0 to 1
Return Air Fan Speed Control Signal	AHU return air fan speed	range 0 to 1
AHU: Cooling Coil Valve Control Signal	Control signal for AHU cooling coil valve	range 0 to 1
AHU: Supply Air Duct Static Pressure	Measured AHU supply air duct static pressure	psi
Occupancy Mode Indicator	Indicator if the system operates in occupied mode	boolean
Fault Detection Ground Truth	Indicator if there is a fault present during the day	boolean

Table 6-1: Data point list for DREXEL test data set, (Granderson and Lin, 2019)

used for the identification of the CPN MTI model. The correlation analysis is used to filter out constants, identical data points and data points with no influence on the state over the whole period. Constants and identical data points would lead to ambiguous parameters and are not useful for diagnostic purposes. With the correlation analysis, 11 measurement signals are chosen for the parameter identification, see Table 6-1.

Algorithm 2 is used as parameter identification method for non-negative alternating least squares (ALS) to identify a diagonal rank-n model from Definition 3-1.3. As the implemented faults last for an entire day, the identification horizon is set to 24 hours. The sampling time is set to 5 minutes. With the chosen data points from Table 6-1, the monomial is of dimension $\mathsf{M} \in \mathbb{R}^{\times^{(2+9)}2}$ and the model has $n \cdot 11 = 22$ parameters in the normalized transition matrix $\mathbf{H}$ from (3-31).

To investigate the occurrence of undetected anomalies and false alarms, the tuning of section 5-2 is applied. The results of the classification with LS-SVM over the ten-dimensional parameter clouds for the state x_{Air_mixed} are clustered into nominal and anomalous behavior. Three nominal and three anomalous days were used to train the LS-SVM classifier, resulting in 24 % of the total data set, as shown by the blue shaded part in Figure 6-11.

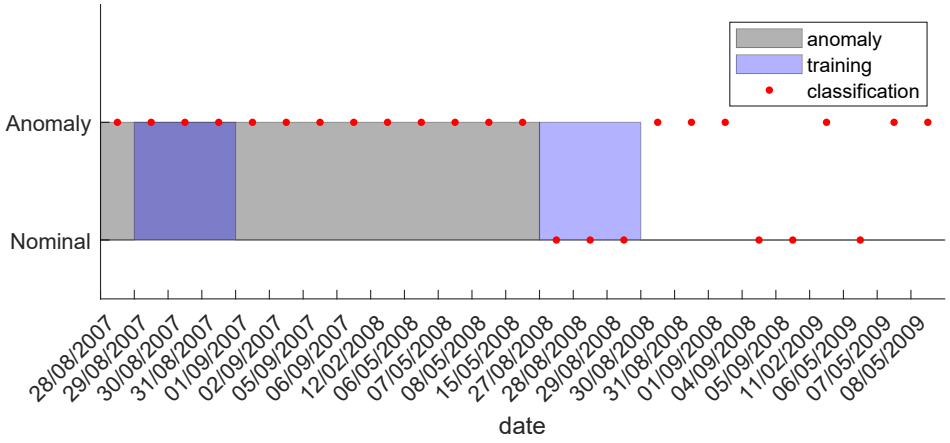

Figure 6-11: Anomaly detection result for DREXEL test data set

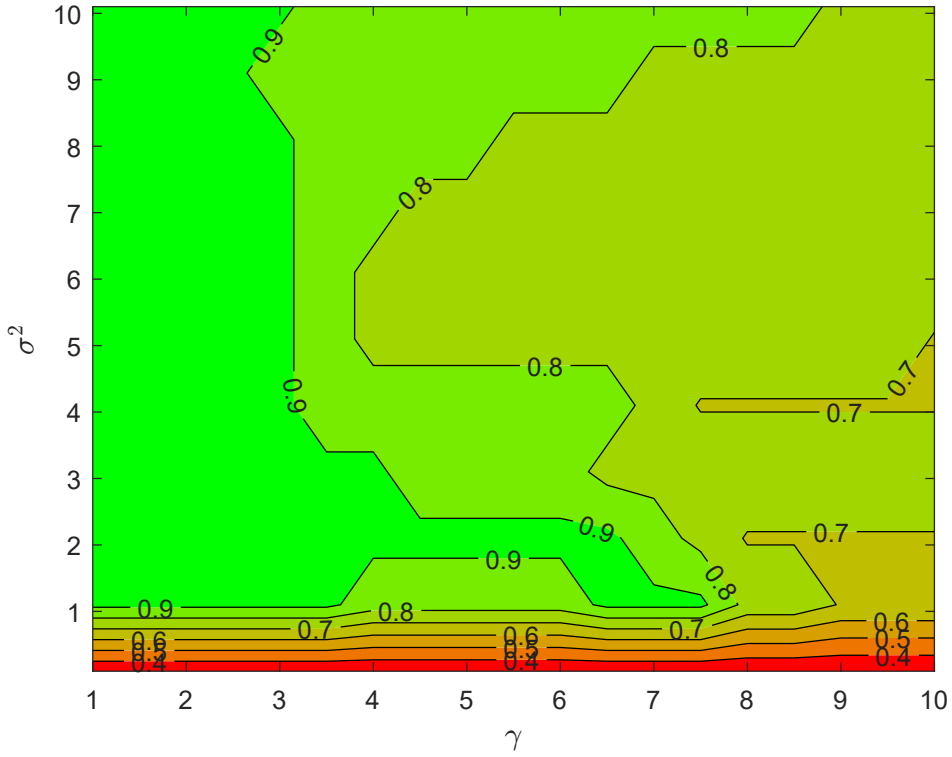

Figure 6-12: Correct positive detections for DREXEL test data set

The true positive anomalies and true negative nominals are shown in Figure 6-12 and in Figure 6-13. Here, the anti-proportionality of undetected anomalies and false alarms is clear, as with decreasing σ^2 and increasing γ false alarms decrease with increasing undetected anomalies. The total percentage of correctly detected events is shown in Figure 6-14. The accuracy of over 80 % of all events covers a wide range of tuning parameters, but similar to the test case in section 5-2, the boundaries where the tuning parameters are either very small or very large, lead to the most incorrectly detected events. The values of the best results and the direction of more false alarms seem to be data and training-dependent and not general. The best result is achieved with $\sigma^2 = 1$ and $\gamma = 8$ with over 85%. The result of the anomaly detection with tuned LS-SVM over the measurement period is shown in Figure 6-11. The tuning is done to avoid false alarms. False alarms are usually undesirable as they are costly if the operator has to send the facility manager or an engineer to each alarm. Although the training only covers a short period, all nominal cases are clustered correctly and 10 out of 13 anomalies of various types were identified.

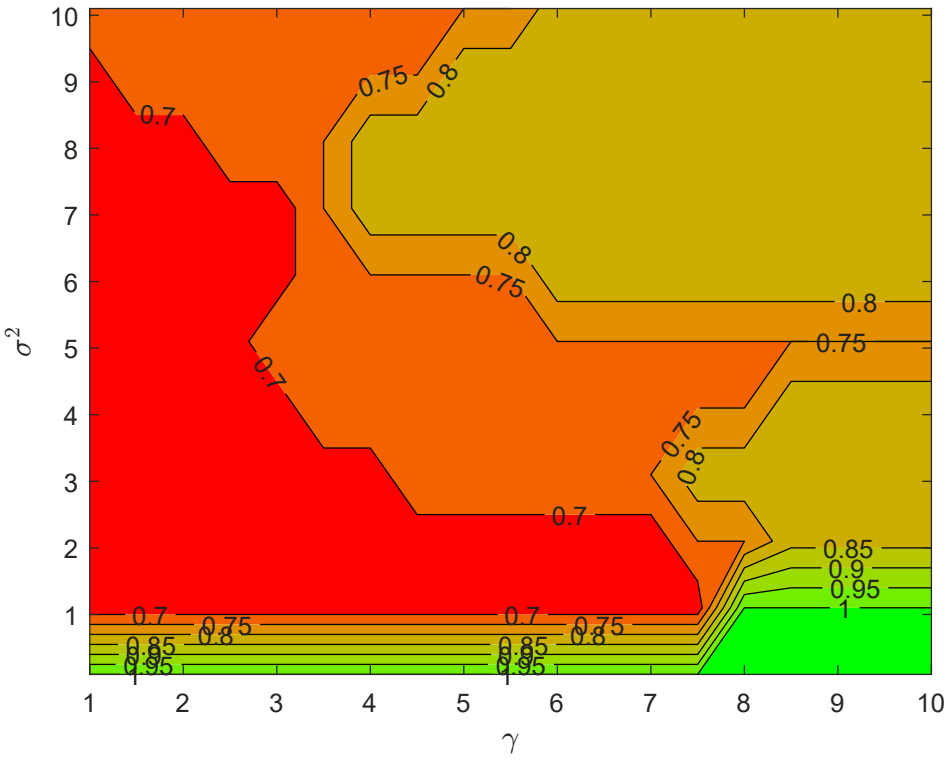

Figure 6-13: Correct negative detections for DREXEL test data set

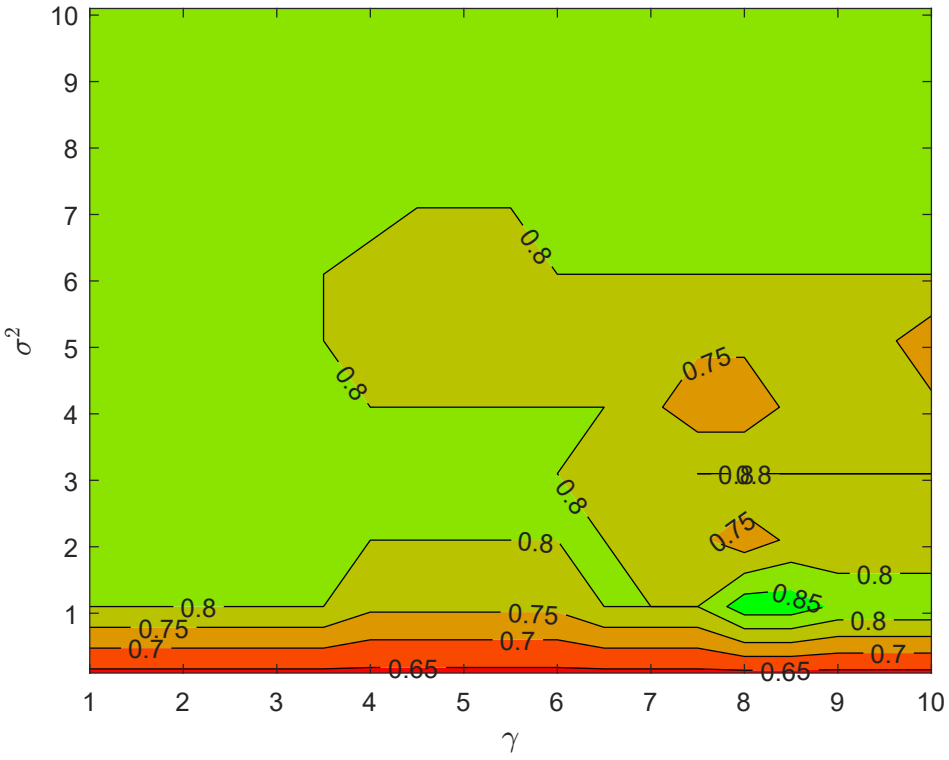

Figure 6-14: Accuracy for DREXEL test data set

6-2-2 HVAC system with temperature sensor bias

System and Building

The second set of standard data for the evaluation of fault diagnosis algorithms is provided by the Pacific Northwest National Laboratory (PNNL) as simulated monitoring data of a large office building model. The simulation of the data is done with a model using a combination of building simulation in EnergyPlus and engineering simulation of the HVAC system in the Modelica tool Dymola, (Granderson and Lin, 2019). The building has three floors. On each floor there are five zones served by a single-duct AHU with variable air volume (VAV) system. The heating and reheating coils are supplied by a natural gas boiler. A central chiller provides chilled water for cooling. The studied AHU system is located in the middle floor and includes components, such as supply and return air fans, cooling and heating coils, and control valves. For recirculation and fresh air outdoor air (OA) and return air (RA) dampers are installed. The control system is installed similarly to the previous data set, distinguishing between an occupied and an unoccupied mode. The occupied mode is scheduled from Monday to Saturday from 6 am to 8 pm. In this mode, the supply air temperature is modulated to 55° F. Heating and cooling are controlled by the zone temperature, with the heating and cooling flow rates modulated by the valve positions. In unoccupied mode, the supply fan is off, the OA damper is closed, and a system cycle is turned on and off to maintain the zone temperature within the unoccupied mode temperature limits. The data set provides set points, control signals, and sensor measurements including temperature signals, damper and valve positions, duct pressure, and fan speed. The data set provides 188 days of faulty and fault-free data. The fault in this data set is a temperature bias of ±1°F to ±4°F in the outdoor temperature sensor measurement.

Anomaly detection for sensor bias test data set

The anomaly detection for the second standard data set is similar to the procedure for the first data set. To preprocess the data, correlation analysis is used to filter out the data points that are not correlated with the states or are almost identical. The list of eleven data points used for parameter identification is shown in Table 6-1. The ground truth is used for training and evaluation.

Identification is performed using the non-negative ALS algorithm from subsection 4-3-2 with a one-day sequential horizon to identify the parameters $\mathbf{H}$ and $\widetilde{\mathbf{F}}_{\phi}$ in 1-norm representation of the diagonal rank-n MTI from Definition 3-1.3 with 5-minute sampled data. The data are causally sorted into states and inputs by defining the mixed air temperature and the return air temperature as states. To check whether the second order rank-2 CPN model can represent the data, an identified parameter set for a daily sequence is used for parameter identification of the CPN model. The result of the rank-2 approximation compared with the real data is shown in Figure 6-15. The figure shows the result for one day of test data used for identification, and 16 days of evaluation data. The rank-2 MTI model is able to represent the main dynamics and the magnitude of the state signal.

For the anomaly detection result, the LS-SVM classifier is tuned with 25 % of the daily identified parameter sets of nominal and anomalous days. Then, all 188 daily parameter

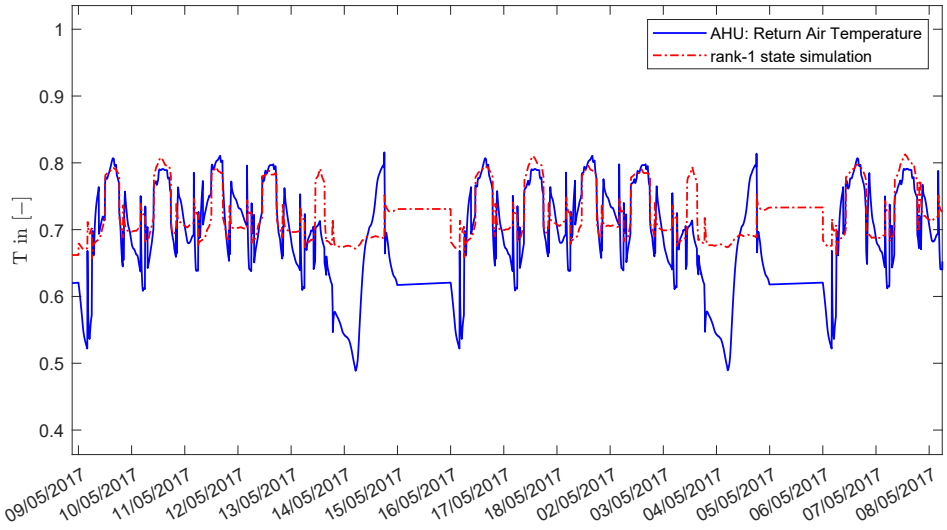

Figure 6-15: Rank-1 identification of sensor bias test data set

Table 6-2: Confusion matrix for sensor bias test data set

		Classification state	
		Positive (P)	Negative (N)
Real State	Positive (P)	119/162 (TP)	43/162 (FN)
	Negative (N)	12/27 (FP)	15/27 (TN)

sets are given to the LS-SVM classifier. The result of correctly and incorrectly classified events is shown in Table 6-2. The total percentage of correctly classified events is 71.28 %. The classification of the events over time is shown in Figure 6-16. It can be seen that many anomalies are classified well, even those that were not in the training set but in the nominal case, most events outside the training set are misclassified, which triggers many false alarms. The result shows that the anomaly detection algorithm with LS-SVM classification is not able to reliably distinguish between the anomalous and nominal behavior by classifying the daily identified parameter sets for this standard data set with the outdoor temperature sensor failure. The temperature bias of ±1°F to ±4°F degrees does not seem to have a significant influence on the system dynamics that differ from the nominal case. One reason may be the difference in disturbance intensity. A temperature difference of one degree between measured and actual value is within the accuracy range of standard outdoor temperature sensors, as they are often affected by changing weather conditions, such as solar irradiation and wind exposure, leading to measurement bias. The binary classifier is not able to discriminate between faulty and nominal behavior hen trained on error intensities between one and four degrees.

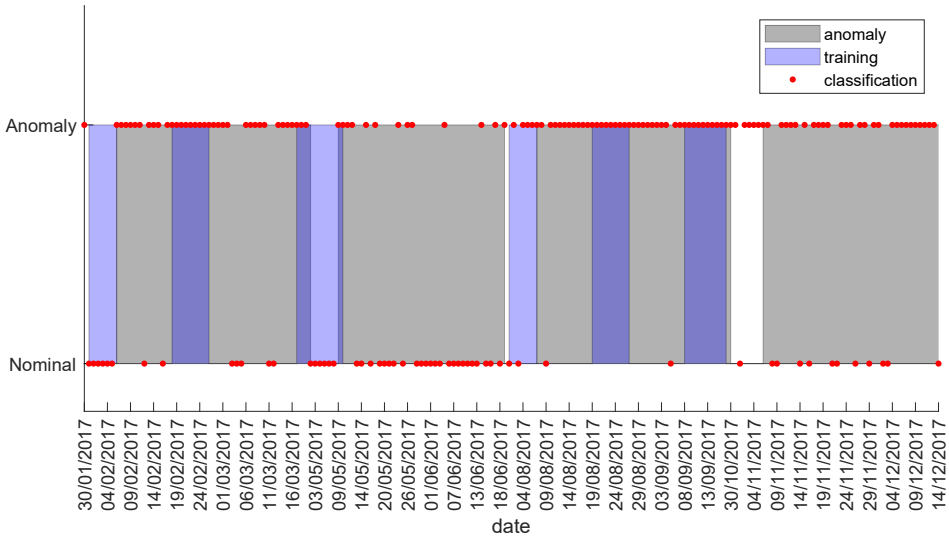

Figure 6-16: Anomaly detection result for sensor bias test data set

6-2-3 Flexlab test cell data with HVAC system anomalies

System and Data

The third standard data set is provided by the Lawrence Berkeley National Laboratory (LBNL) in the FLEXLAB test facility. The FLEXLAB facility is a laboratory for collecting experimental data on building systems. FLEXLAB is equipped with planned internal loads similar to those found in real commercial buildings. The FLEXLAB test cell is served by a single-zone AHU. The main components of the AHU are again heating and cooling coils, valves and dampers for outside fresh air, recirculated air and exhaust air. The control sequence varies between occupied and unoccupied mode. In occupied mode, the zone temperature is maintained by modulating the heating coil valve between a minimum and maximum temperature set point based on demand. The outdoor air control depends on the mode. In economizer mode, when the outdoor temperature is 3.6 $^{\circ}F$ lower than the return air temperature, the outdoor and exhaust air dampers are 100 % open and the return air damper is closed. In unoccupied mode, the zone air temperature is modulated by heating and cooling in the same way as in occupied mode, and the conditioning is deactivated when the set point ($\pm 3.6^{\circ}F$) is reached. The supply fan runs at minimum speed, which is 10 %.

The data set contains data from the summer season, with fault and fault-free scenarios. The faults occur in September and concern stuck and leaking outdoor air dampers and heating and cooling coils. All faults again last for an entire day.

Identification and Anomaly Detection

For the identification of the diagonal rank-n CPN MTI model from Definition 3-1.3 using this data set, the same procedure was applied as for the previous two data sets. Based on the correlation analysis, nine data points were selected for model identification, as listed

in Table 6-3. Using the sequential identification approach described in subsubsection 5-1-1, a CPN MTI model was estimated for each day with the algorithm presented in Algorithm 2.

Data Point Name	Description	Unit/Type
Supply Air Temperature	Measured AHU supply air temperature	°F
Outdoor Air Temperature	Measured AHU outdoor air temperature	°F
Mixed Air Temperature	Measured AHU mixed air temperature	°F
Return Air Temperature	Measured AHU return air temperature	°F
Supply Air Fan Speed Control Signal	AHU supply air fan speed	range 0 to 1
Outdoor Air Damper Control Signal	Control signal for AHU outdoor air damper	range 0 to 1
Return Air Damper Control Signal	Control signal for AHU return air damper	range 0 to 1
Cooling Coil Valve Control Signal	Control signal for AHU cooling coil valve	range 0 to 1
Heating Coil Valve Control Signal	Control signal for AHU heating coil valve	range 0 to 1
Fault Detection Ground Truth	Indicator if there is a fault present during the day	boolean

Table 6-3: Data point list for FLEXLAB (Granderson and Lin, 2019)

The result of the simulated versus the measured first state (return air temperature) of the CPN MTI model with normalized data to the interval $\in [0\ 1]$ is plotted in Figure 6-17.

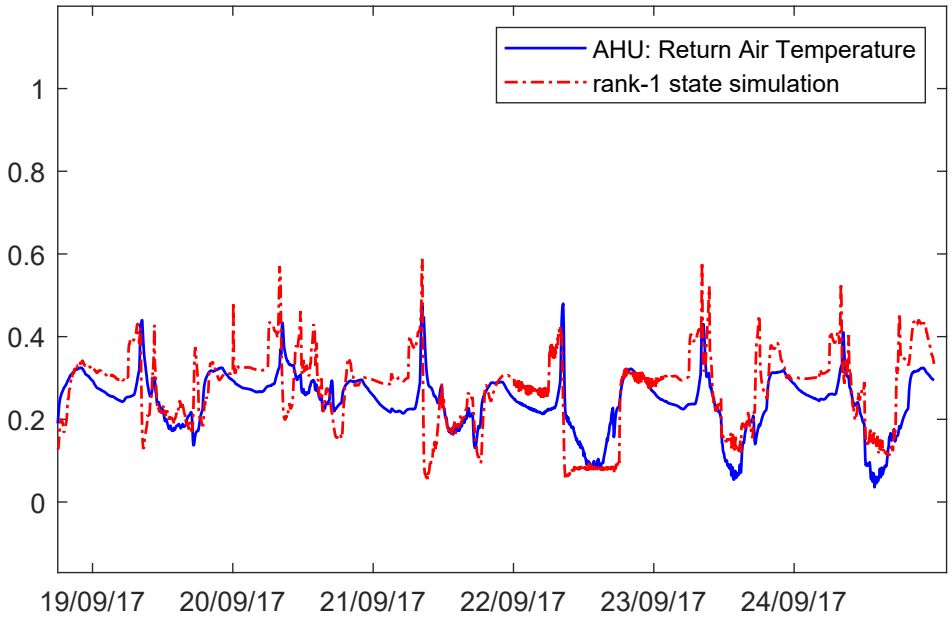

Figure 6-17: Identification result for FLEXLAB data

It shows a high similarity for the identification interval (19/09/17) as well as for the testing interval (20/09 -24/09/17) between the model and the measurements. The performance of the anomaly detection algorithm using LS-SVM is illustrated in Figure 6-18 and shows a good performance with only one false alarm and one undetected fault.

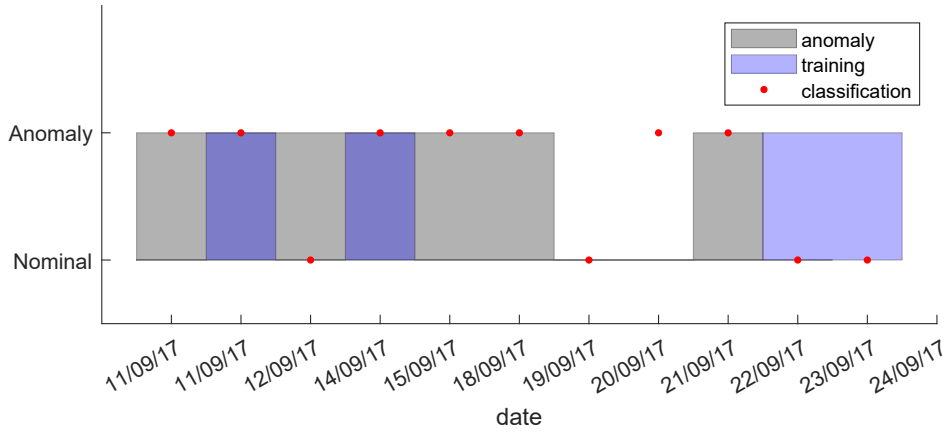

Figure 6-18: Anomaly detection result for FLEXLAB data

6-3 Application to not clustered data sets of real buildings

In this section, the anomaly detection algorithm evaluated in the previous sections on clustered monitoring and simulation data from a single building zone is tested on monitoring data from a real large office building. The main difference and the main challenge is that a large number of unsorted data points are available, but no monitored data labeled as nominal and anomalous behavior. The large amount of unsorted data cannot be used directly, but must first be structured, as described below.

Building and Data

The building under consideration is an office building in Hamburg. It is mainly used as a coworking space for young companies. The building has an area of 7317 m^2 spread over five levels. The basement contains creative offices, laboratories and studios, while the other levels are used as coworking spaces with large open spaces and smaller offices and project rooms. The building uses smart building technology with sensor technology and network protocols to collect and upload weather data and indoor environment parameters, such as temperatures, CO_2 levels and humidity. In addition to measured values, the building management system (BMS) provides other signals such as set points, control signals and operating modes of the HVAC system. Occupancy schedules and occupancy sensors are used in the control system to operate the offices and halls based on actual demand and additional data points are available. A total of 17,000 data points are recorded, collecting time series data from the entire building operating system. In order to use this data, preprocessing and data preparation is required. The structure of the data has to be identified in order to use it for modeling, as described in the next section.

Identification and anomaly detection

For model identification the sparse representation of the CPN MTI from (4-10) is used, as in unstructured full representation the parameter matrix would have $17,000$ diagonal entries

in $\widetilde{\mathbf{F}}_\Phi$ and $17,000 \times 17,000$ parameters in the normalized parameter matrix $\mathbf{H}$, even for the limited diagonal rank-n MTI model from Definition 3-1.3. Since it is very unlikely that every data point has an influence on any other data point in that building a structured sparse MTI model from subsubsection 3-1-2 is used. Furthermore, this assumption would lead to huge models, which would require large storage capacities and computing times, a structured sparse MTI model from subsubsection 3-1-2 is used. For structuring the natural language processing (NLS), as proposed in Algorithm 1, is performed. With the implementation from García (2022), it was able to structure the data in a dependency matrix, which shows less than 50 dependent data points per state. With the diagonal rank-n CPN MTI, each row of the parameter matrix (3-30) represents a submodel for a state; several rows can be used together to create submodels for individual zones or rooms. To neglect duplicate data points or data points showing low excitation or constant values, the linear correlation is used again, as described in section 6-2.

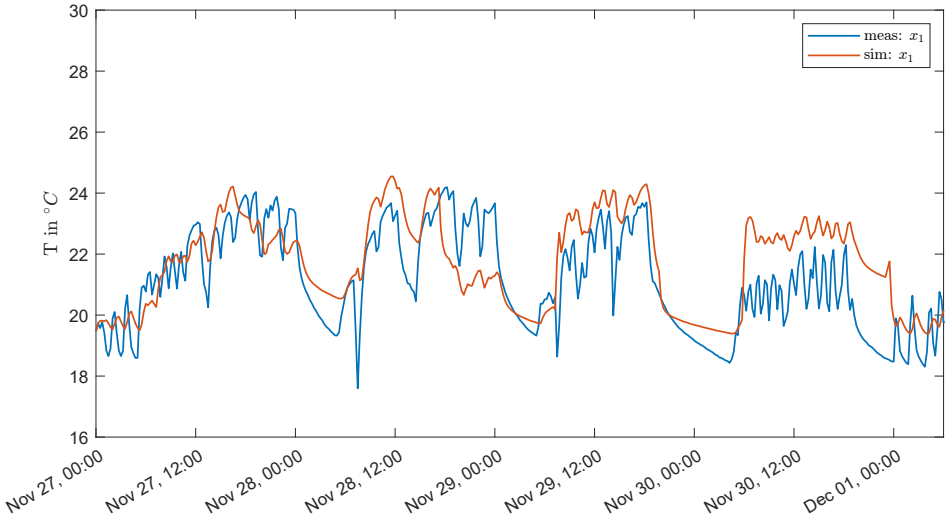

Figure 6-19: Identification result of first state x_1

To evaluate the new parameter identification method using Algorithm 2, a second order submodel for one room of the office building was identified using the ALS optimizer for CPN MTI. Figure 6-19 shows how the identified model represents the measured data of the first state, which is the room temperature. The first day was used as identification data and the next three shown days were used for testing. As it can be seen, the automatically identified structured black-box MTI model with rank limitation is able to represent the main dynamics of the room temperature behavior.

Remark: To avoid numerical problems with data points with a large deviation in magnitude, all data for the identification process are scaled to the unit interval $[0, 1]$ and then rescaled for comparison.

Figure 6-20 shows the results for the second state, the CO_2 sensor measurement. Here, the deviation between the measured and simulated states is larger, especially on the second simulated day. The identified MTI model for one day is not sufficient to represent the changing dynamics of the CO_2 values for different days. This can be explained by the fact that the one-day identification did not take into account all possible state and input trajectories.

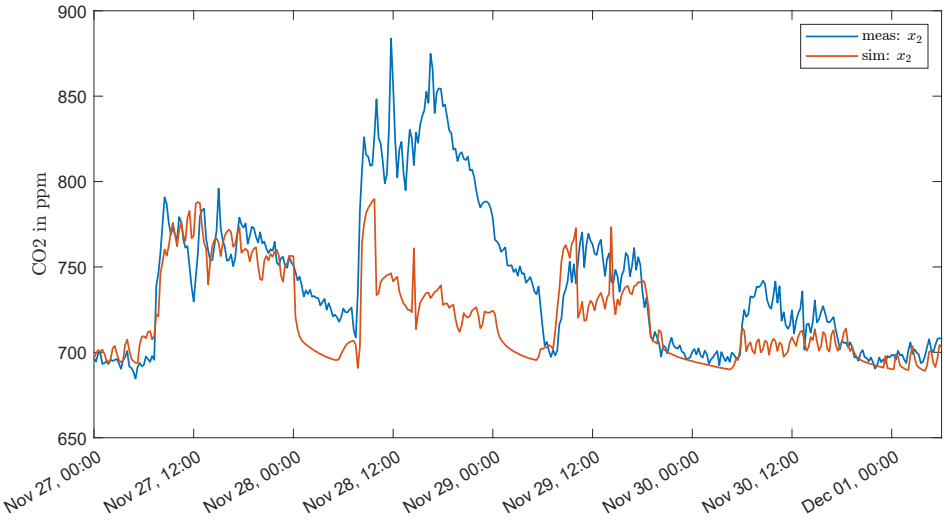

Figure 6-20: Identification result of second state x_2

In addition, the simple occupancy and presence data do not provide any information about the number of people in a room.

To test the anomaly detection algorithm the data were identified for every day over four months in winter. The sampling time was 15 minutes. As no supervised data were available for that building, LS-SVM cannot be used for separation of the parameter sets showing nominal and anomalous behavior. The k-means clustering, see Hartigan and Wong (1979), provides a method to separate a data set into a predefined number of classes. As no training is needed for that method, it is used to test whether it can indicate a possible anomaly during these months. The result of the multidimensional cluster decision is plotted over two weeks in Figure 6-21. Two parameter sets are separated in the second cluster on January 5 and

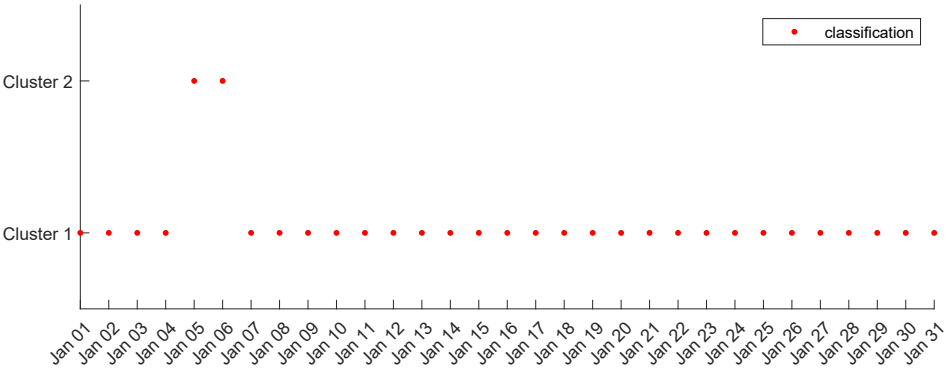

Figure 6-21: Unsupervised clustering result

January 6. Because of the lack of training data, this result cannot be evaluated directly. By analyzing the used data in Figure 6-22, it is noticeable that from January 5, the room temperature (blue line) drops significantly below the temperature set point (orange line). This could actually be anomalous behavior that the algorithm has detected. However, from January 7, the algorithm assigns the nominal cluster again, although the deviation between

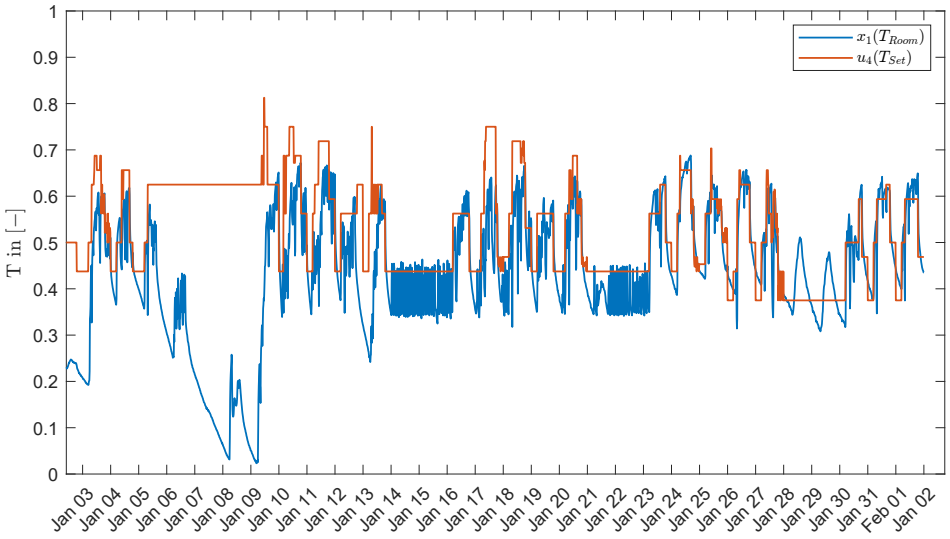

Figure 6-22: Data plot for unsupervised data

room temperature and set point still exists. The untrained model may therefore not be directly suitable for generating an anomaly detection result. It could display a first guess, which could be evaluated and then used as a training data set for LS-SVM.

Chapter 7

Conclusion

7-1 Summary

In this thesis, an anomaly detection method for building systems during their operation is presented using parameter identification of low-rank multilinear models. Chapter 2 first introduces methods for anomaly detection, distinguishing between signal-based and model-based methods. Model-based black-box and gray-box methods require parameter identification methods for the model estimation, which are also explained. Furthermore, multidimensional data structures (tensors) and the Canonical Polyadic (CP) tensor decomposition are introduced as well as the multilinear time-invariant (MTI).

Chapter 3 develops a normalized representation of the normalized CP decomposed MTI models and examines the significance of the normalized model parameters. It is shown that the influence of a signal on the state at the next time step can be read directly from the parameters. In addition, normalization prevents arbitrary linear parameter combinations from representing the same multilinear models and thus lays the foundation for using the new unique model parameters for anomaly detection of multilinear normalized rank-1 models. The sparse representation and prior model structuring allow scaling to large models, which are based on multiple signals.

Chapter 4 deals with the parameter identification of multilinear models. Two different approaches are examined. First, a two-step approach of parameter identification of multilinear models is presented, where the full parameter tensor is identified and then brought into the desired representation by decomposition and normalization. Due to the form of the multilinear equations, which are linear in the parameters, a straightforward solution using the least squares method from linear algebra is possible. However, the number of parameters increases exponentially with each additional state or input, meaning that this method is not scalable. Subsequently, the direct identification of the normalized CP decomposed reduced parameter matrices is developed. An analysis shows that the decomposed optimization problem is non-convex. To solve the optimization problem, a standard nonlinear optimizer is first used, which shows large computation times. Subsequently, a specific optimizer for

the CP decomposed MTI models is developed using an alternating least squares method, which achieves a significant reduction in the optimization time and also a slight improvement in the optimization error.

Chapter 5 then describes an algorithm for anomaly detection by examining the model parameters of the MTI models. A moving horizon or sequential identification of the model parameters using time series data, collected in buildings, is proposed. Residuals of the parameters are then evaluated with existing clustering algorithms to generate a binary anomaly detection result (anomalous/nominal). Due to the uniqueness of the model parameters required for anomaly detection, an example is used to show how a rank-n model can be approximated for a heated room despite the strong limitation of the model dynamics and that a certain robustness against white input noise is given. A tuning procedure is described and investigated with simulation data of a different room, which shows that the sensitivity of the classification model can be tuned between undetected anomalies and false alarms.

The application of the developed anomaly detection algorithm is examined in Chapter 6. Different example data sets are used to evaluate how multilinear rank-1 parameter identification can approximate the monitoring data from building systems. The application to two standard data sets with different faults and fault intensity in the HVAC systems showed good performance, while a bias in the outside temperature sensor from a third standard data set were detected insufficiently. Two more data sets with window/door openings as anomalies were used. One with simulation data from a white-box digital twin and one from real measured data from a test office. Both showed good results. At the end, an example with unsupervised measurement data from a big office and coworking building were used. Here the anomaly detection could not be evaluated, but the interpretation of the identified parameters were reasonable.

7-2 Outlook

Modeling and Identification: The whole algorithm is assuming states as measurable outputs and therefore only the state transition equation is considered in the modelling and identification procedure. Extending the algorithm to output equations would take into account potentially noisy measurements. In the modeling of normalized MTI models the structure of the output equation is known, but the parameter identification procedure would have to be adjusted to estimate the non-measurable states in addition to the parameters. State estimators are not available for the multilinear model class yet. Furthermore, the anomaly detection could be extended to rank-n normalized CP decomposed (CPN) MTI parameters. This could help to reduce the approximation error and represent the dynamics of the building more accurately. Anomalies that only change the dynamic behavior of the building to a small extent could thus be detected more easily, but the problem of ambiguous models and how they affect the classification process should be investigated.

Classification: For unsupervised anomaly detection without the availability of training data, different classification methods could be evaluated. There are methods e.g. for one class classification or unsupervised outlier detection, which could not be investigated in this work. Furthermore, a training for multiple classes could be investigated to train different operation modes as for different seasons. This could also lead to training specific anomalies

and detecting them later in the application phase. This could eventually extend the given algorithm into a fault detection and identification algorithm.

Application: In order to use the developed algorithm for online anomaly detection in real buildings, training data and an evaluation phase are needed to assess whether the alarms generated by the classification model indicate real anomalies. Since training data is not always available, one could investigate whether the nominal parameters of the normalized multilinear models of one building resemble those of another building of similar construction and use. If this is the case, a trained classification model could be transferred for initialization and further tuned during the operation of the specific building.

Bibliography

Al Dakheel, J., Del Pero, C., Aste, N., and Leonforte, F. (2020). Smart buildings features and key performance indicators: A review. *Sustainable Cities and Society*, 61:102328.

Alshibani, A. (2020). Prediction of the energy consumption of school buildings. *Applied Sciences*, 10(17):5885.

Åström, K. J. and Eykhoff, P. (1971). System identification—A survey. *Automatica*, 7(2):123–162.

Awawdeh, M. J., Faisal, T., Bashir, A., Alshbatat, A. I. N., and Momani, R. T. (2024). Moving-horizon estimation approach for nonlinear systems with measurement contaminated by outliers. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 22(1):219–231.

Bader, B. W., Kolda, T. G., et al. (2019). Tensor Toolbox for MATLAB, Version 3.1. https://www.tensortoolbox.org. Accessed: 2020-03-27.

Baskiotis, C., Raymond, J., and Rault, A. (1979). Parameter identification and discriminant analysis for jet engine machanical state diagnosis. In *1979 18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, volume 2, pages 648–650.

Batselier, K., Chen, Z., and Wong, N. (2017). Tensor Network alternating linear scheme for MIMO Volterra system identification. *Automatica*, 84:26–35.

Batselier, K., Ko, C.-Y., Phan, A.-H., Cichocki, A., and Wong, N. (2018). Multilinear state space system identification with matrix product operators. *IFAC-PapersOnLine*, 51:640–645.

Becker, S., Hagen, J., Joshi, S., Krüger, R., and de la Serna, S. (2023). Deutsche Energie-Agentur (DENA) (ed.) : DENA-GEBÄUDEREPORT 2024. Zahlen, Daten, Fakten zum Klimaschutz im Gebäudebestand. https://www.dena.de/infocenter/dena-gebaeudereport-2024/. Accessed: 2024-10-01.

BMWK (2022). Bundesministerium für Wirtschaft und Klimaschutz (BMWK). Energieeffizienz in Zahlen 2022 – Entwicklung und Trends in Deutschland. https://www.bmwk.de/

Redaktion/DE/Publikationen/Energie/energieeffizienz-in-zahlen-2022.html. Accessed: 2023-03-27.

Borda, D., Bergagio, M., Amerio, M., Masoero, M. C., Borchiellini, R., and Papurello, D. (2023). Development of Anomaly Detectors for HVAC Systems Using Machine Learning. *Processes*, 11(2):535.

Boyd, S. and Vandenberghe, L. (2009). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Bürger, D. V., Hesse, D. T., Palzer, D. A., Köhler, B., Herkel, S., Engelmann, D. P., and Quack, D. D. (2020). Klimaneutraler Gebäudebestand 2050 - Energieeffizienzpotenziale und die Auswirkungen des Klimawandels auf den Gebäudebestand. page 289.

Carabantes, M. (2020). Black-box artificial intelligence: an epistemological and critical analysis. *AI & SOCIETY*, 35(2):309–317.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58.

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 & Sons.

Daissaoui, A., Boulmakoul, A., Karim, L., and Lbath, A. (2020). IoT and Big Data Analytics for Smart Buildings: A Survey. *Procedia Computer Science*, 170:161–168.

David, A., Leeb, M., and Bednar, T. (2017). Comparison of the planned and the real energy consumption of the world's first (Plus-)Plus-Energy Office High-Rise Building. *Energy Procedia*, 132:543–548.

De Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., De Brabanter, J., Pelckmans, K., De Moor, B., Vandewalle, J., and Suykens, J. A. (2010). *LS-SVMlab toolbox user's guide: version 1.7*. Katholieke Universiteit Leuven.

De Bruijn, H., Warnier, M., and Janssen, M. (2022). The perils and pitfalls of explainable AI: Strategies for explaining algorithmic decision-making. *Government Information Quarterly*, 39(2):101666.

Domanov, I. and De Lathauwer, L. (2013). On the uniqueness of the canonical polyadic decomposition of third-order tensors—part i: Basic results and uniqueness of one factor matrix. *SIAM Journal on Matrix Analysis and Applications*, 34(3):855–875.

Europäisches Parlament and Rat der Europäischen Union (2024). *Richtlinie (EU) 2024/1275 des Europäischen Parlaments und des Rates vom 24. April 2024 über die Gesamtenergieeffizienz von Gebäuden (Neufassung) (Text von Bedeutung für den EWR)*. Europäische Union. http://data.europa.eu/eli/dir/2024/1275/oj/deu, Accessed: 2024-11-02.

Frank, P. and Wunnenberg, J. (1989). Robust fault diagnosis using unknown input observer schemes. *Fault diagnosis in dynamic systems*, pages 47–97.

Ganesh, H. S., Seo, K., Fritz, H. E., Edgar, T. F., Novoselac, A., and Baldea, M. (2021). Indoor air quality and energy management in buildings using combined moving horizon estimation and model predictive control. *Journal of Building Engineering*, 33:101552.

Gao, Z., Cecati, C., and Ding, S. X. (2015). A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches. *IEEE Transactions on Industrial Electronics*, 62(6):3757–3767. Conference Name: IEEE Transactions on Industrial Electronics,.

García, M. D. E. (2022). *Automated multilinear parameter identification from big data of buildings*. Master's thesis, Universitat Politècnica de València.

Gram-Hanssen, K. and Georg, S. (2018). Energy performance gaps: promises, people, practices. *Building Research & Information*, 46(1):1–9.

Granderson, J. and Lin, G. (2019). Inventory of data sets for afdd evaluation. *Building Technology and Urban Systems Division, Lawrence Berkeley National Laboratory*.

Grasedyk, L., Kressner, L., and Tobler, C. (2013). A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36:53–78.

Groß, J. (2003). *Linear regression*, volume 175. Springer Science & Business Media.

Gugliermetti, L., Cumo, F., and Agostinelli, S. (2024). A Future Direction of Machine Learning for Building Energy Management: Interpretable Models. *Energies*, 17(3):700.

Guo, Z. and Small, D. S. (2016). Control function instrumental variable estimation of nonlinear causal effect models. *Journal of Machine Learning Research*, 17(100):1–35.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108.

Heinrich, J., Schnelle, L., Jacob, D., and Bauer, M. (2022). Modelle von Nichtwohngebäuden für die Gebäudebetriebsoptimierung. In *Proceedings of BauSim Conference 2022: 9th Conference of IBPSA-Germany and Austria*, volume 9, Weimar.

Himeur, Y., Elnour, M., Fadli, F., Meskin, N., Petri, I., Rezgui, Y., Bensaali, F., and Amira, A. (2023). AI-big data analytics for building automation and management systems: A survey, actual challenges and future perspectives. *Artificial Intelligence Review*, 56(6):4929–5021.

Hirschberg, J. and Manning, C. (2015). Advances in natural language processing. *Science*, 349(6245):261–266.

Huffel, S. v. and Vandewalle, J. (1989). Comparison of total least squares and instrumental variable methods for parameter estimation of transfer function models. *International Journal of Control*, 50(4):1039–1056.

Isermann, R. (1994). Integration of fault detection and diagnosis methods. *IFAC Proceedings Volumes*, 27(5):575–590. IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'94), Espoo, Finland, 13-16 June.

Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods — An introduction. *Control Engineering Practice*, 5(5):639–652.

Isermann, R. (2005). Model-based fault-detection and diagnosis – status and applications. *Annual Reviews in Control*, 29(1):71–85.

Isermann, R. (2006). *Fault-Diagnosis Systems*. Springer, Berlin, Heidelberg.

Isermann, R. and Münchhof, M. (2011). *Identification of Dynamic Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Jamaludin, I. W., Wahab, N. A., Khalid, N. S., Sahlan, S., Ibrahim, Z., and Rahmat, M. F. (2013). N4SID and MOESP subspace identification methods. In *2013 IEEE 9th International Colloquium on Signal Processing and its Applications*, pages 140–145.

Jansson, M. and Wahlberg, B. (1996). A linear regression approach to state-space subspace system identification. *Signal Processing*, 52(2):103–129.

Jöres, N., Kaufmann, C., Schnelle, L., Yáñez, C. C., Pangalos, G., and Lichtenberg, G. (2022). Reduced cp representation of multilinear models. In *SIMULTECH*, pages 252–259.

Kalman, R. E. (1958). Design of a self-optimizing control system. *Transactions of the American Society of Mechanical Engineers*, 80(2):468–477.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45.

Kim, W. and Katipamula, S. (2018). A review of fault detection and diagnostics methods for building systems. *Science and Technology for the Built Environment*, 24(1):3–21.

Kolda, T. and Bader, B. (2009). Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500.

Kommenda, M., Burlacu, B., Kronberger, G., and Affenzeller, M. (2020). Parameter identification for symbolic regression using nonlinear least squares. *Genetic Programming and Evolvable Machines*, 21(3):471–501.

Kruppa, K. (2017a). Comparison of Tensor Decomposition Methods for Simulation of Multilinear Time-Invariant Systems with the MTI Toolbox **This work was partly supported by the project OBSERVE of the Federal Ministry for Economic Affairs and Energy Germany (Grant-No.: 03ET1225B). *IFAC-PapersOnLine*, 50(1):5610–5615.

Kruppa, K. (2017b). MTI Toolbox, Version 1.0. https://www.ls.haw-hamburg.de/~prolib/index.php/toolbox/. Accessed: 2024-07-01.

Kruppa, K. (2018). *Multilinear Design of Decentralized Controller Networks for Building Automation Systems*. PhD thesis, Hafen City Universität Hamburg.

Larimore, W. (1990). Canonical variate analysis in identification, filtering, and adaptive control. In *29th IEEE Conference on Decision and Control*, pages 596–604 vol.2.

Lautenschlager, B. (2019). *Data-Driven Learning and Model Predictive Control for Heating Systems*. PhD thesis, HafenCity Universität Hamburg.

Lawson, C. and Hanson, R. (1995). *Solving Least Squares Problems*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics.

Li, K.-L., Huang, H.-K., Tian, S.-F., and Xu, W. (2003). Improving one-class svm for anomaly detection. In *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693)*, volume 5, pages 3077–3081. IEEE.

Lichtenberg, G. (2011). *Hybrid Tensor Systems*. Habilitation, Hamburg University of Technology.

Lichtenberg, G., Pangalos, G., Yáñez, C. C., Luxa, A., Jöres, N., Schnelle, L., and Kaufmann, C. (2022). Implicit multilinear modeling: An introduction with application to energy systems. *at - Automatisierungstechnik*, 70(1):13–30.

Lichtenberg, G., Uhlenberg, E., Warnecke, T., Cateriano, C., Engels, M., Samaniego, L., Schnelle, L., and Kaufmann, C. (2024). MTI Toolbox, Version 2.0. https://www.ls.haw-hamburg.de/~prolib/index.php/toolbox/. Accessed: 2024-07-01.

Ljung, L. (2002). Prediction error estimation methods. *Circuits, Systems and Signal Processing*, 21(1):11–21.

Lunze, J. (2014). *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*. Springer-Textbook. Springer Berlin Heidelberg, 10 edition.

MathWorks (2023a). Matlab global optimization toolbox. https://de.mathworks.com/products/global-optimization.html. Accessed: 2023-05-01.

MathWorks (2023b). Matlab optimization toolbox. https://www.mathworks.com/products/optimization.html. Accessed: 2023-05-01.

Müller-Eping, T. (2020). *Tensordekomposition qualitativer Modelle zur Fehlererkennung*. PhD thesis, Karlsruhe Institute of Technology.

Neumann, C., Jacob, D., Burhenne, S., Florita, A. R., Burger, E., and Schmidt, F. (2011). Modben - modellbasierte methoden für die fehlererkennung und optimierung im gebäudebetrieb. Technical report, Fraunhofer-Institut für Solare Energiesysteme ISE, Freiburg.

Nocedal, J. and Wright, S. J. (2006). Interior-point methods for nonlinear programming. *Numerical Optimization*, pages 563–597.

Oseledets, I. V. (2013). Constructive Representation of Functions in Low-Rank Tensor Formats. *Constructive Approximation*, 37(1):1.

Pajares, R. G. (2024). MySMARTLife - Research project. https://www.mysmartlife.eu/mysmartlife/. Funded from the European Union's. https://www.mysmartlife.eu/mysmartlife. Accessed: 2024-11-11.

Pangalos, G. (2016). *Model-based controller design methods for heating systems*. Thesis, Technische Universität Hamburg.

Pangalos, G., Eichler, A., and Lichtenberg, G. (2013). Tensor Systems: Multilinear Modeling and Applications. In *3rd Int. Conference on Simulation and Modeling Methodologies, Technologies and Applications*, Reykjavik.

Pangalos, G., Eichler, A., and Lichtenberg, G. (2015). Hybrid Multilinear Modeling and Applications. In Obaidat, M. S., Koziel, S., Kacprzyk, J., Leifsson, L., and Ören, T., editors, *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing, pages 71–85, Cham. Springer International Publishing.

Rehault, N., Böhme, G., and Müller, T. (2015). Energetischer Gebäudebetrieb in der Praxis und neue Methoden für eine energieeffiziente Betriebsführung.

Rogers, A. P., Guo, F., and Rasmussen, B. P. (2019). A review of fault detection and diagnosis methods for residential air conditioning systems. *Building and Environment*, 161:106236.

Schnelle, L., Heinrich, J., Schneidewind, J., Jacob, D., and Lichtenberg, G. (2023). Using structured low-rank tensors for multilinear modeling of building systems. *IFAC-PapersOnLine*, 56(2):7306–7311.

Schnelle, L. and Lichtenberg, G. (2024). SONDE – Supervision und Optimierung neuer Gebäude aus Daten Exploration. Abschlussbericht, HAW Hamburg. https://doi.org/10.34657/23824, Accessed: 2025-10-17.

Schnelle, L., Lichtenberg, G., and Warnecke, C. (2022). Using Low-rank Multilinear Parameter Identification for Anomaly Detection of Building Systems. *IFAC-PapersOnLine*, 55(6):470–475.

Sewe, E., Pangalos, G., and Lichtenberg, G. (2019). Approaches to Fault Detection for Heating Systems Using CP Tensor Decompositions. In Obaidat, M. S., Ören, T., and Rango, F. D., editors, *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing, pages 128–152, Cham. Springer International Publishing.

Sewe, E. M. (2018). *Automatisierte Fehlererkennung in Heizungsanlagen*. PhD thesis, Technische Universität Dresden.

Shi, Z. and O'Brien, W. (2019). Development and implementation of automated fault detection and diagnostics for building systems: A review. *Automation in Construction*, 104:215–229.

Simani, S., Fantuzzi, C., and Patton, R. J. (2013). *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. Springer Science & Business Media.

Sridharan, A., Lichtenberg, G., Salvador, A., and Salgado, C. (2020). Approaches to Parameter Identification for Hybrid Multilinear Time Invariant Systems. In *7rd Int. Conference on Simulation and Modeling Methodologies, Technologies and Applications*, pages 255–262.

Starovoitov, V. V. and Golub, Y. I. (2020). Comparative study of quality estimation of binary classification. In *Informatics*, volume 17, pages 87–101.

Steinwart, I. and Christmann, A. (2008). *Support Vector Machines*. Information Science and Statistics. Springer-Verlag, New York.

Stinner, F., Kornas, A., Baranski, M., and Müller, D. (2018). Structuring building monitoring and automation system data. *The REHVA European HVAC Journal-August*, 2018:10–15.

Suykens, J. A. K., Gestel, T. V., and Brabanter, J. D. (2002). *Least Squares Support Vector Machines*. World Scientific.

Söderström, T. and Mahata, K. (2002). On instrumental variable and total least squares approaches for identification of noisy systems. *International Journal of Control*, 75(6):381–389. Publisher: Taylor & Francis.

Van Hueffel, S. (1988). Documented Fortran 77 programs of the extended classical total least squares algorithm, the partial singular value decomposition algorithm and the partial total least squares algorithm. Technical Report 88/1, Dept. of Electrical Engineering, Katholieke Universiteit Leuven.

Van Overschee, P. and De Moor, B. (1993). N4SID: Numerical Algorithms for State Space Subspace System Identification. *IFAC Proceedings Volumes*, 26(2, Part 5):55–58.

Van Overschee, P. and De Moor, B. L. (1996). *Subspace Identification for Linear Systems: Theory — Implementation — Applications*. Springer US.

Wang, W., Brambley, M. R., Kim, W., Somasundaram, S., and Stevens, A. J. (2018). Automated point mapping for building control systems: Recent advances and future research needs. *Automation in Construction*, 85:107–123.

Wicker, J., Hua, Y. C., Rebello, R., and Pfahringer, B. (2019). XOR-Based Boolean Matrix Decomposition. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 638–647.

Xu, D. and Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2):165–193.

Young, P. C. (2015). Refined instrumental variable estimation: Maximum likelihood optimization of a unified Box–Jenkins model. *Automatica*, 52:35–46.

Zhang, J., Swain, A. K., and Nguang, S. K. (2016). *Robust Observer-Based Fault Diagnosis for Nonlinear Systems Using MATLAB®*. Advances in Industrial Control. Springer International Publishing, Cham.