Data-Driven Modeling for Digital Representations in Energy Systems

Marcel Christian Zimmer

Energie & Umwelt / Energy & Environment
Band / Volume 712
ISBN 978-3-95806-927-5

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum

Data-Driven Modeling for Digital Representations in Energy Systems

Datengetriebene Modellierung von digitalen Darstellungen in Energiesystemen

Von der Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Marcel Christian Zimmer

Berichter: Univ.-Prof. Dr.-Ing. Andrea Benigni
Prof. Dr. rer. nat. Dirk Witthaut

Tag der mündlichen Prüfung: 27. März 2026

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Marcel Christian Zimmer

Data-Driven Modeling for Digital Representations in Energy Systems

Forschungszentrum Jülich GmbH
Institute of Climate and Energy Systems (ICE)
Energiesystemtechnik (ICE-1)

Data-Driven Modeling for Digital Representations in Energy Systems

Marcel Christian Zimmer

Schriften des Forschungszentrums Jülich
Reihe Energie & Umwelt / Energy & Environment Band / Volume 712

ISSN 1866-1793 ISBN 978-3-95806-927-5

Bibliografische Information der Deutschen Nationalbibliothek.
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten
sind im Internet über http://dnb.d-nb.de abrufbar.

Herausgeber und Vertrieb:	Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag 52425 Jülich Tel.: +49 2461 61-5368 Fax: +49 2461 61-6103 zb-publikation@fz-juelich.de www.fz-juelich.de/zb
Umschlaggestaltung:	Grafische Medien, Forschungszentrum Jülich GmbH
Druck:	Grafische Medien, Forschungszentrum Jülich GmbH
Copyright:	Forschungszentrum Jülich 2026

Schriften des Forschungszentrums Jülich
Reihe Energie & Umwelt / Energy & Environment, Band / Volume 712

D 82 (Diss. RWTH Aachen University, 2026)

ISSN 1866-1793
ISBN 978-3-95806-926-8 (Print)
ISBN 978-3-95806-927-5 (E-Book)

Vollständig frei verfügbar über das Publikationsportal des Forschungszentrums Jülich (JuSER)
unter www.fz-juelich.de/zb/openaccess.



This is an Open Access publication distributed under the terms of the Creative Commons Attribution License 4.0, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

The growing complexity arising from new technologies, particularly their interconnections, enabling dedicated planning, monitoring, and control of modern energy systems, raises the need for powerful concepts that provide detailed insights about the particular energy system and its components. In contrast to classical models that provide a generalized system description, this thesis investigates the gap between the physical and digital worlds, yielding a replica of the physical reality of energy systems and their components. We propose a definition and classification scheme for digital representations, including digital models, shadows, and twins. To provide digital representations, we investigate data-driven grey-box models that balance known and unknown system properties. In particular, we encode system characteristics into Gaussian processes, yielding hybrid models between equation-based and probabilistic machine learning models. We provide a system-specific modeling scheme and an algorithmic solution that guarantees numerical stability. We propose and analyze applications in three energy-related domains.

First, we propose a digital twin for power system data quality in large-scale monitoring platforms to detect and treat erroneous data. We propose a signal recovery technique that can recover large periods of missing data of power systems signals based on only local power systems topology. We provide information and detailed insights about key characteristics of the proposed digital twin that generate practical benefits for system operators. In particular, a self-aware failing indication allows system operators to directly evaluate the recovered data.

Second, we propose and compare four different types of digital shadows for non-invasive parameter estimation to characterize power electronics converters. We investigate and compare a forward-solver-based approach, a physics-informed neural network, and two different classes of physics-informed Gaussian processes. We discuss their differences as well as their advantages and disadvantages regarding power electronics parameter estimation. In particular, we propose a new class of physics-informed Gaussian processes and a dedicated model-building scheme that allows the application to power electronics converters in the field. By defining affine operator kernels, we show that the proposed digital shadow significantly outperforms previous alternatives when applied to power converters in the field. Moreover, we show that converter parameters can be estimated subject to the particular operational regime, yielding detailed operational insights.

Third, in the area of building operations, we propose a digital twin for the generation of pseudo-measurements usable by advanced control systems. Continuously updated by reference measurements conducted at convenient locations, the proposed digital twin provides pseudo-measurements at user-relevant locations. We provide detailed information on different types of realizations, such as model-building characteristics regarding the number of reference inputs and long-term applicability. The given algorithms can be trained on a minimal data set, which immediately enables practical applications. We show that the proposed digital twin trained on a couple of days in summer can be operated in winter.

To this end, we define digital representations, realize them through hybrid grey-box modeling using Gaussian processes, and provide explicit applications in three energy-related areas.

Kurzfassung

Mit wachsender Komplexität durch die Integration neuer Technologien und insbesondere deren Verknüpfungen, die eine gezielte Planung, Überwachung und Steuerung moderner Energiesysteme ermöglichen, erhöht sich der Bedarf an leistungsfähigen Konzepten, die detaillierte Einblicke in Energiesysteme und deren Komponenten bieten. Während klassische Modelle eine verallgemeinerte Beschreibung von Energiesystemen liefern, untersucht diese Arbeit die Verbindung zwischen der physischen und der digitalen Welt. Hierzu wird ein Definitions- und Klassifizierungsschema für digitale Darstellungen entwickelt, welches digitale Modelle, digitale Schatten und digitale Zwillinge beinhaltet. Zur Realisierung werden datengetriebene Modelle untersucht, die bekannte Systemeigenschaften beinhalten und unbekannte Systemeigenschaften lernen. Energiesystemcharakteristiken werden hierzu in Gauß-Prozesse integriert, welche eine Beschreibung zwischen gleichungsbasierten und Modellen des probabilistischen maschinellen Lernens ermöglichen. Es wird ein systemspezifisches Schema zur Modellgenerierung und eine numerisch stabile Lösung definiert. Explizite Anwendungen werden für drei energierelevante Bereiche vorgestellt und analysieren.

Zunächst wird ein digitaler Zwilling für die Datenqualität von Stromversorgungssystemen in großskaligen IT-Plattformen konstruiert und ein skalierbares Konzept zur Erfassung und Bereinigung von fehlerhaften Daten erarbeitet. Für längere Zeiträume von fehlerhafter oder fehlender Daten in Stromversorgungssystemen wird ein Signalwiederherstellungsverfahren auf der Grundlage der lokalen Netztopologie vorgestellt. Es werden detaillierte Informationen zu den algorithmischen Charakteristiken gegeben, welche unter anderem eine selbstinduzierte Evaluation beinhalten, die eine direkte Bewertung und Auswertung der wiederhergestellten Daten erlaubt.

Für die nicht-invasive Parameterbestimmung von Leistungselektronik werden vier verschiedene Arten von digitalen Schatten konstruiert. Hierfür wird ein auf einer Vorwärtslösung basierender Algorithmus, ein physikalisch-informiertes neuronales Netz, sowie zwei verschiedene Klassen physikalisch-informierter Gauß-Prozesse konstruiert und verglichen. Eine neue Klasse physikalisch-informierter Gauß-Prozesse wird mittels affiner Operatoren konstruiert und ein dediziertes Modellbildungsschema wird entwickelt, das eine Anwendung auf Leistungswandler ermöglicht, welche in realen Energiesystemen operieren. Die gegebene Konstruktion ermöglicht die Bestimmung von Leistungselektronikparametern bezüglich des jeweiligen Operationsbereiches.

Weiter wird eine Anwendung eines digitalen Zwillings im Bereich des Gebäudebetriebs für die Generierung von Pseudo-Messungen entwickelt. Die Konstruktion wird durch Referenzmessungen an herkömmlichen Orten kontinuierlich aktualisiert und liefert Pseudo-Messungen an benutzerrelevanten Positionen. Die Anzahl von der Referenzsensoren und eine langfristige Anwendbarkeit wird diskutiert. Der konstruierte digitale Zwilling kann anhand weniger Tage trainiert werden und ganzjährig betrieben werden.

Zusammenfassend werden in dieser Arbeit digitale Darstellungen definiert, durch hybride Grey-Box-Modellierung unter Verwendung von Gaußschen Prozessen realisiert und explizite Anwendungen in drei energiebezogenen Bereichen präsentiert.

Contents

Abstract iii

Kurzfassung v

Publications xiii

1 Introduction 1

2 Concepts of Digital Representations in Energy Systems 5

2.1 Digital Twins in the Energy Sector 6

2.1.1 Fields of Applications 6

2.1.2 Overview of Definitions of Digital Twins 8

2.2 Definition and Classification of Digital Representations 10

2.2.1 Concepts of Modeling 10

2.2.2 Towards a Definition of Digital Representations 12

2.2.3 Classification Scheme of Digital Representations 15

2.3 Realizations of Digital Representations 17

2.3.1 White-, Grey- and Black-Box Modeling 17

2.3.2 Gaussian Process Characteristics for Digital Representations 21

2.3.3 Road to Gaussian Processes with Multiple-Outputs 24

3 Realizations of Digital Representations by Gaussian Processes 29

3.1 Theoretical Background of Gaussian Processes 30

3.1.1 Machine Learning from a Probabilistic Perspective 30
3.1.2 Basics of Gaussian Processes 31
3.1.3 Kernels 34
3.2 Design of Gaussian Processes with Multiple Outputs 38
3.2.1 Balanced Training Procedure 38
3.2.2 Noise Hyper-Parameter in Multi-Output Structures 39
3.2.3 Regularizations of Kernel Matrices with Block Structure 40
3.2.4 Algorithmic Solution for Block-Structured Gaussian Processes 43
3.3 Linear Operator Kernel Design 43
3.3.1 Electrical Circuit Modeling 43
3.3.2 Linear Operator Kernel Design 46
3.3.3 Balancing, Noise Hyper-Parameter and Regularization of Linear Operator Kernel 48
3.4 Affine Operator Kernel Design 48
3.4.1 Affine Operator Kernel 49
3.4.2 Balancing, Noise Hyper-Parameter and Regularization of Affine Operator Kernel 50
3.4.3 Affine Operator Kernel Algorithmic Solution 51
3.5 Correlated Kernel Design 52
3.5.1 Separable Kernel for System Modeling in Gaussian Processes 52
3.5.2 Balancing, Noise Hyper-Parameter and Regularization of Correlated Kernel 53
3.6 Digital Representation Design by Multi-Output Gaussian Processes 53

4 A Digital Twin for Data Quality in Power Systems 55

4.1 Introduction 56
4.1.1 Motivation 56
4.1.2 Related Work 56
4.1.3 Preliminary Remarks on Technical Realizations 58
4.1.4 Contribution 58
4.2 ICT Platforms for Energy Systems 59
4.2.1 The ICT Platform of the Living Lab Energy Campus 59
4.2.2 Power system at FZJ 59
4.3 Correlated Gaussian Processes for Signal Recovery in Power Systems 59

4.3.1 Gaussian Process Characteristics for Signal Recovery 60
4.3.2 Single-Channel Model for Voltage and Current Magnitudes 60
4.3.3 Multi-Channel Model for Correlated Voltage and Current Magnitudes . . 61
4.3.4 Multi-Channel Recovered Signals 62
4.4 Case Study - Living Lab Energy Campus 63
4.5 Results of Correlated Gaussian Process as Digital Twins for Signal Recovery . . 65
4.5.1 Accuracy, Precision and Basic Characteristics 65
4.5.2 Missing Interval Independent Recovery Accuracy 67
4.5.3 Cross-Correlation Identification 68
4.5.4 Self-Aware Failing Indication 69
4.5.5 Modeling of Multi Cross-Correlated Signals 70
4.6 Online Application of a Digital Twin for Data Quality in Power Systems ICT Platforms 71
4.6.1 Digital Twin Framework in ICT Platforms 72
4.6.2 Modules in ICT Framework 72
4.7 Conclusion and Characterization of Power System Digital Twins 75
4.7.1 Discussion 75
4.7.2 Characterization of Power Systems Data Quality Digital Twins 76
4.7.3 Outlook 77

5 A Digital Shadow for Power Electronics Parameter Estimation 79

5.1 Introduction 80
5.1.1 Motivation 80
5.1.2 Related Work 80
5.1.3 Preliminary Remarks on Technical Realizations 81
5.1.4 Contribution 82
5.2 Power Electronics Parameter Estimation - Basic Modeling and Theoretical Insights 83
5.2.1 LC Model for Buck Converter Parameter Estimation 83
5.2.2 Parameter Estimation by Linear Operator Gaussian Processes 86
5.2.3 Parameter Estimation by Lightweight Physics-Informed Neural Networks 89
5.2.4 Parameter Estimation by Automatic Differentiation Shooting 91
5.2.5 Case Study: LC - Buck Converter 92

5.2.6 Results . . . 93
5.3 Power Electronics Parameter Estimation in the Field . . . 96
5.3.1 Modeling of Power Converters in the Field . . . 96
5.3.2 A Digital Shadow by Affine Physics-Informed Gaussian Processes . . . 96
5.3.3 Case Study: Power Electronic Converters in the Field . . . 100
5.3.4 Results . . . 102
5.4 Conclusion and Characterization of Power Electronics Digital Shadows . . . 107
5.4.1 Discussion . . . 107
5.4.2 Characterization of Power Electronics Digital Shadows . . . 108
5.4.3 Outlook . . . 109

6 A Digital Twin for Building Pseudo-Measurements 111
6.1 Introduction . . . 112
6.1.1 Motivation . . . 112
6.1.2 Related Work . . . 112
6.1.3 Preliminary Remarks on Technical Realizations . . . 113
6.1.4 Contribution . . . 113
6.2 Pseudo-Measurements by Correlated Gaussian Processes . . . 114
6.2.1 Technical Conceptualization of Building Digital Twins . . . 114
6.2.2 Technical Realization of Building Digital Twins . . . 115
6.2.3 Trivial and VARMA Realizations . . . 117
6.3 Case Study: JuLab . . . 118
6.4 Implementation Framework . . . 119
6.5 Results . . . 120
6.5.1 Temperature Digital Twins . . . 121
6.5.2 CO_2 Digital Twins . . . 125
6.6 Conclusion and Characteristics of Building Digital Twins . . . 125
6.6.1 Discussion . . . 126
6.6.2 Characterization of Building Digital Twins . . . 127
6.6.3 Outlook . . . 129

7 Conclusion 131

A Appendix **135**

Acronyms **141**

List of Symbols **143**

List of Figures **147**

List of Tables **149**

Acknowledgments **151**

Publications

Journal Paper

[1] **M. Zimmer**, M. Milton, D. Cucak, T. Pesch, H. Ginn, and A. Benigni, "Parameter estimation of power electronic converter by affine physics-informed gaussian processes", *IEEE Transactions on Power Electronics*, pp. 1–13, 2026

Contribution: M. Zimmer provided the original idea of the proposed solution. The proposed application was designed by A. Benigni. The main implementation, test, and results of the proposed algorithm and the corresponding alternatives are provided by M. Zimmer. The specific case study was designed by A. Benigni, M. Milton, D. Cujak, and H. Ginn. The corresponding data was provided by M. Milton, D. Cujak under the supervision of H. Ginn. The publication was written by M. Zimmer with contributions from M. Milton. A. Benigni supervised the project. A. Benigni and H. Ginn reviewed, edited, and approved the manuscript for possible publication.

[2] **M. Zimmer**, M. Buechel, F. Redder, M. Mork, T. Pesch, A. Xhonneux, D. Müller, and A. Benigni, "Digital twins for building pseudo-measurements", *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–10, 2025. DOI: 10.1109/TIM.2025.3527590

Contribution: M. Zimmer provided the original idea of the proposed solution to the proposed application. The main implementation, test, and results of the proposed algorithm and the corresponding alternatives are provided by M. Buechel under the supervision of M. Zimmer. The specific case study was designed by M. Zimmer. The corresponding data was provided by F. Redder. The publication was written by M. Zimmer with contributions from F. Redder and M. Buechel. A. Benigni supervised the project. A. Benigni and A. Xhonneux reviewed, edited, and approved the manuscript for possible publication.

[3] **M. Zimmer**, D. Carta, T. Pesch, and A. Benigni, "Signal recovery in power systems by correlated gaussian processes", *IEEE Open Journal of the Industrial Electronics Society*, vol. 5, pp. 692–703, 2024. DOI: 10.1109/OJIES.2024.3423405

Contribution: M. Zimmer provided the original idea of the proposed solution to the proposed application. The main implementation, test, and results of the proposed algorithm and the corresponding alternatives are provided by M. Zimmer with minor contributions from D. Carta. The specific case study was designed and the corresponding data was provided by D. Carta. The main part was written by M. Zimmer with contributions from D. Carta. A. Benigni supervised the project. A. Benigni reviewed, edited, and approved the manuscript for possible publication.

[4] M. Wenzel, E. De Din, **M. Zimmer**, and A. Benigni, "Gaussian process supported stochastic mpc for distribution grids", *IEEE Open Journal of Control Systems*, vol. 4, pp. 332–348, 2025. DOI: 10.1109/OJCSYS.2025.3601836

Contribution: M. Wenzel and E. De Din provided the original idea of the proposed solution to the proposed application. The main implementation, test, and results of the proposed algorithm and the corresponding alternatives are provided by M. Wenzel under the supervision of E. De Din. M. Zimmer contributed to the model selection of the proposed algorithm. The specific case study was designed, and the corresponding data were provided by M. Wenzel. The main part was written by M. Wenzel with contributions from M. Zimmer and E. De Din. A. Benigni supervised the project.

Peer-Reviewed Conference Paper

[5] **M. Zimmer**, E. De Din, D. Carta, and A. Benigni, "Power electronics parameter estimation by physics-informed gaussian processes", in *IECON 2025 – 51st Annual Conference of the IEEE Industrial Electronics Society*, 2025, pp. 1–7

Contribution: M. Zimmer provided the original idea of the proposed solution to the proposed application. The main implementation, test, and results of the proposed algorithm and the corresponding alternatives are provided by M. Zimmer. The specific case study was designed, and the corresponding data were provided by M. Zimmer. The publication was written by M. Zimmer. A. Benigni supervised the project.

[6] **M. Zimmer**, J. Augustin, T. Pesch, and A. Benigni, "Parameter estimation of power electronics by forward and backward solutions", in *IECON 2024 - 50th Annual Conference of the IEEE Industrial Electronics Society*, 2024, pp. 1–7. DOI: 10.1109/IECON55916.2024.10905957

Contribution: M. Zimmer provided the original idea of the proposed solution to the proposed application. The main implementation, test, and results of the proposed algorithm and the corresponding alternatives are provided by J. Augustin under the supervision of M. Zimmer. The specific case study and the corresponding data were provided by M. Zimmer. The publication was written by M. Zimmer. A. Benigni supervised the project. A. Benigni reviewed, edited, and approved the manuscript for possible publication. The paper was presented by M. Zimmer.

[7] **M. Zimmer**, M. Buechel, F. Redder, M. Mork, T. Pesch, A. Xhonneux, D. Müller, and A. Benigni, "Building digital twins for thermal pseudo-measurements generation", in *2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2024, pp. 1–6. DOI: 10.1109/I2MTC60896.2024.10561057

Contribution: M. Zimmer provided the original idea of the proposed solution to the proposed application. The main implementation, test, and results of the proposed algorithm and the corresponding alternatives are provided by M. Buechel under the supervision of M. Zimmer. The specific case study was designed by M. Zimmer. The corresponding data was provided by F. Redder. The publication was written by M. Zimmer with contributions from F. Redder and M. Buechel. A. Benigni supervised the project. A. Benigni and A. Xhonneux reviewed, edited, and approved the manuscript for possible publication. The paper was presented by M. Zimmer.

[8] D. Carta, **M. Zimmer**, T. Pesch, and A. Benigni, "Bad Data Detection and Handling in ICT Platforms for Energy Systems", in *2022 IEEE 12th International Workshop on*

Applied Measurements for Power Systems (AMPS), Sep. 2022, pp. 1–6. DOI: 10.1109/AMPS55790.2022.9978880

Contribution: M. Zimmer and D. Carta equally contributed to the original idea of the proposed solution for the proposed application. The main concept, test, and results of the proposed scheme were contributed by M. Zimmer and D. Carta with equal contributions. The main part was written by D. Carta with contributions from M. Zimmer. A. Benigni and T. Pesch reviewed and approved the manuscript for possible publication. The paper was presented by D. Carta.

[9] **M. Zimmer**, T. Pesch, and A. Benigni, "Time-Series Analysis and Forecasting of Power Consumption using Gaussian Process Regression", in *2021 International Conference on Smart Energy Systems and Technologies (SEST)*, Sep. 2021, pp. 1–6. DOI: 10.1109/SEST50973.2021.9543253

Contribution: M. Zimmer provided the original idea of the proposed solution to the proposed application. The main implementation, test, and results of the proposed algorithm and the corresponding alternatives are provided by M. Zimmer. The specific case study was designed, and the corresponding data were provided by M. Zimmer. The publication was written by M. Zimmer. A. Benigni supervised the project. T. Pesch reviewed and approved the manuscript for possible publication. The paper was presented by M. Zimmer.

[10] M. Mork, F. Redder, **M. Zimmer**, M. Buechel, T. Pesch, A. Xhonneux, D. Müller, and A. Benigni, "Thermal pseudo-measurements for model predictive control in buildings", preprint

Contribution: M. Zimmer, M. Mork, and F. Redder provided the original idea of the proposed solution to the proposed application. The main implementation, test, and results of the proposed algorithm and the corresponding alternatives are provided by M. Mork and M. Buechel. The specific case study was designed, and the corresponding data were provided by M. Mork. The preprint was written by M. Mork with contributions from F. Redder and M. Zimmer.

Supervised Student Thesis

[11] M. Buechel, "Building digital twins for energy savings by correlated gaussian processes", Technische Universität Dortmund, M.S. thesis, Technische Universität Dortmund, 2024

[12] J. C. Augustin, "Improved machine learning methods for parameter estimation of ordinary differential equations", Technische Universität Berlin, M.S. thesis, 2023

[13] R. Owens, "Rolling horizon forecasting of time series energy demand using long short-term memory neural networks", Hochschule Darmstadt, M.S. thesis, Hochschule Darmstadt, 2022

[14] K. Stein, "Untersuchung verschiedener lstm-netzwerkarchitekturen zur anomalieerkennung in energieverbrauchsdaten", Hochschule Emden Leer, M.S. thesis, Hochschule Emden Leer, 2022

1

Introduction

Since its first introduction in 2003, the concept of digital twins has attracted a lot of attention in various fields of applications such as power systems and their corresponding components, transportation, smart cities, manufacturing, industrial applications, life science, healthcare, and medicine [15], [16], [17], [18]. Digital twins changed the paradigm of modeling in these fields significantly. While the attempt to provide insights about a specific physical object is not new, the systematic attempt originating in the concept of digital twins highlights the importance of object-specific modeling. The key distinction between a classical model and a digital twin is the tailoring to a particular physical object. A classical model describes a generic or typical version of a physical object utilizing a system's description in terms of equations [19]. Correspondingly, generic or typical physical effects can be described. Contrarily, a digital twin provides a replica of a particular physical object, characterized by its individual specifications. The key idea is that a digital twin provides insights about a single, distinguishable object and not about any other. As such, questions about the current or future status of a particular object or process can be posed, possibly under changing internal or environmental conditions. This provides new types of information to stakeholders, enabling enhanced decision-making. This might be, for example, the decision on possible pathways in manufacturing processes, live monitoring of systems for operational safety and maintenance, or dedicated control schemes capable of adapting to changing operational conditions [20], [21], [22].

The development and potential of digital twins are mainly driven by technological development, such as increasing computational resources and the growing influence of machine learning and big data analysis [17]. While technological aspects of enabling technologies, such as the Internet-of-Things (IoT), advanced IT infrastructures, cyber-security aspects, etc., are intensively addressed in the literature [23], there is no clear, unique, well-defined, and commonly used concept of a digital twin. In Ch. 2 we address this research gap explicitly and propose a definition as well as a classification scheme. More specifically, we propose a more general class of models,

called digital representations, which consists of digital models, digital shadows, and digital twins.

In the realization of digital representations, data-driven modeling plays a key role as it serves the need to include information about a particular physical object. At the same time, it is also evident that not every aspect of a physical object is known or accessible in practical applications. Thus, grey-box modeling, i.e., cases where something is known about a physical object but not necessarily everything, is of major interest. Particular models serving as digital representations might range from highly aggregated to very detailed representations. While in some applications, pure information by a binary classification algorithm on the status of the system might be sufficient, in others, a detailed replica of the physical object might be needed. In the latter case, regression models serve as suitable algorithms and are the main object of investigation in this thesis.

As a compelling class of algorithms in this regard, in Ch. 3 we propose Gaussian processes (GPs) for data-driven grey-box realizations of digital representations. A GP is a supervised probabilistic machine learning regression algorithm [24]. GPs can be seen as a generalization of Gaussian distributions, yielding an extension to function space [25]. Defined by a mean function and a kernel function, a GP is a stochastic process based on conditional probability [26]. Conditioned on a training data set, a GP provides a posterior mean function and a posterior covariance function. The choice of the priors, i.e., the selection of mean function and kernel function, defines the type of function utilized by the GP.

Standard GPs yields representations of individual functions [26]. To realize digital representations of physical objects via a grey-box description, multiple functions and their relative behavior must be considered. For the corresponding GPs this implies a multi-output structure. While these types of GPs yield the desired models with valuable characteristics, the model-building and the algorithmic realization are non-trivial. In Ch. 3, we develop a dedicated model-building scheme and propose a dedicated algorithmic solution for numerically stable computations. While some theoretical and practical effort is needed to construct the proposed models as given in Ch. 3, the underlying issue inherent to this type of GPs can be intuitively understood as follows. In general, kernels defining GP are numerically not optimal. This is due to the simple fact that there are infinitely many functions describing a specific data set. A related effect can be observed in the solution of a set of underdetermined algebraic equations. In the absence of sufficient constraints by the set of equations, infinitely many solutions exist, resulting in a singular, non-invertible matrix. Similarly, when providing finitely many data points to a GP, finitely many constraints are provided, which is not sufficient to fully determine a function consisting of infinitely many data points. For kernel matrices defining the GP, this is established in terms of high values of condition numbers. The condition number measures the kernel's spectrum, i.e., the range of the set of eigenvalues, and indicates the numerical stability of the corresponding algorithms [27]. Summarizing, kernels of GP tend to have a broad spectrum, i.e., high condition numbers, and consequently tend to be non-trivial numerically. In this work, the proposed kernels leading to models with multiple outputs are characterized by a matrix block structure, which is induced by the underlying digital representation. Thus, the individual blocks are not independent of each other but have a common origin linked by the physical system's description. Consequently, the sub-blocks of the kernel have a similar spectrum except for a shift by parameters describing the system interactions. Thus, the multi-output spectrum is significantly broadened compared to the single-output spectrum, leading to serious numerical issues. We propose a general solution

to this issue in Ch. 3.

We propose, investigate, and discuss explicit realizations of digital representations based on data collected in the field for three different domains in the energy sector, i.e., power systems, power electronics, and building operations. With data obtained at the Living Lab Energy Campus (LLEC), a real-life laboratory located at the campus of Forschungszentrum Jülich (FZJ), we address power system signals based on local grid topology and temperature as well as CO_2 signals obtained in building operations. A full characterization of power electronics converters is provided based on data collected at the Energy Routing Lab at the University of South Carolina. The three applications of digital representations can be described as follows.

In modern large-scale power systems Information and Communication Technology (ICT) platforms, data quality plays an important role in monitoring and operation [28], [29]. Especially with the growing size of these systems and corresponding ICT structures, malfunctioning devices and data storing issues might affect reliable systems' operation [30], [31]. Addressing this issue in Ch. 4, we provide a digital twin for data quality in power systems. First, we propose a bad data detection and handling framework applicable in large-scale ICT platforms. The proposed tool can be easily applied to platforms with different structures to perform online and offline analysis aimed at monitoring the data quality, recovering bad data, and investigating the proper operation of the platform's components. Second, we propose a signal recovery algorithm capable of dealing with large missing periods of power system signals. Based on only local power system topology, the presented Correlated Gaussian Process (Corr-GP) algorithms combine cross-channel information of the considered signals with a GP, leading to an algorithm for the recovery of missing intervals in power system signals. We demonstrate the use of the proposed approach for recovering distribution grid signals obtained within the LLEC. We evaluate the performance of the Corr-GP compared to that of other state-of-the-art techniques. In addition to outperformance in terms of recovery accuracy, it is explained when and how the accuracy of the reconstructed signal is independent of the missing interval length. Finally, detailed insights about key characteristics of the proposed approach that generate practical benefits for system operators are provided. A self-aware failing indication allowing system operators a direct evaluation of the recovered data and enabling further improvement of the proposed approach is presented, as well as the detailed description of online operation in dedicated ICT platforms.

Detailed knowledge of converter parameters is crucial in modern power electronics applications, ranging from condition monitoring to control applications [32], [33], [34]. In short, dedicated knowledge about device characteristics, possibly under changing operational conditions, yields enhanced and reliable power electronics applications [35]. In Ch. 5, we propose and investigate digital shadows for power electronics parameter estimation, allowing non-invasive extraction of converter parameters. First, for three types of algorithms, we propose a model-building and training procedure enabling an application in the field of power electronics. We propose a power electronics-specific model-building scheme for Physics-Informed Gaussian Processes (PIGPs) based on linear operators, we provide lightweight Physics-Informed Neural Networks (PINNs) with a dedicated training strategy, and we propose a forward solver-based algorithm, called Automatic Differentiation Shooting (ADShooting), that can be applied to power electronics converters. In particular, for all three types of digital shadows, we present dedicated training and noise-handling strategies, both being major aspects of the given application. Exemplary, we apply both approaches for power electronics parameter estimation of a buck converter in a sim-

ulation environment. We investigate the dependence of sampling rates, as well as, the influence of noise. Second, we propose a new class of kernels, namely, affine operator kernels applicable in PIGP. This enables the application to power electronics in the field, allowing the identification of any physical parameter of interest. We define a dedicated noise-handling scheme that allows outperforming alternative algorithms when applied to hardware tests. We analyze the numerical stability of the presented approach and define a self-consistent normalization scheme allowing a reliable application. As a particular feature of the proposed affine operator kernels, we show that the algorithm is capable of estimating converter parameters for specific operational conditions. The proposed approach is theoretically discussed and experimentally verified with data obtained from a half-bridge converter in the field.

Indoor air quality and thermal comfort are directly related to occupant well-being [36], [37]. However, building control architectures are strongly limited by the systematic lack of measurements at user-relevant locations [38]. We address this issue in Ch. 6 by proposing a digital twin for building pseudo-measurement generation. Tested with thermal and CO_2 measurements collected from the field, close-to-person pseudo-measurements are provided based on the continuous input of remotely located measurement signals. We show that the proposed Corr-GP approach is trainable on only a few days of measurements. This property is especially useful in field applications, where alternative algorithms, such as, for example, Artificial Neural Networks (ANNs) architectures, are not capable of dealing with small amounts of data [39]. Within the given framework, we show how to utilize the proposed digital twin to couple multiple reference sensors to provide close-to-person pseudo-measurements. By extending the Corr-GP approach to a non-zero prior mean formulation, we show how to reduce the included information by the reference sensors. More precisely, the extended approach can be defined as a digital twin with only a single reference sensor. This enables a reliable long-term application by avoiding the need for retraining caused by changing seasonalities within the signal characteristics. That is, we show that the digital twin trained in summer can be operated in winter.

Scientific Contribution The overall scientific contributions of this thesis can be summarised as follows. In Ch. 2, we propose a definition and classification scheme for digital representation, including digital models, digital shadows, and digital twins. In Ch. 3 we summarize the technical contributions to the algorithmic realization of digital representations provided in [1], [2], [3], [5], [6], [7]. In particular, we propose a dedicated model-building scheme as well as a numerically stable algorithm. Published in [3], [8], in Ch. 4, we propose a bad data detection and handling framework for power system ICT platforms and propose a Corr-GP signal recovery algorithm for the recovery of large missing periods of power system signals based on local grid topology. Provided in [1], [5], [6], we propose and investigate digital shadows for power electronics parameter estimation in Ch. 5. Published in [2], [7], we propose a digital twin providing pseudo-measurements at user-relevant positions for building operations in Ch. 6. The outcomes of [4], [9], [10] are not explicitly addressed in this thesis, but are not independent either.

2

Concepts of Digital Representations in Energy Systems

In this chapter, we discuss the concepts, definitions, and applications of digital representations applied to energy systems. In Sec. 2.1, we provide an overview of existing applications in various fields related to the energy sector. We then analyze and show the limitations of existing definitions. In Sec. 2.2, we propose an extended universal definition and a classification scheme of digital representations relevant to this work. In particular, the concepts of digital shadows and digital twins are discussed. This allows us to analyze and classify the developed models in this work within a well-defined framework. In Sec. 2.3, we provide possible technical realizations from a general perspective. We discuss the requirements of digital representations specified by the given definition and provide a high-level discussion on possible algorithmic realizations. To conclude this chapter, we motivate the usage of algorithms based on GPs for the realization of digital representations.

2.1 Digital Twins in the Energy Sector

The concept of digital twins spreads across different fields of application, such as, but not limited to, power systems and their corresponding components, transportation, smart cities, manufacturing, industrial applications, life science, healthcare, and medicine. We focus on applications related to energy systems and review part of the existing concepts and use cases. As will be evident during this discussion, there is no clear, commonly agreed definition of digital twins in the literature, which constitutes the major research gap within the theoretical foundations of digital twins. Sometimes, concepts called digital companion, digital shadow, digital model, or digital version are used synonymously. In this section, we refer to all of these models as digital twins which is the leading term commonly used in literature. In the next section, we provide a more nuanced exploration.

2.1.1 Fields of Applications

In 2003, the concept of digital twins was first proposed in [18] for product lifecycle management. The key idea can be stated as follows:

> "... PLM [product lifecycle management] is an attempt to de-silo information by function and create autonomous information representations of all physical objects that will be available to all functions. All changes to the object during its life will be collected and associated with the object, and everyone who needs to work with the object will have access to the information about it." [18]

Nine years later, the first attempt to give a definition of digital twins for NASA and U.S. Air Force vehicles was given by [40], where the authors define:

> "A Digital twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin." [40]

Since then, the notion of a digital twin has appeared in various fields. Within the electrical domain, digital twin applications range from single components over local energy communities to the system level such as transmission and distribution level.

Electrical Components

Digital twins of electrical components ranging from power electronics applications to modeling of systems like wind turbines or photovoltaic (PV) systems. In [41], a survey on digital twins for power electronics-based energy conversion systems is presented. Power converter parameters are determined by digital twinning in [42]. More precisely, initially fixed parameter values of the power converter turn into probabilistic functions via polynomial chaos expansion yielding a real-time probabilistic digital twin with stochastic parameter characteristics. To address critical

issues relevant to reliable applications of offshore wind farms, the remaining lifetime of power converters is captured by a digital twin in [43]. In [44], a digital twin for the fault diagnosis of PV energy conversion units is proposed. A residual error vector indicates a deviation between the measured output and a model-based estimate. Utilized as health indicators, a particle swarm algorithm is used to estimate the circuit parameters of DC-DC converters in [45]. With the same purpose but based on PINN, in [46] parameters of a DC-DC converter are estimated. In [47], a digital twin approach by a temporal convolution neural network combined with a k-nearest neighbor algorithm for predictive modeling of wind turbines is proposed.

Microgrids

We summarize digital twins for medium-sized power grids. This includes connected and isolated microgrids, local energy communities as well as wind farms and PV systems. Similar to [47] but applied at the system level, in [48] a physics-informed deep learning algorithm is proposed to model the wind farm flow system. The review [49] provides an overview of the operation and maintenance of offshore wind farms using digital twins. In [50], low voltage networks are enhanced by a neural network-based digital twin that provides voltage calculations from smart meter data. In [51], a digital twin evaluation for non-interconnected island power systems as present in Greece is presented. Due to a high penetration of renewable energy sources, these isolated systems benefit from modern digital twin applications as reported in [51]. In [52], challenges and solutions according to big data aspects regarding wind farms are discussed.

Distribution and Transmission Grids

A survey on digital twins in distribution grids with a focus on a real-time application can be found in [53]. In [54], the potential of digital twins in integrated energy systems is discussed. The role of open source for building digital twins for the European energy infrastructure is provided in [55]. In [56], loss measurements of ultra-high voltage direct current systems are addressed by digital twins. A digital twin for the temperature distribution of a high voltage cable is constructed in [57] using a conditional generative adversarial network. In [22], the adaptation of digital twin technology for applications in power system control centers is discussed. Dynamic system observation based on time-synchronized Phasor Measurement Units (PMUs) can be archived by a so-called, dynamic digital mirror.

Transportation

In the field of transportation, digital twins are applied to the components of single vehicles and large-scale traffic structures. A review on digital twins for traffic safety and mobility can be found in [16]. In [58], digital twins of electric drive trains are considered to identify unbalanced rotor system states and for the optimization of run-up routines. For motorways, a digital twin for a continuously synchronized model is proposed with a focus on real-time data integration [59]. In [60], a data-driven model handling crowd-sourced traffic data imputation is given. Beyond modeling of physical entities, the driver behavior is modeled by a digital twin in [61].

Smart Cities

The concept of smart cities combines individual fields of city planning and management with a focus on transportation systems, microgrids, and power grids. A review on digital twins of smart cities can be found in [62]. In [63], the possible impact of digital twin applications within smart cities is discussed. The concept of digital twin cities is proposed and discussed in [64]. In [65], the concept of a city's state by spatiotemporal fluctuations integrated into an analytics platform is discussed to address the human-infrastructure-technology interactions. The management of a decentralized integrated natural gas-electric system by digital twin application is given in [66].

Other fields of Applications

A comprehensive overview of digital twins applied in the energy sector can be found in [15]. In [67], a survey of digital twins for networks can be found. In particular, [67] provides the notion of digital twin networks, which will also be discussed below. A cross-domain review with a focus on software engineering is provided in [68]. A discussion on big data aspects in manufacturing and industry enabling digital twin technologies can be found in [21]. In [69], integrated and intelligent manufacturing including its possible potential is presented. A so-called digital companion for industry applications is discussed in [70]. For product lifecycle management an application framework for digital twins is considered in [71]. In [20], fault diagnosis by deep transfer learning is presented to address maintenance aspects of shop floors during operation. The potential of distributed digital twins compared to a centric approach is discussed in [72]. In [73], a human-robot collaborative assembly is discussed. The latter highlights the potential of digital twin technologies to merge two or more seemingly distinct types of physical systems.

2.1.2 Overview of Definitions of Digital Twins

There are widespread applications of digital twins in different fields of application, across different scales, capturing different levels of detail. One challenge within the field of digital twins is the lack of a proper, universal definition [62], [74]. In this section, we review some common definitions provided in the literature and discuss their potential regarding a universal understanding of the, up to now, vague concept of a digital twin.

The initial definition provided by NASA focuses on as-build vehicles or systems [40]. Demanding in the definition of a digital twin a multiphysics, multiscale, and probabilistic component limits the possible technical realizations significantly, while not being precise at the same time. For example, multiphysics or multiscaling do not restrict the particular model in any practical way as these terms can be interpreted very differently depending on the particular context. At the same time, a lot of possible models are ruled out if they do not include multiple physical effects or multiple scales. To this end, according to the definition of [40] it is easy to exclude models from being a digital twin while not providing a practical definition of digital twins in a useful way.

In [75], within the context of electric power systems, a digital twin is defined as

> "A digital twin virtually and dynamically represents a physical asset, system, or process in the digital world. This digital counterpart mirrors the real-world entity in its behavior, characteristics, and interactions by utilizing real-time data and simulations. Essentially, it serves as a bridge between the physical and digital worlds, offering a comprehensive view of the asset's performance and functionality." [75]

This definition captures the key ideas behind the concept of digital twins. However, there are the following drawbacks to the explicit statement. The requirement of mutually capturing behavior, characteristics, and interactions is challenging to achieve in general applications. Moreover, each requirement depends on the system under consideration and is thus not suitable for a universal characterization. Depending on the interpretation of the term "simulation", cf. [19], the given definition excludes pure data-driven models.

Digital twins in though-life engineering services are defined by

> "A digital twin is a digital representation of a physical item or assembly using integrated simulations and service data. The digital representation holds information from multiple sources across the product life cycle. This information is continuously updated and is visualised in a variety of ways to predict current and future conditions, in both design and operational environments, to enhance decision making." [76]

In connection with the definitions presented in [40], [75], demanding simulations, information from multiple sources which are continuously updated and visualized in a variety of ways to predict current and future conditions is barely met by practical models in general.

On the other side of the spectrum, also purified definitions exist. In the manufacturing context, a digital twin is described as

> "A digital twin is a computerized model of a physical device or system that represents all functional features and links with the working elements." [69]

as well as

> "A Digital twin is a set of virtual information that fully describes a potential or actual physical production from the micro atomic level to the macro geometrical level." [71]

For health indication of power converters, one might define

> "A Digital Twin is a virtual representation of a physical object that shares the same characteristics as its physical counterpart." [45]

Within these definitions, the communication between the physical and the digital part does not play a major role. Moreover, the demand to represent all functional features can be barely met from a theoretical as well as practical perspective.

On top of the definition of a pure digital twin, new concepts emerging from that like, for example, digital twin networks [67] and distributed digital twins [72]. These concepts should not

exist as stand-alone entities but rather be a certain characteristic within the universally defined digital twin.

To this end, no universally applicable definition of the concept of digital twins exists. Previous definitions are too narrow, thus excluding many possible digital twin realizations, or too vague to provide a careful distinction of other types of models. Moreover, related concepts are treated separately instead of being considered inside a universal, conceptual paradigm.

2.2 Definition and Classification of Digital Representations

There is a need for a proper and universally applicable definition of the concept of digital twins. Due to the vast amount of applications within distinct fields and the diverse individual characteristics of each model, referred to as a "digital twin", providing a universal definition is challenging.

For proper analysis and discussion in the discourse of this work, we propose a concept of digital representations, which constitutes digital models, digital shadows, and digital twins. At this point, it should be noted that we semantically switch from "digital twin" to "digital representation" to summarize the concepts presented in Sec. 2.1. For now, "digital representation" should be taken as a working terminology, containing everything that is representing something digitally. A precise meaning of digital representations will be evident after the concepts of digital models, digital shadows, and digital twins have been developed.

The proposed concept should be regarded within the material presented in this work. We do not attempt to provide a final and universal definition valid in all fields of application. However, as we provide a systematic approach, the proposed concept might yield a starting point for an attempt to provide more universal considerations.

Current definitions provide very narrow or too broad notions of the concept of "digital twins" as evident from the review provided in Sec. 2.1.2. The former case provides concepts only valid in a specific domain, limiting comparability, analysis, and discussion of distinct manifestations of digital representations. In the latter case, anything could be called a "digital twin", thus, limiting the explanatory power. In both cases, a precise understanding is prevented. To address the widespread and diverse usage of the term "digital twin", we propose a change of paradigm. We propose a streamlined definition combined with a classification scheme. As such, we can allocate specific realizations within the class of digital representations and provide attributes for a better understanding of each individual model.

2.2.1 Concepts of Modeling

To introduce and clarify the terminology used in modeling and simulation science, we give a short review of the concept of modeling.

System In this work, we follow the definition provided by [77] of a *system*, which states

> "A system is what is distinguished as a system." [77]

Although this appears redundant at first sight, it captures the key insights. A system is a matter of definition as reality is too complex to be addressed in all of its facets simultaneously. Thus, we distinguish between physical reality and the abstract notion of a system, capturing aspects of interest in physical reality. For a detailed discussion on this and a survey of possible alternative definitions of systems, we refer to [19]. Not explicitly stressed in the definition above but the key aspect implicitly present is the ability to distinguish between a system and, loosely speaking, the rest. In other words, a border can be drawn between what belongs to a system and what does not. Note that this does not exclude systems connected to other systems. Accordingly, the definition is not as broad as it might seem at first sight. On one hand, this implies a relative structure of systems. On the other hand, it provides flexibility on the level of detail captured by the system of choice. This is in line with the discussion in [78] on a system as a choice of a list of variables describing an object. We summarize these important aspects:

i) A system is a relative entity that can be distinguished from its environment.

ii) A system has an inherent notion of the level of detail.

For pure convenience, we might specify the relative notion of a system. We call a system a *subsystem* if it is a subset of another system. We call a system *connected* if it shares parts with another system, and *isolated* otherwise.

Experiments According to [19], an experiment is defined as

> "An experiment is the process of extracting data from a system by exerting it through its inputs."[19]

Model To define a *model*, we follow the lines of [79].

> "To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A." [79]

Within simulation science, one can distinguish between qualitative and quantitative models [19], [80]. A survey on qualitative models can be found in [80]. In [19], the main focus is on quantitative models that can be further distinguished into time-continuous and time-discrete modeling. We want to emphasize that the latter distinction significantly influences the model building provided in this work and is further discussed in Sec. 2.3.3.1.

Types of Data In the course of this work, it is important to distinguish between different types of data. Unfortunately, to our knowledge, there is no satisfying, universal, and precise definition in the literature. Although distinguishing specific types of data is important for the definition of digital representations, a contribution to providing a precise notion in this regard is beyond the scope of this work.

In [19], the author distinguishes between quantitative and qualitative models. Contrarily to [19], we assign the attributes qualitative and quantitative to the data instead of the model. This equivalent point of view fits better within the realm of data-driven modeling. We call data *quantitative* if it is obtained by an experiment in the sense of the given definition. This might be in the form of time-series data such as, for example, voltage, current, and temperature measurements. Data about the system itself is called *qualitative*. This might be in terms of, for example, mathematical equations describing the system or other system's attributes. This includes meta-data associated with qualitative data obtained by an experiment. It should be noted that it is possible to provide certain system attributes without associating them with measured quantities. Thus, using meta-data synonymously with qualitative data is not appropriate. This kind of distinction between qualitative and quantitative data is shared in the field of data analytics [81]. We want to emphasize that the term quantitative data does not induce a rating about the quality of the data. We call a model a *data-driven model* if there exists a quantitative data transfer.

2.2.2 Towards a Definition of Digital Representations

The concept of digital representations can be seen as a subclass of general modeling discussed in Sec. 2.2.1. We start with specifying the previously defined quantities in the given context.

In this work, the systems of interest are based on physical items. More precisely, we refer to real-world objects as *physical items*, i.e., things that exist in reality. We call the system defined for a physical item the *physical object*, denoted by Physical object ($\mathcal{P}$). As pointed out in the discussion of the concept of a system above, we have to distinguish between a physical item and what we define as the system representing this physical item. According to the given definition, we refer to a system as an abstract object representing parts of physical reality. To be precise, we use the term physical object to refer to the system of the physical item, not the physical item itself. Within digital twin literature, what we define as the physical object $\mathcal{P}$ is commonly referred to as the "physical twin" [17]. Thus, referring to $\mathcal{P}$ as the "physical representation" immediately suggests itself. However, this might lead to a misleading conceptualization of digital representations as discussed below. To circumvent this, we go with a slight abuse of notation at this point and use the term "physical object". To improve readability in the following we refrain from being too precise and use "physical object" for $\mathcal{P}$ as well as for the physical item. The individual context implies the meaning. Contrarily, $\mathcal{P}$ is exclusively used as defined above. We give the definition of a digital model in (2.I).

2.I. *Definition of a Digital Model*

We call the subclass $\mathcal{D}$ of all models for a given system $\mathcal{P}$ that is provided by a computer program a digital model.

It should be noted that (2.I) holds for general systems, not only for those representing physical items $\mathcal{P}$. In this work, only the subclass $\mathcal{P}$ is of relevance. We want to emphasize, that by definition (2.I) a digital model consists of the tuple

$$(\mathcal{P}, \mathcal{D}) \tag{2.1}$$

As a matter of fact, this is true for all models as there is no relevance of a model without a corresponding system. This aspect is also reflected in the definition of a digital twin provided by [51].

> "A DT [Digital Twin] consists of three essential elements: a tangible object, its corresponding virtual representation, and the vital data link that enables seamless communication between these two." [51]

In this definition, the data linking the physical and the digital object is put into the center. The data exchange between $\mathcal{P}$ and $\mathcal{D}$ is a key aspect when it comes to the concept of digital representations. Thus, the tuple (2.1) extends to

$$\left(\mathcal{P}, \mathcal{D}, \mathcal{P} \xleftrightarrow[\text{transfer}]{\text{information}} \mathcal{D}\right) \tag{2.2}$$

At this point, we used the loose term "information transfer" to indicate the need for clarification. Although [51] put the data link into the center of the definition, no details or specifications are provided. In [17], it is stated that

> "A digital model is described as a digital version of a preexisting or planned physical object, to correctly define a digital model there is to be no automatic data exchange between the physical model and digital model. [...]
>
> A digital shadow is a digital representation of an object that has a one-way flow between the physical and digital object. A change in the state of the physical object leads to a change in the digital object and not vice versus. [...]
>
> If the data flows between an existing physical object and a digital object, and they are fully integrated in both directions, this constituted the reference "Digital Twin". A change made to the physical object automatically leads to a change in the digital object and vice versa. [...]" [17]

This definition puts the data exchange into the center to provide a distinction between a digital shadow and a digital twin. The focus on a directed data exchange is the crucial aspect distinguishing a generic model from what should be called a digital shadow or digital twin. A distinct terminology regarding the data flow direction provides not only a very useful and easily accessible insight into the type of model but is also obvious to determine in practice.

Regarding digital representations, a distinction of the type of data is important. One key intuition behind a digital representation is the reflection of a physical object during operation. If the physical object changes, the digital representation should act accordingly. This can only be archived by an exchange of quantitative data. Note that this does not exclude any qualitative data transfer but puts the quantitative data exchange as a minimal requirement.

An additional point to stress is the modeling at the level of interest. Any system and thus, any model has an inherent notion of the level of detail. This should be stressed, especially for models operating in different regimes, combining different domains, and effects. Thus, the level of detail is explicitly included in the definition of digital shadows and digital twins, which is stated in (2.II).

2.II. ***Definition of Digital Shadows and Digital Twins***

We call a digital model $\mathcal{D}$ of a physical object $\mathcal{P}$ that shares the same characteristics at the level of interest as its physical counterpart a

i) Digital Shadow (DS), if there exists a directed quantitative data flow from $\mathcal{P}$ to $\mathcal{D}$.

ii) Digital Twin (DT), if there exists a bi-directional quantitative data flow between $\mathcal{P}$ and $\mathcal{D}$.

Equally true for digital representations as a subset of general models, the digital counterpart is not a stand-alone quantity as there is no relevance to $\mathcal{D}$ without $\mathcal{P}$. Additionally relevant to DSs and DTs is the presence of a quantitative data transfer by definition (2.II). Thus, we can equivalently state a DS by

$$(\mathcal{D}, \mathcal{P}, \mathcal{D} \rightharpoonup \mathcal{P}) \tag{2.3}$$

and a DT by

$$(\mathcal{D}, \mathcal{P}, \mathcal{D} \rightleftharpoons \mathcal{P}) \tag{2.4}$$

where $\rightharpoonup, \rightleftharpoons$ denotes directed quantitative data flow.

2.III. ***Definition of Digital Representations***

We mutually refer to digital models, digital shadows, and digital twins as digital representation.

Definitions (2.I), (2.II) and (2.III) are visualized in Fig. 2.1 with a focus on their relative notion. We give a discussion of the presented definitions, starting with a comment on the terminology "physical object" compared to "physical representation". Using physical representation for $\mathcal{P}$ easily suggests using digital representation for $\mathcal{D}$. This is the case for the presentation in [17], where only the quantity $\mathcal{D}$ is referred to as the DT. As argued above, this leads to inconsistency as only the triple of $\mathcal{P}$, $\mathcal{D}$, and the quantitative data flow in between leads to a well-defined concept. Despite being inconsistent, phrased like this a misconception is provided. The data exchange is an integral part of the provided concept distinguishing it from other types of models. Thus, it should be presented on equal footing. This is in line with [51].

The proposed definition differs with respect to the following aspects from [17]. In [17], only online applications are allowed. Note that [17] requires this only for DTs in the definition. The demand for online applications of DSs is given in the subsequent explanation, cf. [17, Fig. 1]. This is a very limiting requirement in practical applications. There is no reason to exclude offline-operated models from the definition. Being online or offline is a classifying attribute, which can be associated with individual realization. A further distinction is the requirement of a quantitative data exchange. Stated as in [17] without this distinction, also a model given by an analytic expression without any quantitative data exchange would qualify as a DT.

There are limiting aspects of the given definitions (2.I), (2.II) and (2.III). As a base quantity, sufficient for this work we consider only physical objects $\mathcal{P}$. In general, also digital representations of virtual objects could be possible. Examples of such virtual objects could be energy market designs, surrogate models of computationally intense algorithms or decision-making pro-

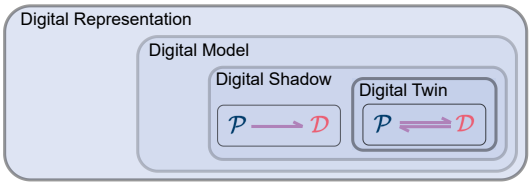

Figure 2.1: Illustration of the concept of digital representations including digital models, DS (directed quantitative data transfer $\rightharpoonup$) and DT (bi-directional quantitative data transfer $\rightleftharpoons$).

cesses by individual car drivers. However, from a conceptual point of view, this immediately implies additional difficulties in proposing a precise definition. For example, a simple copy of some computer script with a comparison of the individual copies' outputs would yield a perfect DT when naively extending the definition to virtual objects. Although nothing fundamentally prevents this particular example from being included, we rather want to emphasize the need for careful consideration of extending the given definitions as explanatory power might be reduced. Although we define digital representation only for physical objects, a digital middle layer might be present. For example, data collected from a power system might be collected and stored in a database. That is, the database yields a digital version of the physical power system, as storing data is required. From a conceptual point of view, this is a consequence of allowing digital representations to be operated offline. In this case, we identify the digital intermediate layer with the physical object. As such, the definitions given apply also to systems digitally mapped to databases.

Another critical aspect already mentioned is the lack of a proper definition of data types. Although intuitively clear which type of data exchange is relevant in the given context, a precise definition is hard to specify.

As a last comment, we want to note that the distinction between models, shadows, and twins is by no means at an evaluation level but rather a pure conceptual separation. All of these concepts might lead to valuable and insightful models.

2.2.3 Classification Scheme of Digital Representations

The previous section conceptualizes the notions of digital models, DSs and DTs. Actual realizations of these concepts have more features worth addressing explicitly. We focus on those characteristics relevant to this work, which are summarized in Fig. 2.2 at the end of this section.

To simplify the discussion, we separate it into three branches, i.e., the system of interest, architectural, and executable aspects. The system of interest refers to the choice of the system under consideration, i.e., the specification of $\mathcal{P}$. With architectural features, we mainly refer to the model chosen for $\mathcal{D}$. The executable class describes the information flow $\rightharpoonup, \rightleftharpoons$ between both constituents.

Classifying the System of Interest $\mathcal{P}$

Level of Detail Although in any application the system under consideration has to be specified, additional aspects have to be considered when dealing with digital representation. First, the choice of the system implies which type of data could be collected and thus, the type of digital representation. In other words, the system's boundary of $\mathcal{P}$, and thus, also the boundary of $\mathcal{D}$ is set. Second, isolated systems, subsystems, and coupled systems demand careful consideration of data collection. For example, it has to be determined if it is possible to define a digital representation of a subsystem on its own and if not, which data from the ambient system should be collected. Note that the level of detail describes a choice for $\mathcal{P}$, but has direct implications for $\mathcal{D}$. More precisely, it sets the level of information within $\mathcal{D}$ that could be maximally expected.

Architectural Classification Scheme for $\mathcal{D}$

Level of Aggregation While the level of detail captures the amount of information captured for $\mathcal{P}$, the level of aggregation describes the subset of this information, that is entered in, processed by, and provided by $\mathcal{D}$. This aspect is simply a restatement of "at the level of interest" from (2.II). The level of aggregation has to be seen relative to the level of detail defined for $\mathcal{P}$.

Static, Dynamic and Updatable Digital Representations The classification attributes of static, dynamic and updatable refers to the flexibility of the digital representation. A static representation captures a single effect under unchanged conditions. Contrarily, a dynamic digital representation can adjust to certain input changes during runtime. Lying between both extremes, the feature of being updatable to new conditions is the third option. The update is meant as an active external action by an operator, adjusting the digital representation to new conditions.

Self-Validation This feature indicates whether the digital representation can indicate its capability of representing the physical counterpart. As $\mathcal{P}$ might change during operation, a self-validation or self-failing indication possibly allows the adjustment of the digital representation to the new conditions or indicates a failing.

Probabilistic and Deterministic These classes refer to which type of model is chosen for $\mathcal{D}$, i.e., a probabilistic or deterministic model. It should be noted that in order to be a useful attribute for digital representations, the specification of which aspect of $\mathcal{P}$ is modeled probabilistically or deterministically by $\mathcal{D}$ is needed. This becomes relevant in hybrid models that are partly driven by data. For example, a variable of a classical probabilistic algorithm can turn deterministic if the particular variable is given by measurements instead of being treated like a random variable.

Executable Classification Classes

Online and Offline Application The terminology online and offline application refers to the degree of automation. We call the information transfer online if an automated data transfer exists and offline otherwise.

Real-Time Application We call a model operated in real-time, if the information provided and transferred between $\mathcal{P}$ and $\mathcal{D}$ needs equally or less time to be provided and transferred compared to the system dynamics to be represented. Both directions of data flow have to be considered separately.

Explicit and Implicit Data Transfer We refer to the standard data transfer between $\mathcal{P}$ and $\mathcal{D}$ in terms of physical bits as explicit. If some data or information flow is provided by the ambient system, i.e., by a physical process not provided by collected data, we call the data transfer implicit. The idea of implicit data transfer might seem counter-intuitive regarding the demand for a quantitative data flow in definition 2.II. However, it is not stated that the data is collected by a measurement device. Although this is not the usual case in common applications, an example is provided in Ch. 6.

Summarizing, the classification classes relevant to this work can be depicted in Fig. 2.2.

2.3 Realizations of Digital Representations

We propose and discuss possible classes of models and how they meet the requirements of being utilized as a digital representation. In Sec. 2.3.1 we discuss grey-box modeling and technical aspects thereof with respect to data-driven modeling in digital representations. In Sec. 2.3.2, we propose realizations by GPs and how they fit into the paradigm of digital representations. We motivate and discuss the specific GP models developed and used in this work in Sec. 2.3.3.

2.3.1 White-, Grey- and Black-Box Modeling

2.3.1.1 Shades of Grey

We discuss the notion of white-, grey-, and black-box modeling with respect to the scope of this work. We focus on hybrid models based on sets of equations. Suppose we have a set of equations $\mathcal{E}$, its solutions $\mathcal{S}_\mathcal{E}$, inputs $\mathcal{I}$ and parameters ϕ. We refer to parameters ϕ that describe the system's characteristic via $\mathcal{E}$ as *system parameters*. An equivalent formulation of the set of equations can be given by an operator $\mathcal{F}_\mathcal{E}$. Then, $(\mathcal{E}, \mathcal{S}_\mathcal{E}, \mathcal{I}, \phi)$ can be written as

$$\begin{aligned} \mathcal{I}_1 &= \mathcal{F}_{\mathcal{E}_1}^{\phi_1}[(\mathcal{S}_1, \dots \mathcal{S}_n), (\mathcal{I}_1, \dots \mathcal{I}_m)] \\ &\vdots \qquad\qquad\qquad\qquad\qquad \Leftrightarrow \quad \mathcal{I} = \mathcal{F}_\mathcal{E}^{\phi}[(\mathcal{S}_\mathcal{E}, \mathcal{I})] \\ \mathcal{I}_m &= \mathcal{F}_{\mathcal{E}_m}^{\phi_m}[(\mathcal{S}_1, \dots \mathcal{S}_n), (\mathcal{I}_1, \dots \mathcal{I}_m)] \end{aligned} \tag{2.5}$$

with m equations given by m operators $\mathcal{F}_\mathcal{E}^\phi = (\mathcal{F}_{\mathcal{E}_1}^\phi, \dots, \mathcal{F}_{\mathcal{E}_m}^\phi)$ which depend on system parameters ϕ, n solutions $\mathcal{S}_\mathcal{E} = (\mathcal{S}_1, \dots \mathcal{S}_n)$ and m inputs $\mathcal{I} = (\mathcal{I}_1, \dots, \mathcal{I}_m)$.

The set of equations could be of algebraic, differential, integral, or mixed nature. Most relevant to this work are linear differential equations dependent on a time variable t and algebraic

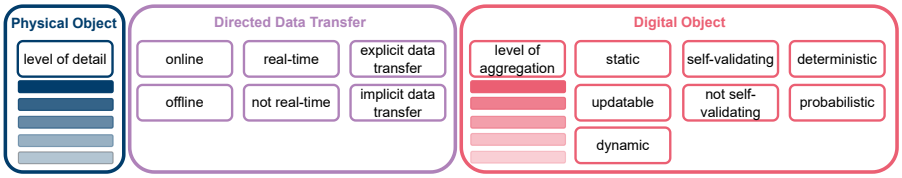

Figure 2.2: Classification scheme of digital representations.

sets of equations. In this case, the operator $\mathcal{F}_i^{\phi_i}$ reads

$$\mathcal{F}_{\mathcal{E}_i}^{\phi_i} = \phi_{i_0} + \phi_{i_1}\frac{d}{dt} + \cdots + \phi_{i_k}\frac{d^k}{dt^k},\ t \in \mathbb{R} \tag{2.6}$$

for differential equations. In the algebraic case, we have

$$\mathcal{F}_{\mathcal{E}_i}^{\phi_i} = \phi_{i_0} + \phi_{i_1} + \cdots + \phi_{i_k} \tag{2.7}$$

In general, the solution $\mathcal{S}_\mathcal{E}$ might be given by analytic expressions or numerical solvers. Initial conditions specify the particular solution $\mathcal{S}_\mathcal{E}$ among infinitely many possible solutions. In this work, we omit to state the initial conditions explicitly. This is because the digital representations presented in this work are based on measurements of $\mathcal{S}_\mathcal{E}$. Thus, the information on the initial condition is implicitly contained in the measured signal represented by $\mathcal{S}_\mathcal{E}$. Note that this also means that the equations do not have to be solved explicitly.

A set of equations is called *forward solved* if the solutions $\mathcal{S}_\mathcal{E}$ are computed according to $\mathcal{I}, \phi$. A set of equations is called *backward solved* if the system parameter ϕ are computed according to $\mathcal{I}, \mathcal{S}$.

White-box models are given by analytic solutions or provided by numerical solvers. A technical distinction between both solutions might be a time-continuous or a time-discrete description [19]. We call a model *white-box* if $(\mathcal{E}, \mathcal{I}, \phi)$ is sufficiently known such that the computed solution $\mathcal{S}_\mathcal{E}$ matches the corresponding physically observed quantities.

We call a model *black-box* if $\mathcal{S}_\mathcal{E}$ is given without any knowledge of $(\mathcal{E}, \mathcal{I}, \phi)$. An obvious realization of pure black-box modeling is the usage of ANN in their basic formulation.

The full spectrum between white- and black boxes might yield useful digital representations. As the focus of this work is the exploration between both extremes, we focus on grey-box aspects. We call a model a *grey box*, if $(\tilde{\mathcal{E}}, \tilde{\mathcal{S}}_\mathcal{E}, \tilde{\mathcal{I}}, \tilde{\phi})$ are partly known where we use $\tilde{\cdot}$ to express that some but not necessarily complete knowledge exist. A visualization of the spectrum between white- and grey-box modeling is given in Fig. 2.3. The particular stages of included systems' knowledge are given by the color-coded boxes. The spectrum ranges from full system knowledge such as particular system solutions and system parameters to qualitative notions such as the relative signal behavior.

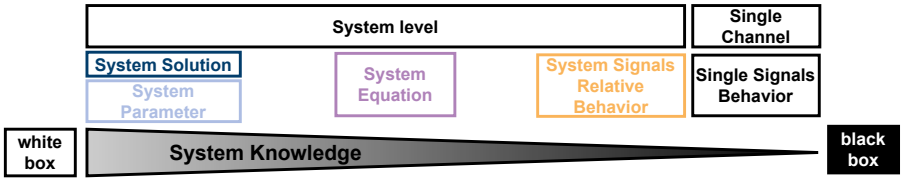

Figure 2.3: Visualization of white-, grey- and black-box modeling with relation to system's description.

2.3.1.2 Operator Classification in Grey-Box Modeling

In the last section, we denoted a grey-box model by $(\tilde{\mathcal{E}}, \tilde{\mathcal{S}}_{\mathcal{E}}, \tilde{\mathcal{I}}, \tilde{\phi})$. This should express that some knowledge about each individual constituent is available, but not necessarily all. Classically, a set of equations is solved forward or backward. Within grey-box modeling, more possibilities exist, depending on the prior knowledge about $(\tilde{\mathcal{E}}, \tilde{\mathcal{S}}_{\mathcal{E}}, \tilde{\mathcal{I}}, \tilde{\phi})$. This implies the need for careful consideration when it comes to model description. A particular example is the notion of linear operators. We illustrate this with a simple artificial example.

Consider a differential operator decorated with parameter a as a minimal example of (2.6) acting on a term $f(t)g(t)$ as a part of a differential equation, that is

$$\dots = a\frac{d}{dt}[f(t)g(t)] + \dots \tag{2.8}$$

A forward solution is the specification of $f(t)$ and $g(t)$ subjected to the known parameter a. Note that the operator $a\frac{d}{dt}$ does not act linearly on the product $f(t)g(t)$ due to the product rule. A backward solution provides the parameter a subjected to the known product $f(t)g(t)$, e.g., by a dedicated measurement. In this case, the product $f(t)g(t)$ or its derivative turns into a discrete list of measurements, i.e., $\{f(t_i)g(t_i)|i = 1, \dots\}$ or $\{\frac{d}{dt_i}f(t_i)g(t_i)|i = 1, \dots\}$. Note that in this case, it does not make any sense to talk about the linearity of the operator as it is simply given by a list of measurements. Beyond forward and backward solutions, more possibilities exist. For example, suppose that $g(t)$ is given by measurements. Then, as evident from

$$a\frac{d}{dt}[f(t)\{g(t_i)|i = 1, \dots\}] = a\frac{d}{dt}[f(t)g(t_i)],\ i = 1, \dots, \quad \forall t = t_i \tag{2.9}$$

the operator acts linearly on $f(t)$ as there is no product of analytical functions present. This is because the product of functions reduces to the product of a function with a scalar given by the measurements. Therefore, depending on the particular setup of $(\tilde{\mathcal{E}}, \tilde{\mathcal{S}}_{\mathcal{E}}, \tilde{\mathcal{I}}, \tilde{\phi})$, classically known restrictions might not straightforwardly apply, which can be a particular advantage of grey-box modeling by data-driven algorithms.

2.3.1.3 Machine Learning for Grey-Box Modeling in Digital Representations

Good candidates to be applied for grey-box modeling can be found in the field of machine learning. One can roughly distinguish between supervised, unsupervised, and reinforcement learning [24], [39], [82]. For the application of digital representations, supervised methods are an obvious choice as the inclusion of quantitative data is a key ingredient. This data is usually

obtained by conducting measurements of physical objects under normal conditions. Although unsupervised and reinforcement methods are not excluded in general, additional justification is needed if known system knowledge is not included in the particular realization. To contextualize the chosen class of algorithms, we briefly discuss unsupervised and reinforcement methods in the context of digital representations before elaborating on supervised machine learning in more detail.

Reinforcement Learning Reinforcement learning can be described as a framework that learns a certain output by reward and punishment signals according to some predefined goal [82]. As such, the application of reinforcement learning is made for systems where the desired output is unknown or can not be pre-determined. This is not the case for digital representations as the measurement of the physical object is a key ingredient.

Unsupervised Learning Unsupervised machine learning does not require labeled data [39]. In line with the previous argument, this is rarely the case for digital representations as the signals of interest are explicitly measured, commonly under normal operational conditions. Thus, a natural label can be assigned.

Supervised Learning Within supervised machine learning, classification, and regression models can be considered [24]. The choice depends on the level of aggregation desired for the specific application. In general, both types of algorithms might yield insightful digital representations. For example, for a power system operator, it might be crucial to have a detailed representation of each node and line within the system for a detailed understanding of the system dynamics. In this case, a regression-based machine learning tool is in favor of this particular application. However, it might be also sufficient to provide a binary classification of the system which simply states if the system is working as desired are not.

2.3.1.4 Algorithmic Realizations

In this work, we focus on models yielding a full, non-aggregated representation of the physical object. This is given by regression types of algorithms, which are summarized visually in Fig. 2.4. Independent of the particular application, any new algorithm should be able to outperform a trivial solution. The Trivial Approach (TRIV) might take another related data point for the regression of the one that should be provided. A simple classical algorithm dependent on a single signal is Linear Interpolation (Lin-Int) [83]. A more evolved technique based on matrix completion might be given by Compressive Sensing (CS) [84]. All three types of algorithms are based on a single channel's information, without any reference to a systematic relationship. Thus, they can be rooted on the darker side of the spectrum. On a multi-signal level including relative signals behavior, Vector-valued Auto-Regressive Moving Average (VARMA) models might be considered [85]. Most prominent within the field of machine learning is the usage of neural networks [39]. While ANN are pure black-box models, certain neural network architectures exist that include more information about the system. For example, a temporal structure might be implemented by a Long Short-Term Memory (LSTM) neural network [39]. Beyond that, PINNs

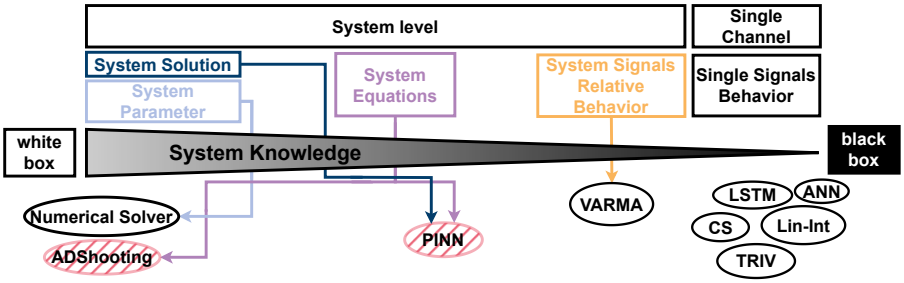

Figure 2.4: Embedding of possible digital representation realizations in the realm of grey-box modeling.

include information on the set of equations describing the physical object [86]. Thus, PINNs can be placed halfway as the set of equations, but not their solutions are known. Based on a classical solver but combined with automatic differentiation, an algorithm called ADShooting can be considered. Although not the focus of this work, we propose technical contributions to PINNs and ADShooting as indicated with the red color in Fig. 2.4. We provide a detailed exploration of these algorithms in Sec. 5.2.

2.3.2 Gaussian Process Characteristics for Digital Representations

As described above, many possible algorithms might yield useful realizations of digital representations. In this work, we focus on GPs, a supervised probabilistic machine learning regression algorithm [24], [26]. In particular, we consider a multi-output version thereof. Standard GPs, which we refer to as Single-output Gaussian Process (S-GP) if clarification is needed, can represent functions

$$f : D \subset \mathbb{R}^n \to \mathbb{R} \tag{2.10}$$

for a compact set D. When modeling systems, usually more than one signal is of relevance. Thus, there is a need to model different types of signals in a systematic way. In other words, maps

$$f : D \subset \mathbb{R}^n \to \mathbb{R}^m, m > 1 \tag{2.11}$$

have to be considered. We provide a high-level discussion of the individual benefits of GPs regarding their application in digital representations in this section. This should give an overview of the framework of choice without any attempt to provide a mathematically deep and rigorous discussion. A detailed and precise discussion is provided in the remaining part of this work.

2.3.2.1 Conceptual Benefits of Gaussian Processes

Probabilistic Mathematical Structures GPs can be routed in the field of probabilistic machine learning. This includes mathematically well-defined probability spaces with an interpretation of their constituents and the notion of a, so-called, Reproducing Kernel Hilbert Space (RKHS). The latter is uniquely given by the GP's kernel and defines the space of functions utilized by the GP [25]. Contrarily, the function space provided by neural networks lacks this level

of underlying structure. This is because the space of functions is generated by arbitrary combinations of activation functions, called artificial neurons. In particular, there is no well-defined notion of accuracy and precision beyond point estimates.

In principle, a mathematically rich structure does not qualify any technique to be preferable. However, with respect to digital representations, there are the following advantages. From a technical perspective, the approach followed in this work is the inclusion of domain knowledge given by a set of equations into the machine learning algorithm yielding a hybrid model. In other words, the data-driven machine learning algorithm is tailored to the specific physical object $\mathcal{P}$ which is described by a set of equations. Loosely speaking, including mathematical equations into a framework consisting of a rich underlying mathematical structure is technical beneficial. It should be noted that the system's description in terms of equations does not have to fit the observed physical object perfectly. Due to the additional presence of a trainable machine learning algorithm, a set of equations at the level of interest is sufficient. In mathematical terms, this refers to *reduced* or *effective modeling*. In the engineering community, the notion of *equivalent models* is used.

Gaussians in Gaussian Processes By definition of GPs, the underlying assumption of being Gaussian distributed holds at a function space level. A precise mathematical description is beyond the scope of this work and we refer to [87] for a detailed description. Schematically, we illustrate the Gaussian distribution present in GPs in Fig. 2.5. For a given data set, there are infinitely many functions fitting this data set. As illustrated in Fig. 2.5-a, three possible functions (colored lines) fitting the data (black crosses) are displayed. To each function, a probability (colored dots) can be associated which is displayed in Fig. 2.5-b. This probability assigns how likely the given function is the best possible choice. The main assumption of GPs is, that the associated probability distribution follows a Gaussian distribution. Thus, demanding the Gaussian distribution is assigned to the functions, which fit the data and not to the data set itself.

Posterior Covariance Directly related to the previous paragraph is the notion of posterior covariances. Similar to Gaussian distributions, a GP is specified by a mean (function) and a covariance (function). The covariance of the trained GP is called the posterior covariance. As the probability distribution is on a function space level, the posterior covariance can be interpreted as a measure of how well the GP suits the data. Note that this depends on the prior assumptions made for the GP. It should be noted that this is not the same as the accuracy of the provided regression results. In other words, the posterior covariance indicates if subjected to selected priors, the GP model fits the data and not the accuracy of the fit. The choices made for the prior assumptions imply the posterior behavior. Thus, if the prior does not fit the data, the quantities computed for the posterior do not provide a meaningful insight. Strictly speaking, the only scenario where the posterior covariance can be interpreted as an error measure of the predicted data can be described as follows. Suppose we fix the prior mean, covariance, and hyper-parameter of a GP and sample from this to generate some data. The accuracy of an additional GP with the same mean and covariance trained on this data set can be measured by its posterior covariance.

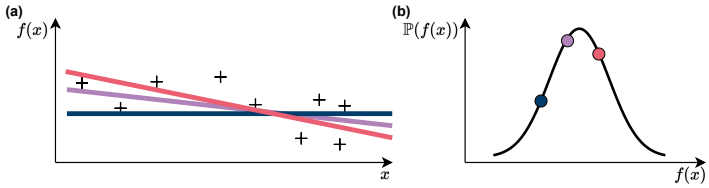

Figure 2.5: Gaussians in Gaussian Processes: (a) Data (black crosses) and possible functions explaining the data (colored lines), (b) Corresponding probabilities (colored dots) of the functions provided in (a).

While this might seem to be a limiting aspect at first sight, it turns out that the proper interpretation of posterior covariances yields a useful property for the application of digital representations. While the absolute value of the posterior covariance has no direct meaning, this is fundamentally different from a relative point of view. A key aspect of digital representation is the adaption of the current state of a physical object by incoming data. If the object is changing during operation this is reflected in the data entering the digital representation. In this case, the GP's posterior variance significantly increases. A practical use case is discussed in Ch. 4. Thus, the posterior covariance yields a self-validation characteristic of digital representation. This is very useful as the validity of the provided digital representation model is a major concern in data-driven modeling.

Universal Approximator Certain GPs have the universal property, that is, they can fit any continuous function up to arbitrary precision [88]. More precisely, those GPs yield a function representation that is dense in the space of continuous functions. All GPs provided in this work have this property. Stated differently, there are no practical limitations regarding the types of signals that can be addressed.

2.3.2.2 Technical Benefits of Gaussian Processes

Noise Hyper-Parameter In the formulation of GPs, noise is added explicitly to the model by an accessible hyper-parameter [26]. This hyper-parameter can be a trainable quantity or pre-determined. Contrarily in ANNs, noise is treated implicitly via regularization [39]. Explicit access to the noise level is especially useful in digital representations as noise is a key ingredient of any data collected from physical objects. Moreover, this benefits the GP models to be restricted to the level of interest.

Non-Equidistant Data Sets In its basic formulation, GPs model functions $f : D \subset \mathbb{R}^n \to \mathbb{R}$ with a training data set $(D, f(D))$ [26]. That means a data point in the domain D is associated with the function's value. Consequently, GPs can be trained on arbitrarily distributed sets D, not necessarily on uniform, i.e., equally spaced data. While this is a convenient property in general, its true benefits will be evident in multi-output modeling for digital representations as discussed below.

Numerically Tractable In general, probabilistic modeling requires significant computational resources [89]. This is because spaces defined by probabilistic models are much larger compared to logical models as they contain all possibilities. Consequently, when dealing with these types of models, dedicated approximation schemes such as Monte Carlo integration or Markov Chain Monte Carlo simulations have to be considered. Compared to general probabilistic modeling, GPs are computationally relatively cheap. Technically, this is because of the handy properties of Gaussian distributions, implying computations at a linear algebra level. Moreover, posterior mean and covariance can be computed without any approximations. Only the objective function, called Negative Log Marginal Likelihood (NLML), requires approximations in order to be computationally tractable [26]. However, in general, GPs have a computational complexity of $\mathcal{O}(N^3)$, where N is the number of training data points.

2.3.3 Road to Gaussian Processes with Multiple-Outputs

In this section, we propose an application and model-building framework usable for digital representation by GPs. Besides reviewing different modeling perspectives, we propose and discuss the system's point-of-view, yielding continuous single-input multi-output GPs models as suitable realizations of digital representations.

2.3.3.1 The System's Point of View

In general, the input-output relation of grey-box models needed for digital representations can be visualized as in Fig. 2.6. We focus on the model-building aspects, highlighted with light blue in Fig. 2.6. For simplicity, the connection to the physical object $\mathcal{P}$ is not considered for now. The input signals that enter the digital representation are denoted by $S_1, S_2, \ldots$ while we denote the output signals by $S_1^*, S_2^*, \ldots$.

Visualized in Fig. 2.7, there is no clear distinction between inputs and solutions. In panel a) of Fig. 2.7, a classical solution of a set of equations is depicted with input $\mathcal{I}$ and output solution $\mathcal{S}_\mathcal{E}$. Usually, $\mathcal{S}_\mathcal{E}$ is called a state in this context. Classically, this scenario frames the terminology input and solution/state. In panel c) the backward solution is visualized with inputs (of the digital representation) $\mathcal{I}$ and $\mathcal{S}_\mathcal{E}$ and digital representation output ϕ. In panel b) a particular example of switching the notion of input and output from a classical perspective can be seen. In this work, we stick to the classical notation of inputs and states as this is the common terminology used. However, it should be noted that this does not automatically imply, that an input is also the input of the digital representation nor a solution is synonymously used for an output. We speak of signals instead of inputs, states, and solutions if this should be highlighted.

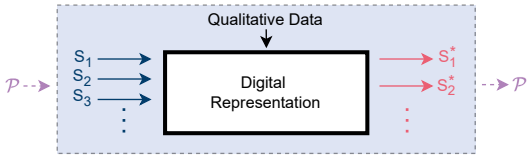

Figure 2.6: Input-output of digital representations.

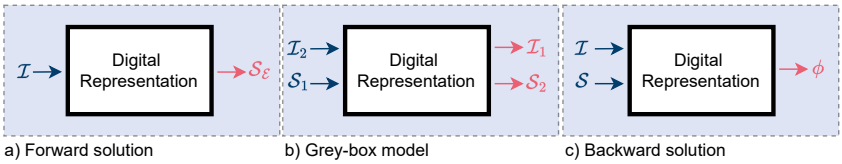

Figure 2.7: Variable input-output relations depending on the specific application.

Moreover, the definition of the input and output of the digital representation does not imply that the model chosen as a realization takes these quantities also as inputs and outputs, respectively. Highlighted in light blue in Fig. 2.8, the model's input (x-axis) and the output (y-axis, i.e., red and blue functions), which yields the digital representations, do not have to coincide with the input/output of the digital representation. It should be noted, that from the perspective described in Fig. 2.8, the model has multiple outputs but just a single input.

We call the known signals entering the digital representation the *input of the digital representation* and the output, i.e., prediction of the digital representation the *output of the digital representation*. These notions might not coincide with the input and output (states) of a given set of equations or the particular algorithms used.

We illustrate this with the following, classical analogue. A general discussion is given in the next section. Suppose we consider a system of coupled linear differential equations with analytic solutions. Then, one might state the solution by their analytic expressions. When comparing the data obtained by an experiment, only discrete instances of the solution are evaluated and compared to the analytic expression. Alternatively, a numerical solution at finitely many points might be given. Although both solutions are evaluated at discrete data points, there is a major difference between both approaches. The analytic solution exists for every instance in the domain, i.e., in a continuous space, while the second one only exists in a discrete subset thereof. It should be noted that the latter is a pure consequence of solving the equations numerically and by no means intrinsic to the general solution itself.

2.3.3.2 Continuous and Discrete Models

We discuss the difference between discrete and continuous modeling regarding its application for digital representations from a broader perspective. First, note that at the level of governing equations (2.5), the functions of interest are of the form

$$\begin{aligned} f_i : D \subset \mathbb{R}^l &\to \mathbb{R} \\ x &\mapsto f_i(x) \end{aligned} \tag{2.12}$$

where $f_i = \mathcal{S}_i, \mathcal{I}_i$ are mapped into a continuous space. The set of equations $\mathcal{E}$ provides a relation between $f_i(x)$ and $f_j(x)$ with $\forall i, j, i \neq j$ and $x \in \mathbb{R}^l$. In this work, $x \in \mathbb{R}^l$ usually denotes the time variable t. When solved numerically, starting from some initial condition x_0, (2.12) turns discrete, i.e.,

$$f_i(x_0) \mapsto f_i(x_1) \mapsto f_i(x_2) \mapsto \dots \tag{2.13}$$

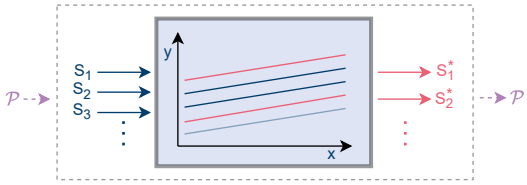

Figure 2.8: Input (x-axis) and output (y-axis) of the model yielding the digital representations.

This is a consequence of numerically approximating derivatives and integrals as they are not exactly tractable in numerical solvers. More formally, we have the following diagram.

$$\begin{array}{ccc} \{x_0,\dots,x_N\} & \longrightarrow & \{f_i(x_0),\dots,f_i(x_N)\} \\ \uparrow{\scriptstyle |_{\{x_0,\dots,x_N\}}} & & \uparrow{\scriptstyle |_{\{f_i(x_0),\dots,f_i(x_N)\}}} \\ D \subset \mathbb{R}^l & \xrightarrow{\quad f_i \quad} & \mathbb{R} \end{array} \qquad \begin{array}{ccc} \{x_0,\dots,x_N\} & \longrightarrow & \mathbb{R} \\ \uparrow{\scriptstyle |_{\{x_0,\dots,x_N\}}} & & \uparrow{\scriptstyle Id} \\ D \subset \mathbb{R}^l & \xrightarrow{\quad f_i \quad} & \mathbb{R} \end{array} \tag{2.14}$$

Although, both domains $D \subset \mathbb{R}^l$ are restricted to discrete spaces by the restriction map $|_{\{x_0,\dots,x_N\}}$, i.e., the vertical maps of (2.14) on the left, the image of f_i does not have to be restricted necessarily. In the left diagram of (2.14), the image is to be considered in a discrete space by applying $|_{\{x_0,\dots,x_N\}}$ while on the right, the identity Id maps to continuous space, even if evaluated at discrete positions $\{x_0,\dots,x_N\}$ afterward.

Summarizing, viewing the functions-image as a discrete space is a direct consequence of using a numerical solver and is by no means intrinsic to the problem's formulation. As such, also the accuracy and precision of the provided digital representation constructed in continuous space might be independent of the underlying discrete structure. In other words, avoiding a discrete step-wise realization as in (2.13) might also avoid error propagation by the individual steps.

2.3.3.3 Continuous Space System's Point of View Model for Gaussian Processes

By construction, S-GPs model continuous functions. Moreover, the system's description in terms of a set of equations is formulated in continuous space relating different signals. Thus, a digital representation of the system in continuous space by a GP with multiple outputs, where each individual output is linked to the other outputs by the system equations, is a straightforward approach. The explicit construction is provided in Ch. 3 in detail. In the following, we highlight certain facts and consequences of this approach.

First, turning from multi-input to multi-output models has the advantage of avoiding the curse of dimensionality, which is a common issue in the field of machine learning [24]. When inputs and outputs of the digital representation match the inputs and outputs of the GP model, a relatively high dimension of the GP's domain compared to the one-dimension representing the time coordinate appears. Thus, the needed sampling in the GP's domain, and thus, computational complexity can be reduced.

Second, each kernel comes with an associated RKHS, a function space with certain characteristics. In other words, models based on GPs are not black boxes but always encode some

chosen predefined properties. While each signal present is provided in the form of a time series with certain physically interpretable properties, the inherent function characteristics get altered when put into a function with multiple inputs. This complicates the process of kernel selection in practical applications.

Third, by utilizing single-input, multi-output GP models, the system's dynamics can be encoded into the kernel as will be developed in the remaining part of this work. For multi-input GPs there is no such construction.

Fourth, by staying in the continuous time regime, error propagation during operation can be avoided. Refraining from a discrete solver-like iteration also avoids an accumulation of errors. In particular, posterior covariances do not increase during operation.

3

Realizations of Digital Representations by Gaussian Processes

In this chapter, we propose and discuss the technical realizations of digital representations in energy systems developed in this work. We focus on a general discussion for now and provide details regarding specific applications in Chs. 4 - 6. We start with reviewing the basics of GPs in Sec. 3.1. In Secs. 3.2 - 3.6, we propose technical realizations of digital representations by GPs. We define, construct, and discuss two classes of GP kernels for digital representations of energy systems. In Sec. 3.2, for both classes, we propose a dedicated model-building scheme allowing a well-defined and numerically stable application within practical settings. In 3.3, we provide implications thereof for physics-informed kernel matrices based on linear operators. These models are known in literature and illustrated with mathematical toy problems. By the proposed model-building scheme, we provide an application to more elaborated cases. In Sec. 3.4, we propose affine operator kernel matrices. The extension from linear to affine operator kernels allows the application in real-world applications. In 3.5, we provide the construction of correlated GPs based on separable kernels. We summarize and classify the given constructions in Sec. 3.6. Provided in [1], [2], [3], [5], [6], [7], this chapter summarizes the key technical contributions developed in this work.

3.1 Theoretical Background of Gaussian Processes

3.1.1 Machine Learning from a Probabilistic Perspective

In this section, we provide a review of the field of machine learning from a probabilistic perspective. We mainly follow the line of reasoning of, and refer to [24]. We give an informal discussion on the probabilistic aspects of machine learning without any attempts of mathematical rigor. This is to classify the techniques used in this work in the field of machine learning and to convey its underlying concepts. We assume some familiarity with Bayesian probability theory which is reviewed in App. A.3 - A.7.

Machine learning can be described as the attempt to learn from data. Contrary to classical approaches, explaining physical effects by heuristic or first-principle derivations, machine learning in its pure form tries to extract certain information directly from data avoiding an intermediate description by an analytic mathematical description.

We informally discuss how supervised and unsupervised machine learning can be viewed from a probabilistic perspective. Unsupervised learning can be seen as conditional probability $\mathbb{P}(X|\Theta)$ for some probability measure $\mathbb{P}$, hyper-parameters Θ, and a data set X. The hyper-parameters Θ summarize the model specification assumed together with a certain probability measure $\mathbb{P}$ yielding the prior to extract data from a given data set X. Supervised learning can be associated with the concept of conditional probability in the form of $\mathbb{P}(X^*|X,\Theta)$. With some prior beliefs Θ about the underlying data structure, the system is conditioned, called trained in the field of machine learning, on the training data set X to make predictions on the test data set X^*. An example of the explicit construction of the probability measure $\mathbb{P}$ and its corresponding probability space for continuous functions can be found in [90]. In this work, the space of interest is called the RKHS for which details will be given in Sec. 3.1.3. For point estimators like, for example, neural networks, the (generalized) probability measure could be given by the Dirac delta distribution $\delta(x)$ which is 1 when evaluated at x and zero otherwise.

When X^* is categorical, the supervised machine learning algorithm is utilized for a classification task. A real-valued X^* yields a regression algorithm. It is convenient to distinguish between two types of regression models, namely parametric and non-parametric. A model is said to be *parametric* if the amount of parameters grows with the amount of training data. Otherwise, the model is called *non-parametric*. On top of parameters, both types of models have hyper-parameters, i.e., those quantities that technically define the model. To distinguish, in the machine learning literature parameters are trained while hyper-parameters are tuned. It should be noted that the distinction between parameter and hyper-parameter depends on the given context and is by no means well-defined or strictly used in the field of machine learning. As the main technique of GPs used in this work is non-parametric, we use the term training for adapting hyper-parameters to the given data sets as common in this field.

Well-known problems in machine learning are overfitting and the curse of dimensionality. Overfitting occurs when highly variable models are trained to data sets in such a way that noise is treated as a reasonable data structure. This is, for example, a common issue when dealing with ANNs as they might consist of hundreds or thousands of parameters. In this work, non-

parametric models are considered. Intrinsically, these types of models are less flexible, and thus, overfitting is not an issue for non-parametric models. The curse of dimensionality can be rooted in the simple fact that the volume of a space grows exponentially with the dimension of the space. Thus, sampling with a reasonable resolution gets computationally intractable for higher dimensions. This is not an issue for the models in this work. Roughly stated, by turning from multi-input to multi-output models as proposed in this work, we can incorporate knowledge about the systems under consideration into the model. This has two advantages. First, the domain of the constructed algorithms is one-dimensional (time coordinate), yielding the possibly lowest dimension. Second, by incorporating the system's description into the multi-output structure, the amount of data needed is significantly reduced. Consequently, overfitting and the curse of dimensionality are not relevant concerns for the models developed in this work.

3.1.2 Basics of Gaussian Processes

We review the basic definitions and derive the explicit expressions of GPs following [24], [26].

3.1.2.1 Definition of Gaussian Processes

We start with the definition from a function space perspective.

Definition: A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution. (3.1)

The statement of about any finite number indicates that implicitly a limit is considered which turns (3.1) to a statement about function spaces. This has to be seen in contrast to spaces consisting of (fixed) data sets in which Gaussian distributions can be defined. We discuss these function spaces, i.e., RKHS, below.

We derive the basic equations from definition (3.1). Let $\mathscr{X}$ be a compact subset of $\mathbb{R}^N$, i.e., $\mathscr{X}$ is bounded and closed. We call

$$\mu : \mathscr{X} \longrightarrow \mathbb{R} \tag{3.2}$$

a *prior mean function*, *mean function* or simply *mean*. A function

$$\kappa : \mathscr{X} \times \mathscr{X} \longrightarrow \mathbb{R} \tag{3.3}$$

is called a *prior covariance function* or *covariance function*, if κ is symmetric positive semi-definite. That is, κ must be symmetric

$$\kappa(x, x') = \kappa(x', x), \quad \forall x, x' \in \mathscr{X} \tag{3.4}$$

and positive semi-definite,

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \kappa(x_i, x_j) c_i c_j \geq 0, \quad \forall x_i, x_j \in \mathcal{X}, c_i, c_j \in \mathbb{R} \tag{3.5}$$

for any finite subset $\mathcal{X} \subset \mathscr{X}$ of dimension N. It is worth noting that the property (3.5) is very restrictive, significantly limiting the existence of possible realizations. Having definition (3.3), for $\mathcal{X}, \mathcal{X}' \subset \mathscr{X}$ finite, we define the prior covariance matrix as

$$K(\mathcal{X}, \mathcal{X}') = (\kappa(x_i, x_j))_{ij},\ \forall x_i \in \mathcal{X}, \forall x_j \in \mathcal{X}' \tag{3.6}$$

As (3.6) establishes the relation between the covariance function $\kappa : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ and the corresponding matrix representation $K(\mathcal{X}, \mathcal{X}')$, we use the terms *covariance matrix*, *kernel matrix* or simply *kernel*, interchangeably. It can be shown that positive semi-definiteness according to (3.5) coincides with the notion of positive semi-definiteness for matrices [25], i.e.,

$$c^T K(\mathcal{X}, \mathcal{X}) c \geq 0, \quad \forall c \in \mathbb{R}^N \tag{3.7}$$

Suppose a function $f(x)$ can be represented as a GP. In that case, we write

$$f(x) \sim \mathcal{GP}(\mu(x), K(x, x')) \tag{3.8}$$

It can be shown that a GP is fully characterized by a prior mean function $\mu(x)$ and a prior kernel function $K(x, x')$ [91]. We have

$$\mu(x) = \mathbb{E}[f(x)] \tag{3.9}$$

$$K(x, x') = \mathbb{E}\left[(f(x) - \mu(x))(f(x') - \mu(x'))\right] \tag{3.10}$$

where $\mathbb{E}[\cdot]$ denotes an expectation value. Although (3.9) and (3.10) are defined point-wise, i.e., for every x individually, we want to emphasize that $\mu(x)$ and $K(x, x')$ are functions of x. This has to be seen in contrast to a multivariate Gaussian distribution, where μ is a fixed vector and Σ is a fixed matrix.

3.1.2.2 Predictions of Gaussian Processes

Suppose a function f can be represented by a GP. Moreover, let f be given by noisy observations y, that is, $f(x) + \epsilon = y$ for some $x \in \mathcal{X}$ and noise $\epsilon \in \mathbb{R}^+$. We denote the training data set by

$$(\mathcal{X}, \mathcal{Y}) = \{(\mathcal{X}, \mathcal{Y}) | (x, y) \in \mathbb{R}^N \times \mathbb{R}, N \in \mathbb{N}\},$$

i.e., the set of the noisy function observations $\mathcal{Y}$ at positions $\mathcal{X}$. The test set defined as

$$\mathcal{X}^* = \{x \in \mathbb{R}^N, N \in \mathbb{N}\} \tag{3.11}$$

is the set at which f^* should be predicted. By definition (3.1) the function evaluations $f(x)$ as well as the predicted function evaluations $f^*(x^*)$ have a joint Gaussian distribution. That is,

$$\begin{pmatrix} f(\mathcal{X}) \\ f^*(\mathcal{X}^*) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu(\mathcal{X}) \\ \mu^*(\mathcal{X}^*) \end{pmatrix}, \begin{pmatrix} K(\mathcal{X}, \mathcal{X}) & K(\mathcal{X}, \mathcal{X}^*) \\ K(\mathcal{X}^*, \mathcal{X}) & K(\mathcal{X}^*, \mathcal{X}^*) \end{pmatrix}\right) \tag{3.12}$$

If $f(x)$ is given by noisy observation $y(x)$, noise is included in the kernel dependent on the training set via

$$K(\mathcal{X}, \mathcal{X}) \mapsto K(\mathcal{X}, \mathcal{X}) + I\sigma_n^2, \tag{3.13}$$

where I denotes the unit matrix of the corresponding dimension, $\sigma_n \in \mathbb{R}^+$ is the noise hyper-parameter, and we assume additive independent identically distributed Gaussian noise. Consequently,

$$\begin{pmatrix} y(\mathcal{X}) \\ f^*(\mathcal{X}^*) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu(\mathcal{X}) \\ \mu^*(\mathcal{X}^*) \end{pmatrix}, \begin{pmatrix} K(\mathcal{X}, \mathcal{X}) + I\sigma_n^2 & K(\mathcal{X}, \mathcal{X}^*) \\ K(\mathcal{X}^*, \mathcal{X}) & K(\mathcal{X}^*, \mathcal{X}^*) \end{pmatrix}\right) \tag{3.14}$$

By standard rules for conditioning, with details provided in App. A.7, we can write the posterior as

$$\mathbb{P}(f^*|\mathcal{X}^*, \mathcal{X}, \mathcal{Y}) = \mathcal{N}(f^*|\mu^*, \Sigma^*) \tag{3.15}$$

with posterior mean

$$\mu^* = \mu(\mathcal{X}^*) + K(\mathcal{X}^*, \mathcal{X})^T \left[K(\mathcal{X}, \mathcal{X}) + I\sigma_n^2\right]^{-1} (f - \mu(\mathcal{X})) \tag{3.16}$$

and posterior covariance

$$\Sigma^* = K(\mathcal{X}^*, \mathcal{X}^*) - K(\mathcal{X}^*, \mathcal{X})^T \left[K(\mathcal{X}, \mathcal{X}) + I\sigma_n^2\right]^{-1} K(\mathcal{X}^*, \mathcal{X}) \tag{3.17}$$

Equation (3.16) describes the prediction of the GP at positions $\mathcal{X}^*$. Equation (3.17) computes the posterior covariance with the variance on its diagonal.

3.1.2.3 Training of Gaussian Processes

The Bayesian approach can be divided into three parts. First, the prior is selected, that is the mean function and prior covariance function. Second, the prior is trained on the given training data set. Third, the trained prior gives the posterior from which predictions can be made.

The training process itself consists of two individual parts, where the notion of training is often used interchangeably. On the one hand side, we have the conditioning of the chosen prior on the given training data set. On the other hand, hyper-parameters of the GP are tuned to match the training data. We follow the common language used in the literature and refer to both aspects as training. If a distinction is necessary in some cases, we point them out specifically. As the first part of conditioning was described in the previous section, we turn to the process of hyper-parameter tuning in the following.

From a theoretical perspective, GPs are rooted in the mathematical framework of Bayesian inference. Thus, there is a natural notion of an objective function in terms of the marginal likelihood. The marginal likelihood is given by

$$\mathbb{P}(\mathcal{Y}|\mathcal{X}) = \int \mathbb{P}(\mathcal{Y}|f, \mathcal{X})\mathbb{P}(f|X)df \tag{3.18}$$

In other words, (3.18) is the probability of obtaining Y given X, where we marginalize over all

possible function realizations $f : \mathcal{X} \to \mathcal{Y}$. Under the main assumption of Gaussian probability distributions and by applying the negative logarithm, equation (3.18) turns into the NLML

$$\begin{aligned}
\text{NLML} &= -\ln\left(\mathbb{P}(\mathcal{Y}|\mathcal{X})\right) \\
&= \frac{1}{2}\left(\mathcal{Y}-\mu(\mathcal{X})\right)^T\left(K(\mathcal{X},\mathcal{X})+I\sigma_n^2\right)^{-1}\left(\mathcal{Y}-\mu(\mathcal{X})\right) && \text{(3.19a)} \\
&+ \frac{1}{2}\ln\det\left(K(\mathcal{X},\mathcal{X})+I\sigma_n^2\right) && \text{(3.19b)} \\
&+ \frac{\dim(\mathcal{X})}{2}\ln(2\pi) && \text{(3.19c)}
\end{aligned}$$

Any hyper-parameters specifying the kernel $K(\cdot,\cdot)$ are trained by minimizing the NLML (3.19).

The three terms of (3.19) can be interpreted as follows. The first term (3.19a), consists of the two-sided multiplication of the training data $\mathcal{Y}$ with the inverse kernel, i.e., the precision of the Gaussian. As such, we can interpret (3.19a) as the data fitting term. The second term (3.19b) consists of the determinant of the kernel, which gives a notion of the model complexity. The last term (3.19c) is a constant resulting from computing the marginal likelihood.

In the relevant equations (3.15) and (3.19) for GPs, the inverse of the kernel matrix is present. For numerical reasons, the inverse matrix is not computed directly. Instead, a Cholesky decomposition of the symmetric, positive semi-definite kernel matrices is performed followed by forward and backward substitution.

3.1.3 Kernels

So far, kernels have been treated on an abstract level based on their defining property. In this section, we give some specific kernels used in this work following [26]. Moreover, we elaborate on the function spaces utilized by the corresponding GPs.

3.1.3.1 Function Spaces of Gaussian Processes

We comment on some theoretical foundations without any attempt for completeness. This discussion is based on [25], [26], [88], on which we refer to for a complete and mathematically rigorous treatment. As stated before, by definition (3.1) a GP yields a probability distribution over functions. Thus, the spaces of interest, called RKHS, are function spaces, defined as follows.

Let $\mathcal{H}$ be the Hilbert space of real functions defined on an index set $\mathcal{X}$, i.e., $\{f|f : \mathcal{X} \to \mathbb{R}\}$, endowed with a scalar product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. We call $\mathcal{H}$ a RKHS if there exists a function $\kappa : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ such that

$$\begin{aligned}
&(i) \quad \text{for every } x,\ \kappa(x,x') \text{ as a function of } x' \text{ belongs to } \mathcal{H} \\
&(ii) \quad \kappa \text{ has the reproducing property } \langle f(\cdot), k(\cdot,x)_{\mathcal{H}}\rangle = f(x)
\end{aligned} \tag{3.20}$$

As shown in [25], for every positive semi-definite function κ there exists a unique RKHS and vice versa. In other words, the kernel defines the particular space of functions used by the GP.

The first property of (3.20) states that for the first argument fixed, the function also belongs

to the same space $\mathcal{H}$ as a function of the second argument. By rewriting part (i) of (3.20) for x' fixed, we define

$$\kappa_{x'} : \mathscr{X} \to \mathbb{R} \tag{3.21}$$

and the corresponding space of functions spanned by (3.21) as

$$\mathcal{K}(\mathscr{X}) = \overline{\text{span}}\{\kappa_{x'} : x' \in \mathscr{X} \text{ compact}\} \tag{3.22}$$

In [88] it is shown, that some kernels have the universal property in the following sense. One can show that $\mathcal{K}(\mathscr{X})$ is dense in the space of continuous functions defined on $\mathscr{X}$. In other words, GPs based on this kernel can fit any continuous function up to arbitrary precision. Thus, no constraints regarding the signals that should be represented by a GP exist. All kernels used in this work have this universal property.

The second property of (3.20) can be interpreted as a generalized analog of the Dirac delta distribution, linking to the discussion on point estimators in a probabilistic context given in Sec 3.1.1. This can be seen as follows. Scalar products $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ in function spaces can be defined by integrals. Informally, we can compare

$$\int f(x')\delta(x - x')dx' = f(x) \tag{3.23}$$

with

$$\int f(x)\kappa(x, x')dx' = f(x) \tag{3.24}$$

which illustrates the analogy. Contrarily to the Dirac delta function, the kernel is non-zero in the neighborhood of x describing the probabilistic aspect of every point x within the GP.

Each RKHS defined by a particular kernel consists of functions displaying certain characteristics. This might be, for example, the order of continuity and differentiability [26], [92].

3.1.3.2 Standard Single-Output Kernels

Based on [26], we give details on the single-output kernels used in this work. Within this work, the single-input dimension is given by the time coordinate. Following common notation in the literature, we define all kernels including a hyper-parameter $\sigma \in \mathbb{R}^+$. This hyper-parameter σ describes a scaling along the function's values, i.e., along the y-axis. Note that σ (without a subscript) as part of the covariance function is whether related to the noise hyper-parameter nor the (prior/posterior) variance of the GP. The inclusion of σ is the reason why standard single-output GPs do not require normalization in general. Let $x, x' \in \mathscr{X} \subset \mathbb{R}$ compact in the following.

Squared Exponential Kernel: The Squared Exponential Kernel, also called the Radial Basis Function Kernel, reads

$$K_{\text{RBF}}(x, x') = \sigma^2 \exp\left(-\frac{|x - x'|^2}{2l^2}\right) \tag{3.25}$$

with hyper-parameter $\sigma, l \in \mathbb{R}^+$. The hyper-parameter l can be interpreted as a lengthscale, specifying the GP behavior with respect to the distance $|x - x'|$. This kernel has the universal property [88] and provides smooth functions [26]. The RBF can be obtained by a limit of Gaussian basis functions centered around x [26]. As such, in common applications, the prediction of the GP based on the RBF kernel tends to drop down to zero beyond the provided training data set. This property is hereditary from the used Gaussian basis functions.

Rational Quadratic Kernel: The Rational Quadratic Kernel reads

$$K_{\mathrm{RQ}}(x, x') = \sigma^2 \left(1 + \frac{|x - x'|^2}{2\alpha l^2}\right)^{-\alpha} \tag{3.26}$$

with $\sigma, l, \alpha \in \mathbb{R}^+$. The hyper-parameters l, α can be seen as a scale mixture of lengthscale parameters [26]. In the limit $l \to \infty$, the RBF kernel (3.25) can be obtained [26]. The Rational Quadratic Kernel (3.26) has the universal property [88] and yields smooth function [26].

Matérn Kernel The family of Matérn Kernels can be defined as

$$K_{\mathrm{M}^{\nu+1/2}}(x, x') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|x - x'|^2}{l}\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}|x - x'|^2}{l}\right) \tag{3.27}$$

with $l, \sigma, \nu \in \mathbb{R}^+$, Gamma function Γ and modified Bessel function K_ν [26]. The Matérn kernel is $\lceil \nu \rceil - 1$ times differentiable [26]. For $\nu \to \infty$, we obtain the RBF kernel as a limit [26]. The kernel of the Matérn class has the universal property which can be proven as follows. In [93], it is shown that for characteristics obtained by (3.27), universal in the sense of [88] is equivalent to a property called cc-universal. With properties of (3.27) given in [26], it is shown that (3.27) is cc-universal [94]. The relevant members of the Matérn family for this work are the kernels generated fort $\nu = 1, 2$. The 1-time differentiable Matérn 3/2 kernel

$$K_{\mathrm{M}^{3/2}}(x, x') = \sigma^2 \left(\frac{l + \sqrt{3}|x - x'|}{l}\right) \exp\left(\frac{-\sqrt{3}|x - x'|}{l}\right) \tag{3.28}$$

is used for roughly varying signals. The two-times differentiable Matérn 5/2 kernel

$$K_{\mathrm{M}^{5/2}}(x, x') = \sigma^2 \left(1 + \frac{\sqrt{5}|x - x'|}{l} + \frac{5(x - x')^2}{3l^2}\right) \exp\left(\frac{-\sqrt{5}|x - x'|}{l}\right) \tag{3.29}$$

is used for less roughly varying signals and is a choice between (3.28) and smooth kernels such as (3.25) and (3.26).

3.1.3.3 Operations on Kernels

Let $f \sim \mathcal{GP}[f]$ and $g \sim \mathcal{GP}[g]$ be independent functions represented by GPs with individual kernels K_f and K_g, respectively.

Addition The sum of two kernels K_f, K_g yields a kernel for the function $f + g$ with

$$K_{f+g}(\mathcal{X}, \mathcal{X}') = K_f(\mathcal{X}, \mathcal{X}') + K_g(\mathcal{X}, \mathcal{X}') \tag{3.30}$$

Note that here we refer to the standard addition, i.e., the componentwise addition of the kernel entries as scalars. Thus, two single-output GP representations provide a single-output GP describing the addition of two functions.

Multiplication Similar to the addition, kernel matrices can be multiplied componentwise. The multiplication

$$K_{fg}(\mathcal{X}, \mathcal{X}') = K_f(\mathcal{X}, \mathcal{X}') \cdot K_g(\mathcal{X}, \mathcal{X}') \tag{3.31}$$

yields a kernel for the function $f(x) \cdot g(x)$.

It should be noted, that by addition and multiplication, function characteristics can be encoded in the kernel. For example, having two different characteristics present in a single process that is uniformly present in the entire data set, one could describe each individually by a suitable kernel and add them together. A multiplicative structure would be preferred if certain characteristics are only present in certain regions of the data set. This can be rooted in the individual characteristics of the scalar operations $\cdot$ and $+$. Loosely speaking, a product is large if one of the factors is, while the same holds for the addition if both summands are.

Direct Sum The direct sum $\oplus$ of kernels for two functions reads

$$K_f(\mathcal{X}_1, \mathcal{X}_1') \oplus K_g(\mathcal{X}_2, \mathcal{X}_2') = \begin{pmatrix} K_f(\mathcal{X}_1, \mathcal{X}_1') & 0 \\ 0 & K_g(\mathcal{X}_2, \mathcal{X}_2') \end{pmatrix} \tag{3.32}$$

Tensor Product At the level of kernel matrices, the tensor product $\otimes$ of two GPs is established in terms of the Kronecker product, also denoted by $\otimes$. Let $B^{[d]}$ be a symmetric positive semi-definite $d \times d$-matrix. The Kronecker product reads

$$B^{[d]} \otimes K(\mathcal{X}, \mathcal{X}') = \begin{pmatrix} b_{11} & \dots & b_{1d} \\ \vdots & \ddots & \vdots \\ b_{d1} & \dots & b_{dd} \end{pmatrix} \otimes K(\mathcal{X}, \mathcal{X}') = \begin{pmatrix} b_{11}K(\mathcal{X}, \mathcal{X}') \dots b_{1d}K(\mathcal{X}, \mathcal{X}') \\ \vdots \quad \ddots \quad \vdots \\ b_{d1}K(\mathcal{X}, \mathcal{X}') \dots b_{dd}K(\mathcal{X}, \mathcal{X}') \end{pmatrix} \tag{3.33}$$

Linear Operator Linear operators applied to kernels yield valid kernels. Let $\mathcal{L}$ be a linear operator. The joined kernel for $f(x_1), x_1 \in \mathcal{X}_1$ and $\mathcal{L}[f(x_2)], x_2 \in \mathcal{X}_2$ reads

$$K_{f,\mathcal{L}_x[f]} = \begin{pmatrix} K_f(\mathcal{X}_1, \mathcal{X}_1) & \mathcal{L}_{x_2}\left[K_f(\mathcal{X}_1, \mathcal{X}_2)\right] \\ \mathcal{L}_{x_1}\left[K_f(\mathcal{X}_2, \mathcal{X}_1)\right] & \mathcal{L}_{x_1}\mathcal{L}_{x_2}\left[K_f(\mathcal{X}_2, \mathcal{X}_2]\right) \end{pmatrix} \tag{3.34}$$

Note that in general the sets $\mathcal{X}_1$ and $\mathcal{X}_2$ do not have to be identical.

3.2 Design of Gaussian Processes with Multiple Outputs

In this section, we propose a detailed algorithmic solution for the design of the developed GP models and their corresponding kernels as provided in [1], [2], [3], [5], [7]. We provide details about the peculiarities of the multi-output kernel and propose an algorithmic scheme allowing a consistent and well-defined application. In particular, we propose a balanced training procedure, the usage and handling of noise hyper-parameters, and a dedicated regularization scheme necessary for numerically stable applications.

It should be noted that in literature, sometimes multi-output GPs are synonymously used for separable kernels which are discussed in Sec. 3.5. However, there exist more kernels that yield multi-output GPs as will be discussed in this work.

For simplicity, we derive the proposed algorithmic solution for a two-fold decomposition, i.e., a GP with two outputs. Generalized by iterative application, the results hold for an arbitrary but finite number of outputs. In general, a kernel $K(\mathcal{X}, \mathcal{X}')$ with a two-fold output can be written in block form as

$$K(\mathcal{X}, \mathcal{X}') = K((\mathcal{X}_1, \mathcal{X}_2), (\mathcal{X}_1', \mathcal{X}_2')) = \begin{pmatrix} K_{11}(\mathcal{X}_1, \mathcal{X}_1') & K_{12}(\mathcal{X}_1, \mathcal{X}_2') \\ K_{21}(\mathcal{X}_2, \mathcal{X}_1') & K_{22}(\mathcal{X}_2, \mathcal{X}_2') \end{pmatrix} \tag{3.35}$$

with sub-kernels $K_{11}(\mathcal{X}_1, \mathcal{X}_1')$, $K_{12}(\mathcal{X}_1, \mathcal{X}_2')$, $K_{21}(\mathcal{X}_2, \mathcal{X}_1')$, and $K_{22}(\mathcal{X}_2, \mathcal{X}_2')$.

3.2.1 Balanced Training Procedure

When considering GP with multiple outputs, data normalization has to be explicitly taken into account. In particular, the training process on multiple signals would not be accurate without proper balancing as can be seen by the following derivation. For a kernel in block form (3.35), the Schur complement $K(\mathcal{X}, \mathcal{X}')/K_{22}(\mathcal{X}_2, \mathcal{X}_2')$ of $K(\mathcal{X}, \mathcal{X}')$ with respect to $K_{22}(\mathcal{X}_2, \mathcal{X}_2')$ is defined as

$$K(\mathcal{X}, \mathcal{X}')/K_{22}(\mathcal{X}_2, \mathcal{X}_2') = K_{11}(\mathcal{X}_1, \mathcal{X}_1') - K_{12}(\mathcal{X}_1, \mathcal{X}_2')K_{22}(\mathcal{X}_2, \mathcal{X}_2')^{-1}K_{21}(\mathcal{X}_2, \mathcal{X}_1') \tag{3.36}$$

Then, we can write the inverse of (3.35) as

$$K(\mathcal{X}, \mathcal{X}')^{-1} = \begin{pmatrix} K_{11}^{-1} & K_{12}^{-1} \\ K_{21}^{-1} & K_{22}^{-1} \end{pmatrix} \tag{3.37}$$

with

$$K_{11}^{-1} = (K(\mathcal{X}, \mathcal{X}')/K_{22}(\mathcal{X}_2, \mathcal{X}_2'))^{-1} \tag{3.38a}$$

$$K_{12}^{-1} = -(K(\mathcal{X}, \mathcal{X}')/K_{22}(\mathcal{X}_2, \mathcal{X}_2'))^{-1} K_{12}(\mathcal{X}_1, \mathcal{X}_2')K_{22}(\mathcal{X}_2, \mathcal{X}_2')^{-1} \tag{3.38b}$$

$$K_{21}^{-1} = -K_{22}(\mathcal{X}_2, \mathcal{X}_2')^{-1}K_{21}(\mathcal{X}_2, \mathcal{X}_1')(K(\mathcal{X}, \mathcal{X}')/K_{22}(\mathcal{X}_2, \mathcal{X}_2'))^{-1} \tag{3.38c}$$

$$\begin{aligned} K_{22}^{-1} =& K_{22}(\mathcal{X}_2, \mathcal{X}_2')^{-1} + K_{22}(\mathcal{X}_2, \mathcal{X}_2')^{-1}K_{21}(\mathcal{X}_2, \mathcal{X}_1')(K(\mathcal{X}, \mathcal{X}')/K_{22}(\mathcal{X}_2, \mathcal{X}_2'))^{-1} \\ & K_{12}(\mathcal{X}_1, \mathcal{X}_2')K_{22}(\mathcal{X}_2, \mathcal{X}_2')^{-1} \end{aligned} \tag{3.38d}$$

Mathematical details of inverse matrices by the Schur complements are given in App. A.1. Correspondingly, the NLML (3.19) of (3.35) can be partitioned as

$$\begin{aligned}
\text{NLML} = &\frac{1}{2}\mathcal{Y}_1^T \left(K(\mathcal{X},\mathcal{X}')/K_{22}(\mathcal{X}_2,\mathcal{X}_2')\right)^{-1}\mathcal{Y}_1 && \text{(3.39a)}\\
&+\frac{1}{2}\mathcal{Y}_1^T \Big(-\left(K(\mathcal{X},\mathcal{X}')/K_{22}(\mathcal{X}_2,\mathcal{X}_2')\right)^{-1} K_{12}(\mathcal{X}_1,\mathcal{X}_2')K_{22}^{-1}(\mathcal{X}_2,\mathcal{X}_2')\Big)\mathcal{Y}_2 && \text{(3.39b)}\\
&+\frac{1}{2}\mathcal{Y}_2^T \Big(-K_{22}^{-1}(\mathcal{X}_2,\mathcal{X}_2')K_{21}(\mathcal{X}_2,\mathcal{X}_1')\left(K(\mathcal{X},\mathcal{X}')/K_{22}(\mathcal{X}_2,\mathcal{X}_2')\right)^{-1}\Big)\mathcal{Y}_1 && \text{(3.39c)}\\
&+\frac{1}{2}\mathcal{Y}_2^T \Big(K_{22}^{-1}(\mathcal{X}_2,\mathcal{X}_2') + K_{22}^{-1}(\mathcal{X}_2,\mathcal{X}_2')K_{21}(\mathcal{X}_2,\mathcal{X}_1')\left(K(\mathcal{X},\mathcal{X}')/K_{22}(\mathcal{X}_2,\mathcal{X}_2')\right)^{-1}\\
&\quad K_{12}(\mathcal{X}_1,\mathcal{X}_2')K_{22}^{-1}(\mathcal{X}_2,\mathcal{X}_2')\Big)\mathcal{Y}_2 && \text{(3.39d)}\\
&+\frac{1}{2}\det[K_{22}(\mathcal{X}_2,\mathcal{X}')_2]\cdot\det\left[K(\mathcal{X},\mathcal{X}')/K_{22}(\mathcal{X}_2,\mathcal{X}_2')\right] && \text{(3.39e)}\\
&+\frac{N_1+N_2}{2}\ln(2\pi) && \text{(3.39f)}
\end{aligned}$$

where we used $\mathcal{Y} = (\mathcal{Y}_1, \mathcal{Y}_2)^T$ for each individual signal $\mathcal{Y}_i, i = 1, 2$. From (3.39a) - (3.39d) it is evident that each signal $\mathcal{Y}_i$ enters the NLML (3.39) in the form of

$$\mathcal{Y}_i(\dots)\mathcal{Y}_j \tag{3.40}$$

where $(\dots)$ abbreviates combinations of the corresponding sub-kernel and its inverse. Thus, if the order of magnitude of $\mathcal{Y}_i$ and $\mathcal{Y}_j$ for $i \neq j$ is very different with respect to each other, the numerical contribution to the NLML of the individual parts (3.39a) - (3.39d) differs significantly. Thus, the non-normalized multi-output GP would be primarily trained on just one of the signals, ignoring the other. Practically, this is established by a drop out of the optimizer as the abortion conditions are fulfilled due to a non-significantly change of the value of the objective function NLML. As the GP operated as a digital representation should capture the structure of the system, i.e., the coupling between several signals, normalization of each signal $\mathcal{Y}_i$ is a crucial aspect of block-structured GP. It should be noted that the type of normalization should be consistent with the structure of the original, i.e., the physical system. We summarize as follows.

3.IV. ***Balanced Training Procedure***

A balanced training procedure via a consistent normalization of individual signals is crucial for the design of digital representations via GPs with multiple outputs.

3.2.2 Noise Hyper-Parameter in Multi-Output Structures

GPs display a practical feature regarding noise handling. There exists a noise hyper-parameter $\sigma_n \in \mathbb{R}^+$ in (3.13) that accounts for the level of detail that should be captured by the GP. In other words, direct access to the level of noise is given. This is fundamentally different from, for example, neural networks, where noise is usually addressed indirectly via regularization schemes. Contrarily to other hyper-parameters present in GPs, σ_n plays a distinguished role as, for example, any signal can be described by a linear function for a sufficiently high value of σ_n.

It should be noted that the GP noise hyper-parameter does not have to coincide with the

measurement noise of the signals. There are three contributions that might be included in the value of σ_n. Besides measurement noise, effects or behavior within the signals that should be discarded by the GP can be addressed. Additionally, a third noise-like contribution related to matrix regularization as discussed in the next section can be associated.

In general, the noise hyper-parameter can be trained or pre-determined. Although in certain circumstances the GP algorithm is able to determine σ_n from the training data set, a pre-determination benefits the training process significantly. It should be noted that this is only advantageous if a proper pre-determination in the corresponding application can be guaranteed. As GP models are very sensitive concerning the value of σ_n, care must be taken to fix this value in the model-building process.

The inclusion of the noise hyper-parameter (3.13) into the GP can be extended to multi-output structures as follows. As multiple signals are present, each signal is provided with its inherent noise level in the sense described above. Thus, we define individual noise hyper-parameter $\sigma_{\{\mathcal{Y}_i\}}, i = 1, 2, \ldots$. Then, for $i = 1, 2$ equation (3.13) turns into

$$\begin{pmatrix} K_{11}(\mathcal{X}_1, \mathcal{X}_1') & K_{12}(\mathcal{X}_1, \mathcal{X}_2') \\ K_{21}(\mathcal{X}_2, \mathcal{X}_1') & K_{22}(\mathcal{X}_2, \mathcal{X}_2') \end{pmatrix} \longmapsto \begin{pmatrix} K_{11}(\mathcal{X}_1, \mathcal{X}_1') & K_{12}(\mathcal{X}_1, \mathcal{X}_2') \\ K_{21}(\mathcal{X}_2, \mathcal{X}_1') & K_{22}(\mathcal{X}_2, \mathcal{X}_2') \end{pmatrix} + \begin{pmatrix} I\sigma^2_{\{\mathcal{Y}_1\}} & 0 \\ 0 & I\sigma^2_{\{\mathcal{Y}_2\}} \end{pmatrix} \tag{3.41}$$

where I denotes the unit matrix of the corresponding dimension. It should be noted that the value of the noise hyper-parameter or its domain, should be balanced according to the considerations in Sec. 3.2.1 for a consistent model. We summarize in (3.V).

3.V. *Gaussian Process Noise Hyper-Parameters for Digital Representations*

Contributions to the signals that should be discarded by the GP can be addressed by noise hyper-parameters $\sigma_{\{\mathcal{Y}_i\}}, i = 1, 2$ according to

$$\begin{pmatrix} K_{11}(\mathcal{X}_1, \mathcal{X}_1') & K_{12}(\mathcal{X}_1, \mathcal{X}_2') \\ K_{21}(\mathcal{X}_2, \mathcal{X}_1') & K_{22}(\mathcal{X}_2, \mathcal{X}_2') \end{pmatrix} \longmapsto \begin{pmatrix} K_{11}(\mathcal{X}_1, \mathcal{X}_1') & K_{12}(\mathcal{X}_1, \mathcal{X}_2') \\ K_{21}(\mathcal{X}_2, \mathcal{X}_1') & K_{22}(\mathcal{X}_2, \mathcal{X}_2') \end{pmatrix} + \begin{pmatrix} I\sigma^2_{\{\mathcal{Y}_1\}} & 0 \\ 0 & I\sigma^2_{\{\mathcal{Y}_2\}} \end{pmatrix}$$

Fixing the noise hyper-parameters in the model-building process significantly benefits the models' quality if an accurate pre-determination can be guaranteed.

3.2.3 Regularizations of Kernel Matrices with Block Structure

Analogous to the normalization of signals, care has to be taken when constructing the kernel. This is due to its internal structure caused by the kernel's block form. We develop a dedicated regularization scheme. This has two advantages. First, similar to signal normalization, it regularizes the contribution of each sub-block of the kernel. That is, each sub-kernel contributes equally. It should be noted that this is not only important for the NLML during the training period, but also for the prediction made by the multi-output GP. Second, it solves numerical issues by reducing the kernel's condition number. For kernel matrices, the condition number can be defined as the quotient of the highest and the lowest eigenvalue [27]. It can be shown that the accuracy of the Cholesky decomposition is bounded by the condition number of the kernel matrix [27]. A common issue, also in standard GPs, is relatively high condition numbers of kernel

matrices [95]. The consideration in this work indicates that this characteristic is amplified for kernels with multiple outputs. The threshold of an acceptable value for the condition number depends on the given application, the precise characteristic of the matrix A, the specific algorithm chosen for the Cholesky decomposition, and the corresponding machine precision. In the course of this work, we experienced numerical issues if the condition number roughly approaches 10^6. We propose the following regularization scheme. We describe the individual parts separately.

Two-Sided Regularization First, a two-sided regularization according to [27] is performed. Let $K = K(\mathcal{X}, \mathcal{X})$ be a kernel matrix with Cholesky decomposition $K = L_K L_K^T$. We define a diagonal matrix D such that

$$D = \operatorname{diag}(k_{ii}^{1/2})_{i=1,\dots\dim(K)} \tag{3.42}$$

Then, we define the two-sided transformation

$$K \;\longmapsto\; D^{-1} K D^{-1} \tag{3.43}$$

Note that according to (3.42), the resulting matrix transformed by (3.43) has a unit diagonal. In particular, this holds for both sub-kernels on the block-diagonal, which initially might have different orders of magnitude. The transformation (3.43) provides a change of basis such that the resulting kernel has the lowest condition number, up to $\epsilon > 0$ [27]. The transformation (3.43) has to be taken into account when computing the GP. For (3.19a) we have

$$\begin{aligned} K^{-1} &= \left(DD^{-1}KD^{-1}D\right)^{-1} \\ &= \left(DL_{D^{-1}KD^{-1}}L_{D^{-1}KD^{-1}}^T D\right)^{-1} \\ &= D^{-1}\left(L_{D^{-1}KD^{-1}}^T\right)^{-1}\left(L_{D^{-1}KD^{-1}}\right)^{-1} D^{-1} \end{aligned} \tag{3.44}$$

for the Cholesky decomposition. We denote the solution $x \in \mathbb{R}^n$ of $Ax = b$ for a $n \times n$ matrix A and $b \in \mathbb{R}^n$ using the backslash operator, i.e., $x = A \backslash b$. Thus, we have

$$L_K^T \backslash (L_K \backslash \mathcal{Y}) \;\longmapsto\; D^{-1} L_{D^{-1}KD^{-1}}^T \backslash (L_{D^{-1}KD^{-1}} \backslash D^{-1}\mathcal{Y}) \tag{3.45}$$

For (3.19b), we get

$$\begin{aligned} \frac{1}{2}\ln(\det K) &= \frac{1}{2}\ln(\det[DD^{-1}KD^{-1}D]) \\ &= \frac{1}{2}\ln\left(\det[D]\cdot\det\left[D^{-1}KD^{-1}\right]\cdot\det[D]\right) \\ &= \frac{1}{2}\ln\left(\det[D]\cdot\det\left[L_{(D^{-1}KD^{-1})}L_{(D^{-1}KD^{-1})}^T\right]\cdot\det[D]\right)\Big) \\ &= \frac{1}{2}\ln\left(\det[D]^2\cdot\det\left[L_{(D^{-1}KD^{-1})}\right]^2\right) \\ &= \ln\det[D] + \ln\det\left[L_{(D^{-1}KD^{-1})}\right] \\ &= \ln\prod_{i=1}^{N} D_{ii} + \ln\prod_{i=1}^{N} L_{(D^{-1}KD^{-1})ii} \\ &= \sum_{i=1}^{N}\ln D_{ii} + \sum_{i=1}^{N}\ln L_{(D^{-1}KD^{-1})ii} \end{aligned} \tag{3.46}$$

That is, we have an additional contribution to the determinant by $\sum_{i=1}^{N} \ln D_{ii}$. Similarly, the posterior mean of the transformed multi-output GP reads

$$\begin{aligned}\mu^*_{\text{post}} &= \mu(\mathcal{X}^*) + K(\mathcal{X}^*, \mathcal{X}) \left(DD^{-1}K(\mathcal{X}, \mathcal{X})D^{-1}D\right)^{-1} (\mathcal{Y} - \mu(\mathcal{X})) \\ &= \mu(\mathcal{X}^*) + K(\mathcal{X}^*, \mathcal{X})D^{-1} \left(D^{-1}K(\mathcal{X}, \mathcal{X})D^{-1}\right)^{-1} D^{-1}(\mathcal{Y} - \mu(\mathcal{X})) \\ &= \mu(\mathcal{X}^*) + K(\mathcal{X}^*, \mathcal{X})D^{-1} \left(L^T_{D^{-1}KD^{-1}} \setminus \left(L_{D^{-1}KD^{-1}} \setminus \left(D^{-1} \left(\mathcal{Y} - \mu(\mathcal{X})\right)\right)\right)\right)\end{aligned} \tag{3.47}$$

For the posterior covariance, we have

$$\begin{aligned}\Sigma^*_{\text{post}} &= K(\mathcal{X}^*, \mathcal{X}^*) - K(\mathcal{X}^*, \mathcal{X}) \left(DD^{-1}K(\mathcal{X}, \mathcal{X})D^{-1}D\right)^{-1} K(\mathcal{X}, \mathcal{X}^*) \\ &= K(\mathcal{X}^*, \mathcal{X}^*) - K(\mathcal{X}^*, \mathcal{X})D^{-1} \left(D^{-1}K(\mathcal{X}, \mathcal{X})D^{-1}\right)^{-1} D^{-1}K(\mathcal{X}, \mathcal{X}^*) \\ &= K(\mathcal{X}^*, \mathcal{X}^*) - K(\mathcal{X}^*, \mathcal{X})D^{-1} \left(L^T_{D^{-1}KD^{-1}} \setminus \left(L_{D^{-1}KD^{-1}} \setminus \left(D^{-1}K(\mathcal{X}, \mathcal{X}^*)\right)\right)\right)\end{aligned} \tag{3.48}$$

Note that, contrary to (3.47), the backslash operator acts on a matrix in (3.48). This operation is meant to be column-wise.

Tikhonov Regularization As a second regularization, a basic version of Tikhonov regularization can be performed [96]. The idea of Tikhonov regularization is to change the initial task of solving

$$Ax = b \tag{3.49}$$

for x, i.e., $x = A\backslash b$ to

$$(A + I\epsilon)x = b \tag{3.50}$$

for $\epsilon > 0$ and unit matrix I. By adding a small positive constant ϵ to the diagonal of A, the eigenvalues close to zero are positively shifted. This minimizes numerical errors as the Cholesky decomposition utilizes division by diagonal entries of A. In implementations like, for example, [97], a constant $\epsilon = 10^{-8} \cdot \text{mean}(\text{diag}(K))$ is always added and incrementally increased if the Cholesky decomposition fails. Note that adding ϵ to the diagonal of the kernel is equivalent to (3.13), i.e., adding noise to the GP. Also, note that the signals remain unchanged. The drawback of this approach is that the initial task might be altered in a non-acceptable manner. While not a problem in arbitrary black-box-like GP applications, in this work, hybrid models between GPs and mechanistic models are considered. Thus, certain aspects of these hybrid models have a physical interpretation which might get altered or lost due to this numerical necessity.

Adding Noise In addition to increasing the value of the noise hyper-parameter, noise could also be added to the signals themselves. It turns out, that especially for clean, non-fluctuating signals this might be beneficial. In Sec. 3.2.2, the ability of GPs to handle noise is discussed. At this point, the need for noisy data is stressed. In particular, when working with simulation scenarios a good amount of noise is crucial for numerical stability. In certain test cases investigated in this work, clean signals lead to condition numbers in the order of 10^{33}. For data sets obtained by measurements in the field, this depends on the signal quality and the corresponding individual kernel.

3.2.4 Algorithmic Solution for Block-Structured Gaussian Processes

We summarize the individual regularization steps and propose an algorithmic solution for arbitrarily GP block structures. Note that the order of applications of the regularization scheme matters. First, artificial noise should be added to the individual signals. This is an optional regularization task and depends on the signal's characteristics. Second, a two-sided regularization has to be performed, which unit-diagonalizes the kernel. This procedure does not alter the initial task and should be always taken into account. Third, Tikhonov regularization has to be applied if the Cholesky decomposition is numerically unstable after applying the previous regularization steps. Note that if the kernel is not unit-diagonal before applying Tikhonov regularization, the different orders of magnitude of the kernels on the block-diagonal are altered non-coherently. While on one of the blocks, the added ϵ is minimal in the sense that it causes numerical stability without altering the underlying task too much, this does not necessarily hold for the other block.

When minimizing the NLML, it turns out that several restarts of the optimizer benefit convergence to the minimum of NLML due to local minima. In this work, an L-BFGS-B optimizer is used [98]. Summarizing, we propose the algorithm (3.VI) for block-structured GP.

3.3 Linear Operator Kernel Design

We review the basics of circuit modeling in Sec. 3.3.1. In Sec. 3.3.2, we encode the model's state-space equation into the GP kernel. This is based on a (Linear) PIGPs, which are defined and tested with artificial toy problems in [99]. Provided in [1], [5], [6], we propose a detailed algorithmic procedure for applications as digital representation in Sec. 3.3.3. Moreover, we show and discuss mathematical restrictions of [99] which drastically limits the applicability in real-world problems. We extend the given construction to affine operators as given in [1] in Sec. 3.4. This allows an application of the given framework to real-world problems.

3.3.1 Electrical Circuit Modeling

The core concept underlying the modeling of various systems of interest, such as power electronics and power systems can be addressed via graph theory. We review basic notions thereof [100]. A graph G is an ordered pair

$$\begin{aligned} &G = (N, E), \text{ s.t.} \\ &\quad N = \{n_1, \dots n_i, i \in \mathbb{N}\} \text{ is a set of nodes and} \\ &\quad E \subset \{\{n_1, n_2\} | n_1, n_2 \in N, n_1 \neq n_2\} \text{ is a set of edges} \end{aligned} \tag{3.51}$$

We call G directed, if E consists of ordered tuple $(n_1, n_2) \in N \times N, v_1 \neq v_2$. We label the directed edges according to the ordered nodes, i.e., edge e_{kl} refers to the ordered tuple $(n_k, n_l), k, l = 1, \dots i, i \in \mathbb{N}$. In the given context, edges are also called lines.

A directed edge between k and l can be associated with its instantaneous current value $I_{kl}(t)$. We say that a current is entering a node if the current's direction points towards the node,

3.VI. ***Algorithm for Block-Structured Gaussian Processes***

Denote the multi-signal training set by $(\mathcal{X}, \mathcal{Y}) = ((\mathcal{X}_1, \mathcal{X}_2, \dots), (\mathcal{Y}_1, \mathcal{Y}_2, \dots))$*, with signals* $\mathcal{Y}_i(\mathcal{X}_i) : D_i \to \mathbb{R}, D_i \subset \mathbb{R}^N$ *compact, i.e.,* $\mathcal{Y}_i(\mathcal{X}_i) = \{y_{i_1}(x_{i_1}), y_{i_2}(x_{i_2}), \dots\}$ *and the test set by* $\mathcal{X}^* = (\mathcal{X}_1^*, \mathcal{X}_2^*, \dots)$*. Denote the normalizing constant of signal* $\mathcal{Y}_i$ *by* $n_{\{\mathcal{Y}_i\}}, \forall i$*. For prior mean function* $\mu(\mathcal{X}) = (\mu_1(\mathcal{X}_1), \mu_2(\mathcal{X}_2), \dots)$ *and kernel*

$$K(\mathcal{X}, \mathcal{X}') = K((\mathcal{X}_1, \mathcal{X}_2, \dots), (\mathcal{X}_1', \mathcal{X}_2', \dots)) = \begin{pmatrix} K_{11}(\mathcal{X}_1, \mathcal{X}_1') & K_{12}(\mathcal{X}_1, \mathcal{X}_2') & \dots \\ K_{21}(\mathcal{X}_2, \mathcal{X}_1') & K_{22}(\mathcal{X}_2, \mathcal{X}_2') & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

which depends on hyper-parameter h *we propose the following algorithm for GPs with block structure.*

Algorithm for GPs with block structure.

Prior $K(\cdot,\cdot)$, $\mu(\cdot)$

Training
Input: $(\mathcal{X}, \mathcal{Y})$
Output: $\mu^*(\mathcal{X}^*)$, Σ^*, h

if clean signals **then** ▷ optional
  add noise
$(\mathcal{Y}_i, ...) \mapsto \left(\mathcal{Y}_i / n_{\{\mathcal{Y}_i\}}, ...\right)$ ▷ balance signals
$(\sigma_{\mathcal{Y}_i}, ...) \mapsto \left(\sigma_{\mathcal{Y}_i} / n_{\{\mathcal{Y}_i\}}, ...\right)$ ▷ balance noise
for random restarts **do**
  for minimizer w.r.t. h **do**
    $D = \operatorname{diag}\left(K(\mathcal{X}, \mathcal{X}) + 1(\sigma_1^2, \sigma_2^2, \dots)^T\right)^{\frac{1}{2}}$
    $K = D^{-1}\left(K(\mathcal{X}, \mathcal{X}) + 1(\sigma_1^2, \sigma_2^2, \dots)^T\right) D^{-1}$
    try
      $L = \operatorname{chol}(K)$
    except
      $L = \operatorname{chol}(K + I\epsilon)$ ▷ increase ϵ if needed
    $\alpha = D^{-1} L^T \setminus (L \setminus D^{-1}(\mathcal{Y} - \mu(\mathcal{X}))$
    $\text{NLML} = -\frac{1}{2}(\mathcal{Y} - \mu(\mathcal{X}))^T \alpha - \sum_i \ln L_{ii} - \sum_i \ln D_{ii} - \frac{N}{2} \ln 2\pi$

Posterior
Input: $\mathcal{X}^*$,$\mathcal{X}$, α, L
$\mu^*(\mathcal{X}^*) = \mu(\mathcal{X}^*) + K(\mathcal{X}^*, \mathcal{X})\alpha$ ▷ posterior mean (prediction)
$v = L \setminus \left(D^{-1} K^T(\mathcal{X}^*, \mathcal{X})\right)$
$\Sigma^* = K(\mathcal{X}^*, \mathcal{X}^*) - v^T v$ ▷ posterior covariance
return μ^*, Σ^*, h

---

otherwise, we say that the current is leaving the node. Given an ordered pair of nodes (k, l) the voltage drop v_{kl} across the nodes k and l can be measured via $V_{kl}(t) = V_k(t) - V_l(t)$, where $V_k(t), V_l(t)$ are absolute electrical potentials. A path can be defined via node labels or line labels. A path is said to be closed if the first node index equals the second node index of the last path segment. Two fundamental laws exist, known as Kirchhoff's current and voltage law.

Kirchhoff's Current Law: The sum of all currents entering or leaving a node must be zero at any instant of time. That is, for node l we have for $I_{1l}(t), \dots, I_{1n}(t)$

$$\sum_{k=1}^{n} I_{kl}(t) = 0, \ \forall t \tag{3.52}$$

Kirchhoff's Voltage Law: The sum of all voltage drops around any closed path of the circuit is zero at any instant of time. That is, for the closed path $v_{12}(t), \ldots, v_{k,k+1}(t) \ldots v_{n1}(t)$

$$\sum_{k=1}^{n} V_{kk+1}(t) = 0, \text{ where } n+1 = 1, \ \forall t \tag{3.53}$$

In the given graph-theoretical framework, we have described electrical circuits via their graph structure including a consistency requirement formulated in terms of Kirchhoff's laws. This constitutes an algebraic structure between voltages and currents in the graph. On top of the algebraic structure, an additional differential structure can be included in the graph. This is due to passive electrical components. The set of first-order differential equations arising from this construction is called the state-space representation, where functions are divided into states and inputs.

Passive Components We give the basic functional relationships for passive components as can be found in [100], [101]. It should be noted that the given expressions are approximations of the real behavior of electrical components in the fields, usually valid under, so-called normal conditions or in the linear regime. More formally, we assume that passive components can be described by constants. In particular, physical effects like thermal degradation or drifting due to high-induced energy can treated individually by parameter values constant in corresponding periods.

The resistance R of a resistor is defined by the voltage drop V_R across the resistor and the current I_R through the resistor by Ohm's law

$$V_R(t) = RI_R(t) \tag{3.54}$$

As a consequence of (3.52) and (3.53), the equivalent resistance R_{eq} of resistances $R_1, \ldots R_i, i \in \mathbb{N}$, can be computed as $R_{eq} = R_1 + \cdots + R_i$ if in series and as $R_{eq} = 1/R_1 + \cdots + 1/R_i$ if in parallel.

The capacity C of a capacitor is defined by the current $I_C(t)$ entering the capacity and the temporal change of voltage across the capacity, i.e.,

$$I_C(t) = C\frac{d}{dt}V_C(t) \tag{3.55}$$

Via (3.52) and (3.53), the equivalent capacity C_{eq} of capacities $C_1, \ldots C_i, i \in \mathbb{N}$, can be computed as $C_{eq} = 1/C_1 + \cdots + 1/C_i$ if in series and as $C_{eq} = C_1 + \cdots + C_i$ if in parallel.

The inductivity L of an inductor is defined by the voltage V_L across the inductor and the temporal change of current through the inductor, i.e.,

$$V_L(t) = L\frac{d}{dt}I_L(t) \tag{3.56}$$

Following from (3.52) and (3.53), the equivalent inductivity L_{eq} of inductivities $L_1, \dots L_i, i \in \mathbb{N}$, can be computed as $L_{eq} = L_1 + \dots + L_i$ if in series and as $L_{eq} = 1/L_1 + \dots + 1/L_i$ if in parallel.

3.3.2 Linear Operator Kernel Design

A generic state-space equation for an electrical circuit model can be written as

$$\dot{x} = Ax(t) + Bx(t) \tag{3.57a}$$

$$y = Cx(t) + Dx(t) \tag{3.57b}$$

with state vector $x(t)$, input (or control) vector $u(t)$, state matrix A, input matrix B, output matrix C and feedforeward matrix D. It should be noted that (3.57a), can be written using a linear operator $\mathcal{L}_t$, i.e.,

$$\mathcal{L}_t[x(t)] = \left(\frac{d}{dt} - A\right)[x(t)] = Bu(t) \tag{3.58}$$

Often synonymously called linear, in the context of this work the difference between a linear and an affine map should be emphasized. For a variable or function x, the map

$$x \;\mapsto\; ax + b \tag{3.59}$$

is called affine if $a, b \neq 0$ and linear if $a \neq 0, b = 0$.

3.3.2.1 Linear Operator Kernel

In [99], the application of linear operators (3.34) given by differential equations to GPs is presented. The author focuses on providing a solution for the differential equation which is illustrated by a mathematical toy model. We transfer this construction to the usage of digital representations for real-world problems. Contrarily to [99], we refer to models like these as linear operator PIGP instead of PIGP as there will be another class of PIGP proposed in this work.

Let $x_i(t) \sim \mathcal{GP}[x_i(t)]$ be an arbitrary state of (3.57) that is represented by a GP. Using the state-space equations (3.57), according to [99], we write

$$\mathcal{GP}[x_1(t_1)] = x_1(t_1) \;, t_1 \in \mathcal{T}_1 \tag{3.60a}$$

$$\frac{d}{dt_2}\mathcal{GP}[x_1(t_2)] = a_1 x_1(t_2) + a_2 x_2(t_2) + \dots + b_1 u_1(t_2) + \dots \;, t_2 \in \mathcal{T}_2 \tag{3.60b}$$

for the joint GP, where $\mathcal{T}_1, \mathcal{T}_2$ denote the sets of time instances at which the measurements of states and inputs are performed. In general, $\mathcal{T}_1$ and $\mathcal{T}_2$ do not have to coincide. From (3.60b), we can define a linear operator

$$\mathcal{L}_{t_2} = \frac{d}{dt_2} \tag{3.61}$$

acting on state $x_1(t_2)$. If $\mathcal{T}_1 = \mathcal{T}_2$, we can define the linear operator as

$$\mathcal{L}_t^{a_1} = \frac{d}{dt} - a_1 \tag{3.62}$$

where $t = t_1 = t_2$. Formulated like in (3.62), the linear operator $\mathcal{L}_t^{a_1}$ contains the parameter a_1 of the state-space equations. Let $x_1(t)$ be represented by a GP with kernel $K_{x_1}(t_1, t_1')$, for $t_1, t_1' \in \mathcal{T}_1$. According to (3.34), we can write the kernel of the system (3.60) as

$$K_{x_1,\mathcal{L}_{t_2}[x_1]} = \begin{pmatrix} K_{x_1}(t_1, t_1') & \mathcal{L}_{t_2'} K_{x_1}(t_1, t_2') \\ \mathcal{L}_{t_2} K_{x_1}(t_2, t_1') & \mathcal{L}_{t_2}\mathcal{L}_{t_2'} K_{x_1}(t_2, t_2) \end{pmatrix} \tag{3.63}$$

or

$$K_{x_1,\mathcal{L}_t^{a_1}[x_1]} = \begin{pmatrix} K_{x_1}(t, t') & \mathcal{L}_{t'}^{a_1} K_{x_1}(t, t') \\ \mathcal{L}_t K_{x_1}(t, t') & \mathcal{L}_t^{a_1}\mathcal{L}_{t'}^{a_1} K_{x_1}(t, t) \end{pmatrix} \tag{3.64}$$

with $t_1, t_1' \in \mathcal{T}_1, t_2, t_2' \in \mathcal{T}_2$ or $t, t' \in \mathcal{T}$. Note that in (3.64), the parameter a_1 of system (3.60) turns into a hyper-parameter of the GP.

We call the state $x_1(t)$ the *coupling signal*, as it is the quantity used for coupling the set of equations by the linear operator. Note that any other state or input might also be used as the coupling object. For example, also the linear operator $\mathcal{L}^{a_2} = a_2$ could be defined, yielding equivalent expressions for (3.64). In this case, $\dot{x}_1(t)$ appears on the right-hand side of (3.60b) and has to be measured explicitly. Note that, the GP model is trained on the right-hand side of (3.60), implying a sensitivity with respect to these qualities. When not measuring $\dot{x}_1(t)$ directly, obtaining $x_1(t)$ and taking numerical derivatives might cause inaccuracies regarding data quality. In machine learning jargon commonly referred to as "garbage in, garbage out", care has to be taken when processing data under consideration. This holds especially for applications of digital representation as noisy measurements are a key aspect thereof. Note that there are no numerical derivatives present if $x_1(t)$ is selected as the coupling signal. This is because derivatives are calculated analytically due to the representation of $x_1(t)$ as a GP.

3.3.2.2 Limitations of Linear Operator Kernel

Most generally, i.e., with the least amount of prior system knowledge, the PIGP based on the linear operator (3.60) can be stated as

$$\mathcal{GP}[x_1(t)] = x_1(t) \tag{3.65a}$$

$$\mathcal{L}_t^{a_1}[\mathcal{GP}[x_1(t)]] = \mathrm{res}(t) \tag{3.65b}$$

with

$$\mathrm{res}(t) = a_2 x_2(t) + \cdots + a_n x_n(t) + b_1 u_1(t) + \cdots + b_m u_m(t) \tag{3.66}$$

As $\mathrm{res}(t)$ remains on the right-hand side of (3.65) all quantities present have to be known. In particular, this includes system parameters $a_2, \ldots, a_n, b_1, \ldots b_m \in \mathbb{R}$. While states and inputs might be measurable in practical applications, the precise determination of system parameters is quite challenging during operation. Although in principle every component could be analyzed individually, practically speaking this produces a huge overhead of additional measurements. Despite lucky circumstances of trivial $\mathrm{res}(t)$, this prevents an application in any practical setup.

3.3.3 Balancing, Noise Hyper-Parameter and Regularization of Linear Operator Kernel

We propose the application of the multi-output kernel design developed in Sec. 3.2 to linear operator kernels. This is crucial for well-defined and reliable applications of linear operator kernels to digital representations for energy systems.

In common machine learning applications, feature normalization is archived by affine transformations like min-max-, mean-, or Z-score normalization [39]. However, this is not applicable in the given context as affine normalization schemes violate the system of equations (3.65). Evident from (3.65), only the linear part of each scaling is consistent with (3.65). That is the multiplication by a scalar. We denote the normalizer of signal s by $n_{\{s\}} \in \mathbb{R}$. Then, (3.65) can be written as

$$\mathcal{GP}\left[\frac{x_1(t)}{n_{\{x_1(t)\}}}\right] = \frac{x_1(t)}{n_{\{x_1(t)\}}} \tag{3.67a}$$

$$\frac{n_{\{x_1(t)\}}}{n_{\{\mathrm{res}(t)\}}}\left(\frac{d}{dt} - a_1\right)\mathcal{GP}\left[\frac{x_1(t)}{n_{\{x_1(t)\}}}\right] = \frac{a_2 x_2(t) + \cdots + b_m u_m(t)}{n_{\{\mathrm{res}(t)\}}} \tag{3.67b}$$

The normalization factor can be included in the operator by linearity via redefining

$$\left(\frac{d}{dt} - a_1\right) \mapsto \frac{n_{\{x_1(t)\}}}{n_{\{\mathrm{res}(t)\}}}\left(\frac{d}{dt} - a_1\right) \tag{3.68}$$

Suppose the noise level can be pre-determined, i.e., $\sigma_{\{x_2(t)\}}, \ldots, \sigma_{\{x_n(t)\}}, \sigma_{\{u_1(t)\}}, \ldots, \sigma_{\{u_m(t)\}}$ are given. According to uncertainty propagation [102], for the noise level of $\mathrm{res}(t)$ we have

$$\sigma_{\mathrm{res}(t)} = \sqrt{\sum_{i=2}^{M}\left(a_i \sigma_{\{x_i(t)\}}\right)^2 + \sum_{j=1}^{N}\left(b_j \sigma_{\{u_j(t)\}}\right)^2} \tag{3.69}$$

The noise hyper-parameter $\sigma_{\{x_1(t)\}}$ and $\sigma_{\{\mathrm{res}(t)\}}$ have to be balanced according to

$$\sigma_{\{x_1(t)\}} \mapsto \frac{\sigma_{\{x_1(t)\}}}{n_{\{x_1(t)\}}} \quad \text{and} \quad \sigma_{\{\mathrm{res}(t)\}} \mapsto \frac{\sigma_{\{\mathrm{res}(t)\}}}{n_{\{\mathrm{res}(t)\}}} \tag{3.70}$$

Despite a two-sided regularization which should always be performed, the precise regularization scheme depends on the individual use case. We give details in the corresponding applications provided in Ch. 5.

3.4 Affine Operator Kernel Design

As presented in [1], [5], we propose and develop a framework to extend the class of operators used for PIGP. Thereby, we overcome restrictions provided by linear operators. In Sec. 3.4.1, we propose the construction of affine operator kernels. This enables a wide range of real-world applications. In Sec. 3.4.2, we propose a detailed algorithmic scheme for a well-defined and numerical stable application.

3.4.1 Affine Operator Kernel

Distinct GPs can be added by a direct sum (3.32). That is, for two independent functions $f(t_1) \sim \mathcal{GP}[f(t_1)]$ and $g(t_2) \sim \mathcal{GP}[g(t_2)]$, where $\mathcal{T}_i,\ i = 1, 2$ denotes the sets of time coordinates, the joint kernel is given by

$$K_f(\mathcal{T}_1, \mathcal{T}_1) \oplus K_g(\mathcal{T}_2, \mathcal{T}_2) = \begin{pmatrix} K_f(\mathcal{T}_1, \mathcal{T}_1) & 0 \\ 0 & K_g(\mathcal{T}_2, \mathcal{T}_2) \end{pmatrix} \tag{3.71}$$

Note that two non-independent signals can be represented by two independent GPs.

Although (3.71) seems to be a trivial statement for the modeling of two signals f and g as there are no correlations, i.e., non-zero off-diagonal sub-blocks, it allows the construction of a new class of operators applicable to GPs, called affine operators. Recall from (3.65) and (3.66) that the mathematical restriction of linearity implies a non-trivial $\mathrm{res}(t)$-term. Using (3.71), the right-hand side of (3.65) can be decomposed such that also the signals present in (3.66) can be represented as GP. In other words, signals not serving as a coupling state via a linear operator and their associated system parameters can be included in the representation by the GP. Accordingly, the present system parameters turn into linear operators. Thus, we can write the system of equations with GP representations as

$$\mathcal{GP}[x_1(t)] = x_1(t) \tag{3.72a}$$
$$\vdots$$
$$\mathcal{GP}[x_n(t)] = x_n(t) \tag{3.72b}$$
$$\mathcal{GP}[u_1(t)] = u_1(t) \tag{3.72c}$$
$$\vdots$$
$$\mathcal{GP}[u_n(t)] = u_n(t) \tag{3.72d}$$
$$\left(\frac{d}{dt} - a_1\right)\mathcal{GP}[x_1(t)] - a_2\mathcal{GP}[x_2(t)] - \cdots - b_1\mathcal{GP}[u_1(t)] - \ldots = \tilde{\mathrm{res}}(t) \tag{3.72e}$$

where $\tilde{\mathrm{res}}(t)$ includes all states, inputs, and system parameters that should not be represented by the joint GP. Note that by construction, $\tilde{\mathrm{res}}(t) = 0$ is always possible. That is, the given construction might provide a complete decomposition of $\mathrm{res}(t)$. Moreover, arbitrarily many system parameters $a_1, \ldots, b_1, \ldots$ turn into hyper-parameters of the GP and are thus determinable in the training process.

We want to emphasize that the construction allows the addition of linear operators in general. In the given application, the second linear operator $\mathcal{L}$ is given by terms like for example, $\mathcal{L} = -a_i$ or $\mathcal{L} = -b_i$. In general, also adding differentiable operators is possible. Due to the focus on the application of state-space equations in this work, this is not explicitly addressed in the following.

We call states and inputs appearing on the left side of (3.72e) the *coupling signals*. Note that the corresponding system parameters appearing in the individual associated linear operators

$$\mathcal{L}_t = \left(\frac{d}{dt} - a_1\right),\ \mathcal{L}_{a_2} = -a_2,\ \ldots,\ \mathcal{L}_{b_1} = -b_1 \tag{3.73}$$

turn into hyper-parameters of the affine operator GP model. Thus, they are trainable quantities.

3.4.2 Balancing, Noise Hyper-Parameter and Regularization of Affine Operator Kernel

Similar to the linear case, the multiple signals entering the affine-operator GP model have to be balanced, i.e., normalized. While including normalization constants into the linear operator is a useful option when one signal is given by a GP, for systems consisting of more than one signal, a redefinition is crucial. Otherwise, each signal could not be normalized independently of the remaining ones leading to practical constraints. Accordingly, (3.72) turns into

$$\mathcal{GP}\left[\frac{x_1(t)}{n_{\{x_1(t)\}}}\right] = \frac{x_1(t)}{n_{\{x_1(t)\}}} \tag{3.74a}$$

$$\vdots$$

$$\mathcal{GP}\left[\frac{x_n(t)}{n_{\{x_n(t)\}}}\right] = \frac{x_n(t)}{n_{\{x_n(t)\}}} \tag{3.74b}$$

$$\mathcal{GP}\left[\frac{u_1(t)}{n_{\{u_1(t)\}}}\right] = \frac{u_1(t)}{n_{\{u_1(t)\}}} \tag{3.74c}$$

$$\vdots$$

$$\mathcal{GP}\left[\frac{u_n(t)}{n_{\{u_n(t)\}}}\right] = \frac{u_n(t)}{n_{\{u_n(t)\}}} \tag{3.74d}$$

$$\begin{aligned}\left(\frac{n_{\{x_1(t)\}}}{n_{\{\tilde{\mathrm{res}}(t)\}}}\left(\frac{d}{dt} - a_1\right)\right)\mathcal{GP}\left[\frac{x_1(t)}{n_{\{x_1(t)\}}}\right] - \frac{a_2 n_{\{x_2(t)\}}}{n_{\{\tilde{\mathrm{res}}(t)\}}}\mathcal{GP}\left[\frac{x_2(t)}{n_{\{x_2(t)\}}}\right] - \dots \\ - \frac{b_1 n_{\{u_1(t)\}}}{n_{\{\tilde{\mathrm{res}}(t)\}}}\mathcal{GP}\left[\frac{u_1(t)}{n_{\{u_1(t)\}}}\right] - \dots = \frac{\tilde{\mathrm{res}}(t)}{n_{\{\tilde{\mathrm{res}}(t)\}}}\end{aligned} \tag{3.74e}$$

Balancing the pre-determined noise hyper-parameter $\sigma_{\{x_1(t)\}}, \dots, \sigma_{\{u_1(t)\}}, \dots, \sigma_{\{\tilde{\mathrm{res}}(t)\}}$ yields

$$\frac{\sigma_{\{x_1(t)\}}}{n_{\{x_1(t)\}}}, \dots, \frac{\sigma_{\{u_1(t)\}}}{n_{\{u_1(t)\}}}, \dots, \frac{\sigma_{\{\tilde{\mathrm{res}}(t)\}}}{n_{\{\tilde{\mathrm{res}}(t)\}}} \tag{3.75}$$

There is one peculiarity for the noise level within affine operator kernels. Suppose the decomposition by the affine construction is extensive, i.e., $\tilde{\mathrm{res}}(t) = 0\ \forall t$. This means that all signals present are represented by a GP, leaving the zero signal for the coupling in (3.74e) by the affine differential operator. Zero noise level for this signal leads to numerical issues. Interpreted physically, for a zero noise level of (3.74e) we demand the coupled GP to fulfill the state-space equations exactly. Having noisy signals in general, this is already a contradiction. Despite that, in exact simulation scenarios, where the differential operator generating the use case is applied to the GP, zero noise leads to an ill-posed kernel. Consequently, in the case of $\tilde{\mathrm{res}}(t) = 0\ \forall t$ a small amount of noise $\sigma_{\{\tilde{\mathrm{res}}(t)=0\}}$, i.e., Tikhonov regularization, has to be utilized for physical consistency and numerical stability.

A two-sided regularization is crucial before applying Tikhonov regularization for the coupling part (3.74e). The full regularization scheme is specified by the individual use case. We give details in Ch. 5.

3.4.3 Affine Operator Kernel Algorithmic Solution

We summarize the given construction of affine operator kernels in (3.VII).

3.VII. ***Proposed Affine Operator Construction***

Suppose we have state-space equations with component

$$\frac{d}{dt}x_1(t) - a_1x_1(t) = a_2x_2(t) + \cdots + b_1u_1(t) + \cdots = res(t)$$

To decompose the individual signals contained in res(t) and represent them by individual GP, we utilize the direct sum $\oplus$. The balanced system reads

$$\begin{aligned}
\mathcal{GP}\left[\frac{x_1(t)}{n_{\{x_1(t)\}}}\right] &= \frac{x_1(t)}{n_{\{x_1(t)\}}}\\
&\vdots\\
\mathcal{GP}\left[\frac{x_n(t)}{n_{\{x_n(t)\}}}\right] &= \frac{x_n(t)}{n_{\{x_n(t)\}}}\\
\mathcal{GP}\left[\frac{u_1(t)}{n_{\{u_1(t)\}}}\right] &= \frac{u_1(t)}{n_{\{u_1(t)\}}}\\
&\vdots\\
\mathcal{GP}\left[\frac{u_n(t)}{n_{\{u_n(t)\}}}\right] &= \frac{u_n(t)}{n_{\{u_n(t)\}}}\\
\mathcal{L}_t\left[\mathcal{GP}\left[\frac{x_1(t)}{n_{\{x_1(t)\}}}\right]\right] + \mathcal{L}_{a_2}\left[\mathcal{GP}\left[\frac{x_2(t)}{n_{\{x_2(t)\}}}\right]\right] + \cdots + \mathcal{L}_{b_1}\left[\mathcal{GP}\left[\frac{u_1(t)}{n_{\{u_1(t)\}}}\right]\right] - \ldots &= \frac{r\tilde{e}s(t)}{n_{\{r\tilde{e}s(t)\}}}
\end{aligned}$$

with operators

$$\mathcal{L}_t = \left(\frac{n_{\{x_1(t)\}}}{n_{\{r\tilde{e}s(t)\}}}\left(\frac{d}{dt} - a_1\right)\right),\ \mathcal{L}_{a_2} = -\frac{a_2n_{\{x_2(t)\}}}{n_{\{r\tilde{e}s(t)\}}},\ \ldots,\ \mathcal{L}_{b_1} = -\frac{b_1n_{\{u_1(t)\}}}{n_{\{r\tilde{e}s(t)\}}}$$

In particular, for basis kernels $K_{x_1}(t,t'),\ldots,K_{u_1}(t,t'),\ldots$ for the individual normalized signals $x_1(t)/n_{\{x_1(t)\}},\ldots,u_1(t)/n_{\{u_1(t)\}},\ldots$, we have

$$\begin{pmatrix}
K_{x_1}(t,t') & 0 & 0 & 0 & 0 & \mathcal{L}_{t'}\left[K_{x_1}(t,t')\right]\\
0 & K_{x_2}(t,t') & 0 & 0 & 0 & \mathcal{L}_{a_2}\left[K_{x_2}(t,t')\right]\\
0 & 0 & \ddots & 0 & 0 & \vdots\\
0 & 0 & 0 & K_{u_1}(t,t') & 0 & \mathcal{L}_{b_1}\left[K_{u_1}(t,t')\right]\\
0 & 0 & 0 & 0 & \ddots & \vdots\\
\mathcal{L}_t\left[K_{x_1}(t,t')\right] & \mathcal{L}_{a_2}\left[K_{x_2}(t,t')\right] & \ldots & \mathcal{L}_{b_1}\left[K_{u_1}(t,t')\right] & \ldots & \begin{array}{r}\mathcal{L}_t\mathcal{L}_{t'}\left[K_{x_1}(t,t')\right] + \mathcal{L}_{a_2}^2\left[K_{x_2}(t,t')\right]\\ + \cdots + \mathcal{L}_{b_1}^2\left[K_{u_1}(t,t')\right] + \ldots\end{array}
\end{pmatrix}$$

for the affine operator kernel.

3.5 Correlated Kernel Design

When considering signals in the energy domain at a system level, it turns out that certain signals are not independent of each other. Depending on the operational regime, correlations among the individual constituencies might exist. This holds, for example, for temperature measurements at distinct locations within a room or for voltage magnitudes in a balanced power grid. In both cases, physical constraints imply similar signal characteristics yielding a correlation between them. Thus, assuming the same underlying GP model combined with a correlation term is a straightforward approach. GPs encoding this property are based on separable kernels, called correlated kernels hereafter. While the mathematical term separable describes the construction, the term correlated indicates the physical interpretation. We refer to models within this class as Corr-GP.

We propose the application of Corr-GPs to provide digital representations in the energy domain in Sec. 3.5.1 and define a concrete application scheme in Sec. 3.5.2

3.5.1 Separable Kernel for System Modeling in Gaussian Processes

We give the construction of separable kernels at the level of kernel matrices [103]. For a derivation in terms of RKHS, we refer to [103].

Suppose multiple signals display the same characteristics. Consequently, they can be described by the same kernel $K(\cdot,\cdot)$. We established in Sec. 3.1.2.3, that the GP training consists of hyper-parameter training and conditioning on the explicit data set. By the same kernel $K(\cdot,\cdot)$, we refer to the selected covariance function and its hyper-parameter values. Each kernel representing the individual signal is conditioned on its own data set.

The key mathematical operation describing the correlation among the signals is the tensor product $\otimes$ with a symmetric positive semi-definite matrix. Using (3.33), we have

$$B^{[d]} \otimes K(\mathcal{T},\mathcal{T}') = \begin{pmatrix} b_{11} \dots b_{1d} \\ \vdots \ \ddots \ \vdots \\ b_{d1} \dots b_{dd} \end{pmatrix} \otimes K(\mathcal{T},\mathcal{T}') = \begin{pmatrix} b_{11}K(\mathcal{T},\mathcal{T}') \dots b_{1d}K(\mathcal{T},\mathcal{T}') \\ \vdots \qquad \ddots \qquad \vdots \\ b_{d1}K(\mathcal{T},\mathcal{T}') \dots b_{dd}K(\mathcal{T},\mathcal{T}') \end{pmatrix} \tag{3.76}$$

for symmetric positive semi-definite matrix $B^{[d]}$ of dimension d.

The matrix $B^{[d]}$, called *coregionalization matrix*, describes the coupling between the d signals and can be interpreted as follows. The entries $\{b_{ij}|i,j=1,\dots,d\}$ of $B^{[d]}$ are trainable hyper-parameters of the given GP. Diagonal entries $\{b_{ii}|i=1,\dots,d\}$ describe the self-contribution of each individual signal to the joint GP model. Off-diagonal terms $\{b_{ij}|i\neq j; i,j=1,\dots,d\}$ account for cross-correlations among related signals. With this at hand, we may define two types of models, one being the generalization of the other.

Intrinsic Coregionalization Model (ICM) Let $K(\mathcal{T}, \mathcal{T}')$ be a kernel describing all signals under consideration. The ICM is defined as

$$K_{\mathrm{ICM}}(\mathcal{T}, \mathcal{T}') = B^{[d]} \otimes K(\mathcal{T}, \mathcal{T}') \tag{3.77}$$

This type of model is used in the case of a single class of correlated signals.

Linear Coregionalization Model (LCM) For multiple classes of correlated signals, several ICMs can be added. This type of model is called LCM and is given by

$$K_{\mathrm{LCM}}(\mathcal{T}, \mathcal{T}') = \sum_{q} B_q^{[d]} \otimes K_q(\mathcal{T}, \mathcal{T}') \tag{3.78}$$

where q denotes the number of considered similarity classes.

3.5.2 Balancing, Noise Hyper-Parameter and Regularization of Correlated Kernel

The kernels of the ICM and the LCM are designed to be applied to signals sharing the same characteristics including similar order of magnitudes. Thus, balancing all signals with the same factor is a natural choice. As this type of model and its corresponding hyper-parameter contained in $B^{[d]}$ does not have a physical interpretation, rescaling via an affine transformation is in principle possible. For the models developed in this work, we used linear rescaling. For the same reason, rescaling each signal by an individual factor is also possible. As the hyper-parameters of $B^{[d]}$ and its diagonal entries in particular describe the relative scaling of each signal, there is no need for individual rescaling.

In this work, for correlated kernels, a pre-determination of noise hyper-parameter was not performed. In the given applications as described in Ch. 4 and Ch. 6, a pre-determination is practically hard to realize. Second, in the given implementation using GPy [97], the noise hyper-parameter heavily interferes with the implemented regularization scheme as can be understood as follows. Due to the lack of direct physical interpretation of hyper-parameters of GP models with correlated kernels, altering the initial task is acceptable if numerical stability can be guaranteed. Additionally to Tikhonov regularization for the basis kernel $K(\mathcal{T}, \mathcal{T}')$, another Tikhonov regularization term is added to the correlation matrix $B^{[d]}$ as the default option in [97]. Thus, additional contributions to the diagonal coming from the signal's noise play numerically a minor role.

3.6 Digital Representation Design by Multi-Output Gaussian Processes

A summary and contextualization of the proposed models in this chapter can be seen in Fig. 3.1. The developed models with linear, affine, and correlated kernels are classified according to their inherent system knowledge. As shown in Fig. 3.1, Corr-GPss encode the least prior

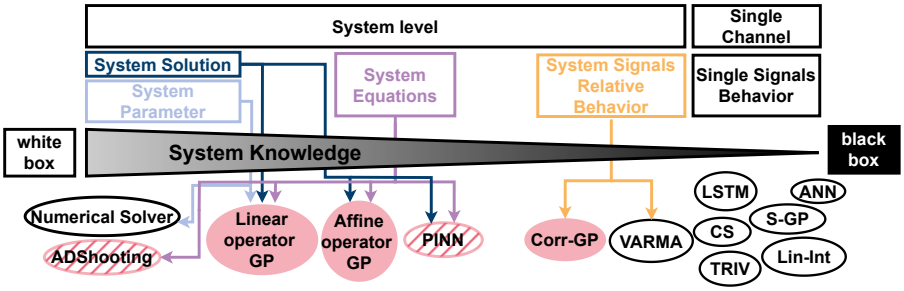

Figure 3.1: Embedding and contextualization of digital representations provided by linear, affine, and correlated kernel GPs.

information about the underlying system. This might be, for example, the local power system topology relevant in Ch. 4 or similar temperature measurements in Ch. 6. Based solely on the relative behavior of systems signals, Corr-GPs models are comparable to VARMA models in terms of the degree of system information included. From a model-building perspective, the construction allows for tailoring of the Corr-GP model to the specific application which has two advantages. First, compared to VARMA models, Corr-GPs are kernel methods that have the universal property instead of pure statistical weights. Second, the coupling by the coregionalization matrix has a physical interpretation, yielding a user-accessible adaptability to the specific application.

On the other side of the spectrum, linear operator GPs are based on a system's description via state-space equations. An application as a power electronics parameter estimator for the toy problem of a buck converter with LC-topology is presented in Sec. 5.2. Compared to solver-based algorithms, linear operator GPs encodes the system's equation but not their solutions. Actually, the solver is replaced by the GP.

Compared to linear operators, less prior information is needed for affine operator GP. Although linear and affine operators utilize the description of the system by state-space equations, less information on system parameters is required. Precise knowledge of system parameters is often not given in practical applications. Unless explicitly measured, allowing system parameters to be trainable hyper-parameters of the GP models enables an application to a widespread class of use cases. The most prominent alternatives to affine operator GPs are PINNs. In general, PINNs allow any kind of operator to be included in the loss function, not only linear ones. Contrary to GPs, where the system's description is encoded into the core process via the kernel, in PINNs only the loss function is affected. The neural network architecture plays no role in this regard. Although not an issue in general, in certain circumstances this might have some drawbacks. This is because no distinct consideration of signal is possible as the individual contributions are added to the loss function.

4

A Digital Twin for Data Quality in Power Systems

In this chapter, we propose a DT for data quality in power systems ICT platforms. We develop a DT based on Corr-GPs of local subsystems of power systems with the purpose of signal recovery for large missing periods of data. We describe the automated application as a bad data detection and signal recovery DT of an ICT platform monitoring a power system operated at FZJ. In Sec. 4.1, we start with contextualizing and motivating the provided DT application. The ICT platform of the considered power system operated at the LLEC at FZJ is introduced in Sec. 4.2. The realization of the DT as a signal recovery tool is developed in Sec. 4.3. The case study is described in Sec. 4.4. We evaluate results of the application of a Corr-GP DT in Sec. 4.5. We provide details on the algorithm's performance and accuracy which is independent of the missing interval length within the proposed application. Detailed insights about key characteristics of the DT that generate practical benefits for system operators are provided. A self-aware failing indication allows system operators a direct evaluation of the recovered data. In Sec. 4.6, the online application of the Corr-GP models is described. To finalize this chapter, we conclude and characterize the presented DT in Sec. 4.7. This chapter is based on the publications [3], [8].

4.1 Introduction

4.1.1 Motivation

Nowadays, power systems are monitored with an increasing number of measurement devices, aimed at providing the system operator with fundamental information about the operating conditions of the system. From smart meters and power quality meters [104], largely applied in low-voltage networks, to PMUs [105], mostly used in transmission and distribution systems, every day huge datasets are generated in the field. In control centers, this large amount of data is provided as input to routines for monitoring, e.g., state estimation [105] or the estimation of line parameters [106], automation, e.g., harmonic analysis [29] and control, e.g., for voltage stability [107] which strongly rely on these measurements. Within this framework, it is evident that the successful operation of these algorithms depends not only on the number and accuracy of measurements but also on the data quality [108]. In particular, bad data is the term commonly used when referring to corrupted data and data loss. Such issues can originate from malfunctioning of the devices in the monitoring chain (instrument transformers and measurement devices), issues with the communication network, or failures when storing the data. An example with reference to a PMU-based measurement system is reported in [31], while in [109] the problem of outliers in a Supervisory Control And Data Acquisition (SCADA) based monitoring environment is addressed. Bad data can play a key role in the monitoring of health conditions of components where, as underlined in [28], data considered incorrect in other fields are indicators of faults in the machinery. Moreover, it is worth mentioning that bad data can also originate from malicious cyberattacks, which are causing an increasing concern [110]. Since the issues mentioned above are getting more and more common, being able to recover a missing section of a signal would help the system operator overcome the issues due to bad data and ensure the proper operation of network managing routines.

4.1.2 Related Work

Detecting bad data might be trivial for certain anomalies, like, for example, missing entries or zeros but could be challenging in case of more subtle anomalies, such as measurements that are apparently reasonable but characterized by a large error. Different approaches have been considered to address such issues. In [111], a computationally efficient largest normalized residual method has been presented for the identification of bad data in large-scale power systems. In [112], a graph signal processing-based bad data detection method for three-phase distribution systems has been proposed. In [113], bad data detection has been approached through backpropagation of neural networks incorporating K-means clustering.

Signal recovery methods applied to large-scale monitoring systems can roughly be classified into single-channel and multi-channel algorithms [114]. Classical single-channel methods range from expectation-maximization methods [115] to mean methods [114]. As linear interpolation, denoted by *Lin-Int* in the following, is a common representative of those methods, we use it to evaluate the Corr-GP algorithm. Modern single-channel approaches include both regression and matrix completion algorithms. In [116], the authors addressed the problem of data losses

in a monitoring framework composed of PMUs utilizing OLAP-t. Such an algorithm allows the recovery of signals exhibiting temporal patterns subjected to prolonged data outages. Alternatively, based on the method of multipliers in the alternating direction, in [117] the recovery of missing PMU data has been addressed by exploiting the low-rank property of the corresponding measurement matrix. Addressing the matrix completion algorithms, we evaluate the proposed approach against a compressive sensing-based recovery technique, abbreviated as *CS* in the following, which is presented in [84]. One such technique, which has been shown to provide the best performance in case of varying signals, uses a discrete cosine transform to reformulate the recovery problem on a sparse basis, such that a norm-1 minimization algorithm can be applied to recover not-null components. Due to the lack of information within larger missing intervals, single-channel methods have a natural limitation.

A straightforward cross-channel approach utilizes related signals. Referred to as *Rep-phase* in this work, such as other phases of a multi-phase system. Although this approach is highly dependent on the balance of the power system and is not universally applicable, any dedicated algorithm has to outperform this trivial method. More elaborated multi-channel algorithms include the power system's graph structure, extended matrix completion algorithms, as well as machine learning techniques. Similarly to [116], [117], but combined with Kirchhoff's laws and three-phase circuit relations as regularization terms, in [118] the recovered values are given by using cross-correlations of synchrophasor measurements to complete a low-rank tensor. In [119], the authors considered the spatial and temporal graph to define the correlations between system components and quantities, before applying a regression neural network for the recovery of missing data in large power systems. Similarly, in [118] cross-correlations of synchrophasor measurements are recovered by the completion of a low-rank tensor using Kirchhoff's laws and three-phase circuit relations as regularization terms. In [120], a jointly low-rank tensor-completion is applied to recover data from logistic systems. In [117], the recovery of missing PMU data has been addressed by exploiting the low-rank property of the corresponding measurement matrix. In [121], a network expectation-maximization algorithm is proposed to recover data in spatiotemporal applications for small missing intervals. Using neural networks including graph structures, missing power system synchrophasors caused by communication delays are recovered as reported in [122]. In [123], a generative adversarial network is used for missing data reconstruction.

To summarize, single-channel recovery algorithms capture typical signal behaviors and are mainly capable of dealing with short missing data intervals. Multi-channel approaches can incorporate simultaneously appearing information but might be limited by unknown system information in practical applications. For example, in [123], the entire power system has to be known first, a simulation model has to be built on top, and a decent amount of scenarios, e.g., 12000 in the application in [123], has to be simulated to generate training data. In other words, the need for big data in artificial intelligence might be a limitation in a practical setting. Moreover, a changing system topology, which once in a while appears in real power systems, can not be tracked without dedicated effort. On top, the inherent strong dependency of artificial intelligence on the specific data set, i.e., on a specific regularisation scheme, makes universal application difficult. Consequently, there is a need for a multi-channel approach based only on local system topology that is capable of recovering large missing intervals from a relatively small amount of known data.

Similar to the given Corr-GP approach, applications in different domains including distinct domain-specific peculiarities, have been investigated [60], [124]. In [60], a Corr-GP is utilized for model building of crowdsourced traffic data. With a focus on real-time information processing, the application of data collection from a weather sensor network is presented in [124].

4.1.3 Preliminary Remarks on Technical Realizations

We motivate the selected Corr-GP algorithm for the application as a signal recovery tool for power system signals. Single missing values might be recovered easily by single-channel methods. This is because power system signals do not vary rapidly under normal operational conditions with respect to usual reporting rates. For larger missing periods this certainly does not hold. This is because generation and loads might change quite significantly during this period. Thus, without any additional simultaneous information providing insights during this period, single-channel algorithms are naturally limited. A quite obvious way to incorporate simultaneous information is to utilize the knowledge about the power system, i.e., the grid topology and the implications for the respective power system signals. However, the drawback of working at the system level, i.e., the inclusion of physical equations describing the power grid, is the need for full system knowledge. Phrased differently, if a quantity appears in the system equations, it also has to be obtained by system operators from physical measurements. In practice, this might be challenging. Although system operators monitor a significant part of the grid, usually they do not monitor it at a level of detail needed for a descriptive and complete system model. The solution to this, as provided in this chapter, is a DT based on only local system topology. From a data collection perspective, this means also that only local measurements have to be provided for the DT.

4.1.4 Contribution

We propose a Corr-GP framework for the application within the field of power systems' data recovery. First, we provide a construction of single and multi-channel models for use in power systems, utilized for signal recovery. We then emphasize cross-channel information transfer that encodes the local power system's topology. In particular, we show that the amount of needed data to recover a signal is comparable to the desired signal to be recovered. Thus, the Corr-GP can handle large periods of missing data, while getting along with comparably small amounts of training data. We propose, analyze and demonstrate the signal recovery algorithm's self-aware failing indication, yielding trustworthy recovery results. More precisely, we show that the approach allows for a direct and inherent evaluation of performance, which can be utilized as a possibly automated reliable recovery procedure applied to large databases. We demonstrate that the results provided by the Corr-GP approach are independent of the length of the missing window if sufficient local cross-channel information is available. Moreover, supported by a detailed mathematical analysis, we show that the approach is almost unconstrained within the desired application. The entire evaluation and analysis of the above statements is performed using real data collected from the field. We apply the proposed framework to the LLEC. We propose a modular tool for bad data detection and the operation of the developed Corr-GP algorithm. The structure of the tool is presented generically, to emphasize its applicability to different types of ICT platforms with specific references to real scenarios.

4.2 ICT Platforms for Energy Systems

4.2.1 The ICT Platform of the Living Lab Energy Campus

The ICT platform of FZJ is based on the open-source IoT platform FIWARE [125]. A schematic representation of its implementation in the LLEC framework is shown in Fig. 4.1. Via an MQTT broker, the platform allows gathering data from various components in the field, e.g., PV systems, batteries, monitoring devices, and so on in real-time. Then, via the IoT agent, data is converted into the NGSIv2 data format and sent via REST API to the FIWARE Orion Context Broker (OCB). Here, only the most updated data are stored in the MongoDB database, while historical data is stored in the InfluxDB database. Since this large-scale framework is used to store data from different components and sensors, the internal structure of the databases consists of multiple subdatabases customized to properly store data from specific components. It is worth noting that different devices and sensors are typically characterized by different reporting rates. For example, temperature sensors installed in the rooms of the FZJ campus provide change-of-value measurements, i.e., no fixed reporting rate can be given. Instead, the power system data considered in this chapter, RMS voltages, and currents collected from the energy meters installed in the field, are provided with a fixed reporting rate of 5 minutes. All these data are first saved independently in the MongoDB, overwriting the data stored in the previous reporting interval, and then saved in the InfluxDB, concatenated to the corresponding set of historical data. In addition to the specific quantities for each device/sensor, the metadata structure is also composed of more general information like timestamps, unique identification number of the device/sensor, location, etc. External applications and services can read/write data directly from the databases. In particular, the ICT platform is modular, which allows different applications to run independently, suppressing malfunctioning due to chain effects.

4.2.2 Power system at FZJ

Over the campus, the power distribution is carried out on the 10 kV level via two main busbars. In total, 136 three-phase medium-voltage/low-voltage (MV/LV) transformers are used to supply the low-voltage loads on the 400 V level. On each transformer, the RMS values of voltages and currents on each phase are monitored by means of measurement devices in compliance with class 1 of the standard IEC 61036 [126], as well as the total active and reactive powers. The collected data are stored in internal databases, introduced in the previous section, for monitoring, control, and planning purposes.

4.3 Correlated Gaussian Processes for Signal Recovery in Power Systems

Published in [3], in this section, we propose the application of Corr-GP as a signal-recovering digital representation in power systems.

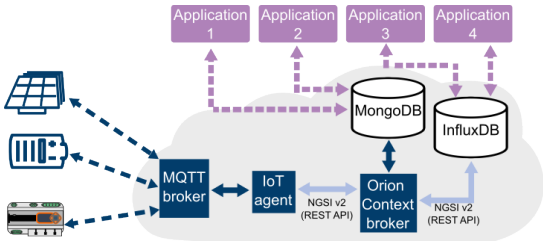

Figure 4.1: Simplified scheme of ICT FIWARE-based platform.

4.3.1 Gaussian Process Characteristics for Signal Recovery

Implied by definition (3.1), the Gaussian assumption in Gaussian processes holds at a function space level. Thus, a generic signal of interest interpreted as a time series does not have to explicitly obtain a Gaussian distribution, e.g., with respect to temporal structures or measurement errors. Nevertheless, it is worth underlining, that the Gaussian assumption on individual measurement values can still be valid for conventional measurement devices. As discussed in Sec. 3.1.3.2, the considered GPs are universal approximators, meaning any continuous function can be approximated by the selected GPs. Thus, within the proposed application there are no constraints according to the signals under consideration.

In the framework of signal recovery, we refer to the available or known data as the training set. The temporal component of the data to be recovered is called the test set. The posterior variance, σ_{post}, is given by the square root of the diagonal of the posterior covariance Σ_{post}. As discussed in Sec. 2.3.2, σ_{post} cannot be interpreted as an error interval in the usual sense, and thus cannot be associated with the model's accuracy. However, as being a major concern of this chapter, it provides a measure of the model's precision, as well as, a self-contained failing indication. Thus, on top of recovering missing data, the suggested models provide a reliable, in a sense self-aware technique, conveniently usable by system operators.

In the given power system application, due to the continuous power demand fluctuations, the signals of interest are non-deterministic. To this end, imposing a pre-build structure on the data by a non-zero prior mean function is not applicable due to the required flexibility and the considered time scale. Thus, we choose $\mu_{\text{prior}} = 0$ in this chapter.

4.3.2 Single-Channel Model for Voltage and Current Magnitudes

We start by selecting a single-channel kernel for the modeling of power system signals. A roughly varying voltage or current RMS signal, collected from the low-voltage side of a transformer with a reporting rate of one minute, as investigated in this chapter, is best described by a one-time continuous differentiable kernel, e.g., the $\nu = 3/2$ Matérn kernel. Recall from (3.28) that

$$K_{\text{M}^{3/2}}(t,t') = \sigma^2 \left(\frac{l + \sqrt{3}|t-t'|}{l} \right) \exp\left(\frac{-\sqrt{3}|t-t'|}{l} \right), \text{ for } t,t' \in \mathcal{T} \tag{4.1}$$

where $\mathcal{T}$ denotes the set of time variables, hyper-parameter $l \in \mathbb{R}^+$ defines the timescale of the given process and $\sigma \in \mathbb{R}^+$ accounts for a scaling in magnitudes' direction.

In order to take slowly varying trends into account, a smooth kernel such as (3.25) of the form

$$K_{\mathrm{RBF}}(t,t') = \sigma^2 \exp\left(-\frac{|t-t'|^2}{2l^2}\right), \text{ for } t,t' \in \mathcal{T} \tag{4.2}$$

has to be selected with $l, \sigma \in \mathbb{R}^+$ and time coordinates $\mathcal{T}$. It is worth to note two important aspects. First, due to the universal property, for data characterized by a high time resolution, the specific choice appears to be less important as the presented kernels yield a universal approximation. In the case of coarser data sets, the signals of interest tend to appear smoother, which implies the usage of smooth kernels. Second, as described in the next section, the key feature of the presented approach to power system signals is the coupling of different single-channel signals into a mutual multi-channel signal. As this appears to be the dominant effect of the proposed approach, the single-channel kernel selection becomes less significant in Corr-GPs.

4.3.3 Multi-Channel Model for Correlated Voltage and Current Magnitudes

Intrinsically induced by their local topological nature, power system signals display a certain degree of similarity depending on the level of balance present in the given system. The Corr-GP approach takes these characteristic cross-correlations between certain signals explicitly into account. In other words, by construction, the proposed method accounts for the balance induced by the power system topology while being flexible with respect to the considered type of signal. Developed in Sec. 3.5, the mathematical structure encoding this property is given by (3.76) as

$$B^{[d]} \otimes K(t,t') = \begin{pmatrix} b_{11} \dots b_{1d} \\ \vdots \ddots \vdots \\ b_{d1} \dots b_{dd} \end{pmatrix} \otimes K(t,t') \tag{4.3}$$

where the superscript d denotes the number of coupled signals and basis kernel $K(t,t')$ as in the previous Sec. 4.3.2.

Recall the interpretation of $B^{[d]}$. First, it should be noted that the entries $\{b_{ij}|i,j=1,\dots,d\}$ of $B^{[d]}$ are trainable hyper-parameters of the given model. That is, no prior knowledge about the quantitative coupling, i.e., the actual balance between power system signals, is required. Diagonal entries $\{b_{ii}|i=1,\dots,d\}$ describe the self-contribution of the signal to be recovered to the given model while off-diagonal terms $\{b_{ij}|i \neq j; i,j=1,\dots,d\}$ account for cross-correlations among related signals induced by the local system's topology. The latter is the key component of the proposed approach to incorporate simultaneous local power system information, improving the accuracy of recovery results. Emphasizing the consequences of this technicality regarding the given application, we have the following comments.

First, note that the possibility of including simultaneously appearing information of the power system of arbitrary size allows the Corr-GP approach to handle large missing periods. Practically constrained by the availability of sufficient simultaneous system information, theoretically, the

proposed approach can recover missing intervals of arbitrary size. Second, the different interpretation of elements of $B^{[d]}$ induces a particular choice of the data set from which the missing interval can be recovered. To train diagonal elements $\{b_{11}, b_{22}, \dots\}$, the available data on both sides of the missing period is selected. We considered the missing window to be in between two known intervals of almost the same size, as described in Sec. 4.4 and visualized for this case study in Fig. 4.3 and Fig. 4.4. It should be noted that this is done for a matter of convenience and by no means mandatory. In fact, having known data before and after the missing period is recommended as it conditions the given model on both ends of the missing window. For off-diagonal entries $\{b_{ij} | i \neq j; i, j = 1, \dots, d\}$, correlated signals are chosen, which are selected by the local power system's topology.

Selecting a basis kernel $K_{\text{basis}}(t, t')$ according to Sec. 4.3.2, both the ICM (3.77) and the LCM (3.78) can be defined. That is, for each similarity class, we have

$$K_{\text{ICM}}(t, t') = B^{[d]} \otimes K_{\text{basis}}(t, t') \tag{4.4}$$

which can be extended to

$$K_{\text{LCM}}(t, t') = \sum_q B_q^{[d]} \otimes K_{\text{basis},q}(t, t') \tag{4.5}$$

for q similarity classes.

4.3.4 Multi-Channel Recovered Signals

We summarize the given construction. For j signals $\left(s_1^*(t_1^*), \dots s_j^*(t_j^*)\right)$ to be recovered from i signals $(s_1(t_1), \dots s_i(t_i))$, the proposed approach reads

$$\begin{pmatrix} s_1^*(t_1^*) \\ \vdots \\ s_j^*(t_j^*) \end{pmatrix} = \sum_q \begin{pmatrix} b_{11}^q K^q(t_1^*, t_1) \dots & b_{1d}^q K^q(t_1^*, t_i) \\ \vdots & \vdots \\ b_{d1}^q K^q(t_j^*, t_1) \dots & b_{dd}^q K^q(t_j^*, t_i) \end{pmatrix} \cdot \begin{pmatrix} b_{11}^q K^q(t_1, t_1) + \sigma_{s_1}^2 \dots & b_{1d}^q K^q(t_1, t_i) \\ \vdots & \vdots \\ b_{d1}^q K^q(t_i, t_1) & \dots b_{dd}^q K^q(t_i, t_i) + \sigma_{s_i}^2 \end{pmatrix}^{-1} \begin{pmatrix} s_1(t_1) \\ \vdots \\ s_i(t_i) \end{pmatrix} \tag{4.6}$$

where q runs over the corresponding similarity classes and $d = i + j$.

For the base kernel $K^q(t, t')$ we choose (4.1) for each similarity class. That is, for example, $q = 1$ for balanced three-phase current magnitudes of a single transformer. For p transformers, we have $p \geq q$, depending on the balance between the transformers. For larger models, i.e., those with $q > 2$ similarity classes, an additional kernel (4.2) should be included to account for an underlying mutual trend.

4.4 Case Study - Living Lab Energy Campus

For the test cases, two different buildings located at FZJ as part of the LLEC have been considered, as shown in Fig. 4.2. The corresponding different internal configurations can be described as follows:

- *Building 1* is characterized by two MV/LV transformers (in the following labeled as B1-TR1 and B1-TR2) in parallel;
- *Building 2*, directly connected to Building 1, being in the same ring, is characterized by 3 MV/LV transformers (in the following labeled as B2-TR1, B2-TR2 and B2-TR3), two of which are connected in parallel.

For these two buildings, a 10-hour interval of voltages and currents without missing data has been identified in the historical database. Then, in order to be able to evaluate and discuss key characteristics of the proposed Corr-GP, different missing intervals have been simulated during the tests. The considered voltage and current signals are reported in Fig. 4.3 and in Fig. 4.4. Black curves denote training data, while green sections underline the maximum interval of the considered missing data, i.e., the sets under test. In particular, during the tests, the total duration of these intervals will vary from 5 minutes to a total period of 4 hours. Different patterns are used to represent the different phases. In order to properly evaluate the performance of the proposed method and to underline its associated coupling features, three different test cases have been considered. In Case 1, the recovery of the voltage magnitude on phase B of the first three-phase MV/LV transformer in Building 2, i.e., B2-Tr1-B, will be investigated by taking into account the magnitude of the voltages on the two other phases of the same transformer, i.e., B2-TR1-A and B2-TR2-C. First, case 1-a analyses the performance with respect to a missing 4-hour interval compared to the state-of-the-art methods Single-output Gaussian process (S-GP), Compressive Sensing (CS) [84], Linear Interpolation (Lin-Int) and the naive approach of substitution by data collected from phase A and C, denoted Rep-A and Rep-C, respectively. Second, case 1-b analyses the effect of the missing interval length, and a comparison is made with the stated alternatives.

In Case 2, in addition to Case 1, the voltage magnitudes collected from all transformers present in Building 1 and 2 are provided as input to the proposed method. More specifically, all of the three phases from the first transformer in Building 1, B1-Tr1, and the last transformer in

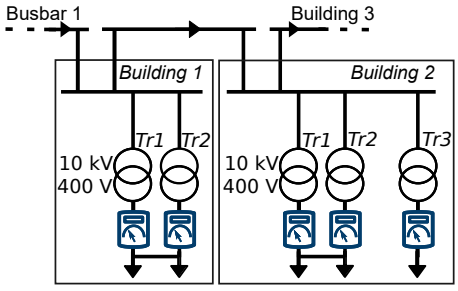

Figure 4.2: Overview of the considered buildings.

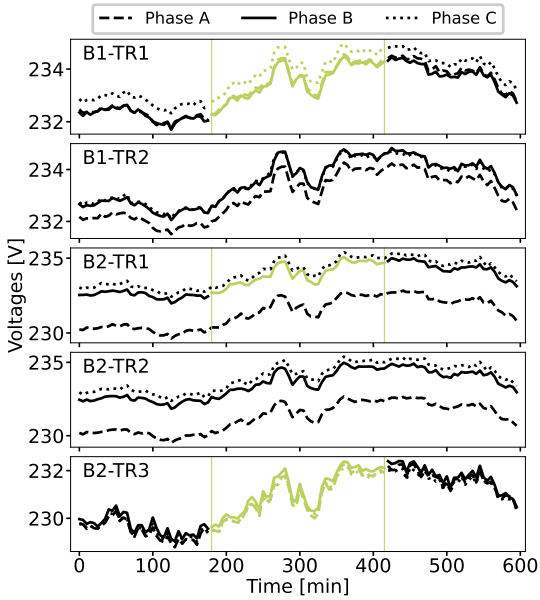

Figure 4.3: Reference signals - Voltages.

Building 2, B2-Tr3, are considered missing for 4 hours. In case 2-a, scalability and the ability to identify relevant cross-correlations of Corr-GPs are discussed. In case 2-b, a self-failing indication is presented.

Finally, to show the applicability of Corr-GPs to signals with different characteristics, the recovery of the current magnitudes of Building 2 is considered in Case 3. In particular, the current profile of B2-Tr2 will be recovered on the basis of the current magnitudes collected from B2-Tr1 and B2-Tr3.

In Table 4.1, a summary of missing and available data for each considered test case is reported,

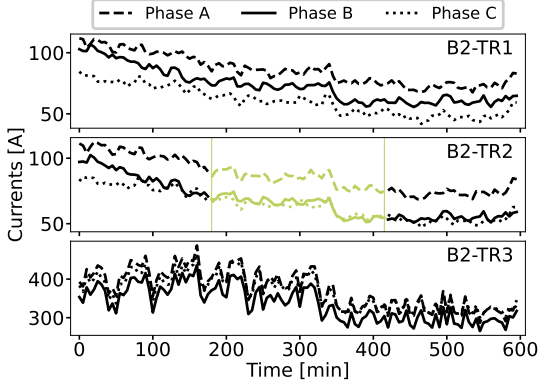

Figure 4.4: Reference signals - Currents.

in terms of RMS samples. In the table, also the number of channels (i.e., signals) considered in the test cases is indicated. Thus, with reference to Case 1-A, a total of 360 samples are considered from 3 channels (i.e., 120 samples per signal) 48 of which are considered missing. It is worth noting that, although the proposed approach is rooted in the realm of probabilistic machine learning, the amount of available data (training data) to recover missing data (test data), is comparable in size.

4.5 Results of Correlated Gaussian Process as Digital Twins for Signal Recovery

In this section, the validity of the proposed approach will be investigated by discussing its performance with reference to the test cases described in the previous section. All of the results are evaluated with respect to the normalized root mean squared error (NRMSE), stated as

$$\mathrm{NRMSE}(y_{\mathrm{true}}, y_{\mathrm{rec}}) = \frac{1}{\bar{y}_{\mathrm{true}}} \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{\mathrm{true},i} - y_{\mathrm{rec},i})^2} \tag{4.7}$$

where y_{true} and y_{rec} denote the true and the recovered values, respectively, and $\bar{y}_{\mathrm{true}}$ denotes the arithmetic mean of $y_{\mathrm{true},i}$ over the N samples.

4.5.1 Results Case 1-a: Accuracy, Precision and Basic Characteristics

The basic characteristics and the performance of Corr-GPs as a recovery tool will be presented. We consider signals derived from the collection of voltage magnitudes of the first three-phase transformer in Building 2, B2-Tr1. The explicit model reads

$$K_{\mathrm{Case1}}(t, t') = B^{[3]} \otimes K_{M^{3/2}}(t, t') \tag{4.8}$$

In Fig. 4.5, the recovery results for a missing 4-hour interval are presented. The color-contrasting upper subplot recaps the available data of 10 hours taken into account. Focusing on the missing interval, the green crosses denote the signals considered missing while blue crosses describe the recovered values by the Corr-GP. It has to be noticed that in the figure it is not possible to see the blue error bars representing the posterior variances $2\sigma_{post}$, since their value is in the order of magnitude of 0.08 V. This is in line with the interpretation of posterior variances of Corr-GPs provided in Sec. 2.3.2.1.

The comparison to alternative methods is reported in Table 4.2, in terms of the NRMSE, and

Table 4.1: Summary of missing and available data or each test case.

Case	# Missing-Channels	# Available-Channels	# Missing Data	# Available Data
1-A,B	1	2	48, 1:48	312, 359:312
2	3	12	144	1656
3	3	6	144	936

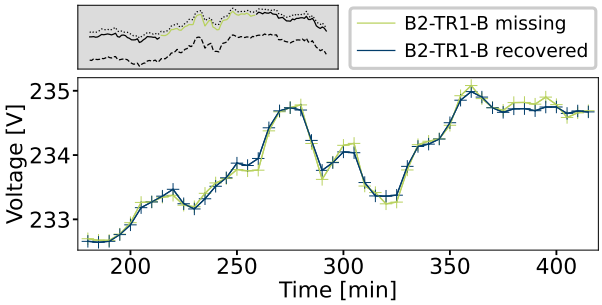

Figure 4.5: Case 1-a - True (green) and recovered voltage profiles (blue) of the three-phase transformer B2-Tr1, with Corr-GP.

displayed in Fig. 4.6. Contextualizing recap and color coding of Fig. 4.6 is used the same way as before. From Fig. 4.5, Fig. 4.6 and Table 4.2, the first two main results can be deduced. In particular, the overlapping green and blue curves in Fig. 4.5, with an NRMSE of 0.031%, underline the accuracy of the Corr-GP recovery. Compared to the alternative methods, an improvement by a factor between 5.3 and 32.0 can be archived by using the Corr-GP. With the naive approach Rep-A and Rep-C, which can also be seen as a model correlating different phases, the accuracy of this rudimentary method is unpredictable in practice as it fully relies on the network balance. While Rep-C is less accurate than the Corr-GP by a factor of 5.3, Rep-A deviates by a factor of 32.0. Linear interpolation, compressive sensing, and single-output Gaussian Processes with kernel (3.28) are less accurate by factors of 7.4, 10.7, and 10.4, respectively.

Furthermore, being in a range too small to be visualized in Fig. 4.5, the posterior covariance indicates a high precision of the given Corr-GP. This is in particular evident when compared to the S-GP, shown in Fig. 4.6-d. In both cases, the same (base) kernel (4.1) is used. This comparison clearly indicates the value of taking cross-correlations between different phases into account. While the single-phase GP is not able to accurately recover the missing signal due to missing information during the recovery period, the use of a separable kernel, given by (4.8), enables the inclusion of simultaneously occurring information. Contrarily, if no cross-correlation information is available, the proposed Corr-GP approach reduces to an uncoupled multi-output model, i.e., the correlation matrices have zero off-diagonal terms. Note that in this case, the tensor product (3.33) effectively reduces to the direct sum (3.32). As this is equivalent to multiple S-GP, in the absence of sufficient correlations, the accuracy of Corr-GP reduces to the given single-output GP alternatives.

Table 4.2: Comparison of different methods.

	Corr-GP	**Rep-A**	**Rep-C**	**Lin-Int**	**CS**	**S-GP**
NRMSE [%]	0.031	0.995	0.1651	0.230	0.333	0.322

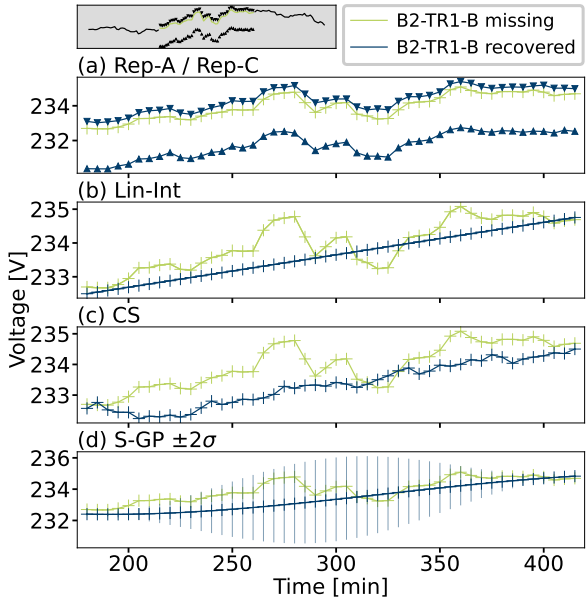

Figure 4.6: Case 1-a - Comparison between recovered voltage profiles of phase B of transformer B2-Tr1.

4.5.2 Results Case 1-b: Missing Interval Independent Recovery Accuracy

Besides the high accuracy and precision of Corr-GPs, a main characteristic is the independence of the recovery performances with respect to the missing interval length as long as a good level of correlation exists between the signal to be reconstructed and other signals in the systems. In practical applications, this property is met for sufficient large power systems by the local topological system characteristics. Using the same data set as above, this property is shown by varying the missing interval length from 5 minutes (1 sample) to 4 hours (48 samples), with an increasing step of 5 minutes. Individual NRMSEs of the methods discussed above with respect to the percentage of missing values in reference to the total interval length are reported in Fig. 4.7. The blue line is used for the results of the Corr-GP while the remaining methods are reported in black. From the figure, it is possible to deduce that the performance of Corr-GP is almost independent of the percentage amount of missing data. Due to the lack of further network unbalances in the observed period, such stability also holds for the replacement by the other phases Rep-A and Rep-C. As for the other methods considered in the comparison, it is possible to observe that the corresponding NRMSE is sensitive to the increase in the portion of missing data. In other words, these results provide an experimental validation, within the given case study, of the independence of the proposed algorithm's accuracy with respect to the missing period size, theoretically discussed in Sec. 2.3.3.1.

Furthermore, the presence of peaks in the curves of these methods underlines how their performances are also affected by the different characteristics of the available and the missing

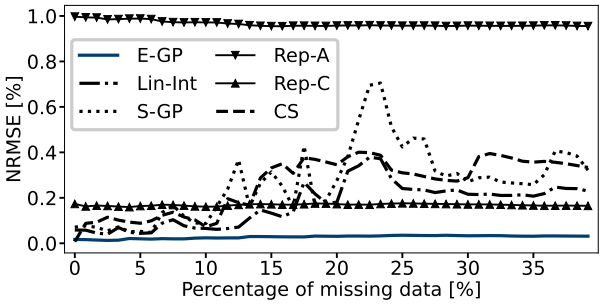

Figure 4.7: Case 1-b - Comparison of NRMSE of different methods with respect to the length of the missing signal.

signals. These considerations are not only shown in the given test case but are also clear from a theoretic perspective. As these methods do not take into account any information within the missing time frame, they cannot be expected to provide precise results for larger intervals.

4.5.3 Results Case 2-a: Cross-Correlation Identification

In the second case study, the amount of information provided as input to the Corr-GP method is increased, by considering the voltage magnitudes of all five transformers present in both Building 1 and Building 2. Instead of recovering the missing window only from one phase of a single transformer, here, three phases from two different transformers located in different buildings, B1-Tr1 and B2-Tr3, will be considered as missing. The comparison of the results achieved with the previous ICM model for 5 three-phase transformers, formulated as

$$K^{\text{ICM}}_{\text{Case2}}(t,t') = B^{[15]} \otimes K_{M^{3/2}}(t,t') \tag{4.9}$$

and the generalized LCM model, formulated as

$$K^{\text{LCM}}_{\text{Case2}}(t,t') = B^{[15]}_1 \otimes K_{M_1^{3/2}}(t,t') + B^{[15]}_2 \otimes K_{M_2^{3/2}}(t,t') \tag{4.10}$$

are reported in terms of the NRMSE, in Table 4.3.

While the results for transformer B1-Tr1 are comparable for both considered Corr-GP's and similar to those discussed in Case 1, different considerations should be made for the accuracy of B2-Tr3. We elaborate on this and discuss a key feature of Corr-GPs in the remaining part of this section.

In general, signals collected across different buildings are expected to have fewer similarities than signals collected within a single transformer. Exemplarily shown in Fig. 4.8, we report the current magnitudes of phase A for the first three hours of the considered interval for all five transformers. From Fig. 4.8, it is possible to appreciate that transformers B1-Tr1, B1-Tr2, B2-Tr1 and B2-Tr2, denoted in black, are quite balanced while transformer B2-Tr3, denoted in blue, slightly deviates. Thus, taking two classes of signal types into account by using an LCM instead of an ICM model, leads to the better performance of the Corr-GP for transformer B2-

Table 4.3: Case 2 - NRMSE of Corr-GP for B1-Tr1 and B2-Tr3.

	NRMSE [%]					
	B1-Tr1			**B2-Tr3**		
	Ph. A	*Ph. B*	*Ph. C*	*Ph. A*	*Ph. B*	*Ph. C*
$K^{\text{ICM}}_{\text{Case2}}(\mathcal{T},\mathcal{T}')$	0.023	0.022	0.024	0.685	0.350	0.186
$K^{\text{LCM}}_{\text{Case2}}(\mathcal{T},\mathcal{T}')$	0.016	0.012	0.023	0.153	0.153	0.143

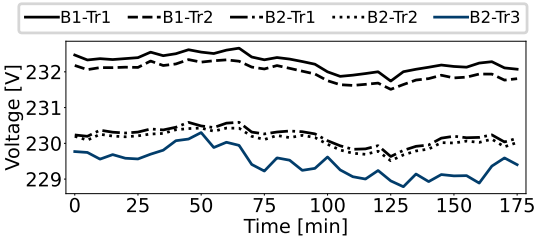

Figure 4.8: Phase A for three hours of available signals of all five transformers.

Tr3, reported in Table 4.3. The overall better performance of the LCM Corr-GP model (4.10) for the recovery of B1-Tr1 compared to those of B2-Tr3 is also evident, as B2-Tr3 is the only transformer displaying this isolated behavior. In fact, there are fewer cross-correlations available to recover voltage magnitudes of B2-Tr3. In summary, the Corr-GP learns during training which cross-correlation to take into account. If less cross-information is available in this regard, the corresponding signals are treated separately, giving results that are similarly accurate to the single-phase methods provided in Case 1.

4.5.4 Results Case 2-b: Self-Aware Failing Indication

So far, the discussion mainly focused on the Corr-GP's performance. In this subsection, an additional feature of Corr-GP for practical, and possibly automated, usage by system operators is underlined. In Fig. 4.9, posterior variances are shown for both models under test with respect to the overall 4-hour missing interval as well as the corresponding accuracy results.

As shown in Fig. 4.9-a, the ICM approach is characterized by a significant increase of σ_{post}-values when compared to the LCM model depicted in Fig. 4.9-b. Moreover, a dependence between σ_{post} (l.h.s. of Fig. 4.9) and the NRMSE (r.h.s of Fig. 4.9) is evident. Thus, not only does the proper Corr-GP model, in this case, the LCM, come with high accuracy, but the less fitting Corr-GP approach can be detected by a high posterior variance. Contrarily to the common usage as an error estimate of the resulting values, the posterior variance of GPs displays an error estimate of the selected model. The posterior variance describes the uncertainty of the Corr-GP associated with each recovered sample, where the latter is the corresponding Corr-GP mean value. This means that the higher the posterior variance, the less the selected Corr-GP model fits the recovery task of a desired accuracy. Thus, the notion of posterior variances yields a self-failing indication allowing a direct evaluation of the performance of the given model. Consequently, an automated iterative procedure could be performed by system operators, by gradually increasing model complexity. In cases without detailed prior knowledge of the similarity classes

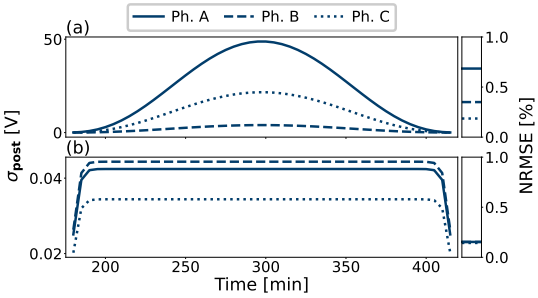

Figure 4.9: Posterior variances (left) for transformer B2-Tr3, by a) ICM and b) LCM, with corresponding NRMSE (right).

provided by the topology and load conditions of the power system, or within an automated operation, a trial-and-error approach could be performed. First, a basic ICM model of the proposed approach corresponding to a single similarity class including posterior variances is computed. If, based on the posterior variance, the desired model accuracy is not sufficient, an additional ICM term is appended and the extended Corr-GP is retrained. This procedure might be iterated until an acceptable value for the posterior variance is reached. In other words, the iterations terminate when the number of added ICM terms corresponds to the number of similarity classes. However, in practical applications, the number of possible iterations is bounded by computational resources, thus, limiting a fully automated operation in cases where no cross-channel information is available. In cases where detailed information can be provided by system operators or a dedicated pre-analysis of the given signals is performed, the trial and error procedure might be neglected. In such a case, the posterior variance serves as a consistency check.

4.5.5 Results Case 3: Modeling of Multi Cross-Correlated Signals

Due to their different characteristics, current signals cannot be modeled as the voltage ones. Evident from Fig. 4.10 and Fig. 4.4, in Case 3 the current magnitudes are considered which are characterized by less homogeneity and cross-correlation compared to the voltages. Taking into account the 10-hour current profiles of all phases of B2-Tr1 and B2-Tr-3, as well as the preceding and following 3-hour intervals of B2-Tr2, the recovered 4-hour interval profiles are displayed in blue in Fig. 4.10. In particular, missing current magnitudes to be recovered are denoted in green.

By a closer inspection of Fig. 4.4, it is possible to observe that there exist three similarity classes in the given data set, yielding (4.11b). Moreover, to also have a correlation term across distinct classes, we take long-term correlations into account, established by (4.11a). Thus, the proposed Corr-GP model reads

$$K_{\text{Case3}}(t,t') = B_0^{[9]} \otimes K_{\text{RBF}}(t,t') \tag{4.11a}$$

$$+ \sum_{q=1}^{3} B_q^{[9]} \otimes K_{M_q^{3/2}}(t,t') \tag{4.11b}$$

and gives accuracies with respect to the NRMSE of 1.2%, 1.0% and 1.3% for phase A, B and C,

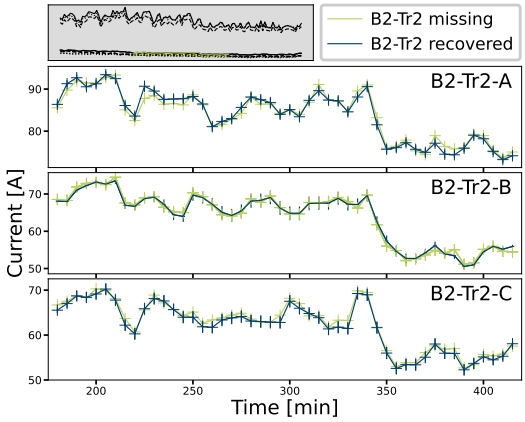

Figure 4.10: Case 3 - True (green) and recovered current profiles (blue) for the three phases of transformer B2-Tr2.

respectively. Contrarily, in this scenario linear interpolation results in 4.6%, 7.5% and 5.5%.

From Fig. 4.4, it is possible to observe that the current profiles of B2-Tr1 and B2-Tr2 are very similar, while the currents flowing in the third transformer are almost 4 times higher, and also characterized by a more volatile profile. Considering the results shown in Fig. 4.10, we can thus observe that the Corr-GP recovery method is able to distinguish between the characteristics of the different coupled signals and to give more weight to the information coming from the more similar inputs.

4.6 Online Application of a Digital Twin for Data Quality in Power Systems ICT Platforms

The main area of application of the presented Corr-GP signal recovery algorithm is the data recovery of large missing periods. Although nothing prevents the Corr-GP model to be applied to single missing data points, the computational overhead compared to comparably accurate but computationally cheaper algorithms such as, for example, linear interpolation has to be taken into account. This is in particular relevant for large-scale databases like the presented use cases. To take this into account, in the given ICT platform we provide the following modular framework. In its core functionality, the framework consists of two modules, one for detection and one for recovery of bad data. Both modules consist of submodules themselves, which allows an easy and modular application of different detection and recovery algorithms. On top, to provide a useful framework for system operators, additional modules are included. We describe the framework in Sec. 4.6.1 and each module in Sec. 4.6.2.

4.6.1 Digital Twin Framework in ICT Platforms

We propose a modular tool for possibly automated anomaly detection and signal recovery. In Fig. 4.11, an overview of the components of the approach is shown. To provide a clean solution from a data processing point of view, we do not alter the data stored in the original database, in the following referred to as *Measurements DB*. The proposed tool includes a dedicated database, namely *Applications DB*, for internal data handling. Then, the tool itself is composed of six modules which can be grouped into four levels:

- control level, where the modules aimed at ensuring the reliable operation of the tool are defined;
- data processing, where the scan and recovery modules are allocated;
- data storage, where the databases are allocated;
- user interaction, where the dashboard module allows the users to interact with the databases and the execution module.

4.6.2 Modules in ICT Framework

4.6.2.1 Scan Module

The scan module constitutes the first part of the data processing level in the proposed tool, the one focusing on the identification of anomalies in the signals, thus on the bad data detection. Its general scheme is shown in Fig. 4.12, and it is composed of three main sections.

The first part of the module consists of reading the most updated data from the *Measurements DB*. This action could be performed in real-time, as soon as a new value, or set of values, is available in the database. Nevertheless, there are cases in which data cannot be scanned in real-time, because of computational and/or communication network limitations, thus a buffer to deal with data characterized by too high reporting rates and/or queues would be required.

The data are then provided as input to the data-processing sub-modules, each one aimed at detecting a specific anomaly. For example, anomalies of interest could be zeros, not-a-numbers (NaNs), unexpected sudden changes or large missing periods of data. It is worth mentioning that missing values are typically replaced by zeros or NaNs, thus included within the above-mentioned categories. Nevertheless, there are also cases, such as in the *Measurements DB* at FZJ, where

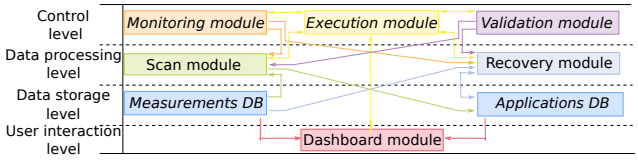

Figure 4.11: Overview of the proposed approach.

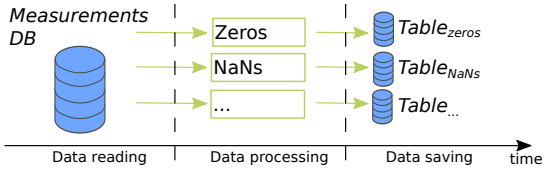

Figure 4.12: Scan module scheme

missing values are not replaced and thus they can be identified by analyzing the distance between two following time tags.

Having dedicated sub-modules aimed at discovering specific anomalies allows to easily implement ad-hoc developed algorithms depending on the needs. The sub-modules could be implemented by APIs running on dedicated hardware, which empower their computation on the most suitable platform. For example, scan sub-modules characterized by a low computational burden (e.g., detecting zeros and NaNs) could be executed directly on the server where the ICT platform interface runs. More computationally intensive sub-modules, such as those aimed at recognizing values that seem reasonable but are affected by high errors, should run on more powerful platforms, such as dedicated servers. Nevertheless, it is worth mentioning that in certain applications having a scan module that captures slightly fewer anomalies with respect to another one, while being characterized by a lower computational burden could be a trade-off one would be willing to accept.

Once the anomaly is detected, each sub-module records key information associated with it, such as original value, time tag, and sensor ID. These information are stored in dedicated tables of the *Applications DB*, in order to avoid modifications on the values stored in the *Measurements DB*.

4.6.2.2 Recovery Module

The recovery module constitutes the second unit of the data processing level. Also in this case, the main structure of the module is organized into three parts: data reading, data processing, and data saving, as shown in Fig. 4.13.

Individual sub-modules could be implemented for specific recovery methods. In particular, each recovery sub-module is enabled to read the data from the tables previously populated by the scan module. According to the type of anomaly, and the specific requirements of the implemented recovery algorithm, the recovery sub-module queries the *Measurements DB* to get the additional information needed for the recovery, e.g., a specific set of "correct" samples. With all of the necessary information available, the recovery algorithm implemented in the sub-module provides a new estimation for the anomaly under analysis and stores it in a dedicated column of the corresponding table.

Another interesting feature of the recovery module is the *statistics* sub-module. Similarly to the other sub-modules, this component can read the data from the various tables in the *Applications DB*, but it is aimed at collecting information on the number of failures of a specific

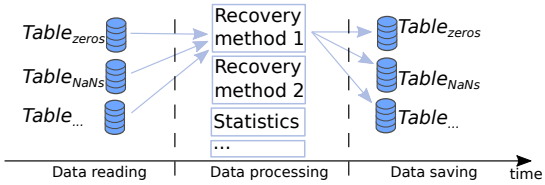

Figure 4.13: Recovery module scheme.

device, the number of occurrences of a specific anomaly, and others. Its role is of particular importance since this information can be used by the system operator to conduct dedicated investigations on specific devices and/or plan maintenance.

4.6.2.3 Execution Module

A decentralized approach like the one presented, with many instances and possible resource-binding algorithms requires dedicated control. The execution module is the key of the control level and coordinates the self and mutual activities of the other modules of the tool. For example, it could avoid query delays in the *Measurements DB*, preventing reading operations from running in parallel. If large communication traffic is expected, the execution module could impose the scan and recovery modules described above to be executed during reduced communication phases. With the same principle, the modules of the data processing level can be also used to perform analysis on historical data, or other databases.

Both versions, i.e. the automated as well as the on-demand executions should also have a stop, break and rollback functionality to avoid blocking the databases for other processes, like updates or cross-system applications. In general, the operations of the execution modules could be automatized, thanks to the coordination with the information gathered from the monitoring module, described in the following subsection. Nevertheless, it is worth mentioning that, for safety reasons, the system operator must always be capable of directly manage the execution module, and thus every other module of the tool.

4.6.2.4 Monitoring Module

The monitoring module is a self-monitoring component that keeps track of all metadata. This includes but is not limited to start and end time of active scripts, devices and servers of execution, as well as every access to databases, tables and specific entries. The module guarantees a full reconstructability of every process. This is not only a benefit for the operator, but constitutes a key aspect of scalability and decentralized execution. For example, the more instances mutually query the same entry of a specific table, the more caution with respect to read/write errors is needed. Moreover, the gained knowledge can be used for a safe and automated operation of the execution module.

4.6.2.5 Validation Module

The validation module gives the possibility to test different applications with respect to each other. In particular, it defines an interface to artificially include typical anomalies described above. This could be used to validate the activities of the modules described above. By partly mirroring a chosen table, an automatic insertion of predefined artificial anomalies is performed. The automated aspect is a key feature as it supports the large scale analysis instead of single values or events.

4.6.2.6 Dashboard Module

Finally, the dashboard module constitutes the interface between the tool and the users. As shown in Fig. 4.14, the dashboard module should be able to read data from both the *Measurements* and the *Applications DB*. From the former, the original data collected from the network are read with reference to both real-time and historical data, while from the latter only the recovered values are considered. Depending on the specific needs, the GUI of the dashboard can be customized with various functionalities. Nevertheless, for the generic signal of interest, it would be sufficient to give an overview of the real data with the recovered values marked accordingly.

4.7 Conclusion and Characterization of Power System Digital Twins

4.7.1 Discussion

ICT Framework for Bad Data Detection & Handling The structure of a modular tool for bad data detection and handling in ICT platforms for modern energy systems has been proposed. The functionalities of the six modules composing the tool have been discussed based on the ICT platform of FZJ. While the concept is theoretically proposed, actual implementations on large-scale platforms come along with additional obstacles. For example, bottlenecks in interfaces might affect the communication between individual substations. On top of this, writing/reading processes of the bad data detection and handling routines might cause additional bad data. Moreover, needed computational capacities must be determined and monitored regularly as they might change during operation.

Correlated Gaussian Processes for Signal Recovery Power systems data recovery by Corr-GP has been proposed, discussed, and evaluated. Tested with voltage and current signals collected from the LLEC, the proposed Corr-GP is capable of outperforming the alternative state-of-the-art methods. In contrast to the considered alternatives, the accurate performance is mainly independent of the missing interval length. Besides the approximately one order of magnitude higher accuracy when cross-correlated information is available, Corr-GPs perform comparable accurately to the stated alternatives if no such correlated information is available. The proposed method yields a self-failing indication in terms of posterior variances, thus recovery

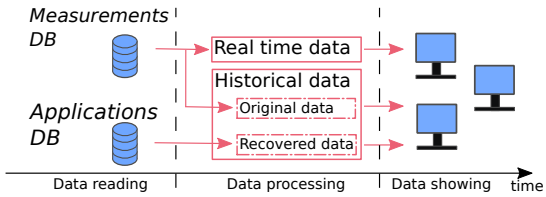

Figure 4.14: Dashboard scheme

precision, allowing system operators a direct evaluation and further improvement if needed.

The proposed Corr-GP approach is neither limited by the missing window length nor the type of correlation under consideration. However, it is worth emphasizing that the absence of any correlated signal makes the given approach not applicable. Thus, regarding an automated application, the relevant domain is within large-scale monitoring systems, in which the existence of correlated signals can be guaranteed in practical scenarios.

One aspect to consider is the computational complexity associated with the matrix inversion by Cholesky decomposition. In [124], an updated Cholesky decomposition is proposed to accelerate the calculation of the matrix inverse. However, this means a continuous operation, i.e., the inclusion of every data point into the Corr-GP. In the recovery of large missing windows, this is not beneficial since the outage of sensors, communication problems within the database, or maintenance work do not frequently occur but are limited to individual occasions.

4.7.2 Characterization of Power Systems Data Quality Digital Twins

ICT Framework for Bad Data Detection & Handling The characterization of the presented approach for the ICT framework for bad data detection is vizualized in Fig. 4.15. The given framework can cover an arbitrary level of detail as the monitoring and recovery abilities are only limited by physical resources. The data transfer can be classified as online and explicit. The classification according to real-time aspects only applies to the modules accessing the MongoDB, which stores the most recent data points. As the InfluxDB only stores historic data, no notion of (power) system dynamics exists, and thus, no real-/not-in-real-time classification applies. Limitations thereof are only due to computational resources as the given structure displays a high level of parallelization. The given framework displays no aggregation as each signal is represented in a one-to-one deterministic correspondence. Due to the modular approach, the framework is updatable with self-validation due to the dedicated validation module.

Correlated Gaussian Processes for Signal Recovery The Corr-GP approach can be characterized as shown in Fig. 4.16. In its basic formulation, the Corr-GP model maps signals displaying a missing period to the recovered signals. Thus, these models can be characterized as DSs. As shown in Fig. 4.16, with reference to the entire power system, the level of detail can be characterized as medium due to the local power system consideration. The data transfer can be classified as explicit, while the type depends on the given implementation. In the given case study, the missing signals were recovered offline. An online application is possible as described

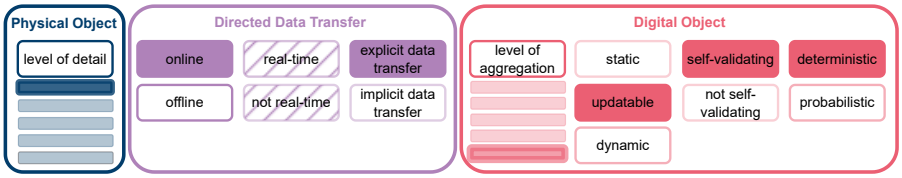

Figure 4.15: DT characterization of the provided ICT framework.

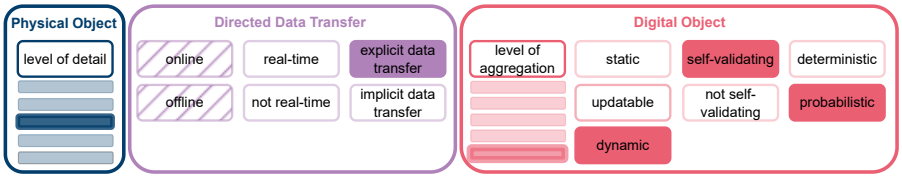

Figure 4.16: DS characterization of the provided Corr-GP signal recovery algorithm.

in the ICT framework given above. As the recovery algorithm is applied subsequentially in the InfluxDB, a classification in terms of real-/not-in-real-time does not apply. The Corr-GP represents each signal one-to-one, thus, no aggregation is performed. Due to the interpretation of the posterior covariance, the proposed DS can be characterized as self-validating, and consequently dynamic. The DS is probabilistic.

4.7.3 Outlook

The next step for the ICT framework of bad data detection and handling is the implementation within the LLEC. To keep the Corr-GP computationally tractable in large-scale platforms, the computational burden has to be analyzed and possibly reduced. A possible reduction can be obtained with the use of a minimal data set in each recovery task, i.e., the ratio between known and missing data has to be minimized. Within a large-scale application, this requires a statistical analysis to ensure reliable applicability. Another possibility is the limitation of hyper-parameters to a relevant domain. Although re-training can not be avoided completely as timescales and magnitudes might change during online operation, the number of optimizer restarts might be decreased. Another aspect to analyse within the application at large-scale platforms is an analytical pre-analysis of the given signals to pre-determine the number of needed similarity classes. This might reduce the number of possible model iterations, and could thus, reduce computational costs. Despite reducing data and algorithmic solutions, a tailored GPU-based computational environment including a specialized Corr-GP implementation might be worth investigating.

5

A Digital Shadow for Power Electronics Parameter Estimation

In this chapter, we provide a DS for power electronics parameter estimation. By encoding the power electronics state-space equations into the machine learning algorithm, we construct a backward solution for the power electronics component. Purely based on input/output measurements, this provides a non-invasive characterization of the specific power electronics device. Although the presented DSs are based on supervised machine learning algorithms, the application of estimating the device parameters is unsupervised. More precisely, there is no need to measure the device parameters in a dedicated experiment to provide training data. In Sec. 5.1 we motivate and contextualize the provided DS. In Sec. 5.2, within a simulation environment, the parameters of a DC-DC buck converter described by an LC model are estimated. We test, analyze, and compare a PIGP based on linear operators, a forward solver-based approach, called ADShooting, and provide an application-tailored PINN. In particular, we discuss the limitations of these algorithms when applied to real power electronics devices. In Sec. 5.3, we propose affine operator kernels that apply to devices operating in the field. Tested with measurements conducted in the field, parameters of half-bridge converters operated in buck and boost topologies are provided. To conclude this chapter, we give a discussion and characterization in Sec. 5.4. This chapter is based on [1], [5], [6].

5.1 Introduction

5.1.1 Motivation

Parameter estimation of power electronic converters leads to useful insights into the converter characteristics, which can be utilized to improve various applications such as control [35], fault diagnosis [32], as well as condition and health monitoring [33]. In control applications, adaptive control depending on the device characteristics and the operational regime can lead to enhanced control schemes, such as model predictive control (MPC) [35], finite-time control [127], and system identification control schemes [34]. Within the field of fault diagnosis and condition monitoring, parameter estimation leads to safe and reliable operation [45], [128]. In particular, parameter degradation has to be taken into account [129]. Although nominal values of parameters are given in data sheets, the stated values are often given with relatively large uncertainty margins, which might not be sufficient for the desired application [130], [131]. Besides accuracy and robustness, additional properties, such as adaptability and sufficiently fast online execution, are needed [34], [127], [130]. Consequently, there is a need for flexible parameter estimation techniques that are capable of extracting non-invasively converter parameters from power electronics devices.

5.1.2 Related Work

For power electronics parameter estimation, a broad variation of approaches ranging from parametric and non-parametric to dual hybrid modeling exists, yielding various system identification algorithms [132]. In [42], initially fixed parameter values turn into probabilistic functions via polynomial chaos expansion, yielding a real-time probabilistic digital twin with stochastic parameter characteristics. In [133], the effects on power electronics parameters are considered via thermal modeling of the physical devices based on cross-coupling effects using an artificial intelligence-aided thermal model. In [134], the equivalent series resistance for the output capacitor of a boost converter is estimated based on a model derived by the output ripple. In [135], an artificial neural network is trained to obtain the DC-link capacitance in a diode-bridge front-end converter system. With an explicit focus on measurement noise, in [136] a state-space model is used to identify parameters via full-state observation. In forward solver-based approaches, many possibilities for the backward update have been tested in different domains [137]. For the backward update with respect to parameters of interest, among others, automatic differentiation has been used [138]. Regarding solver-based approaches, the main drawback is the profound sensitivity to noise, which complicates a practical application.

Artificial intelligence has found many applications, including power electronics [139]. With a focus on condition monitoring, machine learning-based parameter estimation techniques are reviewed in [33]. More specificall,y concerning applications to DC-DC buck converter parameter estimation, the authors of [46], [140] utilize distinct architectures of neural networks to extract power electronics parameters from state and input measurements. In [46], a fully connected PINN with five layers, 50 neurons each, is used. With a sampling rate of 20 kHz, the given approach is tested in simulation and experimentally verified. Although the results of [46] appear

quite promising when tested in simulation scenarios with clean data sets, the same method, when tested using real measurement results, in estimation errors above 10%. In [140], within a simulation scenario with a sampling frequency of 200 kHz, a physics-informed LSTM network is constructed. As both approaches [46], [140] consist of rather large neural network structures, scaling to large systems might be challenging. Moreover, [46], [140] formulate the PINN for a discrete solver-step-like change of states instead of the states themselves. As the capability of a PINNs to estimate parameters is independent of the particular forward solver, the given constructions seem not to be the natural formulation for the given application of power electronics parameter estimation [86]. Despite the neural network architecture itself, the training procedure plays a significant role in the given application [86]. This has not been explicitly explored in literature. Similar to pure solver-based approaches, solver-based neural network approaches like in [46], [140] are sensitive to the presence of noise. In [141], linear operators PIGPs are used to extract line parameters of small power systems. As the methods used in [141] only utilize linear differential operators as developed in [99], the application is limited to very special circumstances and thus, not applicable to power electronics in general. Moreover, the result presented in [141] indicates that the given algorithmic solution is not applicable in real-world applications.

To summarize, the existing power electronics parameter estimation techniques discussed above are limited in their ability to reliably extract parameters from measurement signals of power electronic converters. Thus, there is a need for a DS with application to power electronics parameter estimation capable of dealing explicitly with noisy measurements to overcome these limitations.

5.1.3 Preliminary Remarks on Technical Realizations

The construction of DSs with the purpose of power electronics parameter estimation can be given by the inclusion of the device state-space equations into the machine learning algorithm. However, a priori, it is not evident which specific system description should be used. More precisely, in practice, it is not obvious how to choose the model description matching a given data set obtained by measurements of devices operated in the field. Thus, immediate questions arise such as which set of equations displays the underlying physics, which level of detail is obtainable subjected to a significant signal-to-noise ratio, and how this depends on internal and external operational conditions, possibly corrupted by the measurement process. Therefore, educated guesses by dedicated experts are needed to provide an analytically precise system description. However, the hybrid approach of combining mechanistic and machine learning modeling opens new possibilities. First, note that a key assumption of any algorithm providing a backward solution is the validity of the given equation. Wrongly chosen equations lead to physically non-interpretable parameters. Restated positively, as the underlying equation is taken for granted, the backward solution provides device parameters of effective/equivalent models. Consequently, not every single effect within the data set has to be captured by the set of equations but only at an effective/equivalent level.

Apart from expert knowledge, some conclusions for the selection of a sufficient description in terms of state-space equations can be drawn from a purely mathematical perspective. As

described in Sec. 3.3.1, we can write a general state-space equation as

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}(t) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}(t) + \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}(t) \tag{5.1}$$

with states x_i and inputs u_i. Typically true for power electronics applications and in a certain sense for a large part of any electrical power-related systems is its topological characteristic. That is, due to the spreading of electric and magnetic fields throughout the entire system, each individual branch affects the remaining part of the network. With regard to (5.1), this indicates to include all parameters $a_{11}, a_{12}, a_{21}, a_{22}, b_{11}, b_{12}, b_{21}, b_{22}$ in the model building process. This guarantees the (partial) presence of all states and inputs in each signal's characteristics. A counterexample in terms of a toy model is provided in Sec. 5.2, which is a toy model exactly because of the absence of coupling. On top of non-zero parameters, additional internal coupling might be present. For example, a mutual parameter R might be shared among distinct matrix entries, e.g. $a'_{ij} = R \cdot a_{ij}$ and $a'_{kl} = R \cdot a_{kl}$ for $i, j \neq k, l$. Depending on the specific application, the training process benefits from internal parameter correlations yielding a physical interpretation of the given parameter. Nevertheless, from a purely effective/equivalent modeling point of view, this is not required.

Another consequence from the effective/equivalent perspective is using average state-space equations to estimate the parameters of interest. The switching dynamics do not influence the converter parameters. Thus, average state-space equations should be sufficient. Moreover, the average level significantly simplifies the model-building process. This is due to the change in the device topology caused by the switching in on- and off-phase. Mathematically, this implies a switching between two state-space equations that have to be integrated into the corresponding machine learning method, including precise temporal monitoring of the present state. At the average level, only a single (average) state-space equation has to be taken care of.

Recall from Sec. 2.3.3, that we argued for a continuous-time model from a system's perspective, i.e., states are modeled as a function of time $t \mapsto x(t)$. Contrarily, the models developed in [46], [140] do not follow this point of view. In [46], [140], the neural networks model the state evolution $x(t_i) \mapsto x(t_{i+1})$. Although this procedure might seem natural from a simulation or control perspective, it is by no means crucial in the realm of physics-informed machine learning. A counterargument to this approach is the profound sensitivity to noise, which is a major aspect of power electronics.

5.1.4 Contribution

In Sec. 5.2, we propose, discuss, and compare three power electronics parameter estimation techniques. For all three methods, we propose application-tailored model formulation and corresponding training strategies. We illustrate the proposed approaches as parameter estimators of a DC-DC buck converter in simulation scenarios. First, we propose an application scheme for linear PIGP enabling an application in power electronics parameter estimation. We define an algorithmic scheme for PIGP based on linear operator kernels, enabling a well-defined and numerically stable application. We discuss the need and how to approach a balanced training procedure in the field of power electronics. We explicitly show how to determine noise hyper-

parameters from power electronic signals. We provide a dedicated algorithmic scheme for kernel regularization. Second, we propose a lightweight PINN architecture as well as a dedicated training strategy. The lightweight construction enables the scaling of system size. Compared to 10654 parameters present in [46], we use 754 parameters. Conceptually, the proposed PINN approach differs from [140] and [46] in the following sense. We represent the state $x(t)$ as a function of time t with temporal system dynamics given by the state-space equations. Contrarily, in [46], [140], the state evolution $x(t_i) \mapsto x(t_{i+1})$ is modeled by the neural network. In other words, the state-space equations are implicitly/explicitly forward-solved and backward-trained. However, parameter estimation is the backward solution of the state-space equations. We propose the following dedicated training strategy. We utilize relative weights in the loss function and corresponding suppression of learning rates. Converter parameter updates are limited to suppress premature learning directions. We further use neural network representations of power electronics inputs in the physics-informed part of the loss function. The latter solves the issue of computing residuals of differential equations, as reported in [140]. Third, based on a forward solver with backward training update by automatic differentiation, a method called ADShooting is presented. For training, we propose to use a cubic root for gradient updates as a smoothing operator, and for low-gradient amplification, a rolling solver starts to suppress the noise dependence of initial values and a K-wise iterative training strategy. The latter allows the application of the given approach to second-order systems, generalizing [142].

In Sec. 5.3, we propose the construction of affine operator kernels yielding PIGP applicable to power converters in the field. We show and discuss the limitations of linear operator kernels and clarify the need for as well as the value of the extension to the affine case. We show how affine operator PIGP outperforms PINN and ADShooting algorithms in its ability to deal with measurement noise. Phrased differently, we show how explicit access to the noise level in terms of hyper-parameters of the proposed approach can be utilized. We propose an application-tailored model-building process including a proper self-consistent normalization scheme. We show how to pre-determine noise hyper-parameters and discuss differences from the linear construction. We provide a dedicated algorithmic scheme solving numerical issues arising in this type of model. In particular, numerical differences and their influence on the model-building between power converters in the field and simulation scenarios are discussed. We explicitly address the proposed algorithm's ability to extract any parameter of interest from the corresponding measurement signals. As a case study, we consider the application of the proposed approach to the parameter estimation of a half-bridge converter with buck and boost topology operated in the field.

5.2 Power Electronics Parameter Estimation - Basic Modeling and Theoretical Insights

5.2.1 LC Model for Buck Converter Parameter Estimation

Consider a DC-DC buck converter with a topology depicted in Fig. 5.1. With inductor current

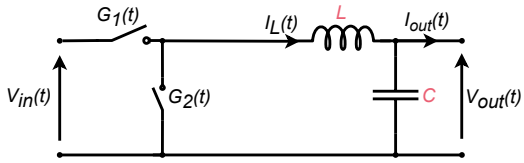

Figure 5.1: LC model of a DC-DC buck converter.

$I_L(t)$ and output voltage $V_{out}(t)$ as states, we have

$$\begin{pmatrix} \dot{I}_L \\ \dot{V}_{out} \end{pmatrix}(t) = \begin{pmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{pmatrix} \begin{pmatrix} I_L \\ V_{out} \end{pmatrix}(t) + \begin{pmatrix} \frac{\delta}{L} \\ 0 \end{pmatrix} V_{in}(t) + \begin{pmatrix} 0 \\ -\frac{1}{C} \end{pmatrix} I_{out}(t) \tag{5.2}$$

with converter parameters L, C describing inductivity and capacity, respectively, duty cycle δ and inputs $V_{in}(t)$ and $I_{out}(t)$. The gate signals measured by the voltages across the switches are denoted by G_1 and G_2 in Fig. 5.1. Note that there are two input terms present in (5.2), which can be motivated as follows. When deriving (5.2) from the topology as given in Fig. 5.1, one obtains a load resistance R_{load} decorating the output voltage $V_{out}(t)$ [143]. As this parameter describes the equivalent resistance with origin from the connection to the grid at the output side and not an inherent converter property, we use $V_{out}(t)/R_{\text{load}} = I_{out}(t)$ to replace this parameter.

Equation (5.2) describes an average model with duty cycle δ. From a theoretical point of view, this is sufficient as the converter parameters $\phi = \{C, L\}$ are fully determined at the average level as can be seen in (5.2).

5.2.1.1 The Role of Duty Cycle

From a power electronics application perspective, the system (5.2) has two device parameters L, C, and a controllable duty cycle δ. From the backward solution point of view, all three quantities L, C, δ are parameters that might be estimated. Due to its different origin as an effective, average quantity, we investigate the role of the duty cycle δ separately. By definition, the duty cycle is defined as

$$\delta_{\text{Gate}} = \frac{PW}{T_{\text{period}}} \tag{5.3}$$

with pulse width PW and total period T_{period}. This quantity can be computed from the two gate signals G_1 and G_2 via a voltage measurement across the switches. Alternatively, we can consider the first equation of (5.2), i.e.,

$$L\frac{d}{dt}I_L(t) - \underbrace{(-V_{out}(t) + \delta V_{in}(t))}_{=\text{rhs}_\delta(t)} = 0 \tag{5.4}$$

and minimize with respect to δ. That is,

$$\delta_{min} = \min_{\delta \in [0,1]} \sqrt{\sum_{i=1}^{N-1} \left(\frac{I_L(t_{i+1}) - I_L(t_{i-1})}{t_{i+1} - t_{i-1}} - \frac{\delta V_{in}(t_i) - V_{out}(t_i)}{L} \right)^2} \tag{5.5}$$

using discrete measurements and the numerical derivative. Although the duty cycle is set by the considered controller to a specific value δ_{set}, the effective duty experienced by the LC-component might differ in general. This might be due to, for example, dead times present for safety reasons during operation. Also, when computed via measurements, the sampling rate affects the actual value significantly. We provide an illustrative example in Table 5.1. For a current load step with $\delta_{\text{set}} = 0.5$, results according to (5.3) and (5.5) are provided. We use the BFGS algorithm for minimization [98] in (5.5). The evaluation of the computed results for the effective duty cycle provided in Table 5.1 can be seen in Fig. 5.2. In Fig. 5.2-(a), we display the right-hand side of the first state-space equation as defined in (5.4) for the computed duty cycles δ_{G_1} (red), δ_{G_2} (green), $\delta_{\min}$ (blue) and the left-hand side provided by $L\frac{d}{dt}I_L(t)$, where we used the numerical derivative. A zoom is given in Fig. 5.2-(b). First note, that the switching ripple is present in $L\frac{d}{dt}I_L(t)$ (light blue), which should be treated as noise in the following proposed algorithms as we give an averaged construction. In 5.2-(c), equation (5.4) is given. A zoom on states obtained in steady-state are provided in Fig. 5.2-(d). Evident from Fig. 5.2-(d), the duty cycle $\delta_{\min}$ given by the minimization scheme matches the given state-space equation best. The duty cycles determined from the gates cause a mismatch in (5.4) of 2 kV in steady-state. It should be noted that (5.5) is only applicable if all converter parameters are known. Thus, independent of the proposed parameter estimation model, we can deduce that the duty cycle can not be predetermined by the gate signals. A pre-determination by minimization is only applicable in simulation scenarios with full system knowledge. Note also that according to (5.2), the duty cycle decorates $V_{in}(t)$, i.e., the signal with the highest numerical value present. On top, the term $\delta V_{in}(t)$ is part of the data that defines the training set. Thus, an imprecise determination of the duty cycle significantly affects all follow-up processes including the parameter estimation. Consequently, in any real-world parameter estimation application, the duty cycle has to be treated as a parameter, which can be estimated. Note that this also holds if one is not interested in the precise determination of the duty cycle itself.

In order to discuss the basic properties of the DS approaches, in this section we assume that the duty cycle is known. The effective value is set to $\delta = \delta_{\min}$ for simplicity.

5.2.1.2 Expectable Error Estimates

In the presence of noise, expectable inaccuracies of any parameter estimation algorithm exist. We give an estimate thereof. Formally, we can solve (5.2) for L and C, incorporate averaged measurements and compute the uncertainty propagation [102] as

$$\begin{aligned}\varepsilon L^2_{\sigma_n} = \left(\frac{\Delta t}{N}\right)^2 \sum_{i=1}^{N} \Bigg[&\left(\frac{\Delta V_{in}\delta}{I_{L_{i+1}} - I_{L_i}}\right)^2 + \left(\frac{\Delta V_{out}}{I_{L_{i+1}} - I_{L_i}}\right)^2 \\ &+ \Delta I_L^2 \left(\frac{\delta V_{in_{i+1}} - V_{out_{i+1}}}{(I_{L_{i+2}} - I_{L_{i+1}})^2} - \frac{\delta V_{in_i} - V_{out_i}}{(I_{L_{i+1}} - I_{L_i})^2}\right)^2 \Bigg]\end{aligned} \tag{5.6}$$

Table 5.1: Nominal and calculated duty cycle values.

δ_{set}	δ_{G_1}	δ_{G_2}	$\delta_{\min}$
0.5000	0.4818	0.4817	0.4774

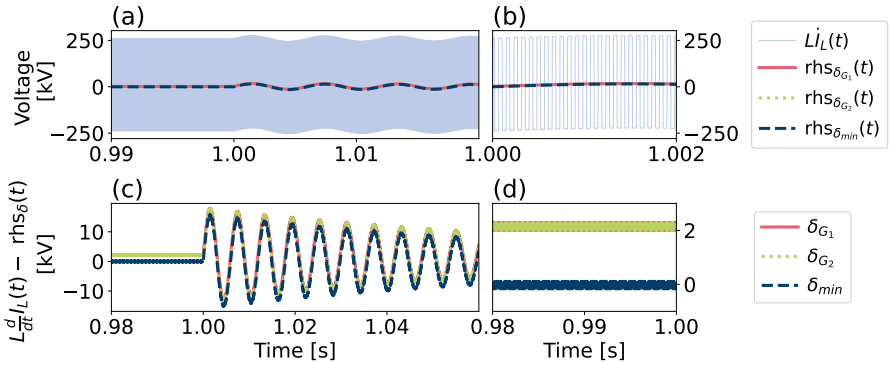

Figure 5.2: Evaluation of duty cycle determination. (a) $L\frac{d}{dt}I_L(t)$ (numerical derivative), rhs_δ according to (5.4), (b) zoom of panel (a), (c) equation (5.4), (d) zoom of panel (c)

and

$$\varepsilon C_{\sigma_n}^2 = \left(\frac{\Delta t}{N}\right)^2 \sum_{i=1}^{N} \left[\left(\frac{\Delta I_L}{V_{out_{i+1}} - V_{out_i}} \right)^2 + \left(\frac{\Delta I_{out}}{V_{out_{i+1}} - V_{out_i}} \right)^2 + \left(\frac{\Delta V_{out}^2 (I_{L_{i+1}} - I_{out_{i+1}})}{(V_{out_{i+2}} - V_{out_{i+1}})^2} - \frac{\Delta V_{out}^2 (I_{L_i} - I_{out_i})}{(V_{out_{i+1}} - V_{out_i})^2} \right)^2 \right] \tag{5.7}$$

Solely based on the state-space equations and corresponding single signal error estimates, (5.6) and (5.7) serve as a reasonable measure of the expected inaccuracies. Technically, this corresponds to the direct hands-on approach of solving state-space-equations with respect to converter parameters and averaging over all given signals per time instance.

5.2.2 Parameter Estimation by Linear Operator Gaussian Processes

We propose a model-building scheme that enables linear operator PIGP to be applied in power electronics parameter estimation. In particular, we propose a balanced training procedure, show how to pre-determine the noise level, and propose a regularization scheme for a stable numerical computation.

No Filter Needed Data collected from measurements as well as simulation data display a switching ripple on top of the transient. As we consider the system at an average level, the switching ripple can be interpreted as noise to the proposed GP model. Moreover, experimental data is corrupted by measurement noise. As argued in Sec. 3.2.3, for DS based on GP this is not only an acceptable but rather a needed circumstance. More precisely, linear operator GPs can not be applied to clean signals due to ill-defined kernel matrices. Within this application, no data pre-processing to deal with noisy signals is needed.

Linear Operator Kernel We give the construction of the linear operator GP as provided in Sec. 3.3 for the buck converter (5.2) explicitly. Representing states $I_L(t)$ and $V_{out}(t)$ by GP, the

set of equations reads

$$\mathcal{GP}[I_L(t)] = I_L(t) \tag{5.8a}$$

$$L\frac{d}{dt}\left[\mathcal{GP}[I_L(t)]\right] = \delta V_{in} - V_{out}(t) \tag{5.8b}$$

and

$$\mathcal{GP}[V_{out}(t)] = V_{out}(t) \tag{5.9a}$$

$$C\frac{d}{dt}\mathcal{GP}[V_{out}(t)] = I_L(t) - I_{out}(t) \tag{5.9b}$$

Balancing Balancing (5.8) and (5.9) yields

$$\mathcal{GP}\left[\frac{I_L(t)}{n_{\{I_L(t)\}}}\right] = \frac{I_L(t)}{n_{\{I_L(t)\}}} \tag{5.10a}$$

$$\frac{L\ n_{\{I_L(t)\}}}{n_{\{\delta_{\min}V_{in}-V_{out}(t)\}}}\frac{d}{dt}\left[\mathcal{GP}\left[\frac{I_L(t)}{n_{\{I_L(t)\}}}\right]\right] = \frac{\delta_{\min}V_{in} - V_{out}(t)}{n_{\{\delta_{\min}V_{in}-V_{out}(t)\}}} \tag{5.10b}$$

and

$$\mathcal{GP}\left[\frac{V_{out}(t)}{n_{\{V_{out}(t)\}}}\right] = \frac{V_{out}(t)}{n_{\{V_{out}(t)\}}} \tag{5.11a}$$

$$\frac{C\ n_{\{V_{out}(t)\}}}{n_{\{I_L(t)-I_{out}(t)\}}}\frac{d}{dt}\left[\mathcal{GP}\left[\frac{V_{out}(t)}{n_{\{V_{out}(t)\}}}\right]\right] = \frac{I_L(t) - I_{out}(t)}{n_{\{I_L(t)-I_{out}(t)\}}} \tag{5.11b}$$

For the specific choice of normalization constants, we have the arithmetic mean for $n_{\{I_L(t)\}}$, $n_{\{\delta_{\min}V_{in}-V_{out}(t)\}}$, and $n_{\{V_{out}(t)\}}$. For $n_{\{I_L(t)-I_{out}(t)\}}$, we select a min-max normalization. This is because of a numerically vanishing amplitude of the transient when applying mean normalization for $n_{\{I_L(t)-I_{out}(t)\}}$.

Noise For the application of power electronics parameter estimation, it is beneficial to predetermine noise hyper-parameter. For the states $\mathrm{s} = I_L(t), V_{out}(t)$ this can be directly determined from the signals in steady state $s_{\text{st-st}}$. The variance of one sigma computes as

$$\sigma^2_{\{s_{\text{st-st}}\}} = \frac{1}{N}\sum_{q=1}^{N}\left(s_{\text{st-st,q}} - \overline{s}_{\text{st-st}}\right)^2 \tag{5.12}$$

where $\overline{s}_{\text{st-st}}$ denotes the arithmetic mean for N data points. For implicitly entering signals as in (5.8) and (5.9), the noise value can be determined by uncertainty propagation [102]. That is, for signal $s_{\text{st-st}}$ depending on $s_{\text{st-st},1}, \ldots, s_{\text{st-st},M}$ we have

$$\sigma^2_{s_{\text{st-st}}} = \sum_{p=1}^{M}\sum_{q=1}^{N}\left(\frac{\partial s_{\text{st-st}_q}(s_{\text{st-st},1_q}, \ldots, s_{\text{st-st},m_q})}{\partial s_{\text{st-st},p_q}}\right)^2 \sigma^2_{s_{\text{st-st},p}} \tag{5.13}$$

Regularization Due to the present switching ripple within the data set and relatively high measurement noise present in typically power electronics applications, the GP's noise level is

relatively high. Thus, there is no need for a full regularization scheme in the sense of Sec. 3.2.3. For the given application, a two-sided regularization according to (3.43) is sufficient.

Linear Operator Gaussian Processes for Buck Converter Parameter Estimation Subjected to a current load step, the signals $I_L(t)$ and $V_{out}(t)$ vary smoothly in time at the average level. This can be seen, for example, in Fig. 5.2. Thus, as a basis kernel describing both signals individually, we choose the smooth kernel (3.25). The derivates of (3.25) can be computed as

$$\frac{d}{dt}K_{\text{RBF}}(t,t') = -(t-t')\frac{\sigma^2}{l^2}\exp\left(-\frac{(t-t')^2}{2l^2}\right) \tag{5.14}$$

$$\frac{d}{dt'}K_{\text{RBF}}(t,t') = (t-t')\frac{\sigma^2}{l^2}\exp\left(-\frac{(t-t')^2}{2l^2}\right) \tag{5.15}$$

$$\frac{d}{dt}\frac{d}{dt'}K_{\text{RBF}}(t,t') = \frac{\sigma^2}{l^2}\left(\frac{(t-t')^2}{l^2}+1\right)\exp\left(-\frac{(t-t')^2}{2l^2}\right) \tag{5.16}$$

for $\sigma, l \in \mathbb{R}^+$. Due to the normalization needed for a balanced training, the hyper-parameter σ can be set to the mean of the training signals $\bar{I}_L$ and $\bar{V}_{out}(t)$, respectively. The interval of the lengthscale parameter l can be determined via the system's dynamics. For example, for a transient period of $3 \cdot 10^{-3}$ s, we choose $l \in [10^{-3}s, 10^{-4}s]$. Normalized noise hyper-parameter determined via signals in steady-state are given by

$$\frac{\sigma^2_{\{I_{L_{\text{st-st}}}\}}}{n_{\{I_{L_{\text{st-st}}}\}}},\quad \frac{\sigma^2_{\{\delta_{\min}V_{in_{\text{st-st}}}-V_{out_{\text{st-st}}}\}}}{n_{\{\delta_{\min}V_{in_{\text{st-st}}}-V_{out_{\text{st-st}}}\}}},\quad \frac{\sigma^2_{\{V_{out_{\text{st-st}}}\}}}{n_{\{V_{out_{\text{st-st}}}\}}},\quad \frac{\sigma^2_{\{I_{L_{\text{st-st}}}-I_{out_{\text{st-st}}}\}}}{n_{\{I_{L_{\text{st-st}}}-I_{out_{\text{st-st}}}\}}} \tag{5.17}$$

Then, (5.8) yields the kernel for training (3.13) as

$$\begin{pmatrix} \left(\frac{\bar{I}_L^2}{l^2}\exp\left(-\frac{(t-t')^2}{2l^2}\right)+\frac{\sigma^2_{\{I_{L_{\text{st-st}}}\}}}{n_{\{I_{L_{\text{st-st}}}\}}}\right)_{t,t'} & \left(L(t-t')\frac{\bar{I}_L^2}{l^2}\exp\left(-\frac{(t-t')^2}{2l^2}\right)\right)_{t,t'} \\ \left(-L(t-t')\frac{\bar{I}_L^2}{l^2}\exp\left(-\frac{(t-t')^2}{2l^2}\right)\right)_{t,t'} & \left(L^2\frac{\bar{I}_L^2}{l^2}\left(\frac{(t-t')^2}{l^2}+1\right)\exp\left(-\frac{(t-t')^2}{2l^2}\right) + \frac{\sigma^2_{\{\delta_{\min}V_{in_{\text{st-st}}}-V_{out_{\text{st-st}}}\}}}{n_{\{\delta_{\min}V_{in_{\text{st-st}}}-V_{out_{\text{st-st}}}\}}}\right)_{t,t'} \end{pmatrix} \tag{5.18}$$

Similarly, (5.9) yields

$$\begin{pmatrix} \left(\frac{\bar{V}_{out}^2}{l^2}\exp\left(-\frac{(t-t')^2}{2l^2}\right)+\frac{\sigma^2_{\{V_{out_{\text{st-st}}}\}}}{n_{\{V_{out_{\text{st-st}}}\}}}\right)_{t,t'} & \left(C(t-t')\frac{\bar{V}_{out}^2}{l^2}\exp\left(-\frac{(t-t')^2}{2l^2}\right)\right)_{t,t'} \\ \left(-C(t-t')\frac{\bar{V}_{out}^2}{l^2}\exp\left(-\frac{(t-t')^2}{2l^2}\right)\right)_{t,t'} & \left(C^2\frac{\bar{V}_{out}^2}{l^2}\left(\frac{(t-t')^2}{l^2}+1\right)\exp\left(-\frac{(t-t')^2}{2l^2}\right) + \frac{\sigma^2_{\{I_{L_{\text{st-st}}}-I_{out_{\text{st-st}}}\}}}{n_{\{I_{L_{\text{st-st}}}-I_{out_{\text{st-st}}}\}}}\right)_{t,t'} \end{pmatrix} \tag{5.19}$$

The parameter bounds for the device parameters L and C are chosen in the order of magnitude of the nominal value. Note that each kernel consists of only a single initial GP hyper-parameter $\Theta = \{l\}$ and converter parameters $\phi = \{L, C\}$, which turn to be also trainable hyper-parameters of the constructed linear operator PIGP.

5.2.3 Parameter Estimation by Lightweight Physics-Informed Neural Networks

We propose a lightweight PINN architecture as well as a dedicated training strategy for power electronics parameter estimation. In particular, we propose to utilize relative weights in the loss function, corresponding suppression of learning rates, and ANN representations of power electronics inputs in the physics-informed part of the loss function.

Basics of Neural Networks In general, an ANN can be written as

$$\begin{aligned} f_{\mathrm{NN}}^{x}(x) &= x^{l} = W_l x^{l-1} + b_l \\ x^{k} &= G^{k}(W_k x^{k-1} + b_k), k = 1, \ldots, l-1 \\ x^{0} &= x \in \mathbb{R}^{d} \end{aligned} \tag{5.20}$$

with activation functions G^k, weights W_k and biases b_k [39]. We write f_{NN}^{x} dependent on $\theta = (W, b)$.

Filter ANN can handle noise implicitly via regularization schemes in the loss function. In practice, there is a certain threshold. As we consider the switching ripple as noise concerning the ANN, the results of [12] indicate that proper pre-processing is needed. We apply a FIR filter to get rid of measurement noise as well as the switching ripple. It should be noted that systematic errors are easily produced by filtering. In particular, an offset added to the given data set might be produced, if not carefully applied. Although a small offset is negligible in many applications, the proposed application is very sensitive with respect to this. This is because the parameters to be estimated are given precisely by their signal's relative behavior. Thus, a small deviation in the signal mean caused by an offset yields a shift in the relative behavior and consequently very different converter parameters.

Neural Network Architecture We proposed the following PINN architecture and training strategy, which is illustrated in Fig. 5.3. Each power electronics signal is modeled by an individual neural network. Denote N_1-measurements of M states as $x_j(t) = \{x_{j_i} = x_j(t_i) | i = 1, \ldots, N_1\}$. Correspondingly, we write $u_j(t) = \{u_{j_i} = u_j(t_i) | i = 1, \ldots N_2\}$ for inputs. For each $x_j(t)$ and $u_j(t)$, respectively, a separate neural network is defined in which the individual sizes account for the type of signal. By that, a minimal amount of neurons can be achieved. For example, volatile signals are treated with larger neural network architectures than slowly varying ones. Independent of computational aspects, the usage of relatively small neural networks benefits regularization [86]. The individual outputs $f_{\mathrm{NN}}^{x_j}$ and $f_{\mathrm{NN}}^{u_j}$, depending on network parameters θ_j, do not only have to fit the measured quantities but also match the state-space equations (5.2)

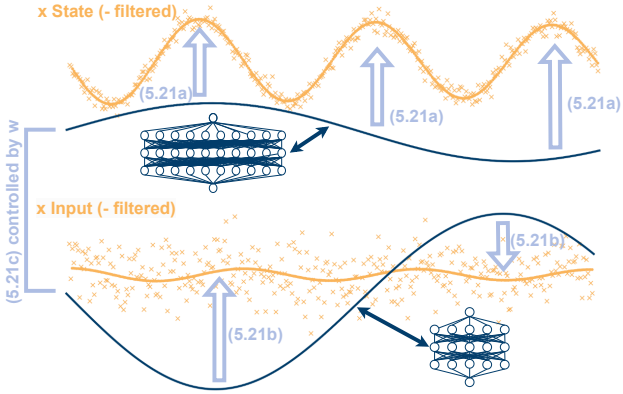

Figure 5.3: Illustration of PINN approach: Data set (orange crosses), (filtered) states and inputs (orange line), neural network architectures and corresponding fit (dark blue) and training strategy with loss function (5.21a) - (5.21c) (light blue).

with converter parameters ϕ. Thus, the loss function reads

$$\mathcal{L}_{\text{PINN}}(\phi,\theta) = \frac{w}{MN_1}\sum_{j=1}^{M}\sum_{i=1}^{N_1}\left(x_{j_i} - f_{\text{NN}}^{x_j}(t_{j_i})\right)^2 \tag{5.21a}$$

$$+\frac{w}{MN_2}\sum_{j=1}^{M}\sum_{i=1}^{N_2}\left(u_{j_i} - f_{\text{NN}}^{u_j}(t_{j_i})\right)^2 \tag{5.21b}$$

$$+\frac{1}{MN_3}\sum_{j=1}^{M}\sum_{i=1}^{N_3}\left(f_{\text{NN}}^{u_{j_i}} - \mathcal{F}_j^{\phi}\left[f_{\text{NN}}^{x_j}(t_{j_i})\right]\right)^2 \tag{5.21c}$$

with $w \in \mathbb{R}^+$. Note that the number of measurements, i.e. N_1, N_2, N_3 may vary in the individual terms of (5.21).

Training Strategy For the proposed lightweight physics-informed neural network architecture with four individual neural networks for states $x_j(t)$, inputs $u_j(t)$, and loss function (5.21), we propose the following training strategy as illustrated in Fig. 5.3. First, note that we introduced a relative parameter w in (5.21) that controls the strength of the contribution of (5.21a) and (5.21b) with respect to (5.21c). During training, the parameter w is controlled. Initially, the randomly initialized network parameters are tuned into a reasonable range by emphasizing (5.21a) and (5.21b). With converging fits of the individual networks, w is decreased to amplify the contribution of the state-space equation part (5.21c). In that way, at least for small architectures like in the proposed approach, (5.21c) yields a sufficient regularization to avoid over-fitting [86]. Note that, in practice this enables the usage $f_{\text{NN}}^{u_j}$ instead of u_j in (5.21c), which solves the convergence issue reported in [140]. Moreover, the learning rate of the converter parameters is also reduced by the factor w to suppress oscillations of converter parameters. Second, as evaluations in (5.21c) can be selected arbitrarily by construction, for N_3 evaluation points, we randomly chose evaluation instances within equally sized intervals in each iteration step. Third, each converter parameter is updated by subtracting the product of the current parameter and

the corresponding gradient. To avoid precipitate direction selection, the converter parameter adaption rate is limited to 10% per training step. Fourth, the learning rate is automatically decreased if the parameter starts oscillating.

5.2.4 Parameter Estimation by Automatic Differentiation Shooting

We propose a forward solver-based parameter estimation approach, called ADShooting. While a similar approach has been investigated previously [138], [142], we propose a training scheme extending the applicability to power electronics in general, and power electronics parameter estimation specifically.

Filter Numerically solving a differential equation is always an initial value problem. Taking the initial value for granted, the solver computes all consecutive states accordingly. However, when provided by a measurement value, the initial value is corrupted by noise. Thus, there exists a quite natural discrepancy between solutions provided by a solver and measured solutions corrupted by measurement noise. Compared to PINNs described above, the requirement and influence of proper pre-processing via an FIR filter is even more important. However, this is only part of the noise-handling strategy needed as will be shown below.

Shooting A classical method to extract parameters from physical systems is similar to a so-called shooting approach. With the initially guessed converter parameter, a forward simulation is performed and compared to the measured signals [137]. For the backward update with respect to the parameters of interest, several methods have been investigated [144]. Among others, automatic differentiation has been used for first-order systems [142]. As forward solvers do not have any parameters up to physical ones, no solver-inherent parameters are present, i.e., $\Theta = \varnothing$.

The ADShooting algorithm is illustrated in Fig. 5.4. The solution for one of the M states x_j is denoted by $f_{\text{solver}}^{x_j,\phi}(t_{j_i})$. Note that the solution of the corresponding state, provided by the solver, depends on the converter parameter ϕ. At first, converter parameters ϕ are randomly initialized. By automatic differentiation, a gradient-based minimization is performed with respect to the converter parameters ϕ. We propose the following loss function

$$\mathcal{L}_{\text{ADShooting}}(\phi) = \frac{1}{MN} \sum_{j=1}^{M} \sum_{i=1}^{N} \left(\frac{x_{j_i} - f_{\text{solver}}^{x_j}(t_i, \phi)}{\eta_{x_j}} \right)^2 \tag{5.22}$$

where η_{y_j} denotes the min-max normalizer of state measurments $\{x_{j_i} = x_j(t_i)\}_{i=1,\dots N}$. The normalization is applied due to different units and orders of magnitude of the involved signals, i.e. voltages and currents. Forward simulation, backward differentiation with respect to the converter parameters, and their corresponding update are iteratively performed.

Training Strategy For the proposed ADShooting approach, we propose the following training procedure as illustrated in Fig. 5.4. First, the update of the converter parameters is done by multiplying the normalized root-mean-square error of the forward simulation with the current parameter value and the cubic root of the loss function's gradient with respect to the converter

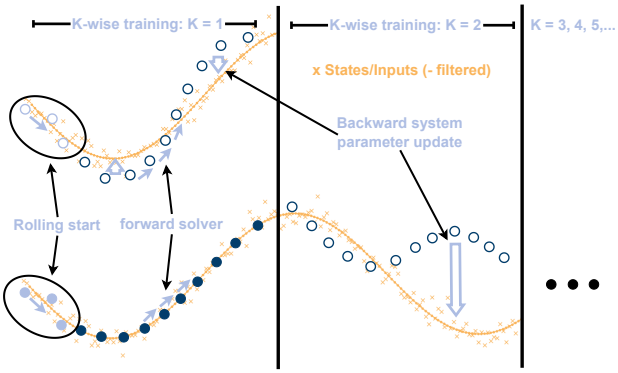

Figure 5.4: Illustration of ADShooting approach: (Filtered) states and inputs (orange) with forward solver (dark blue) and training strategy (light blue).

parameters. The cubic root is chosen as a smoothing operator for high values and as an amplifier for low values of the gradient. Second, we perform a K-wise training procedure to avoid local optima due to periodic signals. This allows us to extend the approach given in [145] to the given application. More precisely, after training with respect to converter parameters for the first subinterval, consecutive data points are appended and initially trained before iterating this process. Third, a premature selection of the training direction is suppressed through update limits by a factor of two in either direction. Fourth, a lowering of learning rates is applied in case of sudden fluctuations in converter parameters. Fifth, as the performance of the solver is especially influenced by noise on the initial state values, a dedicated strategy to mitigate this effect is proposed. Instead of initializing the solver from the same data point, which might be corrupted by noise, we use the first l data points as initial values and change in each iteration step, respectively. In that way, an averaging over noisy initial data points might be provided.

5.2.5 Case Study: LC - Buck Converter

We consider the transient response to a current load step to identify converter parameters L and C. The DC-DC buck converter is simulated with a switching frequency of 12 kHz and a sampling rate of 1000 kHz. We fix the duty cycle $\delta_{\text{set}} = 0.5$. The effective duty cycle is determined according to Sec. 5.2.1.1 and states $\delta = 0.4774$. Besides the noise-free simulation $\sigma_0 = 0$, we add and discuss white noise σ_n. Averaged magnitudes for voltages and currents and noise specifications are given in Table 5.2. We consider three full periods of the transient, which corresponds to 0.018 ms. We investigate the performance of the proposed methods with respect to the sampling rate and the effect of noise. As converter parameters of interest, we use

$$L_{\text{true}} = 1 \text{ mH}, \quad C_{\text{true}} = 900 \text{ µF}$$

for the simulation model. For five different sampling rates, i.e. 20 kHz, 10 kHz, 5kHz, 2 kHz, and 1kHz, the transient response to a single current load step is considered for both cases.

Both cases are down-sampled using a FIR filter with a Nyquist rate of half the desired

Table 5.2: Noise variances σ, absolute average signals $\bar{y}$ and percentage errors by inverse signal-to-noise ratio $\sigma_{\%} = (\bar{y}/\sigma)^{-1}$.

	$\mathbf{V_{out}}[V](\sigma_{\%}[\%])$	$\mathbf{V_{in}}[V](\sigma_{\%}[\%])$	$\mathbf{I_{out}}[A](\sigma_{\%}[\%])$	$\mathbf{I_L}[A](\sigma_{\%}[\%])$
$\sigma_{\mathbf{0}}$	0	0	0	0
$\sigma_{\mathbf{A}}$	2 (0.8)	2 (0.4)	1 (2.5)	1 (2.5)
$\bar{\mathbf{y}}$	237.7	498.1	40.0	40.1

sampling rate. Note that this procedure is needed for a correct down-sampling in the given validation study. However, it should be noted that this is conceptually independent of the proposed pre-processing applied to each method. In this specific case study, preparing the data set for the desired tests coincides with the pre-processing proposed for the PINN and the ADShooting approach. In the case of the linear PIGP, additional noise is added after filtering and down-sampling such that the data sets obey characteristics as provided in Table 5.2.

5.2.6 Results

Model Specifications For the linear PIGP, parameter bounds are set to $l \in [10^{-3}\ \mathrm{s}, 10^{-2}\ \mathrm{s}]$ for both states. The converter parameter bounds are set to $L \in [0.8\ \mathrm{mH}, 1.2\ \mathrm{mH}]$ and $C \in [800\ \mu\mathrm{F}, 1000\ \mu\mathrm{F}]$. While broadened bounds are possible in principle, it should be noted that kernel conditioning turns worse if parameters deviate too much from the optimal one, which might occur during optimization. Thus, relatively tight bounds induced by nominal values support well-defined kernel matrices. Optimization is performed using an L-BFGS-B algorithm with 100 restarts to avoid local minima [98]. For the PINN architecture, we chose three hidden layers, single neuron input and output layers, respectively, and *tanh* activation functions. For $I_L(t)$ and $V_{out}(t)$, the hidden layers consist of 10 neurons each. Five neurons per hidden layer are used for $I_{out}(t)$ and $V_{in}(t)$. For system (5.2), we have $\mathcal{F}_1^{\phi} = L \cdot \frac{\partial}{\partial t}$ and $\mathcal{F}_2^{\phi} = C \cdot \frac{\partial}{\partial t}$ and corresponding input terms $u_1(t) = \delta_{\min} V_{in}(t) - V_{out}(t)$ and $u_2(t) = I_L(t) - I_{out}(t)$. The evaluation number N_3 of (5.21c) is set to 20. For ADShooting, a 4-step Runge Kutta solver is used. The first 5 data points are chosen for a rolling start of the simulation in each iteration step to minimize noise effects. A single normalized value for the expectable error estimates described in Sec. 5.2.1.2 is provided for all cases as the computed values do not vary with respect to the sampling rates in the stated precision.

Data Pre-Processing The pre-processing by an FIR-filter, which is needed for the proposed PINN and ADShooting algorithms and present to prepare the given case study for the linear PIGP, is cross-validated in terms of mean values to ensure the absence of a potential artificial offset, unintentionally added via filtering. Note that the key aspect considered in this cross-validation is the signal mean, while potential artificial harmonics play a minor role in the given application. Artificial harmonics are neglected by all three proposed algorithms by the presented noise-mitigating strategies.

Comparison of Linear PIGP, PINNs, and ADShooting Algorithms The estimated converter parameter ϕ_i is evaluated with respect to the true converter parameter $\phi_{\mathrm{true},i}$ according

to the normalized absolute percentage error as

$$\Delta\phi_i = \left| \frac{\phi_{\text{true},i} - \phi_i}{\phi_{\text{true},i}} \right| \tag{5.23}$$

The obtained results for each converter parameter are provided in Table 5.3 and displayed in Fig. 5.5. Dotted and a dashed line are selected for C and L, respectively, according to the sampling rate and the noise level for the three proposed approaches. Results for ADShooting (blue), PINN (green), and linear PIGP (red) are visualized in Fig. 5.5. Additionally, the expected error estimates (orange) are stated.

In noise-free scenarios, from Table 5.3 and the upper panel of Fig. 5.5, we can deduce that linear PIGPs and PINNs can be applied to all considered sampling rates with relative errors below 1%. Compared to the results obtained by the PINNs, the linear operator PIGP provides better results by factors ranging between 40 and 94. We want to emphasize that the linear operator PIGP does not need any data pre-processing, reducing the risk of error induction by data manipulation schemes. It should be noted that the 1 kHz scenario consists of only 17 data points per signal. This strongly indicates the ability of PINNs and PIGPs to deal with small data sets. The proposed ADShooting algorithm gives only sufficient results for sampling rates with 20 kHz and 10 kHz. Lower sampling rates can not be sufficiently utilized. This can be understood as follows. As the ADShooting needs a relatively small step size by the solver-based construction in the forward simulation, a low sampling rate of the training data causes a lot of intermediate iteration steps before hitting the next training data point. During each iteration, small numerical errors propagate. Thus, the ratio between produced data by the forward solver pronoun to error propagation and training data is relatively low. For low sampling rates of the training data set, the proposed ADShooting algorithm is not capable of dealing with the described mismatch by construction.

In the noisy scenarios, the PINN accuracy range remains unchanged when subjected to noise. The linear operator PIGP accuracy is slightly reduced, still outperforming the PINN results in most cases. Contrarily, noise is an issue for ADShootings. This is evident from the lower panel in Fig. 5.5. Note that not only lower accuracies can be archived, but no clear trend regarding sampling rates can be deduced. This can be understood as follows. The distinct estimation accuracy in the same scenario with respect to both parameters can be rooted in the

Table 5.3: Relative percentage error of converter parameters.

Sampling rate [kHz]	**Noise**	**PINN**		**ADShooting**		**Linear PIGP**	
		ΔC[%]	**ΔL**[%]	**ΔC**[%]	**ΔL**[%]	**ΔL**[%]	**ΔC**[%]
20		0.86	0.27	0.41	0.38	0.02	<0.01
10		0.52	0.64	0.48	0.40	<0.01	<0.01
5	σ_0	0.46	0.88	1.95	1.89	0.09	<0.01
2		0.36	0.23	2.23	1.17	0.23	<0.01
1		0.04	0.10	4.73	2.10	0.05	0.3
εC_{σ_n}[%]	εL_{σ_n}[%]	(8.05)	(9.96)	(8.05)	(9.96)	(8.05)	(9.96)
20		0.55	0.22	0.15	0.14	0.25	0.18
10		0.44	0.13	0.89	1.84	0.26	0.18
5	σ_n	0.15	0.09	0.60	0.48	0.35	0.21
2		0.22	0.34	2.20	0.43	0.46	0.05
1		0.20	0.74	5.97	6.94	0.22	1.43

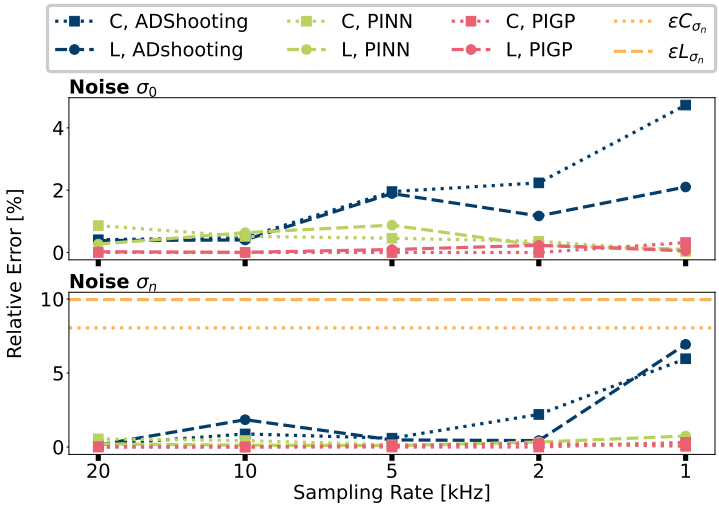

Figure 5.5: Relative percentage error of converter parameters.

loss functions construction. The individual contribution enters the loss additively and thus, balancing effects within both parameter contributions to the loss function exist. Moreover, we use the first five data points for changing initial values for the solver in all scenarios as a noise-handling strategy. Although using the same number enables comparability within the case study, in practical applications, a dedicated analysis of the corresponding signals might be necessary. For a general application, we propose two solutions. First, a high number of starting points for rotating initialization yields stronger averaging effects. This is only possible if sufficient data is available. Second, using multiple, differently initialized ADShootings in parallel yields a consistency check and allows outliers to be detected and sorted out. A comparison to the results obtained in [46] is given in Table 5.4. Compared to the presented case study with the same sampling frequency of 20kHz, lower signal-to-noise values $\sigma_{\%}$ are considered in [46] for the states. In Table 5.4, it is evident that in noisy scenarios, all three proposed approaches outperform results obtained in [46]. It should be noted, that we only refer to the simulation scenarios at this point. The synchronization error discussed in [46] is not relevant for the given PINN architecture due to the solver-free construction, highlighting an additional feature of this lightweight PINN approach. It should be noted, that in the proposed approaches several noise mitigation strategies have been established. Consequently, this smoothing operation allows parameter estimation accuracies to be below the analytical expectable estimates given by uncertainty propagation.

Table 5.4: Comparison of results from noisy scenario as in Table 5.3 to results provided in [46].

	Sampling rate [kHz]	$\sigma_{\%}(I_L)$ [%]	$\sigma_{\%}(V_{out})$ [%]	ΔC [%]	ΔL [%]
Linear PIGP	20	2.5	0.8	0.25	0.18
PINN	20	2.5	0.8	0.55	0.22
ADShooting	20	2.5	0.8	0.15	0.14
Ref. [46]	20	1.8	0.5	0.7	0.3

5.3 Power Electronics Parameter Estimation in the Field

We propose the construction of affine operator kernels enabling PIGP to be applied to parameter estimation for power electronics converters in the field.

5.3.1 Modeling of Power Converters in the Field

5.3.1.1 Buck Converter Topology

The average model for the converter topology given in Fig. 5.6-(a) for inductor current $I_L(t)$ and output voltage $V_{out}(t)$, can be written as

$$\begin{pmatrix} \dot{I}_L \\ \dot{V}_{out} \end{pmatrix}(t) = \begin{pmatrix} -\frac{R_L}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR_B} \end{pmatrix} \begin{pmatrix} I_L \\ V_{out} \end{pmatrix}(t) + \begin{pmatrix} \frac{\delta}{L} \\ 0 \end{pmatrix} V_{in}(t) + \begin{pmatrix} 0 \\ -\frac{1}{C} \end{pmatrix} I_{out}(t) \tag{5.24}$$

with converter parameters L, C describing inductivity and capacity, respectively, duty cycle δ, series resistance R_L, bleeding resistance R_B and input voltage $V_{in}(t)$ and output current $I_{out}(t)$. The gate signals measured by the voltages across the switches are denoted by G_1 and G_2 in Fig. 5.6. Note that there are two input terms present in (5.24), which can be motivated as follows. The second input $I_{out}(t)$ is needed to represent the load. Although we consider the converter as a unit separated from the rest of the system, we can not neglect loading effects. For this reason, an additional parameter Z which accounts for an impedance at the load side, i.e., an equivalent RL-component, has to be included in the converter model.

5.3.1.2 Boost Converter Topology

For the boost topology as in Fig. 5.6-(b), the state-space equation can be stated as

$$\begin{pmatrix} \dot{I}_L \\ \dot{V}_{out} \end{pmatrix}(t) = \begin{pmatrix} -\frac{R_L}{L} & -\frac{1-\delta}{L} \\ \frac{1-\delta}{C} & -\frac{1}{CR_B} \end{pmatrix} \begin{pmatrix} I_L \\ V_{out} \end{pmatrix}(t) + \begin{pmatrix} \frac{1}{L} \\ 0 \end{pmatrix} V_{in}(t) + \begin{pmatrix} 0 \\ -\frac{1}{C} \end{pmatrix} I_{out}(t) \tag{5.25}$$

5.3.2 A Digital Shadow by Affine Physics-Informed Gaussian Processes

We define the proposed affine operator kernels for power electronics parameter estimation. We provide a detailed construction with applications to buck converters in Sec. 5.3.2.1. In Sec. 5.3.2.2, we transfer the construction to boost converters.

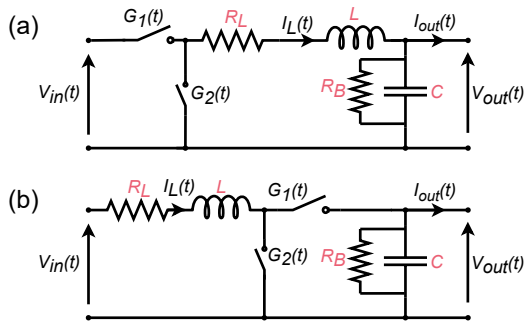

Figure 5.6: Converter topology with converter parameters $\phi = L, R_L, C, R_B$ for (a) buck topology and (b) boost topology.

5.3.2.1 Affine Operator Kernels

Limitations of Linear Operator Kernels for Power Converters in the Field Stating (5.24) including the load parameter Z reads

$$\left(L\frac{d}{dt} + R_L\right) I_L(t) = -V_{out}(t) + \delta V_{in}(t) \tag{5.26a}$$

$$\left(C\frac{d}{dt} + \frac{1}{R_B}\right) V_{out}(t) = I_L(t) - ZI_{out}(t) \tag{5.26b}$$

In (5.26a) the duty cycle δ has to be known. As discussed within the LC toy model in Sec. 5.2.1.1, this is not a trivial task. Additionally, we explicitly state pre-determined duty cycle values for the given case study below. For (5.26b), the parameter Z has to be pre-determined. In practice, this means that additional measurements of the ambient system have to be conducted. In this case, the parameter estimation task can not be solved by pure input/output measurements. Rewriting (5.26) with desired training data set appearing on the right-hand side yields

$$\left(L\frac{d}{dt} + R_L\right) I_L(t) - \delta V_{in}(t) = -V_{out}(t) \tag{5.27a}$$

$$\left(C\frac{d}{dt} + \frac{1}{R_B}\right) V_{out}(t) + ZI_{out}(t) = I_L(t) \tag{5.27b}$$

Thus, in addition to the linear operator appearing on the left-hand side of (5.26) an additional signal must be added and represented by a GP.

Affine Operators To achieve this, we propose the following affine construction. Distinct GPs might be added by a direct sum (3.32). That is, for two independent functions $f(t_1) \sim \mathcal{GP}[f(t_1)]$ and $g(t_2) \sim \mathcal{GP}[g(t_2)]$ with $t_i \in \mathcal{T}_i = \{t_{i_k} | k = 1, \ldots, N \text{ temporal index}, i = 1, 2\}$, the joint kernel is given by the direct sum $\oplus$, i.e.,

$$K_f(\mathcal{T}_1, \mathcal{T}_1) \oplus K_g(\mathcal{T}_2, \mathcal{T}_2) = \begin{pmatrix} K_f(\mathcal{T}_1, \mathcal{T}_1) & 0 \\ 0 & K_g(\mathcal{T}_2, \mathcal{T}_2) \end{pmatrix} \tag{5.28}$$

Although (5.28) seems to be a rather trivial statement for the modeling of states, in terms of power electronics parameter estimation it has a significant impact as it allows the selection of arbitrary parameters to be estimated. Signals not serving as a coupling state via a differential operator, and thus, also their associated parameters (δ, Z), can be included as trainable hyper-parameter into the joint GP. Thus, with GP representations and (5.27) we define

$$\mathcal{GP}[I_L(t)] = I_L(t) \tag{5.29a}$$
$$\mathcal{GP}[V_{in}(t)] = V_{in}(t) \tag{5.29b}$$
$$\left(L\frac{d}{dt} + R_L\right)[\mathcal{GP}[I_L(t)]] - \delta\mathcal{GP}[V_{in}] = -V_{out}(t) \tag{5.29c}$$

and

$$\mathcal{GP}[V_{out}(t)] = V_{out}(t) \tag{5.30a}$$
$$\mathcal{GP}[I_{out}(t)] = I_{out}(t) \tag{5.30b}$$
$$\left(C\frac{d}{dt} + \frac{1}{R_B}\right)\mathcal{GP}[V_{out}(t)] + Z\mathcal{GP}[I_{out}(t)] = I_L(t) \tag{5.30c}$$

for the system representation.

Balancing, Noise Hyper-Parameters and Regularization Balancing (5.8) and (5.9) yields

$$\mathcal{GP}\left[\frac{I_L(t)}{n_{\{I_L(t)\}}}\right] = \frac{I_L(t)}{n_{\{I_L(t)\}}} \tag{5.31a}$$
$$\mathcal{GP}\left[\frac{V_{in}(t)}{n_{\{V_{in}(t)\}}}\right] = \frac{V_{in}(t)}{n_{\{V_{in}(t)\}}} \tag{5.31b}$$
$$\frac{L\ n_{\{I_L(t)\}}}{n_{\{-V_{out}(t)\}}}\frac{d}{dt}\left[\mathcal{GP}\left[\frac{I_L(t)}{n_{\{I_L(t)\}}}\right]\right] - \frac{\delta\ n_{\{V_{in}(t)\}}}{n_{\{-V_{out}(t)\}}}\left[\mathcal{GP}\left[\frac{V_{in}(t)}{n_{\{V_{in}(t)\}}}\right]\right] = \frac{-V_{out}(t)}{n_{\{-V_{out}(t)\}}} \tag{5.31c}$$

and

$$\mathcal{GP}\left[\frac{V_{out}(t)}{n_{\{V_{out}(t)\}}}\right] = \frac{V_{out}(t)}{n_{\{V_{out}(t)\}}} \tag{5.32a}$$
$$\mathcal{GP}\left[\frac{I_{out}(t)}{n_{\{I_{out}(t)\}}}\right] = \frac{I_{out}(t)}{n_{\{I_{out}(t)\}}} \tag{5.32b}$$
$$\frac{C\ n_{\{V_{out}(t)\}}}{n_{\{I_L(t)\}}}\frac{d}{dt}\left[\mathcal{GP}\left[\frac{V_{out}(t)}{n_{\{V_{out}(t)\}}}\right]\right] + \frac{Z\ n_{\{I_{out}(t)\}}}{n_{\{I_L(t)\}}}\left[\mathcal{GP}\left[\frac{I_{out}(t)}{n_{\{I_{out}(t)\}}}\right]\right] = \frac{I_L(t)}{n_{\{I_L(t)\}}} \tag{5.32c}$$

It should be noted that the inclusion of balancing constants into the operators via redefining

$$L\frac{d}{dt} \mapsto \frac{L\ n_{\{I_L(t)\}}}{n_{\{-V_{out}(t)\}}}\frac{d}{dt} = \mathcal{L}_t^L \qquad -\delta \mapsto -\frac{\delta\ n_{\{V_{in}(t)\}}}{n_{\{-V_{out}(t)\}}} = \mathcal{L}^\delta \tag{5.33}$$
$$C\frac{d}{dt} \mapsto \frac{C\ n_{\{V_{out}(t)\}}}{n_{\{I_L(t)\}}}\frac{d}{dt} = \mathcal{L}_t^C \qquad Z \mapsto \frac{Z\ n_{\{I_{out}(t)\}}}{n_{\{I_L(t)\}}} = \mathcal{L}^Z \tag{5.34}$$

allows an independent scaling of the individual signals. This is possible due to the linearity of the individual operators. For the specific choice of normalization constants, we choose the arithmetic mean for all signals. Careful consideration is not needed as no cancelation effects occur in this

type of model as only one signal appears on the right-hand side of (5.31) and (5.32). Thus, amplitudes of corresponding converter signals are not significantly suppressed.

Similar to the linear case, the pre-determination of noise hyper-parameters for each signal can be archived by

$$\sigma^2_{\{s_{\text{st-st}}\}} = \frac{1}{N}\sum_{q=1}^{N}\left(s_{\text{st-st,q}} - \overline{s}_{\text{st-st}}\right)^2 \tag{5.35}$$

where $\overline{s}_{\text{st-st}}$ denotes the arithmetic mean for N data points. Note that uncertainty propagation according to (5.13) is not required.

A two-sided regularization according to (3.43) in combination with the schemes described above is sufficient for numerically stable application.

Affine Operator Gaussian Processes for Buck Converter Parameter Estimation We summarize the proposed affine operator approach described above and specify details for power electronics applications. A graphical representation of the given construction is displayed in Fig. 5.7. For the same reason as in the linear case, we choose the squared exponential kernel (3.25) as a base kernel K_i with the same fixed hyper-parameters and their corresponding bounds.

From (5.31) we get

$$\begin{pmatrix}
\left(K_{\frac{I_L}{n_{\{I_L\}}}}(t,t') + \frac{\sigma^2_{\{I_L\}}}{n_{\{I_L\}}}\right)_{t,t'} & 0 & \left(\mathcal{L}^L_{t'}\left[K_{\frac{I_L}{n_{\{I_L\}}}}(t,t')\right]\right)_{t,t'} \\
0 & \left(K_{\frac{V_{in}(t)}{n_{\{V_{in}\}}}}(t,t') + \frac{\sigma^2_{\{V_{in}\}}}{n_{\{V_{in}\}}}\right)_{t,t'} & \left(\mathcal{L}^\delta\left[K_{\frac{V_{in}}{n_{\{V_{in}\}}}}(t,t')\right]\right)_{t,t'} \\
\left(\mathcal{L}^L_t\left[K_{\frac{I_L}{n_{\{I_L\}}}}(t,t')\right]\right)_{t,t'} & \left(\mathcal{L}^\delta\left[K_{\frac{V_{in}}{n_{\{V_{in}\}}}}(t,t')\right]\right)_{t,t'} & \left(\mathcal{L}^L_t\mathcal{L}^L_{t'}\left[K_{\frac{I_L}{n_{\{I_L\}}}}(t,t')\right] + \mathcal{L}^\delta\mathcal{L}^\delta\left[K_{\frac{V_{in}}{n_{\{V_{in}\}}}}(t,t')\right] + \frac{\sigma^2_{\{V_{out}\}}}{n_{\{V_{out}\}}}\right)_{t,t'}
\end{pmatrix} \tag{5.36}$$

and (5.32) yields

$$\begin{pmatrix}
\left(K_{\frac{V_{out}}{n_{\{V_{out}\}}}}(t,t') + \frac{\sigma^2_{\{V_{out}\}}}{n_{\{V_{out}\}}}\right)_{t,t'} & 0 & \left(\mathcal{L}^C_{t'}\left[K_{\frac{V_{out}}{n_{\{V_{out}\}}}}(t,t')\right]\right)_{t,t'} \\
0 & \left(K_{\frac{I_{out}(t)}{n_{\{I_{out}\}}}}(t,t') + \frac{\sigma^2_{\{I_{out}\}}}{n_{\{I_{out}\}}}\right)_{t,t'} & \left(\mathcal{L}^Z\left[K_{\frac{I_{out}}{n_{\{I_{out}\}}}}(t,t')\right]\right)_{t,t'} \\
\left(\mathcal{L}^C_t\left[K_{\frac{V_{out}}{n_{\{V_{out}\}}}}(t,t')\right]\right)_{t,t'} & \left(\mathcal{L}^Z\left[K_{\frac{I_{out}}{n_{\{I_{out}\}}}}(t,t')\right]\right)_{t,t'} & \left(\mathcal{L}^C_t\mathcal{L}^C_{t'}\left[K_{\frac{V_{out}}{n_{\{V_{out}\}}}}(t,t')\right] + \mathcal{L}^Z\mathcal{L}^Z\left[K_{\frac{I_{out}}{n_{\{I_{out}\}}}}(t,t')\right] + \frac{\sigma^2_{\{V_{out}\}}}{n_{\{V_{out}\}}}\right)_{t,t'}
\end{pmatrix} \tag{5.37}$$

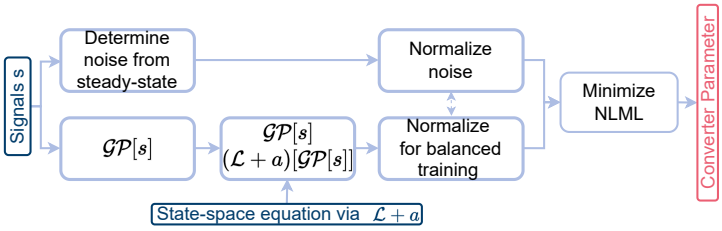

Figure 5.7: Summary proposed approach: input (dark blue), algorithm workflow (light blue), output (red).

5.3.2.2 Affine Operator Kernels of Boost Converters

The proposed approach for (5.25) reads

$$\mathcal{GP}[I_L(t)] = I_L(t) \tag{5.38a}$$

$$\mathcal{GP}[V_{out}(t)] = V_{out}(t) \tag{5.38b}$$

$$\left(L\frac{d}{dt} + R_L\right)[\mathcal{GP}[I_L(t)]] + (1-\delta)\mathcal{GP}[V_{out}] = V_{in}(t) \tag{5.38c}$$

and

$$\mathcal{GP}[V_{out}(t)] = V_{out}(t) \tag{5.39a}$$

$$\mathcal{GP}[I_L(t)] = I_L(t) \tag{5.39b}$$

$$\mathcal{GP}[I_{out}(t)] = I_{out}(t) \tag{5.39c}$$

$$\left(C\frac{d}{dt} + \frac{1}{R_B}\right)\mathcal{GP}[V_{out}(t)] - (1-\delta)\mathcal{GP}[I_L(t)] - Z\mathcal{GP}[I_{out}(t)] = 0 \tag{5.39d}$$

Contrary to the corresponding model (5.30) in buck topology, (5.39) displays an additional dimension due to four converter parameters present in (5.25) for state $V_{out}(t)$. This highlights the capability of the proposed approach to estimate any parameter of interest. Accordingly, (5.39d) is trained on the zero-channel.

5.3.3 Case Study: Power Electronic Converters in the Field

As a case study of the proposed method, we consider a half-bridge converter in buck and on boost configuration, Fig. 5.6. We consider both a simulation scenario as well as a hardware implementation of the same converter. We use the transient response of the converter to a current load step to identify inductivity L, inductive series resistance R_L, capacity C and bleeding resistance R_B from the input voltage $V_{in}(t)$, the output voltage $V_{out}(t)$, the inductor current $I_L(t)$ and the output current $I_{out}(t)$. We refer to the gate signals of the switching elements by $G_1(t)$ and $G_2(t)$, respectively, and δ denotes the duty cycle.

We consider roughly three periods of the transient after the current load step for the parameter estimation. The sampling frequency is 100 kHz, which corresponds to 1000 data points per signal.

For the determination of the duty cycle by the gate signals, a sampling rate of 1000 kHz is used.

Case 1: Simulation Scenario - Buck Converter Topology The simulation of a switching model with a switching frequency of $F_{sw} = 12$ kHz is performed with Simulink using a time step of $1\mu s$. A current load step from 25 A to 40 A is performed. The signals are down-sampled to 100 kHz to match experimental conditions. Dead times and finite slew rates are taken into account. Specific details of the simulation are given in Table 5.5. We consider three different sampling rates, i.e., 100 kHz, 10 kHz, and 1kHz. In addition to the noise-free cases, we consider three different noise levels, where 1%, 2%, and 3% of the corresponding mean signal values are added to the individual signals as white noise.

Case 2: Hardware Measurement Data - Buck Converter Topology A hardware testbed in Fig. 5.8 was constructed to capture measurements for parameter estimation. The testbed centers around a PM1000 IGBT-based switching converter, configured into the buck topology of Fig. 5.6, and rated for 100kW maximum power and 12kHz switching frequency. Control of the converter is performed using a custom in-house digital controller, modulating the converter open-loop with duty of 0.5 with $F_{sw} = 12kHz$ switching frequency. To provide power to the converter, a Keysight N8900 DC power supply rated for 14kW max draw is applied, set to $V_{in} = 300VDC$. For the current load from the converter during testing, a 24kW Chroma DC electronic load is used in the current mode. Measurement data captured from the testbed hardware is taken from probes connected to said hardware and recorded to textual data files by 350MHz 8-channel oscilloscope; the measured quantities are labeled in Fig. 5.6. The converter hardware was operated with current load stepping between 10A and 20A, for four different voltage steps, i.e., 200V to 100V, 400V to 200V, 600V to 300V, and 800V to 400V. Measurements were recorded during the step transition at a 100 kHz sampling rate.

To identify the parameters of the inductor and capacitor separately from the converter for comparison to parameter estimations for the complete system, said components were removed from the converter and placed into their own characterization testbeds. The inductor was connected into series with a 40kW power resistor bank and an Egston power amplifier acting as a programmable AC current source. The amplifier was configured to operate between 5A-40A peak with frequencies from 60Hz to 6kHz and RMS voltage V, current I, and phase θ between these quantities of the inductor were measured for inductance L estimation via impedance using $2\pi FL = im(V/Ie^{j\theta})$. For the capacitor, it was placed in parallel to a 25kΩ resistor R and the Keysight DC power supply set to $V_0 = 300VDC$. With the capacitor fully charged to V_0, the supply was disconnected from the capacitor by a contactor and the voltage V of the capacitor measured as it discharged into the resistor. The measurement was applied to estimate the

Table 5.5: Simulation Specifications.

Converter specifications				
L	C	R_L	$1/R_B$	δ
500 μH	900 μF	0.1 Ω	1/24150 1/Ω	0.5

Load Slew Rate	T_{dead}	F_{sw}	
12 $V/\mu s$	$2 \cdot 10^{-6} s$	12 kHz	

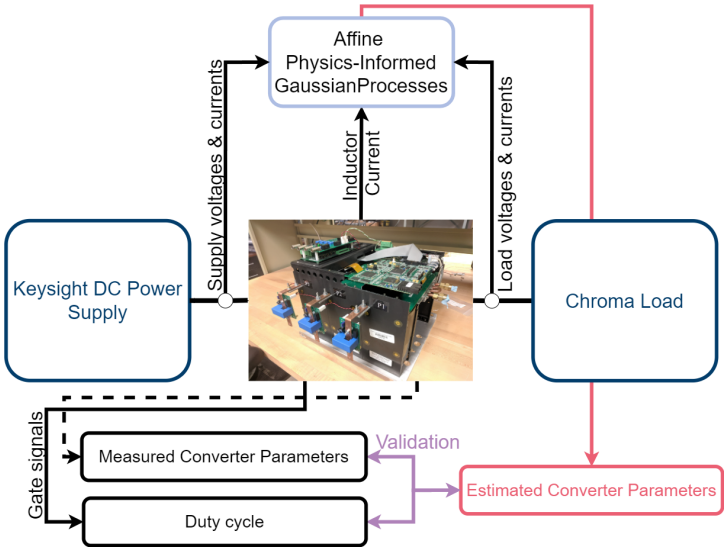

Figure 5.8: Power electronic testbed for data acquisition.

capacitance C of the capacitor via the RC time constant using

$$V(t) = V_0(e^{-t/RC}) \tag{5.40}$$

Case 3: Hardware Measurement Data - Boost Converter Topology The same setup as described in Case 2 is used, but operated in boost topology, i.e., swapped DC source input and halved current load to stay within the specified power supply. The boost converter hardware was operated with current load stepping between I_{out} =10A and 20A for output voltage $V_{out} = 300V$.

5.3.4 Results

For the evaluation of the system parameter ϕ, normalized errors according to

$$\Delta\phi = \left| \frac{\phi_{\text{estim}} - \phi_{\text{true}}}{\phi_{\text{true}}} \right| \tag{5.41}$$

are given in percent.

5.3.4.1 Model Specifications

Selected parameter bounds for the L-BFGS-B optimizer are naturally given by the system characteristics and provided in Table 5.6. The nominal value for the duty cycle serves as an upper bound. This is because dead times reduce the interval length of the open-state period. The lengthscale bounds for the GP hyper-parameter l_1 and l_2 are in the range of the period of one transient, which is approximately given by $3.3 \cdot 10^{-3}s$.

Table 5.6: Range of parameter bounds with nominal values.

	$L[\mu H]$	$C[\mu F]$	$R_L[\Omega]$	$1/R_B[1/\Omega]$
bounds	300-700	700-1100	0.01-1	10^{-8}-10^{-2}
nominal	500	900	0.1	$4.14 \cdot 10^{-5}$
	δ	l_1	l_2	
bounds	0.45-0.5	10^{-3}-10^{-2}	10^{-3}-10^{-2}	
nominal	0.5	$\approx 3.3 \cdot 10^{-3}$	$\approx 3.3 \cdot 10^{-3}$	

Regarding the duty cycle, the computed numerical values by the pulse width of the gate signals with a sampling frequency of 1000 kHz are given in Table 5.7. The range in the determination for each duty cycle arises from the finite sampling rate, measurement errors, and truncation errors specifying the gates' state. Consequently, variations in the corresponding interval length of the pulse width and the switching period occur.

5.3.4.2 Results Case 1: Theoretical Insights from Simulation Scenarios

The test for the simulation scenarios of Case 1 has been performed on a laptop equipped with a 1.60 GHz Intel Core i5-8265U CPU with 24 GB of RAM. The proposed approach is implemented in Python using standard libraries like Numpy and Pandas. For Case 1, 50 restarts of the optimizer are selected to avoid local minima.

The results obtained by (5.29) and (5.30) are given in Table 5.8, where the normalized relative error according to

$$\Delta\phi = \left| \frac{\phi_{\text{estim}} - \phi_{\text{true}}}{\phi_{\text{true}}} \right| \tag{5.42}$$

are given in percent with respect to the true parameter values ϕ_{true} provided in Table 5.5.

Two facts are worth noting. First, a systematic error might be introduced into the converter signals by down-sampling using an anti-alias filter as discussed in [5]. In other words, although a well-known simulation scenario is considered, the data pre-processing needed to provide the case study might cause some accuracy loss with respect to converter parameter estimation. Second, the accuracy of the estimated parameters by the proposed approach is consistently obtained. More specifically, the same results are obtained for each execution of the proposed algorithm. This has to be seen in contrast to neural network approaches like [6], [46], where a certain amount of variation in the obtained accuracy naturally occurs as neural networks almost always end up in local minima.

The first thing to note from Table 5.8 is that the proposed approach yields better results for the noisy scenarios than for the clean cases. This is because the proposed approach is not only able to address noisy data, but in fact, noise is a requirement. This is due to numerical errors caused by ill-conditioned kernels resulting from clean data. The proposed approach is

Table 5.7: Computed values for the duty cycle.

	δ_{nom}	δ_{G_1}	δ_{G_2}
Case 1	0.5000	$0.4756 - 0.4878$	$0.4756 - 0.4878$
Case 2	0.5000	$0.4814 - 0.4938$	$0.4814 - 0.4938$

Table 5.8: Relative errors and runtime for simulation test with buck topology - Case 1.

Noise	Sampling rate	ΔL [%]	ΔR_L [%]	$\Delta\delta$ [%]	ΔC [%]	$\Delta 1/R_B$ [%]	Execution time
0%	100 kHz	2.6	100.4	4.0	0.3	45.4	3.9 d
	10 kHz	19.9	362.9	1.6	4.8	5.2	4.2 min
	1 kHz	35.2	745.7	6.9	4.3	117.9	4.3 s
1%	100 kHz	0.6	2.4	0.1	0.9	11.5	2.3 d
	10 kHz	0.1	4.8	4.3	0.1	60.5	16.1 min
	1 kHz	0.3	6.4	3.2	0.1	75.85	50.9 s
2%	100 kHz	0.9	2.5	0.1	0.2	141.5	1.6 d
	10 kHz	1.5	4.4	4.4	1.5	141.5	12.9 min
	1 kHz	1.9	12.5	4.1	2.9	141.5	45.9 s
3%	100 kHz	0.8	29.5	1.0	1.5	141.5	1.8 d
	10 kHz	0.3	33.8	3.4	1.0	141.4	14.4 min
	1 kHz	0.6	36.8	3.2	0.6	141.4	49.6 s
σ_{meas}		2.0	47.4	4.2	-	-	

sufficient to provide a stable algorithm for noisy data, but it comes to its limits in the clean setting. However, this is an artificial problem with no significance in real-world applications.

For the noisy test cases, we regard accuracies below a relative measurement deviation σ_{meas} obtained for converter parameters L, R_L, and δ as obtained from the hardware test as sufficient. The corresponding values are provided in the last row of Table 5.8. For C and R_B, such values are not provided in the given case study, as these parameters depend on the operational conditions of the converters. The proposed approach successfully estimates converter parameters L and R_L. Also, the duty cycle δ is estimated accurately, except for two cases where the relative error is 0.1% and 0.2% above the benchmark deviation. The capacity C is obtained with good accuracy below 1% for 1% noise case. The higher noise cases are also quite accurate, but no trend with respect to sampling rate or noise level can be obtained. We expect that this is caused by a systematic error introduced by down-sampling, cf. [5]. Nevertheless, the accuracy for all noise cases is comparable with the expected measurement error from hardware tests. The bleeding resistor, i.e., $1/R_B$, is estimated with a relative error between 11.5% and 75.9% for the 1% noise level. The value for the higher noise accuracies is the upper limit of the parameter bounds.

Concluding more generally, converter parameters L, C, and δ can be estimated with higher accuracy than resistances R_L and R_B, which can be explained in terms of sensitivity. That is, these parameters contribute less significantly to the converters' dynamics due to their comparably small numerical value. While, for example, the product of the (nominal) capacity and mean value of $V_{out}(t)$ is of the order $\mathcal{O}(C \cdot \bar{V}_{out}(t)) \approx 1.3$, we have for $\mathcal{O}(1/R_B \cdot \bar{V}_{out}(t)) \approx 0.006$. Thus, the contribution of $1/R_B$ is suppressed by a factor of approximately 200 with respect to the contribution of C, and consequently, is less significant. As the proposed algorithm does not have to solve the state-space equation but is instead 'physically informed', the more relevant the corresponding contribution, the better the estimation performs. It should be noted that similar effects occur for individual hardware measurements, as small resistances are harder to determine

due to the contact resistances of the measurement devices.

The computational complexity of the proposed approach scales cubically with the number of data points included, as it is governed by matrix inversion via Choleski decomposition present in (3.19). Run times of the proposed approach are provided in Table 5.8, ranging from seconds to days. It is worth noting that the results obtained for the 1 kHz test cases are quite accurate, as the data set for each converter signal consists of only 9 measurement points. Regarding RAM used, the limit of the given setup concerning the laptop used can be estimated at around 10000 data points taken into account. That limits the number of states/inputs that can be taken into account to ten, subject to the given test case (three transient periods, 100 kHz sampling rate). The 100 kHz test cases comprise 3,000 data points.

Some general conclusions can be drawn from the simulation test cases. The proposed approach requires noisy data and estimates converter parameters independent of the noise level. The performance of the proposed approach depends on the sensitivity of the corresponding converter parameters to the respective dynamics. Down-sampling of the data should be avoided, with the drawback of significant extensions of run times from seconds (1kHz) to days (100 kHz).

5.3.4.3 Results Case 2: Parameter Estimation under Operational Conditions - Hardware Measurement Data

For the second case study, the proposed approach is tested with measurements conducted from hardware tests with buck topology. Results are provided in Table 5.9. First, the gap between nominal and measured parameter values should be noted, which illustrates the need for the proposed approach.

The results provided for (5.29) in Table 5.9 show that the proposed approach can estimate the converter parameters L, R_L, δ up to experimental characterization accuracy. More precisely, the provided results by the proposed approach, as well as the corresponding standard deviation, match the experimentally characterized values.

Converter parameters estimated by (5.30) are stated in Table 5.9 and values for the capacitance C are visualized in Fig. 5.9. It can be seen that the estimated parameter values of C follow the voltage dependence provided in the experimental characterization. It should be noted that

Table 5.9: Estimated converter parameter values for hardware test with buck topology - Case 2.

Model	Parameter	Nominal	Measured	Estimated
	$L\ [\mu H]$	500.0	356.6 ± 7.2	361.2 ± 8.1
(5.29)	$R_L\ [\Omega]$	-	0.19 ± 0.09	0.13 ± 0.05
	δ	0.5	$(0.48 - 0.49) \pm 0.02$	0.4975
	$C_{100V}[\mu F]$	900.0	815.1	816.9
	$_{200V}[\mu F]$	900.0	797.9	798.6
(5.30)	$_{300V}[\mu F]$	900.0	801.4	810.0
	$_{400V}[\mu F]$	900.0	802.2	787.1
	$\overline{1/R_B}[1/k\Omega]$	-	-	0.02

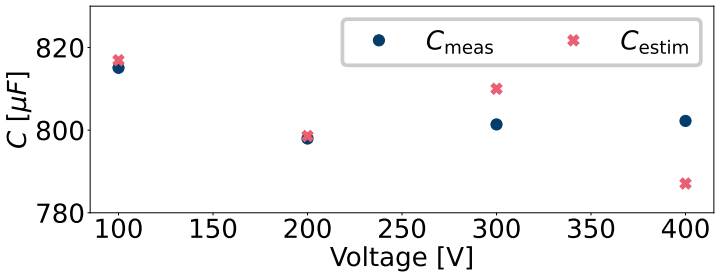

Figure 5.9: Estimated capacity $C_{\text{estim}}[\mu F]$ (red crosses) and measured capacity $C_{\text{meas}}[\mu F]$ (blue dots).

the experimental identification of component parameters via (5.40) displays increasing errors with increasing output voltage, which might explain the growing deviation with higher voltage values. For simplicity, we state the average value $\overline{1/R_B}$ over the four voltage levels to provide results for the bleeding resistor due to the lack of experimental verification.

Compared to the results provided in [46] with the relative error above 10%, the proposed approach outperforms the estimation algorithm provided by [46] in real applications where noise can not be neglected. It should be noted that the estimation errors provided in [46] are averaged error values. The proposed approach estimated L with a relative error of 1.3% with respect to averaged measured values and capacities C_{meas} depending on the output voltage $V_{out}(t)$ within a range between $0.1\% - 1.9\%$. For the equivalent series resistance R_L, the relative error of the estimation is 31.1% and appears to be quite large. However, when taking measurement errors due to probe contact and the internal resistances of the measurement device itself into account, which affect small resistances in particular, the results might be interpreted in such a way that the estimated parameter value reflects the real converter parameter closer than the measured value corrupted by measurement errors. However, due to the data used in this case study, we can not finally validate and answer this indication. Although a precise determination might yield theoretical insights as well as possibly increase confidence in the proposed approach, from an application perspective, the relevance is questionable.

We conclude that the proposed approach enables an estimation of converter parameters, subject to the specific point of operation.

5.3.4.4 Results Case 3: Generalizability - Boost Topology and Hardware Measurement Data

The results for the hardware test with the boost topology are given in Table 5.10. It can be seen that the results provided by model (5.38) are very close to the experimentally found parameter values within the corresponding error bounds. The normalized relative error between the experimentally found (including measurement uncertainties) value $(356.6 + 7.2)\ \mu H$ and the estimated parameter value of 365.2 μH is 0.4%. Similarly, the deviation of R_L is 0.01 Ω. A possible explanation for the small deviation compared to the buck converter results is as follows. When the gate signals to the converter are off, such as during dead-time in switching, the anti-

Table 5.10: Estimated converter parameter values for hardware test with boost topology - Case 3.

Model	Parameter	Nominal	Measured	Estimated
	$L\ [\mu H]$	500.0	356.6 ± 7.2	365.2
(5.38)	$R_L\ [\Omega]$	-	0.19 ± 0.09	0.09
	δ	0.5	$(0.48 - 0.49) \pm 0.02$	0.4968
(5.39)	$C_{300V}[\mu F]$	900.0	801.4	809.5
	$1/R_B[1/k\Omega]$	-	-	0.06

parallel diodes across the switching elements will conduct based on their across voltage biasing and inductor current flow. For the buck converter in the presented configuration, the diode across the bottom switch will primarily conduct during this off/dead time. In the boost converter configuration during the dead time, the diode across the top switch will instead primarily conduct. As the diodes of the two switching elements can have slightly different characteristics due to component tolerances and age, along with having different conduction intervals in the two different topologies, slight deviation can occur between buck and boost topology parameter estimations in the proposed approach.

Regarding model (5.39), we obtain the same estimation an estimated capacitance value of 809.5 μF, which is remarkably close to the value found in buck topology of 810.0 μF. This might be interpreted as a further hint that the proposed approach is closer to the real parameter values as present during operation than a measurement of the disconnected component. For the inverse bleeding resistor, we obtain a deviation between the estimation of the buck and the boost topology by a factor of 3. While this seems to be a rather big deviation, it should be noted that the corresponding numerical value of the parameter is significantly lower than the capacitance. Thus, less numerical sensitivity is present to determine this comparably small value.

5.4 Conclusion and Characterization of Power Electronics Digital Shadows

5.4.1 Discussion

In the first part of this chapter, we propose, analyze, and compare three power electronics parameter estimation algorithms. We propose and test a linear PIGP, a lightweight PINN, and a ADShooting algorithm with data obtained in a simulation scenario. The obtained results show that the linear PIGPs provides the best results among all presented methods. Moreover, we show that no data pre-processing is needed, yielding a robust application. We show that the linear PIGP and the PINN approach can be utilized to estimate power electronics parameters, independent of the considered sampling rate. In particular, small data sets are sufficient, e.g., 1 kHz sampling rates with a 0.018 ms training period. Moreover, the results indicate an immunity of the proposed linear PIGP and PINNs to noise. For the PINNs, the noise immunity is only achieved by dedicated pre-processing. Given the high sensitivity of ADShooting to noise, a dedicated strategy to improve the performance of the method is proposed and tested. The

results show that the mixed forward-backward ADShooting approach is preferably trained on larger data sets, i.e., higher sampling rates.

In the second part of this chapter, we propose a power electronics parameter estimation algorithm by defining affine operator kernels applied in PIGPs. We develop the concept and provide the construction of PIGPs using affine differential operators, discuss the training procedure including a proper self-consistent normalization scheme, show how to utilize noise hyperparameters, and provide a dedicated algorithmic scheme explicitly. We show that the proposed algorithm is able to accurately estimate physically relevant parameters from non-invasive state and input measurements of power electronic converters in the field. In particular, we show that the proposed approach is applicable in hardware tests where noise can not be neglected.

5.4.2 Characterization of Power Electronics Digital Shadows

In the given application, the provided algorithms for power electronics parameter estimation can be classified as a DS. An extension to provide DTs could be given by follow-up processes, where the estimated parameters are provided to enhance modeling and control schemes. For example, after estimating device parameters accurately, the given results might be provided to simulators to produce a precise copy of the specific component. This allows the inclusion of particular devices into the simulation context, instead of typical, average devices. Alternatively, the given estimated parameters could be included in dedicated model-based control schemes.

A classification of the provided models is visualized in Fig. 5.10. Although not every physical effect has to be displayed, based on an effective/equivalent description of the power electronics devices, the proposed DSs provides a high level of detail. The data transfer can be classified as offline and explicit. The provided models do not show any aggregation. The proposed DSs are updatable, in the sense that they provide the parameters for the specific data set, which might depend on the specific operational conditions. The algorithms are not self-validating, meaning they always provide a parameter value subjected to the implicit assumption of a valid device description in terms of state-space equations. A slight exception to this can be observed in both PIGPs. In cases where the provided data set and the assumed state-space equations deviate significantly, the conditioning of linear and affine operator kernels significantly increases, yielding a strongly ill-defined kernel matrix that is numerically not computable. We give an illustrative example. In the course of this work, it turns out that numerically acceptable errors occur for condition numbers below $5 \cdot 10^5$, with issues starting for condition numbers around 10^6. For the affine operator buck converter model the latter is the case when neglecting the load parameter Z. While this yields bad but computable results, in the absence of the bleeding resistor R_B, the condition number rises above 10^{10}. In this case, the proposed PIGP is not computable. Although the contribution of R_B in terms of signals present in the state-space equation is negligible on top of the fact that the determination of the actual value of R_B might not be very relevant from an engineering point of view, an explicit inclusion of R_B can not be avoided. Note that this underlines the argument for full matrices given in Sec. 5.1.3. Even worse condition numbers of the order of 10^{33} can be reached when working with clean data sets, i.e., in the absence of noise. Finally, the proposed linear and affine PIGP can be classified as a probabilistic model. States and inputs are modeled probabilistically by the GP, while converter parameters are scalars. The

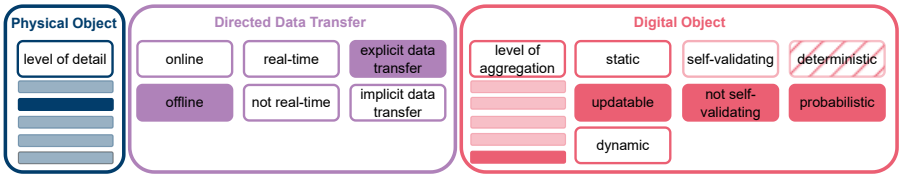

Figure 5.10: Classification of DSs for power electronics parameter estimation.

PINN and the ADShooting algorithm are point estimators for states, inputs, and parameters.

5.4.3 Outlook

As a future research direction, the proposed parameter estimation methods could be applied in different fields. This might be, for example, power systems, heat and gas nets, or a combination thereof. Especially within multi-energy systems, a model reduction by effective/equivalent models might yield a significant computational benefit while displaying real systems, in the sense of a DSs or DTs. Another promising candidate to be investigated is the effective modeling of battery systems. While exact chemical states and reactions are hard to monitor in practice, an effective model trained on individual batteries might be a reasonable description. Beyond pure parameter estimation, the active integration into control schemes could be investigated. This might be in terms of knowledge about the specific devices as described in this chapter. Alternatively, parts of the control loop could be effectively modeled by the proposed methods. A simple example of this has been presented in terms of effective duty calculations in this chapter.

6

A Digital Twin for Building Pseudo-Measurements

In this chapter, DTs are provided to generate pseudo-measurements for building operations. For practical reasons, measurements of the building's state are usually performed at accessible locations. Integrated into light switches, heater valves, or mounted at the ceiling, typical sensor locations only estimate information at user-relevant locations. We construct a DT framework providing this information usable in control strategies in terms of pseudo-measurement. In Sec. 6.1, we start with contextualizing and motivating the provided DT application. The developed algorithm is presented in Sec. 6.2. With the case study described in Sec. 6.3, we provide results of the given DT in Sec. 6.5. To finalize this chapter, we conclude and characterize the presented DT in Sec. 6.6. This chapter is based on the publications [2], [7].

6.1 Introduction

6.1.1 Motivation

People spend approximately 60 % to 70 % of their time indoors [146], and this share is increasing even further [147]. As indoor air quality (IAQ) and thermal comfort directly relate to occupant well-being, productivity, and building energy consumption [36], [37], monitoring and control of IAQ and thermal comfort are essential elements of successful building operation.

Guidelines on thermal comfort have been implemented in the form of international standards, e.g., through ISO 7730:2005 [148]. These standards are based on the calculation of a predicted mean vote and a predicted percentage of dissatisfied. Recent studies dealt with more adaptive approaches or even individual thermal comfort models that make use of numerous in-depth measurements such as skin temperature or heart rate measured via wearables [38]. Modern IoT-enabled technologies and sensor networks can significantly improve occupant comfort and outperform classical approaches [38]. However, they also require a high number of additional sensors, a sophisticated IoT-setup [149], and a high level of data protection [150], [151].

Two prominent and widespread indicators for building indoor comfort are the temperature [152] and the CO_2 level representing air quality [153]. Both quantities are commonly measured once per building space, e.g., in the form of a sensor integrated into a room control panel. Based on the obtained air temperature and a user-defined temperature set-point, either the building's central heating or a room-individual heating appliance, e.g., a radiator, is operated to establish thermal comfort [154]. In this typical scenario, the control performance, and thus the thermal comfort, is strongly affected by sensor placement and accuracy. Nowadays, the trend is moving towards multiple stationary sensors per room at different locations, even in existing buildings [154]. It would therefore be of great benefit to be able to use these measurements to predict individual, close-to-person temperature profiles and thereby unlock comfort improvements and energy-saving potentials in building operation. Both standard (e.g., P/PI/PID) and advanced control concepts, e.g., MPC could thus be upgraded to provide comfort specifically to the occupied zone. At the same time, additional (personal) sensor hardware would be avoided, which otherwise would result in additional financial effort and, as being located in the occupied zone, could constitute a disturbance to the occupants.

6.1.2 Related Work

Reviewed in [155], DTs are applied for thermal comfort and energy efficiency in buildings. In this field, applications range from specific quantities, such as energy saving by lighting [156], to heterogeneous data approaches for energy management [157]. However, so far, dedicated algorithms applied as in our developed conceptual structure have not been used to provide indoor close-to-person temperature or CO_2 pseudo-measurements [155]. In general, pseudo-measurements have been considered in the domain of distribution grids [158], uncertainty quantification of power system state estimation [159] as well as for estimation of GPS outages [160].

6.1.3 Preliminary Remarks on Technical Realizations

We motivate the selected Corr-GP algorithm for the application in building pseudo-measurement generation. In line with the considerations in Ch. 5, the physical description of rooms and buildings could be given in terms of state-space equations as visualized in Fig. 6.1-(a) or describes in [161]. The states are given by the room and the heater temperature T_r and T_h, respectively. The system inputs are the ambient outside temperature T_a, the ambient building temperature T_b, and the heat flow into the heater $\dot{Q}$. For positive system parameters, we have transition coefficients R_{ra}, R_{hr}, R_{rb} and heat capacities C_h, C_r. In common building operations such as the LLEC, temperatures are measured with a sampling rate of 15 minutes [154]. For the application of digital representations based on state-space equations, this is not sufficient. This is because the timescale of the heater dynamics is in the range of several minutes. While more frequent temperature measurements are not an issue for modern temperature measurement devices, this is different for the measurement of the heat flow $\dot{Q}$. Depending on the energy carrier utilized for heating, at the building level, this quantity might be provided on a coarse temporal basis. However, at room level, there is no such measurement possible due to the lack of suitable measurement devices. While there are sensors on the market that could provide an estimated value for $\dot{Q}$ based on temperature differences, the provided measurements lack accuracy and temporal resolution. In particular, it turns out that the sampling rate of $\dot{Q}$ should be in the range of 30 seconds due to the sensitivity of the corresponding state-space equation on this specific input [12]. Measurement devices possibly included within the LLEC might provide a value once per day. In addition to the issues mentioned, installing measurement devices at each heater, in each room, for each building causes a fair amount of effort in terms of workload and financial investments. In this regard, whether the effort is worth it in terms of energy savings provided by the DT pseudo-measurement is questionable and has to be determined.

An alternative approach, which is visualized in Fig. 6.1-(b), is the assumption that signals obtained within a room, such as temperature profiles, display a certain amount of correlations. We propose a detailed construction, use case, and results in this chapter.

6.1.4 Contribution

We develop the concept of DTs for indoor temperature and CO_2 pseudo-measurements and propose a realization by Corr-GP. First, we provide temperature and CO_2 pseudo-measurements for a user-relevant position based on two permanently installed sensors, called *reference measurements* hereafter, in a real-life office building. In addition to the reference measurements installed at standard locations, an additional measurement is collected at a user-relevant position. Based on the three measurements, a Corr-GP is trained. Then, we show how to define a non-zero prior mean function in the context of thermal- and CO_2-DTs for pseudo-measurement generation. This accounts for training the proposed approach by deviations from typical patterns instead of the signals themselves. With that, not only a reliable long-term application can be achieved without the need for retraining the DT, but also a single reference sensor suffices as input. We compare the proposed approach to a trivial algorithm TRIV model as well as a standard time series VARMA model [85]. We give a two-fold evaluation. First, the performance of the DT is evaluated and compared to the given alternative algorithms with respect to the consecutive 14

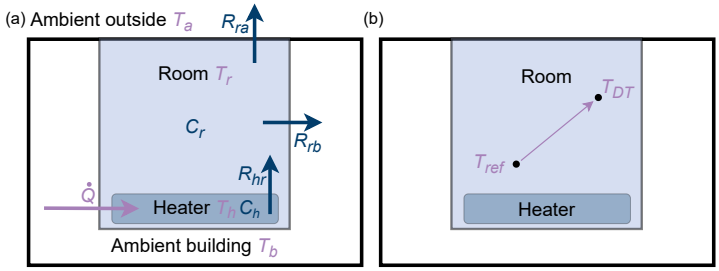

Figure 6.1: Different modeling approaches: (a) state-space approach, (b) correlation-based approach.

days after the training period. Second, a seven-month long-term analysis is provided. We show that the approach with multiple DT-inputs outperforms the given alternatives, due to its ability to determine inherent correlations across different measurements. The provided results indicate the superior ability of Corr-GPs to incorporate multiple signals, providing an algorithmic solution for multi-sensor-equipped rooms and buildings. Moreover, we show that the Corr-GP model with a single DT-input can be reliably utilized in long-term applications. That is, a model trained in summer can be accurately operated during the winter month.

6.2 Pseudo-Measurements by Correlated Gaussian Processes

Published in [2], [7], in this section, we propose a Corr-GP framework for the pseudo-measurement generation of temperature and CO_2 signals at user-relevant locations.

6.2.1 Technical Conceptualization of Building Digital Twins

The proposed concept of a DT for pseudo-measurements is presented in Fig. 6.2. In Fig. 6.2-(a), the training phase of the model is shown. For a certain period, all collected signals are used as training data for the DT. In the execution phase, depicted in Fig. 6.2-(b), the reference signals are continuously included in the model, while the signal of interest is predicted at times t^* of interest. Recall from Sec. 2.3.3.1, that there are two notions of inputs and outputs, visualized in Fig. 6.2. Reference signals are continuously included (DT input) while the signal of interest is predicted (DT output). For further clarification, we write $\mathrm{DT}_{\mathrm{input}}^{\mathrm{output}}$ in the remaining part of this chapter. However, inside the DT realization by a Corr-GP, the reference signals as well as the predictions are both outputs of the multi-output GP (GP output), as functions of time (GP input).

On an intraday basis, every new reference measurement is continuously included in the DT to update the adaptive model in a rolling horizon fashion. As visualized in Fig. 6.3, for every new day N, the data of the previous day $N-1$ is deleted on a daily basis. The proposed approach is only updated with the data from the day on which the pseudo-measurements should be provided. The DT runs between 8:00 am (t_{start}) and 7:00 pm (t_{end}). In addition, an initialization phase, labeled as *init* in Fig. 6.3, is used, where prior to the operational period, reference measurements

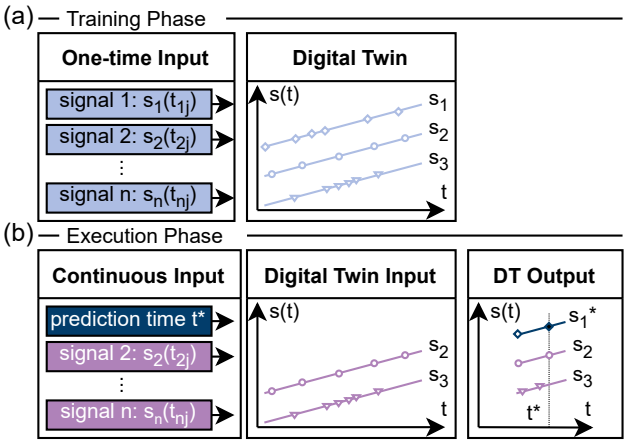

Figure 6.2: Proposed DT concept.

are included.

6.2.2 Technical Realization of Building Digital Twins

6.2.2.1 Correlated Kernel for Building Pseudo-Measurements

Correlations between differently located measurements of temperature signals $T(t), t \in \mathcal{T}$, as well as CO_2 signals, within a room are given. For the given application with q reference sensors, the kernel can be written according to (3.78), as

$$K_{\mathrm{LCM}}(t,t') = \sum_{q} B_q^{[q+r]} \otimes K_{\mathrm{basis},q}(t,t'),\ t,t' \in \mathcal{T} \tag{6.1}$$

The amount of LCM-terms results from the number of reference sensors q plus the number of pseudo-measurements r that should be provided.

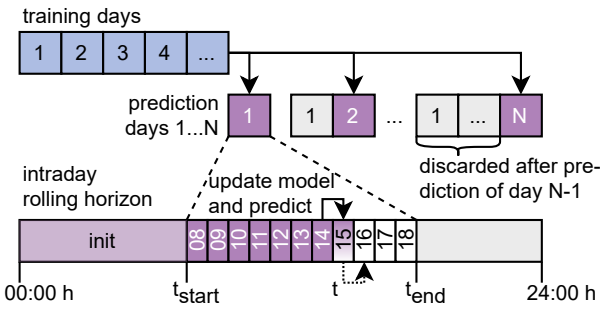

Figure 6.3: Intraday rolling horizon for DT pseudo-measurement.

The required data set defining the training period can be deduced from the following facts. The single-signals hyper-parameters describing intraday characteristics can be determined by a single day of data. To determine the correlation hyper-parameters $\{b_{1,1}, \ldots, b_{q,q}\}$ data reflecting the correlations must be provided. For the given case study, the minimal training period reflecting this is given by a data set of five days.

For the given approach, the noise hyper-parameters σ_n reflect the input sensor noise level. As part of the GP, in this application, these hyper-parameters can be trained on the provided data set. Thus, the sensor accuracy does not have to be individually considered but is determined from the training data set.

Temperature Profiles Room temperature profiles do not vary extensively during the day. This type of smooth variation can be captured by the rational quadratic kernel (3.26), given by

$$K_{\mathrm{RQ}}(t, t') = \left(1 + \frac{|t - t'|}{2\alpha l^2}\right)^{-\alpha} \tag{6.2}$$

with $\alpha, l \in \mathbb{R}^+$. It should be noted, that the prediction scheme described in Fig. 6.3, is a consequence of this choice as (6.2) is defined by an Euclidean distance. An operation like this implies that the model is staying close to the data points of the training data set. In addition, a scaling by a constant factor is applied to ensure stable scaling during operation.

CO_2 Profiles Contrarily to temperature profiles, CO_2 profiles tend to vary more rapidly. Thus, a less smooth kernel, i.e., the Matérn kernel (3.29)

$$K_{\mathrm{M}^{5/2}}(t, t') = \left(1 + \frac{\sqrt{5}|t - t'|}{l} + \frac{5(t - t')^2}{3l^2}\right) \cdot \exp\left(\frac{-\sqrt{5}|t - t'|}{l}\right) \tag{6.3}$$

with lengthscale hyper-parameter $l \in \mathbb{R}^+$ is the appropriate choice.

6.2.2.2 Non-zero Mean Functions for Digital Twins

Within general GP applications, a non-zero mean function might be used to incorporate prior knowledge about the functions to be described. This might be realized by a parametric model of the function if detailed knowledge about the signal characteristics is available. In the given data-driven application, an analytic function can not be given as temperature and CO_2 signal characteristics are determined by non-measured quantities like occupancy within the room or solar radiation. However, it turns out to be sufficient to select a trivial approach that averages the signal values of the training period per time step and is updated by the reference signal. It should be noted that this type of procedure requires equidistant measurements. Thus, for the signal of interest $S_{\mathrm{DT}}(t)$ with a single reference sensor $S_{\mathrm{ref}}(t)$, the prior mean function at time instance t_j reads

$$\mu_{S_{\mathrm{ref}}}^{S_{\mathrm{DT}}}(t_j) = \frac{1}{2}\left(S_j^{\mathrm{offset}} + S_{\mathrm{ref}}(t_j)\right) \tag{6.4}$$

with

$$S_j^{\text{offset}} = \frac{1}{i} \sum_i \left(S_{DT_i}(\hat{t}_j) - S_{\text{ref}_i}(\hat{t}_j) \right) \tag{6.5}$$

where $\hat{t}_j$ runs over discrete, equidistant time instances during the day in the training set and i is indexing training days.

As will be experimentally verified in Sec. 6.5, this construction allows to rely on only a single reference sensor as input for the DT. Thus, (6.4) is stated only for one reference sensor. Contrary to the zero-mean approach, this type of model is trained on the correlation between deviation from the trivial expectation instead of the correlation across signals themselves. This construction enables a long-term application without the need for dedicated retraining strategies. This is due to the fact that external changes, like for example seasonality, are affecting all measurements as well as pseudo-measurements present likewise. Thus, the DT model trained on correlations of deviations is not significantly affected by changing seasons. It should be noted that in the case of a single-input reference signal, (6.1) reduces the ICM

$$K_{\text{ICM}}(t, t') = B^{[2]} \otimes K_{\text{basis}}(t, t') \tag{6.6}$$

6.2.3 Trivial and VARMA Realizations

The alternative models TRIV and VARMA as defined below, require equidistant time steps considering only the latest change of value (COV) measurement per time step.

The TRIV-model provides pseudo-measurements S_{DT} similar to (6.4). For one $M = \{1\}$ or two $M = \{1, 2\}$ reference measurements $S_{\text{ref}_m}, m \in M$ we define

$$\text{TRIV}(t_j) = \frac{1}{M} \sum_{m=1}^{M} \left(S_{mj}^{\text{offset}} + S_{\text{ref}_m}(t_j) \right) \tag{6.7}$$

with

$$S_{mj}^{\text{offset}} = \frac{1}{i} \sum_i \left(S_{DT_i}(\hat{t}_j) - S_{\text{ref}_{mi}}(\hat{t}_j) \right) \tag{6.8}$$

where $\hat{t}_j$ runs over discrete, equidistant time instances during the day in the training set and i is indexing training days.

The VARMA-model for one $S = (S_{\text{ref}_1}, S_{DT})^T$ and two $S = (S_{\text{ref}_1}, S_{\text{ref}_2}, S_{DT})^T$ reference measurements, respectively, reads

$$\text{VARMA} = \nu(t_j) + \text{AR}(p) + \epsilon_j + \text{MA}(q) \tag{6.9}$$

with autoregression $\text{AR}(p) = A_1 S_{j-1} + \cdots + A_p S_{j-p}$, moving average $\text{MA}(q) = M_1 \epsilon_{j-1} + \cdots + M_q \epsilon_{j-q}$, vector-valued Gaussian white noise $\epsilon_j \sim \mathcal{N}(0, \sigma^2)$ and a linear trend $\nu(t_j) = a + bt_j$ for 2 or 3 dimensional square matrices $A_1, \cdots, A_p$, $M_1, \cdots, M_q$ and vectors $a, b \in \mathbb{R}^3$ or $\mathbb{R}^2$, respectively. A least square algorithm is used for training. In the given case study, we have p=q=4. Stationarity is achieved by subtracting the average daily profile (6.8), analogously to the trivial approach.

6.3 Case Study: JuLab

The validity of the given approach is evaluated in a case study based on data collected at a building of the LLEC. For this case study, we select a multi-purpose room that is used as both student laboratory and meeting room, displayed in Fig. 6.4. The room is furnished with multiple workstations and has three glass facades oriented in SE, SW, and NW directions. All three facades can be fully shaded through manual blind operation via a wall-mounted human machine interface (HMI) control panel.

There are three air temperature signals available for the selected room: 1) one permanent measurement integrated into the HMI, which is located next to the main entrance door at a height of 1.60 m providing data on COV with an accuracy of $\pm 0.5\,\mathrm{K}$ and bound to a resolution of 0.1 K, 2) one additional permanent measurement as part of a ceiling-mounted presence multisensor (PM) providing data on COV with an accuracy of $\pm 0.2\,\mathrm{K}$ and bound to a resolution of 0.1 K, and 3) a temporary measurement via an on-desk personal IAQ sensor providing data every 5 min to 15 min (depending on the dynamics of the measured values) with an accuracy of $\pm 0.5\,\mathrm{K}$ and bound to a resolution of 0.2 K.

The temperature measurement from the HMI constitutes a thermal comfort parameter that is quite commonly available in modern buildings nowadays **Althaus2022**. This is due to the use of the room air temperature information to control the heating system based on a temperature set-point provided by the occupants. As the HMI is anyway required to obtain these set points, integrating a temperature measurement represents the industry's best practice. However, the location of the HMI does not allow for a measurement close to the occupants, which leads to a mismatch between the measurable and the occupant-experienced thermal comfort. An increasingly common second sensor at a different location, in this case, the ceiling-mounted PM sensor, can be used to better approximate the real room air temperature. The third temperature signal from the desk-mounted IAQ sensor is the best approximation of the real temperature experienced by a person in the room.

The on-desk IAQ sensor and the PM not only measure the air temperature but also the CO_2 concentration. In addition to these two CO_2 signals, a third one is available through a sensor that is also mounted on a desk multi-sensor (MS) but at a comparably remote position in a presumably better-ventilated part of the room. We choose this spot to achieve a measurement comparable to that of a conventional wall-mounted sensor while still being able to capture the important dynamics inside the occupied area of the room. Regarding the sensors' properties in terms of CO_2, both IAQ and MS have an accuracy of ± 125 ppm. The accuracy of the PM sensor is slightly higher at ± 90 ppm. All three devices have a resolution of 10 ppm and report new CO_2 values with the same sampling rate as for the temperature observations. We summarize sensor specifications in Table 6.1.

Measurements from the four sensors are collected using a cloud-based infrastructure [154], [162]. Each sensor is connected to a programmable logic controller (PLC) via a KNX field-bus network. The IAQ and MS sensors are wireless sensors that use the radio-based EnOcean protocol and connect to the KNX bus through a dedicated, bidirectional EnOcean-to-KNX gateway. The other two sensors (HMI, PM) are directly connected to the KNX bus. The collected device

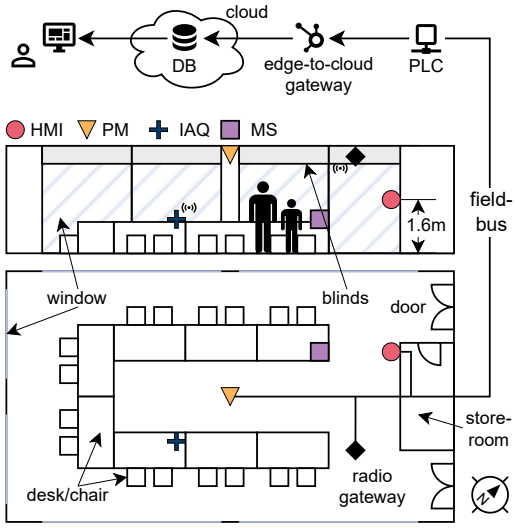

Figure 6.4: Measurement setup for obtaining temperature data at different locations inside the test room.

data is transferred from the PLC into a time-series database. This is done via a data transfer script, which fetches the data from the PLC using the ADS protocol. The fetching mechanism is driven by changes in the data on the PLC to achieve low latencies in data transmission. Finally, the sensor data is retrieved from the database, which is deployed in a cloud environment, using Structured Query Language (SQL) statements.

The data set used in this case study contains five full days of measurements for training, 14 days for the short-term, and seven months for the long-term evaluation, respectively. Temperature and CO_2 data from the four sensors HMI, PM, IAQ, and MS, with varying temporal resolutions ranging from seconds to hours, was obtained during 01 June to 31 December 2023.

6.4 Implementation Framework

The proposed DT framework is implemented in Python. A minimal set of preprocessing steps is applied, where the daytimes of each day are selected from the individual signals, and NaN values are filtered. The proposed Corr-GP is implemented using GPy [97]. For the given application, hyper-parameters specifications are given by GPy's default settings. For the continuous update of the DT during offline operation, signals are retrieved from a database (CrateDB [163]).

The measurements obtained through the various sensors are collected and standardized at a PLC (cf. Table 6.1). Via an edge-to-cloud adapter, based on PyADS [164] and the Eclipse Paho MQTT python client [165], and implemented in Python, we retrieve the standardized data from the PLC and forward it to an Eclipse Mosquitto MQTT broker [166]. From there, the data is written into the database via a script based on the CrateDB python client library [167]. All

Table 6.1: Summary of important specifications of the devices used in the case study.

Device	**Manufacturer**	**Model**	**Protocol**	**Quantities**
IAQ, MS	Pressac Communications	60_CO2_SLR_-TMP_HUM_868	EnOcean[1] (A5-09-04)	Temperature CO_2
HMI	MDT Technologies	BE-GT2TW.02	KNX-TP	Temperature
PM	STEINEL	True Presence Multisensor KNX	KNX-TP	Temperature CO_2
PLC	Beckhoff Automation	CX5130	ADS	n/a
Device	**Accuracy**	**Resolution**	**Range**	**Sampling**
IAQ, MS	±0.5 K ±125 ppm	0.2 K 10 ppm	0–51 °C 0–2555 ppm	5–15 min[2]
HMI	±0.5 K	0.1 K	0–40 °C	on COV
PM	±0.2 K ±90 ppm	0.1 K 10 ppm	0–40 °C 400–10 000 ppm	on COV
PLC	n/a	n/a	n/a	10 ms[3]

[1] The EnOcean Equipment Profile is provided here as it defines not only the payload of the telegram but also the resolution of the transmitted data.
[2] The sampling rate depends on the dynamics of the measured values; the higher the fluctuations, the higher the sampling rate and vice versa.
[3] The sampling rate, here, represents the cycle time of the PLC.

scripts, the MQTT broker and the database are deployed using Docker [168] in our self-hosted cloud environment built on OpenStack [169]. More details on the data collection infrastructure can be found in [162].

To apply the Corr-GP online for live generation of pseudo-measurements, we developed an MQTT interface that allows us to feed the GP directly with live data from the sensors.

6.5 Results

The evaluation of the overall performance and the comparison to alternative algorithms is given with respect to the NRMSE, i.e.,

$$\mathrm{NRMSE}(S_{\mathrm{DT}}, S_{\mathrm{IAQ}}) = \frac{1}{\bar{S}_{\mathrm{IAQ}}} \sqrt{\frac{1}{N} \sum_{j=1}^{N} (S_{\mathrm{IAQ},j} - S_{\mathrm{DT},j})^2} \tag{6.10}$$

with N true measurements S_{IAQ} and, correspondingly, predicted pseudo-measurements S_{DT}. $\bar{S}_{\mathrm{IAQ}}$ denotes the arithmetic signals average for this specific period. It should be noted that (6.10) provides a relative error per day. That is, j runs over N time instances for each individual day. Long-term evaluations are performed per month, where the arithmetic mean over daily relative errors is computed.

6.5.1 Temperature Digital Twins

6.5.1.1 Multi-Input Temperature Digital Twins

The multi-input temperature DT based on Corr-GP$^{\text{IAQ}}_{\text{HMI,PM}}$ is calculated according to Fig. 6.2 and Fig. 6.3 and is defined by (6.1) with $q = 2$ using (6.2), i.e.,

$$\text{DT}^{\text{IAQ}}_{\text{HMI,PM}} = \mathcal{GP}\left(0, \sum_{q=\text{HMI,PM}} B_q^{[3]} \otimes K_{\text{RQ}}(t, t')\right) \tag{6.11}$$

Detailed Short-Term Analysis In Fig. 6.5, the comparison of the Corr-GP$^{\text{IAQ}}_{\text{HMI,PM}}$ (blue) with the TRIV$^{\text{IAQ}}_{\text{HMI,PM}}$ (orange) and VARMA$^{\text{IAQ}}_{\text{HMI,PM}}$ (green) model evaluated by NRMSE [%] per day can be seen. The Corr-GP$^{\text{IAQ}}_{\text{HMI,PM}}$ approach outperforms the given alternatives, for the entire evaluation set except for day 7. This can be explained via a distinct correlation on that specific day, which can be quantified, for example, in terms of the Pearson correlation coefficient [170], given by

$$r_i^{\text{IAQ}} = \frac{\sum_j \left(T_{i_j} - \bar{T}_i\right)\left(T_{\text{IAQ}_j} - \bar{T}_{\text{IAQ}}\right)}{\sqrt{\sum_j (T_{i_j} - \bar{T}_i)^2}\sqrt{\sum_j (T_{\text{IAQ}_j} - \bar{T}_{\text{IAQ}})^2}} \tag{6.12}$$

where $T_i \in \{T_{\text{HMI}}, T_{\text{PM}}\}$, j runs over times instances during a day and bars denote the arithmetic average. The calculated r_i^{IAQ} for the given time-series are displayed in Fig. 6.6-(a). It can be seen that on day seven, a lack of correlation between the IAQ measurements and both the HMI and PM measurements is evident. In line with the given theoretical construction of the Corr-GP in terms of correlated kernel, this result indicates a strong connection between the proposed Corr-GP approach and the level of correlation across the considered signals. Considering Fig. 6.6-(b), this can be explained by a transition in the shading position on day 7 at 10:24 am, in which the previously fully opened blinds are completely closed by the user. Solar radiation that is transmitted into the room heats up the room surfaces and, if hitting a sensor, results in an increased measured temperature. Hence, changes in shading intensity affect the correlation between the temperature sensors. For a prospective field application of Corr-GPs, this relation has to be investigated in detail and corresponding dedicated strategies have to be developed. As an example, the information on the blinds could be included in the DT.

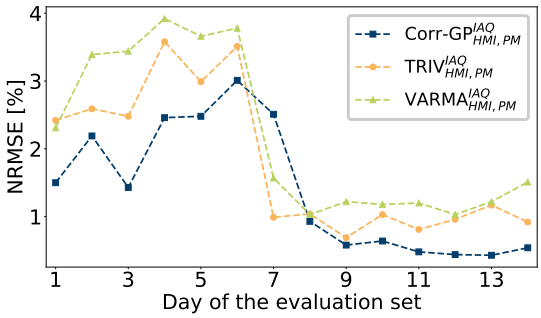

Figure 6.5: Daily NRMSE-based comparison among the considered methods.

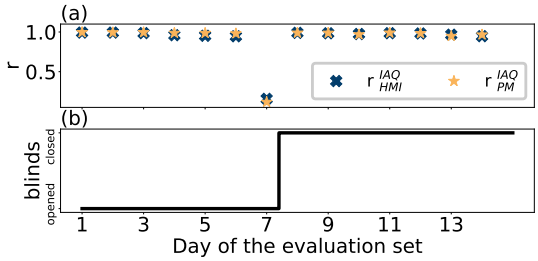

Figure 6.6: (a) Pearson correlation coefficient of IAQ and HMI, PM measurements per day. (b) Blind position.

Long-Term Statistical Analysis As an additional test of the proposed approach, we give a long-term evaluation. The daily-based NRMSEs are aggregated monthly and displayed in Table 6.2. It can be seen that the Corr-GP$^{\text{IAQ}}_{\text{HMI,PM}}$ yields good results for the first months with consecutive increasing errors afterward. This can be explained by seasonality effects and thus, a change in signal correlation. Consequently, for this type of model (6.11), dedicated retraining would be required, i.e., additional measurements have to be performed which limits the usage of the DT model (6.11). It should be noted that although the performance of the trivial approach gets slightly worse over time, the NRMSE does not increase as much as the Corr-GP model. Before presenting a possible solution for the proposed Corr-GP approach in terms of the prior-mean function, we give more details on the individual signal's contribution in the next section.

Model Selection, Limits and Analysis of Correlated Gaussian processes In order to evaluate a possible model extension, we first consider single-input models. In Fig. 6.7, the NRMSE of the continuous single-input models Corr-GP$^{\text{IAQ}}_{\text{PM}}$ and Corr-GP$^{\text{IAQ}}_{\text{HMI}}$ with zero prior mean function, i.e.,

$$\mathcal{GP}\left(0, B_q^{[3]} \otimes K_{RQ}(t,t')\right) \tag{6.13}$$

with $q =$ HMI or PM and TRIV$^{\text{IAQ}}_{\text{PM}}$ and TRIV$^{\text{IAQ}}_{\text{HMI}}$ are displayed. From Fig. 6.7 it can be deduced that the single-input GP architectures perform worse than the single-input trivial algorithm when evaluated for 14 days in June. Contrarily, when having multiple measurement devices available, the results shown in Fig. 6.5 indicate that Corr-GPs are superior with a combination of several reference signals. This is due to the fact that the correlations are modeled as trainable hyper-parameters of the Corr-GP model.

In addition to the NRMSE, intraday performance and individual outliers play a significant role in all follow-up processes based on the pseudo-measurements, e.g., building operation. More

Table 6.2: NRMSE per month [%] of long-term evaluation of multi-input temperature DT.

	Jun	Jul	Aug	Sep	Oct	Nov	Dez
Corr-GP$^{\text{IAQ}}_{\text{HMI,PM}}$	1.3	2.2	3.3	3.8	6.3	7.3	6.4
TRIV$^{\text{IAQ}}_{\text{HMI,PM}}$	2.1	2.2	3.5	4.2	3.3	3.7	3.9
VARMA$^{\text{IAQ}}_{\text{HMI,PM}}$	2.0	2.1	3.5	4.0	3.1	3.5	3.8

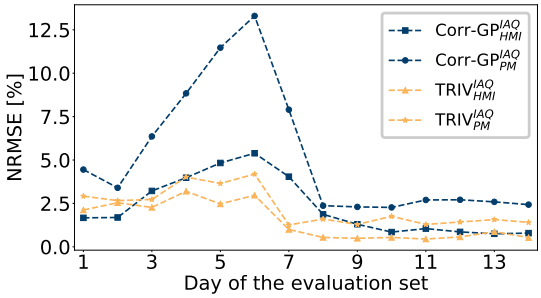

Figure 6.7: NRMSE of single-input zero-mean Corr-GP and single-input .

precisely, the intraday prediction accuracy and the correct estimation of the current system state directly affect the controller's ability to maximize individual thermal comfort and save energy. In Fig. 6.8, the normalized maximum deviation

$$\Delta T_{max} = \frac{1}{\bar{T}_{\mathrm{IAQ}}} \max_{i=1,\dots,M_j} \{|T_{\mathrm{IAQ},i} - T_{DT,i}|\} \tag{6.14}$$

is displayed in addition to the NRMSE. The max-operator determines the maximum deviation between the true T_{IAQ} and the predicted temperature T_{DT} for a single day normalized by the daily average $\bar{T}_{\mathrm{IAQ}}$.

As can be seen in Fig. 6.8, a connection between NRMSE (blue) and ΔT_{max} (orange) is evident. More precisely, we can deduce that mostly if ΔT_{max} rises, so does the NRMSE. In other words, the performance of approach is primarily driven by short-term deviations. An example of this can be seen in Fig. 6.9, where two particular days are depicted.

Fig. 6.9-(a) displays an example with a comparably high NRMSE of 2.46 % and Fig. 6.9-(b) shows a well-performing day with an NRMSE of 0.93%. In Fig. 6.9-(a), it can be seen that the dynamics of the true IAQ measurements (blue crosses) clearly deviate from the two input reference sensors HMI (red) and PM (orange) after 4:30 pm. The local rise in temperature can be explained by an increase in the solar irradiation hitting the IAQ sensor but not the permanent ones. The filled grey area in Fig. 6.9 shows the measured indoor luminosity for the two days, indicating that, (i) high luminosity (i.e., solar irradiation) generally leads to high air temperatures, and, (ii) solely the IAQ sensor reacts to the significant changes in measured luminosity.

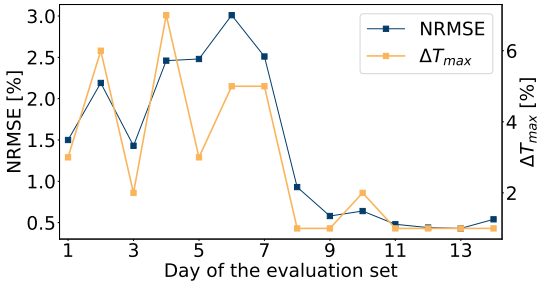

Figure 6.8: Normalized maximum deviation ΔT_{max} and NRMSE.

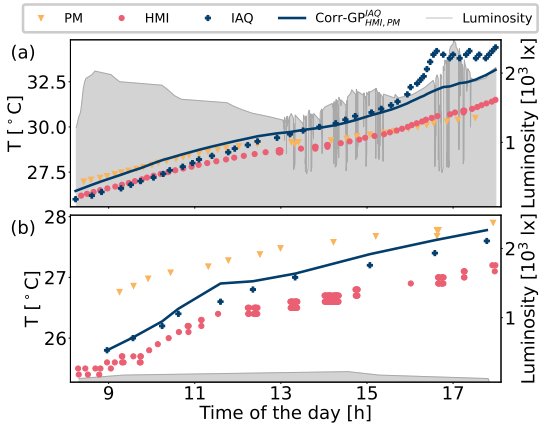

Figure 6.9: Reference measurements, IAQ measurements, the predicted Corr-GP DT and the luminosity for (a) day 4 and (b) day 8 are given.

Due to a low solar elevation angle in the afternoon, the IAQ sensor, which is mounted at a lower height, is hit by incoming solar radiation resulting in an increase in measured temperature. However, the two permanent sensors which are mounted at higher heights are not exposed and, therefore, do not report a temperature increase. As this information is not provided to the DT Corr-GP$^{\text{IAQ}}_{\text{HMI,PM}}$ by the reference signals, it cannot be reproduced by the proposed approach.

6.5.1.2 Single-Input Temperature Digital Twin

Summarizing the discussion in Sec. 6.5.1.1, the Corr-GP approach is superior to the given alternatives regarding the incorporation of multiple inputs but remains sensitive to broken correlations. In the remaining part of this work, we solve the latter issue as periodical retraining would limit the application in practical settings. As the incorporation of additional measurement signals is out of the scope of this work, we leave a dedicated analysis for future work.

The key observation from the previous discussion is the good performance of the trivial alternative. This allowing us to extend the given approach. More precisely, the trivial method TRIV can be included in the Corr-GP in terms of a prior mean function as constructed in (6.4). It should be noted that in the given application it is not possible to give an analytic expression for the prior mean function. Thus, we propose

$$DT^{\text{IAQ}}_{\text{HMI}} = \mathcal{GP}\left(\mu^{\text{IAQ}}_{\text{HMI}}(t), B^{[2]}_{\text{HMI}} \otimes K_{\text{RQ}}(t,t')\right) \tag{6.15}$$

The results for model (6.15) in terms of a long-term evaluation are displayed in Table 6.3. Only results for the model with HMI-input are explicitly given due to better performance compared to the PM sensor as input.

In Table 6.3, it can be seen that the Corr-GP model (6.15) outperforms the alternatives and provides a sufficient prediction capability. It should be noted that the independent accuracy of the Corr-GP model regarding seasonalities is a result of the given analysis as no seasonal pattern can

Table 6.3: NRMSE per month [%] of long-term evaluation of the single-input temperature DT.

	Jun	Jul	Aug	Sep	Oct	Nov	Dez
Corr-GP$^{\text{IAQ}}_{\text{HMI}}$	1.2	1.1	3.0	3.7	3.0	2.4	2.5
TRIV$^{\text{IAQ}}_{\text{HMI}}$	1.7	2.0	3.3	4.2	3.2	3.1	3.1
VARMA$^{\text{IAQ}}_{\text{HMI}}$	1.8	1.8	3.1	4.0	3.1	3.1	3.2

be observed for the results of Corr-GP$^{\text{IAQ}}_{\text{HMI}}$ in Table 6.3. However, as discussed in Sec. 6.2.2.2, this type of independence can be understood by construction. Contrary to the zero-mean approach (6.11) and (6.13), the (6.15) approach is not trained on correlations across signals. Instead, it captures the deviations from an expected mean provided by the trivial approach. As seasonal changes affect all instances simultaneously, DT model (6.15) is also more resistant regarding changing seasonalities.

6.5.2 CO_2 Digital Twins

For the second case study concerning CO_2 pseudo-measurements, both models, i.e., single- and double-input, are tested and evaluated, while the single-input model is individually tested with both possible reference sensors PM and MS. Summarizing the Corr-GP models, we have

$$\text{DT}^{\text{IAQ}}_{\text{MS,PM}} = \mathcal{GP}\left(0, \sum_{q=\text{MS,PM}} B^{[3]}_q \otimes K_{M^{5/2}}(t,t')\right) \tag{6.16}$$

$$\text{DT}^{\text{IAQ}}_{\text{PM}} = \mathcal{GP}\left(\mu^{\text{IAQ}}_{\text{PM}}, B^{[2]}_{\text{PM}} \otimes K_{M^{5/2}}(t,t')\right) \tag{6.17}$$

$$\text{DT}^{\text{IAQ}}_{\text{MS}} = \mathcal{GP}\left(\mu^{\text{IAQ}}_{\text{MS}}, B^{[2]}_{\text{MS}} \otimes K_{M^{5/2}}(t,t')\right) \tag{6.18}$$

The results are provided in Table 6.4. It can be seen that only the Corr-GP$^{\text{IAQ}}_{\text{MS}}$ outperforms the alternative algorithms. As exemplarily shown in Fig. 6.10, this is rooted in the fact that the MS sensor yields measurements sufficiently correlated to the desired IAQ measurements. Contrarily, the PM sensor displays a lower amount of correlation as evident from Fig. 6.10. Thus, both models, including readings from PM perform worse.

6.6 Conclusion and Characteristics of Building Digital Twins

In this chapter, a DT architecture providing temperature and CO_2 pseudo-measurements realized by two types of Corr-GP is proposed. In particular, a multi-input zero-mean as well as a single-input non-zero-mean Corr-GP is proposed and analyzed. It is shown that the proposed Corr-GP digital twin outperforms alternative approaches in predicting a close-to-person temperature and CO_2 pseudo-measurement based on remotely mounted measurement devices.

We conclude this chapter with a comparison of the presented DTs, address limitations, and discuss future research directions. Moreover, we provide a characterization of the developed

Table 6.4: NRMSE per month [%] of CO_2 .

		Jun	Jul	Aug	Sep	Oct	Nov	Dez
PM MS	Corr-$GP^{IAQ}_{MS,PM}$	8.1	6.8	13.7	11.3	12.4	13.1	14.2
	$TRIV^{IAQ}_{MS,PM}$	7.0	7.1	11.5	10.9	10.7	10.4	9.6
	$VARMA^{IAQ}_{MS,PM}$	7.0	6.9	11.2	13.8	11.9	8.3	12.3
PM	Corr-GP^{IAQ}_{PM}	10.1	12.9	22.9	23.7	24.4	24.4	23.1
	$TRIV^{IAQ}_{PM}$	8.8	10.7	21.0	21.9	22.7	22.7	21.5
	$VARMA^{IAQ}_{PM}$	8.3	9.6	20.0	20.9	21.6	21.8	20.6
MS	Corr-GP^{IAQ}_{MS}	6.9	6.3	5.6	5.5	5.2	4.9	4.7
	$TRIV^{IAQ}_{MS}$	7.7	7.6	6.3	6.9	6.1	5.8	5.9
	$VARMA^{IAQ}_{MS}$	7.9	7.7	6.5	7.1	6.3	6.0	6.0

framework of digital representations.

6.6.1 Discussion

Comparison Regarding temperature pseudo-measurements, both approaches, i.e., mean-zero multi-input DT (6.11) and TRIV-mean single-input (6.15), outperform the alternatives subjected to an unchanged seasonality. In case of changing seasonality, the long-term results of Sec. 6.5.1.2 and Sec. 6.5.2 indicate that the TRIV-mean single-input model Corr-GP^{IAQ}_{MS} is applicable without retraining. Thus, long-term pseudo-measurements can be provided by Corr-GP^{IAQ}_{MS} with TRIV-mean function. The comparison of Table 6.2 and Table 6.3 show a possible reduction from two reference sensors to one reference sensor if the GP approach with TRIV-mean function Corr-GP^{IAQ}_{MS} is considered. With application to CO_2 pseudo-measurements, only the TRIV-means single-input with the MS as a reference provides sufficient results.

Concerning the class of vector autoregressive approaches, like VARMA, it should be noted that they require non-seasonal data as they assume an underlying stationary stochastic process.

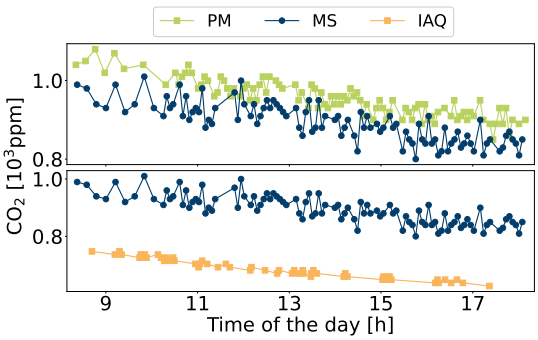

Figure 6.10: Exemplary CO_2 data of a random day (Dec 2).

As this is not the case for the given application, the data must be preprocessed to be usable by VARMA models. In the given case study, a trend-stationarity is not applicable as the test was aborted after 10 evaluation days at most. This is due to a changing seasonal behavior during operation, which is not assignable by the training set. A common method is to consider the difference between neighboring data points. However, this is not applicable as there is no initial point to reverse this transformation to the actual signals. The proposed Corr-GP approaches do not require any preprocessing.

In the given case study, small training data sets are considered. Although more data could be collected in principle, in practical applications, additional measurements should be limited to reduce hardware costs, implementation effort and user disturbances to a minimum. In the presented case study with five training days, the number of available training data points is summarized in Table 6.5. Thus, neural network architectures are not applicable due to the small amount of training data [171].

Limitations A key aspect of the proposed DT framework is the sensitivity to correlations among signals. However, this might also be a limiting fact in practical application. For example, as described in Fig. 6.9, only one of the sensors might be affected by individual effects like shading. Turning the roles of the considered signals around, that is, using the IAQ as a reference sensor predicting HMI or PM in Fig. 6.9 would display a sensitive response to the single-sensor effect. Thus, care must be taken in sensor layout omitting single-sensor effects in reference sensors.

Depending on the type of sensors and signals included, sensor drifting can also not be neglected. As the proposed DT is trained on a specific correlation, retraining can not be avoided in this case. Practically, this implies the installation of the sensor conducting training data, at which the pseudo-measurements should be performed. In general, retraining is always mandatory in cases, in which the correlations between the considered reference measurements and pseudo-measurements change.

In the case study provided, we show that the proposed approach can be updated with respect to new, possibly sparsely sampled COV measurements. However, if the reference sensor fails to provide any data, also the DT can not be updated such that no pseudo-measurements can be provided.

6.6.2 Characterization of Building Digital Twins

A classification of the proposed DT framework is provided in Fig. 6.11. The first thing to specify is the type of digital representation. The explicit data flow between the building and its digital counterpart qualifies the provided model as a DS. Regarding the opposite direction, a more

Table 6.5: Number of available training data points per sensor.

	Temperature			CO_2		
Sensor type	PM	HMI	IAQ	PM	MS	IAQ
# data points	155	462	197	377	543	562

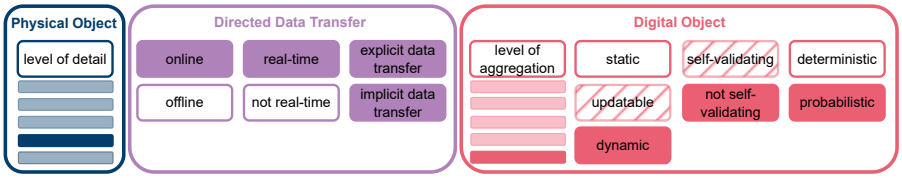

Figure 6.11: Classification of building DTs for pseudo-measurement generation.

refined consideration is needed. Within the application described above, there is no explicit data flow from the digital object to the physical object in terms of a quantitative data stream in the form of bits. However, an implicit data transfer is present. Due to the coupling of the corresponding signals caused by thermodynamics within the room, a change of values in the pseudo-measurement (caused by a change in the reference measurement) is also present in the reference sensor. In this case, we speak of an implicit data transfer. An explicit data transfer in terms of a back-coupling to provide pseudo-measurements influencing a MPC algorithm is investigated in [10].

Compared to the level of detail of a physics-based approach as discussed in Sec. 6.1.3, the given approach lacks physically interpretable parameters. Only based on several temperature or CO_2 measurements, the Corr-GP displays a relatively low level of detail that is captured. The included measurements are represented in a one-to-one correspondence yielding a low level of aggregation. The Corr-GP model is implemented such that it can be operated online. The test run for the given case study was performed offline on a laptop equipped with a 1.60 GHz Intel Core i5-8265U CPU with 24 GB of RAM. The average execution time of the update triggered by the reference sensor is in the range of roughly 10 seconds. Regarding common control schemes such as the MPC described in [10], the proposed algorithm can be operated in real-time at the necessary sampling rate of 15 minutes. Among all sensors, the HMI delivers the most COV measurement values with an average reporting rate of roughly 5 minutes. Even below, including a buffer the proposed DT could be safely operated when updated on a half-minute basis. Regarding the rate of pseudo-measurement generated, no such restriction exists. When not subjected to an update by a reference sensor, the delivery of pseudo-measurements is computationally cheap as no matrix inversion is needed. The provided TRIV-mean single-input models are dynamic in the sense that they adapt to changing seasonalities. For the zero-mean multi-input models, dedicated retraining can not be avoided, thus, they can be classified as updatable. The update would require performing additional measurements, which include periodically installation of sensors at the user-relevant locations. Thus, for these types of models, an application is not practically feasible. In the given case study, the posterior covariance of the Corr-GP that could possibly provide a self-validation has not been mentioned. Similar to the application of power system signal recovery in Ch. 4, the posterior covariance can be utilized as an indicator of the correlation among coupled signal changes during operation. In general, this might be applied to the multi-input DT to monitor the correlations among the two reference sensors. However, this change is simply not present in the given data set of the case study. Thus, we were not able to show this effect within the given case study explicitly. Since it is possible to do so in general, we marked the self-validating class in Fig. 6.11 hatched. For the single-input models, self-validation is not possible as these models are based on a single reference sensor because no relative change to another signal exists. Regarding the change of the posterior covariance of the single reference

signal itself, a deviation from a typical behavior would mean that the inherent time scale of the signal changes dramatically during operation. Considering the order of magnitude required to be observable by the posterior covariance would require a thermodynamically non-realistic scenario.

6.6.3 Outlook

In future works, the demonstrated DTs could be applied to rooms and buildings with different characteristics. In particular, transfer learning could be investigated to reduce costs and implementation of additional measurement devices for training. It is expected that the benefits of the approach are larger in spaces where the distance and height between the remotely mounted sensors and the occupied zone are large, which could be examined as well. The prediction accuracy of the pseudo-measurements was demonstrated to be sensitive to various shading variables, like the position of the blinds and indoor illuminance. Adding such additional information by dedicated measurements could be used to feed the Corr-GP in order to improve the overall performance. The proposed approach is trainable on small data sets which practically benefits a quick application omitting long periods of collecting training data. Nevertheless, including data obtained during long-term application might be beneficial. In particular, the analysis of changes within a particular signal as well as across multiple reference signals might be used and could render retraining unnecessary. The influence on control strategies in terms of thermal comfort or energy saving is investigated in [10].

Beyond the Corr-GP approach, a physics-based approach might be considered at the building level. As the heat flow is usually known for whole buildings, effective/equivalent RC models could be considered to physically model the buildings as a whole based on multi-room-averaged temperatures. Contrarily to single-room characteristics, the physics-informed approach would yield effective models for an equivalent heat capacity and heat transfer parameter, respectively. This might be useful in model reduction of buildings, which might be beneficial to large-scale multi-energy simulation models.

7

Conclusion

In this thesis, we propose the concept of digital representations in energy systems, construct realizations by dedicated GP models, and provide three different applications in energy-related domains. We address power systems data quality, power electronics parameter estimation, and building pseudo-measurement generation. A detailed discussion of each aspect is provided in the corresponding sections of Chs. 2 - 6. In the following, we discuss, address limitations, and propose future research directions from a high-level perspective.

Definition and Classification Scheme of Digital Representations In Ch. 2 we propose a definition and a classification scheme for digital representations within the context of energy systems. The minimal characteristic of the proposed definition (2.II) allows broad applicability while the specific realizations are further specified by a classification scheme. Future possible research routes could include an extension to digital representations of digital objects. This might be desirable to establish an overall well-defined framework applicable to a broader range of applications. Besides that, there is no precise definition of different types of data. As the type of information transferred between both constituents of digital representations is crucial, further clarification would benefit a precise understanding of the proposed concept. On top of these inherent aspects, the proposed definition is developed in the framework of this work, and should thus be evaluated and tested in a broad context. This might identify weak points regarding an extension towards a generalized conception. Moreover, it should be noted that the given classification, as presented in Fig. 2.2 addresses characteristics present in the proposed applications. Other applications and realizations might need further classes to be defined and discussed.

User-Friendly Applications In Ch. 3 we propose a model-building scheme and a dedicated algorithmic solution for digital representations by GPs with multiple outputs. The scheme allows us to address individual peculiarities of specific data sets. Although we have provided sufficient information for the general applicability of the proposed digital representation realizations, a simplified, user-friendly algorithmic solution might be beneficial for an extensive application of the presented models. As of now, the proposed models have to be manually adapted to the individual data sets. A generic solution in terms of, for example, a software package might be desirable. The current GP implementations in Python do not allow for dedicated model-building as described in this thesis. Common implementations in Python are within scikit-learn [172], GPyTorch [173], and GPy [97]. Although the GP implementation in scikit-learn allows the definition of a customized kernel, access to the underlying training procedures is rather limited. GPyTorch is based on PyTorch [174], which demands a tensor structure of the provided data set. As such, the application to the proposed applications, for example, missing data intervals or forward applications, is prohibited by the underlying required data structure. On top, GPyTorch utilizes an approximation for the Cholesky decomposition, which might cause issues regarding the physics-informed models as the original task of computing the kernel's inverse is altered. The Python library GPy, which is also used in this work, allows for a fast and easy implementation of GP with multiple outputs. The drawback of GPy is the brute force handling of ill-conditioned kernel matrices by Tykhonov regularization, which is a disadvantage for physics-informed models and prohibits the application of finer, noise-specific algorithms.

Types of Correlations The DT provided for signal recovery in Ch. 4 and for pseudo-measurement generation in Ch. 6 are both based on Corr-GP models. For both cases, the applicability of the proposed DT is presented with sufficient accuracy. Nevertheless, the individual model building is worth discussing. The models for temperature and CO_2 signals in Ch. 6 are defined via an individual term of the LCM for each signal. For applications in power systems, fewer ICM-terms are needed to define the LCM. This is because of the underlying balanced power system. In voltage magnitudes of power systems, all three phases can be described by the same GP. On top, closely located transformers within the power system might also be described by the same GP. As a possible research direction in this regard, other fields of applications displaying a high share of correlation might be addressed by Corr-GPs.

From Correlations to Physically Informed Systems and Beyond The models provided by Corr-GPs, which are applied in Ch. 4 and Ch. 6 and linear/affine operator kernels applied in Ch. 5, are both utilized to model systems. However, their individual assumptions differ. As explained in (7.1), physics-informed kernels are tight on the coupling and allow looser modeling of the individual signals, while correlated kernels are loose on the coupling and tight on the individual signal modeling. The relation between both models can be seen with the following minimal example. We define a linear operator as $\mathcal{L} = \pm a\tilde{\mathcal{L}}$ with $a \in \mathbb{R}^+$ and $\tilde{\mathcal{L}} = d/dt$ or $\tilde{\mathcal{L}} = Id$. Then, we have the following special cases

$$\begin{pmatrix} K & \pm a\tilde{\mathcal{L}}[K] \\ \pm a\tilde{\mathcal{L}}[K] & a^2\tilde{\mathcal{L}}^2[K] \end{pmatrix} \xrightarrow{\tilde{\mathcal{L}}=Id} \underbrace{\begin{pmatrix} 1 & \pm a \\ \pm a & a^2 \end{pmatrix}}_{A} \otimes K \xleftarrow{b_{11}=1,\dots} \underbrace{\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}}_{B} \otimes K \tag{7.1}$$

The left-hand side of (7.1) is well defined as A is symmetric and $x^T Ax \geq 0 \; \forall x \in \mathbb{R}^2$ holds. In (7.1) it can be seen that both types of models coincide if no differential structure and only a specific coregionalization matrix B is defined. While the Corr-GPs only allows modeling each signal by the same kernel K, PIGPs differ in the sense that they also allow a relative coupling in terms of linear operators, i.e., $\tilde{\mathcal{L}}[K]$. For power-related applications, i.e., power electronics and power systems, this is a beneficial feature due to the differential coupling introduced by passive components. For other types of systems, this might not be a valid assumption. A more general class of individual signal models might be worth investigating in future research activities. This would allow a more application-tailored model-building concerning the grey-box level obtained in these representations. Another aspect to consider is the coupling matrices. While A has a single degree of freedom, B has three degrees of freedom (B symmetric). This flexibility allows the Corr-GP models to determine the system's correlations as demonstrated in Ch. 4 and Ch. 6. However, the matrix B is not unique as an additional scaling parameter could be included, yielding the same model. Consequently, B is not unique, which makes it difficult to relate it to physically interpretable quantities. In future research activities, more general application-tailored coupling structures might be worth investigating.

Decoupled Coupling Equations for Affine Operators Kernels In this work, we develop affine operator kernels based on a single coupling equation. While utilizing a single equation is always possible as the set of equations can be reduced to a single one, in certain circumstances, it might be beneficial to keep individual equations separate. Mathematically this implies a handling of multiple operators and their relative relation on multiple blocks, which might be an interesting future research direction.

Forward and Backward Applications In this work, linear and affine operator kernels were applied to parameter estimation, i.e., backward solutions of differential equations. Alternatively, also forward applications might lead to useful digital representations. In particular, the proposed models in Ch. 3 are constructed independently of the type of applied direction. Thus, the forward application scheme as described in Ch. 6 could also be applied to affine operator PIGPs, yielding a forward digital shadow. The estimated signal could then be fed back to an online control scheme, establishing a digital twin for control purposes. A particular advantage could be that there is no distinction between states and inputs within the digital shadow component, leading to flexible solutions regarding inputs and outputs of the digital twin.

Fields and Purposes of Applications In addition to the given applications, further fields could be explored by the proposed digital representation realizations of which we give some possible examples. In the field of battery modeling, data-driven algorithms could yield promising candidates to provide battery models. The issue with battery systems is that it is not possible to determine the state of the battery by dedicated measurements. The possibility of selecting the digital representation's input and output signals could allow us to provide a digital representation that can be built with the signals available. A further aspect worth mentioning is the posterior covariance with interpretation as provided in this work. Correspondingly, operational changes in systems might be detectable. This might be applied from, for example, the detection of sensor drifts to the monitoring of power systems operation. Within the modeling of buildings, parameter

estimation by the proposed algorithms could yield effective/equivalent models of entire buildings. In practice, it is hard to determine building parameters such as the heat capacity on the building level due to its special configuration. A model reduction in terms of the proposed methods might reduce computational cost, enabling multi-energy system simulation based on a realistic component description. More generally, due to simple key aspects of the proposed GP models, e.g., non-equidistant input data, noise hyper-parameter, and the possibility of incorporating grey-box aspects of physical objects, a more systematic investigation of multi-energy systems could be applied. Especially the incorporation of different timescales into one digital representation is an open question.

Beyond Gaussian Processes Beyond GP, there are many other possible realizations of digital representations yielding useful insights about physical objects. A prominent source of models is within the field of neural networks, which are explored intensively in the literature. More in line with the presented probabilistic machine learning approach, a generalized version of GPs might be worth investigating as a class of models for digital representations. A particular example, generalizing the features obtained by GP, are (generalized) Wishart processes. Contrarily to GP, (generalized) Wishart processes allow for time-varying covariance matrices [175], [176]. This is, in particular, interesting for time-varying parameter estimation as system parameters might turn into functions of time. As such, online monitoring of current system states under changing conditions might be accurately addressed by this type of digital representation.

APPENDIX A

Appendix

A.1 The Inverse of a Block Matrix

Suppose a matrix M can be written in block form

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{A.1}$$

such that the submatrices A and D are invertible. The Schur complement M/A of M with respect to A is defined as

$$M/A = D - CA^{-1}B \tag{A.2}$$

and the Schur complement M/D of M with respect to D is given by

$$M/D = A - BD^{-1}C \tag{A.3}$$

With the theorem on the inverse of a partitioned matrix [24, Ch. 4.3.4], the inverse of M can be written as

$$M^{-1} = \begin{pmatrix} A^{-1} + A^{-1}B(M/A)^{-1}CA^{-1} & -A^{-1}B(M/A)^{-1} \\ -(M/A)^{-1}CA^{-1} & (M/A)^{-1} \end{pmatrix} \tag{A.4}$$

$$= \begin{pmatrix} (M/D)^{-1} & -(M/D)^{-1}BD^{-1} \\ -D^{-1}C(M/D)^{-1} & D^{-1} + D^{-1}C(M/D)^{-1}BD^{-1} \end{pmatrix} \tag{A.5}$$

A proof can be given by direct computation of MM^{-1} and $M^{-1}M$.

A.2 Determinant of a Block Matrix

Suppose M is a symmetric block matrix of the form

$$M = \begin{pmatrix} A & B \\ B^T & D \end{pmatrix} \tag{A.6}$$

If A is invertible, we can write M as

$$\begin{pmatrix} I & 0 \\ B^T A^{-1} & I \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & M/A \end{pmatrix} \begin{pmatrix} I & A^{-1}B \\ 0 & I \end{pmatrix} \tag{A.7}$$

which can be proven by direct computation. Applying the determinant yields

$$|M| = |A| \cdot |M/A| \tag{A.8}$$

Similarly, one can show that

$$|M| = |D| \cdot |M/D| \tag{A.9}$$

A.3 Basics of Probability

We summarize basic definitions of probability from a set-theoretic perspective. We refer to [177, Ch. 1] for references and further details. Let Ω be a sample space, i.e., the space of possible outcomes of an experiment, and $\omega \in \Omega$. We call a subset $A \subset \Omega$ an event. A function $\mathbb{P} : A \to \mathbb{R}$ is called a probability measure if

$$\mathbb{P}(A) \geq 0 \ \forall A \tag{A.10}$$

$$\mathbb{P}(\Omega) = 1 \tag{A.11}$$

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(A_i) \tag{A.12}$$

with $A_1, A_2, \ldots$ disjoint. Let A and B be two events. We write $\mathbb{P}(AB) := \mathbb{P}(A \bigcap B)$. We say that two events are independent if

$$\mathbb{P}(AB) = \mathbb{P}(A)\mathbb{P}(B) \tag{A.13}$$

If $\mathbb{P}(B) > 0$, the conditional probability of A given B is

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(AB)}{\mathbb{P}(B)} \tag{A.14}$$

Let $A_1, \ldots, A_k$ be a partition of Ω. Then, the law for total probability reads

$$\mathbb{P}(B) = \sum_{i=1}^{k} \mathbb{P}(B|A_i)\mathbb{P}(A_i) \tag{A.15}$$

for any B. For a proof, we refer to [177, Thm 1.16]. Let $A_1, \ldots, A_k$ be a partition of Ω such that $\mathbb{P}(A_i) > 0 \; \forall i$. If $\mathbb{P}(B) > 0$ then Bayes' Theorem states that for each $i = 1, \ldots k$,

$$\mathbb{P}(A_i|B) = \frac{\mathbb{P}(B|A_i)\mathbb{P}(A_i)}{\sum_{j=1}^{k} \mathbb{P}(B|A_j)\mathbb{P}(A_j)} \tag{A.16}$$

We call $\mathbb{P}(A_i)$ the prior probability of A_i and $\mathbb{P}(A_i|B)$ the posterior probability of A_i. A proof of Bayes' Theorem can be given by direct computations using conditionals and the law of total probability.

A.4 Basics of Random Variables

We summarize basic definitions of random variables as can be found in [177, Ch. 2]. A random variable is a measurable (see [178] for details) map

$$X : \Omega \to \mathbb{R} \tag{A.17}$$

A random variable X is called discrete if it takes countably finite many values. We define the probability mass function for discrete X as

$$f_X(x) = \mathbb{P}(X = x) \tag{A.18}$$

A random variable X is called continuous if there exists a function f_X such that

$$f_X(x) \geq 0 \; \forall x \tag{A.19}$$

$$\int_{-\infty}^{\infty} f_X(x)dx = 1 \tag{A.20}$$

$$\mathbb{P}(a < X < b) = \int_a^b f_X(x)dx \tag{A.21}$$

for every $a \geq b$. The function f_X is called the probability density function. We give the following definitions for bivariate distributions. The multivariate case holds accordingly. For discrete random variables X and Y the joint probability mass function reads

$$f_{X,Y} = f(x, y) = \mathbb{P}(X = x, Y = y) \tag{A.22}$$

In the continuous case, we have for $f(x, y)$

$$f(x, y) \geq 0 \; \forall (x, y) \tag{A.23}$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)dxdy = 1 \tag{A.24}$$

$$\mathbb{P}((X, Y) \in A) = \iint_A f(x, y)dxdy \tag{A.25}$$

for any $A \in \mathbb{R} \times \mathbb{R}$. For discrete joint distributions of (X, Y) with $f_{X,Y}$ the marginal functions read

$$f_X(x) = \sum_{y \in \Omega} f(x, y) \quad \text{and} \quad f_Y(y) = \sum_{x \in \Omega} f(x, y) \tag{A.26}$$

while in the continuous case we have

$$f_X(x) = \int_{\mathbb{R}} f(x, y)dy \quad \text{and} \quad f_Y(y) = \int_{\mathbb{R}} f(x, y)dx \tag{A.27}$$

for their density analogs. Two random variables are independent if

$$\mathbb{P}(X \in A, Y \in B) = \mathbb{P}(X \in A)\mathbb{P}(Y \in B) \tag{A.28}$$

The conditional probability reads

$$f_{X|Y}(x|y) = \frac{f_{X,Y}}{f_Y(y)} \tag{A.29}$$

if $f_Y(y) > 0$.

A.5 Basics of Expectations, Variance and Covariances

We summarize basic notions of expectations, variance, and covariances definitions as can be found in [177, Ch. 3]. The expectation value, also called the mean, of a random variable X is given by

$$\mathbb{E}(X) = \mu_X = \mu = \begin{cases} \sum\limits_{x \in \Omega} x f_X(x), & \text{if } X \text{ is discrete} \\ \int\limits_{\mathbb{R}} x f_X(x)dx, & \text{if } X \text{ is continuous} \end{cases} \tag{A.30}$$

assuming that the sum or integral exists. For a random variable X with mean μ_X, the variance, denoted by σ^2, is defined as

$$\sigma^2 = \mathbb{E}(X - \mu_X) \tag{A.31}$$

The standard deviation σ is given by $\sqrt{\sigma^2}$. Let X and Y be a random variables with mean μ_X and μ_Y. The covariance Σ between X and Y is given by

$$\Sigma = \Sigma(X, Y) = \mathbb{E}\left((X - \mu_X)(Y - \mu_Y)\right) \tag{A.32}$$

The conditional expectation of X given $Y = y$ is

$$\mathbb{E}(X|Y = y) = \begin{cases} \sum\limits_{x \in \Omega} f_{X|Y}(x|y), & \text{if } X, Y \text{ are discrete} \\ \int\limits_{\mathbb{R}} x f_{X|Y}(x|y)dx, & \text{if } X, Y \text{ are continuous} \end{cases} \tag{A.33}$$

A.6 Gaussian Distribution

A random vector $X = (X_1, \ldots, X_k)$ has a multivariate Gaussian distribution, denoted $X \sim N(\mu, \Sigma)$, if it has the density

$$f(x|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{k}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{A.34}$$

where Σ is a symmetric positive definite matrix, i.e., $x^T\Sigma x > 0, \forall x \neq 0$ [177, Ch. 2].

A.7 Marginals and Conditioning for Jointly Gaussian Distributions

We give details for the computation of marginals and conditionals of Gaussian distributions [24, Ch. 4.3]. Suppose $x = (x_1, x_2)$ is jointly Gaussian with parameters

$$\mu = \begin{pmatrix}\mu_1 \\ \mu_2\end{pmatrix} \text{ and } \Sigma = \begin{pmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{pmatrix} \tag{A.35}$$

Then, the marginals are given by

$$p(x_1) = \mathcal{N}(x_1|\mu_1, \Sigma_{11}) \tag{A.36}$$

$$p(x_2) = \mathcal{N}(x_2|\mu_2, \Sigma_{22}) \tag{A.37}$$

and the conditional is given by

$$p(x1|x2) = \mathcal{N}\left(x_1|\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right) \tag{A.38}$$

We give the following proof. The joint Gaussian $p(x_1, x_2)$ is given with (A.35) by

$$p(x1, x2) = \exp\left(-\frac{1}{2}\begin{pmatrix}x_1 - \mu_1 \\ x_2 - \mu_2\end{pmatrix}^T\begin{pmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}^{-1}\begin{pmatrix}x_1 - \mu_1 \\ x_2 - \mu_2\end{pmatrix}\right) \tag{A.39}$$

Using (A.5) written as

$$\begin{aligned}\Sigma^{-1} &= \begin{pmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}^{-1} \\ &= \begin{pmatrix}(\Sigma/\Sigma_{22})^{-1} & -(\Sigma/\Sigma_{22})^{-1}\Sigma_{12}\Sigma_{21}^{-1} \\ -\Sigma_{22}^{-1}\Sigma_{21}(\Sigma/\Sigma_{22})^{-1} & \Sigma_{22}^{-1} + \Sigma_{22}^{-1}\Sigma_{21}(\Sigma/\Sigma_{22})^{-1}\Sigma_{12}\Sigma_{22}^{-1}\end{pmatrix} \\ &= \begin{pmatrix}1 & 0 \\ -\Sigma_{22}^{-1}\Sigma_{21} & 1\end{pmatrix}\begin{pmatrix}(\Sigma/\Sigma_{22})^{-}1 & 0 \\ 0\Sigma_{22}^{-1}\end{pmatrix}\begin{pmatrix}1 & -\Sigma_{12}\Sigma_{22}^{-1} \\ 0 & 1\end{pmatrix}\end{aligned} \tag{A.40}$$

we can factorize (A.39) as

$$\begin{aligned} p(x1, x2) &= \exp\left(-\frac{1}{2}\left(x_1 - \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)\right)^T (\Sigma/\Sigma_{22})^{-1}\right. \\ &\quad \left.-\frac{1}{2}\left(x_1 - \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)\right)\right) \cdot \exp\left(-\frac{1}{2}(x_2 - \mu_2)^T\Sigma_{22}^{-1}(x_2 - \mu_2)\right) \\ &= p(x_1|x_2)p(x_2) \end{aligned} \tag{A.41}$$

and thus, obtain (A.38). For the normalization constants, we can use (A.9) to verify

$$\begin{aligned} (2\pi)^{\frac{\dim(x1)+\dim(x2)}{2}}|\Sigma|^{\frac{1}{2}} &= (2\pi)^{\frac{\dim(x1)+\dim(x2)}{2}}(|\Sigma/\Sigma_{22}| \cdot |\Sigma_{22}|)^{\frac{1}{2}} \\ &= (2\pi)^{\frac{\dim(x1)}{2}}|\Sigma/\Sigma_{22}|^{\frac{1}{2}}(2\pi)^{\frac{\dim(x2)}{2}}|\Sigma_{22}|^{\frac{1}{2}} \end{aligned} \tag{A.42}$$

Acronyms

ANN Artificial Neural Network 4, 18, 20, 23, 30, 89

ADShooting Automatic Differentiation Shooting 3, 21, 79, 83, 91, 93–95, 107–109

COV Change of Value 117, 118, 120, 127, 128

CS Compressive Sensing 20

Corr-GP Correlated Gaussian Process 3, 4, 52–56, 58, 59, 61, 63–71, 75–77, 113, 114, 119–129, 132, 133, 147, 148

$\mathcal{D}$ Digital object 12–17

DS Digital Shadow 14, 15, 76, 77, 79, 81, 85, 86, 108, 109, 127, 147, 148

DT Digital Twin 14, 15, 55, 58, 77, 108, 109, 111–115, 117, 119, 121, 122, 124–129, 132, 147–149

DC Direct Current 7, 79, 80, 82, 83, 92

FZJ Forschungszentrum Jülich 3, 55, 59, 60, 63, 72, 75, 147, 151

GP Gaussian Process 2, 3, 5, 17, 21–24, 26, 27, 29–44, 46, 47, 49–54, 60, 66, 67, 69, 86, 87, 89, 97, 98, 108, 114, 116, 120, 122, 126, 131, 132, 134, 147

HMI Human Machine Interface 118–125, 127, 128, 148

IAQ Indoor Air Quality 112, 118–127, 148

ICT Information and Communication Technology 3, 4, 55, 58–60, 71, 73, 75–77, 147, 148

IoT Internet-of-Things 1, 59, 112

ICM Intrinsic Coregionalization Model 53, 62, 68–70, 117, 132, 148

LCM Linear Coregionalization Model 53, 68–70, 115, 132, 148

Lin-Int Linear Interpolation 20

LLEC Living Lab Energy Campus 3, 55, 58, 59, 63, 75, 77, 113, 118

LSTM Long Short-Term Memory 20, 81

MPC Model Predictive Control 80, 112, 128

MS Multi Sensor 118–120, 125–127

NLML Negative Log Marginal Likelihood 24, 34, 39, 40, 43

NRMSE Normalized Root Mean Squared Error 65–70, 120–123, 125, 126, 147–149

PMU Phasor Measurement Unit 7, 56

PV Photovoltaic 6, 7

$\mathcal{P}$ Physical object 12–17, 22, 24

PIGP Physics-Informed Gaussian Process 3, 4, 43, 46–48, 79, 81–83, 86, 89, 93, 94, 96, 107, 108, 133

PINN Physics-Informed Neural Network 3, 7, 20, 21, 54, 79–81, 83, 89–91, 93–95, 107, 109

PM Presence Multisensor 118–127, 148

PLC Programmable Logic Controller 118–120

P Proportional control 112

PI Proportional-Integral control 112

PID Proportional-Integral-Derivative control 112

RKHS Reproducing Kernel Hilbert Space 21, 26, 30, 31, 34, 35

S-GP Single-output Gaussian process 21, 26

SQL Structured Query Language 119

SCADA Supervisory Control And Data Acquisition 56

TRIV Trivial Approach 20, 113, 117, 121, 123, 124, 126, 128, 148

VARMA Vector-valued Auto-Regressive Moving Average 20, 54, 113, 117

List of Symbols

$\tilde{\mathcal{I}}$ Approximately known inputs of differential/algebraic equations

$\tilde{\mathcal{E}}$ Approximately known set of equations

$\tilde{\mathcal{S}}_{\mathcal{E}}$ Approximately known solutions of differential/algebraic equations

$\tilde{\phi}$ Approximately known system parameter

$f(t), g(t)$ Arbitrary (continuous) functions

X Arbitrary Set

ϵ Arbitrary small positive number

C Capacity

chol[$\cdot$] Cholesky decomposition of $\cdot$

L_K Cholesky factor of K

$\overline{span}\{\cdot\}$ Closure of the span

$\mathcal{X}$ Compact and finite subset of $\mathbb{R}^n$

D Compact set

$\mathscr{X}$ Compact subset of $\mathbb{R}^n$

$\mathbb{P}(A|B)$ Conditional probability measure

κ Covariance function

$I(t)$ Current

det Determinant

diag Diagonal operator

$\mathcal{F}$ Differential/Algebraic Operator

dim$(\cdot)$ Dimension

$\delta(x)$ Dirac's delta distribution

$\oplus$ Direct sum

δ Duty cycle

$\varnothing$ Empty set

$|_{\{\cdot\}}$ Evaluation map

$\mathbb{E}[\cdot]$ Expectation value

exp Exponential function

$\mathcal{Y}$ Function Observations

Γ Gamma function

$\mathcal{GP}[\cdot]$ Gaussian process representation of $\cdot$

$G = (N, E)$ Graph with nodes N and edges E

C_a Heat capacity of a

$\dot{Q}$ Heat flow

R_{ab} Heat transfer coefficient between a and b

$\mathcal{H}$ Hilbert space

Id Identity map

I Identity matrix

L Inductivity

$\mathcal{I}$ Inputs of differential/algebraic equations

$u(t)$ Inputs, input function

$\sigma_{\%}$ Inverse signal-to-noise ratio

$K(\cdot, \cdot)$ Kernel

$K_{\mathbf{ICM}}$ Kernel of Intrinsic Coregeionalization Model

$K_{\mathbf{LCM}}$ Kernel of Linear Coregeionalization Model

l Lengthscale parameter

$\mathcal{L}_t$ Linear differential operator

$\mathcal{L}$ Linear operator

ln Logarithm

$\mathcal{L}_{\mathbf{ADShooting}}$ Loss function of ADShooting algorithm

$\mathcal{L}_{\mathbf{PINN}}$ Loss function of the PINN

$K_{M^\nu/2}$ Matérn Kernel

A, B, C, D Matrices

$\max_{a \in A}\{\cdot\}$ Maximum of $\cdot$ in A

K_ν Modified Bessel function

N Natural number

G^k Neural network activation function

b_l Neural network offsets

$f_{NN}^{s(t)}$ Neural network representation of $s(t)$

W_l Neural network weights

σ_n Noise hyper-parameter

$\sigma_{\{s(t)\}}$ Noise level of $s(t)$

$n_{\{s(t)\}}$ Normalizer of $s(t)$

$\mathcal{O}(\cdot)$ Order of the set

ν Parameter of Matérn kernel

$a_1, \ldots, b_1, \ldots$ Parameters

r_i Pearson correlation coefficient

Σ^* Posterior covariance function, posterior covariance

μ^* Posterior mean, posterior mean function

f^* Predicted function value at test set x^*

$\mu(\cdot)$ Prior mean function, mean function, mean

$\mathbb{P}$ Probability measure

$K_{\mathbf{RQ}}$ Rational quadratic kernel

$\mathbb{R}^n$ Real numbers in n dimensions

$\Delta\phi$ Relative error of ϕ

$\mathbf{res}(t)$ Residuum/rest of equation

R Resistance

$\langle \cdot, \cdot \rangle_{\mathcal{H}}$ Scalar product in Hilbert space $\mathcal{H}$

K/K_{22} Schur complement of K with respect to K_{22}

$\mathcal{E}$ Set of equations

Θ Set of hyper-parameters

$t \in \mathcal{T}$ Set of time coordinates

s Signal

$A \backslash b$ Solution of $Ax = b$

$\mathcal{S}_{\mathcal{E}}$ Solutions of differential/algebraic equations

$f_{\mathbf{solver}}^{s(t)}$ Solver representation of $s(t)$

$K_{\mathbf{RBF}}$ Squared exponential kernel

$x(t)$ States

$B^{[d]}$ Symmetric positive semi-definite matrix in d dimensions

ϕ System parameter

T_a Temperature of a

$\otimes$ Tensor product

X^* Test set

$T_{\mathbf{period}}$ Time interval for one period

$V(t)$ Voltage

σ Y-scaling of Gaussian process kernels

List of Figures

2.1 Illustration of the concept of digital representations including digital models, DS (directed quantitative data transfer ⇀) and DT (bi-directional quantitative data transfer ⇌). 15
2.2 Classification scheme of digital representations. 18
2.3 Visualization of white-, grey- and black-box modeling with relation to system's description. 19
2.4 Embedding of possible digital representation realizations in the realm of grey-box modeling. 21
2.5 Gaussians in Gaussian Processes: (a) Data (black crosses) and possible functions explaining the data (colored lines), (b) Corresponding probabilities (colored dots) of the functions provided in (a). 23
2.6 Input-output of digital representations. 24
2.7 Variable input-output relations depending on the specific application. 25
2.8 Input (x-axis) and output (y-axis) of the model yielding the digital representations. 26

3.1 Embedding and contextualization of digital representations provided by linear, affine, and correlated kernel GPs. 54

4.1 Simplified scheme of ICT FIWARE-based platform. 60
4.2 Overview of the considered buildings. 63
4.3 Reference signals - Voltages. 64
4.4 Reference signals - Currents. 64
4.5 Case 1-a - True (green) and recovered voltage profiles (blue) of the three-phase transformer B2-Tr1, with Corr-GP. 66
4.6 Case 1-a - Comparison between recovered voltage profiles of phase B of transformer B2-Tr1. 67
4.7 Case 1-b - Comparison of NRMSE of different methods with respect to the length of the missing signal. 68
4.8 Phase A for three hours of available signals of all five transformers. 69

4.9 Posterior variances (left) for transformer B2-Tr3, by a) ICM and b) LCM, with corresponding NRMSE (right). 70
4.10 Case 3 - True (green) and recovered current profiles (blue) for the three phases of transformer B2-Tr2. 71
4.11 Overview of the proposed approach. 72
4.12 Scan module scheme 73
4.13 Recovery module scheme. 74
4.14 Dashboard scheme 76
4.15 DT characterization of the provided ICT framework. 77
4.16 DS characterization of the provided Corr-GP signal recovery algorithm. 77

5.1 LC model of a DC-DC buck converter. 84
5.2 Evaluation of duty cycle determination. (a) $L\frac{d}{dt}I_L(t)$ (numerical derivative), rhs_δ according to (5.4), (b) zoom of panel (a), (c) equation (5.4), (d) zoom of panel (c) 86
5.3 Illustration of PINN approach: Data set (orange crosses), (filtered) states and inputs (orange line), neural network architectures and corresponding fit (dark blue) and training strategy with loss function (5.21a) - (5.21c) (light blue). . . . 90
5.4 Illustration of ADShooting approach: (Filtered) states and inputs (orange) with forward solver (dark blue) and training strategy (light blue). 92
5.5 Relative percentage error of converter parameters. 95
5.6 Converter topology with converter parameters $\phi = L, R_L, C, R_B$ for (a) buck topology and (b) boost topology. 97
5.7 Summary proposed approach: input (dark blue), algorithm workflow (light blue), output (red). 100
5.8 Power electronic testbed for data acquisition. 102
5.9 Estimated capacity $C_{\text{estim}}[\mu F]$ (red crosses) and measured capacity $C_{\text{meas}}[\mu F]$ (blue dots). 106
5.10 Classification of DSs for power electronics parameter estimation. 109

6.1 Different modeling approaches: (a) state-space approach, (b) correlation-based approach. 114
6.2 Proposed DT concept. 115
6.3 Intraday rolling horizon for DT pseudo-measurement. 115
6.4 Measurement setup for obtaining temperature data at different locations inside the test room. 119
6.5 Daily NRMSE-based comparison among the considered methods. 121
6.6 (a) Pearson correlation coefficient of IAQ and HMI, PM measurements per day. (b) Blind position. 122
6.7 NRMSE of single-input zero-mean Corr-GP and single-input 123
6.8 Normalized maximum deviation ΔT_{max} and NRMSE. 123
6.9 Reference measurements, IAQ measurements, the predicted Corr-GP DT and the luminosity for (a) day 4 and (b) day 8 are given. 124
6.10 Exemplary CO_2 data of a random day (Dec 2). 126
6.11 Classification of building DTs for pseudo-measurement generation. 128

List of Tables

4.1 Summary of missing and available data or each test case. 65
4.2 Comparison of different methods. 66
4.3 Case 2 - NRMSE of Corr-GP for B1-Tr1 and B2-Tr3. 69

5.1 Nominal and calculated duty cycle values. 85
5.2 Noise variances σ, absolute average signals $\bar{y}$ and percentage errors by inverse signal-to-noise ratio $\sigma_{\%} = (\bar{y}/\sigma)^{-1}$. 93
5.3 Relative percentage error of converter parameters. 94
5.4 Comparison of results from noisy scenario as in Table 5.3 to results provided in [46]. 95
5.5 Simulation Specifications. 101
5.6 Range of parameter bounds with nominal values. 103
5.7 Computed values for the duty cycle. 103
5.8 Relative errors and runtime for simulation test with buck topology - Case 1. . . 104
5.9 Estimated converter parameter values for hardware test with buck topology - Case 2. 105
5.10 Estimated converter parameter values for hardware test with boost topology - Case 3. 107

6.1 Summary of important specifications of the devices used in the case study. 120
6.2 NRMSE per month [%] of long-term evaluation of multi-input temperature DT. . 122
6.3 NRMSE per month [%] of long-term evaluation of the single-input temperature DT. 125
6.4 NRMSE per month [%] of CO_2 126
6.5 Number of available training data points per sensor. 127

Acknowledgments

This PhD thesis is the scientific outcome of my time at ICE-1 at FZJ. Along the way, many people contributed to the success of this work, and I am thankful for the time we spent together. First, I would like to thank my doctoral advisor, Andrea Benigni, for hiring me, although I was not and will probably never be a true engineer. I appreciate the chance to work in an interdisciplinary environment, and I am thankful for Andrea's supervision, help, and active contribution to my successful time at FZJ. I hope we keep on doing some cool stuff together in the future. I would also like to thank Dirk Witthaut for accepting the call to report on this thesis. I am aware that this is a lot of work, and I am glad you have not refused to be part of my PhD journey. Andrea and Dirk, you are probably the only two people reading this thesis thoroughly. I hope you enjoy it. The outcome in the form of publications listed at the beginning of this thesis would not have been possible without many contributions from very nice and fruitful collaborations. I would like to thank Florian Redder, Maximilian Mork, Max Büchel, Joshua Augustin, Daniele Carta, Edoardo De Din, Moritz Wenzel, André Xhonneux, Matthew Milton, Dejana Cujak, and Herbert Ginn for the effort you put into our work. If some of you want to collaborate again, I am in. Besides collaborations with active scientific output, I would like to thank all my colleagues at ICE-1 for the nice and helpful discussions in the hallway, on Skype and Teams, or at the coffee machine. In particular, I would like to thank Yifei Lu, Junjie Zhang, Jan Stock, Eike Cramer, Danimir Doncevic, Mehmet Velioglu, and Daniel Mayfrank for the nice discussions. Furthermore, I would like to thank Philipp Glücker and Sina Dibos for their help, empathy, and friendship during my time at FZJ. Honestly, without you, the time at FZJ would have been much harder. Many thanks go to Laura Köhler and Stephanie Matern for reviewing this thesis. I can imagine how hard it is to read the actual text and not only look at the figures' captions as physicists. I really appreciate the work you put in. Nevertheless, let's not do this again and just keep on spending our vacations together.

Bibliography

[1] **M. Zimmer**, M. Milton, D. Cucak, T. Pesch, H. Ginn, and A. Benigni, "Parameter estimation of power electronic converter by affine physics-informed gaussian processes", *IEEE Transactions on Power Electronics*, pp. 1–13, 2026.

[2] **M. Zimmer** et al., "Digital twins for building pseudo-measurements", *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–10, 2025. DOI: 10.1109/TIM.2025.3527590

[3] **M. Zimmer**, D. Carta, T. Pesch, and A. Benigni, "Signal recovery in power systems by correlated gaussian processes", *IEEE Open Journal of the Industrial Electronics Society*, vol. 5, pp. 692–703, 2024. DOI: 10.1109/OJIES.2024.3423405

[4] M. Wenzel, E. De Din, **M. Zimmer**, and A. Benigni, "Gaussian process supported stochastic mpc for distribution grids", *IEEE Open Journal of Control Systems*, vol. 4, pp. 332–348, 2025. DOI: 10.1109/OJCSYS.2025.3601836

[5] **M. Zimmer**, E. De Din, D. Carta, and A. Benigni, "Power electronics parameter estimation by physics-informed gaussian processes", in *IECON 2025 – 51st Annual Conference of the IEEE Industrial Electronics Society*, 2025, pp. 1–7.

[6] **M. Zimmer**, J. Augustin, T. Pesch, and A. Benigni, "Parameter estimation of power electronics by forward and backward solutions", in *IECON 2024 - 50th Annual Conference of the IEEE Industrial Electronics Society*, 2024, pp. 1–7. DOI: 10.1109/IECON55916.2024.10905957

[7] **M. Zimmer** et al., "Building digital twins for thermal pseudo-measurements generation", in *2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2024, pp. 1–6. DOI: 10.1109/I2MTC60896.2024.10561057

[8] D. Carta, **M. Zimmer**, T. Pesch, and A. Benigni, "Bad Data Detection and Handling in ICT Platforms for Energy Systems", in *2022 IEEE 12th International Workshop on Applied Measurements for Power Systems (AMPS)*, Sep. 2022, pp. 1–6. DOI: 10.1109/AMPS55790.2022.9978880

[9] **M. Zimmer**, T. Pesch, and A. Benigni, "Time-Series Analysis and Forecasting of Power Consumption using Gaussian Process Regression", in *2021 International Conference on Smart Energy Systems and Technologies (SEST)*, Sep. 2021, pp. 1–6. DOI: 10.1109/SEST50973.2021.9543253

[10] M. Mork et al., "Thermal pseudo-measurements for model predictive control in buildings", preprint.

[11] M. Buechel, "Building digital twins for energy savings by correlated gaussian processes", Technische Universität Dortmund, M.S. thesis, Technische Universität Dortmund, 2024.

[12] J. C. Augustin, "Improved machine learning methods for parameter estimation of ordinary differential equations", Technische Universität Berlin, M.S. thesis, 2023.

[13] R. Owens, "Rolling horizon forecasting of time series energy demand using long short-term memory neural networks", Hochschule Darmstadt, M.S. thesis, Hochschule Darmstadt, 2022.

[14] K. Stein, "Untersuchung verschiedener lstm-netzwerkarchitekturen zur anomalieerkennung in energieverbrauchsdaten", Hochschule Emden Leer, M.S. thesis, Hochschule Emden Leer, 2022.

[15] "IEEE electrification", *IEEE Electrification Magazine*, vol. 12, no. 3, pp. C2–C2, Sep. 2024, ISSN: 2325-5897, 2325-5889. DOI: 10.1109/MELE.2024.3442328

[16] M. S. Irfan, S. Dasgupta, and M. Rahman, "Toward transportation digital twin systems for traffic safety and mobility: A review", *IEEE Internet of Things Journal*, vol. 11, no. 14, pp. 24 581–24 603, Jul. 2024, ISSN: 2327-4662. DOI: 10.1109/JIOT.2024.3395186

[17] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research", *IEEE Access*, vol. 8, pp. 108 952–108 971, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2998358

[18] M. W. Grieves, "PLM–beyond lean manufacturing", *Manufacturing Engineering*, vol. 130, no. 3, p. 23, 2003, ISSN: 0361-0853.

[19] F. E. Cellier, *Continuous System Modeling*. New York, NY: Springer, 1991, ISBN: 978-1-4757-3924-4 978-1-4757-3922-0.

[20] Y. Xu, Y. Sun, X. Liu, and Y. Zheng, "A digital-twin-assisted fault diagnosis using deep transfer learning", *IEEE Access*, vol. 7, pp. 19 990–19 999, 2019, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2890566

[21] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison", *IEEE Access*, vol. 6, pp. 3585–3593, 2018, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2793265

[22] C. Brosinsky, D. Westermann, and R. Krebs, "Recent and prospective developments in power system control centers: Adapting the digital twin technology for application in power system control centers", in *2018 IEEE International Energy Conference (ENERGYCON)*, Jun. 2018, pp. 1–6.

[23] Q. Qi et al., "Enabling technologies and tools for digital twin", *Journal of Manufacturing Systems*, Digital Twin towards Smart Manufacturing and Industry 4.0, vol. 58, pp. 3–21, Jan. 1, 2021, ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2019.10.001

[24] K. P. Murphy, *Machine learning: a probabilistic perspective* (Adaptive computation and machine learning series). Cambridge, MA: MIT Press, 2012, 1067 pp., ISBN: 978-0-262-01802-9.

[25] N. Aronszajn, "Theory of reproducing kernels", *Trans. Amer. Math. Soc.*, vol. 68, pp. 337–404, 1950.

[26] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning* (Adaptive computation and machine learning). Cambridge, Mass: MIT Press, 2006, 248 pp., ISBN: 978-0-262-18253-9.

[27] N. J. Higham, *Accuracy and stability of numerical algorithms*, 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2002, 680 pp., ISBN: 978-0-89871-521-7.

[28] X. Xu, Y. Lei, and Z. Li, "An incorrect data detection method for big data cleaning of machinery condition monitoring", *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 2326–2336, 2020. DOI: 10.1109/TIE.2019.2903774

[29] Y. Wang, H. Ma, X. Xiao, Y. Wang, Y. Zhang, and H. Wang, "Harmonic State Estimation for Distribution Networks Based on Multi-Measurement Data", *IEEE Transactions on Power Delivery*, pp. 2311–2325, 2023. DOI: 10.1109/TPWRD.2023.3239479.

[30] J. Ling, K. Feng, T. Wang, M. Liao, C. Yang, and Z. Liu, "Data Modeling Techniques for Pipeline Integrity Assessment: A State-of-the-Art Survey", *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–17, 2023, ISSN: 1557-9662. DOI: 10.1109/TIM.2023.3279910

[31] Y. Liu et al., "Robust event classification using imperfect real-world PMU data", *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 7429–7438, 2022.

[32] J. Miao, Y. Liu, Q. Yin, B. Ju, G. Zhang, and H. Wang, "A novel soft fault detection and diagnosis method for a dc/dc buck converter based on contrastive learning", *IEEE Transactions on Power Electronics*, vol. 39, no. 1, pp. 1501–1513, 2024. DOI: 10.1109/TPEL.2023.3320878

[33] S. Zhao and H. Wang, "Enabling Data-Driven Condition Monitoring of Power Electronic Systems With Artificial Intelligence: Concepts, Tools, and Developments", *IEEE Power Electronics Magazine*, vol. 8, no. 1, pp. 18–27, Mar. 2021, ISSN: 2329-9207. DOI: 10.1109/MPEL.2020.3047718

[34] H. Gietler et al., "On-Time Mismatch Based System Identification Technique for Buck Converters", *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 7898–7908, Sep. 2020, ISSN: 1557-9948. DOI: 10.1109/TIE.2019.2939982

[35] H. Zhang, Y. Li, R. Xie, J. Song, B. Liang, and Y. Huangfu, "Adaptive model predictive control of an interleaved boost converter using real-time updated model", *IEEE Transactions on Power Electronics*, vol. 38, no. 2, pp. 1720–1731, 2023, ISSN: 1941-0107. DOI: 10.1109/TPEL.2022.3216600

[36] L. Yang, H. Yan, and J. C. Lam, "Thermal comfort and building energy consumption implications – a review", *Applied Energy*, vol. 115, pp. 164–173, 2014, ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2013.10.062

[37] Y. Geng, W. Ji, B. Lin, and Y. Zhu, "The impact of thermal environment on occupant ieq perception and productivity", *Building and Environment*, vol. 121, pp. 158–167, 2017, ISSN: 0360-1323. DOI: https://doi.org/10.1016/j.buildenv.2017.05.022

[38] Y. Feng, S. Liu, J. Wang, J. Yang, Y.-L. Jao, and N. Wang, "Data-driven personal thermal comfort prediction: A literature review", *Renewable and Sustainable Energy Reviews*, vol. 161, p. 112 357, 2022, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2022.112357

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[40] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and u.s. air force vehicles", Honolulu, Hawaii: American Institute of Aeronautics and Astronautics, Apr. 23, 2012. DOI: 10.2514/6.2012-1818

[41] H. Chen, Z. Zhang, P. Karamanakos, and J. Rodriguez, "Digital twin techniques for power electronics-based energy conversion systems: A survey of concepts, application scenarios, future challenges, and trends", *IEEE Industrial Electronics Magazine*, vol. 17, no. 2, pp. 20–36, Jun. 2023, ISSN: 1941-0115. DOI: 10.1109/MIE.2022.3216719

[42] M. Milton, C. De La O, H. L. Ginn, and A. Benigni, "Controller-embeddable probabilistic real-time digital twins for power electronic converter diagnostics", *IEEE Transactions on Power Electronics*, vol. 35, no. 9, pp. 9850–9864, 2020. DOI: 10.1109/TPEL.2020.2971775

[43] K. Sivalingam, M. Sepulveda, M. Spring, and P. Davies, "A review and methodology development for remaining useful life prediction of offshore fixed and floating wind turbine power converter with digital twin technology perspective", in *2018 2nd International Conference on Green Energy and Applications (ICGEA)*, Mar. 2018, pp. 197–204. DOI: 10.1109/ICGEA.2018.8356292

[44] P. Jain, J. Poon, J. P. Singh, C. Spanos, S. R. Sanders, and S. K. Panda, "A digital twin approach for fault diagnosis in distributed photovoltaic systems", *IEEE Transactions on Power Electronics*, vol. 35, no. 1, pp. 940–956, Jan. 2020, ISSN: 0885-8993, 1941-0107. DOI: 10.1109/TPEL.2019.2911594

[45] Y. Peng, S. Zhao, and H. Wang, "A digital twin based estimation method for health indicators of DC–DC converters", *IEEE Transactions on Power Electronics*, vol. 36, no. 2, pp. 2105–2118, Feb. 2021, ISSN: 1941-0107. DOI: 10.1109/TPEL.2020.3009600

[46] S. Zhao, Y. Peng, Y. Zhang, and H. Wang, "Parameter Estimation of Power Electronic Converters With Physics-Informed Machine Learning", *IEEE Transactions on Power Electronics*, vol. 37, no. 10, pp. 11 567–11 578, Oct. 2022, ISSN: 1941-0107. DOI: 10.1109/TPEL.2022.3176468

[47] M. Fahim, V. Sharma, T.-V. Cao, B. Canberk, and T. Q. Duong, "Machine learning-based digital twin for predictive modeling in wind turbines", *IEEE Access*, vol. 10, pp. 14 184–14 194, 2022, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3147602

[48] J. Zhang and X. Zhao, "Digital twin of wind farms via physics-informed deep learning", *Energy Conversion and Management*, vol. 293, p. 117 507, Oct. 1, 2023, ISSN: 0196-8904. DOI: 10.1016/j.enconman.2023.117507

[49] J. Xia and G. Zou, "Operation and maintenance optimization of offshore wind farms based on digital twin: A review", *Ocean Engineering*, vol. 268, p. 113 322, Jan. 15, 2023, ISSN: 0029-8018. DOI: 10.1016/j.oceaneng.2022.113322

[50] V. Bassi, L. F. Ochoa, T. Alpcan, and C. Leckie, "Electrical model-free voltage calculations using neural networks and smart meter data", *IEEE Transactions on Smart Grid*, vol. 14, no. 4, pp. 3271–3282, Jul. 2023, ISSN: 1949-3061. DOI: 10.1109/TSG.2022.3227602

[51] D. Lagos, N. Hatziargyriou, A. Dimeas, G. Sideratos, T. Anastopoulou, and T. Patsaka, "Digital twin for the evaluation of data-driven-based dynamic security assessment of non-interconnected islands: Introducing digital twin on rhodes island", *IEEE Electrification Magazine*, vol. 12, no. 3, pp. 21–30, Sep. 2024, ISSN: 2325-5889. DOI: 10.1109/MELE.2024.3423088

[52] H. Pargmann, D. Euhausen, and R. Faber, "Intelligent big data processing for wind farm monitoring and analysis based on cloud-technologies and digital twins: A quantitative approach", in *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, Apr. 2018, pp. 233–237. DOI: 10.1109/ICCCBDA.2018.8386518

[53] P. Arboleya and A. Méndez, "Real-time grid digital twins: The backbone of the next generation of network technology for distribution system operators", *IEEE Electrification Magazine*, vol. 12, no. 3, pp. 39–49, Sep. 2024, ISSN: 2325-5889. DOI: 10.1109/MELE.2024.3423110

[54] P. Palensky, P. Mancarella, T. Hardy, and M. Cvetkovic, "Cosimulating integrated energy systems with heterogeneous digital twins: Matching a connected world", *IEEE Power and Energy Magazine*, vol. 22, no. 1, pp. 52–60, Jan. 2024, ISSN: 1558-4216. DOI: 10.1109/MPE.2023.3324886

[55] A. Monti, B. Dolley, P. Palensky, and A. Parisot, "Building a digital twin for european energy infrastructure: The role of open source", *IEEE Electrification Magazine*, vol. 12, no. 3, pp. 78–84, Sep. 2024, ISSN: 2325-5889. DOI: 10.1109/MELE.2024.3423168

[56] Y. Sun, Y. Shi, Q. Hu, C. Xie, and T. Su, "DTformer: An efficient digital twin model for loss measurement in UHVDC transmission systems", *IEEE Transactions on Power Systems*, vol. 39, no. 2, pp. 3548–3559, Mar. 2024, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2023.3278300

[57] H. Zhao, Z. Zhang, Y. Yang, J. Xiao, and J. Chen, "A Dynamic Monitoring Method of Temperature Distribution for Cable Joints Based on Thermal Knowledge and Conditional Generative Adversarial Network", *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–14, 2023, ISSN: 1557-9662. DOI: 10.1109/TIM.2023.3317485

[58] H. Brandtstaedter, C. Ludwig, L. Hübner, E. Tsouchnika, A. Jungiewicz, and U. Wever, "DIGITAL TWINS FOR LARGE ELECTRIC DRIVE TRAINS", in *2018 Petroleum and Chemical Industry Conference Europe (PCIC Europe)*, Jun. 2018, pp. 1–5. DOI: 10.23919/PCICEurope.2018.8491413

[59] K. Kušić, R. Schumann, and E. Ivanjko, "A digital twin in transportation: Real-time synergy of traffic data streams and simulation for virtualizing motorway dynamics", *Advanced Engineering Informatics*, vol. 55, p. 101 858, Jan. 1, 2023, ISSN: 1474-0346. DOI: 10.1016/j.aei.2022.101858

[60] F. Rodrigues, K. Henrickson, and F. C. Pereira, "Multi-output gaussian processes for crowdsourced traffic data imputation", *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 594–603, Feb. 2019, ISSN: 1558-0016. DOI: 10.1109/TITS.2018.2817879

[61] X. Chen, E. Kang, S. Shiraishi, V. M. Preciado, and Z. Jiang, "Digital behavioral twins for safe connected cars", in *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, ser. MODELS '18, New York, NY, USA: Association for Computing Machinery, Oct. 14, 2018, pp. 144–153, ISBN: 978-1-4503-4949-9. DOI: 10.1145/3239372.3239401

[62] B. Ketzler, V. Naserentin, F. Latino, C. Zangelidis, L. Thuvander, and A. Logg, "Digital twins for cities: A state of the art review", *Built Environment*, vol. 46, no. 4, pp. 547–573, Dec. 1, 2020. DOI: 10.2148/benv.46.4.547

[63] H. Wang, X. Chen, F. Jia, and X. Cheng, "Digital twin-supported smart city: Status, challenges and future research directions", *Expert Systems with Applications*, vol. 217, p. 119 531, May 1, 2023, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2023.119531

[64] T. Deng, K. Zhang, and Z.-J. (Shen, "A systematic review of a digital twin city: A new pattern of urban governance toward smart cities", *Journal of Management Science and Engineering*, vol. 6, no. 2, pp. 125–134, Jun. 1, 2021, ISSN: 2096-2320. DOI: 10.1016/j.jmse.2021.03.003

[65] N. Mohammadi and J. E. Taylor, "Smart city digital twins", in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov. 2017, pp. 1–5. DOI: 10.1109/SSCI.2017.8285439

[66] M. Yan, W. Gan, Y. Zhou, J. Wen, and W. Yao, "Projection method for blockchain-enabled non-iterative decentralized management in integrated natural gas-electric systems and its application in digital twin modelling", *Applied Energy*, vol. 311, p. 118 645, Apr. 1, 2022, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2022.118645

[67] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey", *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, Sep. 2021, ISSN: 2327-4662. DOI: 10.1109/JIOT.2021.3079510

[68] M. Dalibor et al., "A cross-domain systematic mapping study on software engineering for digital twins", *Journal of Systems and Software*, vol. 193, p. 111 361, Nov. 1, 2022, ISSN: 0164-1212. DOI: 10.1016/j.jss.2022.111361

[69] Y. Chen, "Integrated and intelligent manufacturing: Perspectives and enablers", *Engineering*, vol. 3, no. 5, pp. 588–595, Oct. 1, 2017, ISSN: 2095-8099. DOI: 10.1016/J.ENG.2017.04.009

[70] M. Kritzler, J. Hodges, D. Yu, K. Garcia, H. Shukla, and F. Michahelles, "Digital companion for industry", in *Companion Proceedings of The 2019 World Wide Web Conference*, ser. WWW '19, New York, NY, USA: Association for Computing Machinery, May 13, 2019, pp. 663–667, ISBN: 978-1-4503-6675-5. DOI: 10.1145/3308560.3316510

[71] Y. Zheng, S. Yang, and H. Cheng, "An application framework of digital twin and its case study", *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 3, pp. 1141–1153, Mar. 1, 2019, ISSN: 1868-5145. DOI: 10.1007/s12652-018-0911-3

[72] A. Aziz, S. S. Chouhan, O. Schelén, and U. Bodin, "Distributed digital twins as proxies-unlocking composability and flexibility for purpose-oriented digital twins", *IEEE Access*, vol. 11, pp. 137 577–137 593, 2023, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3340132

[73] A. Bilberg and A. A. Malik, "Digital twin driven human–robot collaborative assembly", *CIRP Annals*, vol. 68, no. 1, pp. 499–502, Jan. 1, 2019, ISSN: 0007-8506. DOI: 10.1016/j.cirp.2019.04.011

[74] V. Maillard, X. Wu, and U. Muenz, "Digital twin: An asset to address sustainability challenges", *IEEE Electrification Magazine*, vol. 12, no. 3, pp. 31–38, Sep. 2024, ISSN: 2325-5889. DOI: 10.1109/MELE.2024.3423109

[75] P. Chongfuangprinya, Y. Ye, C. Venkatraman, P. Mital, H. Osaki, and T. Kato, "Unlocking the potential of digital twins for a greener grid: Applications for the whole range of the electric power system", *IEEE Electrification Magazine*, vol. 12, no. 3, pp. 10–20, Sep. 2024, ISSN: 2325-5889. DOI: 10.1109/MELE.2024.3423070

[76] R. Vrabič, J. A. Erkoyuncu, P. Butala, and R. Roy, "Digital twins: Understanding the added value of integrated models for through-life engineering services", *Procedia Manufacturing*, Proceedings of the 7th International Conference on Through-life Engineering Services, vol. 16, pp. 139–146, Jan. 1, 2018, ISSN: 2351-9789. DOI: 10.1016/j.promfg.2018.10.167

[77] B. R. Gaines, "Progress in general systems research", in *Applied General Systems Research: Recent Developments and Trends*, G. J. Klir, Ed., Boston, MA: Springer US, 1978, pp. 3–28, ISBN: 978-1-4757-0555-3. DOI: 10.1007/978-1-4757-0555-3_1

[78] W. R. Ashby, *An Introduction to Cybernetics*. Martino Fine Books, Jan. 15, 2015, 306 pp., ISBN: 978-1-61427-765-1.

[79] M. Minsky, "Matter, mind and models", *Proceedings of IFIP Congress*, vol. 1, pp. 45–49, 1965.

[80] B. Kuipers, *Qualitative Reasoning: Modeling and Simulation With Incomplete Knowledge*. Cambridge, Mass London: MIT Press, Aug. 29, 1994, 450 pp., ISBN: 978-0-262-11190-4.

[81] R. Hill and S. Berry, *Guide to Industrial Analytics: Solving Data Science Problems for Manufacturing and the Internet of Things* (Texts in Computer Science). Cham: Springer International Publishing, 2021, ISBN: 978-3-030-79103-2 978-3-030-79104-9. DOI: 10.1007/978-3-030-79104-9

[82] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[83] S. D. Conte and C. de Boor, *Elementary Numerical Analysis* (Classics in Applied Mathematics). Society for Industrial and Applied Mathematics, Dec. 27, 2017, 475 pp., ISBN: 978-1-61197-519-2. DOI: 10.1137/1.9781611975208

[84] D. Carta and A. Benigni, "Performance evaluation of a missing data recovery approach based on compressive sensing", in *2021 IEEE 11th International Workshop on Applied Measurements for Power Systems (AMPS)*, 2021, pp. 1–6. DOI: 10.1109/AMPS50177.2021.9586042

[85] S. Jain, Sanjay, Pawan, R. Arora, and P. Behera, "Wind Power Forecasting using VARMAX", in *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, Oct. 2022, pp. 353–360. DOI: 10.1109/ICTACS56270.2022.9988181

[86] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", *Journal of Computational Physics*, vol. 378, no. C, Nov. 2018, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.10.045

[87] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Boston, MA: Springer US, 2004, ISBN: 978-1-4613-4792-7 978-1-4419-9096-9. DOI: 10.1007/978-1-4419-9096-9

[88] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels", *The Journal of Machine Learning Research*, vol. 7, pp. 2651–2667, Dec. 1, 2006, ISSN: 1532-4435.

[89] R. Y. Rubinstein, *Simulation and the Monte Carlo Method*. Newark, UNITED STATES: John Wiley & Sons, Incorporated, 2016, ISBN: 978-1-118-63220-8.

[90] T. Liggett, *Continuous Time Markov Processes: An Introduction* (Continuous Time Markov Processes). American Mathematical Society, 2010, ISBN: 978-0-8218-4949-1.

[91] "Kolmogorov extension theorem", in *Stochastic Processes*, ser. Cambridge Series in Statistical and Probabilistic Mathematics, R. F. Bass, Ed., Cambridge: Cambridge University Press, 2011, pp. 382–384, ISBN: 978-1-107-00800-7. DOI: 10.1017/CBO9780511997044.048

[92] R. J. Adler, *The geometry of random fields* (Classics in applied mathematics 62). Philadelphia, Pa: SIAM, Soc. for Industrial and Applied Mathematics, 2010, 280 pp., ISBN: 978-0-89871-693-1.

[93] B. K. Sriperumbudur, K. Fukumizu, and G. R. G. Lanckriet, "Universality, characteristic kernels and RKHS embedding of measures", *Journal of Machine Learning Research*, vol. 12, no. 70, pp. 2389–2410, 2011, ISSN: 1533-7928.

[94] "Injective hilbert space embeddings of probability measures", English (US), 21st Annual Conference on Learning Theory, COLT 2008, 2008, pp. 111–122.

[95] S. Basak, S. Petit, J. Bect, and E. Vazquez, "Numerical issues in maximum likelihood parameter estimation for gaussian process interpolation", in *Machine Learning, Optimization, and Data Science*, Cham: Springer International Publishing, 2022, pp. 116–131, ISBN: 978-3-030-95470-3. DOI: 10.1007/978-3-030-95470-3_9

[96] D. Gerth, "A new interpretation of (tikhonov) regularization", *Inverse Problems*, vol. 37, no. 6, p. 064 002, Jun. 1, 2021, ISSN: 0266-5611, 1361-6420. DOI: 10.1088/1361-6420/abfb4d arXiv: 2103.08218[math].

[97] GPy, *GPy: A gaussian process framework in python*, http://github.com/SheffieldML/GPy, since 2012.

[98] S. J. W. Jorge Nocedal, *Numerical Optimization* (Springer Series in Operations Research and Financial Engineering). Springer New York, 2006, ISBN: 978-0-387-30303-1. DOI: 10.1007/978-0-387-40065-5

[99] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Machine learning of linear differential equations using Gaussian processes", *Journal of Computational Physics*, vol. 348, pp. 683–693, Nov. 2017, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2017.07.050

[100] P. Rajan and A. Sekar, "Ch.1 - linear circuit analysis", in *The Electrical Engineering Handbook*, W.-K. CHEN, Ed., Burlington: Academic Press, 2005, pp. 3–29, ISBN: 978-0-12-170960-0.

[101] E. M. Purcell and D. J. Morin, *Electricity and Magnetism*, 3rd ed. Cambridge University Press, 2013.

[102] M. P.-L. Ooi, A. Rajan, Y. C. Kuang, and S. Demidenko, *Analytical Evaluation of Uncertainty Propagation for Probabilistic Design Optimisation:* IOP Publishing, Jun. 1, 2023, ISBN: 978-0-7503-4931-4 978-0-7503-4929-1. DOI: 10.1088/978-0-7503-4931-4

[103] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review", *Foundations and Trends® in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012, ISSN: 1935-8237. DOI: 10.1561/2200000036

[104] V. L. Srinivas and J. Wu, "Topology and parameter identification of distribution network using smart meter and µpmu measurements", *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022. DOI: 10.1109/TIM.2022.3175043

[105] W. Song, J. He, J. Lin, H. Ye, X. Ling, and C. Lu, "Bias Analysis of PMU-Based State Estimation and Its Linear Bayesian Improvement", *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, pp. 1607–1617, 2024. DOI: 10.1109/TII.2023.3280328

[106] P. A. Pegoraro, C. Sitzia, A. V. Solinas, S. Sulis, D. Carta, and A. Benigni, "Compensation of Systematic Errors for Improved PMU-based Fault Detection and Location in Three-Phase Distribution Grids", *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2024. DOI: 10.1109/TIM.2024.3400340

[107] Y. Li, S. Zhang, Y. Li, J. Cao, and S. Jia, "Pmu measurements-based short-term voltage stability assessment of power systems via deep transfer learning", *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023. DOI: 10.1109/TIM.2023.3311065

[108] A. Gholami et al., "Detection and classification of anomalies in power distribution system using outlier filtered weighted least square", *IEEE Transactions on Industrial Informatics*, vol. 20, no. 5, pp. 7513–7523, 2024. DOI: 10.1109/TII.2024.3360523

[109] V. S. Patel, S. Chakrabarti, and A. Sharma, "A Weighted Deep Neural Network for Processing Measurements for State Estimation", *IEEE Transactions on Industrial Informatics*, vol. 20, no. 3, pp. 3530–3538, Mar. 2024, ISSN: 1941-0050. DOI: 10.1109/TII.2023.3312402

[110] L. Liang and S. Liu, "Event-triggered distributed attack detection and fault diagnosis", *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2022. DOI: 10.1109/TIM.2022.3216674

[111] Y. Lin and A. Abur, "A highly efficient bad data identification approach for very large scale power systems", *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 5979–5989, 2018. DOI: 10.1109/TPWRS.2018.2826980

[112] O. Anderson and N. Yu, "Distribution system bad data detection using graph signal processing", in *2021 IEEE Power & Energy Society General Meeting (PESGM)*, 2021, pp. 01–05. DOI: 10.1109/PESGM46819.2021.9638014

[113] T. Zhang, Y. Wang, J. Ning, and M. Zhai, "Detection and identification of bad data based on neural network and k-means clustering", in *2019 IEEE Innov. Smart Grid Technol. - Asia (ISGT Asia)*, 2019, pp. 3634–3639. DOI: 10.1109/ISGT-Asia.2019.8881625

[114] J. Du, M. Hu, and W. Zhang, "Missing data problem in the monitoring system: A review", *IEEE Sensors Journal*, vol. 20, no. 23, pp. 13 984–13 998, 2020. DOI: 10.1109/JSEN.2020.3009265

[115] D. Wang, S. Zhang, M. Gan, and J. Qiu, "A Novel EM Identification Method for Hammerstein Systems With Missing Output Data", *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2500–2508, Apr. 2020, ISSN: 1941-0050. DOI: 10.1109/TII.2019.2931792

[116] S. Konstantinopoulos, G. M. De Mijolla, J. H. Chow, H. Lev-Ari, and M. Wang, "Synchrophasor Missing Data Recovery via Data-Driven Filtering", *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4321–4330, Sep. 2020, ISSN: 1949-3061. DOI: 10.1109/TSG.2020.2986439

[117] M. Liao, D. Shi, Z. Yu, Z. Yi, Z. Wang, and Y. Xiang, "An alternating direction method of multipliers based approach for pmu data recovery", *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4554–4565, 2019. DOI: 10.1109/TSG.2018.2864176

[118] A. Ghasemkhani, I. Niazazari, Y. Liu, H. Livani, V. A. Centeno, and L. Yang, "A Regularized Tensor Completion Approach for PMU Data Recovery", *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1519–1528, Mar. 2021, ISSN: 1949-3061. DOI: 10.1109/TSG.2020.3030566

[119] T. Wu, Y.-J. A. Zhang, Y. Liu, W. C. Lau, and H. Xu, "Missing Data Recovery in Large Power Systems Using Network Embedding", *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 680–691, Jan. 2021, ISSN: 1949-3061. DOI: 10.1109/TSG.2020.3014813

[120] Y. Gao, L. T. Yang, J. Yang, D. Zheng, and Y. Zhao, "Jointly Low-Rank Tensor Completion for Estimating Missing Spatiotemporal Values in Logistics Systems", *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1814–1822, Feb. 2023, ISSN: 1941-0050. DOI: 10.1109/TII.2022.3190549

[121] A. Grover and B. Lall, "A recursive method for estimating missing data in spatio-temporal applications", *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2714–2723, 2022. DOI: 10.1109/TII.2021.3100501

[122] J. J. Q. Yu, A. Y. S. Lam, D. J. Hill, Y. Hou, and V. O. K. Li, "Delay Aware Power System Synchrophasor Recovery and Prediction Framework", *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3732–3742, Jul. 2019, ISSN: 1949-3061. DOI: 10.1109/TSG.2018.2834543

[123] J. Fang, L. Zheng, and C. Liu, "A Novel Method for Missing Data Reconstruction in Smart Grid Using Generative Adversarial Networks", *IEEE Transactions on Industrial Informatics*, vol. 20, no. 3, pp. 4408–4417, Mar. 2024, ISSN: 1941-0050. DOI: 10.1109/TII.2023.3306366

[124] M. Osborne, S. Roberts, A. Rogers, S. Ramchurn, and N. Jennings, "Towards Real-Time Information Processing of Sensor Network Data Using Computationally Efficient Multi-output Gaussian Processes", in *International Conference on Information Processing in Sensor Networks, IPSN 2008*, May 2008, ISBN: 978-0-7695-3157-1. DOI: 10.1109/IPSN.2008.25

[125] *FIWARE - Open APIs for Open Minds*, en-US, Feb. 2021. [Online]. Available: https://www.fiware.org/

[126] IEC, "Alternating current static watt-hour meters for active energy (classes 1 and 2)", *Standard IEC 61036*, 2000.

[127] H. Wang, Z. Zhang, C.-Y. Weng, and X. Tang, "Robust Finite-Time Control for DC–DC Buck Converter With Inductor Current Constraint", *IEEE Transactions on Industrial Electronics*, vol. 71, no. 8, pp. 9631–9638, Aug. 2024, ISSN: 1557-9948. DOI: 10.1109/TIE.2023.3325589

[128] Z. Zhao, P. Davari, W. Lu, H. Wang, and F. Blaabjerg, "An overview of condition monitoring techniques for capacitors in dc-link applications", *IEEE Transactions on Power Electronics*, vol. 36, no. 4, pp. 3692–3716, 2021. DOI: 10.1109/TPEL.2020.3023469

[129] V. Raveendran, M. Andresen, G. Buticchi, and M. Liserre, "Thermal stress based power routing of smart transformer with chb and dab converters", *IEEE Transactions on Power Electronics*, vol. 35, no. 4, pp. 4205–4215, 2020. DOI: 10.1109/TPEL.2019.2935249

[130] B. Yao, Y. Peng, Y. Zhang, H. Wang, and H. Wang, "An Estimation Method of High-Order LC Circuits in Power Electronic Converters", *IEEE Transactions on Industrial Electronics*, vol. 71, no. 5, pp. 5274–5284, May 2024, ISSN: 1557-9948. DOI: 10.1109/TIE.2023.3285917

[131] Y. Avenas, L. Dupont, N. Baker, H. Zara, and F. Barruel, "Condition monitoring: A decade of proposed techniques", *IEEE Industrial Electronics Magazine*, vol. 9, no. 4, pp. 22–36, 2015. DOI: 10.1109/MIE.2015.2481564

[132] M. Al-Greer, M. Armstrong, M. Ahmeid, and D. Giaouris, "Advances on System Identification Techniques for DC–DC Switch Mode Power Converter Applications", *IEEE Transactions on Power Electronics*, vol. 34, no. 7, pp. 6973–6990, Jul. 2019, ISSN: 1941-0107. DOI: 10.1109/TPEL.2018.2874997

[133] Y. Zhang, Z. Wang, H. Wang, and F. Blaabjerg, "Artificial Intelligence-Aided Thermal Model Considering Cross-Coupling Effects", *IEEE Transactions on Power Electronics*, vol. 35, no. 10, pp. 9998–10 002, Oct. 2020, ISSN: 1941-0107. DOI: 10.1109/TPEL.2020.2980240

[134] L. Ren, C. Gong, and Y. Zhao, "An online ESR estimation method for output capacitor of boost converter", *IEEE Transactions on Power Electronics*, vol. 34, no. 10, pp. 10 153–10 165, Oct. 2019, ISSN: 1941-0107. DOI: 10.1109/TPEL.2018.2890617

[135] H. Soliman, I. Abdelsalam, H. Adaptive Optima, and F. Blaabjerg, "Artificial Neural Network based DC-link capacitance estimation in a diode-bridge front-end inverter system", in *2017 IEEE 3rd International Future Energy Electronics Conference and ECCE Asia (IFEEC 2017 - ECCE Asia)*, Jun. 2017, pp. 196–201. DOI: 10.1109/IFEEC.2017.7992442

[136] B. X. Li and K. S. Low, "Low Sampling Rate Online Parameters Monitoring of DC–DC Converters for Predictive-Maintenance Using Biogeography-Based Optimization", *IEEE Transactions on Power Electronics*, vol. 31, no. 4, pp. 2870–2879, Apr. 2016, ISSN: 1941-0107. DOI: 10.1109/TPEL.2015.2472459

[137] M. Giftthaler, M. Neunert, M. Stäuble, M. Frigerio, C. Semini, and J. Buchli, "Automatic differentiation of rigid body dynamics for optimal control and estimation", *Advanced Robotics*, vol. 31, Sep. 2017. DOI: 10.1080/01691864.2017.1395361

[138] T. J. Peifer M, "Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting", in *IET Syst Biol.*, vol. 1(2), 2007, pp. 78–88. DOI: 10.1049/iet-syb:20060067

[139] S. Zhao, F. Blaabjerg, and H. Wang, "An Overview of Artificial Intelligence Applications for Power Electronics", *IEEE Transactions on Power Electronics*, vol. 36, no. 4, pp. 4633–4658, Apr. 2021, ISSN: 1941-0107. DOI: 10.1109/TPEL.2020.3024914

[140] S. Chen, J. Zhang, S. Wang, P. Wen, and S. Zhao, "Circuit Parameter Identification of Degrading DC-DC Converters Based on Physics-informed Neural Network", in *2022 Prognostics and Health Management Conference (PHM-2022 London)*, May 2022, pp. 260–268. DOI: 10.1109/PHM2022-London52454.2022.00053

[141] A. G. Priyanka, A. Monti, and F. Ponci, "A data-driven technique for network line parameter estimation using gaussian processes", in *2023 IEEE Power & Energy Society General Meeting (PESGM)*, 2023, pp. 1–5. DOI: 10.1109/PESGM52003.2023.10252495

[142] D. Park, S. Park, J. H. Kim, and J. M. Lee, "Bayesian parameter estimation of nonlinear differential equations using automatic differentiation", in *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, 2018, pp. 1–6.

[143] R. H. Tan and L. Y. H. Hoo, "DC-DC converter modeling and simulation using state space approach", in *2015 IEEE Conference on Energy Conversion (CENCON)*, Oct. 2015, pp. 42–47. DOI: 10.1109/CENCON.2015.7409511

[144] Y. Ma, V. Dixit, M. J. Innes, X. Guo, and C. Rackauckas, "A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions", in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, 2021, pp. 1–9. DOI: 10.1109/HPEC49654.2021.9622796

[145] J. Poon, P. Jain, C. Spanos, S. K. Panda, and S. R. Sanders, "Fault prognosis for power electronics systems using adaptive parameter identification", *IEEE Transactions on Industry Applications*, vol. 53, no. 3, pp. 2862–2870, 2017. DOI: 10.1109/TIA.2017.2664052

[146] I. Khajehzadeh and B. Vale, "How new zealanders distribute their daily time between home indoors, home outdoors and out of home", *Kōtuitui: New Zealand Journal of Social Sciences Online*, vol. 12, no. 1, pp. 17–31, 2017. DOI: 10.1080/1177083X.2016.1187636

[147] E. C. S. O. of the European Union., *Eurostat regional yearbook: 2022 edition*. Publications Office, 2022. DOI: 10.2785/915176 [Online]. Available: https://data.europa.eu/doi/10.2785/915176

[148] "Ergonomics of the thermal environment – Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria (ISO 7730:2005)", International Organization for Standardization, Geneva, CH, Standard, Nov. 2005.

[149] S. C. Folea and G. D. Mois, "Lessons Learned From the Development of Wireless Environmental Sensors", en, *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 3470–3480, Jun. 2020, ISSN: 0018-9456, 1557-9662. DOI: 10.1109/TIM.2019.2938137

[150] S. Sun, H. Huang, T. Peng, and D. Wang, "An improved data privacy diagnostic framework for multiple machinery components data based on swarm learning algorithm", *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–9, 2023. DOI: 10.1109/TIM.2023.3310057

[151] V. Scotti, "Big data or big (privacy) problem?", *IEEE Instrumentation & Measurement Magazine*, vol. 20, no. 5, pp. 23–26, 2017. DOI: 10.1109/MIM.2017.8036692

[152] A. Čulić, S. Nižetić, P. Šolić, T. Perković, and V. Čongradac, "Smart monitoring technologies for personal thermal comfort: A review", *Journal of Cleaner Production*, vol. 312, p. 127 685, 2021, ISSN: 0959-6526. DOI: https://doi.org/10.1016/j.jclepro.2021.127685

[153] A. K. Kanál and K. Tamás, "Assessment of Indoor Air Quality of Educational Facilities using an IoT Solution for a Healthy Learning Environment", in *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, May 2020, pp. 1–6. DOI: 10.1109/I2MTC43012.2020.9129231

[154] P. Althaus, F. Redder, E. Ubachukwu, M. Mork, A. Xhonneux, and D. Müller, "Enhancing Building Monitoring and Control for District Energy Systems: Technology Selection and Installation within the Living Lab Energy Campus", en, *Applied Sciences*, vol. 12, no. 7, Jan. 2022, ISSN: 2076-3417. DOI: 10.3390/app12073305

[155] V. A. Arowoiya, R. C. Moehler, and Y. Fang, "Digital twin technology for thermal comfort and energy efficiency in buildings: A state-of-the-art and future directions", *Energy and Built Environment*, 2023, ISSN: 2666-1233.

[156] H. Seo and W.-S. Yun, "Digital twin-based assessment framework for energy savings in university classroom lighting", *Buildings*, vol. 12, no. 5, 2022, ISSN: 2075-5309. DOI: 10.3390/buildings12050544

[157] A. Schweigkofler et al., "Digital Twin as energy management tool through IoT and BIM data integration", *CLIMA 2022 conference*, Apr. 2022. DOI: 10.34641/clima.2022.46

[158] M. Pau, F. Ponci, A. Monti, C. Muscas, and P. A. Pegoraro, "Distributed state estimation for multi-feeder distribution grids", *IEEE Open Journal of Instrumentation and Measurement*, vol. 1, pp. 1–12, 2022. DOI: 10.1109/OJIM.2022.3198470

[159] M. Vanin, T. Van Acker, R. D'hulst, and D. Van Hertem, "Exact modeling of non-gaussian measurement uncertainty in distribution system state estimation", *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023. DOI: 10.1109/TIM.2023.3287253

[160] S. Lu, Y. Gong, H. Luo, F. Zhao, Z. Li, and J. Jiang, "Heterogeneous multi-task learning for multiple pseudo-measurement estimation to bridge gps outages", *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–16, 2021. DOI: 10.1109/TIM.2020.3028438

[161] P. Althaus, S. Johnen, A. Xhonneux, and D. Müller, "Cloud-based controller architecture for the testing of conventional and model predictive room heating controllers in a real-world environment", in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, Jul. 2024, pp. 1792–1797. DOI: 10.1109/CoDIT62066.2024.10708077

[162] F. Redder et al., "Information and Communication Technologies (ICT) for Holistic Building Energy System Operation in Living Labs: Conceptualization, Implementation, Evaluation", *Preprint available at SSRN: https://ssrn.com/abstract=4743282*, 2024. DOI: 10.2139/ssrn.4743282

[163] Crate.io, *CrateDB: The Database for Real-Time Analytics and Hybrid Search*, https://cratedb.com/product/self-managed, since 2013.

[164] PyADS, *PyADS: A a python wrapper for TwinCATs ADS library*, https://github.com/stlehmann/pyads, since 2015.

[165] Eclipse Foundation, *Eclipse Paho MQTT Python client library*, https://github.com/eclipse-paho/paho.mqtt.python, since 2014.

[166] Eclipse Foundation, *Eclipse Mosquitto: An open source MQTT broker*, https://mosquitto.org/, since 2009.

[167] Crate.io, *Python DB API client library for CrateDB, using HTTP.* https://github.com/crate/crate-python, since 2013.

[168] Docker, Inc., *Docker: Develop faster. Run anywhere.* https://www.docker.com/, since 2013.

[169] Open Infrastructure Foundation, *OpenStack: The Most Widely Deployed Open Source Cloud Software in the World*, https://www.openstack.org/, since 2010.

[170] X. Zhi, S. Yuexin, M. Jin, Z. Lujie, and D. Zijian, "Research on the pearson correlation coefficient evaluation method of analog signal in the process of unit peak load regulation", in *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, 2017, pp. 522–527. DOI: 10.1109/ICEMI.2017.8265997

[171] A. Dairi, F. Harrou, S. Khadraoui, and Y. Sun, "Integrated multiple directed attention-based deep learning for improved air pollution forecasting", *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021. DOI: 10.1109/TIM.2021.3091511

[172] F. Pedregosa et al., "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[173] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration", in *Advances in Neural Information Processing Systems*, 2018.

[174] A. Paszke et al., *Pytorch: An imperative style, high-performance deep learning library*, 2019. arXiv: 1912.01703 [cs.LG].

[175] P. Graczyk, J. Małecki, and E. Mayerhofer, "A characterization of wishart processes and wishart distributions", *Stochastic Processes and their Applications*, vol. 128, no. 4, pp. 1386–1404, Apr. 1, 2018, ISSN: 0304-4149. DOI: 10.1016/j.spa.2017.07.010

[176] A. G. Wilson and Z. Ghahramani, *Generalised wishart processes*, Dec. 31, 2010. DOI: 10.48550/arXiv.1101.0240 arXiv: 1101.0240[stat].

[177] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference* (Springer Texts in Statistics). Springer New York, 2013, ISBN: 978-0-387-21736-9.

[178] P. Billingsley, *Probability and measure* (Wiley series in probability and mathematical statistics), 3. ed. New York, NY: Wiley, 1995, 593 pp., ISBN: 978-0-471-00710-4.

Band / Volume 700
Materials Design, Processing and Application of Proton-Conducting Oxides for Electrochemical Energy Conversion
Y. Zeng (2026), vii, 117 pp
ISBN: 978-3-95806-899-5

Band / Volume 701
Accelerating the discovery of alkaline-stable anion exchange membrane materials via computational exploration
F. P. Tipp (2026), xii, 135 pp
ISBN: 978-3-95806-900-8

Band / Volume 702
Methods for Investigating the Structure-Performance Correlation in Membrane Electrode Assemblies
N. Utsch (2026), XIII, 138, LXVII pp
ISBN: 978-3-95806-901-5

Band / Volume 703
Hot-Spot Formation in Cu(In,Ga)Se_2 Thin Film Solar Cells
S. Nofal (2026), xvii, 131 pp
ISBN: 978-3-95806-903-9

Band / Volume 704
Machine Learning Guidance of Manufacturing Process of Solid Oxide Cells
T. Le Dinh (2026), xxi, 122 pp
ISBN: 978-3-95806-904-6

Band / Volume 705
Drivers of spatiotemporal variability of European terrestrial ecosystem processes
C. Poppe Téran (2026), 254 pp
ISBN: 978-3-95806-905-3

Band / Volume 706
Characterization of Nitrogen Reduction Reaction Catalysts for the electrochemical Ammonia Synthesis
D. Sachse (2026), xi, 235, 6 pp
ISBN: 978-3-95806-907-7

Band / Volume 707
Autonomous Image Analysis to Accelerate the Discovery and Integration of Energy Materials
A. Colliard Granero (2026), xiv, 156 pp
ISBN: 978-3-95806-910-7 (Print)
ISBN: 978-3-95806-911-4 (E-Book)

Band / Volume 708
Effect of Soil Mechanical Properties, Root Tip Geometry, and Mucilage on Penetration Restistance to Root Growth
R. K. Janakiram (2026), xxviii, 98 pp
ISBN: 978-3-95806-914-5 (Print)
ISBN: 978-3-95806-915-2 (E-Book)

Band / Volume 709
European Energy Transition – Germany in the Heart of Europe
T. Klütz, P. Dunkel, T. Busch, J. Linssen, D. Stolten (2026), V, 54 pp
ISBN: 978-3-95806-916-9 (Print)
ISBN: 978-3-95806-917-6 (E-Book)

Band / Volume 710
Development of High-Efficiency Perovskite/Silicon Tandem Solar Cells
Q. Yang (2026), 264 pp
ISBN: 978-3-95806-920-6 (Print)
ISBN: 978-3-95806-921-3 (E-Book)

Band / Volume 711
No-Regret-Maßnahmen der integrierten Netzplanung mit Fokus auf Wasserstoff
T. G. Busch (2026), xx, 325 pp
ISBN: 978-3-95806-922-0 (Print)
ISBN: 978-3-95806-923-7 (E-Book)

Band / Volume 712
Data-Driven Modeling for Digital Representations in Energy Systems
M. C. Zimmer (2026), xv, 166 pp
ISBN: 978-3-95806-926-8 (Print)
ISBN: 978-3-95806-927-5 (E-Book)

Weitere ***Schriften des Verlags im Forschungszentrum Jülich*** unter
http://wwwzb1.fz-juelich.de/verlagextern1/index.asp

Energie & Umwelt / Energy & Environment
Band / Volume 712
ISBN 978-3-95806-927-5

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum