

Offline Zeitdatenreihensynchronisierung von Produktionsdaten

Offline Time-Series Data Synchronization of Production Data

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

vorgelegt von

Arno Schmetz

Berichter:

Univ.-Prof. Dr.-Ing. Achim Kampker, MBA

Univ.-Prof. Dr.-Ing. Jürgen Fleischer

Tag der mündlichen Prüfung: 17.02.2026

Vorwort

Diese Arbeit entstand während meiner Tätigkeit als Mitarbeiter am Fraunhofer-Institut für Produktionstechnologie IPT in Aachen und an der Fraunhofer-Forschungsfertigung Batteriezeile FFB in Münster.

Ich danke meinem Doktorvater Prof. Achim Kampker, Inhaber des Lehrstuhls Production Engineering of E-Mobility Components PEM an der RWTH Aachen University, für die Möglichkeit zur Promotion und das entgegengebrachte Vertrauen. Ebenso danke ich Prof. Jürgen Fleischer als zweitem Beisitzer.

Weiterhin bedanke ich mich bei den Kolleg*innen, die mich während der Arbeit unterstützt haben, für intensive Diskussionen und Feedbacks offen waren und mir bei der Durchführung der Experimente geholfen haben. Ebenso bedanke ich mich bei dem großartigen Team, dessen ich ein Teil sein darf und wir gemeinsam durch lustige, schwierige, stressige und schöne Situationen zwischen Stammtisch, Rallye, Feiern und vielen Memes gegangen sind. Danke für Korrekturen, Feedbacks, Technik und Diskussionen bei der Erstellung an Florenz, Jonas, Thomas, Jürgen, Axel, Maik, Simon, Tae Hun. Besonderer Dank geht dabei an Jonathan, der mich intensiv unterstützt hat, abenteuerliche Wort- und Satzschöpfungen, unzählige schlechte Wortwitze und interdisziplinäres Arbeiten mit mir ertragen hat und immer offen und ehrlich zur Stelle war.

Herzlich bedanken möchte ich mich bei meiner Familie, Freundinnen und Freunden, die mich stets unterstützen. Vielen Dank an meine engen Freunde Kristina, Svenja, Jenny, Lena, Kon, Henri, Sebi, Martin. Und natürlich Danke an Eva, die mich stets stärkt und begleitet.

Münster, April 2025

Arno Schmetz

Zusammenfassung

Datenbasierte Produktionsverbesserungen bieten signifikante Möglichkeiten, insbesondere bei der Optimierung von Produktionsprozessen. Hierzu ist die Qualität der zugrunde liegenden Daten entscheidend. In der Batterieproduktion kommt eine Vielzahl von verschiedenen Sensoren, Maschinen und Systemen zum Einsatz, sodass die Integration dieser Daten kritisch ist und die Qualität der Daten stark beeinflusst. Ein besonderer Aspekt der Datenqualität ist die Synchronizität, also die korrekte zeitliche Zuordnung der verschiedenen Daten zueinander. In der Praxis ist diese oft nicht gewährleistet, was die Nutzbarkeit der Daten erheblich einschränkt.

In dieser Arbeit wird das Problem der Asynchronizität der Produktionsdaten betrachtet, nachdem der Produktionsprozess abgeschlossen wurde. Es wird eine Methode vorgestellt, die in der Lage ist, eine nachträgliche Synchronisierung der Daten für die hochskalige Batterieproduktion zu ermöglichen.

Dieses Ziel wurde im Rahmen der Arbeit durch die Entwicklung einer Methode erreicht, die für die Synchronisierung Informationen über den Produktionsaufbau, Wissen zu Wirkzusammenhängen sowie die Eigenschaften der Zeitdatenreihen einbezieht. Mithilfe dieser Einsichten bewertet die Methode Synchronisierungsalgorithmen und bestimmt einen Vorschlag für die Synchronisierungsabläufe. Zusätzlich detektiert die Methode gleichzeitige Datenpunkte und richtet die Synchronisierung entlang dieser aus. Sie wurde in einer Reihe von Szenarien und unter verschiedenen Störgrößen systematisch getestet und evaluiert. Die Ergebnisse der Verifikation und Validierung zeigen eine grundsätzliche Nutzbarkeit und Anwendbarkeit der Methode sowie eine mögliche Automatisierung in einer hochskaligen Batterieproduktion. Abschließend deckt die Arbeit weitere Potenziale und Forschungsfelder auf, die künftig weiter untersucht werden können.

Summary

Data-driven production improvements offer significant potential, particularly in the optimization of manufacturing processes. The quality of the underlying data plays a crucial role in this context. In battery production, a wide range of sensors, machines, and systems is used, making data integration a critical challenge that significantly influences data quality. One key aspect of data quality is synchronicity, i.e., the correct temporal alignment of different data streams. In practice, such synchronicity is often lacking, which severely limits the usability of the data.

This thesis addresses the problem of asynchronous production data after the completion of the production process. A method is presented that enables the retrospective synchronization of data in large-scale battery production.

To achieve this, a method was developed that incorporates structural information about the production process setup, knowledge of causal relationships, and the characteristics of the time series data. Based on these inputs, the method evaluates synchronization algorithms and proposes a suitable synchronization procedure. In addition, it detects simultaneous data points and aligns the synchronization process accordingly. The method was systematically tested and evaluated across a range of scenarios and disturbance conditions. The results of the verification and validation demonstrate its fundamental applicability and potential for automating data synchronization in large-scale battery production. Finally, the work outlines further opportunities and research directions that may be explored in the future.

Inhaltsverzeichnis

Table of Contents

Abbildungsverzeichnis	V
Tabellenverzeichnis	IX
Formelzeichen und Abkürzungen.....	XI
1 Einleitung	1
1.1 Motivation.....	1
1.2 Forschungsfragen	6
1.3 Struktur der Arbeit	7
2 Grundlagen der Zeitdatenreihensynchronisierung	11
2.1 Praxisrelevanz und Industriebefragung	11
2.1.1 Data Science Arbeiten	11
2.1.2 Synchronisierung in der Produktion.....	14
2.2 Zeitformalisierung und -messung.....	16
2.2.1 Daten, Informationen und Wissen	17
2.2.2 Formalisierung der Zeit	18
2.2.3 Uhren in technischen Systemen.....	19
2.2.4 Zeitdatenreihen	26
2.3 Verteilte Systeme	35
2.4 Produktionsnetzwerke	39
2.5 Batteriezellfertigung.....	44
2.5.1 Batteriezellen und Eigenschaften	44
2.5.2 Produktionsprozesse der Batteriezellfertigung.....	46
2.5.3 Digitalisierung der Batteriezellfertigung	49
2.6 Vorverarbeitung von Produktionsdaten	52
2.6.1 Verarbeitungsmodelle Produktionsdaten	52
2.6.2 Untersuchung der Datenqualität	55
2.6.3 Datenqualität und Asynchronizität.....	56
2.7 Aufgabe der Synchronisierung von Zeitdatenreihen	69
2.7.1 Gleichzeitigkeit und Projektion	70
2.7.2 Übertragung auf Zeitdatenreihen	74
2.7.3 Implikationen durch Produktionsnetzwerke	75
2.8 Arten der Synchronisierungen	77
2.9 Online Synchronisierung	78

2.10 Kreuzmodale Bindung als Datenintegration	81
2.11 Anwendung in der Batteriezellfertigung	84
2.12 Zwischenfazit	86
3 Existierende Ansätze und Ableitung der Anforderungen	89
3.1 Offline Synchronisierung	89
3.1.1 Dynamic Time Warping	89
3.1.2 Systemidentifikation	96
3.1.3 PROFINET Sequence of Events	100
3.1.4 Weitere Ansätze	101
3.2 Anforderungen und Metriken für die Offline Zeitdatenreihensynchronisierung	102
3.3 Evaluierung der existierenden Ansätze	106
3.4 Zwischenfazit und Handlungsfelder	112
4 Entwicklung des Synchronisierungssystems	113
4.1 Konzeptionierung des Synchronisierungssystems	113
4.2 Eingangsgrößen	115
4.2.1 Produktionsaufbau und Unsicherheitskette	116
4.2.2 Charakterisierung Systeme	119
4.2.3 Wirkungen und Regeln	120
4.2.4 Manuelle Eingriffsmöglichkeiten	122
4.2.5 Vorbereitung der Synchronisierungsmethoden	122
4.3 Schritte des Synchronisierungssystems	123
4.3.1 Gesamtanalyse	123
4.3.2 Zerlegung	127
4.3.3 Entscheidung	133
4.3.4 Durchführung	140
4.4 Implementierung	142
4.4.1 Generelle Implementierungen	143
4.4.2 Implementierung der Eingangsgrößen	144
4.4.3 Implementierung der Systemmodule	146
4.4.4 Implementierung der Synchronisierungsmethoden	147
4.4.5 Schnittstellen für Menschen	148
4.5 Zwischenfazit Synchronisierungssystem	150
5 Verifikation und Validierung	153
5.1 Verifikation	154
5.1.1 Eingangstests	154

5.1.2	Integrationstest	157
5.1.3	Komplexität	159
5.1.4	Evaluierung gegen Anforderungen	164
5.2	Validierung des Synchronisierungssystems.....	166
5.2.1	Vorgehen für die Validierung	166
5.2.2	Beschreibung der Szenarien.....	169
5.2.3	Metriken der Bewertungen.....	178
5.2.4	Auswertung der Szenarien	179
5.3	Kritische Diskussion der Ergebnisse	198
5.3.1	Ergebnisdiskussion	199
6	Fazit.....	205
6.1	Betrachtung Forschungsfragen.....	205
6.2	Ausblick	207
6.3	Zusammenfassung	213
	Literaturverzeichnis	215
	Anhang.....	241
A.1	Bildverweise	241
A.2	Aufgaben Data Scientists und Bereinigungsschritte.....	242
A.3	Data Science Tasks Survey	244
A.4	Ergebnisse Umfrage Data Scientists Aufgaben, Vergnügen, Vorverarbeitung.....	250
A.5	Time Synchronization in Smart Manufacturing Survey	253
A.6	Time Synchronization in Smart Manufacturing Survey – Ergebnisse	260
A.7	Ablaufdiagramme System.....	262
A.8	Abläufe Methodenbewertung.....	263
A.9	Stellgrößen Synchronisierungssystem	267
A.10	Implementierungen und Software-Code	271
A.11	Verifikation und Validierung – Taxonomie nach BANKS.....	272
A.12	Verifikation und Validierung – Versuchsmatrix.....	274

Abbildungsverzeichnis

List of Figures

1.1	Visualisierung des »Internet of Production«	2
1.2	Vorgehensmodell für maschinelles Lernen in der Produktion	3
1.3	Wissenschaftssystematik und Verankerung nach ULRICH	8
1.4	Forschungsansatz nach ULRICH und HEVNER und Struktur dieser Arbeit ..	9
2.1	Umfrageergebnisse zu Umfang und Vergnügen der Data Science Aufgaben	13
2.2	Ergebnisse der Fragen zum relativen Arbeitsumfang der Vorverarbeitungsschritte.	14
2.3	Ergebnisse der Fragen zum Auftreten und der Größenordnung der Asynchronizität in der Produktion.....	15
2.4	Ergebnisse der Frage zum Adressieren der Asynchronizität in der Produktion	16
2.5	Netzzeitabweichung durch Abweichungen der Netzfrequenzen	21
2.6	Taktversatz in zwei synchronen Systemen.....	24
2.7	Gangfehler in zwei Systemen.....	25
2.8	Systemzeit zweier Systeme mit Gangfehler und Taktversatz	26
2.9	Darstellungen von Zeitdatenreihen.....	30
2.10	Das ISO/OSI Modell der Kommunikation	36
2.11	Die Bereiche Edge, Cloud, Fog und Mist als Architekturbild.....	39
2.12	Automatisierungspyramide und Referenzarchitektur Industrie 4.0	41
2.13	Angekündigte Batterieproduktionen in Europa	45
2.14	Produktionsprozesse der Lithium-Ionen-Batteriezellen	47
2.15	Exemplarischer Aufbau einer Beschichtung und Trocknung mit relativen Positionen	51
2.16	Data Science Schritte nach BISWAS	54
2.17	Pipeline der Datenvorverarbeitung nach FRYE.....	58
2.18	Zwei Zeitdatenreihen versetzt zur gleichen Entität	64
2.19	Zwei Zeitdatenreihen zur gleichen Entität unter Einfluss eines Gangfehlers	65
2.20	Graph der Synchronisierbarkeit verschiedener Systeme.....	72

2.21	Exemplarische Komposition von Projektionsfunktion	73
2.22	Einordnung der Online und Offline Synchronisierung	77
2.23	Darstellung der Strata in NTP-artigen Synchronisierungssystemen	80
2.24	Ablauf der Verarbeitung von auditiven und visuellen Informationen im intraparietalen Sulcus	83
3.1	Eine Synchronisierung von Zeitdatenreihen, durchgeführt mittels Dynamic Time Warping.....	90
3.2	Die durchgeführte Synchronisierung mittels Dynamic Time Warping, mit der internen Distanzmatrix in der Mitte	93
3.3	Kostengitter mit Sakoe-Chiba-Band und Itakura-Parallelogram	94
3.4	Angepasste Zeitstempel in Sequence of Events	100
4.1	Aufbau des gesamten Offline Synchronisierungssystems.....	114
4.2	Darstellung von Elementen des Produktionsnetzwerks als Komponenten und Komponentengruppen	116
4.3	Anreicherung der Zeitstempeldaten mit den Komponentenzeitinformationen	117
4.4	Schematischer Aufbau der Versuche zur Untersuchung der Asynchronizität.	119
4.5	Exemplarische Darstellung von asynchronen Zeitdatenreihen.....	128
4.6	Darstellung von identifizierten Fixpunkten.....	131
4.7	Darstellung von abgeleiteten Subsequenzen als Zerlegung.....	132
4.8	Exemplarische Ausgangslage innerhalb des Synchronisationsgraphs für die Bewertung von Methoden zwischen zwei Knoten.	134
4.9	Exemplarische Darstellung des Synchronisationsgraphs mit mehreren möglichen Kanten während der Optimierung	138
4.10	Darstellung der Komponenten der Implementierung des Offline Synchronisierungssystems gemäß UML.....	146
4.11	User Interface für die Betrachtung und Bearbeitung von Fixpunkten.....	149
4.12	User Interface für die Betrachtung und Bearbeitung von Synchronisationsgraphen	150
5.1	Darstellung der benötigten Zeiten bei veränderlicher Datenlänge.....	155

5.2	Darstellung der benötigten Zeiten bei der Methode <i>Zerlegung</i> , aufgeteilt in die Hauptschritte	156
5.3	Integrationskonzept des Offline Synchronisierungssystems in die IT-Architektur	158
5.4	Zeitabweichungen über eine Stunde von Maschine und Edge PC	167
5.5	Darstellung der Mikro- und Makro-Strukturen in den Asynchronizitäten..	168
5.6	Signalverläufe im Szenario 1	171
5.7	Signalverläufe im Szenario 2	172
5.8	Signalverläufe im Szenario 3	173
5.9	Signalverläufe im Szenario 4	174
5.10	Signalverläufe im Szenario 5	175
5.11	Signalverläufe im Szenario 6	176
5.12	Signalverläufe im Szenario 7	177
5.13	Auswertung der Anwendbarkeit im Szenario 1	180
5.14	Auswertung der Qualität im Szenario 1	181
5.15	Auswertung der benötigten Zeit im Szenario 1	182
5.16	Auswertung der Anwendbarkeit im Szenario 2	183
5.17	Auswertung der Qualität im Szenario 2	184
5.18	Auswertung der benötigten Zeit im Szenario 2	185
5.19	Auswertung der Anwendbarkeit im Szenario 2b	186
5.20	Auswertung der Qualität im Szenario 2b.....	187
5.21	Auswertung der benötigten Zeit im Szenario 2b.....	188
5.22	Auswertung der Anwendbarkeit im Szenario 3	189
5.23	Auswertung der Qualität im Szenario 3	190
5.24	Auswertung der benötigten Zeit im Szenario 3	191
5.25	Auswertung der Anwendbarkeit im Szenario 4	192
5.26	Auswertung der Qualität im Szenario 4	193
5.27	Auswertung der benötigten Zeit im Szenario 4	194
5.28	Auswertung der Anwendbarkeit im Szenario 5	195

5.29	Auswertung der Qualität im Szenario 5	196
5.30	Auswertung der benötigten Zeit im Szenario 5	197
5.31	Auswertung der Anwendbarkeit im Szenario 6	198
5.32	Auswertung der Qualität im Szenario 6	199
5.33	Auswertung der benötigten Zeit im Szenario 6	200
5.34	Auswertung der Anwendbarkeit im Szenario 7	201
5.35	Auswertung der Qualität im Szenario 7	202
5.36	Auswertung der benötigten Zeit im Szenario 7	203
6.1	Übersicht identifizierter Forschungsfelder im Ausblick	208
A.1	Der Ablauf der Segmentierung in Phase 1 der Bewertung von Metriken für die Synchronisierung.....	262
A.2	Die Taxonomie der Methoden für Verifikation, Validierung und Testing nach BANKS.....	273

Tabellenverzeichnis

List of Tables

2.1	Datenprobleme, adressiert in der Datenvorverarbeitung	57
3.1	Bewertung der aktuellen Ansätze gegen die Anforderungen der Batteriezellfertigung	107
4.1	Umsetzung der Wirkdatenbank als Tabellenstruktur in SQL	145
5.1	Bewertung der neuen Methodik gegen die Anforderungen	165
1	Aufgaben Data Scientists der Vorstudie	242
2	Vorverarbeitungsschritte aus der Vorstudie	243
3	Ergebnisse der Umfrage zur Zeit für die Aufgaben als Data Scientist.....	250
4	Ergebnisse der Umfrage zum Vergnügen für die Aufgaben als Data Scientist	251
5	Ergebnisse der Umfrage zur relativen Arbeitslast der Vorverarbeitungsschritte als Data Scientist.....	252
6	Ergebnisse zur Frage, ob Asynchronizität in den Produktionen aufgetreten ist.....	260
7	Ergebnisse zur Frage, in welcher Größenordnung Asynchronizität in den Produktionen aufgetreten ist.....	260
8	Ergebnisse zur Frage, in welcher Frequenz Daten aufgenommen wurden in den Produktionen.....	260
9	Ergebnisse zur Frage der Menge von Geräten in der Produktion.	261
10	Stellgrößen des Synchronisierungssystems.....	270
11	Versuchsmatrix für die Validierung und ihre Störgrößen.....	275

Formelzeichen und Abkürzungen

Symbols and Abbreviations

Formelzeichen

Zeichen	Einheit	Beschreibung
t_0		Moment der Zeit im betrachteten System bei Start der Observation
t_x		Moment der Zeit im betrachteten System bei Abschluss des x ten Zyklus der Uhr
T_α		Ein fester Punkt in der echten Zeit
τ_{ideal}	[s]	Ideale Zykluszeit für die Oszillatoruhr
τ_{real}	[s]	Tatsächliche Zykluszeit für die Oszillatoruhr
C_s		Betrachtetes System s
$C_s(t_x)$		Die errechnete Systemzeit von System s nach Abschluss des x ten Zyklus der Uhr des Systems
$C_s(T_\alpha)$		Die Systemzeit von System s zu echten Zeit T_α
Δ_{skew}	[s]	Taktversatz zwischen zwei Uhren
δ	$[\frac{s}{s}]$	Gangfehler zwischen zwei Uhren
$\Delta_{skew}^s, \delta^s$	[s], $[\frac{s}{s}]$	Taktversatz und Gangfehler der Uhren in System s im Verhältnis zur echten Zeit
$V, {}_lV$		Eine Datenreihe
$U, U_V^{k,m}$		Eine Subsequenz aus der Datenreihe
$Z, {}_lZ$		Eine Zeitdatenreihe
$ V , Z $		Anzahl der Werte in der (Zeit)datenreihe
$\overset{\circ}{U}, {}_l\overset{\circ}{U}$		Eine Subsequenz einer Zeitdatenreihe
$d(z_i, z_j)$		Zeitliche Distanz zwischen zwei Zeitdaten
$\mathfrak{Z}$		Menge aller Zeitdatenreihen
ϵ		Leerer Wert
σ		Standardabweichung
$\gamma(Z)$		Schiefe einer Zeitdatenreihe

$\eta(Z)$		Wölbung einer Zeitdatenreihe
$\bar{t}$	[s]	Durchschnittliche Zeitangabe innerhalb einer Zeitdatenreihe
Var_Z		Variation der Zeitverteilung innerhalb einer Zeitdatenreihe Z
f_Z	[Hz]	Frequenz einer Zeitdatenreihe
SS	[m]	Position auf der Bahn (»Space Stamp«)
TS	[s]	Zeitstempel eines Datenpunkts (»Time Stamp«)
$\vec{v}$	$[\frac{m}{s}]$	Bahngeschwindigkeit
$\equiv$		(Vergleichsoperator) Identische Bedeutung (Semantik) der beiden syntaktischen Darstellungen
$\mathbb{P}$		Menge aller Regeln, die sich aus dem Wissen zur Domäne ergeben
$\doteq$		(Vergleichsoperator) Gilt, wenn beide Seiten denselben Zeitpunkt der echten Zeit beschreiben
Δ_{corr}	[s]	Korrektur-Offset eines Systems
τ_{Error}	[s]	Erwartete Ungenauigkeit der Zykluszeiten
$\Phi_{n,m}$		Projektionsfunktion für Zeiten aus C_n auf C_m
$C_n \succ_l Z$		Zeitdatenreihe $_l Z$ ist System C_n zugeordnet
$\mathfrak{I}$		Gesamtheit aller genutzten Informationen
$\mathfrak{W}$		Gesamtheit allen genutzten Wissens
$\mathfrak{M}$		Menge der manuellen Eingriffe
G_{sync}		Synchronisationsgraph bestehend aus Knoten (KN) und Kanten (KA)
KA		Menge der Kanten eines Graphs
KN		Menge der Knoten eines Graphs
λ		Kantenlabel eines Graphs
M		Menge der Synchronisierungsmethoden
Λ		Knotenlabel eines Graphs

$\mathbb{S}$		Menge der Sensoren in der Produktion
$\mathbb{E}$		Menge der gemessenen physikalischen Effekte in der Produktion
$s_i \perp E_j$		(Relation) Sensor s_i misst Effekt e_j
$\overset{\circ}{d}_{s_m, s_n}$	[s]	Zeitliche Distanz der Propagation zwischen den Sensoren s_m und s_n
Γ		Die Menge aller Ursache-Wirkzusammenhänge
$f \rightsquigarrow g$		Ursache-Wirkzusammenhang von Funktion f zu g
$\Xi = \{\xi\}$		Die Menge aller identifizierten wahrscheinlichen Fixpunkte
Ξ_1		Die Menge aller identifizierten sicheren Fixpunkte
$\Delta(\xi)$	[s]	Ungenauigkeit eines Fixpunktes
$\iota \overset{\circ}{U}(\Xi)$		Zerlegung einer Zeitdatenreihe ιZ in Subsequenzen

Abkürzungen

AAS	Asset Administration Shell
ANSI	American National Standards Institute
API	Application Programming Interface
ARX	AutoRegressive with eXogenous inputs
ARMAX	AutoRegressive Moving Average with eXogenous inputs
BMBF	Bundesministerium für Bildung und Forschung
BPIM	Internationales Büro für Maß und Gewicht
CPU	Central Processing Unit
HMI	Human Machine Interface
HOG1D	Histogram of Oriented Gradients for 1D Data
IEEE	Institute of Electrical and Electronics Engineers
IEC	International Electrotechnical Commission

IETF	Internet Engineering Task Force
IIoT	Industrial Internet of Things
ISO	International Standardization Organization
ISA	International Society of Automation
IoP	Internet of Production
IoT	Internet of Things
KPI	Key Performance Indicator (»Leistungskennzahl«)
LIB	Lithium-Ionen-Batterien
LIMS	Labor-Information-Management-System
LTS	Long Term Support
MES	Manufacturing Execution System
ML	Maschinelles Lernen
NARMAX	Nonlinear AutoRegressive Moving Average with eXogenous inputs
NARX	Nonlinear AutoRegressive with eXogenous inputs
NIB	Natrium-Ionen-Batterien
NMI	Normalized Mutual Information
NTP	Network Time Protocol
OCV	Open Circuit Voltage (Leerlaufspannung)
OCXO	Oven Controlled Crystal Oscillator
OPC UA	Open Platform Communications Unified Architecture
(OPC UA) CS	(OPC UA) Companion Specification
OSI	Open Systems Interconnection
OSS	Offline Synchronisierungssystem
PLC	Programmable Logic Controller
PMI	Pattern Matching Index
PTP	Precision Time Protocol
REST	Representational State Transfer

RFC	Request For Comments (Technische Normen und Standards für das Internet)
SCADA	Supervisory Control and Data Acquisition
SDK	Software Development Kit
SEI	Solid Electrolyte Interface
SI	Systemidentifikation
SOA	Service-orientierte Architektur
SoE	Sequence of Events
SPS	Speicherprogrammierbare Steuerung
SQL	Simple Query Language
SSI	Stochastische Subraumidentifikation
TSN	Time-sensitive Networking
UTC	Coordinated Universal Time (Koordinierte Weltzeit)
ZVEI	Verband der Elektro- und Digitalindustrie

1 Einleitung

Introduction

Im Mittelpunkt dieser Dissertationsschrift steht die Untersuchung der Offline Zeitdatenreihensynchronisierung für Produktionsdaten. Dabei dient die Batteriezellfertigung als Anwendungsbereich der Arbeiten. Mit Blick auf die steigende Verbreitung der datenbasierten Verfahren und Komponenten in modernen Produktionen spielt die Sicherung der Datenqualität eine zentrale Rolle, wozu diese Zeitdatenreihensynchronisierung beiträgt.

Im Folgenden werden die Motivation und die sich daraus ergebenden Forschungsfragen dargestellt. Darüber hinaus wird der Aufbau dieser Arbeit vorgestellt.

1.1 Motivation

Motivation

Die Digitalisierung und die damit verbundenen datenbasierten Dienste bieten enorme Potenziale im Bereich der Produktion, von Qualitätssicherung über Kostensenkungen bis hin zu neuartigen Produkten. Diese Revolution der Produktion entstand insbesondere im Jahre 2011 als Zukunftsprojekt im Rahmen der Hightech-Strategie des Bundesministeriums für Bildung und Forschung (BMBF) [BUND16b, KAGE11]. Im Folgenden entstand die »Plattform Industrie 4.0« als Netzwerk, welches durch Arbeitsgruppen Ergebnisse und Empfehlungen erarbeitet. Eines der Ergebnisse ist dabei das »Leitbild 2030 für die Industrie 4.0«. Darin werden strategische Handlungsfelder der Industrie 4.0 beschrieben. Im Handlungsfeld der Interoperabilität wird dabei betont, dass die Integration, statt getrennter Einzellösungen, sowie die Verknüpfung von verschiedenen Daten aus der Produktion zwingend notwendig ist. [PLAT19]

Seit der ersten Ankündigung ist Industrie 4.0 weltweit bekannt und wird mit internationalen Aktivitäten wie »Smart Manufacturing« [LU20] zusammen beforscht. Weitere Aktivitäten unter dem Begriff »Industrie 5.0« werden teilweise als Weiterentwicklung, um den Menschen in den Vordergrund zu setzen, beschrieben [GOLO23]. Gleichzeitig verwerfen beispielsweise der deutsche Forschungsbeirat und Plattform Industrie 4.0 explizit diese Begrifflichkeit und damit verbundene Unterscheidung zu Industrie 4.0 [LIGG24].

Um die angestrebte Transformation zur Industrie 4.0 zu erreichen, ist das sogenannte »Internet of Production« [BREC17] ein durchgängiger Ansatz. Hierbei wird die Vernetzung vom Shopfloor bis zu den Mehrwertdiensten komplett betrachtet, dargestellt in Abbildung 1.1. Die Daten werden hier aus den verschiedenen Quellen wie Anlagen, Messtechnik und

Systemen extrahiert und als Rohdaten transportiert. Dies geschieht oftmals über proprietäre und nicht standardisierte Schnittstellen. Hierzu hilft die Ebene der Middleware, welche diese Zugriffe verwaltet und die Aggregation und Synchronisierung unterstützt. Die Daten werden weiterverarbeitet und für Analysen angepasst und dort verwendet (Smart Data). Zuletzt kommen diese Daten und extrahierte Informationen in die Agenten- und Entscheidungssysteme (oberste Ebene) um Unternehmen die bestmögliche Entscheidungsfindung datenbasiert zu ermöglichen. [SCHM20]

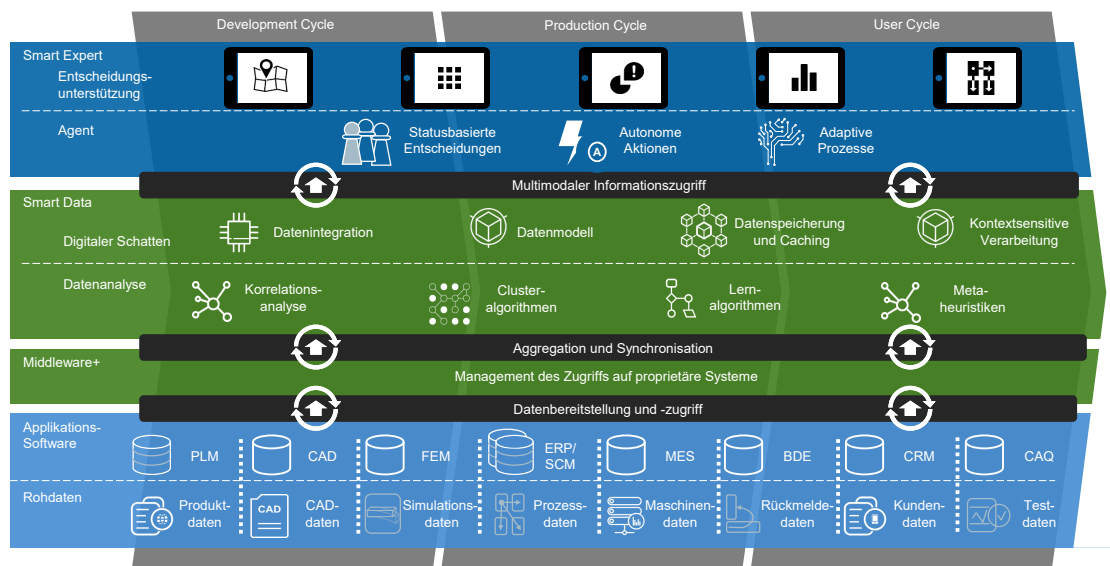


Abbildung 1.1: Visualisierung des »Internet of Production« (IoP). Es besteht aus mehreren Ebenen vom Shopfloor (unten) bis zu den Mehrwertdiensten (oben). Diese Ebenen sind dabei durch Zwischenebenen verbunden. (Basierend auf [BREC17] und [SCHM20])

Datenqualität in der Industrie 4.0

Durch die Darstellung des IoP ist erkennbar, dass die Sicherstellung der Datenqualität in der Aggregation und Synchronisierung einen maßgeblichen Einfluss auf die späteren Schritte und die Güte der Ergebnisse hat. Bei der Nutzung von Daten ist daher eine Überprüfung und Untersuchung der Datenqualität notwendig [PIPI02]. In der Praxis scheitern daher viele der datengetriebenen Anwendungen in der Industrie eben aufgrund der mangelnden Datenqualität und der resultierenden schlechten Ergebnisse der Modelle [FRYE21]. Im

Bereich der Produktion hat FRYE den Einfluss der Datenqualität und die Notwendigkeit einer Vorverarbeitung untersucht. Dabei sind verschiedene Eigenschaften der Datenqualität und Methoden der Adressierung als Datenbereinigung, -transformation, -reduktion und -augmentierung betrachtet worden. Die Synchronizität der Daten aus mehreren Quellen ist dabei einer der Qualitätsindikatoren, die zu beachten sind. [FRYE22]

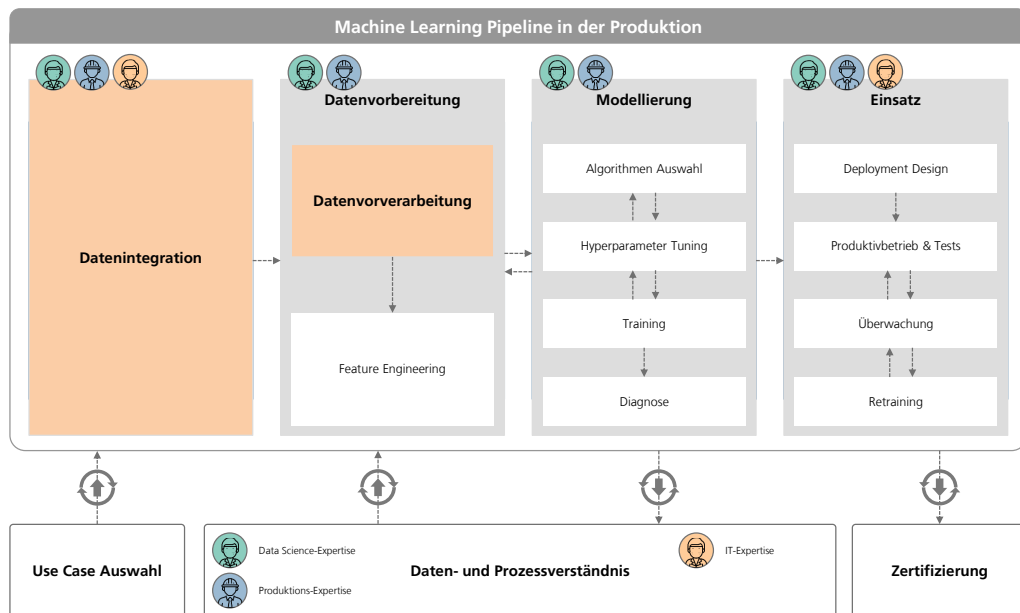


Abbildung 1.2: Vorgehensmodell für den Einsatz von datenbasierten Methoden, konkret maschinelles Lernen, in der Produktion, basierend auf dem AutoML Modell. (Darstellung angepasst aus [KRAU22])

Bei datengetriebenen Entwicklungen in der Produktion kommt oftmals maschinelles Lernen (Machine Learning) zum Einsatz. Hierzu wird in der Praxis auf das Vorgehensmodell »AutoML« zurückgegriffen, worin die Schritte von Datenerhalt bis zur Zertifizierung eines Modells betrachtet werden [HE21]. In Bezug auf die Produktion ist dieses Modell in Abbildung 1.2 dargestellt. Die Pipeline startet mit dem Schritt der Datenintegration. Innerhalb dieses Schrittes müssen die Daten aus verschiedenen Quellen zusammengeführt werden. Bei Zeitdatenreihen umfasst dies ebenso die Aufgabe der Synchronisierung, sodass die nachfolgenden Schritte auf zeitlich korrekte Reihenfolgen der Daten zurückgreifen können. Im folgenden Schritt werden die Daten vorbereitet. Darin werden die Daten untersucht und Maßnahmen zur Korrektur von Fehlern wie fehlenden Werten, Ausreißern und falschen Wertebereichen ergriffen (Datenvorverarbeitung). Hier können ebenfalls Teilschritte der Synchronisierung vorgenommen werden. Aus den Daten mit sichergestellter Qualität kön-

nen dann Features für die Analyse- und Modellierungsarbeiten extrahiert werden (Feature Engineering). Der nächste Schritt (Modellierung) setzt bereits qualitativ hochwertige Daten voraus. Hier werden abhängig von den Zielen der Analysen und Optimierungen die Algorithmen ausgewählt, Parameter angepasst und das Training durchgeführt. Zuletzt wird das Modell auch diagnostiziert für die Anwendbarkeit auf das Problem. Im letzten großen Schritt wird das Modell in Einsatz gebracht (Deployment). Dazu wird diese In-Einsatz-Bringung entworfen, durchgeführt und evaluiert. Auch die Güte der Modellergebnisse wird kontinuierlich überwacht, sodass bei abfallender Güte ein Retraining durchgeführt werden kann. Entlang all dieser Schritte wird besondere Expertise benötigt aus verschiedenen Domänen. Besonders zu betonen ist dabei die Expertise der Produktion, also dem Kontext, in dem die gesamte Entwicklung stattfindet [KRAU22]. Der Fokus dieser Arbeit befindet sich in den orange markierten Schritten »Datenintegration« und »Datenvorverarbeitung«, und liegt damit am Anfang der Pipeline mit hohem Einfluss auf die Güte und Umsetzbarkeit aller folgenden Schritte.

Batteriezellfertigung

Im Zeichen von Klimawandel und Abhängigkeit von nicht-europäischen Produzenten ist der Aufbau und die Innovation in der Batteriezellfertigung in Europa eine strategische Schlüsseltechnologie. Der Markt alleine bei den Lithium-Ionen-Batterien (LIB) zeigte ein Handelsvolumen von 50 bis 60 Milliarden Euro für die entsprechenden Batteriezellen im Jahr 2021. Gleichzeitig gehen aktuelle Studien von einem Wachstum des Handelsvolumens, bei sinkenden Kosten, auf bis zu 300 Milliarden Euro im Jahr 2030 aus. Auf der technischen Ebene lag die Nachfrage nach LIB im Jahr 2021 bei einem Volumen von etwa 470 GWh. Je nach Prognose steigt diese Nachfrage auf 2 bis 4 TWh im Jahr 2030. Dadurch ist die Batteriezellfertigung ein strategisch relevantes Wachstumsfeld. [BÜRK22]

Innerhalb der Batteriezellfertigung sind ebenfalls datenbasierte Verfahren relevant und einsetzbar. Wissenschaftlich ist dies ein aktives Forschungsfeld mit umfangreichen Ansätzen und Entwicklungen [HAGH22, AYER22, VS23]. Industrieseitig ist der Bedarf ebenfalls hoch an Digitalisierung. So wird in einer aktuellen Studie der Mehrwert und die potenziellen Kosteneinsparungen durch Digitalisierung entlang von Use Cases im Bereich von mehreren Milliarden Euro geschätzt [WUND24]. Gleichzeitig stehen diesen Potenzialen Herausforderungen und Blocker gegenüber. Insbesondere ist die Batteriezellfertigung eine hochkomplexe Produktionskette, bei der Abweichungen und Einflüsse in den frühen Schritten kritische Einflüsse auf Folgeprozesse und die finale Qualität haben. Aufgrund der Art der Prozesse ist dabei die Auswirkung oftmals erst nach Abschluss aller Prozessschritte

sichtbar. Dies führt zu hohen Ausschussraten und technisch zu einer hohen Komplexität der Fertigung. [KAMP23, KRAU23]

So ist beispielsweise der Prozess der Beschichtung in der Elektrodenproduktion komplex und in der Prozesskette eng verbunden. Die Qualität der Folien und des gemischten Slurry aus den vorhergegangenen Schritten beeinflussen direkt die Parametrierung und die Qualität der beschichteten Elektrodenfolie. Ebenso umfasst der Prozess in sich eine Vielzahl von Parametern, die direkt die Qualität der Beschichtung beeinflussen. Bereits kleine Veränderungen machen sich in der Trocknung direkt bemerkbar, aber ebenso wesentlich später bei der Finalisierung der Batteriezellen. Dadurch ist dieser Prozess mit der gesamten Prozesskette komplex verknüpft und bedarf einer ganzheitlichen Betrachtung. Allerdings sind eben jene prozessübergreifende Wirkzusammenhänge bisher wenig betrachtet und eine der Hauptherausforderungen in der Batteriezellfertigung. [MICH23]

Synchronizität von Zeitdatenreihen

Wie oben beschrieben sind datenbasierte Verfahren in der Batteriezellfertigung strategisch notwendig und bergen hohe Potenziale. Dem entgegen stehen die Probleme der Datenqualität. Diese Probleme werden durch den Mangel an Wissen über die Wirkzusammenhänge und die Komplexität der Produktion verschärft. Der erste Schritt bei den datenbasierten Verfahren ist hier die Datenintegration, bei der die Daten aus verschiedenen Quellen zusammengeführt werden. Zentral sind hier Daten aus der Messtechnik, der Anlage und der Umgebung, die über die Zeit betrachtet werden. Dies

sind sogenannte Zeitdatenreihen. Innerhalb der Datenintegration werden diese Zeitreihen synchronisiert, sodass gleichzeitige Datenpunkte auch in dem Datensatz entsprechend verbunden sind.

Definition

Eine **Datenreihe** beschreibt eine Abfolge von Werten (z.B. ein Sensorsignal) ohne weitere Angabe von Zeiten.

Eine **Zeitreihe** hingegen ist eine Reihe von Zeitangaben ohne die Angabe von Werten.

Die Kombination von Zeiten und Werten als Paare ist eine **Zeitdatenreihe**.

Die Messung von Zeit in technischen Systemen erfolgt durch Oszillatoren (z.B. Quarzkristalle), deren Schwingungen gezählt werden. Dadurch kann ein System die Uhrzeit errechnen durch einen fixen Punkt T_0 in der Zeit, der Anzahl n an Schwingungen seit diesem Zeitpunkt,

und der Frequenz f des Oszillators. Daraus ergibt sich die Berechnung der aktuellen Zeit T_{jetzt} zu $T_{jetzt} = T_0 + \frac{n}{f}$. In der Praxis allerdings entstehen hier Abweichungen aufgrund von verschiedenen physikalischen Einflüssen wie Temperatur oder Gravitation [HORO89]. In der Konsequenz ist die gleiche dargestellte Zeit in zwei Systemen in Wirklichkeit verschieden. Für Zeitreihen bedeutet dies, dass die Datenintegration entlang der Zeitinformatoren ungenau ist. Wird hier nicht im Betrieb eine hohe Genauigkeit sichergestellt, leidet die Datenqualität erheblich unter diesem Effekt [SCHM22]. Bei bahnbasierten Prozessen wie in der Elektrodenfertigung der Batteriezellfertigung verursachen diese Effekte Ungenauigkeiten bei der Zuordnung von Daten auf die Bahnposition. Dieser Effekt wurde in der Praxis fabrikübergreifend regelmäßig im Bereich von »geringen zweistelligen cm« bei der Bahnposition in der Batteriezellproduktion durch Siemens benannt [WITT23]. In der zeitlichen Auflösung zeigt der kumulierte Effekt eine Größenordnung von Sekundenbruchteilen bis Minuten oder noch länger [SCHM24b]. Diese Größenordnung kann je nach eingesetztem Zellformat bereits die Größe der zugeschnittenen Elektroden überschreiten. Dadurch wäre eine elektrodengenaue Zuordnung von Produktionsdaten nur unzureichend möglich.

Um das Problem der mangelnden Genauigkeit der Zuordnung, aufgrund von Asynchronizität, zu lösen, kommen Methoden der Synchronisierung zum Einsatz. Dabei wird zwischen einer Synchronisierung der Systeme im Betrieb (»Online Synchronisierung«) und im Nachgang (»Offline Synchronisierung«) unterschieden.

1.2 Forschungsfragen

Research Questions

Die Synchronisierung in der Batteriezellproduktion, nachdem die eigentliche Produktion stattgefunden hat (»Offline Zeitdatenreihensynchronisierung«) ist heute **nicht allgemein gelöst**, ist zugleich aber **in der Praxis relevant** für die Güte aller datenbasierten Verfahren. Die Offline Zeitdatenreihensynchronisierung ist eine komplexe und aufwändige Aufgabe, die zudem kontextuelles Wissen benötigt. Diese Aufgabe bestimmt maßgeblich die Güte nachfolgender Schritte in der Analyse und Modellierung. Dadurch besteht aktuell sowohl ein **Theorie- als auch ein Praxisdefizit** zur Lösung dieses Problems in der Domäne der Batteriezellfertigung.

Ziel dieser Arbeit ist der Entwurf und Erprobung eines neuen Ansatzes, wie das Problem der Offline Zeitdatenreihensynchronisierung zu lösen ist mit Blick auf die Anwendbarkeit und besondere Anforderungen in einer hochskaligen Batteriezellproduktion. Daraus ergeben sich für diese Arbeit eine Hauptforschungsfrage (HFF) und einige Unterforschungsfragen (UFF):

HFF Kann eine Offline Synchronisierung von Zeitdatenreihen aus der Batterieproduktion mit Wissen zum Produktionsaufbau automatisiert durchgeführt werden?

- UFF 1 Welche Bewertungsmetriken für eine Offline Synchronisierung können abgeleitet werden?
- UFF 2 Wie lässt sich die temporale Ungenauigkeit der Daten in der Batterieproduktion beschreiben und modellieren?
- UFF 3 Wie kann Wissen zum Aufbau und Prozessverständnis bei der Offline Synchronisierung genutzt werden?
- UFF 4 Lässt sich die Offline Synchronisierung von Zeitdatenreihen für hochskalige Produktionen automatisieren?

Um obige Hauptforschungsfrage zu beantworten, müssen die abgeleiteten Unterforschungsfragen betrachtet und beantwortet werden.

1.3 Struktur der Arbeit

Structure of Thesis

Diese Arbeit nähert sich der Problemlösung durch wissenschaftliche Methodik und ist entsprechend strukturiert. Wissenschaften können in verschiedene Klassen und Schwerpunkte unterteilt werden. Diese Systematik ist in Abbildung 1.3 dargestellt. Dabei wird als Erstes die Unterscheidung zwischen den Formal- und den Realwissenschaften unternommen. Die Formalwissenschaften befassen sich mit der Konstruktion von Sprachen und der Nutzung von Zeichensystemen und Regeln. Hierunter fallen Philosophie, Logik, Mathematik und die Wissenschaftslogik. Im Gegensatz dazu befassen sich die Realwissenschaften mit Beschreibungen, Erklärungen und Gestaltung empirischer Wirklichkeitsausschnitte. Dabei unterscheiden sich diese entlang ihrer Ziele. So adressieren »reine« Realwissenschaften theoretische Ziele durch Erklärung von empirischen Informationen, wie es in den Naturwissenschaften häufig anzutreffen ist. Bei praktischer Zielsetzung hingegen handelt es sich um »angewandte« Realwissenschaften. Diese adressieren die Handlungsalternativen und unterstützen bei der Entscheidungsfindung. Die Realwissenschaften können dabei über ihre Disziplinen verzahnt betrachtet werden. Einen besonderen Fall stellen hier die Ingenieurwissenschaften dar, da diese auf eben jene Verzahnung der Realwissenschaften aufsetzen (hellblau). Diese Arbeit ist daher in diesem Bereich auf der Seite mit einem praktischen Ziel angesiedelt (orange markiert). [ULRI76]

Diese Arbeit folgt in ihrer Struktur und ihrem Forschungsansatz den Schritten von ULRICH

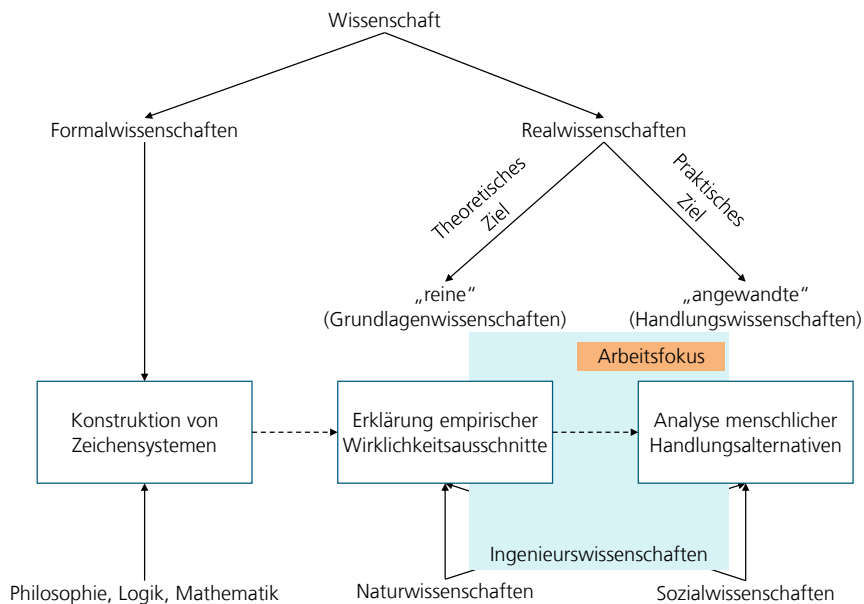


Abbildung 1.3: Die Wissenschaftssystematik nach ULRICH inkl. der Verankerung von Ingenieurwissenschaften und dem Fokus dieser Arbeit. (Basierend auf [ULRI76])

[ULRI81], zusätzlich ergänzt um Aspekte aus den Guidelines nach HEVNER [HEVN04]. Diese Anlehnung ist dargestellt in Abbildung 1.4. Im ersten Schritt wird das Problem beschrieben und die Relevanz in der Praxis hergeleitet. Dies geschieht in der Einleitung (Kapitel 1) und in den Grundlagen (Kapitel 2) dieser Arbeit. Als Nächstes werden die Theorien mit Bezug zu diesem Problem untersucht und im Kontext der Problemstellung interpretiert. Dies geschieht insbesondere in dem Kapitel zu Grundlagen und dem Stand der Forschung (Kapitel 3). Weiterhin werden die Verfahren aus den Formalwissenschaften eingebunden in dem Kontext des Problems untersucht. Dies geschieht in den Kapiteln zu Grundlagen und dem Stand der Forschung. Die Erfassung und Untersuchung des relevanten Zusammenhangs in der Anwendung geschieht einerseits in dem Stand der Forschung, bezogen auf aktuelle Forschungsentwicklungen, sowie im Kapitel 4 zur Entwicklung in Bezug auf die hier neu entwickelten Ansätze und Konzepte. Die Ableitung der Beurteilungskriterien und Modelle geschieht maßgeblich ebenfalls im Kapitel Entwicklung, wo ebenso die Anforderungen abgeleitet und darauf basierende Metriken entsprechend entworfen werden. Eine entsprechende Prüfung der Modelle und Regeln im Kontext der Anwendung findet im Kapitel Verifikation und Validierung (Kapitel 5) statt. Zuletzt wird der Schritt zu Beratung und Implementierung in der Praxis im Kapitel 6 adressiert. Die Guideline »Design as an

artifact« wird im Kapitel zu Entwicklung integriert, da sie methodisch die Entwicklung eines Artefakts, also eines Modells oder Softwarekomponente, adressiert, was wiederum den Anwendungscharakter der Arbeit unterstreicht. Die Guideline »Design Evaluation« beschreibt neben der Überprüfung der inhaltlichen Anwendbarkeit die Effizienz der Artefakte, sodass hier auch praktische Anwendbarkeit weiter fokussiert werden kann.



Abbildung 1.4: Der Forschungsansatz nach ULRICH. Rechts gegenüber ist dabei die Struktur dieser Arbeit dargestellt inklusive der Verlinkung entlang der Schritte auf der linken Seite. Zusätzlich in blau entnommene Guidelines aus [HEVN04]. (Basierend auf [ULRI81])

2 Grundlagen der Zeitdatenreihensynchronisierung

Background on Time-Series Synchronization

In diesem Kapitel werden die Grundlagen zu der Offline Zeitdatenreihensynchronisierung in der Batteriezellfertigung und damit verbundene Themen dargestellt. Zunächst wird dieses Thema in seiner Praxisrelevanz basierend auf Expert*innenbefragungen und Literatur eingeordnet und danach technisch betrachtet. Dies umfasst die Beschreibung und Formalisierung der Zeit, verteilte Netzwerke und die Batteriezellfertigung. Im Anschluss liegt der Schwerpunkt auf der Vorverarbeitung von Produktionsdaten und einer Analyse der Problemstellung mit Expert*innen. Zunächst werden alle Themen und Hintergründe einzeln dargestellt, abschließend werden diese Themen zueinander und in Bezug zur Problemstellung aus der Motivation eingeordnet.

2.1 Praxisrelevanz und Industriebefragung

Practical Relevance and Industry Survey

Das Thema der Synchronizität von Zeitdatenreihen in der Produktion ist wie dargestellt in Kapitel 1 relevant. Zur Einordnung, ob dieses Thema bereits als gelöst betrachtet werden kann, welche Probleme in dem Themenfeld aus Praxissicht existieren und welche Dimensionen hierzu üblich sind, wurden Expert*innenbefragungen durchgeführt. Diese Befragungen sind in zwei Teilen durchgeführt worden. Der erste Teil befasst sich hier mit den grundsätzlichen Herausforderungen der Umsetzung von datengetriebenen Methoden und der Anwender*innenperspektive. Die entsprechende Befragung zielt dabei auf eine Bandbreite von Aufgaben der entsprechenden Personen ab und nimmt die Synchronisierung von Daten als eine dieser Aufgaben auf. Dadurch kann eine Einschätzung zur Relevanz auch in Relation mit anderen Aufgaben getroffen werden. Der zweite Teil befasst sich spezifisch mit der Synchronisierung von Zeitdatenreihen in der Produktion. Hierbei werden die Relevanz des Themas in der Praxis und die Dimensionen der Probleme betrachtet. Die Ergebnisse dieser Befragungen sind in den folgenden Abschnitten dargestellt.

2.1.1 Data Science Arbeiten

Data Science Tasks

Um die Daten aus einer Produktion nutzbar zu machen und Erkenntnisse aus diesen Daten zu generieren, werden entsprechende Methoden und Schritte angewendet. Für die Durchführung dieser Arbeiten ist dabei das Berufsbild des »Data Scientist« entstanden. Der

Begriff selbst geht dabei bereits zurück bis in die 1970er Jahre [ES18]. Seit dieser Zeit haben sich Software, Berechnungsfähigkeiten der Hardware und insbesondere der Durchbruch von künstlicher Intelligenz maßgeblich auf dieses Berufsbild ausgewirkt und dieses verändert. In der Konsequenz ist die Definition einer Data Science Position nicht einheitlich und nicht unbedingt konsistent. Eine Stichprobe von Ausschreibungen in einem Jobnetzwerk ergab dabei, bezogen auf die technischen Aspekte, dass Aufgaben wie Software Entwicklung, künstliche Intelligenz und Statistik üblich sind. Darüber hinaus unterscheiden sich diese Ausschreibungen, sodass Aufgaben wie Datenaufnahme, Datenbankmanagement, Informations- und IT-Sicherheit, Visualisierung, Lehren, Deployment, Datenvorverarbeitung nur vereinzelt genannt werden. [SCHM24b]

Die oben dargestellte Stichprobe sowie große Umfragen von Anaconda [ANAC22, ANAC23], CrowdFlower [CROW16] und Kaggle [KAGG20] der vergangenen Jahre betrachten dabei Data Science in der Breite um branchenunabhängige Muster und Erkenntnisse zu gewinnen. Diese Umfragen legen dabei jeweils nahe, dass ein signifikanter Anteil der Arbeitszeit von Data Scientists für die Bereinigung und Vorverarbeitung von Daten aufgewendet wird. So beschreibt der CrowdFlower die entsprechenden Arbeiten als verantwortlich für etwa 60 % der Arbeitszeit. Eine spezifische Untersuchung im Bereich von Data Science im Produktionsbereich existiert allerdings bis dahin nicht, sodass eine spezifische Bewertung dieser Domäne erschwert ist. Aus diesem Grunde wurde eine spezifische Untersuchung durchgeführt und 2024 veröffentlicht.

Diese Untersuchung fand in mehreren Schritten statt. Im ersten Schritt (Vorstudie) wurde durch eine Reihe von Befragungen mit Expert*innen und Quellen wie BISWAS [BISW22] eine Auswahl von relevanten Arbeitsschritten für Data Scientists erstellt und im Bereich der Datenvorverarbeitung eine detaillierte Liste von Schritten abgeleitet. Diese sind vollständig im Anhang A.2 dargelegt. Aus dieser Vorstudie wurde eine strukturierte Hauptstudie entworfen und durchgeführt. Aufgrund der spezifischen Zielgruppe (»Data Scientists in der Produktionsdomäne«) wurde bei dem Stichprobenverfahren insbesondere auf das Schneeballverfahren zurückgegriffen, da es eine bessere Erreichung von spezifischen Gruppen ermöglicht [BIER81]. Der vollständige Fragebogen ist im Anhang A.3 verfügbar.

Grundsätzlich umfasst die Umfrage drei Kernfragen. Die erste Frage befasst sich mit dem Arbeitsaufwand der Data Scientists entlang der abgeleiteten Arbeitsschritte. Die zweite Frage nimmt die gleichen Arbeitsschritte auf und befasst sich mit der emotionalen Seite, dem Vergnügen dabei, diese Aufgaben auszuführen. Die letzte Kernfrage adressiert die Schritte der Datenvorverarbeitung und ihren relativen Umfang. Für die Umfrage konnten 89 Antworten gesammelt werden. Nach Entfernung von Antworten, die nicht aus der gesuchten Domäne stammten, blieben 81 Antworten übrig. Aufgrund der Stichproben-

verfahren und Art der Verteilung der Umfrage sind die Ergebnisse höchstwahrscheinlich primär auf den deutschen und europäischen Raum zu beziehen.

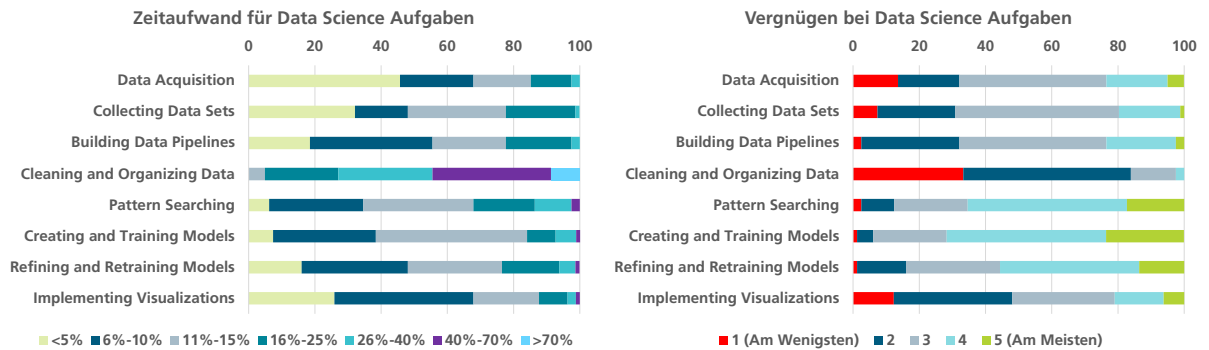


Abbildung 2.1: Ergebnisse der Fragen zu Arbeitsumfang (links) und Vergnügen (rechts) der Data Science Aufgaben aus der Vorstudie. Die Option *Sonstige* ist hier ausgelassen.

Für die ersten beiden Kernfragen sind die Ergebnisse in der Abbildung 2.1 dargestellt. Die Ergebnisse zeigen deutlich, dass die Bereinigung und Organisation von Daten einen dominanten Anteil der Data Science Aufgaben in der Praxis darstellt (linke Darstellung). Andere Aufgaben außerhalb der Modellierungsschritte wie Visualisierung und Datenaufnahme spielen bei den meisten Befragten kaum eine Rolle. Im Bereich Vergnügen (engl. »Joy«) der Aufgaben (rechte Auswertung) lässt sich erkennen, dass die Aufgaben im Kernbereich der Modellierung stärkeres Vergnügen bereiten als die Aufgaben außerhalb dieses Bereiches. So ist insbesondere die Visualisierung für fast 50 % eher keine Arbeit mit Vergnügen. Auch bei dieser Frage zeichnet sich ein dominantes Bild der Datenvorverarbeitung, welche über 80 % der Personen als keine Aufgabe mit Vergnügen beschreiben. Dies zeigt grundsätzlich eine hohe Relevanz der Datenvorverarbeitung in der praktischen Arbeit bei gleichzeitiger negativer emotionaler Einstellung zu dieser Aufgabe.

Für die dritte Kernfrage sind die Ergebnisse in der Abbildung 2.2 dargestellt. Hierbei handelt es sich um die relativen Arbeitsaufwände der Vorverarbeitungsschritte. Hieraus lässt sich ableiten, dass manche der Schritte, wie beispielsweise die Textbereinigung, bei den meisten Data Scientists in der Produktionsdomäne eine untergeordnete Rolle spielen. Die größten Zeitaufwände gehen demnach im Schnitt auf das Ausfüllen fehlender Werte, die Synchronisierung mehrerer Datenströme und die Befragung von Personen zur Bedeutung von Daten und Bezeichnungen. Der Punkt der Synchronisierung mehrerer Datenströme umfasst dabei insbesondere die Synchronisierung der Zeitdatenreihen, also das Hauptthema dieser Arbeit.

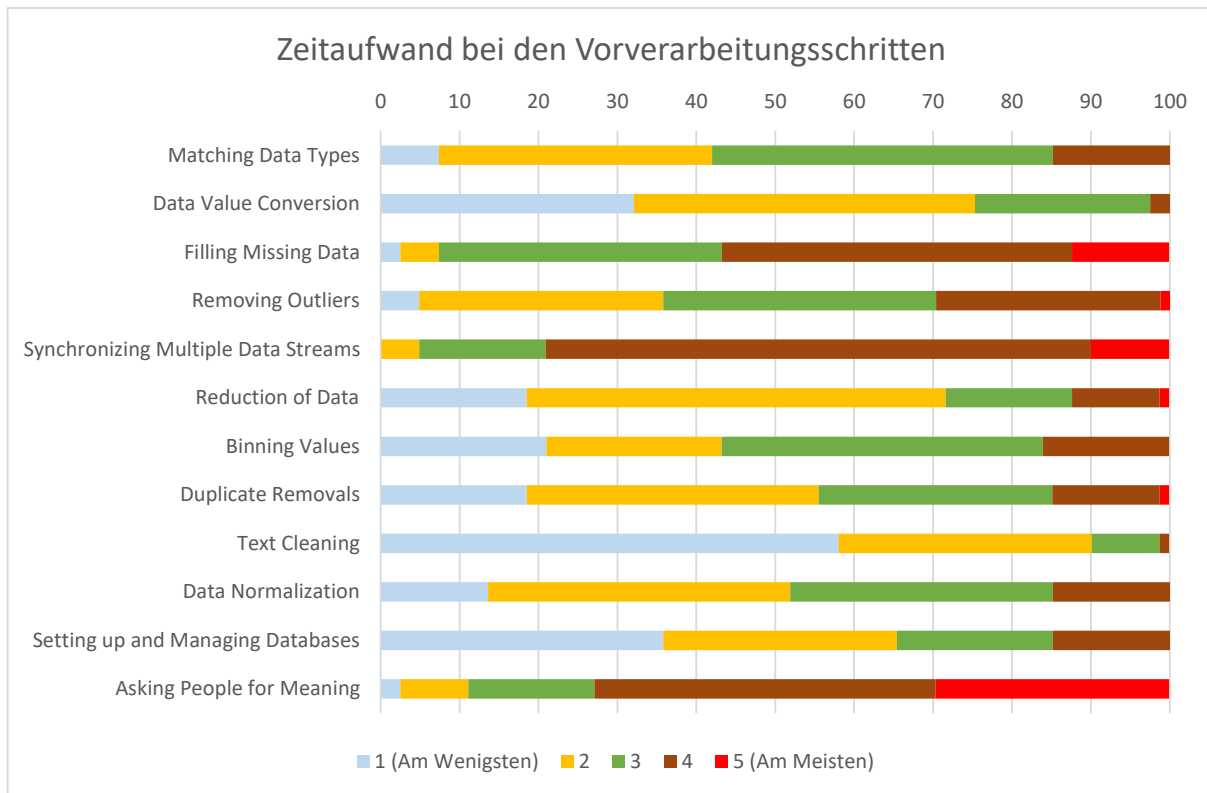


Abbildung 2.2: Ergebnisse der Fragen zum relativen Arbeitsumfang der Vorverarbeitungsschritte. Die Option *Sonstige* ist hier ausgelassen.

Aus diesen Kernfragen der Umfrage ergibt sich, dass für Data Scientists in der Domäne der Produktion die Datenvorverarbeitung den dominanten Teil der Arbeit darstellen kann und die Synchronisierung von Zeitdatenreihen, welche in dieser Arbeit betrachtet wird, darin eine signifikante Rolle einnimmt. Die detaillierten Werte der Auswertung befinden sich im Anhang A.4. Für weiterführende Interpretationen der Ergebnisse, die über den Fokus dieser Arbeit hinaus gehen, sei auf die entsprechende Publikation verwiesen [SCHM24b].

2.1.2 Synchronisierung in der Produktion

Synchronization in Production

In Ergänzung zu der Beschreibung in den vorherigen Abschnitten zur Synchronisierung wurde ebenfalls eine Industriebefragung von Expert*innen durchgeführt um die Relevanz des Themas zu validieren. Hierzu wurde eine spezifische Umfrage durchgeführt. Die detaillierte Befragung ist im Anhang A.5 dargestellt. Die Umfrage wurde analog zur obigen

Studie hauptsächlich im Schneeballverfahren verteilt und konnte am Ende 49 Antworten aggregieren.

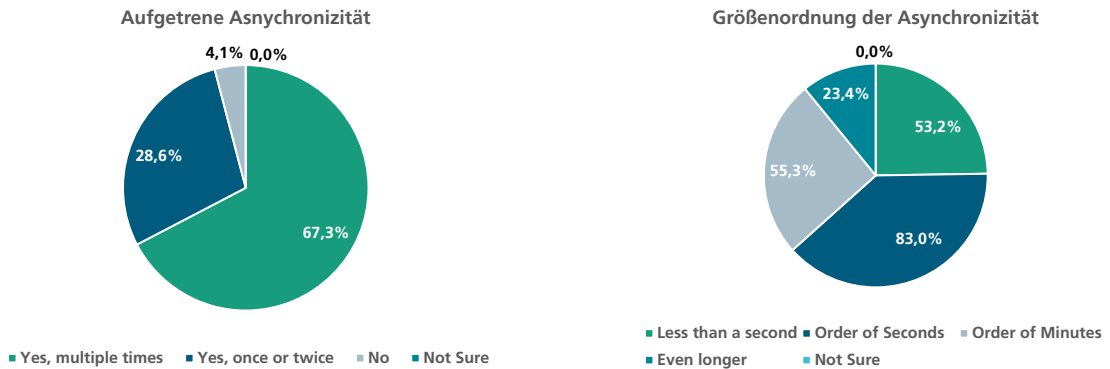


Abbildung 2.3: Ergebnisse der Fragen zum Auftreten und der Größenordnung der Asynchronizität in der Produktion. Die Option *Sonstige* ist hier ausgelassen. Bei der Frage zur Größenordnung war eine Mehrfachnennung möglich, sodass das Diagramm die relative Gewichtung anzeigt.

Bei dieser Umfrage stand das Problem der Asynchronizität im Vordergrund und Befragte gaben Auskunft über diesen Effekt in ihren Produktionen oder den Produktionen ihrer Kunden. Betreffend des Problems selbst sind die Ergebnisse in der Abbildung 2.3 dargestellt. Die Mehrheit der Personen aus der Befragung hat diesen Effekt in der eigenen Produktion mehrfach beobachtet (linke Darstellung). Bei der beobachteten Größenordnung (rechte Darstellung) hat die Mehrheit diesen Effekt in der Größenordnung von Sekunden bemerkt, zusätzlich wurden auch signifikante Anteile von Beobachtungen im Bereich unterhalb einer Sekunde und Minuten genannt. Diese Frage erlaubte Mehrfachauswahl, sodass die Summe der Werte 100% übersteigt.

Bezüglich der Adressierung der Asynchronizität sind die Ergebnisse der Befragungen in Abbildung 2.4 dargestellt. Die meisten Antworten zielen hier auf die Anpassung der Datenaufnahme und des Produktionsaufbaus im Betrieb ab (z.B. zentrale PLC oder regelmäßige Synchronisierung der Uhren). Weitere Methoden umfassen Behelfsmethoden im Betrieb, wie ein definiertes Ereignis zu Beginn zur Synchronisierung aller Signale am Start [MERZ23]. Dennoch verbleibt ein signifikanter Anteil für die manuelle Synchronisierung, nachdem die Daten aufgenommen wurden.

Ein weiterer Einblick durch diese Befragung ist die Menge an Geräten, von denen Daten kommen und deren Zeitdatenreihen synchronisiert werden müssen. So gaben etwa ein Viertel der Befragten an, dass nur zwei bis drei Geräte zu synchronisieren sind, ebenso

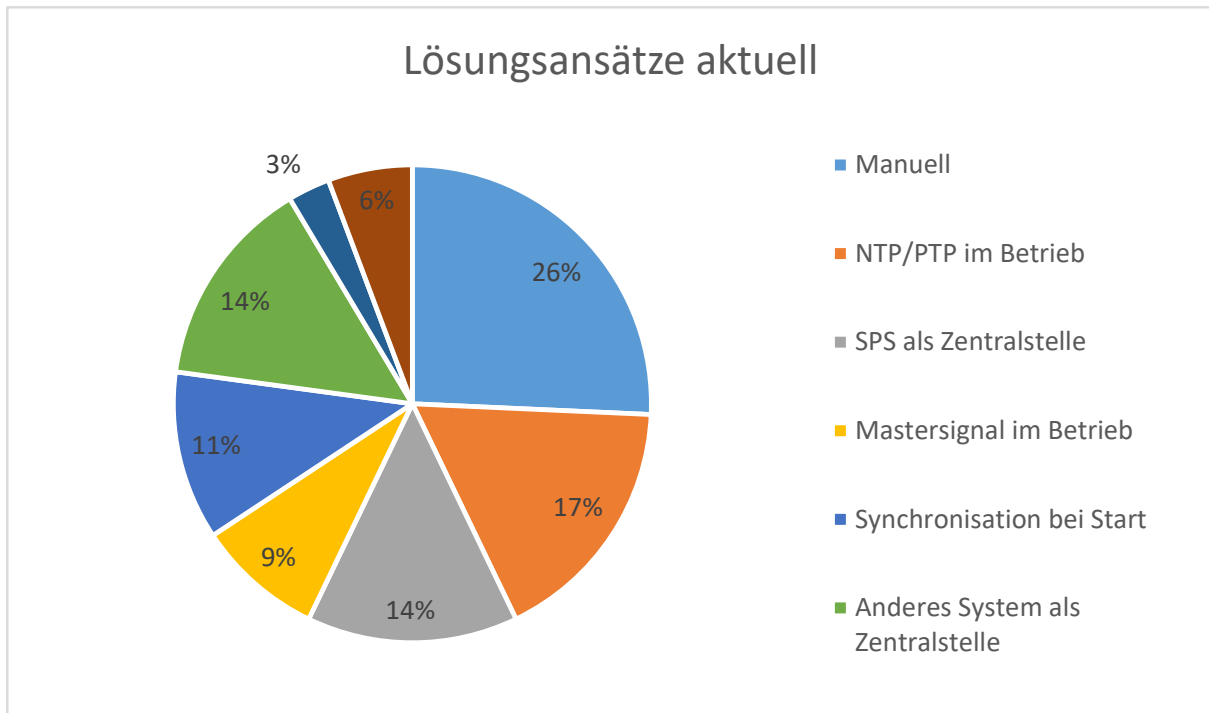


Abbildung 2.4: Ergebnisse der Frage zum Adressieren der Asynchronizität in der Produktion. Die Option *Sonstige* ist hier ausgelassen.

gaben allerdings über ein Fünftel an, dass es mehr als zehn Geräte seien. Daraus ergibt sich ein breites Feld in der Praxis von zu synchronisierenden Zeitdatenreihen und eine hohe Komplexität.

Insgesamt zeigt diese Befragung, dass das Problem der Synchronisierung nicht gelöst ist, und die Größenordnung des Problems in der Praxis oftmals in der Größenordnung von Sekunden und Minuten liegt. Übertragen auf eine Produktion von Batteriezellen kann dies beispielsweise eine falsche Zuordnungen von Elektroden oder kompletten Zellen bedeuten. Dadurch zeigt diese Befragung eine hohe Relevanz der Offline Zeitdatenreihensynchronisierung und die möglichen Größenordnungen, in denen sich der Fehler der Asynchronizität befindet. Weiterführende Ergebnisse und Einblicke befinden sich im Anhang A.6.

2.2 Zeitformalisierung und -messung

Time Formalization and Measurement

Diese Arbeit befasst sich mit Zeit und Formalisierungen. Daher beschreibt dieser Abschnitt zunächst eine abgrenzende Definition von Daten, Informationen und Wissen. Anschließend

folgt eine Beschreibung der Zeit, ihrer Formalisierung und damit verbundene Eigenschaften. Im Anschluss ist die technische Seite der Zeit und die daraus resultierende Definition von Zeitdatenreihen dargestellt. Der Fokus ist dabei auf den technischen Aspekten, die relevant für die Produktion sind, sodass weiterführende Themen und philosophische Sichtweisen ausgelassen sind.

2.2.1 Daten, Informationen und Wissen

Data, Information and Knowledge

Daten, Informationen und Wissen sind verschiedene Konstrukte, welche oftmals synonym benutzt werden und keine einheitliche Definition besitzen [LIEW07]. KELLEY beschreibt diese Begriffe als eng verbunden und nutzt diese zur gegenseitigen Definition. Dabei sind Daten grob und ungefiltert, während Informationen aufbereitet und erstmals nutzbar für Analysen sind [KELL02]. Das Wissen wiederum befindet sich im Menschen und entsteht durch Anwenden der eigenen Erfahrungen auf Daten und Informationen. Der Übergang von einem Datum zur Information ist also die Nutzbarkeit für weiterführende Schritte. Eine andere Definition von DAVENPORT beschreibt Daten als diskrete, objektive Fakten, die zudem strukturiert vorliegen. Informa-

tionen hingegen sind Nachrichten, die in der Kommunikation genutzt werden um Entscheidungen und Bewertungen durch Personen zu beeinflussen. Zuletzt stellt Wissen eine Mischung aus Erfahrungen, Werten und Kontexten dar, welche die Grundlage für Bewertungen durch einen Menschen bilden [DAVE98]. Der Übergang von Daten zu Informationen ist hier also durch die mögliche Auswirkung auf die Entscheidungsfindung definiert. Eine weitere bekannte Definition ist die kompakte Definition durch ACKOFF, welche die Begriffe mit der Wahrnehmung verbindet. So sind Daten lediglich als Symbole definiert und werden verarbeitet, bis Fragen zu »Wer, Was, Wo« und »Wann« gestellt und beantwortet werden können. Die Beantwortung der weiterführenden Frage »Wie« stellt dabei das Wissen dar [ACKO89]. Der Übergang von Daten zu Informationen ist also an die Nutzbarkeit für die

Definition

Daten sind computerisierte Repräsentationen von Modellen und Attributen von realen oder simulierten Entitäten.

Informationen sind Daten, die das Ergebnis eines Berechnungsprozesses darstellen, um Bedeutung zu den Daten zu geben.

Wissen sind Daten, die das Ergebnis eines kognitiven Prozesses (Wahrnehmung, Lernen, Assoziation, Begründung) oder Wissen eines Menschen darstellen.

Beantwortung von Fragen gekoppelt. Eine technischere Definition von CHEN in Daten, Informationen und Wissen ist in der Definitionsbox dargestellt [CHEN08] und wird im Verlauf dieser Arbeit, wenn nicht anders angegeben, entsprechend verwendet.

Daten existieren in verschiedenen Formen. Eine Klassifizierung für die Verarbeitung ist hier die Aufteilung nach strukturierten und unstrukturierten Daten möglich. Strukturierte Daten haben eine feste vorgegebene Struktur und können aufgrund ihrer Struktur adressiert werden. Dies umfasst Tabellen und Datenbanken. Dem gegenüber stehen unstrukturierte Daten, welche regelfreie Texte und Bilddaten umfassen können. Für alle Daten, die sich nicht eindeutig in diese Unterscheidung einordnen lassen, existiert der Bereich der semistrukturierten Daten. Schätzungen gehen davon aus, dass der Großteil der Daten in Unternehmen im Bereich der unstrukturierten Daten anzusiedeln ist [ALWA20]. Dies ist für die Analyse von Daten problematisch, da die Analyse und Verarbeitung von unstrukturierten und semistrukturierten Daten komplexer ist und Herausforderungen mit sich bringt. Der erste Schritt ist daher üblicherweise, diese Daten zu strukturieren [ZAMA20]. Dabei wird eine mathematische Struktur definiert und auf die extrahierten Daten angewendet, sodass eine strukturierte Datenspeicherung, beispielsweise innerhalb eines Datenbankmanagementsystems möglich ist [MAGN04].

2.2.2 Formalisierung der Zeit

Time Formalization

Zeit ist die physikalische Größe, welche üblicherweise als Abfolge von Ereignissen beschrieben ist. Die älteste bekannte Formalisierung der Zeit geht auf NEWTON zurück, welcher die Zeit als gleichförmig und unabhängig jedweder anderer Gegenstände und Größen definierte [NEWT87]. In dieser Definition ergibt sich die Eigenschaft der Monotonie der Zeit. Weiterhin leiten sich die Eigenschaften der absoluten Zeit und der Unabhängigkeit ab. Spätere Forschungsergebnisse wie die spezielle Relativitätstheorie [EINS05] zeigten, dass diese absolute Zeit und Unabhängigkeit nicht korrekt sind, und postulierten Phänomene wie die Relativität der Gleichzeitigkeit. In der Praxis für diese Arbeit sind die Auswirkungen aber vernachlässigbar, sodass die Definition nach Newton genutzt werden kann, sofern nicht anders angegeben.

Im internationalen Einheitensystem SI ist die Zeit durch zwei Basisgrößen bestimmt. Die verbreitetste Einheit ist dabei die Sekunde. In der Vergangenheit wurde diese aus der Erdrotation abgeleitet, jedoch war die Annahme der Gleichmäßigkeit der Erdrotation falsch, sodass die Definition mehrfach überarbeitet wurde. Dabei bildeten bis 1967 astronomische Beobachtungen die Grundlage der Definition. Im Jahre 1967 wurde die heute

gültige Definition der Sekunde eingeführt. Sie basiert auf einem festgelegten Vielfachen (9.192.631.770) der Periode einer elektromagnetischen Welle, die aus dem Niveauübergang eines Caesiumatoms (^{133}Cs Nuklid) resultiert. Formal ergibt sich damit die Definition

$$1\text{ s} = \frac{9.192.631.770}{\Delta_{Cs}},$$

wobei $\Delta_{Cs} = 9.192.631.770\text{ Hz}$ eben jene Periodendauer des Atoms beschreibt. Dadurch ist die Zeit aktuell nicht über astronomische Beobachtungen, sondern über Eigenschaften der Atome und damit physikalischer Basisgrößen definiert [BIDP19]. Aufgrund von technischen Weiterentwicklungen und höherer Präzision der Uhren ist eine präzisere Definition der Sekunde im Jahr 2030 möglich [BURE22]. Eine weitere Definition im Bereich der Zeit ist die Planck-Zeit, welche die minimale Zeitspanne beschreibt, bei der physikalische Gesetze und Theorien noch gelten. Diese Spanne liegt umgerechnet bei ca. $5,4 \cdot 10^{-44}\text{ s}$, allerdings mit einer relativ hohen Ungenauigkeit [NIST18]. In dieser Arbeit wird die aktuelle Definition der Sekunde genutzt, die Planck-Zeit ist weit unterhalb der betrachteten Zeitspannen.

Da die Erdrotation unregelmäßig Abweichungen aufweist, stimmt die Definition der Sekunde nicht perfekt mit der Rotation überein. Als Konsequenz daraus wurden Ausgleichsverfahren eingeführt. Diese umfassen insbesondere Schalttage und Schaltsekunden.

Für die Koordination der Zeit existieren gesetzliche Vorgaben und Definitionen sowie eine Union zur Koordination. In Deutschland ist dabei die Zeit der Physikalisch-Technischen Bundesanstalt als Referenz gesetzlich geregelt [BUND16a]. Das deutsche Gesetz stellt dabei auch die Verbindung zur Koordinierten Weltzeit (UTC) dar. Diese wird durch das Internationale Büro für Maß und Gewicht (BIPM) koordiniert. Zur Koordination gehört auch die Festlegung oben beschriebener Schaltsekunden.

2.2.3 Uhren in technischen Systemen

Clocks in Technical Systems

Die Zeit ist formal definiert und beschrieben, muss zugleich aber auch gemessen werden. Zur Messung von Zeitspannen werden Zeitmessgeräte verwendet. Die ersten technischen Geräte gehen dabei auf Sumerer und Ägypter zurück, welche mit Sonnenuhren arbeiteten. Zusammen mit dem Zahlensystem der Babylonier wurden Tage in zwölf Stunden unterteilt. Ein System, welches heute abgewandelt die Basis des 24 Stunden Tages, der 60 minütigen Stunde und der 60-sekündigen Minute ist. Diese Zeitmesser waren von astronomischen Beobachtungen abhängig und wurden ab dem zweiten Jahrtausend vor Christus durch

Wasseruhren ersetzt. In Europa und Asien wurden genormte Zeitmessgeräte wie gleich gehaltene Kerzen und Sanduhren eingesetzt [ROSS96]. Derartige Zeitmesser sind heute üblicherweise nicht mehr anzutreffen, abgesehen vom benannten Zahlssystem.

Anfang des 14. Jahrhunderts erhielten die Räderuhren als Zeitmesser Einzug in Europa. Diese Art der Uhr umfasst eine Trennung von Anzeige (beispielsweise Uhrblatt mit Zeiger) und dem Uhrwerk im Hintergrund. Das Uhrwerk ist dabei der Mechanismus in diesem Zeitmesser, welcher die Zeit nachhält. Räderuhren und ihre Nachfolger setzen dabei auf mechanische Uhrwerke, welche aus dem Gehwerk und einem Antrieb bestehen. Der Antrieb, durch Gewichte oder Federn, übt eine mechanische Kraft auf das Gehwerk aus. Das Gehwerk hat ein Schwingssystem und eine Hemmung, welche dazu führen, dass in regelmäßigen Abständen eine Bewegung ausgelöst wird, die dann auf die Anzeigevorrichtung übertragen werden kann. So können beispielsweise Antrieb und Schwingssystem so eingestellt werden, dass sekundlich eine Bewegung der Anzeige stattfindet [STOL95]. Dieses Grundkonzept der Trennung nach Uhrwerk und Anzeige sowie Antriebe ist bis heute in Zeitmessern zu finden. So ist eine mechanische Wand- oder Armbanduhr eine solche Räderuhr. Wichtig ist hierbei, dass die Räderuhren kontinuierlich die Zeit durch die Schwingungen zählen, jedoch keine absolute Zeitinformation besitzen. Stattdessen ergibt sich die Zeit als Menge der Schwingungen nach einem festgelegten Zeitpunkt, wie dem letzten Stellen der Uhr.

Mit Aufkommen der Elektrizität sind auch elektrische Zeitmesser entstanden. Der erste Schritt hierbei ist die Erweiterung der Räderuhr um einen elektrischen Taktgeber. Der Antrieb und das Schwingssystem, bisher mechanisch, werden ersetzt durch eine elektronische Komponente, welche regelmäßig eine Schwingung auslöst, sodass der Rest der Räderuhr gleich bleibt. Für diese Komponente wird der piezoelektrische Effekt [MART72] und die Stabilisierung eines Schwingquarzes genutzt, um bei vorhandener Stromversorgung diesen regelmäßigen Takt auszulösen. Derartige elektromechanische Uhren fanden ab dem frühen 20. Jahrhundert zunehmende Verbreitung [VIG92]. Eine Weiterentwicklung dieser Uhr ist die Entfernung der mechanischen Komponenten aus dem Uhrwerk. Anstatt über den Taktgeber eine mechanische Schwingung auszulösen, wird direkt eine elektrische Anzeige betrieben [LOMB11]. Derartige elektromechanische und vollelektrische Quarzuhren sind bis heute im Einsatz. Das Uhrwerk, welches vollelektrische Uhren antreibt, wird daher auch »Quarzwerk« bezeichnet.

Eine weitere Variante der elektrischen Zeitmessung ist die Synchronuhr. Diese nutzt den Zugang zum Stromnetz mit einer bekannten Netzfrequenz als Taktgeber anstatt eines Quarzkristalls [SCHE19]. Diese Art von Uhren findet sich oftmals in Küchengeräten und ist direkt abhängig von der Stabilität der Netzfrequenz. Diese Stabilität ist allerdings nicht

dauerhaft gewährleistet, sodass Schwankungen wahrscheinlich sind. Dadurch sind die Uhren in der Konsequenz ungenauer, je mehr die Netzfrequenz schwankt. Der resultierende Effekt ist in Abbildung 2.5 dargestellt. Anfang 2018 sank die Netzfrequenz so stark ab, dass die kumulierte Abweichung für Synchronuhren mehr als 5 Minuten betrug. Synchronuhren sind damit als präzise Zeitgeber stark von externen Faktoren abhängig und nicht nutzbar.

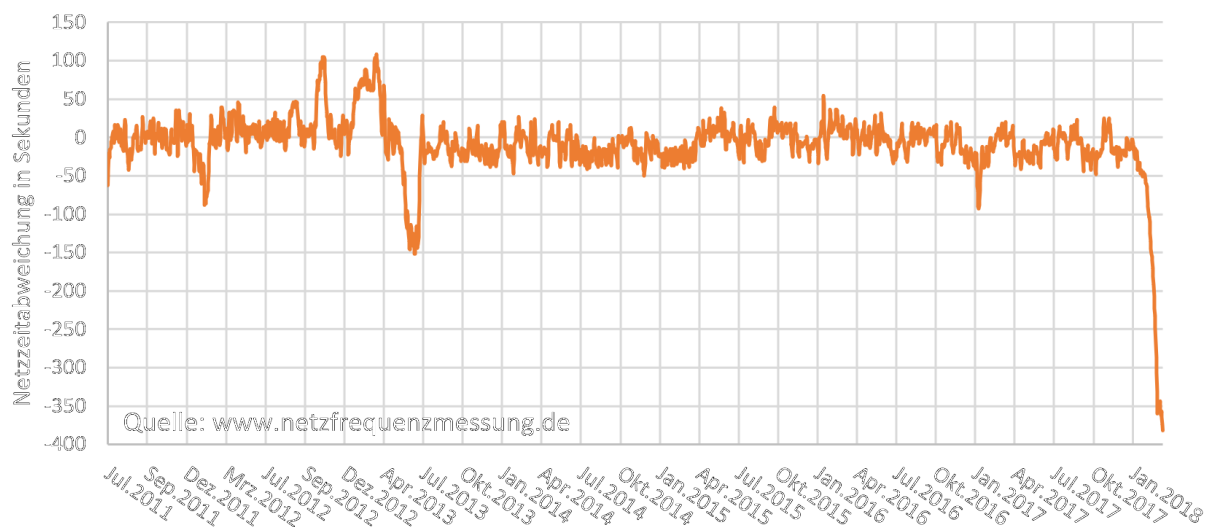


Abbildung 2.5: Die Abweichung der Zeiten von Synchronuhren resultierend aus den Abweichungen der Netzfrequenz (Netzzeitabweichung). (Entnommen aus [NETZ18])

Die aktuell präzisesten Zeitmessgeräte sind sogenannte Atomuhren. Hierbei ist der eigentliche Taktgeber mehrteilig aufgebaut. Ein Quarzoszillator wird genutzt, um ein elektromagnetisches Wechselfeld zu erzeugen. Innerhalb dieses Feldes befinden sich spezifische Atome (z.B. ^{133}Cs). Die eingestellte Frequenz am Quarzoszillator regt jene Atome an und diese strahlen Energie ab. Mittels einer Regelschleife wird die Frequenz des Quarzoszillators nachgeregelt, sollte die abgestrahlte Energie der Atome geringer werden. Dieser Regelkreis erlaubt die hochpräzise Regelung und hohe Stabilität des Quarzoszillators. Dieser wird zugleich als Taktgeber des Zeitmessers verwendet [BAND24]. Neuartige Ansätze für Atomuhren setzen auf alternative Atome, optische Methoden [LUDL15] oder Rubidium anstatt eines Quarzoszillators [CAMP07].

Die dargestellten modernen Arten von Zeitmessern unterliegen Annahmen, um ihre Genauigkeit und Korrektheit sicherzustellen. Synchronuhren sind direkt abhängig von der Netzfrequenz und Quarzuhren sind abhängig von der Stabilität der Frequenz des Kristalls. Diese Stabilität wiederum ist abhängig von Materialeigenschaften, mechanischen Einwirkungen, Temperatur, Alterung und weiteren Einflüssen [HORO89]. Eine verbreitete

Gegenmaßnahme bezüglich der Einflüsse durch die Temperatur sind Quarzöfen (engl. »Oven Controlled Crystal Oscillator OCXO«), welche den Kristall in einer geregelten Temperaturkammer thermisch stabil halten, um die Abweichung zu verringern [XU23]. Trotz der verschiedenen Gegenmaßnahmen sind Abweichungen bei diesen Oszillatoren nicht zu vermeiden, sodass diese nicht als ideal und stabil angenommen werden können [ZHOU08].

Zur Darstellung der Zeit selbst existieren darauf aufbauend Standards zur Beschreibung von Zeiten. Die DIN ISO 8601 beschreibt dabei explizit die Datums- und Zeitformate, die international kompatibel anzuwenden sind. Ein Teil definiert dabei die Darstellung von Daten (Jahre, Monate, Tage), ein anderer Teil die Tageszeit, welche in beliebiger Genauigkeit als Sekundenbruchteile dargestellt werden kann. Ebenso umfasst dieser Standard Regeln für Zeitzonen und Notationen für Zeitspannen [TECH19b]. Konkrete Zeit- und Zeitspannenangaben in dieser Arbeit folgen diesem Standard, wenn nicht anders angegeben.

Computerbasierte Systeme

Mit der durchdringenden Computerisierung muss die Zeit auch für Computersysteme definiert und nachgehalten werden. Moderne Computersysteme bestehen dabei aus zentralen Prozessoren und hochkomplexen Schaltkreisen, die mehrere Milliarden Transistoren umfassen und deren Komplexität nach *Moore's Law* weiter ansteigen wird [INTE22]. Diese zentralen Prozessoren (engl. »Central Processing Unit CPU«) arbeiten in Zyklen. Ein Taktgeber löst dabei regelmäßig einen solchen Zyklus und damit die Rechenwerke in der CPU aus. Die Frequenzen der Quarzoszillatoren liegen je nach Kristallschnitt und Schwingungsform im Bereich von kHz bis MHz [NEUB97]. Moderne CPUs benötigen höhere Frequenzen, daher ist hinter dem Quarzoszillator noch eine Phasenregelschleife eingefügt [HSIE96, GREB69]. Diese Regelschleife erlaubt eine Synthese höherer Frequenzen als Taktgeber.

Um nun die Zeit darzustellen, setzen Computersysteme auf diesen Taktgeber und zählen regelmäßig die Zyklen. Die Information über die Frequenz des Taktgebers f zusammen mit der Anzahl von gezählten Zyklen n erlaubt so eine fortschreitende Beobachtung der Zeit. Um die Zeit nicht nur relativ zueinander zu beschreiben, wird durch das Betriebssystem ein Referenzzeitpunkt definiert. Bei Unix-Systemen ist das die sogenannte »Unix-Zeit«. Diese markiert den Beginn der Zählung ($n = 0$) als den 01.01.1970, 00:00:00 Uhr [LINU24]. Andere Systeme und Standards können abweichende Startpunkte haben. So haben aktuelle Windows-Systeme den 01.01.1601 [MICR24] und im Standard ISO 8601 ist der 00.01.0001 v.Chr. (»0. Januar im Jahre 1 vor Christus« – gleichbedeutend mit dem Jahr null, da die Zählung vom Jahr 1 vor Christus auf Jahr 1 nach Christus springt) als Nullpunkt defi-

niert [TECH19b]. Die ideale Berechnung der aktuellen Zeit T_{jetzt} in einem Computersystem ergibt sich damit zunächst zu

$$T_{jetzt} = T_0 + \frac{n}{f}, \quad (2.1)$$

wobei T_0 eben jener Nullpunkt des Systems ist.

In der Praxis kommt es hierbei zu Abweichungen von der tatsächlichen Zeit. Dadurch ist eine Unterscheidung der echten Zeit und der Zeit im betrachteten System notwendig. Im Folgenden ist daher die Notation T_α genutzt, um einen Zeitpunkt α in der echten Zeit zu bezeichnen. Ein betrachtetes System s ist als C_s beschrieben. Die Systemzeit, errechnet in eben jenem System, ist dargestellt durch

$$C_s(t_x) = T_0 + \frac{x}{f}, \quad (2.2)$$

wobei t_x den Moment bezeichnet, in dem der x -te Zyklus des Taktgebers abgeschlossen und registriert wurde. Zum Vergleich mit der echten Zeit beschreibt $C_s(T_\alpha)$ die Systemzeit in C_s zum echten Zeitpunkt T_α [SCHM22].

Gangfehler und Taktversatz

Wie beschrieben, kommt es in der Praxis zu Abweichungen von Systemzeit und echter Zeit. Hauptursache dafür ist die oben beschriebene Abweichung des Taktgebers von der Idealfrequenz. Dies führt maßgeblich zu zwei Effekten beim Betrachten und Vergleichen der Zeiten. Es handelt sich dabei um den **Taktversatz** (engl. »Clock Skew«) und **Gangfehler** (engl. »Clock Drift«). [LATH10]

Der **Taktversatz** beschreibt einen Effekt zwischen mehreren synchronen Systemen (z.B. Schaltkreise oder Computersysteme). Im Idealzustand erhalten alle Systeme ein Event (z.B. eine Signaländerung oder Nachricht) zur selben Zeit [FRIE01]. In der Praxis allerdings kann dies verzögert zueinander auftreten, beispielsweise aufgrund von anderen Zyklen beim Abtasten von Informationen oder anderem Offset. Dies ist exemplarisch in Abbildung 2.6 dargestellt. Ein Eingangssignal (braun) soll von zwei Systemen C_1 und C_2 detektiert werden. Eine Detektion kann nur zyklensweise stattfinden, mit steigenden Flanken bei den Signalen der Taktgeber in den Systemen. Die Zyklen der Systeme sind synchron, in dem Sinne,

dass die Länge der Zyklen exakt gleich ist. Allerdings sind diese Zyklen statisch versetzt zueinander entlang der Zeitachse. Dies führt dazu, dass das System C_1 das Signal früher als C_2 detektiert. Dieser Versatz der Detektion zwischen den Systemen, ausgelöst durch den Versatz der Taktgeber ist der Taktversatz und ist beschrieben als Δ_{skew} mit der Einheit Sekunden.

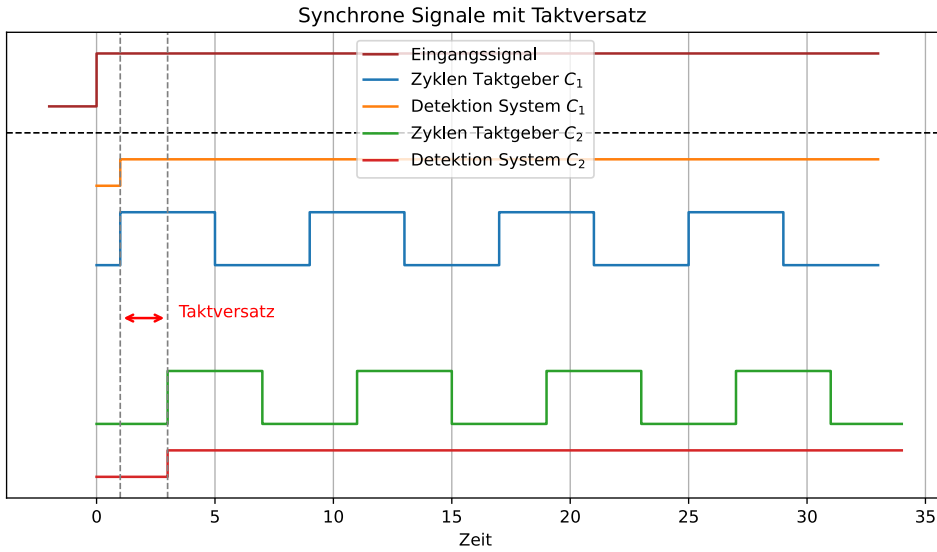


Abbildung 2.6: Der Taktversatz am Beispiel zweier synchroner Systeme, welche ein Eingangssignal (braun, oben) versetzt zueinander detektieren.

Das Gegenstück zum Taktversatz ist der **Gangfehler**, exemplarisch dargestellt in Abbildung 2.7. Während bei der Betrachtung des Taktversatzes eine ideale und identische Zykluszeit bei allen Systemen angenommen wurde, beschreibt der Gangfehler Abweichungen bei den Zykluszeiten. In dem Beispiel ist erkennbar, dass die Zyklen im Betrachtungszeitraum beim System C_2 konstant bleiben. Beim System C_1 hingegen sind die markierten Zyklen erst verlängert, dann verkürzt. Dadurch kommt es zu Verschiebungen bei Detektionen zusätzlich zum benannten Taktversatz. Der Gangfehler ist beschrieben als δ mit der Einheit Sekunde durch Sekunde (Der Fehler in Sekunden auf Sekunden der echten Zeit).

Die Effekte Taktversatz und Gangfehler haben einen direkten Einfluss auf die Systemzeit in den Computersystemen. Ist eine ideale Systemuhr in System C_1 vorhanden und das System C_2 hat einen Taktversatz, dann kann für einen festen Zeitpunkt in der echten Zeit T_α der Taktversatz dargestellt werden als

$$\Delta_{skew} = C_1(T_\alpha) - C_2(T_\alpha) = C_1(t_0) - C_2(t_0), \quad (2.3)$$

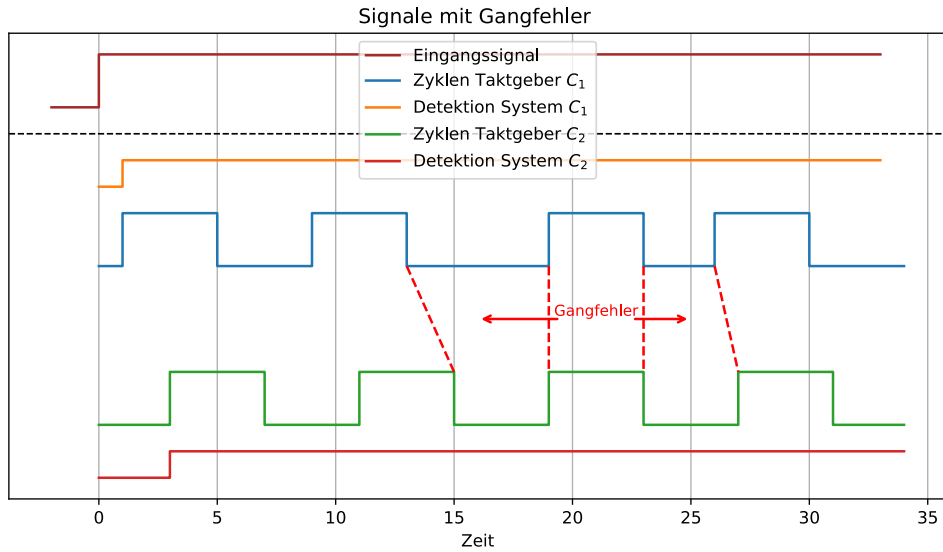


Abbildung 2.7: Der Gangfehler am Beispiel zweier Systeme, deren Zyklen sich dadurch unterscheiden.

wobei $C_m(t_0)$ die Systemzeit zu Beginn der Beobachtung, also dem nullten Zyklus t_0 innerhalb der Beobachtung bezeichnet.

Für den Gangfehler kann zunächst die Zykluszeit eines idealen (stabilen) Taktgebers als τ_{ideal} beschrieben werden. Dem gegenüber steht die reale Zykluszeit τ_{real} , welche größer oder kleiner sein kann. Zur Bestimmung des Gangfehlers δ kann dann die Formel

$$\tau_{real} = \frac{\tau_{ideal}}{1 - \delta} \quad (2.4)$$

genutzt werden. Anders ausgedrückt ergibt sich für den Gangfehler $\delta = 1 - \frac{\tau_{ideal}}{\tau_{real}}$. Unter der Annahme, dass kein Systemausfall stattfindet kann dadurch auch die Annahme getroffen werden, dass $-1 < \delta < 1$ gilt. Dies bedeutet, dass der Gangfehler stets kleiner als die Zykluszeit ist.

Obige Definition beschreibt dabei den Gangfehler δ für einen spezifischen Moment (Zyklus). Für die Betrachtung über eine Zeitspanne von m Zyklen ist die Definition erweitert zu

$$\delta(i) = 1 - \frac{\tau_{ideal}}{\tau_{real}(i, i+1)}, 0 \leq i < m. \quad (2.5)$$

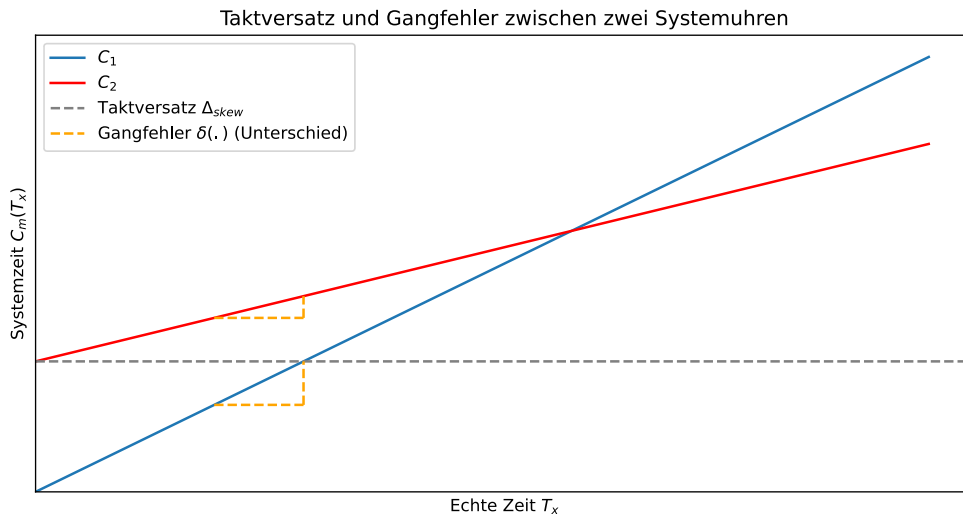


Abbildung 2.8: Die Systemzeit zweier Systemuhren sowie die Effekte von Gangfehler und Taktversatz über die Zeit dargestellt. (Basierend auf [SCHM22])

Die kombinierten Effekte auf Systemzeiten sind exemplarisch in Abbildung 2.8 aufgezeigt. Die Systemzeiten von C_1 und C_2 , $C_1(T_x)$ bzw. $C_2(T_x)$, steigen über die echte Zeit monoton an. In dieser Darstellung sei vereinfachend auch eine lineare Steigung angenommen. Zum Startpunkt der Beobachtung unterscheiden sich die Systemzeiten. Dies entspricht dem Taktversatz der beiden Systemuhren Δ_{skew} . Der lineare Anstieg der Systemzeit geschieht in C_1 schneller als in C_2 . Dies bedeutet, dass diese Systemuhren einen Gangfehler zueinander aufweisen, erkennbar an den verschiedenen Steigungen in der Darstellung. Dies hat zur Folge, dass bei gleicher echten Zeit die Systemzeiten der Systeme C_1 und C_2 nicht garantiert übereinstimmen. [SCHM22]

Die hier gewählte Darstellung einer streng monotonen, linearen Funktion für die Systemzeit muss in der Praxis nicht unbedingt gegeben sein. So kann auch der Gangfehler selbst über die Zeit variieren [HAN22].

2.2.4 Zeitdatenreihen

Time-Series Data

Gemäß der Definitionen und Formalisierungen von Zeit in den vorherigen Abschnitten, sollen nun neben der Zeit selbst auch Daten strukturiert betrachtet werden. In dieser Arbeit ist die Definition angelehnt an ESLING mit spezifischen Erweiterungen, um die Darstellung

der Zeit selbst zu berücksichtigen [ESL12]. Ein einzelnes Datum v_i umfasst zunächst einen Wert, der, sofern nicht anders angegeben, in dieser Arbeit als reelle Zahl ausgedrückt wird ($v_i \in \mathbb{R}$). Eine Serie von Daten lässt sich nun als Sequenz darstellen:

$$V = (v_1, \dots, v_n), v_i \in \mathbb{R} \quad (2.6)$$

$$U = (v_k, v_{k+1}, \dots, v_{k+m-1}). \quad (2.7)$$

Hierbei stellt V die gesamte Serie von Daten dar, während U eine Subsequenz daraus beschreibt. Daher gilt $1 \leq k \leq n - m + 1$. Zur vereinfachten Darstellung kann eine solche Serie als $U_V^{k,m}$ beschrieben werden. Für ein V mit n Daten, gilt die Schreibweise $|V| = n$.

Um auch Zeitangaben aufzunehmen, werden die Daten v_i erweitert. Wie in Abschnitt 2.2.2 gezeigt, kann Zeit auf verschiedene Weisen dargestellt werden. Für die Beschreibung hier wird die Angabe der Zeit als rationale Zahl t_i genutzt. Eine textuelle Darstellung gemäß ISO 8601 kann entlang der dortigen Definitionen und der Kenntnis der Nullstelle direkt umgerechnet werden [TECH19b]. Dadurch ergibt sich für die Definition der Zeitreihen

$$Z = (z_1, \dots, z_n), z_i \in \mathbb{Q} \times \mathbb{R} \quad (2.8)$$

$$\overset{\circ}{U} = (z_k, z_{k+1}, \dots, z_{k+m-1}) \quad (2.9)$$

$$z_i = (t_i, v_i), \quad (2.10)$$

wobei nun die Sequenz Z und Subsequenz $\overset{\circ}{U}$ für die Zeitreihe analog zu den Datenreihen 2.6 und 2.7 zusammenhängen.

Wird nun mehr als eine Datenreihe betrachtet, bestehen zwei Möglichkeiten zur Darstellung. Eine Variante ist die Erweiterung der Zeitreihenpunkte z_i um jeweils eine Dimension. Für die Darstellung von l Datenpunkten ändert sich damit die Definition zu $z_i \in \mathbb{Q} \times \mathbb{R}^l$. Allerdings ergibt sich dadurch das Problem der Darstellung von fehlenden Werten. Wenn zum Zeitpunkt t_i von den l theoretischen Datenpunkten nur $l - 1$ existieren, ist die Konsistenz nicht mehr sichergestellt und fehlende Datenpunkte müssen in der Nutzung stets berücksichtigt werden. Dieses Problem wird in dem Abschnitt 2.6 genauer beleuchtet. Um in der Definition der Zeitreihen zunächst Konsistenz zu halten, wird daher die zweite Option der Modellierung genutzt. Hierbei wird pro Datenreihe eine eigene Sequenz als Zeitdatenreihe erstellt. Analog zu den obigen Definitionen ergibt sich damit

$${}_lZ = (z_1^l, \dots, z_n^l) \in \mathbb{Q} \times \mathbb{R} \quad (2.11)$$

$${}_l\overset{\circ}{U} = (z_k^l, z_{k+1}^l, \dots, z_{k+m-1}^l) \quad (2.12)$$

$$z_i^l = (t_i, v_i^l) \quad (2.13)$$

für die Zeitdatenreihe mit Bezug zur Datenreihe ${}_lV$. Die Gesamtheit aller Zeitdatenreihen wird als $\mathfrak{Z} = \{{}_lZ \mid \exists {}_lV\}$ bezeichnet. An dieser Stelle sei erwähnt, dass in der deutschen Sprache der Begriff »Zeitdatenreihe« und seine Variationen üblich sind, jedoch streng mathematisch die Definitionen keine Reihen, sondern Folgen beschreiben. Im Verlauf dieser Arbeit wird hier die übliche Bezeichnung der Zeitdatenreihe verwendet.

Darstellung von Zeitdatenreihen

Zur grafischen Darstellung und Erhöhung der Lesbarkeit bestehen verschiedene Möglichkeiten. Die üblichen Darstellungen umfassen dabei eine tabellarische Darstellung, die Nutzung von Diagrammen und die Nutzung der Graphentheorie. Diese Möglichkeiten sind in Abbildung 2.9 dargestellt.

In der tabellarischen Darstellung sind entlang der ersten Spalte (»Zeit«) alle bekannten Zeitwerte aufgenommen. In den folgenden Spalten sind die Werte aller Zeitdatenreihen eingetragen. Existiert in einer Zeitdatenreihe für einen Zeitwert aus der ersten Spalte kein Wert, so ist die entsprechende Zelle der Tabelle leer (Bereich *a* der Abbildung) [JIAN20]. Eine weitere Darstellung ist die grafische Darstellung in einem Koordinatensystem (Bereich *b* der Abbildung). Hier ist die Zeit entlang der Abszisse, die Werte entlang der Ordinate aufgetragen. Für jedes z_i ergibt sich dadurch eine eindeutige Position in dem Koordinatensystem. Sollen mehrere Zeitdatenreihen dargestellt werden, können mehrere Ordinaten in der Darstellung verwendet werden. Die Nutzung von Linien oder Splines zwischen den Punkten in dem Koordinatensystem dient der Unterstützung der Lesbarkeit. Diese häufige Methode hat allerdings keine Aussagekraft über die tatsächlichen Werte zwischen den definierten Punkten. [SARA05]

Die letzte vorgestellte Möglichkeit der Darstellung (im Bereich *c* der Abbildung) ist die Nutzung der Graphentheorie. Ein gerichteter Graph G_{sync} besteht aus Knoten (engl. »nodes«) KN und Kanten (engl. »edges«) $KA \subseteq KN \times KN$. Optional können Knoten und Kanten auch Gewichte oder Label zugeordnet werden [STEP15]. Für die Darstellung von

Zeitdatenreihen kann der gerichtete Graph die Abfolge der Daten darstellen. Für Vergleiche von Sequenzen können so direkt auch weiterführende Analysen basierend auf der Graphentheorie angewendet werden, um beispielsweise Ähnlichkeitsberechnungen durchzuführen [ZARE23]. In dieser Arbeit werden alle vorgestellten Darstellungen genutzt.

Eigenschaften von Zeitdatenreihen

Wie zuvor beschrieben, hat Zeit grundsätzlich einen streng monotonen Charakter. Aufgrund der Umsetzung und Einschränkungen der Zeitmessung in technischen Systemen (Abschnitt 2.2.3) gilt diese Eigenschaft bei Betrachtung einer Zeitdatenreihe aus technischen Systemen nur eingeschränkt. Die Zeitdatenreihen wie oben definiert bestehen lediglich aus den Tupeln aus Zeit und Wert. Dadurch kann strukturell keine Aussage über Wert-Tupel abseits der gegebenen Tupel getroffen werden. Aus Abschnitt 2.2 ergibt sich, dass Zeit existiert und in Abschnitte von mindestens der Größe der Planck-Zeit unterteilt werden kann. Nicht für jeden Zeitpunkt ist allerdings ein Wert vorhanden. Um Punkte in der Zeit zu betrachten, die keinen expliziten Wert in den Zeitdatenreihen haben, werden diese entsprechend dargestellt als (t_i, ϵ) . Zur Unterscheidung zu den gegebenen Zeitdatenreihen befinden sich diese Tupel in $Z^\epsilon \supset Z$:

$$\neg(\forall t_i \exists v_i ((t_i, v_i) \in Z)) \quad (2.14)$$

$$\begin{aligned} \implies \forall t_i (\exists v_i ((t_i, v_i) \in Z) \\ \vee \exists \epsilon ((t_i, \epsilon) \in Z^\epsilon)). \end{aligned} \quad (2.15)$$

Durch die Nutzung der rationalen Zahlen für die Zeitangaben kann zudem für die zeitliche Distanz die euklidische Metrik angewendet werden zu $d(z_i, z_j) = |t_j - t_i|$.

Aus Zeitdatenreihen lassen sich weitere deskriptive Eigenschaften beschreiben. Die Eigenschaft der *Schiefe* (engl. »Skewness«) beschreibt die Asymmetrie der Daten um ein Zentrum herum. Dieses Zentrum kann beispielsweise der Mittelwert der Daten in der Reihe sein. Diese Eigenschaft errechnet sich durch

$$\gamma(Z) = \frac{1}{|Z|\sigma^3} \sum_{\forall (t_i, v_i) \in Z} (v_i - \bar{v})^3, \quad (2.16)$$

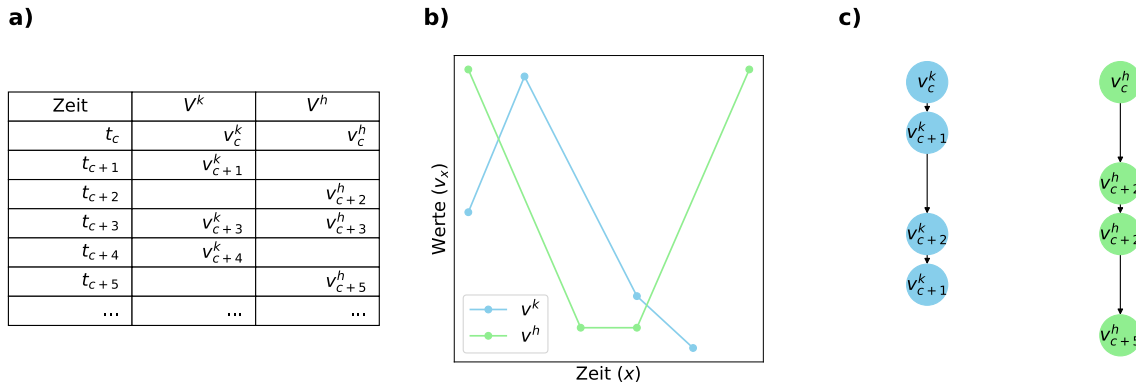


Abbildung 2.9: Darstellungen von Zeitdatenreihen als (a) Tabelle, (b) Koordinatensystem und (c) gerichtete Graphen

wobei σ die Standardabweichung und $\bar{v}$ den Mittelwert der Datenreihe angibt. Sind die Werte der Reihe gleichmäßig oder symmetrisch verteilt, ergibt sich $\gamma(Z)$ zu einem Wert nahe 0. Ein positiver Wert gibt an, dass mehr Werte über dem Mittelwert liegen (*positive Schiefe*), während ein negativer Wert beschreibt, dass eine größere Menge von Werten existiert, die unter dem Mittelwert liegen (*negative Schiefe*). [WANG06]

Ergänzend zur Schiefe beschreibt auch die Wölbung (engl. »Kurtosis«) die Verteilung der Werte einer Zeitreihe. Sie beschreibt wie flach oder spitz die Verteilung der Werte im Vergleich zu einer Normalverteilung ist. Dies ist analog zu den Definitionen der Schiefe beschrieben als

$$\eta(Z) = \left(\frac{1}{|Z|\sigma^4} \sum_{\forall (t_i, v_i) \in Z} (v_i - \bar{v})^4 \right) - 3. \quad (2.17)$$

Der Wert von 0 entspricht der Normalverteilung. Ein positiver Wert bedeutet eine gespitze Verteilung, hingegen ein negativer Wert eine flachere Verteilung [WANG06]. Die Schiefe und die Wölbung sind deskriptive statistische Eigenschaften, welche insbesondere definiert sind für Verteilungen mit einem Gipfel statt mehreren. Die Anwendung auf komplexere Zeitdatenreihen muss daher vorher geprüft werden durch Methoden wie die Kerndichteschätzung nach PARZEN [PARZ62].

Am Übergang von den deskriptiven Eigenschaften zur Analytik befinden sich die Eigenschaften *Trend* und *Saisonalität*. Trend beschreibt dabei die langfristige Bewegung der

Daten unabhängig von lokalen und kurzen Schwankungen. Übliche Methoden hierfür sind lineare Regressionen oder die Theil-Sen Methode, welche weniger anfällig für Ausreißer ist [WANG05]. Die Saisonalität fokussiert wiederkehrende Muster innerhalb der Zeitreihen. Für die Modellierung und den Ausgleich der Saisonalität existieren verschiedene Methoden. So kann sie einerseits in Frequenzkomponenten zerlegt und durch Fourier-Reihen modelliert werden. Andererseits bestehen Methoden wie die Holt-Winters-Methode, welche die Saisonalität durch eine exponentielle Glättung direkt ausgleicht in den Daten [GELP10]. Weiterhin existiert die Methodik der saisonalen Dekomposition. Die Anwendung dieser Methode zerlegt eine Zeitreihe in die verschiedenen Komponenten wie Saisonalität und Trend separat. In den vergangenen Jahrzehnten wurden hierzu Variationen entwickelt, um den Umgang mit Ausreißern, Saisonalitäts-Verschiebungen, Periodenlängen und abrupten Veränderungen zu verbessern [WEN19].

Um die Gleichmäßigkeit der Zeitdatenreihe zu beschreiben, können entsprechende Maße genutzt werden. Für eine betrachtete Zeitdatenreihe Z lässt sich als Hilfsvariable die durchschnittliche Zeit $\bar{t}$ innerhalb der Zeitdatenreihe nutzen. Mit dieser ergeben sich

$$\bar{t} = \frac{\sum_{\forall (t_i, v_i) \in Z} t_i}{|Z|} \quad (2.18)$$

$$Var_Z = \frac{1}{|Z| - 1} \sum_{i=1}^{|Z|-1} |t_i - t_{i-1} - \bar{t}| \quad (2.19)$$

$$\sigma_Z = \sqrt{\frac{1}{|Z| - 1} \sum_{i=1}^{|Z|-1} (t_i - t_{i-1} - \bar{t})^2}, \quad (2.20)$$

wobei Var_Z die Variation der Zeitabstände innerhalb der Zeitdatenreihen Z beschreibt und σ_Z die entsprechende Standardabweichung. Für gleichmäßig zeitlich verteilte Zeitdatenreihen sind die Variation und Standardabweichung minimal, während sie bei nicht-gleichmäßiger Verteilung entsprechend ansteigen.

Weiterhin lassen sich die Frequenzen von Zeitdatenreihen entsprechend definieren:

$$f_Z = \frac{|Z|}{t_{|Z|-1} - t_0}. \quad (2.21)$$

Entlehnt aus der Stochastik kann die Stationarität einer Zeitdatenreihe definiert werden

[BROC91]. Diese Eigenschaft beschreibt, dass für alle Subsequenzen einer Zeitdatenreihe die statistischen Eigenschaften für die Werte aus V erhalten bleiben. Dies ergibt sich zu

$$\mathring{U}_1, \mathring{U}_2 \subset Z \implies F(\mathring{U}_1) = F(\mathring{U}_2) = F(Z), \quad (2.22)$$

wobei F die betrachtete statistische Funktion auf den Werten ist. Die Definition gilt grundsätzlich für Subsequenzen beliebiger Länge, eine Einschränkung auf eine Mindestlänge dieser Sequenzen kann aber sinnvoll sein, um ausreichende statistische Repräsentierbarkeit herzustellen.

Um das erste bzw. letzte Element in einer Zeitdatenreihe zu beschreiben, sind die Funktionen $\min$ und $\max$ entsprechend definiert. Zur Verkürzung der Notation lassen sich auch entsprechende Funktionen für die kleinsten Werte und Zeitangaben formulieren:

$$\min Z = z_i = (t_i, v_i) \mid j > i \forall z_j \in Z \quad (2.23)$$

$$\max Z = z_i = (t_i, v_i) \mid j < i \forall z_j \in Z \quad (2.24)$$

$$\min_t Z = z_i = (t_i, v_i) \mid t_j > t_i \forall z_j \in Z \quad (2.25)$$

$$\max_t Z = z_i = (t_i, v_i) \mid t_j < t_i \forall z_j \in Z \quad (2.26)$$

$$\min_v Z = z_i = (t_i, v_i) \mid v_j > v_i \forall z_j \in Z \quad (2.27)$$

$$\max_v Z = z_i = (t_i, v_i) \mid v_j < v_i \forall z_j \in Z \quad (2.28)$$

Gilt die Monotonie bei den Zeitwerten einer Zeitdatenreihe, so gilt $\min Z = \min_t Z$ und $\max Z = \max_t Z$. Für nicht-monotone Zeitwerte ist so eine Unterscheidung möglich. Soll statt des vollständigen Zeitdatenreihenelements nur die entsprechende Zeitangabe (Zeitwert) beschrieben werden, können die Funktionen $\arg \min_t$ und $\arg \max_t$ analog genutzt werden.

Vergleichsmetriken zwischen Zeitdatenreihen

Neben den Metriken, die sich auf eine einzelne Zeitdatenreihe beziehen, existieren ebenfalls Metriken zum Vergleich von zwei solcher Reihen miteinander. Diese lassen sich in verschie-

dene Kategorien unterscheiden, angefangen bei Strukturanalysen über Form-Analysen zu Korrelationsmaßen und Frequenzmaßen.

Im Bereich der Strukturanalysen beschreibt die Metrik *Normalized Mutual Information* (NMI), wie viele Informationen zwei gegebene Zeitdatenreihen gemeinsam haben in Relation zu ihrer Entropie. Diese Metrik gibt damit an, wie weit Aussagen zu einer Zeitdatenreihe auf die andere übertragbar sind. Die Berechnung ergibt sich aus

$$\text{NMI}(k, l) = \frac{2I(k, l)}{H({}_kZ) + H({}_lZ)} \quad (2.29)$$

$$I(k, l) = \sum_{v^k, v^l} p(v^k, v^l) \log\left(\frac{p(v^k, v^l)}{p(v^k)p(v^l)}\right), \quad (2.30)$$

wobei $H(Z)$ die Shannon-Entropie der jeweiligen Zeitreihe angibt.

Eine weitere Strukturanalyse ist der *Pattern Matching Index* (PMI). Diese Analyse überprüft die Existenz von Mustern und wie weit diese übereinstimmen. Die grundsätzliche Formulierung ergibt sich daher zu

$$\text{PMI}(k, l) = \frac{1}{N} \sum_{i=1}^N \mathbf{1} \left[\text{Pattern um } z_i^k \text{ und } z_i^l \text{ stimmt überein} \right], \quad (2.31)$$

wobei $\mathbf{1}$ ein Vergleichsoperator ist, der überprüft, ob Muster um die gegebenen Elemente übereinstimmen. Durch Anpassung dieses Vergleichsoperators lässt sich die Prüfung so auf bestimmte Muster eingrenzen oder erweitern. In dieser Arbeit wird hier insbesondere die positive und negative Steigung betrachtet als Muster.

Neben den Strukturanalysen kommen auch Form-Analysen zum Einsatz. Dies ist insbesondere die *Fréchet-Distanz*. Diese Distanz gibt an, was der maximale Minimalabstand zwischen den Zeitdatenreihen ist, wenn beide monoton kontinuierlich durchlaufen werden. Daraus ergibt sich eine Einschätzung der Ähnlichkeit der sequentiellen Strukturen der Zeitdatenreihen unter Berücksichtigung der Werte und der Reihenfolge. Diese Distanz lässt sich beschreiben durch

$$d_F({}_kZ, {}_lZ) = \inf_{\alpha, \beta} \max_{t \in [0,1]} d(z_{\alpha(t)}^k, z_{\beta(t)}^l), \quad (2.32)$$

wobei α und β Parametrierungen angeben, also angepasste Durchlaufgeschwindigkeiten.

Der Bereich der Korrelationsmaße umfasst mehrere Maße, welche verschiedene Schwerpunkte darstellen. Die direkte *Kreuzkorrelation* analysiert dabei, wie stark zwei Zeitdatenreihen miteinander korrelieren, wenn eine der beiden eine zeitliche Verschiebung erfährt. Dies bedeutet, dass zeitliche Verzögerungen zwischen den Reihen gefunden werden können. Dies lässt sich beschreiben als

$$\text{XCor}_{k,l}(\sigma) = \sum_{i=1}^{N-\sigma} (v_i^k - \bar{v}^k) (v_{i+\sigma}^l - \bar{v}^l), \quad (2.33)$$

wobei hier σ angibt, wie weit die Verschiebung sein darf. Unabhängig von diesen Verzögerungen prüft hingegen die *Pearson-Korrelation* lineare Zusammenhänge zwischen den Zeitdatenreihen. Diese ergibt sich daher zu

$$\text{Pearson}(k, l) = \frac{\sum_{i=1}^N (v_i^k - \bar{v}^k) (v_i^l - \bar{v}^l)}{\sqrt{\sum_{i=1}^N (v_i^k - \bar{v}^k)^2} \sqrt{\sum_{i=1}^N (v_i^l - \bar{v}^l)^2}}. \quad (2.34)$$

Während die Kreuzkorrelation die Verzögerung und die Pearson-Korrelation die linearen Zusammenhänge in den Fokus der Analyse rückt, fokussiert die *Spearman-Korrelation* auch nicht-lineare Korrelationen. Die Berechnung basiert auf den Rängen R_i^k bzw. R_i^l der Zeitdatenreihen. Dies macht die Berechnung unempfindlicher gegenüber Ausreißern und erlaubt die Berücksichtigung von nicht-linearen Zusammenhängen. Die Berechnung ergibt sich so zu

$$\text{Spearman}(k, l) = \frac{\sum_{i=1}^N (R_i^k - \bar{R}^k) (R_i^l - \bar{R}^l)}{\sqrt{\sum_{i=1}^N (R_i^k - \bar{R}^k)^2} \sqrt{\sum_{i=1}^N (R_i^l - \bar{R}^l)^2}}. \quad (2.35)$$

Es existieren weitergehende Metriken wie Frequenzmaße, die in dieser Ausarbeitung aber nicht weiter betrachtet werden.

2.3 Verteilte Systeme

Distributed Systems

In der heutigen Welt existiert eine Vielzahl an computerisierten Systemen und die Verbindung dieser Systeme («Vernetzung») spielt dabei eine zentrale Rolle. Diese Vernetzung ermöglicht den Austausch von Daten und Informationen über Entfernungen zur Verarbeitung in den computerisierten Systemen und zur gezielten Anzeige und Interaktion mit Menschen. Für die weltweite Vernetzung existieren immense Infrastrukturen, die dezentrale Zugänge zu Gebäuden, über tausend zentrale Internetknoten [IXPD24] und hunderte Unterseekabel zwischen den Kontinenten [TELE24] umfassen, ebenso wie lokale Infrastrukturen innerhalb von Unternehmen und Produktionen.

Die technische Grundlage all dieser Infrastrukturen und Netzwerke bilden Protokolle, welche von den Systemen und Geräten befolgt werden, um die Daten zu verarbeiten und an die richtige Stelle weiterzuleiten. Die grundsätzliche Struktur ist dabei ein stufenweises Ver- und Entpacken von Daten. Die Nutzdaten der Quelle werden pro Aufgabe mit weiteren Daten angereichert und an die nächste Stelle übergeben [PETE07]. Dieses Grundprinzip zeigt sich in dem standardisierten OSI MODELL, welches entlang einer einzelnen Datenverbindung in Abbildung 2.10 dargestellt ist [TECH94]. Die oberen Schichten (Sitzung, Darstellung, Anwendung) sind dabei spezifisch für das Endgerät und die ausgeführte Anwendungssoftware. Dort werden Aufgaben wie eine Verwaltung der aktuellen Sitzung, der Datenkompression und -verschlüsselung sowie der finalen Ein- und Ausgabe übernommen. Die Schichten darunter hingegen beschreiben Aufgaben, die die Transportationsebene in den Fokus nehmen. In der Transportschicht (4) wird der Datenstrom segmentiert und Zusatzinformationen zur Zuordnung vergeben oder gelesen, sodass die Reihenfolge und korrekte Adressierung im System (Port-Angaben) erreicht wird. Die darunter liegende Vermittlungsschicht (3) verpackt die Daten in Pakete und beschreibt Quell- und Zielsystem mit einer Adresse (z.B. IP-Adresse). Geräte im Netzwerk, die diese Adressen verstehen, pflegen Routing-Tabellen und senden eingehende Pakete gemäß dieser Tabellen weiter durch das Netzwerk, sodass diese ihr entferntes Ziel erreichen können, welches nicht direkt mit dem Quellsystem verbunden ist. Darunter liegend hat die Sicherungsschicht (2) zwei Hauptaufgaben. Die eine Aufgabe betrifft die Prüfung der Korrektheit der ankommenden Daten und Flusskontrolle, also die Kontrolle und Planung der Übertragungen. Die andere Aufgabe ist der Medienzugriff. Da die meisten Medien nicht gleichzeitig bidirektional Daten übertragen können, regelt diese Schicht den Zugriff auf das Medium und das Verhalten, wenn eine Kollision festgestellt wird. Zuletzt befasst sich die unterste Ebene, die Bitübertragung (1), mit den Methoden zur Veränderung der physischen, elektrischen oder anderen

physikalischen Eigenschaften der Welt. Dadurch beschreibt diese Schicht, wie einzelne Bits über den Äther gesendet und empfangen werden. Dieses Modell stellt die Basis jeder Kommunikation in Netzwerken dar. Aus den Eigenschaften der Schichten lässt sich auch erkennen, dass keine exakte Zeit der Verarbeitung und Übertragung von Quelle zu Ziel garantiert werden kann [MARI14].

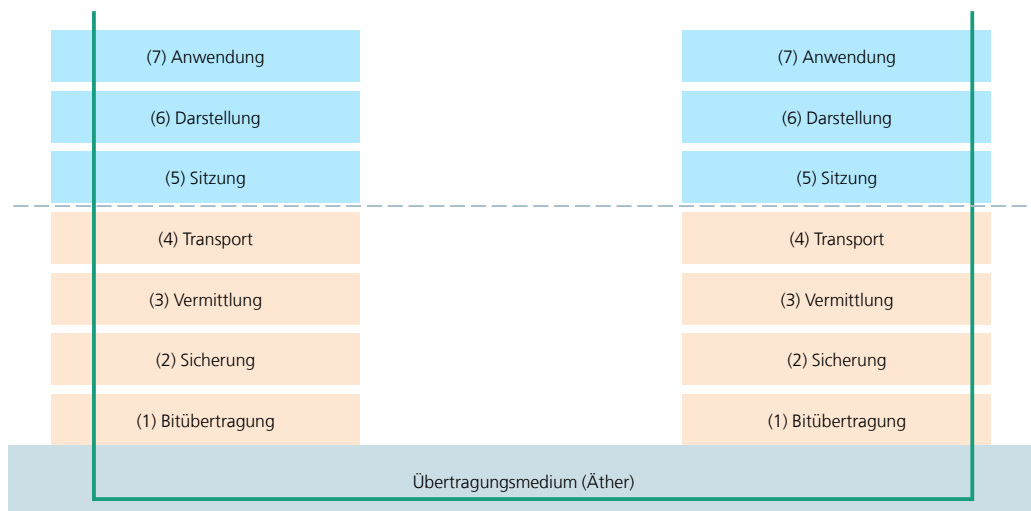


Abbildung 2.10: Das ISO/OSI Modell für die Vernetzung von Systemen und die Datenflüsse durch die Schichten. (Basierend auf [TECH94])

Skalierung der Netzwerk-Infrastruktur

Der Aufbau der Infrastruktur für eine durchgehende Vernetzung zusammen mit der Steigerung der Rechenleistung von Computersystemen hat über die Zeit neue Berechnungs- und Dienst-Strategien hervorgebracht. Die maßgeblichen Entwicklungen hierzu sind einerseits das Feld der hochskaligen Verarbeitung *Big Data* sowie andererseits eine strukturelle Skalierung durch *Edge, Cloud und Fog Computing* [OMET22, ANGE21]. In der klassischen Architektur sind monolithische Systemarchitekturen üblich, welche hauptsächlich vertikal skalieren. Dies bedeutet, dass für eine Skalierung der Aufgaben die zugrunde liegenden Systeme stärker und größer werden müssen. Das Bereitstellen weiterer Systeme ist nicht, oder nur mit Einschränkungen möglich. Für die hochskalige Architektur hingegen ist die

horizontale Skalierung das Kernelement. Dabei profitieren Lösungen weniger von einer Verbesserung der zugrundeliegenden einzelnen Systeme, sondern sind optimiert um verteilte Ressourcen, also beispielsweise hunderte Systeme, bestmöglich zu nutzen. Dadurch ist die horizontale Skalierung zunächst effizienter umsetzbar, da die Leistung eines Einzelsystems schneller an Grenzen der Verfügbarkeit gerät [SHAR10].

Eine Hauptunterscheidung der örtlichen Aufteilung der Berechnungen ist hier die Trennung in Cloud- und Edge-Level. Üblicherweise ist die Cloud selbst ortsunabhängig, beispielsweise aufgeteilt über mehrere Rechenzentren lokal oder weltweit. Durch den Fokus auf die horizontale Skalierbarkeit stehen auf dem Cloud-Level eine signifikante Menge an Ressourcen zur Verfügung. Über den Rechenzentren liegt eine Virtualisierungsebene und auch ein virtuelles Netzwerk, sodass für Anwendungen und Software die genauen Gegebenheiten der Hardware und Örtlichkeit nicht erkennbar und vernachlässigbar sind [ARD16]. Als Konsequenz existieren spezifische Architektur Modelle von Diensten in der Cloud-Umgebung. Die größten Modelle sind hierbei *Platform as a Service (PaaS)*, *Software as a Service (SaaS)* und *Infrastructure as a Service (IaaS)*. *Software as a Service (SaaS)* beschreibt hier das Angebot, dass eine Software in der Cloud-Infrastruktur ausgeführt wird für die Benutzenden, komplett verwaltet und betreut durch den Anbieter. Soll jedoch nur eine Plattform angeboten werden, welche die Entwicklung und Ausführung von Software ermöglicht, ohne Details der Infrastruktur zu kennen, ist dies das Modell *Platform as a Service (PaaS)*. Die größte Kontrolle über die Systeme haben Nutzende bei dem *Infrastructure as a Service (IaaS)* Modell, da hier (virtuelle) Infrastrukturkomponenten wie virtuelle Maschinen und Netzwerke bereitgestellt werden und die Nutzenden so die eigene Plattform konfigurieren können [NADE22]. Begleitend zu diesen Modellen und der horizontalen Skalierung ist Software für die Cloud strukturell anstatt in einem großen Paket in eine Vielzahl kleinerer Komponenten aufgebrochen. Diese sind dann verteilt in der Cloud-Umgebung und können angesprochen werden. Diese Art der Softwarestruktur nennt sich *Microservices* und ist durch Techniken wie der Containerisierung, der Verpackung von Software in einem Container zusammen mit allen Abhängigkeiten zur Verteilung, der Standard im Cloud Computing [BARB23].

Örtlich stärker eingegrenzt und definiert ist das Edge-Level, was auch den Namen herleitet. Systeme auf dieser Ebene befinden sich an der Kante (engl. »Edge«) des Netzwerks. Dies bedeutet, dass ein Gerät im Edge-Bereich am Rande des Netzwerks zur restlichen Infrastruktur wie Cloud zur einen Seite und zu Benutzenden oder Datenquellen zur anderen Seite eingesetzt ist. Geräte in diesem Bereich sind in der Lage, entstehungsnahe Aufgaben mit Daten zu übernehmen [HAMD20]. Die Nähe bringt dabei die Vorteile geringer Latenzen für Verarbeitung und eventuelle Antworten oder Steuerungsbefehle. Wenn insgesamt

viele Daten verarbeitet werden sollen, kann der Einsatz von Edge-Technologien auch die Daten frühzeitig reduzieren und die Übertragungsmenge in die Cloud verringern [SHI16]. Techniken wie die Containerisierung aus der Cloud können ebenfalls auf der Edge-Ebene eingesetzt werden, um notwendige Software bereitzustellen oder diese aus der Cloud für die bessere Latenz aus der Cloud näher an die Kante des Netzwerks zu bringen [PAHL15].

Neben den Bereichen Cloud und Edge werden seit Jahren auch die Begriffe *Fog Computing* und *Mist Computing* diskutiert. Der Begriff und initiale Definition von Fog Computing geht dabei zurück auf das Unternehmen Cisco als Erweiterung der Cloud in Richtung Edge, um Funktionen wie ein geografisches Bewusstsein und geringere Latenz als in der Cloud zu ermöglichen [BONO12]. Neben den Systemen im Edge und Cloud Bereich sind hierbei weitere Verarbeitungssysteme dazwischen vorhanden und können für die Arbeiten genutzt werden [NAAS17]. Während Fog den Verarbeitungsbereich zwischen Edge und der Cloud beschreibt, erweitert Mist den Betrachtungsrahmen und bezieht Edge-Geräte, Sensoren und weitere verfügbare Ressourcen am Rande des Netzwerks ab der Datenaufnahme mit ein [ATLA18, PRED15].

Das Zusammenspiel eben jener Bereiche ist in der Abbildung 2.11 dargestellt. Die oberste Ebene ist die Cloud Ebene, wo die zentralen Systeme der Architektur hochskalig bereitstehen unabhängig von der Lokalität. In hellblau dargestellt ist der Edge-Bereich, wo die Geräte an der Kante zum Netzwerk sind, da sich weiter unten die Geräte und Maschinen befinden, welche mit der Welt interagieren oder Daten aufnehmen. Zwischen Edge und Cloud ist der Fog Bereich, worin zusätzliche Systeme Aufgaben übernehmen mit geringerer Latenz und Bezug zur Lokalität. Zuletzt zeigt der markierte orange Bereich die Erweiterung, Mist Computing, die sich über mehrere Ebenen bis auf die Geräte der untersten Ebene ausdehnt.

Wie in Abschnitt 2.2.3 gezeigt, haben technische Systeme Zeitmesser, welche nicht perfekt sind und daher Effekten wie Taktversatz und Gangfehler ausgesetzt sind. Zur Indikation, wann etwas stattgefunden hat verwenden die Systeme und die darauf laufende Software Zeitstempel. In Netzwerken existieren mehrere Computersysteme und damit auch mehrere Uhren, die alle unter jenen Effekten leiden [ELLI73] und damit Zeitstempel generieren, die von diesen Effekten beeinflusst wurden. Dienste in vernetzten Architekturen setzen dabei teilweise eine Mindestgenauigkeit und Synchronizität der Uhren voraus. Werden diese Annahmen verletzt, können diese Dienste fehlerhafte Daten ausgeben, erhöhte Ressourcenverbräuche geschehen oder die Funktionalität komplett ausfallen [AGUI09].

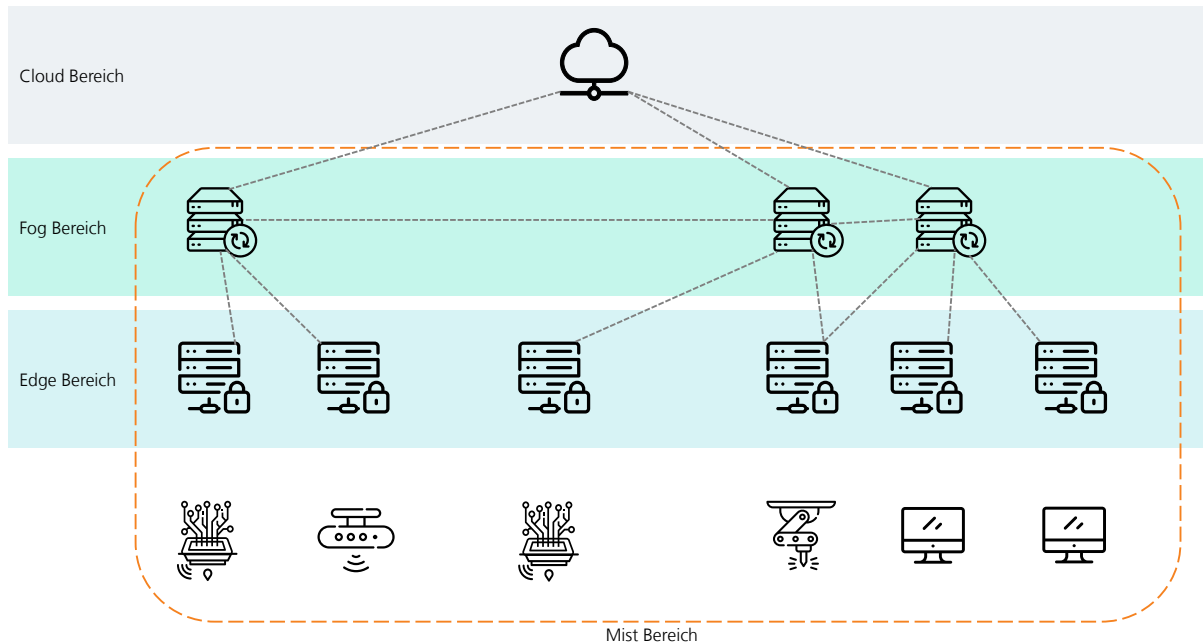


Abbildung 2.11: Die Bereiche Edge, Cloud, Fog und Mist als Architekturbild. (Angenommen an [DD19])

2.4 Produktionsnetzwerke

Production Network

Nicht erst mit der Hightech-Strategie des Bundesministeriums für Bildung und Forschung und dem Begriff *Industrie 4.0* existieren Computersysteme und Netzwerke in den Produktionsumgebungen [KAGE11]. Dabei erstrecken sich die Computersysteme auf verschiedene Ebenen innerhalb einer Produktionsumgebung.

Ein Standard-Modell für diese Ebenen bildet die sogenannte Automatisierungspyramide. Diese basiert auf den Definitionen des Standards ISA-95 und der Erweiterung durch die Norm IEC62264 und ist in Abbildung 2.12(a) dargestellt inklusive der dort üblicherweise verwendeten Systeme [INTE10, TECH13]. Die unterste Ebene (*Prozessebene*) umfasst dabei die Sensorik und Aktorik direkt im Produktionsprozess, welche direkt an die Ebene darüber verbunden sind. Diese *Feldebene* umfasst dann die Ein- und Ausgabe der Prozesssignale, beispielsweise über Feldbusse. Oberhalb davon befindet sich die *Steuerungsebene* und dient der Steuerung und Regelung durch Komponenten wie speicherprogrammierbare Steuerungen (engl. »Programmable Logic Controller PLC«). Darüber ist die *Prozessleitebene*. In dieser Ebene existieren Systeme zur Datenakquise und Überwachung (SCADA) sowie

Schnittstellen zur Interaktion mit der Maschine für Menschen (HMI). Auf dieser Ebene finden auch die Aufnahmen von Betriebsdaten technisch statt. Eng verbunden mit dieser Aufnahme sind die Systeme auf der *Betriebsleitebene*, welche diese Daten nutzen für die Produktionsdatenerfassung und damit verbundene Ermittlung von Produktionskennzahlen. Ebenfalls findet die Produktionsfeinplanung, das Materialmanagement und Qualitätssicherung in dieser Ebene statt. Zentrales System hierzu ist das Manufacturing Execution System (MES), welches diese Aufgaben übernimmt. Für spezifische Aufgaben kann dieses MES durch weitere Systeme wie ein Labor-Information-Management-System (LIMS) für Labordaten erweitert werden. Die oberste Ebene, die *Unternehmensebene*, befasst sich im Gegensatz zum MES nur mit einer Grobplanung der Produktion, kennt aber dafür unternehmerische Größen und Bestellungen, die für die Ebenen darunter zerlegt und weitergegeben werden. Daraus ergibt sich in der Konsequenz, dass die unteren Ebenen kürzere Kommunikationszyklen als die oberen Ebenen beinhalten. Auch wenn diese Definitionen auf die Ebenen und Definitionen auf Normen und Standards zurückgehen, gibt es dennoch Unterschiede und Variationen der Automatisierungspyramide in der Literatur. Die meisten der Variationen fassen dabei unterschiedlich viele der unteren Ebenen zusammen [MEUD17]. Für diese Arbeit wird die Pyramide wie in der Abbildung dargestellt verwendet und keine der Ebenen zusammengefasst.

Ein jüngeres Standard-Modell, welches aus der Zielstellung der Industrie 4.0 entstanden ist [PLAT19], ist das Referenzarchitekturmodell Industrie 4.0, dargestellt in Abbildung 2.12(b). Im Vergleich zur Automatisierungspyramide sind hier weitere Dimensionen hinzugefügt, sodass dieses Modell einem Würfel statt einer Pyramide gleicht. Auch dieses Modell ist einerseits standardisiert als IEC PAS 63088, greift andererseits auch bestehende Standards und Normen in den Dimensionen auf [IEC 17]. Die drei Dimensionen des Würfels beschreiben dabei verschiedene Aspekte und Unterscheidungen. Die Dimension zu *Hierarchy Levels* bezieht sich auf mehrere Standards, welche aus der gleichen Familie stammen, wie weiter oben in Bezug auf die Automatisierungspyramide beschrieben. Der Großteil der Ebenen in dieser Dimension kann aus dem oben beschriebenen Modell übernommen werden als Variation der Namen. Zwei Ebenen sind allerdings neu hinzugefügt. Am unteren Ende ist die Ebene »Product« neu, welche die Möglichkeit beschreibt, dass das Produkt selbst eine aktive Rolle einnimmt und kommunizieren kann. Neu hinzu kommt ebenfalls eine weitere Ebene oberhalb der Unternehmensebene, die »Connected World«. Diese Ebene beschreibt die Möglichkeit, dass Daten und Informationen auch über die eigenen Unternehmensgrenzen ausgetauscht werden können. Die Dimension *Life Cycle & Value Stream* basiert auf dem Standard IEC62890, welcher die Strategien für das Lebenszyklus-Management von Anlagen, Komponenten und Diensten beinhaltet [IEC 20]. Die Grundidee ist hierbei die Unterteilung aller Komponenten und Produkte in der Produktion in ihre Lebensphasen.

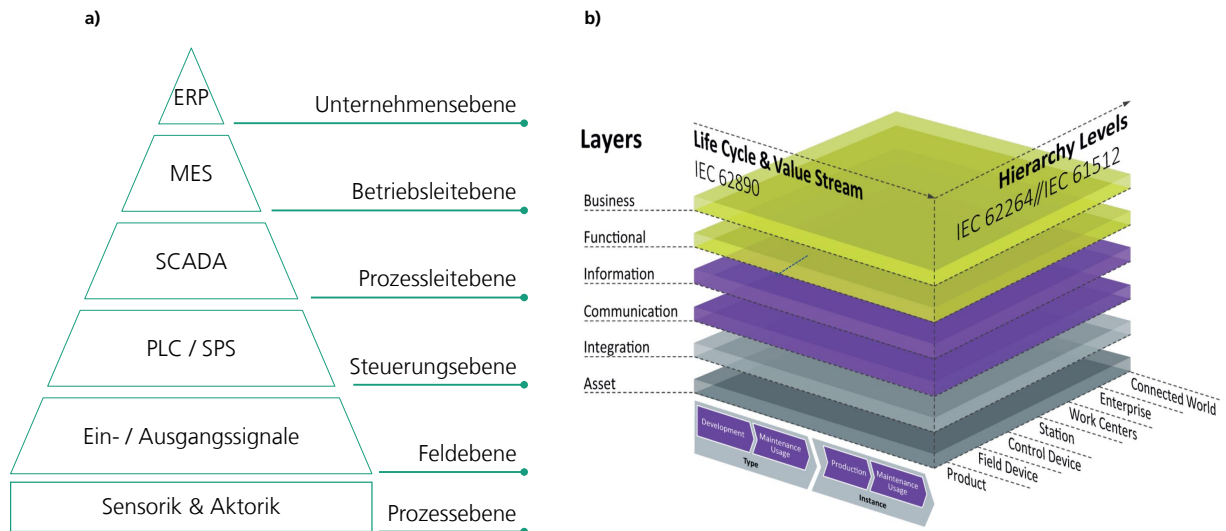


Abbildung 2.12: Darstellungen der Automatisierungspyramide (a) und des Modells der Referenzarchitektur der Plattform Industrie 4.0 (b). Das zweite Modell erhöht dabei die Komplexität durch mehr Dimensionen als die Automatisierungspyramide. ((a) basierend auf [INTE10, TECH13]; (b) entnommen aus [HANK15])

Dabei startet jedes Produkt ohne physische Entsprechung in der Entwicklungsphase als Typ. Durch die Produktion eines solchen Produkts entstehen dann physische Instanzen eben jener Typen. Dadurch kann ein Produkt einem Typen zugeordnet sein, und einem Typ beliebig viele Instanzen. Die letzte Dimension nennt sich *Layers* und ist eine Architektur-Achse. Hier ist die unterste Ebene »Asset« der physischen Welt zugeordnet, während die fünf Ebenen darüber in der Informationswelt existieren. Zur Verbindung dieser Welten existiert die »Integration« Ebene, die als Übersetzer zwischen den Welten agiert. Diese Ebene dient dabei auch als Ort für Daten, Funktionen und Applikationen, welche nicht kompatibel zu den Industrie 4.0 Ansätzen sind oder nicht sein sollen. Für Daten ergibt sich daraus, dass Datenpunkte, die in der physischen Welt erzeugt werden, einen Punkt in der Integrations-Ebene haben. Wenn dieser Datenpunkt relevant und konform zu organisatorischen Vorgaben und Industrie 4.0 Ansätzen ist, wird er an die höheren Ebenen weitergegeben. Die oberen Ebenen befassen sich mit der technischen Kommunikation (analog zu Abschnitt 2.3), semantischen Beschreibungen und Funktionen. Daraus ergeben sich neue Möglichkeiten, sodass Objekte selbstständig kommunizieren, was sie sind und welche

Funktionen sie anbieten können, sodass ein Abgleich von notwendigen und verfügbaren Verarbeitungsressourcen automatisch möglich sein könnte [QIN21]. Die oberste Ebene *Business* bestimmt dabei die nicht-technischen und nicht-fachlichen Regeln, der Objekte [HEID17]. Insgesamt ist dieses Modell damit deutlich komplexer als das Modell als Automatisierungspyramide, bricht damit aber auch die starre Struktur dieser Automatisierung auf und beschreibt die Kommunikation von Systemen direkt untereinander anstelle einer klaren Kommunikationshierarchie. In dieser Arbeit werden einzelne Aspekte dieses komplexeren Modells (wie die Informationsmodellierung und Semantik) aufgegriffen, ansonsten aber das einfachere Modell verwendet, da keine kommunizierenden Produkte betrachtet werden.

Aus diesen Architekturen ergibt sich die zentrale Rolle von Systemen wie einem MES, welches ohne verlässliche Kommunikation und Daten nicht die Aufgaben übernehmen kann, wodurch die gesamte Produktion gefährdet ist [ITSK16]. Externe regulatorische Bestrebungen zu digitalen Produktpässen erhöhen ebenfalls die Notwendigkeit einer verlässlichen Infrastruktur und Weitergabe von Daten [WALH23]. Im Jahr 2022 gaben 90 % der produzierenden Unternehmen in Deutschland an, aktuell bereits Industrie 4.0 Anwendungen zu nutzen oder zu planen [MARK22]. Durch diese dargestellte Relevanz der Daten und technische Entwicklungen erhöht sich die Menge der Daten und Komplexität der benötigten Dateninfrastruktur. Um diesen Anforderungen zu begegnen haben sich spezialisierte Vernetzungsplattformen entwickelt, welche unter dem Begriff *Internet of Things (IoT)* für allgemeine Anwendungsfälle Verbreitung finden [KAHV22]. Sie übernehmen dabei die Rolle einer Middleware, also eine Plattform, die die Verbindung der Systeme und Ebenen koordiniert und vereinfacht. Gemäß der Hierarchien in Abbildung 2.12 sind sie am ehesten in der Prozessleitebene beheimatet, allerdings mit Verbindungen zu allen anderen Ebenen, sodass topologisch die Struktur eher einer Sterntopologie gleicht. Ihr Funktionsumfang umfasst dabei auch Funktionen des Asset-Managements und der Informationsmodellierung, wie in den *Layers* der Referenzarchitektur Industrie 4.0 definiert. Die Definition einer IoT-Plattform ist dabei nicht eindeutig, üblicherweise aber auf die Konnektivität und Funktionen fokussiert, nicht auf Eigenschaften wie Stabilität und Sicherheit, und zudem zunächst unabhängig von der Einsatzdomäne. Zur Unterscheidung ist daher der Begriff *Industrial Internet of Things (IIoT)* für Plattformen genutzt, welche sich direkt an den Bedarfen und Anforderungen einer Produktionsumgebung orientieren. Diese Plattformen befassen sich dabei auch explizit mit der Industrie 4.0-konformen Datenerfassung, -anreicherung und -verteilung [SCHM19].

Einordnung von Uhren in Produktionsnetzwerken

In den obigen Abschnitten wurde gezeigt, dass moderne Produktionen über eine immer komplexere Architektur des Produktionsnetzwerks verfügen zusammen mit einer steigenden Menge von Datenquellen und Daten [BRET14, BEHR23]. Durch neue Paradigmen wie Cloud, Edge und Fog Computing steigt die Komplexität der Systemlandschaft und Anzahl von Systemen weiter an. Ebenfalls müssen mehr Entscheidungen bei der Architektur getroffen werden, da beispielsweise Maschinen wahlweise mit einem Edge Gerät, mit einem lokalen Server oder direkt mit der Cloud verbunden werden können. Neue Lösungen wie IIoT-Plattformen sind Ansätze, um dieser Komplexität entgegenzutreten, allerdings verhindern sie nicht den Anstieg an Datenquellen und Datenmengen. Ein Ausfall von Systemen in dieser Landschaft kann daher schwerwiegende Folgen bis hin zu einem kompletten Stillstand der Produktion haben.

Aus der benannten Komplexität ergibt sich auch, dass eine Vielzahl von Computersystemen existieren in verschiedenen Größen und an verschiedenen Standorten. Wie in Abschnitt 2.2.3 beschrieben, leiden alle diese Systeme unter Effekten wie Taktversatz und Gangfehler. Aus der Komplexität der Architektur lässt sich daher ableiten, dass die Synchronizität der Uhren in dem gesamten Produktionsnetzwerk exponentiell mit der Komplexität des Netzwerks (bezogen auf die Anzahl der System) an Komplexität steigt. Durch Paradigmen der Cloud liegen diese Systeme und deren Kontrolle oftmals nicht mehr in der Hand eines Unternehmens, sondern sind auf mehrere Beteiligte dieser Produktions-Infrastruktur aufgeteilt.

Aus der Definition der Zeiten in den Systemen aus 2.2.3 lässt sich damit festhalten, dass die Zeit für jedes System gleich beschrieben, jedoch pro System unterschiedliche definierte Werte beinhalten kann. Zur Darstellung und Schaffung einer Berechenbarkeit von Zeitbereichen, für die keine explizite Beschreibung vorliegt, kann daher die lineare Hülle der errechneten Zeiten von $\langle C_n \rangle$ als Vektorraum genutzt werden. Da die Darstellung der Zeiten selbst auf $\mathbb{Q}$ (als Körper) zurückgeht, sind mit dem neutralen Element 0 und Einselement 1 alle Bedingungen und Axiome für den entsprechenden Vektorraum erfüllt. Damit lässt sich ein Produktionsnetzwerk mit verschiedenen Uhren als eine Menge von Vektorräumen mit gleichem Körper und verschiedenen Basen darstellen.

2.5 Batteriezellfertigung

Battery Cell Production

Diese Arbeit beschreibt die Anwendung der Batteriezellfertigung. Daher gibt dieses Kapitel eine kurze Einführung in diesen Bereich und die Eigenschaften der Produktion, die für die Entwicklungen und Validierung entscheidend sind.

2.5.1 Batteriezellen und Eigenschaften

Battery Cells and Properties

Aktuell erlebt die europäische Gesellschaft mehrere grundlegende Veränderungen, die durch demografischen Wandel, technologischen Fortschritt und neue politische Dynamiken geprägt sind. Angesichts des Klimawandels und globaler Umwälzungen sowie der Abhängigkeit von nichteuropäischen Akteuren rückt das Thema Energiespeicher zunehmend in den Fokus. Insbesondere Batterien spielen eine zentrale Rolle bei der Umsetzung der Energiewende und der Transformation hin zu einer Mobilität ohne fossile Antriebe. Elektrisch betriebene Fahrzeuge und erneuerbare Energiesysteme, die eine effiziente Speicherung erfordern, sowie der steigende Energieverbrauch schaffen einen stark wachsenden Bedarf an zuverlässigen und leistungsfähigen Batterien. Vor diesem Hintergrund ist die Batterieproduktion zu einer Schlüsseltechnologie für die Energiewende in Europa geworden. Batterien bestehen aus einer oder mehreren Batteriezellen, die den größten Teil der Kosten, Ressourcen und des Produktionsaufwands ausmachen [DEGE22]. Somit bildet die Batteriezellfertigung den Kern dieser Schlüsseltechnologie.

Batterien selbst können in ihrem elektrochemischen Design in verschiedene Kategorien einsortiert werden. Aktuell die höchste Verbreitung in der Praxis haben *Bleibatterien* und *Lithium-Ionen-Batterien (LIB)*. Ein weiteres Zellkonzept ist die *Natrium-Ionen-Batterie (NIB)*, welche in den vergangenen Jahren ein hohes Interesse in der Forschung ausgelöst hat und insbesondere durch ihre unkritischeren Materialien, im Sinne der Lieferketten und ökologischen Fußabdruck, ein hohes Potenzial zugeschrieben wird. Da für die Natrium-Ionen-Batterien die umfassende Marktreife und Skalierung noch nicht erreicht ist, ist die aktuelle Bandbreite der Prognosen zum Marktvolumen sehr groß. So sagen Analysten aktuell Volumen von 50 GWh bis 600 GWh in 2030 vorher. Einen Zweifel an der Skalierung und der Marktrelevanz in Zukunft ist indes nicht zu beobachten [DEGE23]. Technisch bewegen sich die Natrium-Ionen-Batterien dabei in einem vergleichbaren Bereich wie die Lithium-Ionen-Batterien mit Lithium-Eisenphosphat (LFP) Kathoden. Ein anderes Zellkonzept mit hohem Potenzial ist die *Feststoffbatterie*. Im Gegensatz zu anderen Batterien sollen

verortet. Insgesamt ist also die Batterieproduktion, und die Batteriezellproduktion als ihr Kern, ein stark wachsender Markt und von strategischem Interesse für Europa.

Der Fokus dieser Arbeit liegt auf der Produktion der Lithium-Ionen-Batterie (LIB), wobei die Anwendung in Produktionen weiterer Zellkonzepte damit nicht ausgeschlossen ist, insbesondere durch die hohe Ähnlichkeit der Produktionen anderer Zellkonzepte zur LIB in wesentlichen Punkten. Innerhalb der LIB sind drei Formate am häufigsten anzutreffen. Die *prismatische* Zelle hat ein festes prismatisch geformtes Gehäuse mit den Kontaktstellen auf einer oder zwei Seiten. Im Gegensatz zum festen Gehäuse der prismatischen Zelle hat die *Pouchzelle* (engl. »Beutel«) eine Verbundfolie als Außenhülle, welche thermoelastisch signifikante Unterschiede hinsichtlich Sicherheit, Thermomanagement, Energie- und Leistungsdichte und weiterer Parameter aufweist, die sich bis auf die Fertigungsverfahren ausweitet. Das letzte verbreitete Zellformat ist die zylindrische Zelle, auch Rundzelle genannt, welche eine zylindrische feste Außenhülle besitzt. Diese Bauformen bieten verschiedene Vor- und Nachteile im Bereich der Recyclingfähigkeit [GA22], Wärmeabfuhr [PEGE24], Raumnutzung, Kapazität und Sicherheit sowie weiterer Eigenschaften. In bestimmten Domänen wie beispielsweise der Automobilbranche deuten sich Trends der letzten Jahre weg von Pouch-Formaten hin zu prismatischen Formaten an, allerdings sind weiterhin alle Formate relevant [LINK23].

Der grundsätzliche Aufbau einer Lithium-Ionen-Batteriezelle unterscheidet sich dabei nicht bei den Bauformen. Eine solche Batteriezelle ist ein elektrochemischer Energiespeicher, sodass durch chemische Reaktionen elektrische Energie gespeichert oder freigesetzt werden kann. Die Hauptbestandteile sind hierbei positive und negative Elektroden, Separator und Elektrolyt. Üblicherweise ist die positive Elektrode (vereinfacht Kathode genannt) aus einem lithiumhaltigen Metalloxid, die negative Elektrode (vereinfacht Anode genannt) aus Kohlenstoff-basierten Materialien. Der Separator ist dabei eine poröse Membran, die einen direkten Kontakt von Anode und Kathode verhindert, allerdings permeabel für Ionen ist. Diese Komponenten sind umgeben von Elektrolyt, welches diese Ionenmobilität sicherstellt [YOSH09]. Umgeben werden diese Hauptkomponenten, in Form eines Stapels oder Wickels, von einer Außenhülle wie den oben beschriebenen Metallgehäusen oder Folien.

2.5.2 Produktionsprozesse der Batteriezellfertigung

Production Processes of the Battery Cell Production

Der Produktionsprozess der Lithium-Ionen-Batteriezelle ist eine komplexe Sequenz von Produktionsschritten. Diese sind als Übersicht dargestellt in Abbildung 2.14. Der Produktionsprozess selbst ist in drei Hauptbereiche unterteilt [KAMP13]. Innerhalb dieser Bereiche



Abbildung 2.14: Die Produktionsprozesse der Lithium-Ionen-Batteriezelle, vertikal gruppiert in die Hauptbereiche Elektrodenfertigung, Assemblierung und Finalisierung. Innerhalb dieser Hauptgruppen unterscheiden sich die Produktionsabläufe teilweise je nach Zellformat, markiert durch die gerenderten Zellbilder. (Prozesse basierend auf [KAMP13], Renderings ©Fraunhofer-Forschungsfertigung Batteriezelle FFB)

kann sich der genaue Prozess leicht unterscheiden je nach Bauform. Der erste Hauptbereich, die *Elektrodenfertigung*, ist dabei für alle Formate weitgehend gleich (ausgenommen beispielsweise Konfigurationsunterschiede innerhalb der Prozesse) und beschreibt den Bereich mit dem größten Einfluss auf die gesamte Produktion und die finalen Batteriezellen [GONÇ22]. Die Elektrodenfertigung startet mit dem Dosieren und Mischen, wo die Grundmaterialien der Anode und Kathode jeweils separat gemischt werden zum sogenannten »Slurry«. Im Anschluss werden diese Grundmaterialien im Beschichtungsprozess auf die Trägerfolie (z.B. Aluminium oder Kupfer) nass aufgebracht und getrocknet. Diese geschieht üblicherweise in einem kontinuierlichen Rolle-zu-Rolle-Prozess, sodass komplette Folienrollen (»Muttercoils« genannt) als Eingangsmaterial dienen. Diese beschichtete Folie wird im Folgeschritt mittels Kalandrierverfahren verdichtet, bevor sie im Slitten längs getrennt wird. So entstehen aus einem Muttercoil mehrere »Tochtercoils«. Als letzter Schritt der Elektrodenfertigung wird die Restfeuchte in einem Vakuumofen entzogen, in dem diese Tochtercoils bis zu 48 Stunden lagern. Diese Prozesskette der Elektrodenfertigung ge-

schiebt einerseits für die Anodenherstellung und andererseits für die Kathodenherstellung, wobei aufgrund der eingesetzten Materialien eine Kreuzkontamination zu vermeiden ist [KAMP15].

Die zweite Hauptgruppe in der Produktion ist die *Assemblierung*, die sich mit dem Zusammenbau der eigentlichen Zelle befasst. Daher sind in diesen Prozessen Unterschiede entlang der Bauformen vorhanden. Als Halbzeug werden getrocknete Tochtercoils aus der Elektrodenfertigung verwendet. Da die Feuchte einen kritischen, negativen Einfluss auf die Zellqualität darstellt und der Eintrag minimiert werden soll, findet die Assemblierung in Trockenraumumgebung statt. Ziel des ersten Schritts ist ein Zwischenprodukt, welches Anoden, Kathoden und Separatoren in der korrekten Reihenfolge zusammenfasst. Dabei kann wahlweise ein Zuschneiden (Vereinzeln) und Stapeln von Elektroden sheets geschehen oder ein Wickeln auf Rollenebene sowie weitere Schritte des Zuschneidens (»Notching«). Die Ableiter der hieraus entstandenen Zellstapel (»Stacks«) oder Zellrollen (»Jelly-Rolls«) werden kontaktiert und zu den Kontaktpunkten der positiven bzw. negativen Batteriepole geführt. Danach werden die Zwischenprodukte in die jeweilige Hülle (Pouchfolie, prismatisches oder zylindrisches Gehäuse) eingebracht. Der letzte Kernbestandteil, der Elektrolyt, wird über Dosierlanzen in diese Gehäuse eingefüllt und die Zelle final verschlossen [KAMP13]. Die Zelle ist damit fertig montiert.

Der letzte Hauptabschnitt der Produktion, die *Finalisierung*, befasst sich mit der elektrochemischen Fertigstellung der Batteriezelle. Der erste Schritt ist dabei ein Übergang von Assemblierung zur Finalisierung durch eine Hochtemperaturlagerung, welche das Ziel hat, die Durchtränkung der Elektroden sicherzustellen. Danach beginnt die Formierung, in der erstmals die Batteriezellen be- und entladen werden. Bei dieser erstmaligen Ladung bildet sich eine das Gesamtsystem stabilisierende Grenzschicht zwischen Elektrode und Elektrolyt (»Solid Electrolyte Interphase SEI«) [ADEN23]. Je nach Zellformat und Zellgröße können im Nachgang entwickelte Gase abgeführt und durch SEI-Bildung verbrauchter Elektrolyt nachgefüllt werden. Die letzten Schritte befassen sich mit abschließenden Tests der Zellen um Veränderungen von Zelleigenschaften oder Kurzschlüsse zu detektieren. Diese Überwachung kann mehrere Wochen dauern, bevor die Zelle final verpackt und versendet werden kann [HEIM18]. Der Bereich der Finalisierung ist in erster Linie dominiert durch elektrochemische Prozesse statt mechanischer Produktionsprozesse und den abschließenden Qualitätssicherungsschritten. Gleichzeitig sind die Lagerzeiten ein signifikanter Faktor der Produktionszeit einer Batteriezelle und zugleich können viele Fehler aus den Produktionsschritten in der Elektrodenfertigung und Assemblierung erst am Ende festgestellt werden [PLUM23]. In der Praxis sind die sogenannten Formierprotokolle vielfältig, sodass

auch die Reihenfolge dieser Schritte möglicherweise verändert oder iterativ durchlaufen werden kann.

Mit Blick auf die Definition und Unterteilung von Fertigungsverfahren nach DIN 8580 lässt sich erkennen, dass die Batteriezellfertigung eine Vielzahl von grundverschiedenen Verfahren umfasst [DEUT22]. In der Abfolge sind verfahrenstechnische, kontinuierliche, diskrete und elektrochemische Prozesse gleichermaßen vertreten. Dies zeigt die Komplexität der Kette der Batteriezellproduktion. Aus jedem der verschiedenen Schritte und Bereiche ergibt sich eine Vielzahl offener Fragestellungen und laufender Forschungs- und Entwicklungsaktivitäten. Dies stellt eine signifikante Herausforderung für die Anlagenhersteller, Integratoren und Produzenten dar [MICH23].

2.5.3 Digitalisierung der Batteriezellfertigung

Digitalization of Battery Cell Production

Ein signifikanter Hebel, um bestehende Probleme zu lösen, Vorhersagen zu ermöglichen und Optimierungen zu vollziehen ist die Digitalisierung der Batteriezellfertigung. Mithilfe der Daten aus der Produktion der Batteriezellen sind Einblicke und Optimierungen möglich entlang verschiedener Anwendungsfälle. Eine finale Kontrolle der Qualität der Batteriezelle ist produktionsbedingt erst am Ende der Produktionskette möglich. Datenbasierte Vorhersagen der Qualität (engl. »Predictive Quality«) ermöglicht ein Ausschleusen von Schlechtteilen oder die Nicht-Verarbeitung von Bahnmetern, welche eine zu geringe erwartete Qualität haben. Dadurch können Ressourcen reduziert und Ausschuss vermieden werden. Eine andere Art der datenbasierten Vorhersage in der Batteriezellfertigung befasst sich mit den Anlagen der Produktion und deren Ausfall- und Wartungsfenster (engl. »Predictive Maintenance«). Durch eine bessere Planbarkeit und Vermeidung von Ausfällen wird die Gesamteffizienz der Produktion deutlich gesteigert. Auch die Aufnahme der Energiedaten für die proaktive Steuerung der Energieverbraucher resultiert in sinkenden Kosten für die Produktion von Batteriezellen. Diese und weitere Anwendungsfälle bieten erhebliche jährliche Einsparungspotenziale bei den Produktionskosten. Der größte Hebel auf die Produktion besteht bei den Digitalisierungsanwendungen ebenfalls im Bereich der Elektrodenfertigung. [WUND24]

Während die Optimierung von Prozessparametern und Prozesseigenschaften innerhalb eines Prozesses vermehrt umgesetzt wird, ist die Verbindung von mehreren Prozessschritten und der Auswirkungen aufeinander komplexer und aktuell weniger betrachtet. Dafür existieren mehrere Gründe. So ist die zuerst notwendige Identifizierung relevanter Parameter und möglicher Wirkzusammenhänge zwischen Prozessen oder auf die finale Zelle eine zentrale

Herausforderung [FITZ23, MEIN22, OTTE24]. Gleichzeitig ist die Nachverfolgbarkeit von Änderungen, Auswirkungen und Merkmalen über den gesamten Produktionszyklus aufgrund der Komplexität der Produktion und veränderlicher Betrachtungsgegenstände (Granulate, Coils, Sheets, Zellen, etc.) erschwert [WESS23]. Zuletzt ist auch die prozessübergreifende Kommunikation zur Datenaufnahme und Zuordnung der Daten eine Herausforderung, welche bei der Menge an Anlagen, Sensoren und Datenquellen eine hohe Komplexität erreicht [WARD22]. Diese drei Herausforderungen bedingen sich teilweise gegenseitig, sodass sie parallel gelöst werden müssen um eine deutliche Steigerung in der Menge und Qualität von datenbasierten Anwendungen zu ermöglichen. Sie werden aus diesem Grunde in der deutschen Roadmap Batterie-Produktionsmittel als drei notwendige Technologiedurchbrüche (»Red Brick Walls«) für die Batterieproduktion identifiziert [MICH23].

Trotz dieser Herausforderungen steigen die Beiträge und Entwicklungen im Bereich der Optimierung der Batteriezellproduktion durch die Digitalisierung, insbesondere durch die Nutzung von Methoden wie dem maschinellen Lernen und digitalen Zwillingen [KRAU23]. Dabei ist allerdings zu beobachten, dass die meisten Entwicklungen hierbei einerseits nur einzelne oder wenige Prozessschritte betrachten und andererseits Produktionsstrecken von Labor- bis Pilotmaßstäben im Fokus sind. Anwendungen über mehrere Schritte oder auf industriellen Anlagen fehlen aktuell [HAGH23]. Bei der grundsätzlichen Betrachtung von datengetriebenen Anwendungen in der Batteriezellproduktion machen sich die notwendigen Technologiedurchbrüche insbesondere bemerkbar in mangelnder Datengrundlage und -qualität [TURE20].

Teil dieser Datenqualität ist die Zuordnung der Daten aus der Messtechnik und der Anlage zu konkreten Zwischenprodukten und den Orten auf eben jenen. Dies ist besonders deutlich am Beispiel der Elektrodenfertigung, welches hier kurz vorgestellt sei. Für eine datengetriebene Analyse ist die Zuordnung von Daten zu Orten auf den Bahnen der Coils notwendig, wobei eine höhere Genauigkeit bessere Aussagekraft ermöglicht [HAGH22]. Als Beispiel für die Bahnprozesse in der Elektrodenfertigung ist hier die Beschichtung und Trocknung relevant und exemplarisch in Abbildung 2.15 dargestellt, startend mit der Abwicklung der unbeschichteten Elektrodenfolie links, der Beschichtung mittels einer Schlitzdüse (grün) und der Aufwicklung der beschichteten Folie am Ende (rechts). Entlang der Bahn können an verschiedenen Orten zudem Messsysteme wie eine Nass- und Trockenfilmmessung integriert sein. Die Position innerhalb des Arbeitsraums entspricht dabei den Positionen s_1, s_2, s_3 .

Für eine ideale Produktion lässt sich die Bahnposition (Ortsstempel) SS für einen Datenpunkt aus einer Messtechnik daher mit den Produktionsdaten und Zeitstempeln errechnen durch

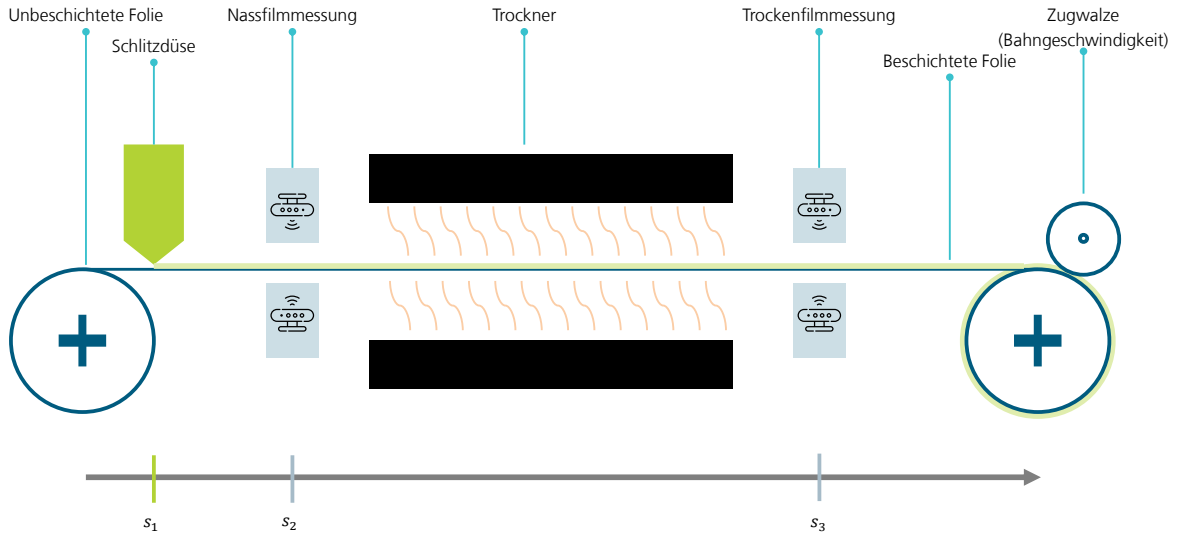


Abbildung 2.15: Ein exemplarischer Aufbau einer Beschichtung und Trocknung in der Elektrodenfertigung. Den Geräten ist eine relative Position s_x zugeordnet entlang der Bahnrichtung.

$$SS(TS_{MT}) = s_0 + s_{MT} + \vec{v} \times (TS_{MT} - TS_{start}), \quad (2.36)$$

wobei s_0 die Startposition der Bahn und TS_{start} den Zeitstempel zum Produktionsbeginn beschreiben. Entlang des Maschinenarbeitsraums ist die Position der entsprechenden Messtechnik durch s_{MT} gegeben sowie der Zeitstempel der betrachteten Messung, welcher der Datenpunkt zugrunde liegt durch TS_{MT} . So kann die Position auf der Bahn für die Datenpunkte berechnet werden, wenn die Bahngeschwindigkeit konstant ist. In der Praxis wird die Geschwindigkeit allerdings am Anfang erst hochgefahren, am Ende heruntergefahren und unterliegt während der Produktion gewissen Schwankungen [NIRI22]. Dadurch kann die Geschwindigkeit nicht als konstant angenommen werden, und die Formulierung 2.36 erweitert sich zu

$$SS(TS_{MT}) = s_0 + s_{MT} + \int_{TS_{start}}^{TS_{MT}} \vec{v}(t) dt. \quad (2.37)$$

Wissenschaftliche Labor- und Testlinien der Batteriezellproduktion nutzen in der Regel niedrige Bahngeschwindigkeiten von wenigen Metern pro Minute. Gleichzeitig zielt die Produktion im industriellen Maßstab darauf ab, die Bahngeschwindigkeiten von $80 \frac{m}{min}$ auf bis zu $100 \frac{m}{min}$ zu erhöhen. Daraus ergibt sich eine signifikant erhöhte Anforderung an die Genauigkeit der Daten, um die gleiche Ortsauflösung und damit verbundene Datenqualität zu erreichen. Wenn die zeitliche Genauigkeit des Zeitstempels für die Messtechnik TS_{MT} nur ± 100 ms beträgt, ergibt sich die orts aufgelöste Genauigkeit von SS zu $\pm 0,167$ cm bei einer geringen Bahngeschwindigkeit von $1 \frac{m}{min}$. Wird nun die gleiche Ungenauigkeit bei einer industriell skalierten Anlage mit $80 \frac{m}{min}$ betrachtet, ist die örtliche Ungenauigkeit hingegen bei $\pm 13,3$ cm. Diese Ungenauigkeit ist derart groß, sodass abhängig vom Zelldesign und Produktionsstrecke damit ein Datenpunkt nicht mehr nur nicht dem korrekten Ort, sondern in den Folgeschritten nicht einmal mehr dem richtigen Elektroden sheet zugeordnet werden könnte. Damit ist die Genauigkeit der Zeitangaben für datengetriebene Methoden in der Batteriezellfertigung von kritischer Bedeutung bei der Skalierung der Produktion auf einen industriellen Maßstab [KIES21].

2.6 Vorverarbeitung von Produktionsdaten

Preprocessing of Production Data

In diesem Abschnitt wird der Ablauf von datenbasierten Modellen mit Fokus auf den vorbereitenden Schritten dargestellt. Im Anschluss werden die konkreten Eigenschaften von Datenproblemen und den entsprechenden Schritten aufgezeigt zusammen mit einer Einordnung dieser Arbeit in diesem Bereich.

2.6.1 Verarbeitungsmodelle Produktionsdaten

Processing Models Production Data

Bei der Nutzung von Methoden der Digitalisierung und Analysen einer Produktion sind die Daten von entscheidender Bedeutung. Dabei hängen Ergebnisse, Güte und Nutzbarkeit der Analysen und Modelle direkt von der Qualität der Datenbasis ab. Dies ist einerseits die Relevanz der Daten, also ob das Ziel der Analyse überhaupt mit der Datenauswahl betrachtet werden kann. Andererseits ist die Qualität der Daten selbst entscheidend [CIOF20]. Dadurch ist die Untersuchung und Sicherstellung der Qualität von Daten notwendiger Schritt der Qualitätssicherung der Produktion.

Um die Qualität zu untersuchen und sicherzustellen wird vor einer eigentlichen Datenanalyse die Qualität geprüft und Korrekturmaßnahmen eingeleitet. Die genaue Aufteilung

und Abfolge der Schritte vor der Analyse unterscheidet sich dabei je nach Autor*innen. Im Kontext der Anwendung von maschinellem Lernen in der Produktion nach KRAUß ist der Ablauf in Abbildung 1.2 bereits dargestellt. Nach der Datenaufnahme (nicht im Bild), jedoch vor der Datenanalyse (Modellierung) finden mehrere Schritte statt. In dieser Darstellung ist dies in drei Schritte unterteilt. Der erste Schritt ist dabei die Datenintegration, bei der Daten aus mehreren Quellen zusammengeführt werden. Danach beginnt der Schritt der Datenvorverarbeitung und endet in der Extraktion von Kerneigenschaften (»Feature Engineering«) für die Analysen. Für all diese Schritte wird neben einer Data Science Expertise auch mindestens noch ein Prozessverständnis benötigt. [KRAU22]

Eine andere Abfolge beschreibt zunächst eine eingehende explorative Analyse der Produktionsdaten und danach eine Vorverarbeitung. Im Anschluss werden dann Kerneigenschaften von Zeitdatenreihen extrahiert und erst zuletzt Kerneigenschaften für den Anwendungsfall weiter bestimmt. Die Aufgabe der Zusammenführung mehrerer Datenquellen wurde hierbei nicht explizit adressiert, sondern implizit als Aufgabe der Datenaufnahme angenommen. [CONR24]

Ein weiterer Ansatz aus der Sichtweise von hochskaligen Produktionen mit mehreren Standorten nach O'DONOVAN beschreibt die Datenqualität als Teil der Datenaufnahme. Die Aufnahme geschieht in mehreren Schritten, wobei für Zeitdatenreihen zunächst diese bei der Aufnahme bereits koordiniert werden, um nur bestimmte Zeitbereiche, definiert durch ein zentrales Steuersystem (»Site Manager«), zuzulassen. In diesem Schritt werden Daten nach Möglichkeit bereits direkt im zentralen Speicher auf andere Daten gematcht. Ein späterer Schritt befasst sich dann mit der Vorverarbeitung jener Daten, bei der die Transformation und Aggregationsstufen im Vordergrund stehen. Ein optionaler Schritt findet im Anschluss statt für weitere Vorverarbeitung mit Wissen von Expert*innen. Nach diesen Schritten können Modelle direkt auf die Daten und ihre Aggregationsebenen zugreifen. [O'15]

Ein wieder anderer Ansatz stellt die explorative Datenanalyse und Vorverarbeitung sowie Zusammenführung der Zeitdatenreihen als einen gemeinsamen Schritt dar. Dabei ist der Fokus auf dem Austausch der Informationen zwischen den Beteiligten Personen aus dem Data Science- und dem Engineering-Team. Dabei stellen die Personen aus dem Data Science-Team Probleme in den Daten fest und beheben diese zusammen mit dem anderen Team, wenn sie diese Probleme als ausreichend kritisch bewerten. Im Anschluss werden die Kerneigenschaften extrahiert, welche durch das Engineering-Team überprüft und freigegeben werden. Dies integriert dadurch die enge Zusammenarbeit der Teams, vergrößert die Komplexität der Vorverarbeitung allerdings auch weiter. [CARB19]

Eine weitere verbreitete Darstellung der Schritte ist die Abfolge nach BISWAS, dargestellt in Abbildung 2.16. Die Vorverarbeitung von Daten ist hier der erste Schritt der Kette. In den Unteraufgaben steht auch hier die Aufnahme von Daten vorne, wobei der letzte Schritt darin die rohe Integration von Datenreihen ist, auf denen dann die Schritte der Vorbereitung stattfinden. Das Extrahieren von Kerneigenschaften ist hier bereits Teil der eigentlichen Modellbildung, erlaubt aber Rücksprünge zu den vorherigen Schritten. Im Weiteren Verlauf ihrer Arbeit beschreiben die Autoren, dass die ersten Schritte in kleinen und großen Projekten jeweils die dominante Rolle und größten Einfluss auf die Ergebnisse haben. [BISW22]

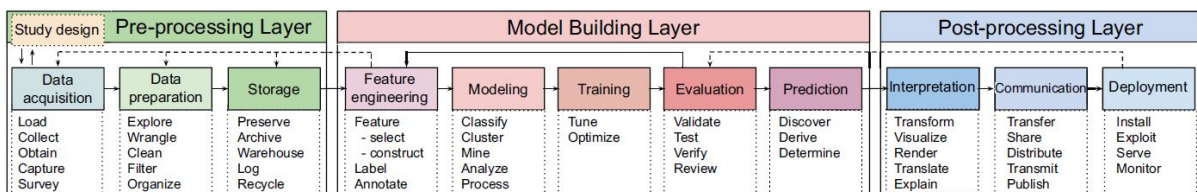


Abbildung 2.16: Abfolge der Schritte in der Data Science, unterteilt in die Hauptbereiche und ihre Aufgaben darin. (Entnommen aus [BISW22])

Aus den verschiedenen Abläufen und Vorgehen für den Einsatz von datenbasierten Modellen und Analysen ist erkennbar, dass Kernelemente allen diesen Ansätzen gemein sind. So beschreiben alle Varianten die Sicherstellung der Datenqualität und Integration verschiedener Datenquellen als essenziell und die Expertise aus dem Kontext der Daten als notwendig. Eine leichte Unterscheidung besteht bei den vorgestellten Ansätzen, ob die Zusammenführung von Datenquellen Teil der Vorverarbeitung oder ein separater Schritt davor ist. Für diese Arbeit wird die Zusammenführung auf einer strukturellen Ebene als Integrationsschritt am Anfang verstanden, bei dem die Zeitdatenreihen in einen gemeinsamen Datenspeicher gegeben werden. Weitere Betrachtungen zu den Zeitdatenreihen und die Prüfung ihrer Synchronizität ist dann wiederum Teil der Datenvorverarbeitung. Diese Definition ist daher auch im Einklang mit der Darstellung in Abbildung 1.2.

Zur Unterscheidung von Zeitdatenreihen und der Datenbasis je nach Ort in der Pipeline (Abbildung 2.17) ist eine angepasste Markierung der Datenbasis möglich. So beschreibt $\mathfrak{Z}^{Pre}$ die Datenbasis vor dem Eintritt in die Integration und Synchronisierung, während $\mathfrak{Z}^{Post}$ den Datensatz bezeichnet, der an die Modellierung weitergegeben wird. Für die Bezeichnung der Datenbasis innerhalb der Pipeline, oder wenn die Unterscheidung irrelevant ist, wird die Notation $\mathfrak{Z}$ genutzt.

2.6.2 Untersuchung der Datenqualität

Data Quality Assessment

Die Qualität von Daten ist entscheidend für die Nutzbarkeit und die Analysen, die mit diesen Daten gemacht werden [REDM18]. Daher ist eine Untersuchung der Datenqualität (engl. »Data Quality Assessment«) notwendig. Dabei lassen sich die Aspekte der Untersuchung der Qualität auf verschiedene Arten unterteilen. Aktuelle Standards wie ISO 25012 und ISO 8000 beschreiben dabei Datenqualität allgemein aus einer Software-Perspektive und definieren dabei Untersuchungsmethoden und Qualitätsmetriken. Die grundsätzliche Unterscheidung der Metriken ist dabei zwischen inhärenten Metriken (Konsistenz, Kreditibilität, Genauigkeit) und systemabhängigen Metriken (Verfügbarkeit, Wiederherstellbarkeit). Alle weiteren Metriken, die sich darin nicht scharf trennen lassen (Compliance, Effizienz, Verständlichkeit) sind Teil des Bereichs dazwischen. [TECH22a, TECH19a]

Die Untersuchung nach PIPINO nutzt dabei objektive Untersuchungsmethoden, als auch subjektive Untersuchungsmethoden. Daten sind erst als hochqualitativ zu bewerten, wenn für alle Eigenschaften die objektiven und subjektiven Methoden dies entsprechend bewerten. Auf diese Art lässt sich verhindern, dass ein nach objektiven Methoden guter Datensatz dennoch bei den nutzenden Personen erst später eine andere Einschätzung und damit Verzögerung verursacht [PIPI02]. Für die Untersuchung der Datenqualität sind dabei übergreifend eine Reihe von Dimensionen zur Bewertung definiert. Diese umfassen organisatorische Dimensionen wie die Regelung von Zugriffen und Manipulationssicherheit, aufgabenorientierte Dimensionen wie die Interpretierbarkeit und ausreichender Umfang für die Aufgabe, zuletzt auch Dimensionen zur technischen Qualität der Daten. Diese Dimensionen wurden erstmals für Informationen definiert [KAHN02], später aber auch auf Daten übertragen. Die zentrale Dimension für die technische Betrachtung der Datenqualität ist dabei die *Free-of-Error* Dimension, welche die Korrektheit und Verlässlichkeit der Daten beschreibt. Die Dimension *Objectivity* ist nicht direkt adressiert, allerdings wird die Datenvorverarbeitung von Menschen durchgeführt, sodass ein Bias darin nicht ausgeschlossen ist. Im Weiteren ist, wenn nicht anders angegeben, die *Free-of-Error* Dimension betrachtet.

Auch existieren einige Taxonomien, um die Probleme der Datenqualität entsprechend zu beschreiben. Diese Taxonomien beschreiben dabei hierarchisch, welche Probleme beobachtet sind und welche möglichen Methoden dabei helfen können, diese Probleme zu adressieren. Dabei ist bei vielen der Probleme hier die Lösung, eine Person mit Domänenwissen manuell eingreifen zu lassen. Auch sind multivariate Datensätze aus mehreren Quellen nur begrenzt betrachtet. So existiert eine spatiale Ungenauigkeit, aber keine temporale Ungenauigkeit in der Taxonomie [KIM03, OLIV05]. Die Verbindung dieser Ungenauigkeiten

existiert, wie in Abschnitt 2.5.3 gezeigt, ist in den Taxonomien allerdings nur indirekt durch die Auswirkungen adressiert.

Um die Probleme der Datenqualität in der Vorverarbeitung zu adressieren, existieren Lösungsansätze. Der Großteil der Lösungen kommt dabei aus den Domänen der jeweiligen Anwendung selbst, da spezifisches Domänenwissen benötigt wird [EHRL22]. Diese Ansätze und Methoden untersuchen dabei die Zeitdatenreihen und wenden Korrekturen an. Allerdings sind die Effekte in den Daten und die Korrekturen nicht unabhängig voneinander und beeinflussen daher gegenseitig ihre Güte und damit auch die Ergebnisqualität von datenbasierten Modellen. Dies hat FRYE untersucht und beschreibt, dass die Auswahl der genauen Methoden und die Reihenfolge der Anwendung eben jener einen maßgeblichen Einfluss auf das Ergebnis von Datenanalysen, in seinem Fall Klassifikation- und Regressionsaufgaben, hat. Auf Basis von Eingangsparametern und einer Analyse der Eingangsdaten erstellt ein Entscheidungssystem Vorschläge für die Pipelines zur Datenvorverarbeitung. Der Aufbau einer solchen Pipeline ist dabei in Abbildung 2.17 dargestellt. Hierbei stehen die Anforderungen und eine eingehende Prüfung der Daten am Anfang, wonach die Integration und Synchronisierung als erster zentraler Schritt vorgenommen wird. Nach dieser Synchronisierung findet ein Quality Check (orange) statt, bevor die Pipeline mit den von FRYE betrachteten Schritten mit Auswahl und Reihenfolge durchgeführt werden (blau). Am Ende der Schritte steht ein vorverarbeiteter Datensatz, der bereit ist für die Modellierungsschritte [FRYE22]. Der Quality Check nach der Synchronisierung umfasst dabei lediglich die Prüfung, ob die Struktur gemäß der Abtastrate (engl. »Sampling Rate«) korrekt ist sowie Start- und Endzeitpunkte der integrierten Datensätze identisch sind [FRYE20]. Weiterführende Metriken zur Überprüfung sind nicht gezeigt und die Synchronizität der Daten im weiteren Verlauf als gegeben angenommen.

2.6.3 Datenqualität und Asynchronizität

Data Quality and Asynchronicity

Eine Zusammenfassung von möglichen Problemen der Datenqualität ist in Tabelle 2.1 dargestellt. Dies ist eine Aggregation aus den beschriebenen Vorarbeiten [FRYE22] und Taxonomien [KIM03, OLIV05] mit Fokus auf die Schritte in der Vorverarbeitung, bevor eine Modellierung oder Analyse stattfindet. Diese Vorarbeiten betrachten die Probleme aus verschiedenen Perspektiven, sodass die Aggregation ein vollständiges Bild in dieser Arbeit angibt. Das Datenproblem der Asynchronizität ist in der folgenden Darstellung ausgeklammert, da es in den vorangegangenen Abschnitten bereits beschrieben wurde.

Datenproblem	Einzelne Datenreihe	Multiple Datenreihen	Einfluss Asynchronizität
Fehlende Werte	•	•	•
Rauschen	•		
Ausreißer / Intervallverletzung	•		
Datenformate / Syntaxverletzung	•	•	
Messgrößen und Skalierung		•	
Dopplungen	•	•	•
Unplausible & Inadäquate Werte	•		
Inkonsistenz zu Entitäten		•	•
Aggregationsebenen		•	
Verletzung der Domänenregeln	•	•	•
Uneindeutige Daten / Fehlender Kontext	•	•	
(Asynchronizität)		•	-

Tabelle 2.1: Datenprobleme, die in der Datenvorverarbeitung adressiert werden. Die Markierungen zeigen, welche bei einzelnen Datenreihen auftreten und welche bei der Betrachtung mehrerer Datenreihen. Eine Einschätzung des Einflusses der Asynchronizität auf diese Probleme und ihrer Korrektur ist rechts markiert. Da der Einfluss von Asynchronizität auf sich selbst nicht bewertbar ist, ist diese Zelle mit »-« markiert. (Aggregiert aus [FRYE22, KIM03, OLIV05])

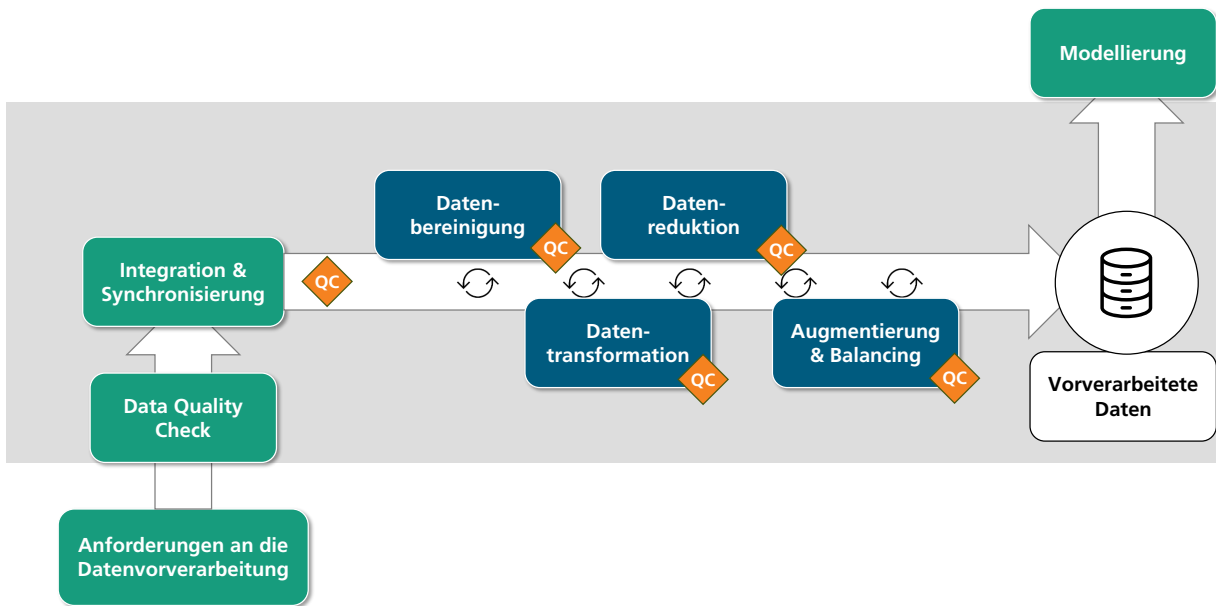


Abbildung 2.17: Die Pipeline der Datenvorverarbeitung nach FRYE mit notwendigem Einbezug von Quality Checks (QC). (Basierend auf [FRYE20] und Anpassungen aus [FRYE22])

Fehlende Werte

Eine Zeitdatenreihe enthält idealerweise für den Betrachtungszeitraum durchgehend Werte in der gegebenen Frequenz. Fehlen nun Werte zu einem erwarteten Zeitpunkt t_k , so ist dies ein fehlender Wert im Sinne der Datenqualität.

$$\{\dots, (t_{k-1}, v_{k-1}), (t_{k+1}, v_{k+1}), \dots\} \subset Z \wedge (t_k, \epsilon) \in Z^\epsilon \quad (2.38)$$

Dieses Problem kann innerhalb einer einzelnen Datenreihe beobachtet werden, sodass die Gegenmaßnahmen sich mit der Bestimmung des entsprechenden ϵ befassen. Auf eine Menge von mehreren Zeitdatenreihen kann dies neben einzelnen fehlenden Werten der Integration auch bedeuten, dass eine Zeitdatenreihe Z_1 nicht den gleichen Zeitraum wie eine Zeitdatenreihe Z_2 beschreibt. Dies kann daher einzelne Werte in der Zuordnung oder komplette Subsequenzen betreffen. Unter Nutzung der Definition von $\hat{U}$ ergibt sich daraus

$$\exists_l \overset{\circ}{U} \mid \forall(t_i)(\exists(t_i, v_i^m, \epsilon)) \quad (2.39)$$

für fehlende Subsequenzen. Für einzelne fehlende Werte lässt sich dies für ${}_lZ$ analog zu 2.38 darstellen.

Rauschen

In technischen Geräten ist ein Rauschen in den Signaldaten üblich. In der Modellierung kann dieses einfach als eine Rauschfunktion auf die Zeitdatenreihen addiert werden. Für die Darstellung ergibt sich dadurch

$$(t_i, v_i) = (t_i, vr_i + r(t)), \quad (2.40)$$

wobei vr_i das angenommene rauschfreie Signal ist, ergänzt um eine Rauschfunktion $r(t)$. In der Praxis kann für dieses Rauschsignal ein additives weißes Gauß'sches Rauschen oder ein Poisson-Rauschen angenommen werden [GHAZ23]. Da dieses Rauschen innerhalb einer einzelnen Datenreihe relevant ist, aber nicht datenreihenübergreifende Eigenschaften hat, wird es bei der Integration zunächst von der Betrachtung ausgelassen.

Ausreißer / Intervallverletzung

Datenpunkte, welche auffällig außerhalb eines definierten Wertebereichs liegen, sind als Ausreißer bezeichnet. Dabei ist der Intervallbereich lediglich strukturell vorgegeben, also bezieht kein Wissen um den Kontext oder Domänenwissen ein. Bei der Betrachtung von Ausreißern ist daher oftmals eine Analyse verbunden, zur Klassifikation, ob dieser Ausreißer entfernt werden muss oder selbst eine Relevanz mit sich bringt [BG21]. Für die Definitionen ergibt sich daraus

$$\forall(t_i, v_i) \in \{(t_i, v_i) \mid v_i > g_{upper} \vee v_i < g_{lower}\} \implies (t_i, v_i) \in \mathbb{O}_{Intervall}, \quad (2.41)$$

wobei g_{upper} und g_{lower} die obere und untere Grenze des erlaubten Intervalls angeben. Damit ist $\mathbb{O}_{Intervall}$ die Menge aller Ausreißer in der Datenreihe. Da die Ausreißer zunächst keine Relevanz für die datenreihenübergreifenden Eigenschaften haben, sind diese bei der Integration zunächst ausgelassen.

Datenformate / Syntaxverletzung

Die Darstellung von Werten kann je nach Aufnahmesystem oder lokalen Eigenschaften abweichen. Dies kann beispielsweise die Darstellung nicht-ganzzahliger Werte darstellen, insbesondere mit Blick auf Trennzeichen in verschiedenen technischen und sprachlichen Gebräuchen. Ebenfalls die Darstellung von Grenzfällen wie der Unendlichkeit ist hier zu beachten. Dieser Effekt kann innerhalb einer Datenreihe selbst vorkommen oder sich grundsätzlich zwischen mehreren Datenreihen beschreiben. Für die Beschreibung derselben Bedeutung bei verschiedenen syntaktischen Ausdrücken ist die Identität $\equiv$ definiert. Für Datenpunkte aus verschiedenen Systemen mit unterschiedlichen syntaktischen Darstellungen kann dieser beispielsweise folgendermaßen eingesetzt werden.

$$1,024.08 \equiv 1024.08 \equiv 1.024,08$$

$$Inf \equiv \infty$$

$$2024-04-20T04:20:01.200 \equiv 1713586801.2 \equiv 20.04.2024\ 04:20:01.2$$

In diesem Beispiel ist oben die Zahldarstellung je nach Lokalisierung anders syntaktisch dargestellt, beschreibt dabei aber den identischen Wert. Im Beispiel darunter ist die Darstellung einer Unendlichkeit ebenfalls syntaktisch verschieden, aber inhaltlich identisch. Die letzte Zeile zeigt die Identität einer Zeitangabe in verschiedenen Darstellungen, welche für implizit in dieser Arbeit verwendet wird, um Zeitdaten zu interpretieren.

Messgrößen und Skalierung

Bei der Betrachtung mehrerer Zeitdatenreihen ist es möglich, dass zwei Datenreihen verschiedene Wertebereiche oder Messgrößen verwenden. Für die Vorverarbeitung ist die Normalisierung dieser Datenreihen möglicherweise notwendig für konsistente Analysen. Dies beschreibt beispielsweise die Anpassung der Messgrößen in Einheiten auf die gleiche Norm. In Bezug auf die Zeitdatenreihen beschreibt das hierbei

$$z_i^l = (t_i, v_i^l \times k^l), k^l \in \mathbb{R}_{\neq 0}, \quad (2.42)$$

wobei k^l den Vektor von konstanten Skalierungskoeffizienten für die Zeitdatenreihen beschreibt. Beispielsweise kann eine Normalisierung von Millimeter auf Meter mit einem Eintrag von 1000 in dem Vektor erreicht werden. Die Asynchronizität selbst hat keinen direkten Einfluss auf diese Störgröße, allerdings ist diese Normalisierung relevant bei der Formalisierung von Wissen, sodass dies in Abschnitt 4.2.3 aufgegriffen wird.

Dopplungen

Doppelte Werte in den Zeitdatenreihen beinhalten keinen zusätzlichen Wert. Sie sind redundant und verbrauchen zusätzlichen Speicherplatz. Für eine einzelne Zeitdatenreihe sind diese einfach identifizierbar durch

$$\begin{aligned} \exists z_i = (t_i, v_i) \wedge \exists z_j = (t_j, v_j) \mid t_i = t_j \wedge v_i = v_j \\ \implies z_i, z_j \in \mathbb{O}_{Dopplung}, \end{aligned} \quad (2.43)$$

wobei $\mathbb{O}_{Dopplung}$ die Menge der doppelten Werte angibt. Sind Zeitwert und Datenwert jeweils identisch ($v_i = v_j$) kann eines der beiden Elemente problemlos entfernt werden. Unterscheiden sich die Datenwerte hingegen muss entschieden werden, welcher Wert beibehalten werden soll.

Im Falle von multiplen Datenreihen, die integriert werden, ist dieser Schritt der Entfernung notwendig, da in dem jeweiligen Zeitdatenreihenvektor z_i^l nur ein Wert vorgesehen ist in der Zuordnung. Wird nun eine Synchronisierung von mehreren Zeitdatenreihen durchgeführt, kann es passieren, dass eine Neu-Zuweisung der Datenwerte zu Zeitwerten eine Dopplung verursacht. Dadurch kann die eigentliche Asynchronizität und deren Behebung das Problem der Dopplungen verschärfen.

Unplausible und Inadäquate Werte

Während die Definition der Ausreißer $\mathbb{O}_{Intervall}$ alleine auf die strukturellen Eigenschaften der Datenreihe und gegebenen Grenzen schaut, werden bei der Prüfung der Plausibilität

auch Informationen und das Wissen zu dem Aufbau der Datenaufnahme einbezogen. Dies umfasst einerseits Grenzwerte, die sich pro Datenpunkt überprüfen lassen (g_{upper} und g_{lower} analog zu den Ausreißer-Grenzwerten g_{upper} und g_{lower}). Andererseits sind auch Verläufe und Sprünge in den Zeitdatenreihen, welche aus dem Wissen um Zusammenhänge und Wahrscheinlichkeiten als nicht plausibel ergeben Teil der Betrachtung. Beispielsweise kann bei einem Temperatursensor ein statischer Plausibilitätsbereich festgelegt werden, und schnelle Sprünge der Datenwerte eingeschränkt werden.

Für den Fall der einzeln überprüfbaren Werte mittels statischer Grenzwerte ergibt sich analog zu den Ausreißern die Definition zu:

$$\begin{aligned} \forall (t_i, v_i) \in \{(t_i, v_i) \mid v_i > g_{upper} \vee v_i < g_{lower}\} \\ \implies (t_i, v_i) \in \mathbb{O}_{Unplausibel} \end{aligned} \quad (2.44)$$

Für die Betrachtung der unplausiblen Sprünge und Veränderungen lassen sich die plausiblen Wertbereiche nach einem Datenpunkt relativ als $g_{plaus_{low}}$ für die untere Grenze und $g_{plaus_{high}}$ angeben. Daraus ergibt sich

$$\begin{aligned} (t_{i+1}, v_{i+1}) \mid v_{i+1} \notin [v_i - g_{plaus_{low}}, v_i + g_{plaus_{high}}] \\ \implies (t_{i+1}, v_{i+1}) \in \mathbb{O}_{Unplausibel} \end{aligned} \quad (2.45)$$

für Betrachtungen, in denen Plausibilität von einem zum anderen Wert geprüft werden kann. Für komplexere Plausibilitätsanalysen kann dann eine Erweiterung der Definition auf Subsequenzen $\overset{\circ}{U} \subseteq \mathbb{O}_{Unplausibel}$ geschehen.

Theoretisch hat eine Asynchronizität im Sinne von Gangfehlern hier einen Einfluss, da eine Verschiebung auf einen Punkt der Zeitdatenreihe die Grenze der plausiblen Wertveränderung verschieben kann. Allerdings wird hier der Gangfehler alleine zwischen zwei einzelnen Zeitdatenpunkten als ausreichen gering eingeschätzt, als dass ein Einfluss auf die Plausibilität direkt erkennbar ist.

Inkonsistenz zu Entitäten

Existiert eine einzelne Datenreihe zur Beschreibung einer Entität, so ist diese per Definition eindeutig, da keine gegenteiligen Daten vorliegen. Bei der Betrachtung von multiplen Datenreihen wiederum besteht die Möglichkeit, dass mehrere Datenreihen dieselbe Entität beschreiben, beispielsweise dasselbe Sensorsignal. Daraus ergeben sich zwei mögliche Fälle. Im ersten Fall existieren zwei Datenreihen, die identisch sind zu einer Entität. In diesem Fall ist dies ein besonderer Fall der Dopplungen (2.43), bei dem nicht einzelne Werte, sondern komplette Reihen Teil von $\mathbb{O}_{Dopplung}$ sind.

Im zweiten Fall hingegen beschreiben beide Datenreihen dieselbe Entität, jedoch unterscheiden sich die Werte:

$$\begin{aligned} \exists z_i^l = (t_i, v_i^l) \wedge \exists z_j^k = (t_j, v_j^k) \mid t_i = t_j \wedge v_i^l = E_{Ident}^{l,k}(v_j^k) \neq v_j^k \\ \implies z_i^l, z_j^k \in \mathbb{O}_{Inkonsistent} \end{aligned} \quad (2.46)$$

Dabei beschreibt $E_{Ident}^{l,k}(v_j^k)$ die Erwartung des Wertes von v_j^k aus dem Wissen der identischen Entität, also denselben Wert v_i^l . Dabei ist es möglich, dass zwei Zeitdatenreihen zu einem Zeitpunkt oder einer Subsequenz eine Dopplung und für den übrigen Teil eine Inkonsistenz beschreiben. In der Konsequenz bedeutet dieses Datenproblem, dass für Analysen eine Entscheidung getroffen werden muss, welche dieser Zeitdatenreihen als valide und geeignet genutzt werden soll, mit entsprechenden Einflüssen auf die Qualität der Auswertung. Die Identifikation von Inkonsistenz zu Entitäten wie oben beschrieben basiert dabei auf der Annahme, dass die betrachteten Zeitdatenreihen die gleichen Zeitwerte (t_x) nutzen. Ist allerdings die Aufnahme beispielsweise etwas versetzt, beschrieben durch

$$\forall t_i \neg (\exists (t_i, v_i^l) \wedge \exists (t_i, v_j^k)), \quad (2.47)$$

hält diese Annahme nicht. Dies beschreibt eine konstante Verschiebung, welche durch die Gegebenheiten der Datenaufnahmen auftreten kann.

Gangfehler und Taktversatz haben hier weitere Einflüsse auf die Existenz und Identifizierbarkeit der durchgehenden Dopplungen und Inkonsistenzen. Ist der Taktversatz, bezogen

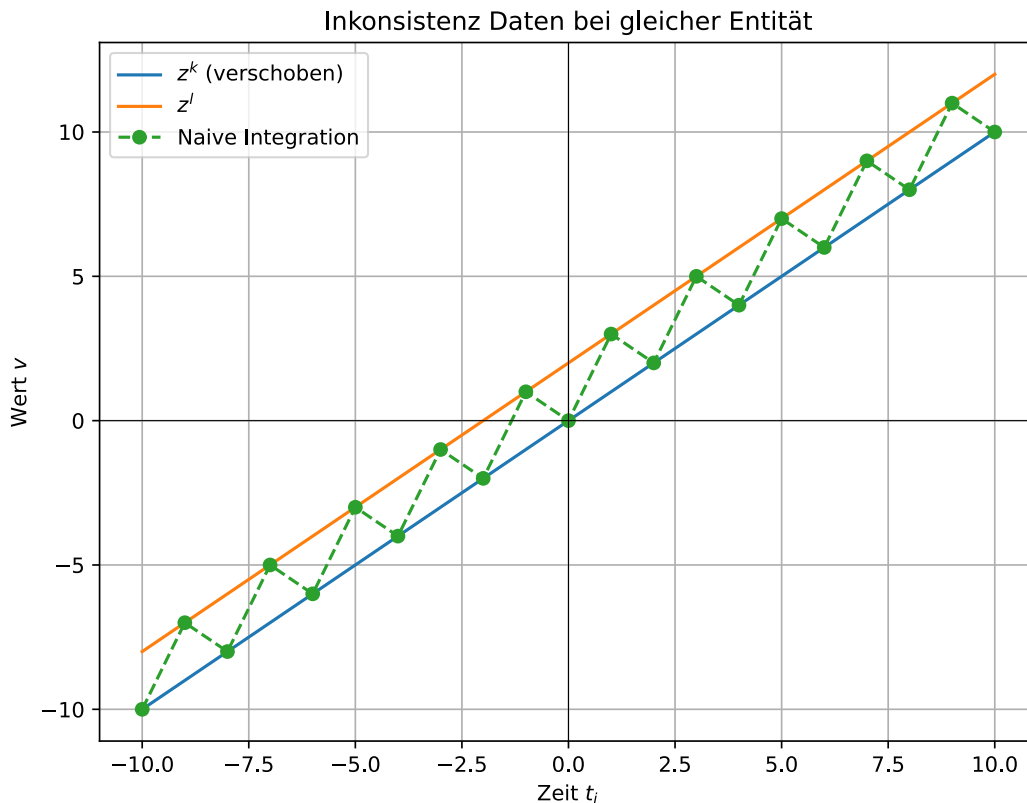


Abbildung 2.18: Zwei Zeitdatenreihen für die gleiche Entität, allerdings versetzt zu einander. Punkte entsprechen Elementen der Zeitdatenreihen, die orange und blaue Linie dienen der besseren Lesbarkeit.

auf zwei Zeitdatenreihen, kein Vielfaches der Abtastrate, so ergeben sich keine direkt erkennbaren Probleme nach $\mathbb{O}_{Dopplung}$ oder $\mathbb{O}_{Inkonsistent}$. Bei der direkten (»naïven«) Nutzung der Daten für die entsprechende Entität ist dann ein Springen der Werte zwischen den Zeitdatenreihen erkennbar, dargestellt in Abbildung 2.18. Bei der Betrachtung dieser Abbildung ist nicht zu erkennen, ob dieser Effekt aus einer systematischen Verschiebung (wie einem Versatz der Datenaufnahme) oder einem Taktversatz des aufnehmenden Systems stammt. Ist hingegen der Taktversatz ein Vielfaches der Abtastrate, so wird dieser Einfluss direkt als Inkonsistenz in $\mathbb{O}_{Inkonsistent}$ identifiziert. Wenn die Daten beide korrekt, nur durch den Taktversatz verschoben sind, wird eine Zeitdatenreihe mit eigentlich korrekten Daten eventuell fälschlicherweise als inkonsistent klassifiziert.

Die Auswirkung des Gangfehlers ist in Abbildung 2.19 dargestellt. Die beiden Zeitdaten-

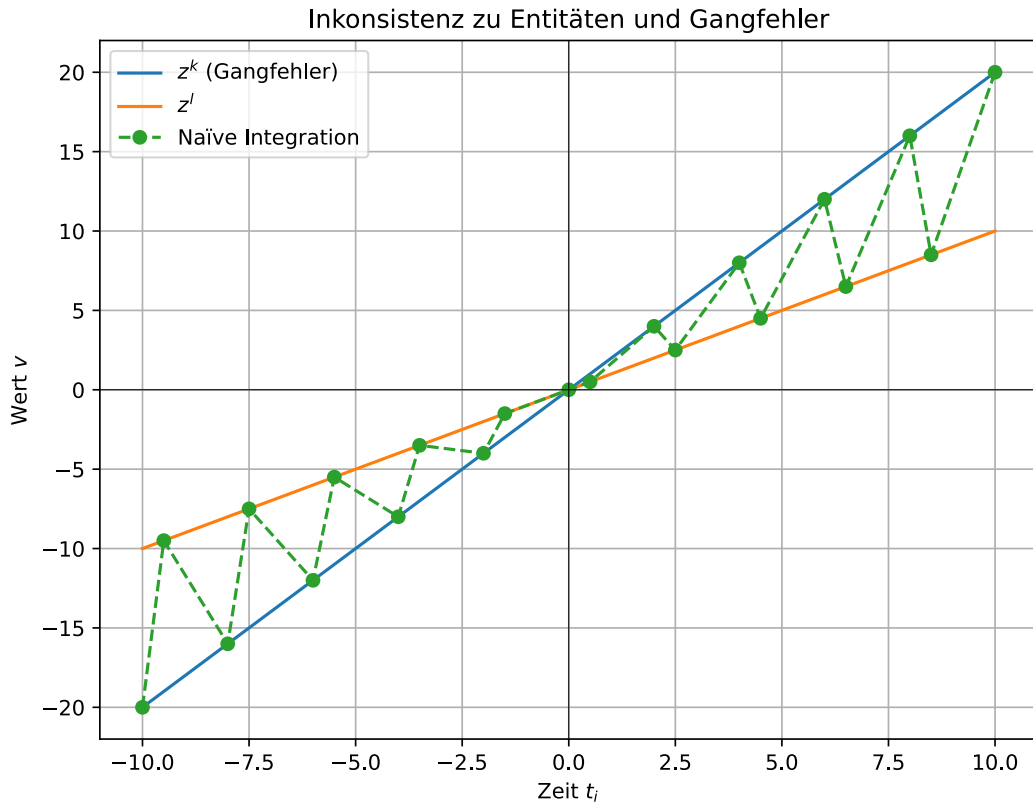


Abbildung 2.19: Zwei Zeitdatenreihen zur gleichen Entität, wobei eine unter dem Effekt der Gangfehler leidet.

reihen beschreiben die gleiche Entität, jedoch durch den Gangfehler ist die Abtaste nicht konstant (grüne Punkte). Bei direkter Nutzung der Daten zeigen sich daher geringer und größer werdende Schwankungen in den Werten, die in der Realität nicht existieren. Möglicherweise existieren einzelne Zeitdatenreihenelemente, bei denen es zudem zu einer Dopplung oder Inkonsistenz wie oben definiert kommt, allerdings ist dies zunächst nicht von Effekten außerhalb der Asynchronizität zu unterscheiden.

Insgesamt ist damit klar, dass Asynchronizität einen Einfluss auf Zeitdatenreihen für die Identifikation und Klassifikation von Inkonsistenzen zu einer Entität hat.

Aggregationsebenen

Bei der Aufnahme von Daten ist es möglich, dass Aggregationen stattfinden vor der weitergehenden Analyse. Dies umfasst beispielsweise eine Rohdatenaufnahme mit hoher Frequenz, die bei der Weitergabe gemittelt in niedriger Frequenz wird. Ebenfalls besteht die Möglichkeit zur inhaltlichen Zusammenfassung mehrerer Zeitdatenreihen, beispielsweise eine Gesamttemperatur innerhalb einer Anlage statt die einzelnen Sensorwerte. Dadurch kann auch die Menge der Daten reduziert werden, da nicht für jede Art Analyse die maximale Auflösung benötigt wird [BAUC25]. Daher tritt der Effekt der Aggregationsebenen bei multiplen Datenreihen möglicherweise in Erscheinung.

Derartige Aggregationen können Datenprobleme verschleiern, da beispielsweise eine Inkonsistenz oder Ausreißer bei der Aggregation behandelt wurden, aber die Güte dieser Behandlung unklar ist. Auch ist die Zuordnung eines gemittelten Wertes für eine Zeitspanne ein Verlust von Daten, da nun einer Subsequenz vollständig ein einzelner Wert zugeordnet wird

$${}_aZ = \{z_i^a \mid agg(\overset{\circ}{U}_l^{m(i),n(i)}), \forall l \in |\mathcal{Z}|\}, \quad (2.48)$$

wobei $agg()$ die Aggregationsfunktion beschreibt, welche eine oder mehrere verschiedenen Subsequenzen nutzt für die Berechnung. Da ein aggregierter Wert auch mehreren Zeitdaten zugeordnet werden kann, ist die Subsequenz-Betrachtung hierbei abhängig von i . Diese Definition impliziert die Existenz der ursprünglichen Zeitdatenreihen, aus denen die Aggregationen gebildet werden. Sollten diese bei einer Betrachtung nicht oder nicht vollständig vorliegen, sind die entsprechenden l in der Betrachtung nicht vorhanden, die Definition wie oben beschrieben bleibt davon aber unberührt.

Daraus ergeben sich Folgefragen bezüglich der Vergleichbarkeit von Zeitdatenreihen und die Ordnung in den Analysen. Diese Aggregationsfunktion kann dabei unter den Effekten der Asynchronizität leiden, allerdings im gleichen Maße wie die zugrunde liegenden Zeitdatenreihen, auf denen diese Aggregation beruht. Dadurch gleichen sich bei der direkten Nutzung diese Effekte grundsätzlich aus. Wird eine Synchronisierung der Daten vorgenommen, muss diese auch in der Aggregationsfunktion angewendet werden. Dadurch ist der Einfluss der Asynchronizität auf das Datenproblem der verschiedenen Aggregationsebenen nur ein indirektes Problem und daher nicht markiert in Tabelle 2.1.

Verletzung von Domänenregeln

Zur Identifizierung und von nicht plausiblen Werten ist der Rückgriff auf Informationen und Wissen zu Aufbau und sensorischen Eigenschaften beschrieben. Weiterführendes Wissen existiert für die Domäne, also dem Bereich der Anwendung. Aus diesem Wissen lassen sich Domänenregeln ableiten. Diese können einzelne Zeitdatenreihen oder multiple Zeitdatenreihen betreffen.

Im Fall singulärer Datenreihen ist dieses Datenproblem eng mit Ausreißern und nicht plausiblen Werten verbunden. Als Beispiel zum Verständnis sei hier ein Temperatursensor in einer Produktionsanlage angenommen. Dabei beschreibt das Problem der Ausreißer einen Wert, der aus Datensicht abseits der grundsätzlichen Wertebereiche liegt. Ist dieser Wert innerhalb des grundsätzlichen Wertebereichs, jedoch aus den Informationen zur Sensorik und Wissen zu Temperatursensorik nicht möglich, ist er unplausibel. Ist der Wert grundsätzlich technisch plausibel, ergibt jedoch im Bereich des konkreten Aufbaus und des laufenden Prozesses keinen Sinn, stellt dies eine Verletzung einer Domänenregel dar. Dies ist beispielsweise eine Temperaturkurve außerhalb des zugelassenen Arbeitsbereichs der Anlage.

Im Falle der multiplen Zeitdatenreihen kommt insbesondere Wissen zum Zusammenhang der Systeme und Komponenten hinzu, welche die Zeitdatenreihen generieren. Bezogen auf Zeitdatenreihen einer Produktion sind dies beispielsweise Bedingungen, dass eine Anlage bei geringem Pumpendruck (Zeitdatenreihe z^l) keinen hohen Messwert beim Durchflusssensor (Zeitdatenreihen z^j) haben kann. Diese Regeln können sich ebenfalls über mehrere Zeitdatenreihen erstrecken und benötigen eine Formalisierung und entsprechendes Wissen von Expert*innen. Daraus ergibt sich die Definition

$$\mathbb{P} = \{p_i \mid \text{Domäne} \models p_i\} \quad (2.49)$$

$$\mathbb{O}_{\text{Domäne}} = \{z_i^l \mid \exists p_j(z_i^l \perp p_j)\}, \quad (2.50)$$

wobei $\mathbb{P}$ die Menge aller Regeln beschreibt, die sich aus dem Wissen zur Domäne ableiten lassen.

Die Asynchronizität macht sich hier direkt bemerkbar, da Domänenregeln, welche auf eine zeitliche Komponente zurückgreifen (z.B. Gleichzeitigkeit), fälschlicherweise eine Zuordnung annehmen, und darauf basierend eine Bewertung durchführen. So können die

Zeitdatenreihen aus dem Beispiel oben bei einer Verschiebung durch Taktversatz bereits fälschlicherweise als Verletzung einer Domänenregel klassifiziert, und andere echte Verletzungen als korrekt eingestuft werden. Dadurch ergibt sich ein direkter Einfluss auf die Bewertung von Zeitdatenreihen bezüglich ihrer Domänenregeln durch die Asynchronizität.

Zuletzt sei erwähnt, dass Domänenregeln je nach Definition auch Sonderfälle zulassen. So sind in dem obigen Beispiel die Zeitdatenreihen trotz Widerspruch zur Domänenregel korrekt, wenn beispielsweise in einem Einrichtungsbetrieb manuell ohne die Anlagenpumpe entlang des Durchflusssensors (Bypass) Betriebsmittel eingefüllt werden. Auch hierfür ist das Wissen um die Situation, Aufbau und Auswirkungen notwendig zur korrekten Einordnung.

Uneindeutige Daten

Ein weiteres Problem von Daten entsteht durch fehlenden Kontext und Uneindeutigkeiten. Bei Daten mit Bezeichnungen oder Abkürzungen existiert dabei das Problem der Homonyme, also eine syntaktische Benennung, die allerdings semantisch verschiedene Bedeutungen haben kann. Beispielsweise kann bei der Beschreibung von Datenreihen in der Batteriezellfertigung die Angabe »Elek.« die Verläufe der Ströme in der Finalisierung, einer einzelnen Anlage oder eine Referenz auf den Elektrolyten bezeichnen. Existieren keine weiterführenden Informationen zu dieser Datenreihe ist sie nicht eindeutig und potenziell unbrauchbar.

Andere Beschreibungen von Datenreihen sind keine Homonyme, allerdings ohne weiterführenden Kontext ebenfalls nicht nutzbar. So ist die Bezeichnung »Schichtdicke« bei einer beidseitigen Beschichtung nicht eindeutig, ob nun die obere, die untere oder beide Seiten summiert gemeint sind. Im Fall von multiplen Datenreihen kann dieses Problem zusätzlich auftreten, wenn dort mehrere Datenreihen die gleiche Bezeichnung (aber verschiedene Werte) haben.

Diese Fehler treten daher bei singulären und multiplen Datenreihen auf, haben selbst aber keine Abhängigkeit von der Asynchronizität der zugrunde liegenden Zeitdatenreihen.

Abschließende Eigenschaften

Aus den oben beschriebenen Datenproblemen ergeben sich Eigenschaften. Bei den Datenproblemen sind Mengen von problematischen Zeitreihenelementen und Subsequenzen definiert

$$\mathbb{O} = \mathbb{O}_{Intervall} \cup \mathbb{O}_{Dopplung} \cup \mathbb{O}_{Unplausibel} \cup \mathbb{O}_{Inkonsistent} \cup \mathbb{O}_{Domaene}, \quad (2.51)$$

die gemäß ihrer Definition im Rahmen der Bereinigung entweder ganz oder teilweise Kandidaten zur Entfernung für $\mathfrak{Z}^{Post}$ sind. So reicht beispielsweise bei Elementen aus $\mathbb{O}_{Dopplung}$ die Entfernung aller Elemente bis auf eines pro Zeitpunkt aus.

Gleichzeitig ergibt sich aus der Diskussion der Datenprobleme, dass potenziell nicht alle Elemente in $\mathbb{O}$ als Kandidaten festgelegt wurden und zugleich andere Elemente aus $\mathfrak{Z}$ fehlen. Dies kann auf spezifische Eigenschaften wie einem leichten Versatz bei der Inkonsistenz zurückzuführen sein.

Asynchronizität von Zeitdatenreihen ist hier in signifikantem Ausmaß für fehlende Kandidaten und falsch detektierte Kandidaten in $\mathbb{O}$ verantwortlich. Damit bestimmt die Güte der Synchronizität der Zeitdatenreihen die Güte und den Aufwand bei der Bereinigung von Produktionsdaten [CHO22].

Für die Aufnahme von Daten in modernen Produktionssystemen kommen Systeme wie IIoT-Plattformen zum Einsatz. Die beschriebenen Datenprobleme werden bei diesen Systemen allerdings nur stark begrenzt betrachtet und bereinigt. Mehrere Anbieter bieten hierzu Möglichkeiten für die Skalierung und Ausreißererkennung an. Für die Domänenregeln und Intervallverletzungen existiert oftmals eine Möglichkeit zur Eingabe von entsprechenden Parametern, um Alarme zu generieren, allerdings keine Funktionen um die Daten selbst zu bereinigen. Die Asynchronizität ist bei den meisten Systemen nicht direkt berücksichtigt, und eine Behebung von Asynchronizität findet nicht statt [SCHM19]. In IIoT-Plattformen wie WinCC OA existieren allerdings Grenzwerte für die Asynchronizität bei der Aufnahme. Wird ein Zeitdatenreihenelement empfangen, welches mehr als eine definierte Zeitspanne von der Zeit des IIoT-Servers abweicht, wird das gesamte Element als fehlerhaft markiert [ETM 15].

2.7 Aufgabe der Synchronisierung von Zeitdatenreihen

Task of Time-Series Data Synchronization

Moderne Produktionen setzen auf datenbasierte Verfahren für die Analytik und Optimierung. Dabei steigen die Datenmengen und die Menge der Datenquellen weiter an. Dies führt dazu, dass spezielle Systeme und Methoden aus der hochskaligen Datenverarbeitung

(»Big Data«) in der Produktion Anwendung finden. Damit einher geht auch, dass manuelle Arbeiten wie eine manuelle Datenvorverarbeitung nicht mehr effizient genug sind und durch Methoden und Algorithmen ersetzt werden müssen [HEID24]. Die Integration eben jener Expertise der Menschen ist hierbei eine zentrale Herausforderung [WANG22].

Für die Qualität jeglicher Auswertungen und Berechnungen in der hochskaligen Datenverarbeitung ist dabei die Qualität der Daten entscheidend. Wie in Abschnitt 2.6.3 gezeigt, hat die Asynchronizität hierbei eine bedeutende Rolle und einen Einfluss auf eben jene Qualität. Daher ist eine Synchronisierung der Zeitdatenreihen notwendig.

2.7.1 Gleichzeitigkeit und Projektion

Simultaneity and Projection

Zur Beschreibung von Zeitpunkten wurde eine Notation für Zeitdatenreihen (t_i), die errechnete Zeit in einem System (C_s) sowie Zeitpunkte in der echten Zeit (T_α) eingeführt. Zur Vergleichbarkeit von Zeiten im Hinblick auf die echte Zeit wird im weiteren Verlauf daher die Notation $T(\cdot)$ genutzt, welche die echte Zeit beschreibt, zu der die jeweiligen Zeitargumente real aufgenommen wurden. Beschreiben also alle Darstellungen die gleiche echte Zeit, bedeutet dies:

$$T_\alpha = T(t_i) = T(C_s(t_x)) = T(C_s(T_\alpha)) = T(z_i) = T(TS_m) \quad (2.52)$$

Zur Vereinfachung der Schreibweise und des direkten Vergleichs von Zeitangaben wird daher der Vergleichsoperator $\doteq$ entsprechend definiert:

$$T_\alpha \doteq t_i \doteq C_s(t_x) \doteq (C_s(T_\alpha)) \doteq z_i \doteq TS_m \quad (2.53)$$

Aus der Betrachtung von Produktionsnetzwerken ergibt sich, dass mehr als ein System existiert, welches Zeitangaben für die entsprechenden Zeitdatenreihen erstellt. Um bei der Modellierung die verschiedenen Systeme C_n zu berücksichtigen, werden die Definitionen für den Gangfehler (2.5) und Taktversatz (2.3) entsprechend erweitert, wobei die Angabe n die Spezifizierung für das System C_n bezeichnet:

$$\delta^n(i) = 1 - \frac{\tau_{ideal}^n}{\tau_{real}^n(i, i+1)}, 0 \leq i < m \quad (2.54)$$

$$\Delta_{skew}^n = C_n(T_\alpha) - T_\alpha \quad (2.55)$$

Für die Beschreibung einer Zeit, die ein System C_n errechnet, ergibt sich daher

$$C_n(t_x) = C_n(t_0) + \sum_{i=0}^{x-1} \frac{\tau_{ideal}^n}{\tau_{real}^n(i, i+1)} + \Delta_{skew}^n + \Delta_{corr}^n(t_x), \quad (2.56)$$

wobei Δ_{corr}^n ein zeitabhängiger Korrekturparameter ist. In diesem kann beispielsweise eine Korrektur für Schaltsekunden (Abschnitt 2.2.2) modelliert werden.

Die eigentliche Synchronisierung der Zeitdatenreihen lässt sich nun beschreiben durch eine Projektionsfunktion $\Phi_{n,m}$, welche die errechneten Zeitangaben in System C_n auf eine Zeitangabe in System C_m projiziert.

$$\begin{aligned} \Phi_{n,m} : \langle C_n \rangle &\rightarrow \langle C_m \rangle \\ \Phi_{n,m}(C_n(t_x)) &= \begin{cases} C_m(t_y) & | \exists t_y : C_n(t_x) \stackrel{\circ}{=} C_m(t_y) \pm \tau_{Error}^{n,m} \\ \text{undefiniert} & \text{andernfalls} \end{cases}, \end{aligned} \quad (2.57)$$

wobei τ_{Error} den Bereich beschreibt, der sich ergibt durch die Ungenauigkeiten der Zyklen in der Zeitberechnung der Systeme. Da die ideale Frequenz sich unterscheiden kann, ergibt sich die Unsicherheit in Abhängigkeit von der größeren Zykluszeit. Daher gilt die Abschätzung:

$$\tau_{Error}^{n,m} \leq \frac{\max(\tau_{ideal}^n, \tau_{ideal}^m)}{2} \quad (2.58)$$

Kann eine solche Funktion gefunden werden für den Betrachtungsbereich für die Systeme C_n und C_m , so ist C_n **auf** C_m **synchronisierbar**. Dies bedeutet, dass die Zeitangaben

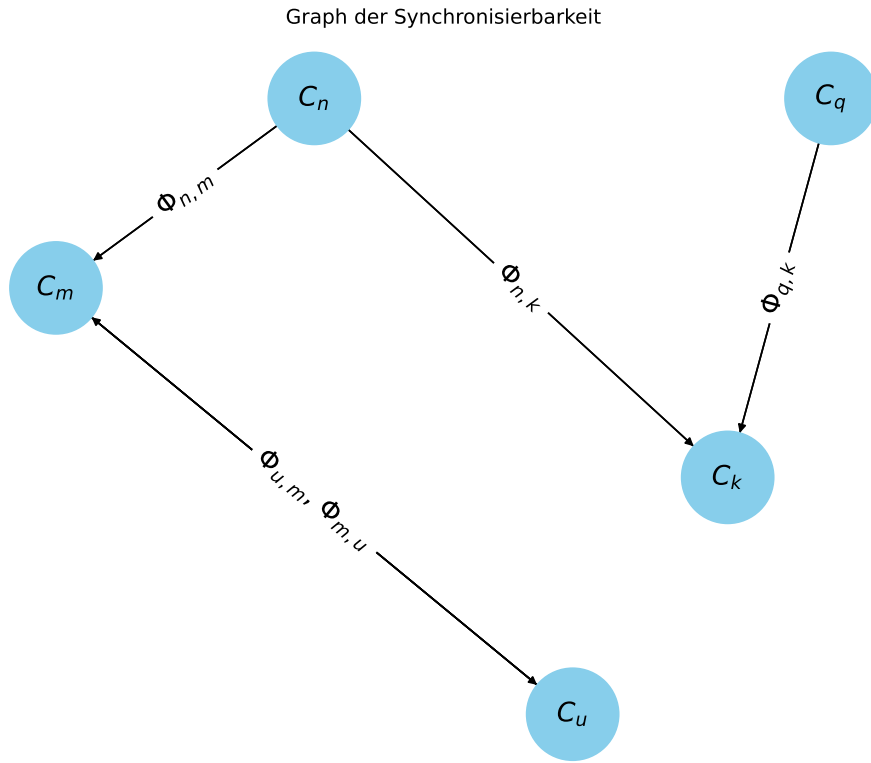


Abbildung 2.20: Beschreibung der Synchronisierbarkeit von Zeitangaben aus verschiedenen Systemen, dargestellt als Graph. Eine Kante entspricht einer verfügbaren Projektionsfunktion zur Synchronisierung auf das System des Zielknotens.

aus dem System C_n innerhalb des relevanten Zeitraums auf Zeitangaben des Systems C_m umgerechnet werden können [SCHM22]. Eine vollständige Umkehrabbildung ist nicht automatisch gegeben, da diese Projektion nicht bijektiv ist. Aus der Definition kann allerdings behelfsmäßig eine triviale Umkehrfunktion konstruiert werden:

$$\begin{aligned}
 \Phi_{n,m}(\langle C_n \rangle) &\subseteq \langle C_m \rangle \\
 \Phi_{n,m}^{-1} : \Phi_{n,m}(\langle C_n \rangle) &\rightarrow \langle C_n \rangle \\
 \forall t_y \in \Phi_{n,m}(\langle C_n \rangle) : \exists t_x \in \langle C_n \rangle \mid \Phi_{n,m}^{-1}(C_m(t_y)) &\doteq C_n(t_x) \pm \tau_{Error}^{n,m}
 \end{aligned} \tag{2.59}$$

Wird nun ein Netzwerk mit mehreren Systemen betrachtet, wie in Abbildung 2.20 dargestellt, so bedeutet dies, dass ein gerichteter Synchronisierungsgraph definiert werden kann. Dabei sind die Systeme C_s die Knoten und die definierten Projektionsfunktionen Φ die gerichteten Kanten. Idealerweise ist für jede Kombination von Knoten (Systemen) eine solche Projektionsfunktion vorhanden, sodass es sich um einen vollständigen gerichteten Graphen handelt. Wie in der Abbildung dargestellt, kann es aber auch sein, dass diese besondere Eigenschaft nicht gegeben ist. Stattdessen ist es möglich, dass Knoten existieren, die als Senke (keine ausgehenden Kanten) oder als Quelle (nur ausgehende Kanten) fungieren. Ebenfalls ist es möglich, dass der Graph nicht zusammenhängend ist.

Für die Synchronisierbarkeit der Zeitangaben eines Netzwerks beschreibt dieser Graph die entsprechenden Möglichkeiten. In einem vollständigen Graphen kann für jeden Zeitpunkt ein Tupel erstellt werden, welches die Zeitangaben der Systeme zu diesem Zeitpunkt benennt. Ist der Graph nicht vollständig und auch nicht zusammenhängend, so ist keine Verbindung der Zeitangaben im gesamten Netzwerk möglich. Eine Synchronisierbarkeit der Teilgraphen (Teilnetzwerke) ist eventuell möglich.

Für den Fall eines zusammenhängenden, nicht vollständigen Graphen sind Senken besonders relevant, da entlang der gerichteten Kanten andere Zeitangaben von Systemen auf eben jene Zeit des der Senke abgebildet wird. Zur Identifikation solcher Senken (»Synchronisierungsziel«) kann gemäß der üblichen Graphentheorie der transponierte Graph G_{sync}^T erstellt und mittels Breiten- oder Tiefensuche betrachtet werden als Erreichbarkeitsanalyse. Die daraus entstandene Liste der Knoten beschreibt übertragen alle Systeme, deren Zeiten auf dieses System projizierbar sind. Bei einer direkten Adjazenz ist eine entsprechende Projektionsfunktion Φ gegeben. Besteht diese nicht, ist die Projektion durch die Komposition der Projektionen $(\Phi_1 \circ \Phi_2)$ zu konstruieren.

C_n		C_m		C_u
0.0	→	3	→	0.2
0.2				0.4
0.4		4		0.6
0.6	→		→	0.9
0.8		5	→	1.1
1.0	→			1.2
1.1	→	6	→	1.3

Abbildung 2.21: Exemplarische Komposition von Projektionsfunktionen $\Phi_{m,u} \circ \Phi_{n,m}$ in rot. Die Spalten geben die Zeitangaben innerhalb des jeweiligen Systems an.

Im Falle von Kompositionen verringert sich möglicherweise die Genauigkeit der Synchronisierung. Dies ist exemplarisch in der Abbildung 2.21 dargestellt. Es existiert keine Projektion von C_n auf C_u , daher wird die Komposition $\Phi_{m,u} \circ \Phi_{n,m}$ verwendet. Die Auflösung der Zeitangaben in C_n ist allerdings höher als in C_m , sodass entweder nicht alle Werte aus C_n abgebildet werden oder mehrere auf die gleichen Ziele in C_m abbilden (zur besseren Lesbarkeit ist hier der erste Fall genutzt). Auch wenn die Auflösung der Zeitangaben möglicherweise vergleichbar zu C_n ist, beschränkt die geringere Auflösung in C_m die Auflösung in der Komposition. Wird diese Auflösung auf die Zyklendauer bezogen, ergibt sich für die Ungenauigkeit eine Abschätzung der Ungenauigkeit von $\Phi_{n,m} \circ \Phi_{m,u}$ zu

$$\tau_{Error}^{n,u} \leq \tau_{Error}^{n,m} + \tau_{Error}^{m,u}. \quad (2.60)$$

Eine mögliche Erweiterung des Synchronisierungsgraphs ist daher die Aufnahme der τ_{Error} als Kantengewichte. Bei der Erstellung von Kompositionen kann damit bei mehreren Optionen der Pfad durch den Graphen mit dem geringsten Gewicht gewählt werden.

2.7.2 Übertragung auf Zeitdatenreihen

Transfer to Time-Series Data

In der obigen Beschreibung ist die Projektion einer Zeitangabe aus einem System auf die Zeitangaben eines anderen Systems dargestellt. Übertragen auf die Definition der Zeitdatenreihen ergibt sich zunächst, dass eine Zeitdatenreihe ${}_lZ$ einem System C_s zugeordnet ist. Damit gilt für alle Zeitdatenelemente in ${}_lZ$:

$$z_i^l = (C_s(t_i), v_i) \quad , z_i^l \in {}_lZ$$

Eine Zuordnung einer Zeitdatenreihe ${}_lZ$ zu einem System C_s ist beschrieben durch:

$$C_s \succ {}_lZ \quad (2.61)$$

Analog zu den Definitionen aus den Systembetrachtungen können daher einige Definitionen übernommen und erweitert werden:

$$C_n(t_i) \doteq C_m(t_j) \implies z_i^l \doteq z_j^k, C_n \succ_l Z \wedge C_m \succ_k Z \quad (2.62)$$

$$\Phi_{n,m}(z_i^l) = (\Phi_{n,m}(C_n(t_i)), {}_l V_i) \quad (2.63)$$

$$\Phi_{n,m}^{-1}(z_j^k) = (\Phi_{n,m}^{-1}(C_m(t_j)), {}_k v_j^k) \quad (2.64)$$

Die Komposition $\Phi \circ \Phi$ gilt ebenfalls analog. Für die Betrachtung des Unsicherheitsfaktors τ_{Error} kann hier auf die Frequenz der Zeitdatenreihe (2.21) zurückgegriffen werden. Daraus ergibt sich für die Parameter von τ_{Error}

$$\begin{aligned} \tau_{ideal}^n &= f_{lZ}, C_n \succ_l Z \\ \tau_{real}^n(i, i+1) &= t_{i+1} - t_i, \end{aligned}$$

sodass die Definition von τ_{Error} selbst zunächst unverändert bleibt. Für die eigentliche Synchronisierung kann für Zeitdatenreihen somit auf das Finden einer Projektionsfunktion Φ zurückgegriffen werden, solange eine Zuordnung der Zeitdatenreihen zu einem System besteht. In der Praxis bedeutet dies die Zuordnung des Quellsystems eines Zeitstempels $TS_i \doteq C_s(t_i)$ auf die entsprechende Zeitdatenreihe ${}_l Z$.

2.7.3 Implikationen durch Produktionsnetzwerke

Implications from Production Networks

Wird nun ein Produktionsnetzwerk, wie in Abschnitt 2.4 beschrieben, betrachtet, ergeben sich weiterführende Implikationen auf die Zeitdatenreihen und ihre Synchronisierung. Aufnehmende Systeme, Edge Geräte, Server und weitere Systeme im Sinne des Fog Computing sind alles Systeme, die potenziell eigenständige Zeitberechnungen beinhalten. Bei den Zeitangaben ist daher die Information notwendig, aus welchem System diese Zeitangabe stammt. Gleichzeitig ist bei einem zentralen System, wo Datenreihen aus verschiedenen Quellen ankommen eine Vereinfachung, da hier eine gemeinsame Zeitberechnung für mehrere Datenreihen existiert. Allerdings beschreibt die Zeit bei Aufnahme in einem zentralen System $C_{zentral}$ die dortige Ankunft und nicht den ursprünglichen Aufnahmezeitpunkt in dem Aufnahmesystem $C_{ursprung}$. Da die Übertragung von Daten durch ein Netzwerk nicht instantan passiert, sondern Zeit benötigt (Abschnitt 2.3), gilt für die jeweiligen Zeitangaben $C_{zentral}(T_{Ankunft}) \neq C_{ursprung}(T_{Aufnahme})$.

Existieren für eine Datenreihe mehrere Systeme, welche auf dem Weg durch das Netzwerk Zeitangaben erfassen, ist dies beschrieben durch

$$C_{ursprung} \succ C_{edge} \succ \dots \succ C_{zentral} \succ {}_lZ, \quad (2.65)$$

wobei jedes System, welches eine Zeitangabe hinzufügt, in der Kette von $\succ$ aufgenommen ist in der Reihenfolge des Durchlaufs der Daten. Systeme, die selbst keine Zeitangaben erfassen, aber ebenfalls durchlaufen werden, sind ebenfalls in dieser Kette aufgenommen, tauchen in den Zeitdatenreihen allerdings nicht auf. Dadurch steht das Ursprungssystem an erster Stelle. Soll von den verfügbaren Zeitangaben eine bestimmte referenziert werden, wird die Kurzschreibweise gemäß 2.61 verwendet.

Dabei gilt grundsätzlich, dass zwischen allen Systemen für das Versenden, Übertragen und Empfangen eine positive Menge an Zeit vergeht. Dies kann in diesem Kontext beschrieben werden durch

$$C_{ursprung} \succ C_{zentral} \succ {}_lZ \implies C_{zentral}(T_{Ankunft}) \doteq C_{ursprung}(T_{Aufnahme}) + D_{Net}^{ursprung,zentral}, \quad (2.66)$$

wobei $D_{Net}^{ursprung,zentral}$ die benannte zeitliche Verzögerung durch das Produktionsnetzwerk ist. Grundsätzlich gilt hier $D_{Net}^{ursprung,zentral} = T_{Ankunft} - T_{Aufnahme}$, jedoch lässt sich so die Modellierung des Netzwerks beschreiben, anstatt auf die echte Zeit zurückzugreifen.

Für die Darstellung als Synchronisierungsgraph bedeutet dies zunächst keine Änderung, da hier die Knoten die Systeme und die Projektionen zwischen diesen darstellen. Die Kantengewichte können optional neben der Angabe von τ_{Error} auch um D erweitert werden.

Eine weitere Verzögerung liegt in der Aufnahme der ursprünglichen Daten selbst. So arbeiten beispielsweise speicherprogrammierbare Steuerungen nach einem Zyklus für die Eingabe, Verarbeitung und Ausgabe. In einem definierten zyklischen Abstand werden die Eingangssignale gelesen und dann an Verarbeitung und Ausgabe gegeben. Dies bedeutet, dass ein eingelesenes Signal an der Steuerung im besten Falle in exakt dem Moment aufgetreten ist. Im schlechtesten Fall hingegen tritt das Signal direkt nach Abschluss des

Einlesevorgangs auf, sodass die Erfassung dieses Signals knapp einen kompletten Zyklus später stattfindet. Auch integrierte Verarbeitungssysteme bei Sensorik arbeiten eventuell analog dazu. Dadurch ergibt sich eine potenzielle weitere Verzögerung $D_{plc}^{ursprung}$, die am Quellsystem zu betrachten ist.

Insgesamt lässt sich damit feststellen, dass die Asynchronizität aus Uhren technischer Systeme, Gegebenheiten der Produktionsnetzwerke und weiteren Effekten in der Produktion relevanten Einfluss hat und daher in der Synchronisierung betrachtet werden muss.

2.8 Arten der Synchronisierungen

Kinds of Synchronization

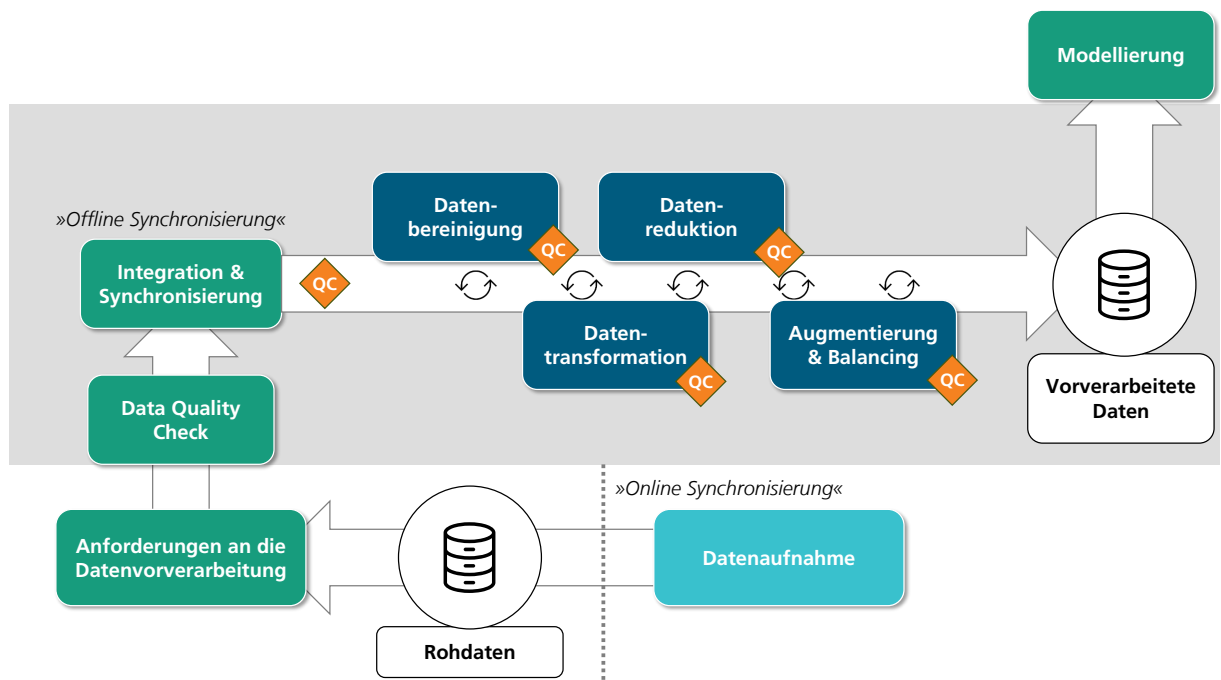


Abbildung 2.22: Einordnung der Online und Offline Synchronisierung im Ablauf von Datenaufnahme bis Modellierung. (Basierend auf [FRYE20] und [FRYE22])

Um eine Synchronizität der Zeitdatenreihen zu erreichen, existieren zwei grundsätzliche Arten von Ansätzen. Derartige Ansätze setzen an verschiedenen Punkten in der Pipeline von Datenaufnahme bis zur Modellierung an. Sie sind in Darstellung 2.22 abgebildet. Diese Arten sind die Online Synchronisierung (unten rechts bei der Datenaufnahme) und die Offline Synchronisierung (links bei der Vorverarbeitung) [PUTT20].

Die Online Synchronisierung beschreibt das Synchronisieren der Systeme zur Laufzeit. Dabei kommunizieren die Systeme miteinander oder mit einer zentralen Instanz. Bei diesen Verfahren kann so die Uhrzeit im System mit der Uhrzeit eines anderen Systems abgeglichen werden und die Abweichung der Zeiten wird bestimmt zur Angleichung der Uhrzeiten in den Systemen. Dadurch erhalten Zeitstempel der Systeme eine höhere Synchronizität zueinander.

Die Offline Synchronisierung hingegen befasst sich mit einer Synchronisierung von zeitbasierten Daten. Dabei kann kein Einfluss mehr auf die aufnehmenden Systeme und den Ablauf der Produktion genommen werden, da diese bereits in der Vergangenheit liegen. Ziel der Offline Synchronisierung ist die Erstellung einer Abbildung von den Zeiten eines Systems auf die Zeiten eines anderen Systems um eine nachträgliche Synchronisierung zu ermöglichen. Dabei können Abschnitte der Zeitdatenreihen verschoben, temporal gestaucht oder gestreckt werden.

Bezogen auf die Formulierungen in Abschnitt 2.7 bedeutet dies, dass die Online Synchronisierung die Zeiten der Systeme C_n (2.56) als Ansatzpunkt wählt. Im Betrieb werden durch Kommunikation der Systeme die Abweichungen durch Taktversatz und Gangfehler regelmäßig bestimmt. Korrekturparameter wie Δ_{corr}^n werden errechnet und eingesetzt. Dadurch reduziert sich die Asynchronizität in den Rohdaten. Bei der Offline Synchronisierung hingegen findet die Berechnung und Korrektur nach der Aufnahme statt, sodass keine Eingriffe und Korrekturen am System vorgenommen werden können. Stattdessen ist die Suche nach den Projektionsfunktionen Φ (2.57) das gewählte Mittel. [SCHM22]

Die Forschungsfragen dieser Arbeit befassen sich mit der Offline Synchronisierung. Da für einige dieser Ansätze und die weiterführenden Arbeiten ein Verständnis der resultierenden Effekte und Rückgriff auf Komponenten der Online Synchronisierung benötigt wird, sind auch grundlegende Methoden der Online Synchronisierung im Bereich der Grundlagen dargestellt.

2.9 Online Synchronisierung

Online Synchronization

Die Online Synchronisierung beschreibt die Synchronisierung der Uhren (bzw. der Berechnung der Uhrzeit in einem System) von Systemen zu deren Laufzeit. Dadurch wird die Genauigkeit der Zeitinformationen der Daten mit Zeitstempeln erhöht, da die Synchronizität beim Erstellen des Zeitstempels genauer ist als ohne Synchronisierung. Die Güte

der Genauigkeit hängt dabei von der Frequenz der Synchronisierung, der eingesetzten Methode und der Kommunikation zum synchronisierenden System ab.

Grundsätzlich kommunizieren ein synchronisierendes System C_{server} und ein zu synchronisierendes System C_{client} miteinander und tauschen Nachrichten zu ihren Zeiten aus. Aus diesen Nachrichten sowie der Zeit zwischen Anfrage und Antwort kann die Differenz der Uhrzeiten in den Systemen auf dem anfragenden System errechnet werden.

Das zu synchronisierende System C_{client} nutzt diese Informationen und stellt die eigene Uhrzeit um. Dies geschieht direkt durch Anpassen der Korrekturparameter Δ_{corr} für das System, was einen Sprung im Uhrzeitverlauf des Systems bedeutet. Ein solcher Sprung kann dabei zu einem Rücklauf der Uhrzeit führen. Diese Methode wird üblicherweise als *step* bezeichnet [MILL10].

Zur Vermeidung solcher Sprünge addiert das zu synchronisierende System C_{client} alternativ einen Korrekturparameter auf die Zählung der Schwingungen des internen Zeitgebers. Dies entspricht einem temporär veränderten Gangabweichung δ bis zur Erreichung der errechneten Synchronizität der Systeme. Dieses Vorgehen wird üblicherweise als *smearing* bezeichnet [THOM20].

Der Standard bei der Online Synchronisierung ist das *Network Time Protocol (NTP)*, von dem es verschiedene Versionen und Varianten gibt. Die erste Version wurde im September 1985 als RFC als NTPv0 veröffentlicht. Darin wurden die notwendigen Komponenten, Nachrichten und Synchronisierungsalgorithmen beschrieben. Ebenfalls umfasst es den grundsätzlichen Ablauf durch regelmäßiges Abfragen durch ein zu synchronisierendes System bei einem synchronisierenden System. Im Laufe der Zeit wurden mehrere Variationen und Erweiterungen entwickelt, die inzwischen in dem aktuellen NTPv4 standardisiert und beschrieben sind [MILL10].

Für die Zustände und Güte der Synchronisierung wird hierbei hierarchisch in **Strata** unterteilt, dargestellt in Abbildung 2.23. An oberster Stelle stehen dabei Systeme im Stratum 0, welche die höchste Güte der Zeitgenauigkeit darstellen. Dies sind beispielsweise Atomuhren oder präzise Zeitgeber in Satelliten, welche daher als Referenzuhren gelten [GUO16]. Ein System, welches direkt eine Synchronisierung gegen eine Referenzuhr durchführt, ist dadurch Stratum 1. Wenn ein System eine Synchronisierung gegen ein System durchführt, welches keine Referenzuhr ist, ist das System damit ein Stratum höher als sein angefragtes System. Dadurch können Systeme die Distanz zu einer Referenzuhr bestimmen und bei Verfügbarkeit mehrerer Zeitserver den Server des niedrigsten Stratum auswählen. Diese Funktionen sind heutzutage standardmäßig in Windows- und Unix-Systemen [MICR21, UBUN18] zu finden.

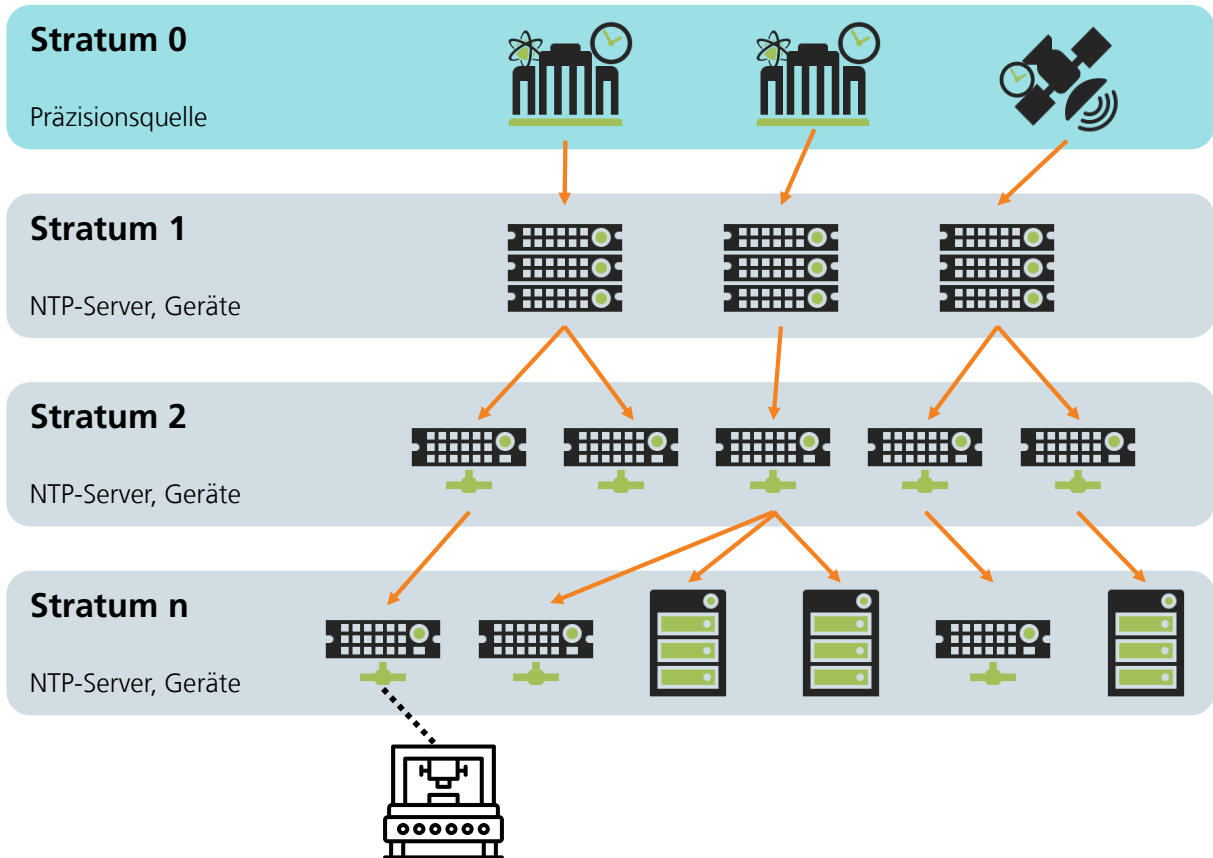


Abbildung 2.23: Darstellung der Strata in NTP-artigen Synchronisierungssystemen. Die oberste Ebene sind die Präzisionsuhren wie Atomuhren und Satelliten (siehe Abschnitt 2.2.3). In den Strata unterhalb von Stratum 0 können dabei NTP-Server und Endgeräte gleichermaßen existieren.

Diese Grundfunktionsweise gilt für alle aktuellen Ansätze, welche die Synchronisierung entlang der echten Zeit erreichen möchten. Weiterentwicklungen wie das *Precision Time Protocol (PTP)* [AD10, IEEE20] und das *Time-sensitive Networking (TSN)* [WANG24] nutzen die gleichen Konzepte und haben ebenfalls die resultierenden *step* und *smearing* Schritte.

Insgesamt lässt sich festhalten, dass die Grundmethodik der Online Synchronisierung bereits weitgehend beforscht wurde [LÉVE16, BALA23]. Aktuelle Forschungen befassen sich hauptsächlich mit Verbesserungen in detaillierten Bereichen und domänenspezifischen Anforderungen, beispielsweise aus den kabellosen Sensornetzwerken [MAHM16, GEET17], der industriellen Automatisierung [BELL19] oder den eingebetteten Systemen im Automotive-Sektor [ASHJ21]. Die Methoden haben damit einen hohen Reifegrad, müssen jedoch bei der Produktion entsprechend aktiviert und konfiguriert sein, um bestmögliche Ergebnisse zu erhalten [GORE20].

Ist nun eine solche Methodik nicht angewendet, oder die Genauigkeit der Synchronisation nicht ausreichend, so kommt auch hier die Offline Synchronisierung zum Einsatz. Effekte wie *step* und *smearing* sind dabei in den Zeitdatenreihen sichtbar. Insbesondere ist dadurch möglicherweise die Monotonie ansteigender Zeitangaben in einem System nicht mehr garantiert, was die Offline Synchronisierung potenziell erschwert.

2.10 Kreuzmodale Bindung als Datenintegration

Cross-Modal Binding as Data Integration

Ein aktueller Trend im Bereich der Produktionswissenschaft ist die bio-inspirierte Fertigung (engl. »Bio-inspired manufacturing«), oder auch »Biologisierung«. Das Grundkonzept ist dabei, Inspiration aus natürlichen biologischen Prozessen und Elementen zu ziehen für das Design von Produkten und Prozessen. Durch die Evolution sind in biologischen Systemen selbstorganisierte, anpassungsfähige und effiziente Strukturen und Prozesse entstanden, die nun für die Produktion als Vorlage dienen [BYRN18]. Dabei zeigt die aktuelle Forschung eine Reihe von Schwerpunkten bei der bio-inspirierten Fertigung. Ein Schwerpunkt beschreibt die Übernahme von biologischen Prozessen zur autonomen Rekonfiguration, um einem System eine Adaption an neue Gegebenheiten zu ermöglichen (engl. »Self-organizing systems«) [JOÃO13]. Die Nachahmung von Mikro- und Nanostrukturen hingegen erlaubt die Verbesserung spezifischer Produkteigenschaften [ZHAN16]. Dies führt auch zu funktionalen Oberflächen, die basierend auf biologischen Phänomenen Eigenschaften wie erhöhte Wasserabweisung ermöglichen [LIU10]. Intelligente Materialien (engl. »Smart fabrics«) integrieren biologische Mechanismen, um erweiterte Funktionen in der Nutzung zu ermöglichen [SING12].

Ein weiterer Schwerpunkt befasst sich mit der intelligenten Datenverarbeitung von Produktionsdaten. Dabei dienen erneut die Prozesse in der Biologie zur Verarbeitung von Daten und Informationen als Vorbild für eine technische Umsetzung. Dies umfasst auch andere Trendthemen wie künstliche Intelligenz, die in einigen Methoden der Funktionsweise eines Gehirns nachempfunden ist. Auch die Nutzung von evolutionären Algorithmen, um iterativ zu Lösungen zu gelangen, basiert auf jenem biologischen Vorbild. [DS19]

Das menschliche Gehirn ist im Rahmen der Datenverarbeitung die zentrale Stelle des Menschen. Dabei ist das Gehirn selbst ebenfalls in mehrere Bereiche unterteilt, die verschiedene Aufgaben übernehmen. Während Hirnstamm und Kleinhirn (»Cerebellum«) auf motorische Themen wie Herzschlag und koordinierte Bewegungen abzielen, übernimmt das limbische System das Gedächtnis und Emotionen. Verschiedene Lappen (engl. »Lobes«) befassen

sich mit der Verarbeitung sensorischer Informationen (im biologischen Sinne) sowie der Zusammenführung eben jener Informationen. Kortizes bezeichnen innerhalb dieser Lappen die dedizierten Bereiche zur Verarbeitung. Konkret sind hier der Parietallappen für taktile (Berührung, Schmerz, Temperatur) Sensorik, der Temporallappen für auditorische (Töne, Sprachverarbeitung) Sensorik und der Okzipitallappen für visuelle Sensorik zuständig mit ihren Kortizes. [HAXB91]

Für die Wahrnehmung eines Menschen treffen zunächst die physikalischen Reize auf die biologische Sensorik (beispielsweise Licht auf die Augen). Die biologische Sensorik wandelt diese in Informationen um, die dann an die Lappen und Kortizes (bei visuellen Informationen der visuelle Kortex im Okzipitallappen) weitergegeben werden. Erst nach der Verarbeitung in den Kortizes werden die verschiedenen Informationen in weitere Bereiche übergeben, in denen multisensorische Informationen zusammengeführt werden (der intraparietale Sulcus). Im Anschluss an diese Zusammenführung gehen diese Informationen weiter an den Frontallappen. Dadurch erscheinen diese Informationen erstmals im Bewusstsein des Menschen. [BRIN18]

Der Prozess der Zusammenführung der sensorischen Informationen ist dabei nicht trivial, sondern umfasst zusätzliche komplexe Verarbeitungen. Anstatt dem Bewusstsein ein unvollständiges oder fehlerhaftes Bild zu übergeben, versucht der intraparietale Sulcus die Informationen anzupassen, sodass eine konsistente und (scheinbar) vollständige Wahrnehmung entsteht. Tatsächlich greift der intraparietale Sulcus dazu nicht nur auf die sensorischen Informationen zurück, sondern ebenfalls auf Wissen und Informationen aus anderen Bereichen des Gehirns. Dies wird als kreuzmodale Bindung (engl. »Cross-modal binding«) bezeichnet. [BUSH03]

Ein besonderer Fall dieser kreuzmodalen Bindung ist die Kausalinferenz (engl. »Causal inference«), welche exemplarisch in der Abbildung 2.24 dargestellt ist. Der intraparietale Sulcus identifiziert dabei kausale Zusammenhänge in den vorliegenden Informationen. Das Wissen über Zusammenhänge nutzt dieser dann, um die eigentlichen Informationen neu zu synchronisieren um dem Bewusstsein eine passende Aggregation weiter zu geben. Beispielsweise ist bei einem Auftreffen eines Fingers auf einem Tisch das visuelle Signal und die daraus erstellte Information (lila) schneller im intraparietalen Sulcus der wahrnehmenden Person als das auditorische Signal (blau). Der intraparietale Sulcus erkennt, dass diese Informationen allerdings wahrscheinlich den gleichen Ursprung (Kausalität) haben und verändert die Informationen, sodass dem Frontallappen eine Gleichzeitigkeit der Informationen gegeben wird [ALSI18], dargestellt als grüner Verlauf. Ist die Wahrscheinlichkeit der Gleichzeitigkeit hingegeben als gering bewertet, so werden die Informationen wie einge-

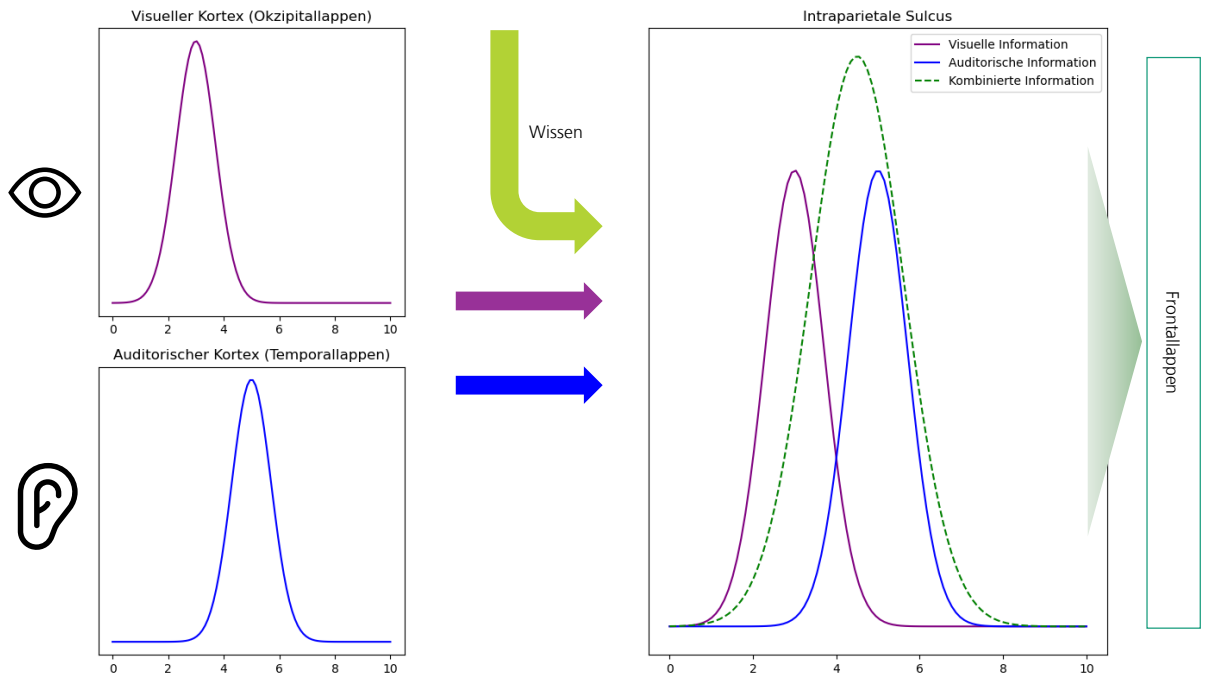


Abbildung 2.24: Ablauf der Verarbeitung von auditiven und visuellen Informationen im intraparietalen Sulcus. Die Reize kommen links über die Sensorik in die Kortexes und mit dem Wissen aus dem Gehirn zusammen in den intraparietalen Sulcus. Die zusammengeführte (grüne) Information wird dem Frontallappen übergeben. (Basierend auf [CAO19])

gangen versetzt zueinander übergeben. Diese Zuordnung basiert auf Wahrscheinlichkeiten und kann daher falsch sein [CAO19].

Dieser Effekt ist im Bereich der Sprachwahrnehmung auch nach seinem Entdecker als *McGurk-Effekt* bekannt und kann positive Konsequenzen mit sich bringen. So fand McGurk heraus, dass sich die Wahrnehmung von Sprache verbessert, wenn zugleich die sich bewegenden Lippen der sprechenden Person (wenn auch leicht versetzt) sichtbar sind [MCGU76]. Gleichzeitig kann diese Veränderung der Wahrnehmung auch zu negativen Konsequenzen führen. So führt diese Veränderung durch Kausalinferenz zu Täuschungen und Fehlwahrnehmungen, die im Extremfall bis zu Illusionen und Fehlwahrnehmungen des eigenen Körpers führen [KILT15].

Die vorgestellten Abläufe der Wahrnehmung des Menschen und der Verarbeitung können teilweise vergleichbar zur Datenaufnahme und -integration in der Produktion betrachtet werden. Sowohl in der Produktion als auch im Körper existiert Sensorik zur Aufnahme und Verarbeitung von physikalischen Reizen. Bei multiplen Datenquellen werden diese Daten

zusammengeführt und vorverarbeitet, bevor sie an ein weiterführendes System (Analytics bzw. Frontallappen) gegeben werden. Mit dieser Analogie entspricht der intraparietale Sulcus dem Schritt der Datenintegration und -synchronisierung bei der Produktion. Dabei greift der intraparietale Sulcus in diesem Schritt bereits auf das Wissen des übergreifenden Systems (Gehirn) zurück um die Qualität der Synchronisierung zu erhöhen [SCHM23b]. Eben jenes Konzept aus der Neurobiologie wird in dieser Arbeit als Inspiration aufgegriffen.

2.11 Anwendung in der Batteriezellfertigung

Application in Battery Cell Production

In den vorangegangenen Abschnitten wurden die Grundlagen der Synchronisierung von Zeitdatenreihen einzeln dargestellt. Der Anwendungsbereich dieser Arbeit fokussiert die Batteriezellfertigung, wie sie in Abschnitt 2.5 dargestellt ist. Die Batteriezellfertigung ist eine komplexe Prozesskette und die Anwendung von Methoden der Digitalisierung verspricht daher Schaffung von Einblicken in die Ursache-Wirkzusammenhänge und Optimierungspotenziale.

In Anbetracht der verketteten Produktion, dargestellt in Abbildung 2.14, kommt eine Vielzahl von Maschinen zum Einsatz. Wird die Pouchzelle betrachtet, ergeben sich alleine durch die Anlagentechnik 17 technische Geräte von Relevanz, die Daten generieren. Diese Menge erhöht sich durch die Integration von Inline-Messtechnik weiter. Da die Ursache-Wirkzusammenhänge in der Batteriezellfertigung nicht abschließend bekannt sind, ist auch die Integration und der Wechsel weiterer Messtechnik wahrscheinlich. Darüber hinaus findet diese Produktion teilweise in besonderen Umgebungen, den Trockenräumen, statt, welche ebenfalls eine Regelung und damit relevanten Datenquellen umfassen. In dem Produktionsnetzwerk (siehe Abschnitt 2.4) befinden sich oberhalb der Anlagen- und Gebäudetechnik in Form von Systemen wie einem Manufacturing Execution System (MES), IIoT-Plattform oder Wareneingangssystem weitere technische Datenquellen. Die besonderen Anforderungen der Batteriezellfertigung sorgen dabei für spezifische Anpassungen des gesamten Produktionsnetzwerks gegenüber anderen Produktionen [MOHR25].

Diese Vielzahl von Datenquellen ist möglicherweise direkt an die IIoT-Plattform oder ein SCADA-System angebunden, können aber auch vorher verbunden und aggregiert werden. So kann Messtechnik direkt in die SPS eingebunden werden, um die Daten über die zentrale Steuerung zu übertragen. Ebenfalls können mehrere Maschinen (beispielsweise mehrere Schritte der Assemblierung) erst an eine gemeinsame weitere SPS angeschlossen werden, welche erst danach an die weiterführenden Plattformen kommuniziert. Dadurch verändert

sich die Kommunikationsstrecke eines Datums von der Quelle ausgehend, und mehrere Systeme mit technischen Uhren (siehe Abschnitt 2.2.3) sind Teil des Datenaufnahmeprozesses. Aus den verschiedenen Uhren, den Kommunikationslatenzen (siehe Abschnitt 2.3) und gestaffelten Aufbauten ergeben sich Ungenauigkeiten und Asynchronizitäten in den Daten der Batteriezellproduktion.

Für die Anwendung von datenbasierten Produktionsverbesserungen ist durch die Menge und unklare Qualität der Daten die Vorverarbeitung von Produktionsdaten (siehe Abschnitt 2.6) notwendig. Die eingebrachte Asynchronizität hat dabei einen direkten und indirekten Einfluss auf die Datenqualität und die möglichen Bereinigungsschritte.

Die Aufgabe, einen Teilbereich der Elektrodenfolie von Beginn der Produktion bis zur fertigen Zelle nachzuverfolgen (»Traceability«) zieht sich entlang der gesamten Produktion. In der Konsequenz führen bereits kleine Abweichungen und Ungenauigkeiten in den einzelnen Schritten bei der Zuordnung von Daten zu hohen Gesamtungenauigkeiten.

Ein Produktionsschritt als Bahnprozess in der Elektrodenfertigung mit einer Messtechnik lässt sich dadurch folgendermaßen beschreiben. Die Zeitdatenreihen der Anlage sind beschrieben durch $_{An}Z$ und die Zeitdatenreihen aus der Messtechnik analog durch $_{MT}Z$. Die Frequenzen der beiden Zeitdatenreihen weichen voneinander ab, sodass nicht für jeden Zeitwert eine direkte Entsprechung in der jeweils anderen Zeitdatenreihe existiert. Daher können fehlende Zeitreihenwerte als Platzhalter in $_{MT}Z^\epsilon$ und $_{An}Z^\epsilon$ abgebildet werden. Zusammen bilden all diese Zeitdatenreihen in der Betrachtung die Gesamtheit der Zeitdatenreihen $\mathfrak{Z}$.

Entlang des Aufbaus des Produktionsnetzwerks werden die beiden Zeitdatenreihen nun nicht in einem gemeinsamen System aufgenommen, sondern separat in C_{An} ($C_{An} \succ_{An} Z$) und C_{MT} ($C_{MT} \succ_{MT} Z$) und erst später bei der Vorverarbeitung zusammengeführt und sollen synchronisiert werden. Daher müssen Taktversatz und Gangfehler zwischen diesen Systemen betrachtet werden. Der Taktversatz ist wie üblich (2.55) als Δ_{skew}^{An} und Δ_{skew}^{MT} definiert. Der Gangfehler der Anlage $\delta^{An}(i)$ und der Messtechnik $\delta^{MT}(i)$ sind ebenfalls gemäß der obigen Definitionen (2.54) bestimmt. Für die orts aufgelösten Daten (Ortsstempel) des Beschichters aus Abschnitt 2.5.3 verändert sich 2.37 zu

$$\begin{aligned}
 SS(TS_{MT}) &= s_0 + s_{MT} + \int_{TS_{start} \pm X_{start}}^{TS_{MT} \pm X_{MT}} \vec{v}(t) dt \\
 &= s_0 + s_{MT} + \int_{C_{An}(TS_{start})}^{C_{MT}(TS_{MT})} \vec{v}(t) dt \\
 &= s_0 + s_{MT} + \int_{C_{An}(TS_{start})}^{\Phi_{MT,An}(C_{MT}(TS_{MT}))} \vec{v}(t) dt \\
 &= s_0 + s_{MT} + \int_{z_{C_{An}}^{An}(TS_{start})}^{z_{\Phi_{MT,An}(C_{MT}(TS_{MT}))}^{An}} \vec{v}(t) dt \\
 &= s_0 + s_{MT} + \int_{z_{C_{An}}^{An}(TS_{start})}^{z_{\Phi_{MT,An}(C_{MT}(TS_{MT}))}^{An}} v_t^{An} dt, \tag{2.67}
 \end{aligned}$$

wobei X die Ungenauigkeiten bezeichnet, die sich bis dahin aufsummiert haben aus Gangfehlern, Taktversatz, Latenzen und vorangegangenen Ungenauigkeiten. Zur Synchronisierung der verschiedenen Zeiten ist daher die Projektionsfunktion $\Phi_{MT,An}$ unerlässlich, da sie die Zeitstempel der Messtechnik auf die Anlage überträgt, sodass die Ortsberechnung stattfinden kann. Da Zeitdatenreihen keine echten kontinuierlichen Funktionen, sondern diskrete Reihen sind, kommt eine weitere Ungenauigkeit durch die numerische Integration hinzu. Alleine durch diese Ungenauigkeiten ergibt sich $SS(TS_{MT})$ zu einem Bereich statt eines Ortes auf dem Band. Diese Ungenauigkeit wird dann auch bei den nächsten Prozessschritten (z.B. Kalandrieren) aufgenommen und potenziell noch erweitert. Dadurch ergibt sich eine potenziell quadratische Fehlerfortpflanzung durch die gesamte Batteriezellproduktion, die durch eine Synchronisierung signifikant reduziert werden kann. Damit ist die Aufgabe der Synchronisierung der Zeitdatenreihen (siehe Abschnitt 2.7) in der Batteriezellfertigung von entscheidender Bedeutung.

2.12 Zwischenfazit

Interim Conclusion

In diesem Kapitel wurden die Industrierelevanz und die Grundlagen der Synchronisierung von Zeitdatenreihen betrachtet und der thematische Kontext der Batteriezellfertigung eingeordnet. Die Synchronisierung von Zeitdatenreihen ist eine zentrale Aufgabe zur Ermöglichung der datenbasierten Produktionsverbesserung, da sie bereits frühzeitig in der

Datenvorverarbeitung stattfindet und Asynchronizität auch andere Probleme in den Daten beeinflusst.

In der Domäne der Batteriezellfertigung existiert eine komplexe Produktionskette mit einer Vielzahl von Systemen und Geräten und einem Ziel der Skalierung verbunden. Diese Kombination resultiert in einer steigenden Menge von Zeitdatenreihen, die synchronisiert werden müssen. Gleichzeitig wächst der Markt und Bedarf an Batteriezellproduktionen erheblich, sodass die Größe des Problems weiter ansteigt und ein Mangel an Expert*innen in der Domäne besteht.

Für die Lösung dieser Probleme und Bedarfe könnte eine verbesserte Methodik zur Synchronisierung von Produktionsdaten in der Batteriezellfertigung einen signifikanten Beitrag leisten. Dabei spielt die erwartete Skalierung der Produktionen eine besondere Rolle, die in der Lösung adressiert werden muss, und daher in der Unterforschungsfrage **UFF 4** besteht. Ebenso ist die begrenzte Verfügbarkeit von Expert*innen in der Domäne kritisch, sodass dieses Wissen effizient eingesetzt werden muss, was in den Unterforschungsfragen **UFF 2** und **UFF 3** aufgegriffen wird.

Insgesamt bedeutet dies, dass eine ideale Lösung mindestens automatisiert und skalierbar sein muss. Für die technischen Anforderungen und Metriken existiert kein entsprechendes Modell, sodass die Metriken zunächst entwickelt werden müssen, was in der Unterforschungsfrage **UFF 1** adressiert ist.

In dieser Arbeit sind damit mehrere Aktionsfelder zu adressieren. Diese lassen sich folgendermaßen zusammenfassen:

1. Die Modellierung und Bewertung von Asynchronizität und Synchronisierung von Zeitdatenreihen - Dieses Feld ist damit auch die Anforderungsbasis, gegen die die entwickelte Methodik bewertet wird.
2. Die Entwicklung einer Methode für die Durchführung der Synchronisierung von Zeitdatenreihen
3. Die Anwendung dieser Entwicklungen an Zeitdatenreihen aus der Batteriezellproduktion und die Bewertung der Ergebnisse mit den entwickelten Metriken

3 Existierende Ansätze und Ableitung der Anforderungen

Existing Approaches and Derivation of Requirements

In diesem Kapitel ist der aktuelle Stand von Forschung und Technik zur Offline Synchronisierung von Zeitdatenreihen von Daten technischer Systeme, wie einem Produktionsnetzwerk vorkommen, dargestellt. Die Ansätze verfolgen dabei verschiedene Konzepte und nutzen entsprechende Annahmen. Daher werden diese Ansätze zusammen mit ihren Annahmen und Bedingungen dargestellt.

Wie in Abschnitt 2.12 beschrieben, existieren keine ausreichenden Metriken für die Bewertung solcher Ansätze. Aus diesem Grund befasst sich dieses Kapitel ebenfalls mit der Entwicklung geeigneter Metriken. Abschließend werden die Anforderungen und Metriken genutzt, um die existierenden Ansätze zu bewerten. Aus dieser Bewertung lassen sich eine Einordnung und der Handlungsbedarf ableiten.

3.1 Offline Synchronisierung

Offline Synchronization

Wie in Abschnitt 2.8 dargestellt, befasst sich die Offline Synchronisierung mit der Erstellung von Projektionsfunktionen Φ , die die Zeitdaten zwischen den Systemen übertragbar machen, um so die Zeitdatenreihen zu synchronisieren. Während die Online Synchronisierung, wie in Abschnitt 2.9 dargestellt, einen hohen Reifegrad und verbreitete Anwendungen bietet, ist dies bei der Offline Synchronisierung weniger ausgeprägt [PUTT20]. Da bei Fehlern, dem Fehlen oder unzureichender Qualität der Online Synchronisierung nur die Offline Synchronisierung helfen kann, ist der Bedarf signifikant [SCHM22].

Für die Offline Synchronisierung existieren verschiedene aktuelle Ansätze, die hier vorgestellt werden. Diese Methoden haben dabei ihren Ursprung in verschiedenen Domänen, was entsprechend eingeordnet wird.

3.1.1 Dynamic Time Warping

Dynamic Time Warping

Zeitdatenreihen, die ähnliche Abläufe beschreiben haben Ähnlichkeiten in Bezug auf die Datenverläufe und Strukturen. Diese Ähnlichkeiten sind der Startpunkt für das sogenannte **Dynamic Time Warping (DTW)**. So basiert DTW darauf, zwei Zeitdatenreihen auf ihre Ähnlichkeit zu untersuchen und den jeweiligen Punkten einen Punkt der jeweils anderen

Zeitdatenreihe zuzuweisen. Dies führt zu einer Art Dehnung und Stauchung von Bereichen in der Zeitdatenreihe, welche gegen eine andere synchronisiert wird.

Die Ursprünge von Dynamic Time Warping liegen in der Sprachverarbeitung der 1970er Jahre bei SAKOE. Aufgrund verschieden schnell gesprochener Wörter wurde eine Methode zur Angleichung notwendig, um die Variationen der Sprechgeschwindigkeit aufzulösen [SAKO78]. Im Laufe der Jahre wurde diese Methodik weiterentwickelt und in anderen Domänen eingesetzt [LI21]. So wird DTW in der Finanzbranche verwendet zur Analyse von Kursverläufen [TSIN14]. In der Medizin ist die Verwendung von DTW verbreitet für Ähnlichkeitsanalysen von Krankheitsverläufen [GIAN18] und wird auch für Datenreihen von medizinischen Geräten [HUAN85] verwendet. In der Produktionsdomäne existieren ebenfalls Anwendungen für DTW. Für Werkzeuge und Verschleißteile kann mit DTW eine Restlebensdauer vorhergesagt werden [KIM20], durch Anwendung auf relevante Eigenschaften über die Verwendungszeit. Das integrierte Distanzmaß findet auch bei der Bewertung von Modellen wie Energievorhersagen in der Produktion Einsatz [STRÖ23]. Für die direkte Synchronisierung von Zeitdatenreihen findet DTW auch in der Produktion Anwendung [NIKO21].

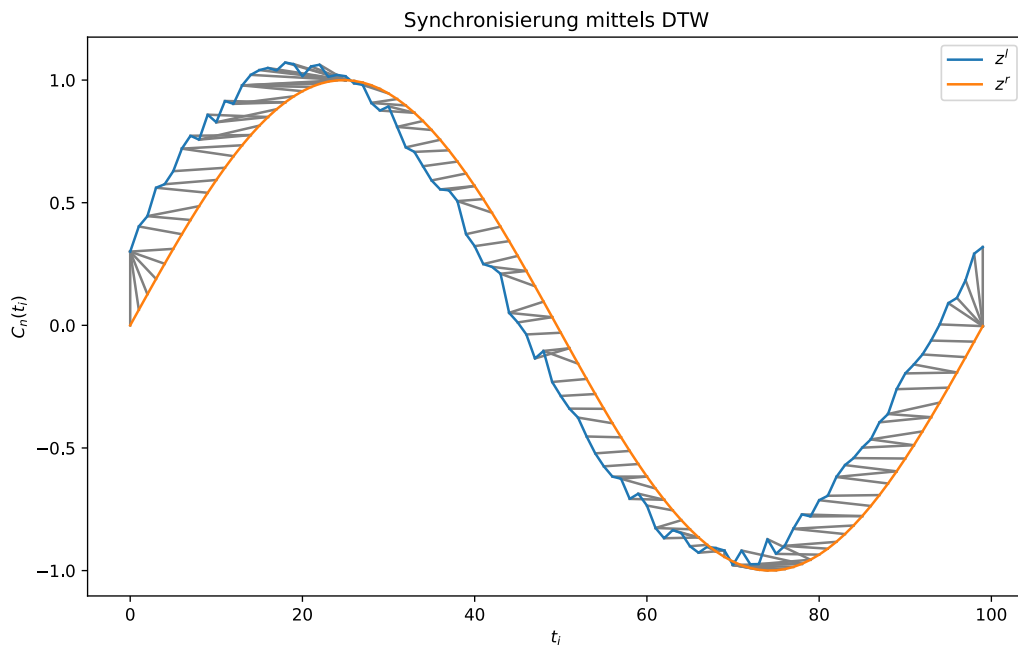


Abbildung 3.1: Eine Synchronisierung von Zeitdatenreihen, durchgeführt mittels Dynamic Time Warping. Die grauen Linien beschreiben dabei die Abbildungsfunktion $\Phi_{l,r}$

Das Ergebnis einer solchen Synchronisierung ist in Abbildung 3.1 dargestellt. Für die Zeitdatenreihe z^l soll eine Synchronisierung auf z^r durchgeführt werden. Mittels DTW wird eine punktweise Projektionsfunktion $\Phi_{l,r}$ (graue Striche) errechnet.

DTW Algorithmus

Der DTW-Algorithmus nutzt die gegebenen Zeitdatenreihen und errechnet zunächst paarweise die Distanzen aller (z_i^l, z_j^r) Paare. Diese Menge an Distanzen wird dann als Distanzmatrix D der Dimension $|_lZ| \cdot |_rZ|$ dargestellt. Jeder Eintrag $D(i, j)$ der Matrix entspricht damit dem Abstand zwischen z_i^l und z_j^r . Für diesen Abstand können dabei verschiedene Metriken Verwendung finden. Üblicherweise wird hier der euklidische Abstand $D(i, j) = \sqrt{(t_i^l - t_j^r)^2}$ verwendet.

Basierend auf der Distanzmatrix wird dann ein Kostengitter K erstellt. Die Startecke bekommt dabei per Definition die Kosten zugewiesen, die der Distanz der jeweils ersten Zeiten entsprechen. Die weiteren Einträge werden als minimale Kostenpfade errechnet:

$$K(\arg \min_i {}_lZ, \arg \min_j {}_rZ) = D(\arg \min_i {}_lZ, \arg \min_j {}_rZ) \quad (3.1)$$

$$K(i, j) = D(i, j) + \min\{K(i-1, j), K(i, j-1), K(i-1, j-1)\}$$

Dieses Kostengitter bildet die Basis, um Pfade zu bewerten, die vom Start- zum Endpunkt gehen. Übertragen auf die Zeitdatenreihen bedeutet dies, dass für jede betrachtete Position ($K(i, j)$) drei Möglichkeiten von Übergängen existieren. Der vertikale Übergang ($K(i-1, j)$) bedeutet hier eine Verzögerung der Zeitdatenreihe ${}_lZ$. Analog dazu beschreibt der horizontale Übergang ($K(i, j-1)$) eine Verzögerung in der Zeitdatenreihe ${}_rZ$. Der letzte Fall beschreibt den synchronen Schritt ($K(i-1, j-1)$) in beiden Zeitdatenreihen.

Nun soll für die Synchronisierung der optimale Pfad gefunden werden zwischen Start- und Endpunkt (»Warping-Pfad«). Dazu wird beim Endpunkt $K(\arg \max_i {}_lZ, \arg \max_j {}_rZ)$ gestartet und der kürzeste Pfad zum Startpunkt (3.1) gesucht. Rekursiv wird der Pfad Übergang für Übergang aufgebaut durch die Auswahl des Nachbarn mit den geringsten kumulativen Kosten $\min\{K(i-1, j), K(i, j-1), K(i-1, j-1)\}$. Der resultierende Pfad W , als Menge der gewählten Gitterpositionen, ist eine Minimierung der Gesamtkosten im Kostengitter und beschreibt die Synchronisierung der beiden Zeitdatenreihen [SENI08]. Für die Betrachtung als Projektionsfunktion ergibt sich damit:

$$\Phi_{l,r}(z_i^l) = \{z_j^r\} \quad | (i,j) \in W \quad (3.2)$$

$$\Phi_{r,l}(z_j^r) = \{z_i^l\} \quad | (j,i) \in W \quad (3.3)$$

Die Umkehrabbildung ist also durch Transposition direkt ableitbar. Durch die Übergänge in horizontalen und vertikalen Richtungen in dem Kostengitter ist die Zuordnung in der Projektion nicht mehr eindeutig und ein Wert kann auf mehrere Werte abgebildet werden.

Als Nebeneffekt dieser direkten Zuordnung müssen keine zusätzlichen Werte (aus ${}_lZ^\epsilon, {}_rZ^\epsilon$) betrachtet werden. Die Berechnung erfolgt ausschließlich mit existierenden Datenpunkten. Ein weiterer Nebeneffekt davon ist die Veränderung der Fehlerbetrachtung. Sie ist nicht mehr direkt von dem Fehler der Zykluszeit $\tau_{Error}^{l,r}$ abhängig, sondern von der Abweichung von W zu einer theoretischen Diagonale von Start- zu Endpunkt.

Solch eine Berechnung der Kostenmatrix und dem resultierenden Pfad ist in Abbildung 3.2 dargestellt. Die Synchronisierung aus 3.1 ist hier durch den Pfad in der Mitte dargestellt. Der Pfad reicht von unten links nach oben rechts und umfasst alle drei verschiedenen Übergänge.

Variationen von Dynamic Time Warping

Dynamic Time Warping ist ein aktives Forschungsfeld, mit vielen Variationen und Weiterentwicklungen in den letzten Jahrzehnten. Die Schwerpunkte der Forschung lassen sich auf vier Bereiche aufteilen. Ein Schwerpunkt ist die Anwendung von Dynamic Time Warping in verschiedenen Domänen, wie bereits oben beschrieben. Dabei wird auch Dynamic Time Warping als Metrik genutzt für die Entwicklung anderer Algorithmen und die Erstellung von Modellen im maschinellen Lernen. [KATE16]

Während das normale Dynamic Time Warping auf gegebenen (statischen) Zeitdatenreihen Anwendung findet, befassen sich neue Ansätze mit der Schaffung einer Online Fähigkeit und der Anpassungen an Veränderungen. Diese Dynamic Dynamic Time Warping Algorithmen erlauben während der Berechnung Aktualisierungen, bei denen neue Elemente in die Zeitdatenreihen eingefügt, verändert oder gelöscht werden können [BRIN24]. Mit Blick auf Streaming von Daten bedeutet dies potenziell endlose Zeitdatenreihen ($|Z| = \infty$). Daher nutzen Anpassungen wie **Online Dynamic Time Warping** dynamische Fenster, in denen die Kostengitter berechnet werden. Durch diese Segmentierungen verändert sich allerdings potenziell auch die mathematische Güte des Pfads im Gesamt-Kostengitter [OREG17].

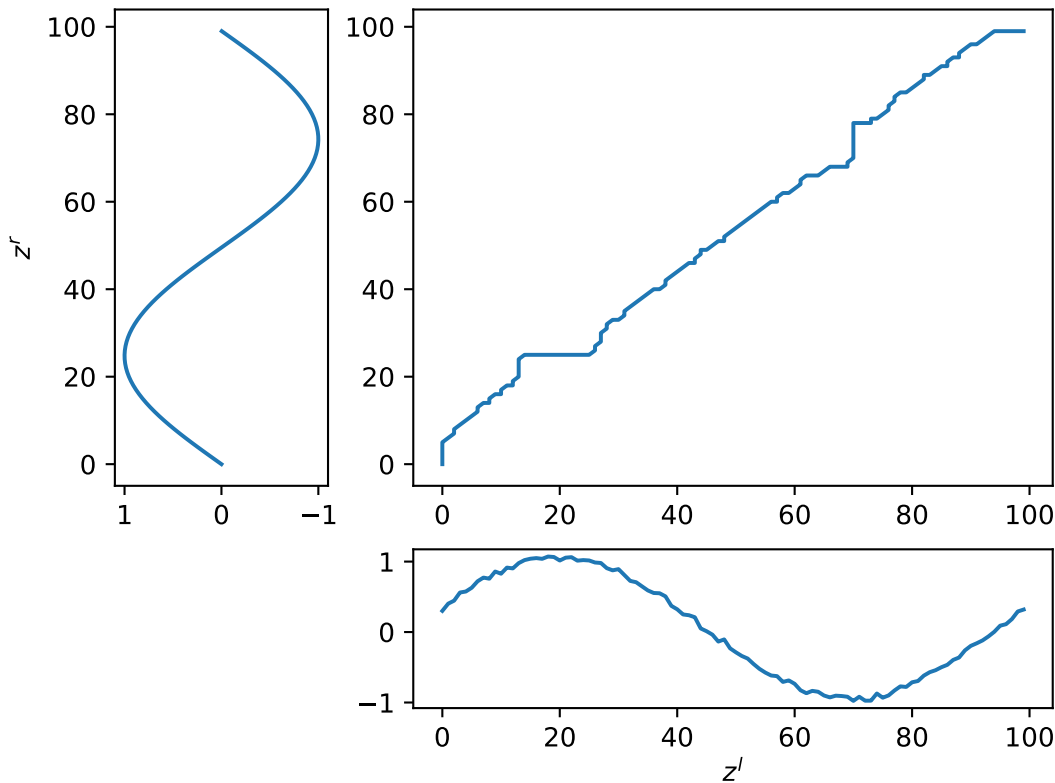


Abbildung 3.2: Die durchgeführte Synchronisierung mittels Dynamic Time Warping, mit der internen Distanzmatrix in der Mitte. Der Pfad durch die Distanzmatrix (blau) entspricht der Warping-Funktion, also der Projektion von z^l auf z^r .

Die Erstellung des Kostengitters nimmt einen signifikanten Teil der Berechnungszeit ein. Beim normalen Dynamic Time Warping wird das gesamte Gitter berechnet, auch wenn am Ende nur ein Teil des Kostengitters genutzt wird. Diesen Umstand adressieren mehrere Anpassungen zur Optimierung. So beschreibt ITAKURA eine globale Einschränkung für die Suche des Pfades durch das Kostengitter. Maximale Winkel β der Betrachtung vom Start- und Endpunkt aus beschreiben diese Einschränkung zur Hauptdiagonalen [ITAK75]. Für das gesamte Gitter beschreibt diese Einschränkung damit ein Parallelogramm, welches heute auch als **Itakura-Parallelogramm** bezeichnet wird. Einen ähnlichen Weg gingen SAKOE & CHIBA, indem sie eine globale maximale Entfernung ($|i - j| \leq b$) zur Hauptdiagonalen definierten [SAKO78]. Heute ist diese globale konstante Einschränkung als **Sakoe-Chiba-Band** bekannt. Diese Beschränkungen sind in Abbildung 3.3 dargestellt. Das Sakoe-

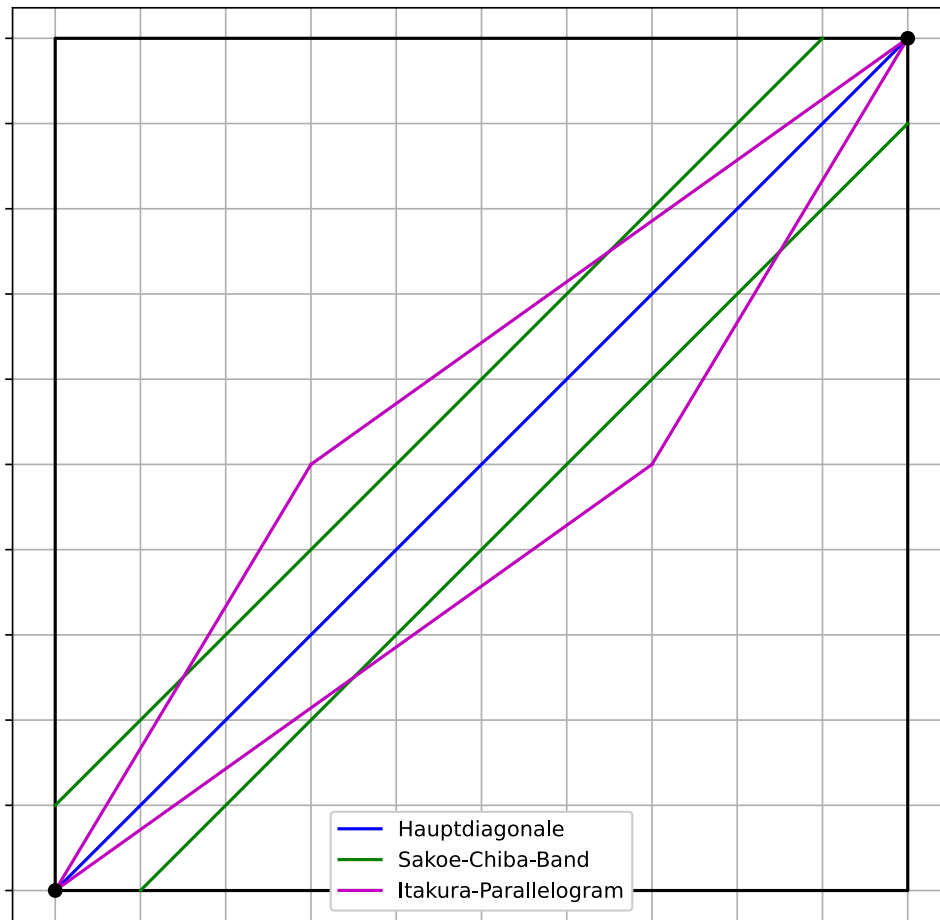


Abbildung 3.3: Dynamic Time Warping Kostengitter mit dem Sakoe-Chiba-Band (grün) und dem Itakura-Parallelogramm (lila). Die Hauptdiagonale zwischen Start- und Endpunkt ist mittig in blau dargestellt.

Chiba-Band (grün) behält einen konstanten Abstand zur Hauptdiagonalen, während das Itakura-Parallelogramm (lila) nahe der Start- und Endpunkte eine stärkere Beschränkung darstellt und in der Mitte des Gitters weniger restriktiv ist. In beiden Fällen wird ein Großteil des Gitters grundsätzlich ausgeschlossen, sodass die Erstellung des Kostengitters deutlich

reduziert wird. Weitere Variationen befassen sich bei der Erstellung des Kostengitters mit Gewichtungen, um bestimmte Bereiche in den Zeitdatenreihen zu betonen [JEON11].

Zusätzliche Varianten nutzen nicht nur die punktuelle Betrachtung der Zeitdatenreihen, sondern nutzen ganze Segmente und die Muster darin [KEOG01]. Dafür wird jedem Element z_i in den Zeitdatenreihen eine Subsequenz $\overset{\circ}{U}_i$ zugeordnet, welche die Subsequenz um dieses Element herum beschreibt. Weiterhin werden Formmerkmale (engl. »Shape Descriptors«) erstellt. Formmerkmale können dabei verschieden definiert werden, beispielsweise über Gradienten. Hierbei kann auf Ansätze wie *Histogram of Oriented Gradients for 1D Data (HOG1D)* [ZHAO16] zurückgegriffen werden. Diese Merkmale M_i werden für jede der Subsequenzen erstellt. Die Menge der Formmerkmale $\mathbb{M} = \{M_i\}$ bilden eine Art Merkmaldatenreihe, welche eine höhere Abstraktion der ursprünglichen Zeitdatenreihe ist. Nun wird der Dynamic Time Warping Algorithmus auf diese erstellte Merkmaldatenreihe zur Synchronisierung angewendet und danach von den Subsequenzen wieder auf die ursprünglichen Elemente zurück übertragen. Auf diese Weise bezieht **ShapeDTW** die Umgebung und Strukturen in den Zeitdatenreihen ein und wird robuster gegenüber lokalen Effekten. Die Erstellung der Subsequenzen und der Formmerkmale erhöht allerdings die notwendigen Berechnungen maßgeblich. [ZHAO18]

Der letzte Schwerpunkt der aktuellen Forschung zu Dynamic Time Warping befasst sich aufgrund der hohen Berechnungsaufwände mit der algorithmischen Optimierung der Laufzeit. Anstatt die Kostenmatrix komplett zu erstellen, wird bei **PrunedDTW** [SILV16] nur ein Teil des Kostengitters zu Beginn berechnet. Während der Pfadsuche werden notwendige Einträge dann befüllt. Dabei wird der bis dahin beste Pfad gespeichert und sobald bei einer Berechnung diese kumulierten Kosten überschritten werden, der dahinterliegende Bereich des Kostengitters beschnitten, also nicht weiter betrachtet. Ein anderer Ansatz ist die Vereinfachung der Zeitdatenreihen (**FastDTW**). So wird eine kleinere Zeitdatenreihe erstellt durch Mittelung mehrerer Punkte. Auf der verkleinerten Zeitdatenreihe wird dann DTW ausgeführt. Entlang des Pfades wird nun wieder auf eine feinere Granularität gewechselt und mit DTW verfeinert. Dieses rekursive Vorgehen erlaubt Abstraktionen beliebiger Menge und beschleunigt die Berechnung bei geringen Einbußen des Ergebnisses [SALV04]. Alternativ kann auch eine Linearisierung bei den Zeitdatenreihen direkt genutzt werden, um die Daten selbst vereinfacht darzustellen und Dynamic Time Warping zu beschleunigen, da die Berechnungen von Distanzmatrix und Kostengitter deutlich geringere Dimensionen betrachten [KEOG99].

Abschließende Eigenschaften

Insgesamt ist Dynamic Time Warping ein aktives Forschungsfeld mit einer großen Menge von Anwendungen in verschiedenen Domänen. DTW erlaubt dabei die Synchronisierung von zwei Zeitdatenreihen, inklusive einer Umkehrfunktion. Durch die direkte Nutzung der Elemente in den Zeitdatenreihen besteht keine Notwendigkeit zur Betrachtung von Zwischenbereichen. Die Berechnung der Distanzmatrix und des Kostengitters ist dabei für die Laufzeit entscheidend. Während bei den normalen Algorithmen eine Laufzeit- und Speicherkomplexität in $\mathcal{O}(|_lZ| \cdot |_rZ|)$, also quadratischem Wachstum, existiert, können Optimierungen dies teilweise reduzieren. So ergibt sich durch die rekursive Methode in FastDTW eine lineare Platzkomplexität und eine Zeitkomplexität in $\mathcal{O}(|_lZ| \cdot \log(|_lZ|) + |_rZ| \cdot \log(|_rZ|))$.

In dieser Arbeit werden verschiedene Varianten von Dynamic Time Warping verwendet. Da viele Eigenschaften für die Varianten übergreifend gelten, werden diese Varianten teilweise zusammenfassend betrachtet.

3.1.2 Systemidentifikation

System Identification

Die Methode der Systemidentifikation stammt aus den Bereichen der Regelungstechnik und der Datenanalyse. In der Systemidentifikation ist das Ziel die Modellierung eines dynamischen Systems auf Basis der Beobachtungen eben jenes Systems. Dabei sind der innere Aufbau, die inneren Abläufe und Parameter im Vorfeld nicht bekannt, sodass das mathematische Modell nicht auf diese zurückgreifen kann. Üblicherweise wird daher das Modell aus der Systemidentifikation verwendet, um Vorhersagen zum Verhalten eines Systems zu erstellen [LUN10].

Die Grundstruktur für Systemidentifikation besteht aus Eingangs- und Ausgangssignalen und der Lösung eines Optimierungsproblems. Bezogen auf Zeitdatenreihen ergibt sich damit eine Darstellung $SI = \{\vec{u}(t_i), \vec{y}(t_i)\}$, sodass die Menge SI die zusammengefassten Eingangssignale $\vec{u}$ sowie die Modellvorhersagen $\vec{y}$ zum gleichen Zeitpunkt t_i umfassen. Im zweiten Schritt werden Modelle gewählt und die Parameter des Modells optimiert.

Modelle

Die verbreitetsten Modelle lassen sich nach Art unterteilen. Die erste Kategorie bezeichnet die linearen Modelle. Das verbreitete **ARX** (AutoRegressive with eXogenous inputs) Modell [BOX15] ist dabei beschrieben durch

$$\vec{y}(t) + a_1\vec{y}(t-1) + \dots + a_n\vec{y}(t-n) = b_1\vec{u}(t-1) + \dots + b_m\vec{u}(t-m) + e(t), \quad (3.4)$$

wobei a und b die Modellparameter und $e(t)$ der Fehlerterm sind. Eine Erweiterung von ARX stellt **ARMAX** (AutoRegressive Moving Average with eXogenous inputs) dar, bei dem zusätzlich zur rechten Seite in 3.4 noch eine bewegliche Durchschnittskomponente $c_1e(t-1) + \dots + c_ke(t-k)$ hinzukommt, um die vergangenen Fehler statt nur den aktuellen Fehlerterm zu berücksichtigen [HANN12]. Neben den obigen Modellen sind auch nicht-lineare Modelle verfügbar. So ist das Modell ARX als **NARX** (Nonlinear AutoRegressive with eXogenous inputs) auf nicht-lineare Fälle erweitert. Dazu werden die einzelnen Komponenten aus 3.4 bis auf den Fehlerterm in eine nicht-lineare Funktion als Argument übergeben. Für die Parametrierung muss damit diese nicht-lineare Funktion als Ganzes gefunden werden statt der einzelnen a und b Parameter wie in ARX. Analog kann hier statt ARX auch ARMAX als Basis verwendet werden, wodurch die Methode **NARMAX** (Nonlinear AutoRegressive Moving Average with eXogenous inputs) entsteht [BILL13]. Für die Annäherung an die unbekannte nicht-lineare Funktion werden Methoden wie neuronale Netze, Wavelet-Funktionen und polynomiale Basen verwendet [DONG23].

Eine Alternative zu den obigen Modellen sind die Zustandsmodelle. Diese gehen im Hintergrund auf Zustandsübergänge zurück. Diese können dabei linear oder nicht-linear sein und auf andere Modelle wie Markov-Modellierungen zurückgehen [RADA17]. So kann ein Zustandsmodell mit Übergängen dargestellt werden als

$$\dot{x}(t) = Ax(t) + B\vec{u}(t) + w(t) \quad (3.5)$$

$$\vec{y}(t) = Cx(t) + D\vec{u}(t) + q(t), \quad (3.6)$$

wobei w und q das Prozessrauschen, die Matrizen A, B, C, D die Systemmatrizen im Zustandsraum beschreiben. Der Vektor x ist damit der Zustandsvektor des Systems zum definierten Zeitpunkt.

Parameterabschätzung

Basierend auf dem gewählten Modell müssen nun die Parameter des Modells bestimmt werden. Eine übliche Metrik hierbei ist die Summe der quadrierten Fehler (engl. »Least Squares«). Seien nun alle abgeschätzten Parameter in dem Schätzvektor $\vec{\theta}$ beschrieben, so ergibt sich

$$\vec{\theta} = \arg \min_{\theta} \sum_{t=1}^N (\vec{y}(t) - \vec{y}(t|\theta))^2, \quad (3.7)$$

wobei $\vec{y}(t|\theta)$ der vorhergesagte Ausgangswert bei gegebenen Parametern θ ist.

Übertragung auf Synchronisierung

Die Struktur der Systemidentifikation selbst ist nicht in erster Linie eine Methode zur Synchronisierung von Zeitdatenreihen, sondern eine umfangreiche Modellierung des Systems. Allerdings kann die Erstellung des Modells für eben jene Synchronisierung angewendet werden.

Übertragen auf die Zeitdatenreihen können die Datenreihen in Eingangs- und Ausgangsparameter aufgetrennt werden. Sind die Werte aus der Zeitdatenreihe ${}_rZ$ die Eingangssignale und die Werte aus ${}_lZ$ die Ausgangssignale, beschreibt das Modell der Systemidentifikation den Zusammenhang zwischen diesen beiden Zeitdatenreihen. Dies kann auch auf mehrere Zeitdatenreihen erweitert werden mit $\mathfrak{Z}_{Out} = \mathfrak{Z} \setminus \mathfrak{Z}_{In}$, sodass die Eingangs- und Ausgangsparameter disjunkte Mengen aus der Gesamtheit der Zeitdatenreihen sind. Daraus ergibt sich:

$$\begin{aligned} SI &= \{\vec{u}(t_i), \vec{y}(t_i)\} \\ \vec{u}(t_i) &= (v_i^k, v_i^l, \dots) \quad | \quad \forall (t_i, v_i) \in \mathfrak{Z}_{In} \\ \vec{y}(t_i) &= (v_i^k, v_i^l, \dots) \quad | \quad \forall (t_i, v_i) \in \mathfrak{Z}_{Out} \end{aligned}$$

Damit können Modelle der Zusammenhänge zwischen Messwerten oder Stellparametern und den Messdaten der Zeitdatenreihen erstellt werden. Solche erstellten Modelle dienen

nun der Vorhersage von Werten, wobei Abweichungen durch Asynchronizität von neuen Daten erkannt werden können und eine Verschiebung durchgeführt werden kann. Daraus ergibt sich die Projektion $\Phi_{r,l}$, beziehungsweise eine Projektionsfunktion $\Phi_{\mathfrak{Z}_{In},\mathfrak{Z}_{Out}}$.

Eine andere Übertragung beschrieb MAES unter Nutzung der **stochastischen Subraumidentifikation**. Dabei wird die Gesamtheit der Zeitdatenreihen nicht in Eingangs- und Ausgangssignale aufgeteilt, sondern die Gesamtheit der Signale als Ausgangssignale behandelt [MAES16]. Aus dem Wissen um die Umgebung wird die Annahme von Dämpfungseffekten und Signalpropagation getroffen. Die nicht synchronisierten Zeitdatenreihen werden genutzt, um ein initiales Dämpfungsmodell zu erstellen. Gegenüber dem idealen Dämpfungsmodell der Annahme wird dieses errechnete Modell verglichen und eine Verschiebung der Zeitdatenreihen vollzogen, um die Abweichung der Dämpfungsmodelle zu minimieren. Dabei wird diese Methode genauer in der Anpassung, wenn mehr Zeitdatenreihen existieren, die von dieser Dämpfungsannahme betroffen sind. Dabei ist die Anpassung allerdings ein globaler Verschiebungsparameter Δ_{SSI} , also ein konstanter Offset für die gesamte Zeitdatenreihe. Als Projektionsfunktion beschreibt dies

$$\Phi_{n,m}(z_i^n) = z_i^m + \Delta_{SSI}^{n,m} \quad (3.8)$$

$$\Phi_{n,m}^{-1}(z_i^m) = z_i^n + \Delta_{SSI}^{m,n}, \quad (3.9)$$

wobei $\Delta_{SSI}^{n,m}$ den Offset von der Zeitdatenreihe $_nZ$ zu $_mZ$ aus der stochastischen Subraumidentifikation bezeichnet. Aus der gemeinsamen Erstellung aller Verschiebungsparameter ist die Umkehrfunktion ebenfalls direkt gegeben.

Abschließende Eigenschaften

Die vorgestellten Ansätze im Bereich der Systemidentifikation erlauben eine Synchronisierung von Zeitdatenreihen. Für die regulären Methoden der Systemidentifikation muss dabei eine Aufteilung in Eingangs- und Ausgangsparameter vorgenommen werden, welche die Erstellung von Modellen erlaubt, die nicht nur auf die Stützstellen, sondern auch darüber hinaus Vorhersagen ermöglichen. Bei der Methode der stochastischen Subraumidentifikation ist diese Auftrennung nicht notwendig, die Synchronisierung erfolgt hierbei allerdings nur über globale Verschiebungsparameter. Dadurch ist diese Methode bei langen Zeitdatenreihen, welche einen Gangfehler aufweisen, nur begrenzt anwendbar.

Bei all diesen Methoden spielt der Aufbau der Messsysteme eine entscheidende Rolle. Nur, wenn es Verbindungen wie Dämpfungen, Korrelationen oder Signalpropagationen zwischen den gemessenen Zeitdatenreihen gibt, kann die Systemidentifikation ein entsprechendes Modell erstellen.

3.1.3 PROFINET Sequence of Events

PROFINET Sequence of Events

In kritischen Infrastrukturen ist die Nachverfolgbarkeit von Ereignissen und ihrer Reihenfolge besonders wichtig. In solchen Systemen kommen daher oftmals Bussysteme mit hohen Anforderungen zum Einsatz, wie beispielsweise PROFINET [SEST17]. Zur Sicherstellung der korrekten Reihenfolge von Ereignissen wurde daher eine Spezifikation entwickelt. Diese **Sequence of Events** Spezifikation beschreibt die Synchronisierung von Ereignisdaten in die korrekte Reihenfolge.

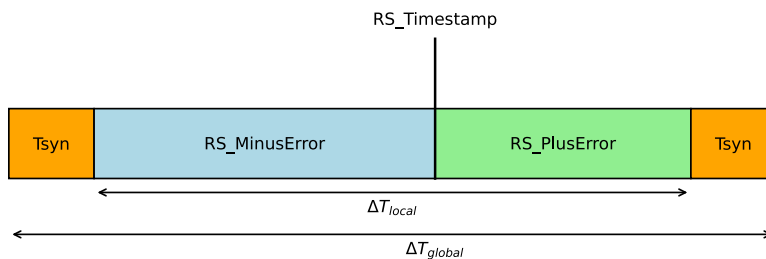


Abbildung 3.4: Angepasste Zeitstempel in Sequence of Events. RS_Timestamp ist der eigentliche Zeitstempel, während RS_MinusError und RS_PlusError die Unsicherheitsbereiche angeben. (Basierend auf [PI W17])

Kern der Spezifikation ist der Systemaufbau für die Produktion oder Infrastruktur. Für alle Ereignisdaten existiert eine zentrale Stelle (Historian), wo diese Daten zusammenlaufen aus allen verteilten Systemen. Alle Datenquellen sind direkt, oder über vermittelnde Systeme, mit dem Historian verbunden. Alle Quellsysteme vergeben erweiterte Zeitstempel für die Ereignisse, die von den vermittelnden Systemen erweitert werden. Diese Erweiterung ist dargestellt in Abbildung 3.4. Anstatt des normalen Zeitstempels (RS_Timestamp) werden zusätzliche Informationen zur Unsicherheit und Asynchronizität in den Feldern darum hinterlegt. Diese Unsicherheiten umfassen dabei auch Informationen zur Unsicherheit, die das System errechnen kann oder abschätzt, wie die interne Latenz oder den Firmware-Jitter.

Die Systeme sollen grundsätzlich regelmäßig synchronisiert werden (Online Synchronisierung gemäß 2.9). Wird ein System synchronisiert und verändert seine Zeitberechnung (Korrekturparameter), wird dies in den Feldern der Ereignisse vermerkt. Ist ein System vorübergehend getrennt oder schlagen Synchronisierungen vorübergehend fehl, wird die Information nach erfolgreicher Synchronisierung zusammen mit den nachträglich zu versendenden Ereignisdaten versendet.

Dies bedeutet, dass für eine Reihe von Ereignisdaten ${}_IZ$ die Kette $C_{ursprung} \succ \dots \succ C_{historian}$ bekannt ist. Weiterhin existiert für jedes System C_n für jedes Datum eine Information zur Ungenauigkeit. Im Historian können nun alle Zeitstempel als Zeitfenster, gemäß ΔT_{global} aus 3.4, interpretiert werden. Liegen Fenster überschneidungsfrei hintereinander, ist die Reihenfolge eindeutig. Andernfalls werden Konfliktlösungsstrategien angewendet. Dies können Priorisierungen von Ereignisarten, ID-basierte Sortierungen oder die Sortierung nach Unsicherheitsbereichen direkt sein [PI W17].

Diese Methode ist nicht spezifiziert als direkte Synchronisierungsmethode von Zeitdatenreihen, sondern stellt eine Reihenfolge von Ereignisdaten sicher. Aus diesem Grunde ist keine direkte Projektionsfunktion Φ für die Zeitdatenreihen aus Sequence of Events zu extrahieren. Allerdings ist die Erstellung der Informationen zu Unsicherheiten relevant, da dadurch die Umrechnung von Historian zu seinen verbundenen Systemen $\Phi_{historian,n}$ möglich. Dadurch entsteht ein Synchronisierungsgraph gemäß 2.20, wobei $C_{historian}$ die einzige Quelle ist. Die Struktur des Graphen ist damit eine Baumstruktur. Als Konsequenz kann $C_{historian}$ auf alle Systeme synchronisiert werden, entweder direkt oder als Komposition. Der Synchronisierungsgraph kann bei vorhandenen erweiterten Zeitstempeln auch dynamisch erstellt werden, falls nicht vorab bekannt [MAIE11].

Auch wenn Sequence of Events nur für Ereignisse spezifiziert ist, könnte eine Anwendung auf Zeitdatenreihen zur Sicherstellung der Reihenfolge von Daten mehrerer Zeitdatenreihen möglich sein, wenn Zeitdatenreihen als hochfrequente Ereignisse interpretiert werden [SCHM23c].

3.1.4 Weitere Ansätze

Further Approaches

Neben den dargestellten Methoden existieren weitere Ansätze, die zwar weniger aktive Forschungsfelder, jedoch relevante Methoden sind. In der Methode des **Subsequence Matching** wird mit dem Wissen zur Durchführung einer Produktion und den dortigen Phasen eine Klassifikation der Subsequenzen der Zeitdatenreihen vorgenommen. Dies ist ein

weitgehend manueller Prozess unter Einbeziehung von Expert*innen. So hat NAVARO-BRULL gezeigt, dass so mehrstündige Mischprozesse (Batch) in drei Hauptphasen unterteilt werden konnten, woran die Zeitdatenreihen manuell ausgerichtet wurden. Die Hauptarbeit liegt hier bei der manuellen Klassifikation und Vorsynchronisierung. Im Anschluss wird eine detaillierte Synchronisierung mittels Dynamic Time Warping durchgeführt. [AG22]

Ein anderer verbreiteter Ansatz zur Synchronisierung sind direkte Methoden der **Linearisierung**. So können mit einfachen linearen Programmen die Zeitdatenreihen zwischen einem definierten Start- und Endpunkt gleichmäßig linear verteilt und damit synchronisiert werden. Dabei stellte LUO fest, dass dieser Ansatz lediglich für verhältnismäßig kurze Zeitdatenreihen funktioniert (Bei LUO unter 20 Minuten) [LUO11].

3.2 Anforderungen und Metriken für die Offline Zeitdatenreihensynchronisierung

Requirements and Metrics for the Offline Time-Series Data Synchronization

Für die Offline Synchronisierung von Produktionsdaten in der Batteriezellfertigung ergeben sich aus den vorangegangenen Abschnitten Anforderungen, die ein Lösungsansatz adressieren sollte. Bestehende Standards und Taxonomien (siehe 2.6.2) befassen sich mit Datenqualität, allerdings nicht mit der Qualität von Offline Synchronisierungen. In den Arbeiten zu den Ansätzen für die Offline Synchronisierung (siehe 3.1) existieren ebenfalls keine übergreifend eingesetzten Metriken, sondern lediglich einzelne verwendete Metriken [SCHM24a]. Daher sind hier die Anforderungen an die Offline Synchronisierung als Metriken für diese Arbeit dargestellt.

Aus den Eigenschaften der Batteriezellfertigung und der zu erwartenden Menge von Zeitdatenreihen ergeben sich drei Bereiche von Anforderungen an die Offline Synchronisierung. Im ersten Bereich geht es um die Skalierbarkeit und Effizienz der Methoden, der zweite Bereich beschreibt die Anwendbarkeit von Methoden und zuletzt fokussiert der dritte Bereich die Qualität und Güte der Synchronisierung. Diese Bereiche sind dabei auf technische Eigenschaften ausgerichtet. Organisatorische und geschäftsprozessseitige Anforderungen sind spezifisch für die anwendenden Unternehmen, daher hier im Sinne einer technischen Betrachtung ausgelassen.

Als Kurzschreibweise der Bewertungen kommen Harvey Balls (○ bis ●) zum Einsatz. Die genauen Stufen je nach Metrik sind im Folgenden ebenfalls beschrieben.

Effizienz und Skalierbarkeit

Die Offline Synchronisierung von mehreren Zeitdatenreihen bedeutet eine entsprechende Menge an Arbeit und Rechenzeit, die aufgewendet werden muss. Wie in der Umfrage (siehe 2.1.2) gezeigt, besteht der Großteil der Offline Synchronisierung von Produktionsdaten in manueller Arbeit. Da manuelle Arbeit von vielen Faktoren abhängt, ist die Quantifizierung nicht allgemein möglich. Daher wird die manuelle Arbeit als Basisfall betrachtet, bei dem paarweise Zeitdatenreihen betrachtet und synchronisiert werden. Die Metriken in diesem Bereich sind daher relative Maße, die einen grundsätzlichen Effizienzgewinn gegenüber diesem Basisfall darstellen. Für die Berechnung der Laufzeiten sind die Größen N als Anzahl der Zeitdatenreihen, T die durchschnittliche Anzahl von Datenpunkten pro Reihe und G als Anzahl der betrachteten Geräte (falls relevant) berücksichtigt.

Die erste Metrik beschreibt daher den **Effizienzgewinn für kleinskalige Produktionen**. Gegenüber dem Basisfall stehen die Effizienzgewinne durch Anwendung der Methoden im Vordergrund, die bei wenigen Zeitdatenreihen eingesetzt werden. Als Grenze ist hierzu eine Größenordnung von maximal drei Zeitdatenreihen und relevanten Geräten gesetzt. Für die Bewertung ist eine dreistufige Bewertung vorgesehen. Stufe 0 entspricht einem überwiegend manuellen Ablauf, der keinen Effizienzgewinn gegenüber dem Basisfall erwarten lässt, oder mehr Aufwand als der Basisfall bedeutet. Die Stufe 1 beschreibt, dass ein signifikanter Teil der Arbeit durch manuelle Konfiguration und Parametrierung bestimmt ist, wonach die Synchronisierung stets manuell ausgelöst werden muss. Auch Methoden, die bereits bei zwei Zeitdatenreihen eine mindestens quadratische Komplexität ($\Omega(m^2)$) aufweisen, fallen in diese Stufe. Die Stufe 2 ist erreicht, wenn die manuellen Schritte reduziert sind und die Synchronisierung automatisiert werden kann bei einer Laufzeit in $o(m^2)$.

Als zweite Metrik befasst sich der **Effizienzgewinn für großskalige Produktionen** mit dem Einfluss von einer steigenden Zahl an Zeitdatenreihen und Geräten. Der Basisfall geht hier von der manuellen paarweisen Synchronisierung aus. Dies führt allein durch die Skalierung zu einer mindestens quadratischen Laufzeit zuzüglich der Komplexität der eigentlichen Methode. Für die Metrik gilt daher die Stufe 0, wenn die Methode paarweise eingesetzt werden muss oder die Laufzeit in $\Omega(N^2 \cdot T^2)$ liegt. Methoden, die maximal ein superlineares Wachstum haben ($\mathcal{O}(N \cdot T \cdot \log(N) \cdot \log(T))$), sind gut skalierbar und damit in Stufe 2. Methoden, deren Laufzeiten zwischen Stufe 0 und Stufe 2 liegen, sind per Definition in Stufe 1.

Als Kurzschreibweise wird für die Metriken der Effizienz und Skalierung die Darstellung

○ für Stufe 0, ◐ für Stufe 1 und ● für Stufe 2 verwendet. Die Belegung ist bei der Gruppe gleichartig, da die gleiche Menge von Stufen existiert.

Anwendbarkeit

Methoden unterliegen potenziell Einschränkungen bezüglich ihrer Anwendung. In diesem Bereich sind potenzielle Einschränkungen aus Literatur und Methoden extrahiert, die in der Batteriezellfertigung relevant sind. Die Bewertungsskala ist hier grundsätzlich sprachlich kontraintuitiv. Eine niedrige Stufe entspricht grundsätzlich einer hohen Einschränkung. Dies dient der besseren Erfassung von Stärken und Schwächen von Methoden konsistent mit den anderen Metriken.

Auch wenn es um die Offline Synchronisierung geht, greifen Methoden möglicherweise auf Informationen oder Eigenschaften aus der Produktion (»in situ«) zurück. Daraus ergeben sich dann **In Situ Bedingungen**, die erfüllt sein müssen, um eine Anwendbarkeit zu ermöglichen. Stufe 0 (○) gilt dabei für Methoden, die harte Anforderungen an den Betrieb der Produktion stellen (z. B. an die Datenaufnahme oder Geräte), und ohne die Einhaltung dieser Anforderungen nicht funktionieren. Existieren diese Anforderungen grundsätzlich, aber ohne die Einhaltung kann die Methode eingeschränkt unter Rückgriff auf Informationen zur Produktion arbeiten, fällt sie in die Stufe 1 (◐). Kann sie sogar eingeschränkt arbeiten ohne weiteren Rückgriff auf diese Informationen, fällt sie in Stufe 2 (◑). Existieren keine harten Anforderungen an den Betrieb selbst, es müssen jedoch Informationen zur Produktion (z.B. Aufbau) bereitgestellt werden, so gilt die Stufe 3 (◒). Nur, wenn keinerlei Anforderungen an den Betrieb gestellt werden und keine Informationen notwendig sind, gilt die höchste Stufe 4 (●).

In einem komplexen Produktionssystem wie der Batteriezellfertigung ist das Einbeziehen weiterer Datenquellen von extern oder intern durch weitere Messtechnik wahrscheinlich. Für die Synchronisierung bedeutet dies ebenfalls, dass die nachträgliche **Erweiterung um Datenquellen** möglich sein muss und diese Zeitdatenreihen auch synchronisiert werden müssen. Ist das nachträgliche Hinzufügen einer weiteren Datenquelle grundsätzlich unmöglich, so gilt die Stufe 0 (○). Ist das Hinzufügen teilweise möglich, bezogen auf bestimmte Anforderungen (z.B. Art der Quelle oder In Situ Bedingungen), so gilt die Stufe 1 (◐). In der Stufe 2 (◑) ist das Hinzufügen grundsätzlich möglich, jedoch resultiert dieses Hinzufügen in umfangreichen Arbeiten zur erneuten Synchronisierungen (z.B. mehrere Φ bestimmen oder Modellparameter neu bestimmen). Die letzte Stufe (●) bedeutet, dass neue Datenquellen hinzugefügt werden können und für die Synchronisierung eine direkte

automatisierbare Integration existiert (z.B. eine einzelne Projektion Φ), sodass der Aufwand gering ist.

Für die **Unabhängigkeit der Datenreihen** ist der Zusammenhang zwischen den aufgenommenen Daten im Fokus. Zwei Messungen sind abhängig voneinander, wenn eine Struktur oder Änderung in einer Messung einen Einfluss auf die andere Messung hat. Für die Metrik bedeutet dies, dass die Nutzung von diesen Abhängigkeiten einen Mangel bei der Anwendbarkeit für unabhängige Datenreihen bedeutet. Daher fallen Methoden, die zwangsweise darauf angewiesen sind, dass die zu synchronisierenden Datenreihen eine bekannte direkte Abhängigkeit aufweisen in die Stufe 0 (○). Funktioniert die Synchronisierung allerdings auch bei Messungen, die nur indirekte oder unbekannte Abhängigkeiten aufweisen, so beschreibt dies die Stufe 1 (◐). Die letzte Stufe (●) besagt hier, dass die Synchronisierung angewendet werden kann bei Messungen, die keinerlei Abhängigkeiten zueinander aufweisen.

Qualität

Der letzte Bereich befasst sich mit der Güte einer Synchronisierung und den Erwartungen, die Benutzende an diese Methoden stellen können. Die Qualitätsmetriken gelten dabei nur für Fälle, in denen die Methode jeweils auch anwendbar ist, da die Qualität sonst per Definition immer auf der niedrigsten Stufe wäre.

Daher ist die erste Metrik die **erwartete Qualität** der Synchronisierung. Hierbei geht es um die algorithmische Güte, die von der Anwendung auf Zeitdatenreihen ausgeht und die Eigenschaften, die die Methoden zur Bewertung zur Verfügung stellen. Die niedrigste Stufe (○) bedeutet, dass die Methode keinerlei Aussagen zur Güte treffen kann (»best effort«). Kann eine Methode hingegen auf Basis des eingesetzten Modells grundsätzliche Abschätzungen und Obergrenzen zu Abweichungen angeben, so fällt diese Methode in Stufe 2 (◐). Ist eine Methode in ihrer Qualität allerdings stark abhängig von lokalen Effekten wie Ausreißern, so gilt sie nur als Stufe 1 (◑). Geht die Methode allerdings noch weiter und beschreibt eine hohe Güte sowie Garantien für diese, so fällt die Methode in die Stufe 3 (●).

Wie in den vorherigen Abschnitten gezeigt, können Asynchronizitäten in langen Zeitdatenreihen nicht-lineare Verhaltensweisen aufzeigen. Daraus ergibt sich die Unterstützung für **Nicht-Linearität** als Metrik zur Einordnung, ob eine Methode mit derartigen Zeitdatenreihen umgehen kann. Die Bewertung startet hier mit der niedrigsten Stufe (○), die besagt, dass die Methode keinerlei Unterstützung für diese Art von Asynchronizität bietet. Bringt

eine Methode nun eine grundsätzliche Unterstützung mit durch partielle Funktionen, so fällt sie in die Stufe 1 (◐). Wenn eine Methode nun eine volle Unterstützung für totale nicht-lineare Modelle mitbringt, so fällt sie in die höchste Stufe (●).

Eine letzte Metrik befasst sich mit den **Plausibilitätsprüfungen**. Eine Synchronisierung kann mathematisch eine hohe Qualität aufweisen, jedoch gegen Regeln der Domäne oder Physik verstoßen. Gerade bei einer Skalierung ist die manuelle Prüfung von Plausibilität in den Daten nicht mehr machbar, sodass die Methoden idealerweise eine Unterstützung für derartige Prüfungen bereitstellen. Stellt eine Methode keinerlei Unterstützung bereit und kann keine Domänenregeln beachten, so fällt sie in die niedrigste Stufe (○). Besteht die Möglichkeit, manuell Anpassungen vorzunehmen, um Regeln aus der Domäne in die Modelle einzubringen, so ist die Stufe 1 erreicht (◐). Die höchste Stufe (●) beschreibt Methoden, die Regeln dynamisch aufnehmen können und nach der Durchführung einer Synchronisierung die Plausibilität gegen diese Regeln prüfen.

Abschließende Eigenschaften

Aus den obigen Bereichen und dazugehörigen Metriken ergibt sich ein Rahmenwerk, mit dem sich Methoden der Offline Synchronisierung untersuchen lassen. Der Betrachtungsrahmen ist dabei auf technische Aspekte beschränkt. Der Basisfall der manuellen Synchronisierung ist dabei schwer zu definieren und zu quantifizieren, sodass die Metriken teilweise relativ formuliert sind. Letztlich sind die Anforderungen aus der skalierenden Batteriezellproduktion in den Metriken integriert und damit die Anwendung des Rahmenwerks in dieser Domäne sichergestellt. [SCHM24a]

3.3 Evaluierung der existierenden Ansätze

Evaluation of Existing Approaches

Im vorangegangenen Abschnitt wurden die Bereiche und Metriken für die Anforderungen an die Offline Synchronisierung von Zeitdatenreihen in der Batteriezellproduktion gezeigt. In Folgenden werden die relevanten Ansätze (siehe 3.1) entlang dieser Metriken bewertet. Die Gesamtbewertung ist mit der Stufenschreibweise von oben als Harvey Balls in Tabelle 3.1 dargestellt.

		Effizienz		Anwendung			Qualität		
		Klein-skalig	Groß-skalig	In Situ	Erweiter-barkeit	Unabhängige Messungen	Erwartung	Nicht-Linearität	Plausibilitäts-prüfungen
	Manuell	○	○	◐	◐	○	○	◐	●
	Linear	●	◐	●	●	●	○	○	○
	DTW	◐	○	◐	◐	○	◐	◐	○
DTW	FastDTW	●	◐	◐	◐	○	◐	◐	○
	ShapeDTW	◐	○	◐	◐	○	◐	◐	○
	ARMAX	◐	◐	◐	◐	○	○	○	◐
SI	NARMAX	◐	◐	◐	◐	○	○	●	◐
	SSI	○	○	◐	◐	◐	◐	○	◐
	SoE	●	●	○	◐	●	●	◐	○

Tabelle 3.1: Die Bewertung der relevanten Ansätze aus 3.1 gegen die Anforderungen an die Offline Synchronisierung von Produktionsdaten aus der Batteriezellfertigung aus 3.2. (Angelehnt an [SCHM24a])

Basisfälle

Die Verbesserung der Effizienz ist beim manuellen Basisfall per Definition die niedrigste Stufe, da dies als untere Grenze dient. Die direkte Linearisierung bedeutet, dass Start- und Endpunkt von zwei Zeitdatenreihen gleichgesetzt werden und der Bereich dazwischen linear gestreckt/gestaucht wird. Dies ist in linearer Laufzeit möglich. Ist dies für ein Paar durchgeführt, so kann jede weitere Zeitdatenreihe auf die gleichen Start- und Endpunkte gesetzt und linearisiert werden. Damit ist bei der Skalierung auch eine Laufzeit von $\mathcal{O}(N \cdot T)$ zu erwarten. Die Auswahl des Start- und Endpunktes bei Zeitdatenreihen, die nicht gleichzeitig gestartet oder gestoppt wurden, ist dabei erneut eine manuelle Arbeit, die der Automatisierung entgegensteht.

Der Basisfall der manuellen Synchronisierung greift im Idealfall zurück auf Informationen und Bedingungen aus der Produktion. Sind lediglich Informationen im Nachhinein verfügbar, so ist die Anwendung beschränkt aber möglich. Für die Linearisierung reicht hier die aus, die Zeitdatenreihen zu übergeben, sodass keine weiterführenden Bedingungen grundsätzlich gestellt werden. Das Hinzufügen einer weiteren Datenquelle führt bei der

manuellen Synchronisierung zu einer nicht-vorhersehbaren Arbeit, sodass dies als Stufe 2 abgeschätzt wird. Bei der Linearisierung ist die Arbeit automatisierbar und umfasst eine einzelne Berechnung, sodass der Aufwand als gering eingeschätzt wird. Bei der manuellen Synchronisierung sind bekannte Zusammenhänge notwendig um diese zu erkennen und eine Synchronisierung durchzuführen. Die Linearisierung hingegen bezieht lediglich die Zeitinformationen ein, sodass die Datenreihen irrelevant sind für die Synchronisierung.

Weder die manuelle Synchronisierung noch die Linearisierung bieten Möglichkeiten zur Abschätzung der Güte des Modells, da sie von den externen Vorgaben, Parametrierungen und Personen abhängen. Aufgrund der direkten Linearisierung bildet diese Methode entsprechend keine nicht-linearen Zusammenhänge ab. Für die manuelle Synchronisierung ist die Betrachtung komplizierter. Grundsätzlich können Menschen lineare Zusammenhänge erkennen. Bei nicht-linearen Zusammenhängen allerdings ist diese Fähigkeit bei Menschen stark eingeschränkt [CHER23]. Dadurch ergibt sich, dass in der manuellen Synchronisierung grundsätzlich nicht-lineare Funktionen abgebildet werden können, aber durch den Menschen stark eingeschränkt ist. Für die Plausibilitätsprüfung erlaubt die manuelle Synchronisierung eine Prüfung und dynamische Hinzunahme von Regeln und Erkenntnissen. Die Linearisierung hingegen nutzt selbst keinerlei Regeln oder Prüfungen, da sie lediglich auf die Zeitinformationen schaut.

Dynamic Time Warping

Für Dynamic Time Warping sind hier die normale Variante sowie die Anpassung ShapeDTW und die optimierte Variante FastDTW ausgewählt. Allen DTW Varianten ist gemein, dass sie nicht vollautomatisch laufen können, sondern eine Konfiguration und einen manuellen Start benötigen (Vorauswahl des Betrachtungsfensters, etc.). Dadurch entscheiden die Laufzeiten über die genaue Zuordnung der ersten Metrik. Bei FastDTW ist eine lineare Laufzeit ($\mathcal{O}(T)$) anzunehmen, während die anderen Varianten in $\mathcal{O}(T^2)$ liegen. Bei der Skalierung auf N Zeitdatenreihen müssen alle Varianten paarweise Synchronisierungen durchführen. Daraus ergeben sich $\binom{N}{2}$ Paare, die als Koeffizient in die Laufzeit eingerechnet werden. Dies entspricht einem dominanten Faktor N^2 . Damit liegen DTW und ShapeDTW in $\mathcal{O}(N^2 \cdot T^2)$, FastDTW in $\mathcal{O}(N^2 \cdot T)$.

Für die In Situ Bedingungen benötigen die DTW Methoden keine harten Anforderungen an den Betrieb, allerdings Informationen zum Aufbau, welcher beschreibt, welche Verläufe in Datenreihen ähnlich aussehen sollten (Stufe 3). Wird eine neue Datenquelle hinzugefügt in das System kann die entsprechende Zeitdatenreihe integriert werden. Dies geschieht

manuell, da zunächst bestimmt werden muss, welche Teile der Zeitdatenreihe gegen welche anderen Zeitdatenreihen synchronisiert werden können. Die Einschränkung für DTW ergibt sich bei der Betrachtung der unabhängigen Zeitdatenreihen. Alle DTW Methoden basieren auf bekannten Signalverläufen, die gegenseitig ausgerichtet werden müssen. Sind zwei Zeitdatenreihen nun komplett unabhängig voneinander, so kann keine Synchronisierung durchgeführt werden und dennoch applizierte DTW führen zu unvorhersehbaren Zuordnungen.

Für die Dynamic Time Warping Varianten existieren mathematische Modelle und Abschätzungen zur Güte der entstehenden Synchronisierung. Dabei kann bei einem klassischen DTW von dem besten Pfad durch das Kostengitter und dadurch geringen Abweichungen ausgegangen werden. Für die Optimierungen wie FastDTW und Anpassungen wie ShapeDTW wird diese Grenze teilweise aufgegeben, allerdings bestehen dennoch obere Schranken. DTW und FastDTW sind dabei in der Qualität direkt abhängig von lokalen Ausreißern und Effekten, die bei ShapeDTW durch die Formmerkmale teilweise adressiert werden können. Durch den Aufbau als partielle Funktionen können DTW und FastDTW partielle nicht-lineare Funktionen abbilden. ShapeDTW ist mehrstufig aufgebaut. Dabei ist der Schritt der Berechnung von Formeigenschaften ähnlich zu einer totalen nicht-linearen Funktionsdarstellung. Die resultierende Rückrechnung auf die Zeitdatenreihen zur Synchronisierung erfolgt allerdings als partielle Funktion, sodass ShapeDTW dennoch eine partielle nicht-lineare Funktionsweise darstellt. Alle DTW Varianten bilden die Formen der Datenreihen aufeinander ab. Dabei beziehen sie kein Domänenwissen oder kontextuelle Beschränkungen ein.

Systemidentifikation

Für die Systemidentifikation sind die lineare Variante ARMAX und die nicht-lineare Variante NARMAX zusammen mit der besonderen Form der stochastischen Subraumidentifikation (SSI) betrachtet. Die Modelle können nicht vollautomatisch erstellt und zur Synchronisierung verwendet werden, sondern müssen manuell parametrisiert werden und danach vom Ergebnis aus die Synchronisierung durchgeführt werden. Daher entscheidet die Laufzeit über die Zuordnung in der Effizienz. Dabei führen ARMAX und NARMAX zunächst zusätzliche Parameter ein. Ein signifikanter Anteil der Laufzeit ist damit die Modellordnung p , also die Menge der Parameter im Modell. Für ARMAX ergibt sich damit eine abgeschätzte Laufzeit in $\mathcal{O}(T \cdot p^2)$. Für NARMAX ist die Abschätzung stark von der Ordnung k der nicht-linearen Terme abhängig. Dadurch liegt NARMAX in $\mathcal{O}(T \cdot p^k)$, was eine hohe Bandbreite darstellt. Zuletzt nutzt SSI mathematisch eine Singulärwertzerlegung, was die Berechnung

unabhängig von Modellparametern wie bei ARMAX macht, jedoch direkt mit der Länge der Zeitdatenreihen skaliert. Mit der dominanten Singulärwertzerlegung kann die Laufzeit abgeschätzt werden zu $\mathcal{O}(T^3)$. Aufgrund des Ansatzes von Systemidentifikation kann bei der Skalierung auf mehrere Zeitdatenreihen anstatt der Erstellung von paarweisen Modellen ein Gesamtmodell erstellt werden. Die quadratische Komponente des paarweisen Vergleichs wird dadurch auf eine lineare Komponente reduziert. Bei steigender Menge von Zeitdatenreihen ist es möglich, dass die Menge der Modellparameter erhöht werden muss. Eine Abschätzung ist hier allerdings nicht allgemein möglich. Bei einer linearen Zunahme von Modellparametern würde dies bei ARMAX und NARMAX zu einem exponentiellen Wachstum der Berechnungskomplexität führen, während eine logarithmische Zunahme nur eine superlineare Komplexität zu erwarten ist. Dabei ist die Komplexität von NARMAX stets höher als bei ARMAX. Diese Unsicherheit führt hier zunächst zu einer Bewertung in Stufe 1. Bei SSI hingegen vergrößert sich die Singulärwertzerlegung in direkter Abhängigkeit der Menge von Zeitdatenreihen. Damit landet SSI bei einer Komplexität von $\mathcal{O}((N \cdot T)^3)$.

Die Anwendung von Systemidentifikation beruht auf dem Finden eines Modells basierend auf den Zeitdatenreihen und nachträglich die Zeitdatenreihen an dieses Modell anzupassen. Daraus ergeben sich keine harten Anforderungen an die Datenaufnahme und die laufende Produktion. Allerdings müssen für die grundsätzliche Auswahl der Modellparameter Informationen zum Aufbau und Zusammenhängen einbezogen werden, um eine Modellqualität sicherzustellen. Soll nun eine weitere Zeitdatenreihe hinzugefügt werden, ist dies grundsätzlich möglich, allerdings bedeutet dies eine Anpassung der Modellparameter oder Neuberechnung des Modells. Im Falle der SSI kann dies auch eine komplette Neuberechnung des Systems bedeuten. In jedem Fall muss auf Informationen zurückgegriffen werden, um die entsprechenden Modellparameter zu überprüfen. Die Systemidentifikation nutzt Zusammenhänge in den Zeitdatenreihen, daher ist die Anwendung bei Unabhängigkeit stark eingeschränkt. Existieren gar keine Zusammenhänge sind diese Methoden grundsätzlich nicht anwendbar. Existieren nun indirekte unbekannte Zusammenhänge, so sind ARMAX und NARMAX direkt von der Modellierung abhängig, was zumindest eine teilweise Aufdeckung der Zusammenhänge voraussetzt. Für SSI reicht hingegen, wenn Effekte korrekt durch das System propagieren, um diese zu detektieren.

Die zu erwartende Güte einer Synchronisierung bei Nutzung von Systemidentifikation hängt von dem Modell und den Modellparametern ab. Dabei ist auch eine Gefahr der Überanpassung (engl. »Overfitting«) gegeben. Eine Abschätzung der Güte ist damit zunächst nicht möglich. Die Abschätzung der Güte bei SSI ist nicht direkt möglich, allerdings zeigen die unterliegenden Methoden eine Erhöhung der Genauigkeit bei mehr Zeitdatenreihen, sodass die Genauigkeit tendenziell genauer wird bei der Skalierung. Dies wird hier analog

mit Stufe 1 bewertet. NARMAX bildet im Modell nicht-lineare Funktionen ab, sodass im finalen Modell eine totale nicht-lineare Funktion entsteht. ARMAX und SSI hingegen setzen auf lineare Funktionen und können daher nicht-lineare Funktionen nicht abbilden. Regeln der Domäne können in ARMAX und NARMAX in den Aufbau der Modelle als Beschränkungen eingetragen werden, sodass dort eine Plausibilitätsprüfung vorkonfiguriert werden können. Bei SSI ist ein analoges Vorgehen möglich durch Erweiterung des Zustandsraums und Einbeziehung als Regeln in der Berechnung. Diese Einbeziehungen beziehen sich alle auf die Erstellung des Modells, nicht der anschließenden Synchronisierung, sodass nur eine teilweise Plausibilitätsprüfung Anwendung findet.

Sequence of Events

Für die Methode der Sequence of Events ist die Effizienz strukturell bedingt hoch einzuschätzen. Die Methodik zielt darauf ab, automatisch ausgeführt zu werden. Die eigentliche Sortierung der Ereignisse läuft in $\mathcal{O}(T \log T)$ ab. Eine Rückrechnung vom zentralen System (Historian) auf die Uhr des Systems davor kann als linear angenommen werden. Damit ist die Sortierung der dominante Anteil. Wird nun die Menge an Geräten und Zeitdatenreihen skaliert, erhöht sich die Berechnungszeit für die Sortierung auf $\mathcal{O}(G \cdot T \cdot \log T)$. Da das Synchronisierungsnetzwerk eine Baumstruktur hat, ist die Berechnung aller Φ hier abzuschätzen als $\mathcal{O}(N \cdot T)$. Damit bleibt die superlineare Sortierung maßgeblich für die Komplexitätsbetrachtung.

Die Anwendbarkeit von Sequence of Events hängt direkt von dem Systemaufbau und den Verbindungen aller Geräte zum Historian ab. Für die Anwendung müssen daher alle Elemente die erweiterten Zeitstempelformate nutzen und alle Geräte müssen ihre Ungenauigkeiten sowie eventuelle Synchronisationen dokumentieren und senden. Dies sind hohe Bedingungen an die laufende Produktion (Stufe 0). Weitere Datenquellen können daher auch nur integriert werden, wenn ihre Aufnahme direkt an einem der Geräte des betrachteten Produktionsnetzwerks stattfindet (Stufe 1). Da die Synchronisierung komplett auf die Betrachtung der Systeme und den erweiterten Zeitstempeln zurückgeht, sind die Inhalte der Datenreihe irrelevant. Damit sind unabhängige Zeitdatenreihen uneingeschränkt möglich.

Wenn die Synchronisierung via Sequence of Events möglich ist, kann die Güte der Synchronisierung im Historian berechnet werden. Dies ist dabei eine klare obere Grenze als Garantie. Die Betrachtung der Nicht-Linearität hängt hier von den Konfliktlösungsstrategien ab. Die grundsätzliche Synchronisierung findet hier linear statt, jedoch kann beim Einsatz

einer Konfliktlösungsstrategie auf den Daten auch eine partielle nicht-lineare Funktion implizit dargestellt werden. Da Sequence of Events nicht auf die Daten selbst eingeht, sondern nur die Systemzeiten und Zeitstempelinformationen nutzt, findet keine weitere Prüfung der Plausibilität auf Datenebene statt.

3.4 Zwischenfazit und Handlungsfelder

Interim Conclusion and Required Actions

In diesem Kapitel wurden die grundsätzlichen Ansätze zur Synchronisierung von Produktionsdaten gezeigt. Der Schwerpunkt lag dabei auf den Methoden zur Offline Synchronisierung. Da keine allgemeingültigen Metriken und Anforderungen existieren, wurden eben jene in diesem Kapitel entwickelt und die bestehenden Ansätze entlang dieser Anforderungen bewertet.

Aus der Bewertung und Betrachtung der aktuellen Ansätze wird klar, dass die Offline Synchronisierung von Produktionsdaten in der Batteriezellfertigung eine aktuelle Herausforderung ohne allgemeingültige Lösung ist. Keine Methode bietet eine breite Anwendbarkeit bei gleichzeitiger Skalierbarkeit und Qualität der Synchronisierung. Jede der vorgestellten Methoden bietet allerdings spezifische Vorteile und Einschränkungen.

Mit Blick auf die Skalierung und Relevanz der Batteriezellfertigung ist daher eine tiefergehende Betrachtung der Offline Synchronisierung notwendig, und der Bedarf an einer Lösung, die diese Anforderungen bestmöglich adressieren kann, gegeben. Diese Anforderungen werden daher in dem Entwurf der neuen Methodik in Kapitel 4 einbezogen und in der Verifikation (Kapitel 5) überprüft.

Die Bewertung der bisherigen Methoden zeigt auch, dass die Hauptforschungsfrage (1.2) zum aktuellen Zeitpunkt nicht positiv beantwortet werden kann. Die vorgestellten Metriken adressieren allerdings bereits die erste Unterforschungsfrage (**UFF 1** »Welche Bewertungsmetriken für eine Offline Synchronisierung können abgeleitet werden?«) und beantworten diese Frage mit der vorgestellten Aufstellung in Abschnitt 3.2.

4 Entwicklung des Synchronisierungssystems

Development of a Synchronization System

Basierend auf den Anforderungen und den Eigenschaften der Offline Synchronisierung von Produktionsdaten wird in diesem Kapitel ein System entworfen, welches die Aufgabe der Offline Synchronisierung lösen kann. Die Auslegung eben jenes Systems und der darin liegenden Aufgaben werden in dem Abschnitt 4.1 allgemein und im Abschnitt 4.3 im Detail beschrieben. Die Modellierung der Eingangsgrößen und ihrer Rolle beschreibt Abschnitt 4.2 genauer. Abschließend werden die Details der Implementierung in Abschnitt 4.4 dargestellt, inklusive der eingesetzten Bibliotheken und Technologie-Stacks.

4.1 Konzeptionierung des Synchronisierungssystems

Design of the Synchronization System

In diesem Abschnitt wird das Konzept des entwickelten Synchronisierungssystems dargestellt. Die Grundidee geht dabei auf die bio-inspirierte Produktion zurück, insbesondere auf das Vorbild des menschlichen Gehirns und dessen Verarbeitung von Eingangsdaten, beschrieben in Abschnitt 2.10. Zunächst wird das Gesamtkonzept vorgestellt und im Anschluss auf die einzelnen Schritte detailliert eingegangen.

Die Grundzüge der Synchronisierung dieses Konzepts sehen die Nutzung von Daten, Informationen und Wissen gleichermaßen vor, um die Synchronisierung kontextuell durchzuführen. Anstatt komplette Zeitdatenreihen zu synchronisieren, werden Punkte gesucht, welche gemäß dem Kontext gleichzeitig sind, und somit als Stützstellen (»Fixpunkte«) dienen. Die Zeitdatenreihen können nun aufgeteilt werden, sodass kürzere Zeitdatenreihen zu synchronisieren sind und einzelne Punkte bereits direkt synchron sind. Für die daraus entstehenden Subsequenzen werden die Eigenschaften und der Kontext genutzt, um Vorschläge zu erstellen, mit welchen Methoden dort synchronisiert werden kann.

Der Ablauf inklusive der Eingangs- und Ausgangsparameter für die Synchronisierung ist im Schaubild 4.1 dargestellt. Als Eingangsgrößen dienen verschiedene Daten, Informationen und Wissen aus und um die Produktion. Auch manuelle Eingriffe und Nachsteuerungen sind möglich, auch wenn das Ziel des Systems ein möglichst hoher Automatisierungsgrad ist. An welcher Stelle diese Eingangsgrößen Verwendung finden ist mit den entsprechenden Markierungen (gelbbraune Kreise) in den Hauptschritten markiert. Auf die Generierung und Beschreibung dieser Eingangsgrößen geht der Abschnitt 4.2 genauer ein. Für die Beschreibung der Eingangsgrößen dient im Folgenden 3 als die Gesamtheit der Zeitdaten-

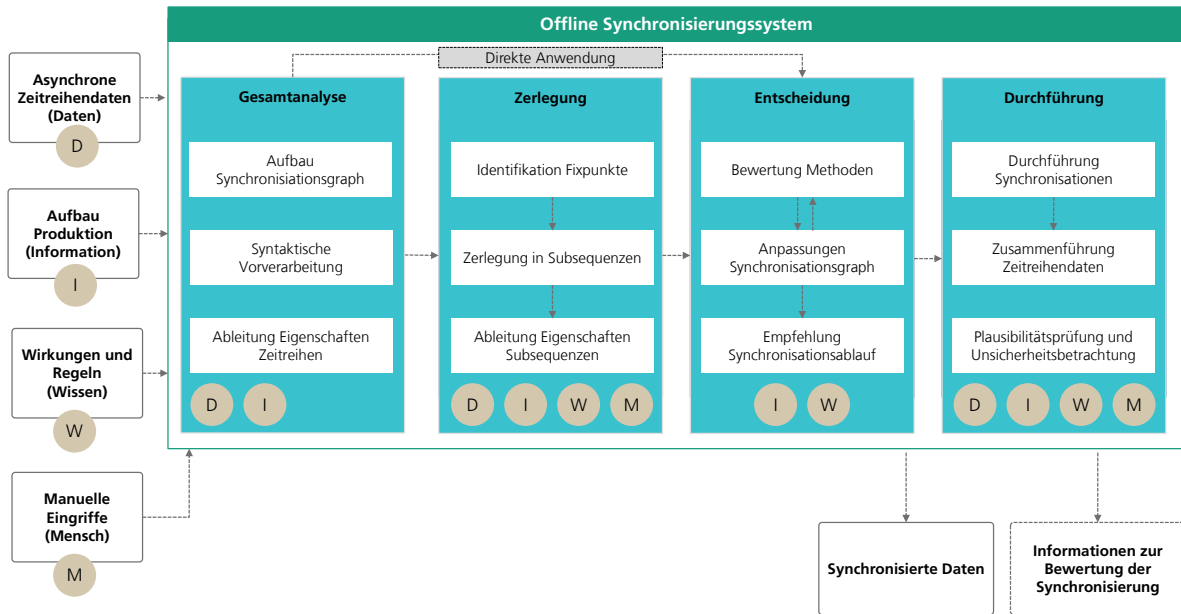


Abbildung 4.1: Aufbau des gesamten Offline Synchronisierungssystems. Die runden Markierungen beschreiben für die Schritte die mögliche Nutzung von Daten, Informationen, Wissen und manueller Eingriffe.

reihen wie zuvor definiert, $\mathcal{I}$ als die Gesamtheit von Informationen, $\mathcal{W}$ als die Gesamtheit des Wissens und zuletzt $\mathcal{M}$ als die Menge aller manuellen Eingriffe.

Das eigentliche Synchronisierungssystem (grüner Kasten) ist in vier Hauptschritte unterteilt. Im ersten Hauptschritt **Gesamtanalyse** werden vorbereitende Aktivitäten durchgeführt durch eine erste Analyse der Daten und Informationen der Produktion. Daraus kann dann der Synchronisationsgraph (siehe Abschnitt 2.7.1) aufgebaut werden, dessen Kanten später ausgestaltet werden. Weitere Eigenschaften, die sich direkt aus den Strukturen der Daten bestimmen lassen, werden ebenfalls in diesem Hauptschritt bestimmt.

Nach der Gesamtanalyse folgt der Hauptschritt der **Zerlegung**. Durch Nutzung von Wissen und möglichen manuellen Eingriffen werden die Zeitdatenreihen auf Bereiche untersucht, welche wahrscheinlich gleichzeitig stattgefunden haben. Solche Bereiche und Elemente werden als »Fixpunkte« bezeichnet. Nachdem diese Fixpunkte identifiziert wurden, werden die Zeitdatenreihen entsprechend in Subsequenzen zerlegt. Für diese Subsequenzen werden abschließend die Eigenschaften bestimmt. Im Falle von nicht zerlegbaren Zeitdatenreihen oder manueller Entscheidung gegen die Zerlegung kann dieser Schritt theoretisch übersprungen werden (Pfeil »Direkte Anwendung«).

Darauf folgt der Hauptschritt der **Entscheidung**. Die vorgestellten Methoden der Offline Synchronisierung haben Eigenschaften, die ihre Anwendbarkeit, Qualität und Skalierbarkeit beschreiben. Mit diesen Eigenschaften und den Eigenschaften der Subsequenzen werden die Optionen der Synchronisierung bestimmt. Diese Optionen werden dann in den Synchronisierungsgraph aufgenommen (Kanten). Nachdem alle Kanten so bestimmt wurden, kann das Entscheidungsunterstützungssystem eine Empfehlung ableiten, welche Synchronisierungsschritte auf den Daten vorgenommen werden sollen. Dabei greifen die Komponenten auf Informationen und Wissen zur Produktion zurück.

Der letzte Hauptschritt ist die **Durchführung** der Synchronisierung. Basierend auf der Empfehlung des vorherigen Hauptschritts oder der manuell angepassten Strategie werden für die einzelnen Subsequenzen die gewählten Synchronisierungen durchgeführt. Anschließend werden abschließende Arbeiten durchgeführt um die Subsequenzen entlang der Fixpunkte wieder zu vollständigen Zeitdatenreihen zu aggregieren. Begleitend wird durch Nutzung von Wissen und Informationen, eventuell auch manuell, die Plausibilität der Synchronisierung überprüft und eine Unsicherheitsbetrachtung durchgeführt.

Als Ausgang des Synchronisierungssystems stehen die synchronisierten Zeitdatenreihen zusammen mit der Unsicherheitsbetrachtung zur Verfügung. Damit folgt auf diese Synchronisierung die übliche Vorverarbeitungs-Pipeline ab der Datenbereinigung, wie in Abbildung 2.17 dargestellt. Die Einzelheiten und Schritte innerhalb der Hauptschritte sind im Abschnitt 4.3 beschrieben.

4.2 Eingangsgrößen

Input parameters

Zentrale Eingangsgrößen für das Synchronisierungssystem umfassen Daten, Informationen und Wissen gemäß der Definition in Abschnitt 2.2.2. Ebenfalls sind manuelle Eingriffe möglich, sollen als Ziel dieser Arbeit allerdings möglichst reduziert werden, da manuelle Eingriffe einer Automatisierung und damit verbundenen Skalierbarkeit entgegenstehen.

Die hier dargestellten Eingangsgrößen sind relevante Größen für die Schritte des Synchronisierungssystems. In der Praxis kann sich die Verfügbarkeit dieser Größen unterscheiden. Für die Eingangsgröße 3, die Daten, wurde bereits in den vorangegangenen Kapiteln die Formalisierung und die Beschreibung als Zeitdatenreihen genauer definiert und dargestellt. In diesem Abschnitt sind Eingangsgrößen, welche eine Herleitung und Bestimmung bedürfen, genauer beschrieben.

Einige Schritte und Methoden benötigen zudem eine Parametrierung, beispielsweise Grenzwerte, diese sind in diesem Abschnitt nicht explizit beschrieben, sondern werden als Information angesehen, welche bei der detaillierten Beschreibung des Systems erwähnt werden.

4.2.1 Produktionsaufbau und Unsicherheitskette

Production Setup and Uncertainty Chain

Eine zentrale Information für das Synchronisierungssystem ist die Information über den Aufbau des Produktionssystems und den Ablauf der Produktion. Wie in Abschnitt 2.4 beschrieben, besteht eine Produktion aus verschiedenen Systemen und Komponenten. So besteht die Information zum Produktionsaufbau aus der Menge aller Systeme C_n im Produktionsnetzwerk. Für diese Systeme gibt es zusätzlich die Angabe, welche Netzwerkverbindungen diese Systeme haben, also entlang welcher Links die Daten gesendet werden. Dies entspricht den Angaben $C_n \succ {}_l Z$, respektive $\dots \succ C_n \succ \dots \succ {}_l Z$.

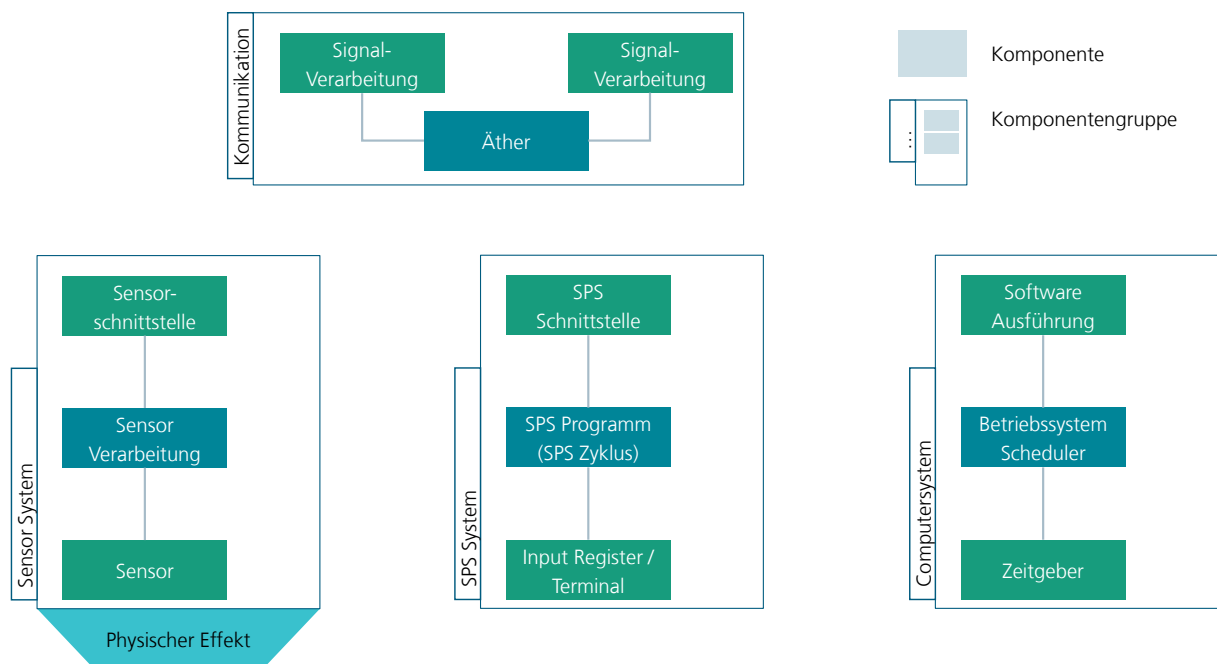


Abbildung 4.2: Darstellung von Elementen des Produktionsnetzwerks als Komponenten und Komponentengruppen. (Basierend auf [SCHM23c])

Da jede Verarbeitung und Übertragung eine Verzögerung verursacht, kann das gesamte Produktionsnetzwerk auch tiefergehend betrachtet werden und dafür in Komponenten und Komponentengruppen zerlegt werden, dargestellt in Abbildung 4.2. So lässt sich ein

Sensorsystem (unten links) zerlegen in den technischen Sensor, welcher den physischen Effekt misst, eine Sensorverarbeitung und die ausgehende Sensorschnittstelle. Dies gilt analog für Systeme wie ein SPS System mit dem innenliegenden SPS Zyklus, allerdings auch für Netzwerkübertragungen, welche ein- und ausgehende Signalverarbeitung sowie die Zeit auf dem Äther beschreiben. Da jede derartige Komponente eine Verzögerung und möglicherweise auch Jitter in die Zeitwerte einbringt, ist die Modellierung und Abschätzung der entsprechenden Zeiten und Jitter hilfreich, soweit diese vollzogen werden können.

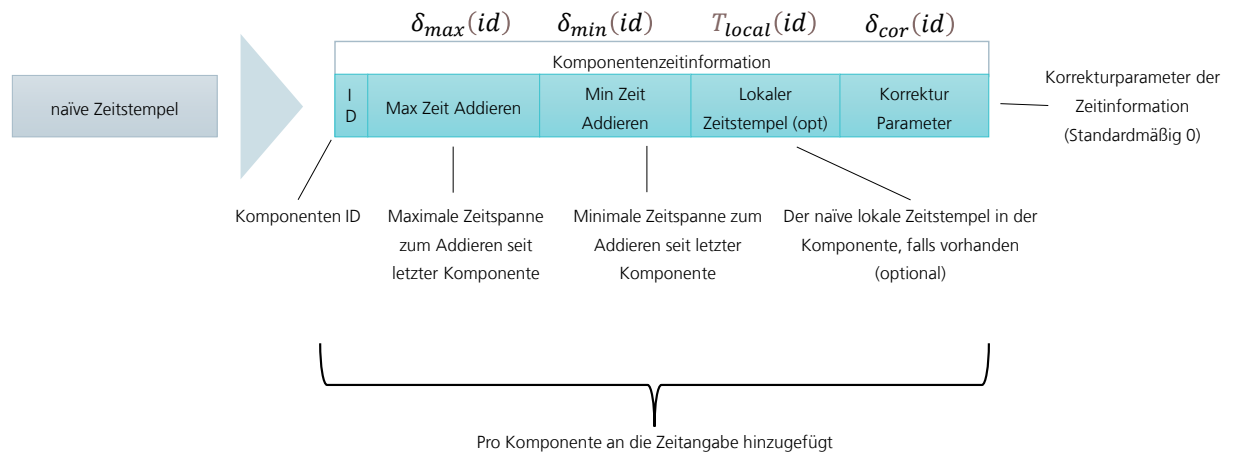


Abbildung 4.3: Anreicherung der Zeitstempeldaten mit den Komponentenzeitinformationen basierend auf der Komponentenzерlegung des Produktionsnetzwerks. (Basierend auf [SCHM23c])

Aus dieser Zerlegung des Produktionsnetzwerks ergibt sich die Möglichkeit der Anreicherung von den Zeitwerten (Zeitstempel) mit den Informationen aus den Komponenten. Dies ist dargestellt in Abbildung 4.3. Der naïve Zeitstempel eines Datums wird hierbei angereichert pro Komponente. Jede Komponente erhält eine eindeutige ID (id), welche als erstes Feld beschrieben ist. Basierend auf den Informationen der Verzögerungen und Jitter zwischen der vorherigen Komponenten und dieser Komponente werden die maximalen ($\delta_{max}(id)$) und minimalen ($\delta_{min}(id)$) Zeiten zum Addieren (Zeitspannen) aufgenommen. Wenn die durchlaufende Komponente selbst auch Zeitwerte generiert ($T_{local}(id)$), werden diese in einem optionalen Feld aufgenommen. Zuletzt existiert ein Feld für die Korrekturparameter $\delta_{cor}(id)$ der Zeitinformationen. Standardmäßig ist dieses Feld zunächst mit 0 belegt. Sollte im Verlauf der Verarbeitungen oder der Synchronisierungen hier Informationen bekannt werden für Korrekturen, so können diese direkt hier aufgenommen werden.

Diese dargestellte Erweiterung der Zeitstempel wird pro durchlaufener Komponente hinzugefügt. Dadurch wird bei einem Produktionsnetzwerk die eigentliche Zeitinformation

mehrfach angereichert. Diese Anreicherungen stellen damit zugleich die Unsicherheitskette der Datenaufnahme entlang des Produktionsnetzwerks dar. Die durch das Produktionsnetzwerk bedingte Unsicherheit der Zeitstempel entspricht damit

$$TS_{Uncert} = \sum_{\forall id} \delta_{max}(id) - \sum_{\forall id} \delta_{min}(id). \quad (4.1)$$

Diese Formulierung beschreibt dabei die Unsicherheiten, die für die Komponenten bestimmt und genutzt werden. Da die Charakterisierung einzelner Komponenten nicht immer möglich ist, kann dies auch auf Ebene von Komponentengruppen oder nur ausgewählter Komponentengruppen geschehen. Dadurch wird die Genauigkeit möglicherweise eingeschränkt.

Messtechnik und Ortung

Eine weitere Art von Informationen zum Aufbau der Produktion bezieht sich auf die räumliche Dimension und Abfolge von Teilprozessen. Für alle Systeme im Produktionsaufbau, welche Sensorik repräsentieren, wird der Kontext bezüglich der räumlichen Koordinaten gespeichert. Dies bedeutet, dass jede Datenaufnahme einem Ort in der Produktion zugeordnet werden kann. Auch die Produktionsanlagen selbst bekommen diesen Kontext zugeordnet. Ein Produkt, welches sich durch die Produktion und die Messtechnik bewegt beschreibt somit auch räumlich den Ablauf des Produktionsprozesses.

Formal wird hierzu die Menge aller physikalischer Effekte, die gemessen werden, als $\mathbb{E}$ beschrieben. Die Menge der Sensoren ist analog gegeben durch $\mathbb{S}$. Der räumliche Kontext der Sensoren ist gegeben durch P . Der Fakt, dass ein Effekt durch einen spezifischen Sensor an einem Ort gemessen wird, ist daher beschrieben durch

$$s_i \perp E_i \mid s_i \in \mathbb{S}, E_i \in \mathbb{E} \quad (4.2)$$

$$P = \{s_i, p_i\} \mid s_i \in \mathbb{S}, p_i \in \mathbb{R}^3, \quad (4.3)$$

wobei die Darstellung der Ortskoordinaten p_i auch in weniger Dimensionen möglich ist, abhängig vom Aufbau der Produktion. Beispielsweise kann bei einem bahngeführten

linearen Prozess eine eindimensionale Beschreibung entlang des Bahnverlaufs ausreichend sein [SCHM23b].

4.2.2 Charakterisierung Systeme

Characterization of Systems

Mögliche Informationen umfassen auch die Charakterisierung von Systemen in dem Produktionssystem. Dies bedeutet, dass der Effekt der Asynchronizität separat im Betrieb oder separat untersucht wird. Dies dient der Modellierung der Asynchronizität von Systemen.

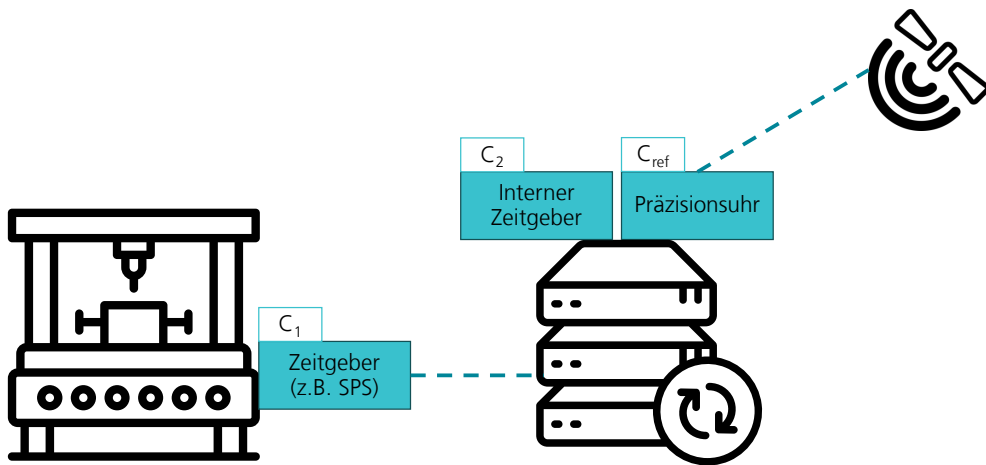


Abbildung 4.4: Schematischer Aufbau der Versuche zur Untersuchung der Asynchronizität. Gestrichelte Linien stellen Datenverbindungen dar.

Der grundsätzliche Aufbau einer solchen Charakterisierung ist in Abbildung 4.4 dargestellt. Eine Produktionsmaschine oder andere Datenquelle C_1 (links) hat eine Verbindung zu einem anderen System wie einem Edge Gerät C_2 (rechts). Eine Präzisionsuhr mit vorhandener Synchronisierung C_{ref} (hier als Satellitenverbindung dargestellt) dient als Referenz. Die Präzisionsuhr kann mit regelmäßiger Synchronisierung eine hohe Genauigkeit halten. Die anderen Zeiten im Versuch sind aus der Datenquelle links (hier SPS) und dem internen Zeitgeber des Systems. Daraus ergibt sich eine Zeitreihe der Form

$$CH_{C_1, C_2} = \{(C_1(t_i), C_2(t_i), C_{ref}(t_i))\}, \quad (4.4)$$

woraus eine direkte Zuordnung zwischen den Zeiten der Systeme C_1 und C_2 für den Betrachtungszeitraum möglich ist, also die Projektionsfunktionen Φ_{C_1, C_2} und Φ_{C_2, C_1} . Findet die angedachte Synchronisierung im Betrachtungszeitraum von CH_{C_1, C_2} statt, so ist diese Information bereits ausreichend.

Für Synchronisierungen außerhalb dieses Betrachtungsbereichs lassen sich aus dieser Zeitreihe Modellierungen ableiten. Das direkte Modell betrifft die Verbindung zwischen den beiden betrachteten Systemen C_1 und C_2 . Dazu lässt sich aus der Menge CH_{C_1, C_2} eine Interpolationsfunktion der Abbildung von C_1 auf C_2 und andersherum erstellen. Hierzu wird eine lineare Interpolation erstellt, welche das Verhalten beschreibt und als mögliche Projektionen Φ_{C_1, C_2} und Φ_{C_2, C_1} dient.

Zusätzlich zu den direkten Modellen können auch Interpolationsmodelle der Systeme auf die Referenzzeit und andersherum analog erstellt werden. Wird eine Charakterisierung von mehreren Systemen oder Systempaaren durchgeführt, kann so direkt eine Komposition der Projektionen definiert werden.

4.2.3 Wirkungen und Regeln

Cause-effect Relationships and Rules

Die Ursache-Wirkzusammenhänge sind zentrales Wissen für das Synchronisierungssystem und üblicherweise einer der Treiber für manuelle Arbeiten, da dieses auf Expert*innen zurückgeht, welche aktiv eingebunden werden. Im Bereich der Batteriezellfertigung ist zudem dieses Wissen weniger vorhanden als in anderen Domänen [FITZ23]. Für dieses Synchronisierungssystem sollen die notwendigen Eingriffe durch Menschen minimiert werden. Daher wird das notwendige Wissen in einer Wirkdatenbank formalisiert.

Die Formalisierung von Expert*innenwissen ist ein aktives Forschungsfeld bei dem technische und nicht-technische Hürden existieren [BIGL22], dessen Details allerdings außerhalb dieser Arbeit liegen. Im Vergleich mit komplexen Ansätzen aus dem Bereich der temporalen Logik [BELL00] und spezialisierter Ontologien [LI20] sind die Anforderungen und notwendige Komplexität durch eine vereinfachte Variante der Darstellung erfüllbar.

Für die Modellierung der Ursache-Wirkzusammenhänge werden Ursachen und Wirkungen als Funktionen über die Zeit in Relation gebracht. Dies bedeutet, dass ein beobachtetes

Verhalten durch $f(E_n, t)$ beschrieben ist. Für eine Zeitspanne gibt f den signifikanten Verlauf des Sensorsignals der Messung von Effekt E_n an. Dies kann beispielsweise entlang der Merkmalsdefinition nach *HOG1D* geschehen. Dem gegenüber steht eine Auswirkung, die analog beschrieben wird durch $g(E_m, t)$. Diese Verbindung von f und g , beschrieben als $\rightsquigarrow$, ist hier aufgrund desselben Zeitrahmens in enger zeitlicher Abfolge. Um die räumliche Distanz einzubeziehen, kann die Definition angepasst werden zu

$$f(E_m, t) \rightsquigarrow g(E_n, t + \overset{\circ}{d}_{s_m, s_n}) \mid s_m \perp E_m, s_n \perp E_n \quad (4.5)$$

$$\Gamma = \{f \rightsquigarrow g\}, \quad (4.6)$$

sodass Γ die Menge aller bekannter Ursache-Wirkzusammenhänge ist. Die Funktion $\overset{\circ}{d}_{s_m, s_n}$ hingegen ist die zeitliche Distanz für die Propagation des messbaren physikalischen Effekts vom Ort des Sensors s_m zu s_n . Damit beschreiben die Elemente in Γ ein detektierbares Merkmal in den Daten, welches nach einem erwarteten Zeitraum an einem anderen Sensor zu einem anderen detektierbaren Merkmal in den Daten führt [SCHM23b]. Γ wird daher als die Wirkdatenbank bezeichnet und repräsentiert das Wissen $\mathfrak{W}$ als Eingangsgröße.

Das Befüllen von Γ kann im Vorfeld einer Synchronisierung durch Expert*innen oder der Nutzung von Literatur zu Ursache-Wirkzusammenhängen geschehen. Zusätzlich kann hier eine Vorverarbeitung in Bezug auf die Normalisierung von Messgrößen und Skalierungen notwendig sein. Diese Wirkdatenbank kann auch für mehrere Offline Synchronisierungen wiederverwendet werden, sodass der manuelle Aufwand der Erstellung einmalig auftritt, beziehungsweise dann nur noch für Ergänzungen mit geringem Aufwand angepasst wird.

Neben dieser Wirkdatenbank besteht auch das Wissen um spezifische Domänenregeln $\mathbb{P}$ (2.49), welche sich nicht aus den Informationen direkt ergeben. Hierfür ist keine spezifische Formalisierung notwendig, stattdessen wird $\mathbb{P}$ als eine Menge von Gleichungen und Relationen bezüglich der Zeitdatenreihen behandelt, die durch Expert*innen erstellt wird. Die Darstellung als Gleichungen und Relationen erlaubt eine direkte Prüfung der Plausibilität am Ende der Durchführung der Synchronisierung.

4.2.4 Manuelle Eingriffsmöglichkeiten

Manual Intervention Possibilities

Aus der Anforderung zur Skalierbarkeit des Offline Synchronisierungssystems ergibt sich ein Bedarf an Automatisierung und der Vermeidung von manuellen Eingriffen durch Menschen. Daher ist das manuelle Eingreifen per Definition in diesem System unerwünscht.

Dennoch gibt es Stellen, an denen explizit manuelle Eingriffsmöglichkeiten vorgesehen werden, um eine optionale Kontrollmöglichkeit und Korrektur zu ermöglichen. Der erste vorgesehene Eingriffspunkt ist vor dem eigentlichen Synchronisierungssystem, bei der Vorbereitung der Eingangsgrößen. Die Aufbereitung der Informationen und die Formalisierung des Wissens der Wirkdatenbank werden mindestens erstmalig einen manuellen Aufwand erfordern. Bei häufigerem Einsetzen dieses Synchronisierungssystems sinkt aber die entsprechende Signifikanz im Sinne des Aufwands.

Im Schritt der Zerlegung wird eine Möglichkeit für Menschen vorgesehen, bei der die Ergebnisse der Zerlegung einerseits überprüft werden können, andererseits auch manuell Zerlegungen und Fixpunkte markiert werden können. Das System arbeitet also grundsätzlich selbstständig, kann aber durch die manuellen Eingaben verbessert werden.

Am Ende der Synchronisierungen finden Plausibilitätsprüfungen statt. Hierzu werden die Gleichungen und Relationen aus $\mathbb{P}$ verwendet, jedoch besteht auch die Möglichkeit, vor Abschluss der Prüfungen eine manuelle Prüfung der Plausibilität durch einen Menschen vornehmen zu lassen. Sollte die Plausibilität nicht gewährleistet sein aus Sicht des Eingreifenden, so hat diese Person die Wahl, welche Schritte wiederholt werden sollen, oder ob diese Einschränkung der Datenqualität hinzunehmen ist. Letzterer Punkt ist in jedem Fall nur von einem Menschen zu bewerten, da dieses Synchronisierungssystem keinerlei Informationen über die spätere Verwendung der Zeitdatenreihen nutzt.

4.2.5 Vorbereitung der Synchronisierungsmethoden

Preparation of Synchronization Methods

Ein Teil des Synchronisierungssystems ist die Sammlung der möglichen Synchronisierungsmethoden M zum Einsatz. Neben den eigentlichen Methoden muss diese Menge noch mit Informationen angereichert werden, um eine automatisierte Auswahl und Bewertung zu ermöglichen.

Die Eigenschaften der Methoden wurden in Tabelle 3.1 beschrieben und dargestellt. Für die Vorbereitung des Offline Synchronisierungssystems muss diese Tabelle entsprechend als

Information zu den Methoden integriert werden. Während die Angaben zu Effizienz und Qualität in dieser Bewertungsform ausreichend sind, bedürfen die Metriken zur Anwendbarkeit einer genaueren Beschreibung. Die In situ Anforderungen werden pro Methode genau beschrieben und mit den Informationen verlinkt. So bedarf die Methode *SoE* der Verfügbarkeit der Informationen zu den Uhren und Zeiten aller Systeme im Netzwerk, was in den Informationen zum Aufbau und den angepassten Zeitstempeln abgebildet ist, wenn diese Anforderungen erfüllt sind. Derartige Verlinkungen der Anforderungen zu den Informationen werden für jede Methode beschrieben und angefügt.

Analog dazu geschieht dies auch bei dem Wissen zur Unabhängigkeit von Messungen. Die Unabhängigkeit von Messungen kann beschrieben werden durch die Angaben zu $\mathbb{S}$ und $\mathbb{E}$. Auch können diese Angaben möglicherweise direkt in der Wirkdatenbank Γ auftauchen. Die Anreicherung, ob zwei Messungen als abhängig oder unabhängig einzustufen sind, ist daher eine zusätzliche Information bei der Vorbereitung der Methoden. Methoden wie *Dynamic Time Warping* können hierbei granularer beschrieben werden, indem nicht nur eine grundsätzliche Abhängigkeit von Messungen, sondern konkrete Ähnlichkeitsmaße und Grenzwerte aufgenommen werden können.

4.3 Schritte des Synchronisierungssystems

Steps of the Synchronization System

Mit den oben beschriebenen Eingangsgrößen können die einzelnen Schritte des Synchronisierungssystems genauer beschrieben werden. Diese Schritte entsprechen allen Elementen innerhalb des grünen Kastens in Abbildung 4.1.

4.3.1 Gesamtanalyse

Base Analysis

Die Gesamtanalyse befasst sich mit der ersten Analyse der Zeitdatenreihen $\mathfrak{Z}$ im Einzelnen und übergreifend. Weiterhin dient diese Gesamtanalyse der Vorbereitung der nächsten Schritte basierend auf den verfügbaren Informationen $\mathfrak{I}$.

Aufbau Synchronisationsgraph

Ein wichtiges Mittel der Synchronisierung ist der Synchronisierungsgraph G_{sync} . In der Gesamtanalyse wird dieser erstmals aufgestellt:

$$\begin{aligned}
 G_{sync} &= (KN, KA, \lambda, \Lambda) & (4.7) \\
 KN &= \{C_n\} \\
 KA &= \{(a, b) \in \{KN \times KN\}\} \\
 \lambda &: KA \rightarrow \mathcal{P}(\{\Phi\} \cup M \cup \{\epsilon\}) \\
 \Lambda &: KN \rightarrow \mathcal{P}(\mathfrak{Z})
 \end{aligned}$$

Die Menge KN beschreibt hier die Menge aller Knoten im Graphen, die Menge KA die Menge der Kanten. Die Kanten erhalten zusätzlich ein Label durch die Funktion λ , welches eine oder mehrere explizite Projektionen Φ oder mögliche Methode zur Erstellung aus der Menge der verfügbaren Synchronisierungsfunktionen M enthält. Ist weder eine Projektion, noch eine Methode zugewiesen ist der leere Wert ϵ zugewiesen. Die Darstellung als Menge hat ihren Ursprung in der folgenden Zerlegung, bei der möglicherweise mehrere Subsequenzen statt gesamter Zeitdatenreihen betrachtet werden. Die Funktion Λ weist den Knoten Label zu, welche jeweils eine Menge von Zeitdatenreihen umfassen.

Im ersten Schritt wird die Knotenmenge bestimmt. Für diese wird auf die Informationen des Produktionsaufbaus zurückgegriffen. Für jedes dort bekannte System C_n wird ein entsprechender Knoten erzeugt: $KN = \{C_n \mid C_n \in \mathfrak{I}\}$. Ergänzend wird die Menge $KN_{sink} \subseteq KN$ definiert und beschreibt die Zielsysteme (Senken) der Datenpipelines, also beispielsweise die Datenbanksysteme im Produktionsnetzwerk. Die Knotenlabel werden aus den Quellen der Zeitdatenreihen bestimmt. Jeder Knoten bekommt daher die Menge aller Zeitdatenreihen zugewiesen, welche Zeitangaben aus diesem System haben (siehe 2.61 und 2.65). Daraus ergibt sich $\Lambda(C_n) = \{{}_lZ \in \mathfrak{Z} \mid C_n \succ {}_lZ\}$. Hat ein System keine Zeitdatenreihen mit Zeitangaben versehen, so ist das Label entsprechend die leere Menge $\emptyset$. Die Kanten KA von G_{sync} werden im ersten Schritt für alle Kombinationen von Knoten (vollständiger Graph) erstellt. Die Labels λ sind zunächst für alle Kanten auf den Wert ϵ gesetzt.

Nachdem der Synchronisationsgraph erstellt wurde, kann die erste Analyse und Anpassung durchgeführt werden basierend auf den Informationen zum Produktionsaufbau. Knoten, welche als Label lediglich $\emptyset$ haben, bringen einerseits Verzögerungen in der Übertragung ein, aber selbst keine Daten zu Zeiten. Daher können die Kanten, die diesen Knoten betreffen, unterschieden werden. Dazu werden die Netzwerkpfade $\dots \succ C_n \succ \dots \succ {}_lZ$ als Pfade analog durch G_{sync} interpretiert. Daraus ergibt sich

$$KN_{\emptyset} = \{C_n \mid \Lambda(C_n) = \emptyset\} \quad (4.8)$$

$$KA^{C_n} = \{e = (a, b) \in KA \mid a = C_n \vee b = C_n\} \quad (4.9)$$

$$KA_{\emptyset}^{C_n} = \{e \in KA \mid \exists_l Z : \dots \succ C_n \succ \dots \succ_l Z\} \quad (4.10)$$

$$KA_{none}^{C_n} = KA^{C_n} \setminus KA_{\emptyset}^{C_n}, \quad (4.11)$$

wobei die Menge $KA_{\emptyset}^{C_n}$ alle Kanten beschreibt, welche eine Zeitdatenreihe betrifft, die im Produktionsnetzwerk durch das System C_n geht, während analog dazu $KA_{none}^{C_n}$ alle Kanten beschreibt, bei denen C_n keine Relevanz für die Zeitdatenreihen aufweist. Als Konsequenz ergibt sich damit

$$KA = KA \setminus \{KA_{none}^c \mid c \in KN_{\emptyset}^{C_n}\} \quad (4.12)$$

$$\lambda((C_n, C_m)) = {}^D\Phi_{n,m} \mid (C_n, C_m) \in KA_{\emptyset}^c \quad (4.13)$$

$$\lambda((C_m, C_n)) = {}^D\Phi_{m,n} \mid (C_m, C_n) \in KA_{\emptyset}^c \quad (4.14)$$

$${}^D\Phi_{n,m} : t_i \mapsto C_n(t_i) + D_{Net}, \quad (4.15)$$

wobei D_{Net} eine feste Verzögerung zwischen zwei Systemen, induziert durch das Netzwerk und die Weiterleitung beschreibt, wie dargestellt in 2.66. Dies bedeutet, dass unnötige Kanten aus dem Synchronisierungsgraph entfernt wurden und Verzögerungen im Netzwerk aus den Informationen zum Netzwerk und des dortigen Verzögerungsmodells in $\mathfrak{T}$ abgeleitet wurden.

Der so erstellte Synchronisierungsgraph wird in den späteren Hauptschritten weiter angepasst und genutzt.

Syntaktische Vorverarbeitung

Wie in Abschnitt 2.6 dargestellt, leiden Daten potenziell unter verschiedenen Problemen im Bereich der Datenqualität. Gemäß der Vorverarbeitungsschritte nach FRYE werden die meisten dieser Probleme erst nach der Integration im Detail betrachtet, zumal auch die Asynchronizität Einfluss auf diese Probleme haben kann. Da das Offline Synchronisierungssystem selbst auf Daten angewiesen ist, müssen Schritte der Datenvorverarbeitung teilweise vorab durchgeführt werden. Dabei sollen die Vorverarbeitungsschritte im Gesamtablauf

allerdings nicht mehrfach durchgeführt werden und keine gegenseitigen Abhängigkeiten schaffen. Somit ist hier eine Eingrenzung dieser Schritte notwendig. Für die Vorverarbeitung werden daher in diesem Schritt lediglich die Zeitwerte betrachtet und auf wenige Datenprobleme hin untersucht und vorverarbeitet.

Im Rahmen der **Syntaxverletzung** werden die Zeitwerte untersucht und, wo möglich, eine syntaktische Konvertierung vorgenommen, sodass alle Zeitwerte einer korrekten Darstellung wie in Abschnitt 2.2.3 beschrieben folgen.

Weiterhin werden die Zeitwerte auf das Problem der **Uneindeutigkeit und fehlendem Kontext** hin untersucht. Sind die Zeitwerte nicht eindeutig, so wird über die Informationen zu dem Produktionsaufbau nach plausiblen Interpretationen gesucht. Ist keine solche Interpretation verfügbar, oder der Kontext (z.B. aufnehmende Systeme) nicht verfügbar, so werden die entsprechenden Zeitwerte oder gar komplette Zeitdatenreihe als defekt und somit nicht synchronisierbar markiert. Sie werden im weiteren Verlauf des Synchronisierungssystems nicht weiter betrachtet und innerhalb des Systems aus $\mathfrak{Z}$ entfernt.

Der Effekt von **unplausiblen und inadäquaten** Zeitwerten kann mit den Informationen zum Aufbau des Produktionssystems und den Informationen zum groben betrachteten Zeitraum untersucht werden. So sind beispielsweise Sprünge in den Zeitwerten, bei denen ein einzelner Wert deutlich über den Betrachtungsraum hinaus und danach zurückspringt unplausibel und sollte entfernt werden.

Die weiteren Probleme der Datenqualität zielen auf die Datenwerte ab, sind für Zeitwerte nicht zutreffend oder bereits in der Formalisierung berücksichtigt (z.B. fehlende Werte durch die Einführung von Z^e). Durch diese Auswahl der Vorverarbeitungsschritte ist eine doppelte Ausführung der Vorverarbeitung und gegenseitiger Abhängigkeiten ausgeschlossen, bei gleichzeitiger Sicherstellung der Qualität von Zeitwerten für die folgenden Schritte.

Ableitung Eigenschaften Zeitreihen

In diesem Schritt werden Eigenschaften abgeleitet, die sich aus dem Aufbau und den Daten selbst ergeben. Dabei werden die Eigenschaften der Zeitdatenreihen und des Gesamtsystems betrachtet.

Für die einzelnen Zeitdatenreihen werden zunächst die jeweiligen Minima und Maxima (siehe 2.23ff.) bestimmt, inklusive der Minima und Maxima der Zeitwerte ($\arg \min_t$ und $\arg \max_t$). Mit diesen Angaben und der Menge von Zeitdatenreihen werden weiterhin

die jeweiligen Frequenzen f_Z (2.21) bestimmt. In Bezug auf das Verhalten der Zeitwerte werden dann zusätzlich die Wölbung $\eta(Z)$ (2.17), Schiefe $\gamma(Z)$ (2.16), Variation Var_Z (2.19), Mittelwert $\bar{t}$ (2.18) und Standardabweichung σ (2.20) bestimmt.

Weiterhin wird für jede Zeitdatenreihe eine Monotonie-Überprüfung durchgeführt. Dabei wird geprüft, ob die Zeitwerte monoton oder streng monoton ansteigen. Sollte es Monotonie-Verletzungen geben, werden die entsprechenden Stellen notiert, sodass implizite monotone Subsequenzen definiert werden können.

Auf das Gesamtsystem bezogen wird zunächst der zeitliche Betrachtungsrahmen identifiziert. Dieser ergibt sich über alle Zeitdatenreihen hinweg aus dem niedrigsten und dem höchsten auftretenden Zeitwert:

$$\arg \min_t \mathfrak{Z} = \arg \min_t \{ {}_tZ \mid Z \in \mathfrak{Z} \} \quad (4.16)$$

$$\arg \max_t \mathfrak{Z} = \arg \max_t \{ {}_tZ \mid {}_tZ \in \mathfrak{Z} \} \quad (4.17)$$

Weiterhin wird für die oben beschriebenen Eigenschaften der einzelnen Zeitdatenreihen jeweils ein Histogramm erstellt, um die Verteilung dieser Eigenschaften zu beschreiben.

Am Ende des Hauptschritts der Gesamtanalyse sind damit syntaktisch vorverarbeitete Zeitdatenreihen, der Synchronisierungsgraph, sowie die Menge aller Eigenschaften der Zeitdatenreihen und des Gesamtsystems verfügbar.

4.3.2 Zerlegung

Slicing

Der Hauptschritt der Zerlegung orientiert sich an dem Konzept der kreuzmodalen Bindung, dargestellt in Abschnitt 2.10. Da das Gehirn hierzu sowohl Daten, Informationen als auch Wissen nutzt, werden diese Eingangsgrößen ebenfalls hier bei der Zerlegung benötigt. Auch manuelle Eingriffe sind in diesem Hauptschritt möglich.

Zur Veranschaulichung ist in Abbildung 4.5 eine exemplarische Darstellung von Zeitdatenreihen, aufgetragen auf einer gemeinsamen Zeitskala, gezeigt. Diese Zeitskala ist nur syntaktisch gleich (gleiche Indizes), sodass die Zeitdatenreihen selbst asynchron zueinander sind. Für diesen exemplarischen Fall seien zudem auch Informationen zum Aufbau und Wissen in Form der Wirkdatenbank gegeben.

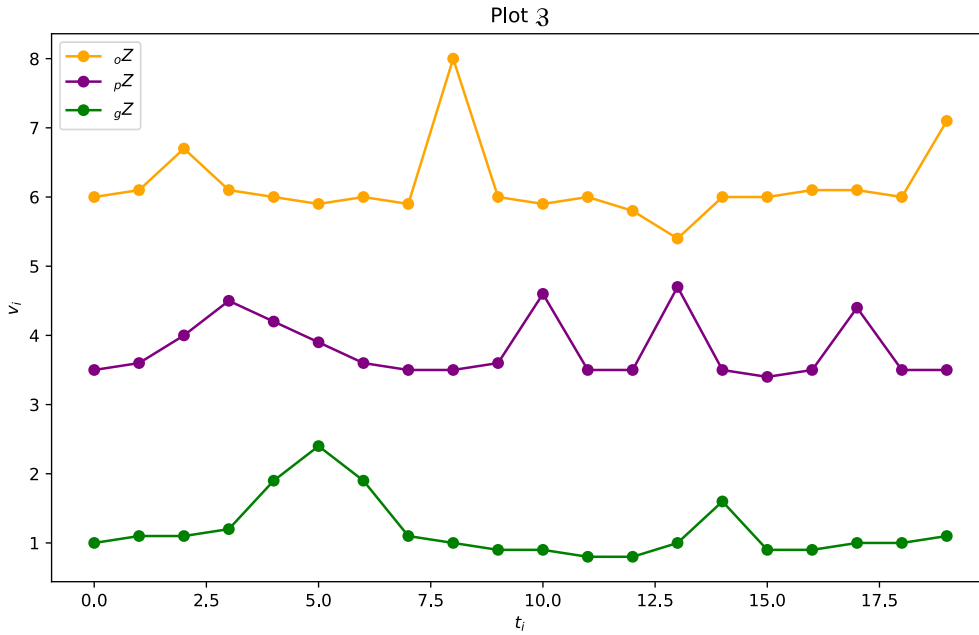


Abbildung 4.5: Exemplarische Darstellung von asynchronen Zeitdatenreihen, aufgetragen auf gemeinsamer Zeitskala.

Identifikation Fixpunkte

Aus den Informationen zum Aufbau der Produktion und dem Wissen über Zusammenhänge lassen sich Punkte mehrerer Zeitdatenreihen identifizieren, die wahrscheinlich gleichzeitig sind. Mit einer Definition von »wahrscheinlich« ($Pr()$) als mindestens 95 % ergibt sich die Menge aller identifizierten Fixpunkte Ξ aus

$$Pr(z_n^l \doteq z_m^k) > 0.95 \mid z_n^l \in {}_lZ, z_m^k \in {}_lZ^k \quad (4.18)$$

$$\Xi = \{\xi_i = (z_n^l, z_m^k, \dots) \mid Pr(z_n^l \doteq z_m^k \doteq \dots) > 0.95\}, \quad (4.19)$$

wobei die transitiven Eigenschaften von $\doteq$ genutzt werden um Fixpunkte beliebiger Dimension, also bezüglich beliebig vieler Zeitdatenreihen, zu beschreiben. Damit ist Ξ die Menge aller identifizierten Fixpunkte ξ_i in den Daten für die betrachtete Domäne.

Für eine Beschränkung auf Fixpunkte, welche sicher als solche identifiziert oder bestimmt wurden, wird die Menge Ξ_1 genutzt. Solche eindeutigen Fixpunkte können durch Wissen

und Informationen abgeleitet sein, oder aber durch manuelle Eingriffe eines Menschen explizit vorgegeben werden. Solange keine externen Eingriffe solche Fixpunkte vorgeben, gilt daher $\Xi_1 \subseteq \Xi$.

Die Ungenauigkeit eines Fixpunktes lässt sich durch die erwartete Ungenauigkeit der Gleichzeitigkeit $\triangleq$ definieren. Damit kann die Ungenauigkeit eines Fixpunkts ξ_i beschrieben werden durch

$$\Delta(\xi_i) = (0, \tau_{Error}^{z_m^k}, \dots), \quad (4.20)$$

wobei der erste Eintrag stets 0 ist, da die Ungenauigkeit das erste Tupelelement als Referenz nutzt, welches damit per Definition genau ist.

Die Identifikation der Fixpunkte kann auf drei verschiedene Weisen geschehen. Die erste Variante ist die manuelle (externe) Vorgabe von Fixpunkten durch einen Menschen. Ein solcher Fixpunkt ξ_{manuell} ist per Definition ein Element von Ξ_1 und wird als genau bewertet, sodass $\Delta(\xi_{\text{manuell}}) = (0, 0, \dots)$.

Die zweite Variante ist die direkte Zuordnung von Zeitreihenelementen durch Nutzung von Informationen. Dies sind beispielsweise im Prozess bestehende Synchronisationspunkte oder besondere punktuelle Ereignisse, wie Start, Ende, Unterbrechung oder ein Referenzsignal. Die Zuordnung an sich ist damit sicher, allerdings hängt die Ungenauigkeit von einem solchen Fixpunkt ξ_{info} von der Detektion und Aufnahmeart ab. Das entsprechende Ereignis E_i wird durch einen Sensor s_i detektiert. Der Sensor selbst, beziehungsweise das messende System, birgt selbst Verzögerungen durch Zykluszeiten oder interne Verarbeitungen. Aufgrund der Systeme bestimmt sich die Ungenauigkeit mit 2.57 in diesem Fall daher zu

$$\begin{aligned} \xi_{\text{info}} &= (z_n^{s_l}, z_m^{s_k}, \dots) \mid C_{s_l} \succ_{s_l} Z, C_{s_k} \succ_{s_k} Z \\ \implies \Delta(\xi_{\text{info}}) &= (0, \tau_{Error}^{l,k}, \dots). \end{aligned} \quad (4.21)$$

Die dritte Variante ist die Zuordnung über die Nutzung von Wissen und den innenliegenden Datenmerkmalen. Die Elemente aus Γ beschreiben Verhalten von Datenmerkmalen in mehreren Zeitdatenreihen. Auf der linken Seite der Wirkregel $w_i \in \Gamma$ (4.5) wird das Verhalten in einer Zeitdatenreihe $_{s_n}Z, s_n \perp E_n$ beschrieben. Dies bedeutet, dass für die

Identifizierung von Fixpunkten die entsprechende Zeitdatenreihe auf solche Merkmale untersucht werden muss. Da es nicht um ein einzelnes Element geht bei diesem Merkmal, wird also eine Subsequenz $\overset{\circ}{U}_{s_n}Z$ gesucht, welche dieses Verhalten aufweist. Analog wird für die rechte Seite von w_i die Suche nach Subsequenzen $\overset{\circ}{U}_{s_m}Z$ durchgeführt. Die Länge der Funktionen in w_i sowie der Versatz $\overset{\circ}{d}_{s_m, s_n}$ erlauben dabei die Bewertung, wie wahrscheinlich diese Merkmale auf die Wirkregel w_i zurückgehen. Ist die Wahrscheinlichkeit ausreichend hoch zu bewerten, können diese beiden Subsequenzen in Relation gesetzt werden. Für die Bestimmung der Fixpunkte ergeben sich hierbei verschiedene Strategien. So können beispielsweise die Anfangspunkte oder bestimmte Repräsentanten aus den Subsequenzen als Fixpunkte im Sinne eines $\xi_{\text{wirk}} \in \Xi$ bestimmt werden.

Die Ungenauigkeit eines ξ_{wirk} ist hier abhängig von mehreren Faktoren. Auch hier trifft die Unsicherheitsbetrachtung analog zu ξ_{info} zu. Weiterhin ist das Verfahren der Suche der Subsequenzen und Wahl der Elemente als Fixpunkte relevant für die Ungenauigkeit. Je kürzer die Funktionen in w_i sind, desto geringer ist diese Ungenauigkeit. Zuletzt bleibt die Wahrscheinlichkeit eines fehlerhaften Fixpunkts bestehen, sodass die potenzielle Ungenauigkeit über den Betrachtungsrahmen hinaus geht. Daher kann keine allgemeine Ungenauigkeitsgrenze abgeleitet werden.

Bezogen auf die exemplarische Darstellung von oben lassen sich die Fixpunkte wie in Abbildung 4.6 darstellen. In diesem Fall konnten durch die Fixpunktidentifikation drei Paare w_i von Fixpunkten gefunden werden. So beschreibt der Fixpunkt w_1 hier die Eigenschaft $z_3^p \doteq z_5^g$. Es wurden Fixpunkte zwischen ${}_gZ$ und ${}_pZ$ sowie zwischen ${}_oZ$ und ${}_pZ$ gefunden, allerdings kein Fixpunkt für das andere Zeitdatenreihen-Paar.

Zerlegung in Subsequenzen

Nach der Identifikation von Fixpunkten Ξ wird für die Menge der Zeitdatenreihen eine Zerlegung durchgeführt. Dies bedeutet, dass jede Zeitdatenreihe in Subsequenzen zerlegt wird, welche jeweils mit einem Fixpunkt starten und enden. Ausnahmen bilden hierbei die erste und die letzte Subsequenz, deren Start, respektive Ende, dem ersten oder letztem Element der gesamten Zeitdatenreihe entsprechen. Diese Subsequenzen lassen sich daher für eine Zeitdatenreihe ${}_lZ$ definieren als

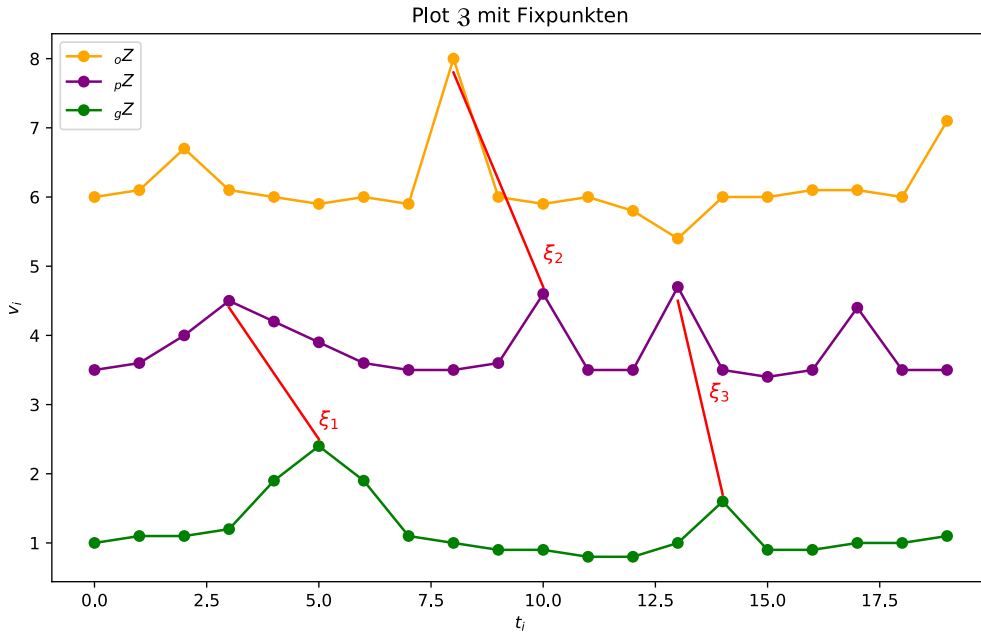


Abbildung 4.6: Die exemplarische Darstellung von asynchronen Zeitdatenreihen mit identifizierten Fixpunkten in rot, inklusive der gegenseitigen Bezugspunkte.

$${}_i\mathring{U}(\Xi) = \{{}_i\mathring{U}^i \mid z_i^l \in \Xi\} \quad (4.22)$$

$${}_i\mathring{U}^i = \{z_k^l \mid \exists z_i^l, z_j^l \in \Xi' : i \leq k \leq j \wedge z_k^l \notin \Xi'\} \quad (4.23)$$

$$\Xi' = \Xi \cup \{\min_t {}_iZ\} \cup \{\max_t {}_iZ\}, \quad (4.24)$$

wodurch die Zerlegung ${}_i\mathring{U}(\Xi)$ entweder mehrere Subsequenzen darstellt oder genau die Zeitdatenreihe ${}_iZ$ selbst.

Die Zerlegung ${}_i\mathring{U}(\Xi)$ ist dabei die maximal mögliche Zerlegung für die jeweilige Zeitdatenreihe ${}_iZ$, da alle Fixpunkte dieser Zeitdatenreihe zu allen anderen Zeitdatenreihen berücksichtigt werden. Für die Synchronisierung von zwei Zeitdatenreihen ${}_iZ, {}_mZ$ sollen daher nur die Subsequenzen betrachtet werden, die sich durch Berücksichtigung der Fixpunkte eben jener Zeitdatenreihen ${}_{l,m}\Xi$ ergeben. Diese Zerlegung ist beschrieben als ${}_{l,m}\mathring{U}({}_{l,m}\Xi)$, mit der Kurzform ${}_{l,m}\mathring{U}$.

Während die Kantenlabel in G_{sync} die beiden Seiten einer Kante als Systeme zum Synchronisieren interpretieren, gehen die Zerlegungen auf die Zeitdatenreihen auf beiden Seiten einer Kante ein. Daher wird die Definition 4.7 um Kantenlabel der Zerlegung $\overset{\circ}{\lambda}$ erweitert:

$$\overset{\circ}{\lambda} : KA \rightarrow \{_{l,m}\overset{\circ}{U} \times (M \cup \{\epsilon\})\} \quad (4.25)$$

Daher wird nach der Zerlegung $\overset{\circ}{\lambda}$ entsprechend erstellt, und für jede Zerlegung ein Element $(_{l,m}\overset{\circ}{U}, \epsilon)$ hinzugefügt.

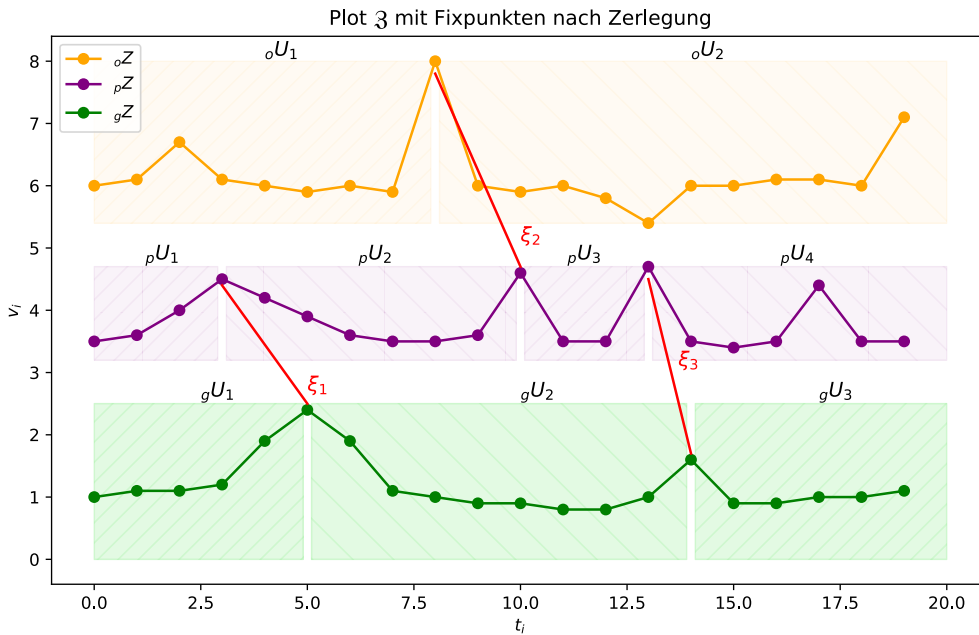


Abbildung 4.7: Die exemplarische Darstellung von asynchronen Zeitdatenreihen mit identifizierten Fixpunkten in rot und den abgeleiteten Subsequenzen U (schraffiert) als Zerlegung.

Bezogen auf die exemplarische Darstellung von oben lassen sich die Zerlegungen wie in Abbildung 4.7 darstellen. So wird die Zeitdatenreihe oZ in die zwei Subsequenzen oU_1 und oU_2 zerlegt. Das Element z_8^o ist Teil des Fixpunktes w_2 und daher in beiden Subsequenzen enthalten nach der Zerlegung. Die Zerlegung der anderen Zeitdatenreihen geschieht analog. Zu erkennen ist hier die maximale Zerlegung von pZ in vier Subsequenzen. Da die gemeinsamen Fixpunkte mit den jeweils anderen Zeitdatenreihen allerdings nicht

dieselben Elemente betrifft, werden bei der Synchronisierung verschiedene Subsequenzen genutzt. Das Gegenstück zu Subsequenz ${}_gU_1$ ist ${}_pU_1$, während das Gegenstück zu ${}_gU_2$ die Vereinigung ${}_pU_2 \cup {}_pU_3$ ist. Diese Angabe, welche Subsequenzen welche Gegenstücke haben, wird in oben beschriebener Art in die Kantenlabel λ hinterlegt.

Ableitung Eigenschaften Subsequenzen

Der letzte Schritt der Zerlegung betrifft die Ableitung von Eigenschaften der Subsequenzen. Dabei werden dieselben Methoden und Metriken eingesetzt, wie eingangs für die Gesamtanalyse in Abschnitt 4.3.1. Dies erlaubt eine genauere Betrachtung der einzelnen Subsequenzen, die bei der Bewertung von Methoden aufgegriffen wird. Zusätzlich zur Bestimmung der Eigenschaften für die einzelnen Subsequenzen werden die Ungenauigkeiten und Distanzen der Fixpunkte aufgenommen. Diese Angaben sind insbesondere wichtig für die Methodenauswahl, genauso wie für die Plausibilitätsprüfung. Beispielsweise ist eine Abweichung weniger plausibel, wenn die Fixpunkte durchweg deutlich geringere Distanzen als eine Distanz aus der Anwendung einer Synchronisierungsmethode aufweisen.

Am Ende des Hauptschritts der Zerlegung wurden damit die Menge der Fixpunkte identifiziert, bewertet und für eine Zerlegung der Zeitdatenreihen genutzt. Basierend darauf wurde der Synchronisationsgraph verfeinert. Auch wurden weitere Eigenschaften der Zeitdatenreihen und allgemeine Informationen zur Asynchronizität für die Nutzung in der Plausibilitätsprüfung aufgestellt.

4.3.3 Entscheidung

Decision

Der Hauptschritt der Entscheidung befasst sich mit der Erstellung einer Auswahl möglicher Synchronisationsmethoden für die Zeitdatenreihen oder Subsequenzen, falls eine Zerlegung stattgefunden hat. Am Ende des Hauptschritts steht dann eine Empfehlung, welche Synchronisierungen durchzuführen sind, um die Zeitdatenreihen zu synchronisieren.

Bewertung Methoden

Im ersten Schritt innerhalb des Hauptschritts der Entscheidung werden Synchronisationsmöglichkeiten betrachtet. Die Möglichkeiten entsprechen hierbei allen existierenden Kanten im Synchronisationsgraph G_{syncr} , daher wird die Bewertung der Methoden einzeln pro

Kante ausgeführt. Aus der Notation der Kanten ergibt sich, gemäß der Definition, Quell- und Zielsystem (C_n, C_m) sowie die dort jeweils existierenden Zeitdatenreihen (Λ). Sollte eine Zerlegung existieren, so ist diese ebenfalls im Label $\overset{\circ}{\lambda}$ der Kante beschrieben.

Diese Ausgangslage pro betrachteter Kante ist exemplarisch in Abbildung 4.8, basierend auf Ergebnissen der Zerlegung in 4.7, dargestellt. Die betrachtete Kante e repräsentiert eine gesuchte Synchronisierung von C_n auf C_m . Im entsprechenden System C_n existieren drei Zeitdatenreihen, während System C_m zwei hat. Es besteht aktuell keine Methode oder Funktion zur Synchronisierung zwischen diesen Systemen ($\lambda(e) = \epsilon$). Da allerdings $\overset{\circ}{\lambda}(e)$ nicht leer ist, existieren Zerlegungen auf Basis von Fixpunkten. Für diese Zerlegungen und darauf basierende Subsequenzen existieren aktuell ebenfalls keine Synchronisierungen (zweites Element der Tupel in $\overset{\circ}{\lambda}$ ist ϵ). Ziel dieses Schritts ist die Ersetzung der ϵ in den Kantenlabels durch Methoden (aus M) oder Funktionen (Φ). Kanten, welche bereits eine Funktion zugewiesen haben, wie etwa ein ${}^D\Phi_{n,m}$ aus 4.15, können in dieser Bewertung übersprungen werden.

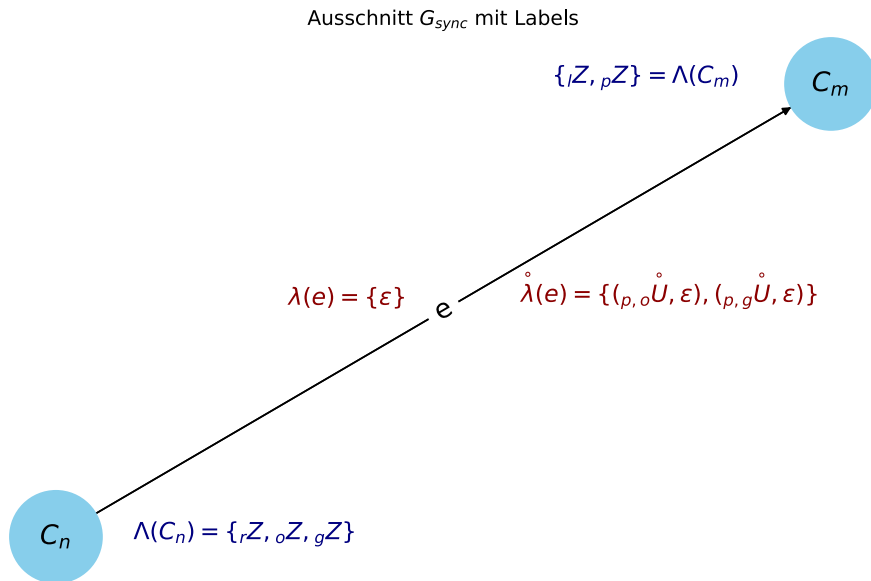


Abbildung 4.8: Exemplarische Ausgangslage innerhalb des Synchronisationsgraphs für die Bewertung von Methoden zwischen zwei Knoten. Der Rest des Graphs ist an dieser Stelle aus Gründen der Lesbarkeit ausgelassen. Dieser Graph ist zudem Ergebnis der exemplarischen Zerlegung aus Abbildung 4.7.

Der Ablauf der Bewertung der Methoden umfasst mehrere Phasen. In der ersten Phase wird die grundsätzliche Strategie für diese Kante bestimmt. Diese Strategie ergibt sich durch eine Fallunterscheidung und Vorbereitung der Zeiträume (im Sinne der Vektorräume). Diese Fälle geben zugleich eine Abschätzung der zu erstellenden Synchronisierungsvorschläge für diese Kante an. Mit K_M als Anzahl der Kandidaten zur Synchronisierung entlang einer Kante können hierbei vier Fälle unterschieden werden:

$$|\Lambda(C_n)| = |\Lambda(C_m)| = 1 \wedge \dot{\lambda}(e) = \emptyset \implies K_M = 1 \quad (4.26)$$

$$(|\Lambda(C_n)| > 1 \vee |\Lambda(C_m)| > 1) \wedge \dot{\lambda}(e) = \emptyset \implies K_M \leq \max\{|\Lambda(C_n)|, |\Lambda(C_m)|\} \quad (4.27)$$

$$|\Lambda(C_n)| = |\Lambda(C_m)| = 1 \wedge \dot{\lambda}(e) \neq \emptyset \implies K_M \leq |\dot{\lambda}(e)| \quad (4.28)$$

$$(|\Lambda(C_n)| > 1 \vee |\Lambda(C_m)| > 1) \wedge \dot{\lambda}(e) \neq \emptyset \implies K_M \leq |\dot{\lambda}(e)| + \max\{|\Lambda(C_n)|, |\Lambda(C_m)|\} \quad (4.29)$$

Im ersten Fall (4.26) existiert keine Zerlegung und pro System jeweils eine Zeitdatenreihe, sodass nur eine Synchronisierung vorgeschlagen werden muss. Bestehen bei den Systemen jedoch mehrere Zeitdatenreihen (4.27), so kann entweder analog zu Fall 1 eine Synchronisierung vorgeschlagen werden oder aber mehrere Optionen dargestellt werden (beispielsweise die Anwendung einer Methode, bei der die Zeitdatenreihe frei gewählt werden kann). Existieren allerdings Zerlegungen (4.28), so können Synchronisierungen pro Subsequenz vorgeschlagen werden. Der letzte Fall (4.29) resultiert daher in der Möglichkeit von Synchronisierungen entlang der Subsequenzen und den verschiedenen Zeitdatenreihen.

Für die beiden Systeme der Betrachtung werden die Zeitbereiche ($\langle C_n \rangle$ und $\langle C_m \rangle$) erstellt. Der Betrachtungsrahmen ergibt sich dabei jeweils aus dem kleinsten und größten Zeitwert in Λ für das System (4.16, 4.17). Für die Fälle mehrerer Zeitdatenreihen wird zudem auf die Informationen zum Produktionsablauf zurückgegriffen um zu prüfen, ob die Sensorik, welche einer Zeitdatenreihe zugehörig ist, während der gesamten Produktion aktiv war, oder nur vorübergehend. Existiert keine Zeitdatenreihe, deren zugehörige Sensorik durchgehend verfügbar war während der Produktion, so kann eine Segmentierung des Zeitbereichs eines Systems entlang der verfügbaren Zeitdatenreihen vorgenommen werden. Die fehlenden Bereiche innerhalb der Zeitbereiche werden für die Synchronisierung dann ausgelassen. Für die Fälle mit einer vorhandenen Zerlegung und jeweils einer Zeitdatenreihe werden die entsprechenden Zeitbereiche gemäß dieser Zerlegungen direkt segmentiert. Dieser beschriebene Ablauf der ersten Phase ist auch in Anhang A.1 dargestellt.

Damit resultiert diese Phase in Zeitbereichen für die beiden Knoten aus G_{sync} inklusive Segmentierung der Zeitbereiche. Für die Fälle eins, zwei und drei ergeben sich dabei die Segmentierungen als disjunkte Segmente, bis auf die Fixpunkte. Lediglich der letzte Fall kann diese Eigenschaft nicht garantieren.

Bei der exemplarischen Darstellung aus 4.8 und 4.7 führt dies zu einer eindeutigen disjunkten Segmentierung für den Zeitbereich in C_m entlang der Zerlegung der Zeitdatenreihe ${}_pZ$ in vier Segmente. Da die Zerlegungen aus C_n Überschneidungen in Bezug auf den Zeitbereich von C_m aufweisen (beispielsweise die Gegenstücke zu ${}_oU_1$ und ${}_gU_1$), kann keine direkte disjunkte Segmentierung gefunden werden. Stattdessen werden alle möglichen Segmentierungen betrachtet.

In der zweiten Phase werden für die Zerlegungen die verfügbaren Methoden M auf ihre Anwendbarkeit hin überprüft. Die Anwendbarkeit geht dabei direkt zurück auf die Evaluierung aus den Metriken der Methoden (siehe 3.3). Für die Bewertung in dieser Phase sind hierzu insbesondere die in situ Bedingungen und Nutzbarkeit auf unabhängigen Messungen sowie die Unterstützung von Nicht-Linearität entscheidend. Um diese Bewertung vorzunehmen, sind Informationen und Wissen notwendig. Eine Beschreibung der Bewertungsabläufe für die Anwendbarkeit der in Abschnitt 3.1 ausgewählten Methoden, inklusive der Parameter, ist in A.8 dargestellt. Nach Abschluss dieser Bewertung ist jeder Methode ein Anwendbarkeitswert a zugeordnet, welcher beschreibt, ob die Methode sicher anwendbar ($a = 1$), eingeschränkt anwendbar ($0 < a < 1$) oder sicher nicht anwendbar ($a = 0$) ist.

Während in der zweiten Phase der Bewertung die Anwendbarkeit der Methoden untersucht wurde, befasst sich die Phase drei mit der Bewertung der erwarteten Güte b . Für einige Methoden lässt sich eine Bewertung der Güte der Synchronisierung direkt aus der Methode ableiten oder bestimmen (siehe 3.3). Dabei ist für diese Bewertung die Metrik der erwarteten Qualität zentral, ergänzt um die integrierten Plausibilitätsprüfungen. Auf diese Art können für die Varianten in »DTW« Abschätzungen zur Qualität errechnet werden, ebenso eine Abschätzung bei »SSI«. Bei »SoE« ist die Genauigkeit direkt aus der Methode als sehr hoch abzuleiten, wenn eine Anwendbarkeit gegeben ist. Für die Methoden »ARMAX« und »NARMAX« sind die Qualitäten nicht direkt abzuschätzen, jedoch durch die Option von Plausibilitätsprüfungen (unter Rückgriff auf Regeln aus dem Wissen) ist die Güte in Abhängigkeit des eingesetzten Wissens zu bewerten. Für den Fall der »Linearisierung« ist die Bewertung der Güte von zwei Faktoren abhängig. Signifikant für die Qualität eine Linearisierung haben sich hier die Länge der Zeitdatenreihen herausgestellt. Je kürzer diese sind, desto besser ist die Güte der Synchronisierung. Zusätzlich spielt die mögliche Charakterisierung eines Systems hier ebenfalls eine positive Rolle, da bei Rückgriff auf eine Charakterisierung die Linearisierung nachgeschärft werden kann. Die entsprechende

Bewertung für die Linearisierung geht daher auf eine Stellgröße und Grenzwerte der Länge von Zeitdatenreihen zurück für das System.

Zuletzt kann noch eine einfache Bewertung der Berechnungskomplexität c durchgeführt werden für die Methoden. Diese Bewertung ergibt sich aus den Methoden selbst und kann damit als einmalige statische Bestimmung zu Beginn durchgeführt werden, muss also nicht pro Kante neu erstellt werden.

In Phase drei wird auf Basis der Bewertungen der Anwendbarkeit a , erwarteten Güte der Synchronisierung b und Berechnungskomplexität c eine Gesamtbewertung pro Methode erstellt. Wie die Verrechnung dieser Parameter genau geschieht, ist eine Stellgröße des Systems. Durch Veränderung der Gewichtung der jeweiligen Einzelbewertungen lässt sich das Synchronisierungssystem grundsätzlich ausrichten.

Anpassungen Synchronisationsgraph

Im Rahmen des vorherigen Schritts entsteht schrittweise eine Liste mit den Gesamtbewertungen für jede untersuchte mögliche Synchronisierung. Basierend auf diesen Gesamtbewertungen werden nun für die Kanten e die Kantenlabel $\lambda(e)$ mit der entsprechend am besten bewerteten Methode befüllt. Für Kanten, bei denen bereits eine existierende Projektionsfunktion Φ existiert, bleibt diese Funktion in $\lambda(e)$ erhalten.

Für Kanten, bei denen eine Zerlegung existiert ($\overset{\circ}{\lambda}(e) \neq \emptyset$), muss die Zerlegung an die gewählten Methoden nach der Gesamtbewertung angepasst werden. Greift die gewählte Menge von Methoden die maximale Zerlegung auf, so ist keine Änderung von $\overset{\circ}{\lambda}(e)$ notwendig. Nutzen die gewählten Methoden jedoch nur teilweise die Zerlegung, so werden die entsprechenden Subsequenzen zusammengeführt (vereint). Wird hingegen bei der gewählten Methode vollständig auf die Zerlegung verzichtet, so gilt $\overset{\circ}{\lambda}(e) = \emptyset$. Dadurch ist aus der Kombination der Kantenlabel die mögliche Anwendung der Methoden und ihrer betrachteten Sequenzen eindeutig.

Wird eine Kante mit einer Methode in $\lambda(e)$ festgelegt, kann eine Optimierung im Graph vorgenommen werden. Ausgehend von einer Senke (KN_{sink}) wird die Datenpipeline für die entsprechenden Zeitdatenreihen geprüft und es werden mögliche alternative Pfade betrachtet. Ist hier eine Projektion nah an der Senke mit hoher Bewertung und Güte vorhanden, so können Pfade höherer Entfernung ausgeschlossen werden. Ist der Pfad hingegen von mäßiger Güte und ein anderer Pfad in höherer Entfernung zur Senke stellt eine bessere Güte sicher, so kann der erste Pfad entfernt werden im Hinblick auf die Synchronisierung.

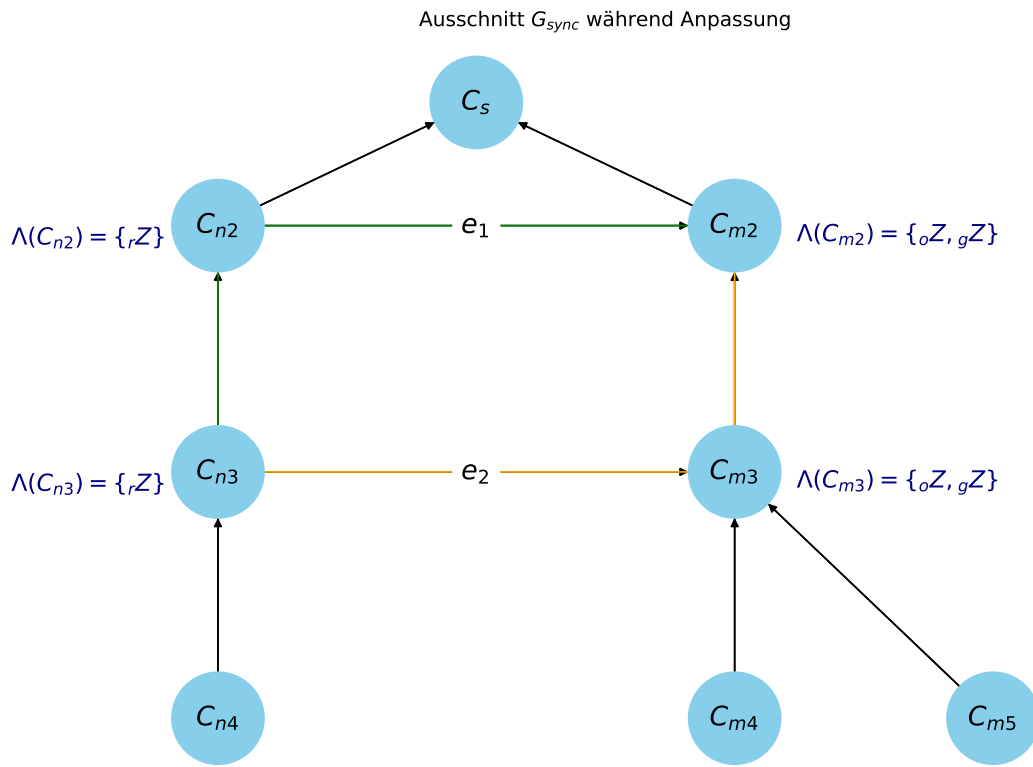


Abbildung 4.9: Exemplarische Darstellung des Synchronisationsgraphs mit mehreren möglichen Kanten e_1 und e_2 . Abhängig der Bewertung der einen Kante kann eine Optimierung in Form eines Ausschlusses der anderen Kante während der Anpassung vorgenommen werden. Aus Gründen der Lesbarkeit sind weitere Label und Kanten nicht dargestellt.

Dies ist exemplarisch in Abbildung 4.9 dargestellt. Für die Synchronisierung von rZ auf die Zeitdatenreihe oZ, gZ bestehen die Möglichkeiten repräsentiert durch Kanten e_1 und e_2 . Von der Senke C_s ausgehend, sollte e_1 zuerst betrachtet werden. Ist die erwartete Güte im Sinne der Gesamtbewertung maximal (oder oberhalb eines Grenzwertes), so kann die Kante e_2 von der weiteren Betrachtung ausgeschlossen werden, da keine Verbesserung der Datenqualität zu erwarten ist. Ist hingegen die Güte von e_1 beschränkt, wird e_2 bewertet. Falls die Güte von e_2 höher als von e_1 ist, so kann analog e_1 entfernt werden. Dadurch werden letztlich der grüne und der orange Pfad im Graph gegeneinander bewertet und dieser Teil optimiert.

Diese Anpassungen der Kantenlabel können teilweise parallel zu den Bewertungen direkt ausgeführt werden oder die Reihenfolge der Bewertung beeinflussen. Dadurch sind diese Schritte miteinander verbunden.

Empfehlung Synchronisationsablauf

Mit dem vollständigen Synchronisationsgraphen G_{sync} wird in diesem Schritt die Menge der Synchronisationen ausgewählt, welche durchgeführt werden sollen, um die Synchronisierung der Offline Produktionsdaten zu erreichen. Hierzu lassen sich Graphalgorithmen anwenden unter der Verwendung der Kantenlabel λ als Gewichtsfunktion. Eine bessere Güte der Synchronisierungen entspricht hierbei einem niedrigeren Gewicht.

Das Ziel des Synchronisierungssystems ist die Synchronisierung der Zeitdatenreihen im Produktionssystem. Bezogen auf den Synchronisationsgraphen bedeutet dies, dass es für jede Kombination von Zeitdatenreihen einen Pfad gibt, der bei einem Knoten beginnt, wo die eine Zeitdatenreihe existiert und zu einem anderen Knoten führt, der die andere Zeitdatenreihe umfasst. Der Teilgraph, welcher die resultierenden Kanten umfasst wird als *TSG* (»Time Spanning Graph«) bezeichnet:

$$\begin{aligned} TSG(G_{sync}) &\subseteq KA \\ TSG(G_{sync}) &= \bigcup_{(nZ, mZ)} W_{nZ, mZ} \\ W_{nZ, mZ} &= ((C_1, C_2), \dots, (C_j, C_k)) \mid nZ \in \Lambda(C_1), mZ \in \Lambda(C_k), \end{aligned} \quad (4.30)$$

wobei $W_{nZ, mZ}$ einen Pfad beschreibt, der bei einem Knoten zur ersten Zeitdatenreihe startet und bei einem Knoten zur zweiten Zeitdatenreihe startet. Zur Optimierung der Synchronisierungen und Vermeidung von Zyklen wird basierend auf den Kantengewichten die Definition angepasst auf ein Minimierungsproblem:

$$\begin{aligned} W_{nZ, mZ} &= \arg \min \sum_{e \in W} \omega(e) \mid W = ((C_1, C_2), \dots, (C_j, C_k)), nZ \in \Lambda(C_1), mZ \in \Lambda(C_k) \\ \omega : KA &\rightarrow \mathbb{R}^+, \end{aligned} \quad (4.31)$$

wobei ω die Gewichtungsfunktion der Kanten beschreibt. Diese Gewichtung kann dabei direkt auf die Gesamtbewertung zurückgehen oder eine zusätzliche Anpassung umfassen. Damit ist die Ausgestaltung von ω ein Stellparameter des Systems.

Da der gesuchte *TSG* auf einem gerichteten Graph berechnet wird, ergeben sich weitere Bedingungen. So ist die Senke im Graph auch die Senke des *TSG*. Damit kann für die Senke angenommen werden, dass lediglich eine Kante zur Senke des *TSG* existiert. Daher erhalten alle Kanten zur Senke das Gewicht $2 \cdot N$ zugewiesen. Gleichzeitig muss der *TSG* von den Quellen ausgehen. Damit entspricht der *TSG* der Definition eines In-Trees für gewichtete gerichtete Graphen. Damit kann grundsätzlich Edmonds' Algorithmus angewendet werden [EDMO67]. In dem Fall des *TSG* soll die Verbindung zur Senke zuletzt geschehen, und das Kantengewicht aus den damit verbundenen Knoten ist überall gleich. Bei der direkten Anwendung von Edmonds' Algorithmus bedeutet dies, dass das Ergebnis nicht deterministisch ist und von der Wahl der ersten Kante aus der Senke abhängt. Um dies zu adressieren, findet die Berechnung nicht für die Senke statt, sondern einmalig für jeden Knoten mit einer Verbindung zur Senke. Daraus ergibt sich eine Menge von In-Trees mit verschiedenen Gewichten. Durch Auswahl des minimalen In-Trees dieser Menge, und hinzufügen der entsprechenden Kante zur Senke, entsteht der gesuchte *TSG*. Die Menge der Kanten des *TSG* entspricht damit der Empfehlung des Ablaufs der Synchronisierung.

Am Ende dieses Hauptschritts sind auf Basis des Synchronisationsgraphen und der eventuellen Zerlegung mögliche Methoden der Synchronisierung für alle Knotenpaare bewertet und entlang der Kantenlabel aufgenommen. Mithilfe einer Gewichtungsfunktion ist eine Empfehlung der durchzuführenden Synchronisierungen und deren Reihenfolge erstellt.

4.3.4 Durchführung

Execution

Im letzten Hauptschritt findet die eigentliche Durchführung der Synchronisierungen der Zeitdatenreihen statt. Ebenfalls können Prüfungen der Plausibilität und Abschätzungen der Unsicherheiten getroffen werden.

Durchführung Synchronisationen

Als Ergebnis des Hauptschritts der Entscheidung existiert eine Liste von empfohlenen Synchronisierungen, der zu nutzenden Methoden, Zerlegungen und die Reihenfolge dieser. Zu Beginn dieses Schritts besteht die Möglichkeit des manuellen Eingriffs (☞), bei

der eine Person diese Liste anpassen und ändern kann. So kann diese Person weitere Synchronisierungen hinzufügen oder entfernen, die Methoden anpassen oder die Reihenfolge umstellen. Dies kann beispielsweise bei Synchronisierungen, welche für spätere Auswertungen unerheblich sind, eine manuelle Optimierung der Durchführung sein.

Gemäß der Liste werden nun die einzelnen Synchronisierungen durchgeführt und die damit ergebenden Projektionsfunktionen Φ erstellt. Dieser Schritt ist im Sinne der Berechnungsaufwände der komplexeste Schritt im gesamten Ablauf des Synchronisierungssystems. Zugleich ist der Ablauf selbst durch die Vorgaben der Liste klar und die Durchführung nutzt die vorgestellten Methoden entsprechend. Am Ende dieses Schritts existieren für alle Kanten des Synchronisierungsgraphs die entsprechenden Projektionsfunktionen, welche die Abbildung der Zeitangaben der jeweiligen Knoten (Systeme) ermöglicht.

Zusammenführung Zeitdatenreihen

Im letzten Schritt der eigentlichen Synchronisierung werden die Projektionsfunktionen genutzt, um die Menge der Zeitdatenreihen $\mathfrak{Z}$ anzupassen. Zunächst wird eine Zeitreihe als Referenz bestimmt. Dies kann die echte Zeit sein, wenn entsprechende Projektionen vorhanden sind, oder die Zeitreihe aus der Zeitdatenreihe mit der besten Bewertung der Zeitinformationen (mit den Eigenschaften in 4.3.1). Diese Referenz REF ist damit das Ziel aller Projektionen.

Entsprechend der Strategie aus Abschnitt 4.3.3 werden die Projektionsfunktionen nun entlang der Datenpipelines (Informationen aus $\mathfrak{J}$) verkettet mit der letztlichen Ausrichtung auf REF . Diese verketteten Projektionen $\Phi_{n,REF}$ werden dann auf alle Zeitdatenreihen aus $\mathfrak{Z}$ angewendet. Daraus ergibt sich die Menge aller synchronisierten Zeitdatenreihen zu:

$$\mathfrak{Z}_{sync} = \{ \{ (\Phi_{n,REF}(t_i), v_i^n) \} \mid \{ (t_i, v_i^n) \} \in \mathfrak{Z} \} \quad (4.32)$$

Damit ist $\mathfrak{Z}_{sync}$ das zentrale Ergebnis des Synchronisierungssystems und der Synchronisierung der Produktionsdaten.

Plausibilitätsprüfung und Unsicherheitsbetrachtung

Zusätzlich zu der eigentlichen Synchronisierung können weitere Bewertungen der Synchronisierung vorgenommen werden. Unter Rückgriff auf Informationen zum Produkti-

onssystem und dem Wissen um Zusammenhänge von physikalischen Effekten können datenbasiert Plausibilitätsprüfungen vorgenommen werden. Hierbei wird insbesondere auf die Domänenregeln $\mathbb{P}$ zurückgegriffen, welche das Wissen entsprechend formalisieren und prüfbar sind. Diese Prüfung selbst kann dabei verschieden detailliert ausfallen. Bei Regeln, welche die Gleichzeitigkeit von Werten beschreiben (beispielsweise eine maximale Abweichung zweier Werte bei gleichzeitiger Betrachtung), kann diese Prüfung daher pro Zeitwert, in Stichproben oder zu definierten bestimmten Zeitpunkten betrachtet werden. Da die Prüfung der Plausibilität dadurch selbst möglicherweise eine nicht-lineare Komplexität erhält, ist die Detail-Stufe der Durchführung der Prüfungen ein Stellparameter für das System. Alternativ kann diese Prüfung auch gänzlich ausgelassen werden oder durch einen manuellen Eingriff geschehen.

Der zweite Teil der Bewertung betrifft die Unsicherheit der durchgeführten Synchronisierung. Für alle $\Phi_{n,REF}$ wird geprüft, ob eine definierte Unsicherheit in den einzelnen Methoden und Projektionen vorliegt. Diese Unsicherheiten werden dann aufsummiert. Im Falle von Zerlegungen wird ebenfalls die Unsicherheit der Fixpunkte $\Delta(\xi)$ addiert. Zuletzt wird die zeitgeberbedingte Unsicherheit (halbe Zykluszeit) addiert. Können alle benannten Unsicherheiten definiert und quantifiziert werden, so beschreibt die Summe dieser Unsicherheiten die Gesamtunsicherheit der Synchronisierung der entsprechenden Zeitdatenreihe. Ist mindestens einer dieser Einträge nicht bestimmbar, so entspricht die Summe der unteren Grenze der Gesamtunsicherheit.

Die bestimmten Gesamtunsicherheiten sowie die Ergebnisse der Plausibilitätsprüfungen stellen neben dem Hauptergebnis, den synchronisierten Zeitdatenreihen, zusätzliche Informationen bereit, welche bei Bedarf aus dem Synchronisierungssystem übergeben werden.

4.4 Implementierung

Implementation

In diesem Kapitel wurde ein Synchronisationssystem ausgestaltet und eine Reihe von Eingangsgrößen beschrieben. All diese Komponenten müssen für eine spätere Validierung programmatisch umgesetzt werden. Da hier auch die Automatisierbarkeit, Skalierbarkeit und Anwendungsnähe relevante Bewertungsgrößen sind, muss eine Implementierung ebenfalls in ein Produktionsnetzwerk integrierbar sein. Daher werden alle dargestellten Komponenten als Software implementiert und deren Verbindungen dargestellt.

4.4.1 Generelle Implementierungen

General implementations

Die Komponenten basieren hier, sofern nicht anders angegeben, auf dem Technologie-Stack C# (.Net), oder Python. Diese Stacks sind in dem Bereich des Data Engineerings und der Produktionsumgebungen verbreitet und eine Vielzahl von Bibliotheken für übliche Aufgaben in diesen Bereichen sind verfügbar. Dies erlaubt eine effiziente Implementierung, da auf diese Art keine Standardimplementierungen vorgenommen werden müssen, sondern die Entwicklung auf die neuen Komponenten fokussiert wird.

Als Laufzeitumgebungen kommen Windows oder Ubuntu Linux zum Einsatz. Windows hat in der Industrie und der Produktion einen hohen Verbreitungsgrad, ebenso sind einige der eingesetzten Hardware-Komponenten ausschließlich mit Windows kompatibel. Linux ist insbesondere in der Virtualisierung, wie Cloud-Umgebungen, dominant. Daher wird für die virtuellen Maschinen auf Linux gesetzt bei der Implementierung und dem Hosting. Im Speziellen wird Ubuntu Linux genutzt aufgrund der hohen Kompatibilität mit den eingesetzten Technologien sowie der langfristigen Unterstützung (»LTS«) und der Konsistenz mit den anderen virtuellen Systemen in dem Produktionsnetzwerk des Anwendungsbereichs (Kapitel 5).

Im Falle von Schnittstellen für Softwarekomponenten, die auf dem gleichen System oder über ein Netzwerk zugänglich gemacht werden, werden entweder die bereits integrierten Schnittstellen (bei Datenbanken) oder entsprechend entwickelte Schnittstellen auf Basis von REST [FIEL00] umgesetzt.

Das Thema Sicherheit im Sinne der IT-Sicherheit wird bei diesen Implementierungen nicht explizit behandelt. Hierbei wird für die Systeme auf Grundmittel wie Verschlüsselung der Kommunikation und der Festplatten gesetzt sowie grundsätzliche Zugangskontrollen und Rollen aus dem Sicherheitskonzept des Produktionsnetzwerks des Anwendungsfalls.

Für alle Definitionen gemäß Kapitel 2, wie die Zeitdatenreihen, Formatierungen von Zeitwerten und Vorverarbeitungsschritte abseits der Synchronisierung wurden direkte Implementierungen mit den integrierten Funktionen der Sprachen selbst direkt umgesetzt und sind unter A.10 zu finden. Auf diese Basisimplementierungen wird daher hier nicht genauer eingegangen.

4.4.2 Implementierung der Eingangsgrößen

Implementation of input parameters

Für die verschiedenen Eingangsgrößen aus Abschnitt 4.2 sind Implementierungen notwendig. Diese geschehen dabei getrennt voneinander und adressieren grundsätzlich verschiedene Aspekte. Daher ist bei diesen Implementierungen stets die Nutzung von Programmschnittstellen (»API«) notwendig.

Charakterisierung Systeme

Der Versuchsaufbau, wie in Abbildung 4.4 dargestellt, wird durch eine einzelne Software realisiert. Im Rahmen der Implementierung wird für die Präzisionsuhr eine Karte der Firma Meinberg mit entsprechenden Oszillatoren verwendet [MEIN23b]. Für die Verbindung zu den Zeitwerten aus der Präzisionsuhr der Meinberg-Oszillatoren wird daher auf das Meinberg SDK [MEIN23a] gesetzt. Für den internen Zeitgeber C_2 wird hierbei eine Implementierung über die Windows *FileTime* statt normaler Systemzeit genutzt, da diese präziser und stabiler ist [MICR24].

Die Zeit aus der angeschlossenen Maschine wird über die OPC UA Schnittstelle aufgenommen. Die Implementierung geht hier davon aus, dass eine direkte Verbindung zu der Maschine existiert, sodass keine Maßnahmen aufgrund von Jitter der Verbindung getroffen werden müssen. Daher wird für die Implementierung auf die offiziellen Referenzimplementierungen [OPC 18] zurückgegriffen.

Für den Ablauf wird eine Subscription auf die Werte im OPC UA Server (C_1), inklusive der Serverzeit, erstellt. Empfängt das Programm nun eine solche Nachricht aus C_1 werden parallel die Zeiten aus dem System selbst (C_2) und der Präzisionsuhr (C_{ref}) bestimmt. Als Resultat ergibt sich ein Tupel $(t_i^1, t_i^2, t_i^{ref}, v_i, \dots)$, welches die Zeitinformationen aller Zeitgeber und die zugehörigen Werte aus der Maschine beinhaltet. Dieses Tupel wird kontinuierlich exportiert, sodass eine fortlaufende Zeitdatenreihe mit mehreren Zeitwerten existiert. Im Rahmen der Auswertung kann nun diese Zeitdatenreihe eingeladen werden und Informationen wie der Verlauf der Asynchronizität durch laufende Differenzen bestimmt werden.

ID	S1	XX	XX_Para	T/D	S2	YY	YY_Para
#1	Druck Pumpe	Anstieg	Ramp: 10 %	600 ms	Dicke Schicht	Anstieg	Ramp: 15 %
#2	Dicke Schicht	Peak	Peak: 20 %	0 m	Dicke Trocken	Peak	Peak: 15 %

Tabelle 4.1: Die Wirkdatenbank als Tabellenstruktur für eine Datenbank im SQL-Server

Wirkdatenbank

Die Wirkdatenbank aus Abschnitt 4.2.3 ist der zentrale Wissensspeicher für das Synchronisierungssystem. Da auf dieses Wissen von Methoden, dem Synchronisierungssystem, als auch Menschen und eventuellen weiteren Systemen zugegriffen werden soll, wird die Wirkdatenbank als zentrale Komponente im Produktionsnetzwerk umgesetzt. Dabei besteht eine virtuelle Maschine mit Netzwerkzugriff für alle möglichen zugreifenden Akteure. Da die Modellierung beliebig komplex sein kann, wird diese hier als eine Tabelle in einem SQL-Server umgesetzt. Die korrekte Formalisierung solchen Expertenwissens ist ein eigenes Forschungsfeld, sodass hier eine pragmatische Lösung genutzt wird, welche im Folgenden beschrieben ist.

Die Struktur dieser Tabelle ist damit in Tabelle 4.1 dargestellt. Die Spalte *ID* ist dabei eine fortlaufende Nummer und stellt den Primärschlüssel dieser Tabelle dar. In Spalte *S1* ist die zu beobachtende physikalische Größe bezeichnet. Die Spalten *XX* und *XX_Para* beschreiben den Effekt zu beobachten in der physikalischen Größe durch den allgemeinen Effekt (*XX*) und eine Quantifizierung (*XX_Para*). Analog dazu beschreiben die Spalten *S2*, *YY* und *YY_Para* eine physikalische Größe und zu beobachtenden Effekt als Konsequenz. Die erwartete Verzögerung zwischen diesen Größen ist durch die Spalte *T/D* gegeben als Zeitinformation. Diese Angabe kann alternativ auch eine Distanzinformation entlang der Länge des Bandes darstellen bei bahnführenden Prozessen. Im gegebenen Beispiel führt ein Anstieg von mindestens 10 % des Pumpendrucks zu einer Erhöhung der Schichtdicke nach 600 Millisekunden um mindestens 15 % und ein Ausreißen der Schichtdicke um 20 % zu einem Ausreißen von 15 % der trockenen Schichtdicke an der gleichen Bandposition.

Der SQL-Server mit dieser Wirkdatenbank ist im Produktionsnetzwerk erreichbar. Für die Umsetzung wurde hier die Software *MariaDB* gewählt, wodurch das Datenbankmanagementsystem über das binäre MySQL-Protokoll erreichbar ist. Alternativ kann auch eine pure lokale Implementierung auf Basis von *LiteDB* zurückgegriffen werden, um die Abhängigkeit von einem Datenbankserver zu reduzieren. Die genaue Auswahl der Einträge ist dabei in

dieser Arbeit zugeschnitten auf die jeweiligen Fälle in der Validierung in Kapitel 5. Eine allgemeingültige und vollständige Aufstellung von Wirkzusammenhängen der gesamten Batterieproduktion ist außerhalb des Rahmens dieser Arbeit.

4.4.3 Implementierung der Systemmodule

Implementation of system modules

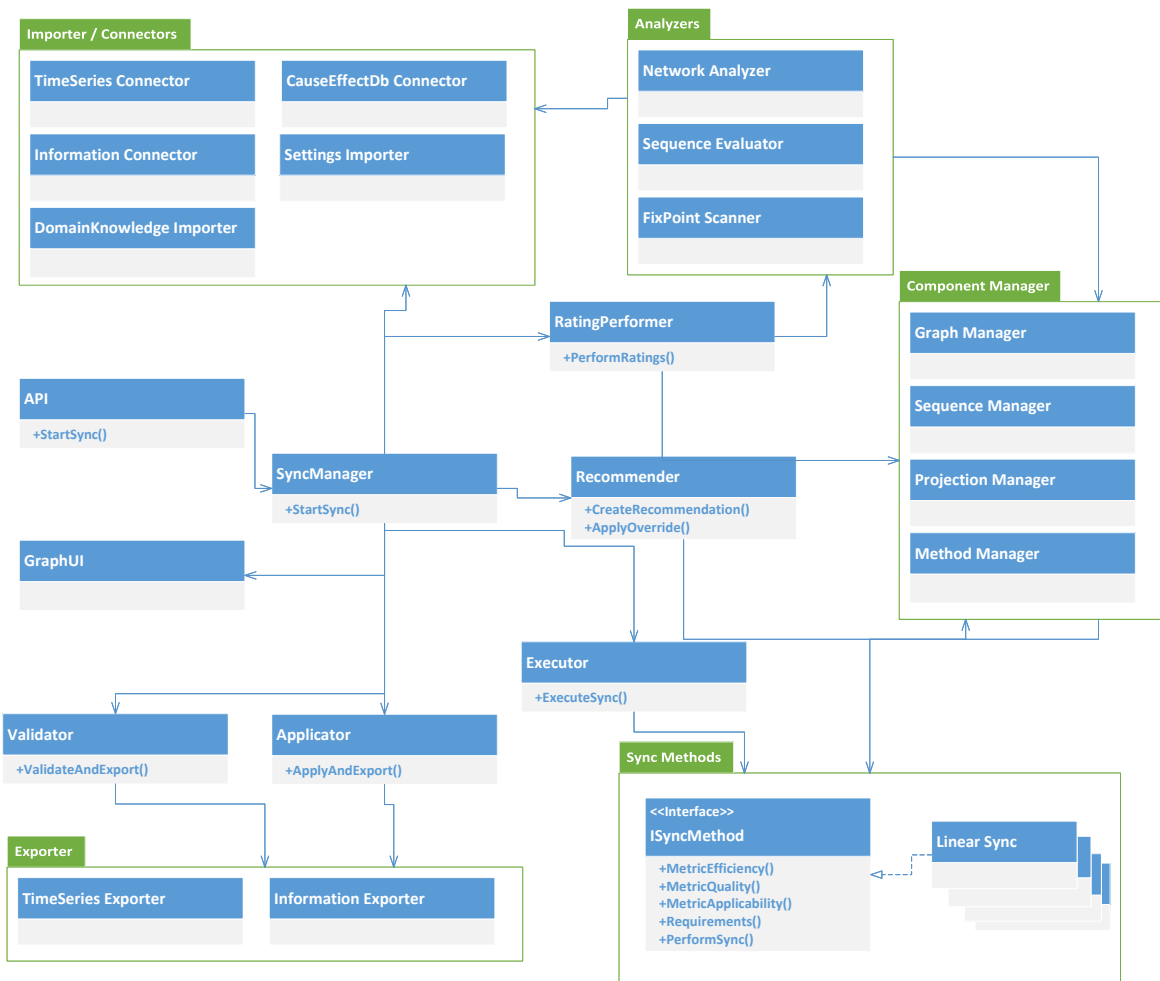


Abbildung 4.10: Darstellung der Komponenten der Implementierung des Offline Synchronisierungssystems gemäß UML. Pfeile (Assoziationen) beschreiben direkte Zugriffe auf entsprechende Komponenten. Im Normalfall ist der Startpunkt der Synchronisierung ein Aufruf über die API durch einen Trigger im Produktionsnetzwerk. Aus Gründen der Lesbarkeit sind nur einzelne Methoden und keine Eigenschaften oder Felder dargestellt.

Die Komponenten der Implementierung des Offline Synchronisierungssystems sind mit ihrem Zusammenspiel gemäß UML in Abbildung 4.10 dargestellt. Insgesamt ist die Software in verschiedene Bereiche und Bibliotheken aufgeteilt. Einstiegspunkt ist die Schnittstelle (*API*), welche per REST Aufruf angesprochen werden kann, um die Synchronisierung zu starten. Der *SyncManager* ist dabei das Herzstück, welches den Daten- und Kontrollfluss steuert. Er startet mit dem Import der Stellparameter des Systems. Die Übersicht der wichtigsten Stellparameter befindet sich in der Tabelle 10.

Bei einer Synchronisierung erstellt der *SyncManager* zunächst die notwendigen Manager (*Component Manager*), Analyzer (*Analyzers*) und Importer (*Importer / Connectors*), die für den weiteren Ablauf benötigt werden. Auch werden hier die Synchronisierungsmethoden registriert (über den *MethodManager*), sodass diese Methoden zur Verfügung stehen bei den Hauptschritten der Offline Synchronisierung. Die Details der Methoden sind im weiteren Verlauf genauer dargestellt.

Nach diesen Vorbereitungsschritten ruft der *SyncManager* die einzelnen Komponenten auf entlang der Hauptschritte, wie in 4.1 beschrieben. An den Stellen der manuellen Interaktion kann die explizite Schnittstelle zu einem Menschen erstellt und aufgerufen werden (*GraphUI*), die später genauer beschrieben ist. Nach Abschluss der Synchronisierung werden die Zeitreihen und die begleitenden Informationen über die *Exporter* außerhalb des Systems verfügbar gemacht.

4.4.4 Implementierung der Synchronisierungsmethoden

Implementation of synchronization methods

Für die Umsetzung des Offline Synchronisierungssystems sind die eigentlichen Methoden in der Implementierung elementar. Innerhalb des Hauptsystems (siehe UML-Darstellung 4.10) sind die Methoden zunächst abstrahiert und vom System über eine Schnittstelle angesprochen. Dadurch ist die Erweiterbarkeit im Nachhinein um weitere Methoden einfach umsetzbar. Weiterhin nutzen die eigentlichen Methoden verschiedene Formatierungen oder Technologien, sodass diese Abstraktion zudem als Übersetzung und Wrapper dient.

Im Bereich Dynamic Time Warping setzt die Implementierung aufgrund der vorhandenen Bibliotheken auf Python. Für die grundsätzliche Dynamic Time Warping Implementierung kommt das verbreitete Paket von Giorgino zum Einsatz [GIOR09] in der Version aus September 2024. Dieses beinhaltet bei Bedarf auch zusätzliche Optimierungen wie das Sakoe-Chiba-Band und das Itakura-Parallelogram. Für die speziellen Varianten sind die entsprechenden Referenz-Implementierungen genutzt. Bei FastDTW entspricht das dem

entsprechenden Code-Stand von 2019 [SLYP19] und bei ShapeDTW dem Stand von 2023 [MIKO23]. Bei all diesen Varianten wird die Synchronisierung wie in Kapitel 3.1.1 durchgeführt. Die dabei entstehende Zuordnung der Punkte der Zeitreihen wird dabei direkt als Projektionsfunktion Φ im Gesamtsystem gespeichert.

Bei den Varianten der Systemidentifikation ist ebenfalls der Weg über Python-Programme notwendig. Für die Umsetzung von ARMAX und NARMAX kommt hier das Paket *sysidentPy* zum Einsatz, welches neben vielen Optimierungen auch die Vorgabe von Verzögerungen ermöglicht [JUNI20]. Für SSI hingegen ist das Paket *nfoursid* genutzt, welches der Definition in Kapitel 3.1.2 entspricht [VO94, VERH07]. Da die Methoden der Systemidentifikation selbst keine direkte Synchronisierungsmethoden sind, geschieht die Anwendung jeweils in zwei Schritten. Im ersten Schritt wird ein Modell erstellt auf Basis der übergebenen Daten. Im zweiten Schritt werden die Zeitdatenreihen auf das Modell verschoben. Daraus entsteht eine Zuordnung, welche die Projektionsfunktion Φ im Gesamtsystem darstellt.

Für Sequence of Events existiert einerseits keine öffentlich zugängliche Referenzimplementierung, die verwendet werden kann, andererseits ist die Methode wie in Abschnitt 3.1.3 dargestellt zunächst für Ereignisse und nicht Zeitdatenreihen ausgelegt. Daher kommt hier eine eigene Implementierung auf Basis der Spezifikation [PI W17] mit Anpassungen für Zeitdatenreihen zum Einsatz. Die Implementierung beschränkt sich dabei auf die Komponenten der Synchronisierung, sodass andere Teile der Spezifikation ausgelassen sind. Es handelt sich damit um eine angepasste Implementierung, welche nicht die vollständige allgemeine Spezifikation erfüllt, was in dem Kontext dieser Arbeit ausreichend ist. Die Implementierung ist in C# und gibt direkt eine Projektionsfunktion Φ zurück ins Gesamtsystem.

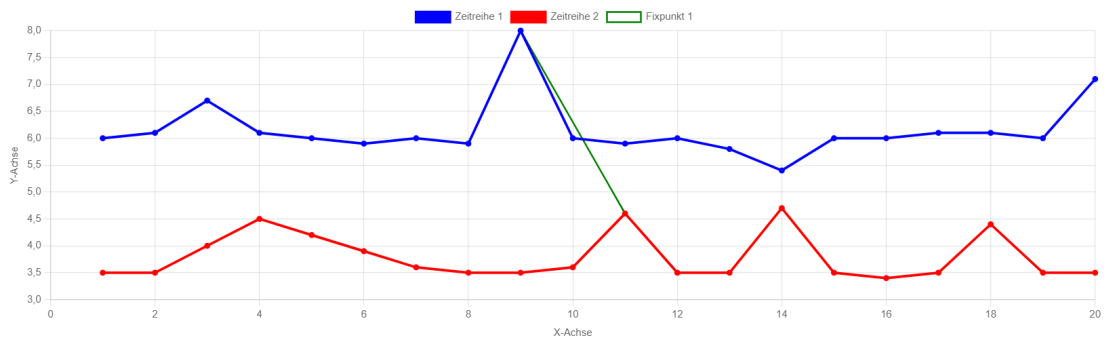
Die Linearisierung ist eine eigene Implementierung in C#. Diese Methode ist direkt und einfach umsetzbar, sodass die Projektionsfunktion Φ zurückgegeben wird.

4.4.5 Schnittstellen für Menschen

User interfaces

Wie in Abbildung 4.1 dargestellt, existieren im Betrieb des Synchronisierungssystems zwei Punkte, wo ein manueller Eingriff stattfinden kann. Der erste Eingriffspunkt betrifft die Zerlegung. Nachdem Fixpunkte identifiziert wurden, besteht die Option, einen Menschen dies überprüfen und bei Bedarf anpassen zu lassen. Diese erste Schnittstelle ist in Abbildung 4.11 als Screenshot dargestellt. Die beiden Zeitdatenreihen sind dargestellt und in Grün die Verbindung als Fixpunkt, die vom System gefunden wurde. Ist ein manueller

Zeitreihen-Visualisierung



Verbindungen bearbeiten

Verbindung 1: Fixpunkt 1

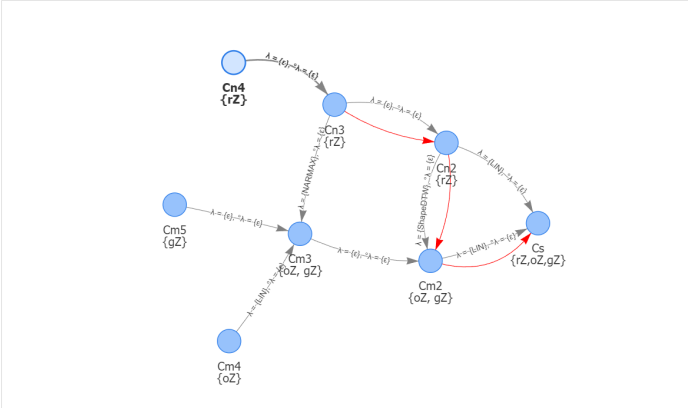
Abbildung 4.11: Schnittstelle (User Interface) für die Betrachtung von Fixpunkten, die vom System im Rahmen der Zerlegung identifiziert wurden. Im unteren Bereich bestehen Eingabemöglichkeiten, sodass ein Mensch diese Fixpunkte anpassen, löschen oder neue hinzufügen kann. Die dargestellten Daten basieren auf Darstellung 4.6.

Eingriff notwendig, kann ein Mensch mit den Elementen unterhalb der Visualisierung die Fixpunkte anpassen, löschen oder neue hinzufügen.

Der zweite Punkt, wo ein Eingriff durch einen Menschen möglich ist, ist zu Beginn der Durchführung der Synchronisierung, dargestellt in Abbildung 4.12. So wird der erstellte Synchronisationsgraph und die Empfehlung darauf basierend als Pfad durch diesen Graphen in Rot dargestellt. Aufgrund der potenziellen Komplexität des Graphen ist die Darstellung interaktiv, sodass ein Mensch die Knoten verschieben kann, um alles zu erkennen. Die Bearbeitungsmöglichkeiten beziehen sich grundsätzlich auf die Knoten, Kanten und den Pfad. Insbesondere durch die Anpassung der Kantenlabel kann so die gewählte Methode einer einzelnen Synchronisierung angepasst werden und bei der Veränderung des Pfads der gesamte Synchronisierungsablauf.

Technisch umgesetzt sind diese Schnittstellen als ASP.NET Core Applikationen mit Rückgriff auf JavaScript für die Visualisierungen mittels Vis.js [B.V.14] und Chart.js [DOWN13]. Der Datenaustausch in beide Richtungen läuft über eine REST Schnittstelle.

Synchronisierungsgraph Anzeigen/Bearbeiten



Bearbeitung des Graphen

Knoten bearbeiten (Nur für Fehler)

Knoten Cn2:	Cn2	{rZ}
Knoten Cn3:	Cn3	{rZ}
Knoten Cn4:	Cn4	{rZ}
Knoten Cm2:	Cm2	{oZ, gZ}
Knoten Cs:	Cs	{rZ, oZ, gZ}
Knoten Cm3:	Cm3	{oZ, gZ}
Knoten Cm4:	Cm4	{oZ}
Knoten Cm5:	Cm5	{gZ}

Kanten bearbeiten (Methodenauswahl)

Kante Cn4 -> Cn3:	$\lambda = \{e\}$	$\lambda = \{e\}$
Kante Cn3 -> Cn2:	$\lambda = \{e\}$	$\lambda = \{e\}$
Kante Cn3 -> Cm3:	$\lambda = \{NARMAX\}$	$\lambda = \{e\}$
Kante Cn2 -> Cm2:	$\lambda = \{ShapeDTW\}$	$\lambda = \{e\}$
Kante Cn2 -> Cs:	$\lambda = \{LIN\}$	$\lambda = \{e\}$
Kante Cm2 -> Cs:	$\lambda = \{LIN\}$	$\lambda = \{e\}$
Kante Cm4 -> Cm3:	$\lambda = \{LIN\}$	$\lambda = \{e\}$
Kante Cm5 -> Cm3:	$\lambda = \{e\}$	$\lambda = \{e\}$
Kante Cm3 -> Cm2:	$\lambda = \{e\}$	$\lambda = \{e\}$

Pfad bearbeiten (Ablauf Synchronisierung)

Pfad: Cn3,Cn2,Cm2,Cs
Daten an API senden

Abbildung 4.12: Schnittstelle (User Interface) für die Betrachtung und Bearbeitung des Synchronisierungsgraphs, der vom System erstellt wurde. Die linke Seite stellt diese Visualisierung dar, während die rechte Seite in der Darstellung die Bearbeitungsmöglichkeiten anzeigt. In der eigentlichen Schnittstelle sind diese Elemente untereinander, aus Gründen der besseren Darstellung sind diese Elemente hier nebeneinander. Der dargestellte Graph basiert auf der Abbildung 4.9.

4.5 Zwischenfazit Synchronisierungssystem

Interim Conclusion Synchronization System

In diesem Kapitel wurden die Grundstruktur des Offline Synchronisierungssystems dargestellt und die einzelnen Schritte sowie die Eingangsgrößen beschrieben. Für die neuen Methoden kann zwischen der direkten Anwendung (Auslassung Hauptschritt 2 in Abbildung 4.1) und der Durchführung mit Zerlegung (Alle Hauptschritte) unterschieden werden. Im Folgenden werden diese Varianten als die Methoden »Direkt« beziehungsweise »Zerlegung« bezeichnet. Die Implementierung des Offline Synchronisierungssystems erfolgt dabei in C# und Python, wobei die einzelnen Methoden in den jeweiligen Sprachen umgesetzt werden. Die Schnittstellen für Menschen sind als Webapplikationen umgesetzt, um eine einfache Bedienung zu ermöglichen.

Mit dieser Implementierung sind die Zugriffspunkte und Automatisierungsabläufe klar

beschrieben, was formal bereits auf die vierte Unterforschungsfrage (**UFF 4** - »Lässt sich die Offline Synchronisierung von Zeitdatenreihen für hochskalige Produktionen automatisieren?«) einzahlt, da hier bis auf die beiden optionalen manuellen Eingriffe alle Schritte automatisiert sind. Die Implementierung ist damit bereit für die Validierung und die Bewertung entlang der Anforderungen aus Abschnitt 3.2.

5 Verifikation und Validierung

Verification and Validation

Basierend auf der Entwicklung, beschrieben in Kapitel 4, sind die Software und ihre Anwendung zu verifizieren und validieren. Für das Offline Synchronisierungssystem kann auf verschiedene Methoden dazu zurückgegriffen werden. Die Verifikation beschreibt die Untersuchung eines Systems, in Hinblick darauf, ob die gestellten Anforderungen erfüllt sind. Dies umfasst dabei ebenso den Prozess der Entwicklung selbst wie die Konformität zu Regularien entlang des Lebenszyklus. Dadurch stellt die Verifikation sicher, dass ein System korrekt erstellt wurde. Die Validierung hingegen beschreibt die Untersuchung des Systems entlang von gestellten Anforderungen an die Nutzung. Damit stellt die Validierung die Nutzbarkeit des Systems zur Aufgabenbearbeitung sicher. [TECH17]

BANKS unterteilt in der Taxonomie die Techniken in informelle, statistische, dynamische und formale Methoden. Die vollständige Taxonomie ist in Abbildung A.2 dargestellt [BANK98]. Während diese Taxonomie zunächst den Fokus auf Simulationen legt, so ist die Taxonomie grundsätzlich einsetzbar. Für den Bereich der technischen Systeme und Software bestehen für die Methoden tiefergehende Beschreibungen und Handlungsempfehlungen durch die IEEE als IEEE 1012 Standard [Soft17]. In dieser Arbeit werden ausgewählte Methoden aus dem IEEE 1012 Standard, genauer im Teilbereich Software, genutzt. Dabei ist die Betrachtung des Lebenszyklus reduziert, sodass Phasen wie Wartung und Entfernung nicht Teil dieses Kapitels sind.

Die Verifikation und Validierung geschieht in mehreren Schritten. Im Kapitel 4 wurden bereits Methoden angewendet zur Verifikation der Umsetzung und Implementierung des Synchronisierungssystems. Diese umfassten insbesondere Methoden zur Daten- und Kontrollflussanalyse. Ebenso wurden Schnittstellen für Menschen und Automatisierung gemäß dem Vorgehen in IEEE 1012 entworfen und überprüft. Weitere Arbeiten zur Verifikation finden sich in Abschnitt 3.3 und bilden die Grundlage zur Bewertung der Einhaltung der Anforderungen.

Die weiteren Schritte der Verifikation und Validierung finden sich in diesem Kapitel. Der Abschnitt 5.1 beschreibt die Anwendung von dynamischen Methoden wie den Eingangstests. Weiterhin werden die Anforderungen aus dem Abschnitt 3.2 genutzt für eine Bewertung des entwickelten Systems, ob der Einhaltung eben jener Anforderungen. Der Abschnitt 5.2 beschreibt die darauf folgende Validierung. Darin wird die Parametrierung des Systems selbst, sowie der Aufbau der Validierung dargestellt. Basierend darauf werden Szenarien beschrieben, welche darauf folgend in Versuchsreihen ausgewertet werden. Diese Auswertungen geschehen für das Synchronisierungssystem ebenso wie für die anderen Methoden,

die in Abschnitt 3.3 dargestellt wurden. Dies entspricht den dynamischen Methoden der Taxonomie. Abschließend findet eine Diskussion und Einordnung der Ergebnisse der Verifikation und Validierung statt, inklusive der Implikationen für die Forschungsfragen aus Abschnitt 1.2.

5.1 Verifikation

Verification

Bei der Verifikation des Systems kommen vier Methoden zum Einsatz. Zunächst wird ein Eingangstest des Synchronisierungssystems durchgeführt. Weiterhin wird die Integration in eine Produktions-Systemlandschaft sowie die theoretische Komplexität der Lösung betrachtet. Zuletzt findet die Bewertung der gestellten Anforderungen an das Synchronisierungssystem statt.

5.1.1 Eingangstests

Input Testing

Zur Charakterisierung des Systems und seiner Verhaltensweisen bei veränderlichen Datenlängen sind Eingangstests genutzt. Hier ist ein Szenario definiert, bei dem die Anzahl der Datenpunkte verschiedene Ausprägungen annimmt. Dazu existieren zwei Zeitreihen, welche mit jedem Versuch länger werden, um eine halbe Zehnerpotenz, angefangen bei 100 Werten. Die Veränderung der Rechenzeit des Systems und der Synchronisierungsmethoden in Abhängigkeit von der Länge der Eingaben ist Untersuchungsgegenstand. Auch Abbrüche aufgrund zu hoher Ressourcenbedarfe (Speicherplatz, Rechenkapazitäten, Zeit) sind betrachtet. Dabei findet einerseits ein Vergleich mit den anderen Methoden statt und andererseits eine tiefere Betrachtung der neuen Methode in Form einer Aufteilung in die verschiedenen Hauptschritte gemäß Abbildung 4.1.

In Diagramm 5.1 sind die verschiedenen benötigten Zeiten der einzelnen Methoden für die steigenden Datenlängen sichtbar. Bei den meisten Methoden steigt mit der Datenlänge auch die benötigte Rechenzeit. Bei den neuen Methoden hingegen sind die Ausführungszeiten bei 10.000 Werten im Bereich von knapp drei Minuten, fallen bei mehr Daten allerdings auf unter zehn Sekunden zurück. Dies ist auf das Verhalten der Bewertung zurückzuführen, wonach die Methodenbewertung sich auch mit der Datenmenge verändert. Dies zeigt, dass diese Grenzwerte in der Praxis auch die Entscheidung und angewandte Methode anpassen. Bei DTW fehlen die Angaben für die zwei größten Fälle. In diesen beiden Fällen

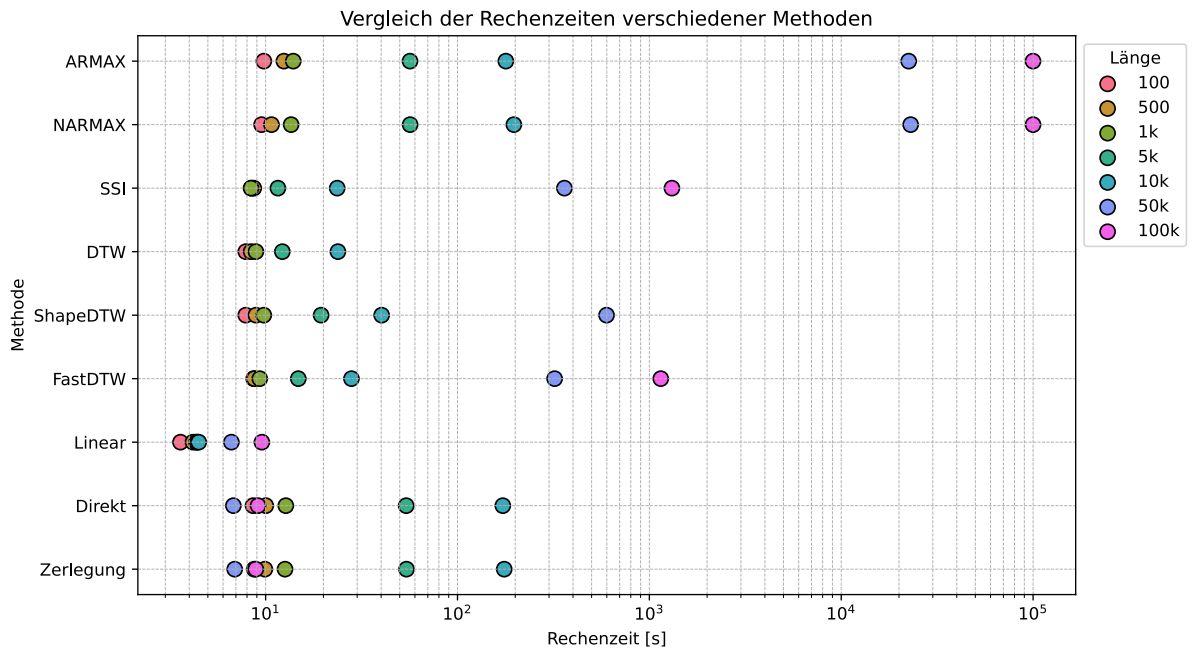


Abbildung 5.1: Darstellung der benötigten Zeiten bei veränderlicher Datenlänge. Fehlende Einträge bedeuten einen Fehlschlag bei der Durchführung.

brach die Ausführung ab mit einem Speicherfehler, da die Erstellung der Matrix innerhalb der Methode mehr Speicher benötigte als Python zu verwalten vermochte. Ebenfalls nicht erfolgreich war die letzte Ausführung von *ShapeDTW*. Dadurch, dass die Werte zufällig erstellt wurden, erstellte diese Methode Sprungdistanzen um ein Vielfaches der eigentlichen Länge der Zeitdatenreihe, sodass die Berechnungen intern letztlich fehlschlagen. Damit schlug die gesamte Ausführung fehl. Bei den neuen Methoden hingegen kam es zu keinen Fehlern, sodass die formale Ausführung für alle Längen erfolgreich umsetzbar ist. In Bezug auf die Dauer der Ausführung ist die Methode der *Linearisierung* erwartungsgemäß die schnellste Methode und bleibt konstant unterhalb der zehn Sekunden. Hingegen ist der größte Auswuchs erkennbar bei den Methoden *ARMAX* und *NARMAX*, welche im letzten Fall deutlich mehr als einen Tag für die Ausführung benötigen.

Zur Betrachtung der Aufteilung der Rechenzeiten der neuen Methodik dient die gestapelte Darstellung in Diagramm 5.2. In allen Fällen ist der Schritt 1 (hellgrün) ein signifikanter Anteil der Gesamtzeit. Im Detail ist hier der Import aus Dateien mit anschließender syntaktischer Vorverarbeitung der dominante Anteil innerhalb dieses Schritts. Da für diese Versuche eine Textdatei genutzt wurde mit anschließendem Parsing, kann sich diese Zeit verringern lassen

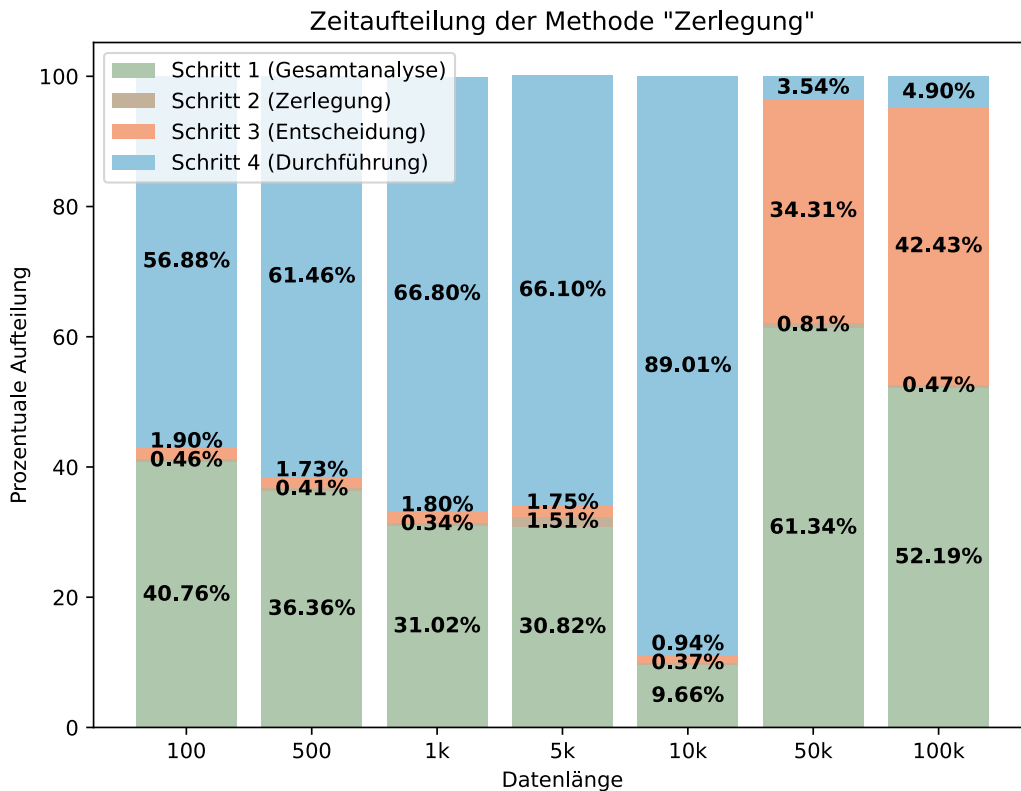


Abbildung 5.2: Darstellung der benötigten Zeiten bei der Methode *Zerlegung*, aufgeteilt in die Hauptschritte des Gesamtsystems. Die Prozentwerte summieren sich nicht immer auf exakt 100 % aufgrund von Rundungseffekten.

durch die Nutzung geeigneter Datenbanken und Speichertechnologien. Im vorliegenden Fall wurden dennoch Dateien gewählt um Varianzen und Caching-Effekte der Datenbanken und des Netzwerks auszuschließen. In den ersten Fällen wählt die Methodik mit *ARMAX* eine Synchronisationsmethode aus, welche eine signifikante Ausführungszeit (Schritt 4, blau) bedeutet. Mit ansteigender Menge von Daten steigt auch die Dominanz dieses Schritts entsprechend. Ab dem vorletzten Schritt wählt die Methodik allerdings aufgrund der erwarteten Berechnungszeit eine andere Methode aus, sodass die eigentliche Ausführung deutlich geringer ausfällt. Im Umkehrschluss steigen damit die Importe und auch die Bewertung (Schritt 3, orange) deutlich an im Anteil. Im Detail fällt dies zurück auf die Berechnung der Vergleichsmetriken zwischen den Zeitdatenreihen. Erkennbar ist hier ebenfalls, dass die Zunahme des Anteils bei dem Schritt der Bewertung am stärksten

ist. Lediglich der Schritt der Fixpunktsuche und Zerlegung (braun) bleibt in allen Fällen verhältnismäßig gering.

Insgesamt zeigen die Eingangstests, dass die Entscheidung auf Basis von Eigenschaften der Daten funktioniert und zu anderen Ergebnissen führt. Während alle Methoden mit der Erhöhung der Datenmenge monoton erhöhte Rechenzeiten benötigen, kann die neue Methodik hier durch Veränderung der Auswahl von Methoden entgegenwirken und zugleich in allen Fällen zu einer erfolgreichen Ausführung kommen. Bei der genaueren Betrachtung der dominanten Schritte innerhalb des Synchronisierungssystems lassen sich drei Einzelaufgaben identifizieren. Sowohl die syntaktische Vorverarbeitung, die Berechnungen von Vergleichsmetriken und die finale Ausführung der Methoden sind dabei primär in ihrer Dauer von der Menge von Daten abhängig.

5.1.2 Integrationstest

Integration Test

Für die Nutzung des Systems in einer Produktion und zur Skalierung der Anwendung ist die Integration in bestehende Produktionsnetzwerke entscheidend. Die Untersuchung der Integrierbarkeit erfordert daher die Erweiterung einer Referenzarchitektur um das hier entwickelte Offline Synchronisierungssystem und die Betrachtung der Kontroll- und Datenflüsse dort. Die Basis für diese Integrationsbetrachtung stellt hier die Architektur des Produktionsnetzwerks der Fraunhofer FFB dar. Für diese Netzwerkarchitektur wurden in der Planungsphase die Anforderungen aufgenommen, das Netzwerk ausgelegt [MOHR25] und in die Fabrikplanungsphasen aufgenommen [RUBE25]. Das Netzwerk ist in der Praxis zudem unterteilt in Entwicklungs- und Produktivbereiche, sodass das dargestellte Offline Synchronisierungssystem bei Fehlverhalten keine Ausfälle im Produktivbereich und der Produktionslinien verursachen darf [TECH22b].

Die Integrationsstrategie ist in Abbildung 5.3 dargestellt. Kernpunkt ist die Einführung des eigentlichen Offline Synchronisierungssystems. Dabei kann es eine virtuelle Maschine auf Windows oder Unix-Basis im Netzwerk sein mit den entsprechenden Laufzeitumgebungen. Der Kontrollfluss startet für die Integration stets außerhalb des Offline Synchronisierungssystems. Die IIoT-Plattform oder ein externer Dienst erstellt eine Anfrage und sendet einen entsprechenden API-Aufruf als REST-Call an das Offline Synchronisierungssystem. Im Falle der Fraunhofer FFB kommt für die IIoT-Plattform die Software »WinCC OA« der Firma Siemens zum Einsatz. Die zentrale Instanz dieser Software ist in der Lage, entsprechende REST-Calls zu erstellen und abzusenden. Bei der Ausführung stellt das Offline Synchronisierungssystem eine Verbindung zu den Datenbanken für Zeitdatenreihen, Informationen und Wissen her.

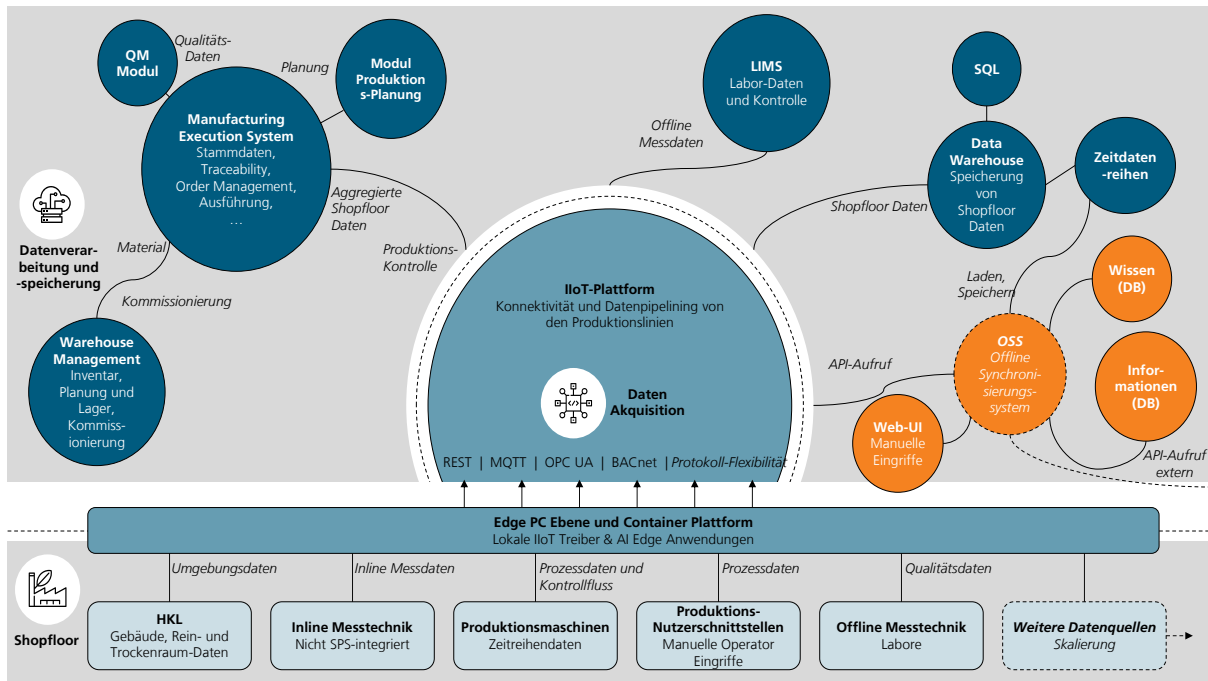


Abbildung 5.3: Darstellung des Aufbaus der IT-Architektur für das Produktionsnetzwerk der Fraunhofer FFB. Die orangen Bereiche stellen Erweiterungen der Architektur für das Offline Synchronisierungssystem dar. (Basierend auf [MOHR25] - Die Größe der Kreise hat keine Bedeutung.)

Für Informationen und Wissen (z.B. die Ursache-Wirkzusammenhangs-Datenbank analog zur SQL-Implementierung in Abschnitt 4.2.3) findet dabei ein ausschließlich lesender Zugriff statt. Für die Zeitdatenreihen findet bei erfolgreicher Synchronisierung auch ein schreibender Zugriff statt. Dabei werden die alten Daten erhalten, lediglich eine alternative Version der Zeitdatenreihen abgelegt. Dies erhöht den Platzbedarf, stellt aber sicher, dass keine Daten unwiederbringlich verloren gehen. Sollte im API-Aufruf die Nutzung von menschlichen Eingriffen vorgegeben sein, so stellt das Offline Synchronisierungssystem die entsprechenden Web-UIs (siehe Abschnitt 4.2.4) bereit im Netzwerk.

Da der Kontrollfluss stets von anderen Systemen ausgeht und keine Daten überschrieben werden, kann das Offline Synchronisierungssystem nicht selbstständig agieren oder direkten Schaden verursachen. Die einzige Einschränkung in Bezug auf die Sicherheit ist die eingeschränkte Bereinigung von Eingangsparametern. So kann ein Aufruf des Offline Synchronisierungssystems mit präparierten Angaben zu den Datenbanken der Zeitdatenreihen theoretisch zu einem Abfluss von Daten aus einer Datenbank in eine andere Datenbank führen. Das Offline Synchronisierungssystem überprüft die Datenbanken nicht auf Freigaben oder Einschränkungen im Bereich der Informationssicherheit. Um hier die

vollständige Sicherheit zu gewährleisten, sollte bei der Integration die Kommunikation des zugrunde liegenden Systems über Firewall-Regeln eingeschränkt werden, sodass kritische Datenbanken gar nicht erst erreichbar sind. Insgesamt lässt sich festhalten, dass mit diesem Integrationskonzept das Offline Synchronisierungssystem manuell und automatisch nutzbar ist im Produktionsnetzwerk der Fraunhofer FFB und zugleich ein relevantes Maß an Sicherheit sichergestellt ist. Basierend auf testweisen Ausführungen kann daher die Integration als erfolgreich verifiziert angesehen werden.

5.1.3 Komplexität

Complexity

Zur Betrachtung der Komplexität dient der Aufbau des Synchronisierungssystems in den einzelnen Schritten, wie in Abbildung 4.1 unterteilt. Diese Schritte bilden in Summe die Komplexität des Gesamtsystems. Dabei haben die Schritte teilweise einen Einfluss aufeinander. Daraus ergibt sich eine Komplexitätsbetrachtung für jeden Schritt, bei dem die Ergebnisse der vorherigen Schritte als Parameter dienen. Daraus folgt

$$\mathcal{O}(OSS_N) = \max(\mathcal{O}(OSS_N^{S1}), \mathcal{O}(OSS_N^{S2}), \mathcal{O}(OSS_N^{S3}), \mathcal{O}(OSS_N^{S4})) \quad (5.1)$$

$$N = \{N_3, N_z, N_C, N_\Gamma, N_{rel}, N_\Xi\}, \quad (5.2)$$

wobei N die Menge aller Parameter für die Komplexität angibt. So ist N_3 die Anzahl der Zeitdatenreihen und N_z die Gesamtzahl aller Zeitreihenelemente. Diese stellen die Stellgrößen aus dem Bereich Daten dar. Für den Bereich Informationen beschreibt N_C die Anzahl aller Systeme im Produktionsnetzwerk. Hingegen beschreibt N_Γ die Anzahl der bekannten Ursache-Wirkzusammenhänge bei der Ausführung, und analog dazu ist die Anzahl der bekannten Relationen durch N_{rel} gegeben. Da Fixpunkte innerhalb des Systems erstellt oder von außen (manueller Eingriff) definiert sein können, ist die Anzahl aller Fixpunkte durch N_Ξ gegeben. Aus der Definition von Zeitdatenreihen und Fixpunkten ergeben sich zusätzlich

$$N_\Xi \ll N_z, \quad N_3 \ll N_z. \quad (5.3)$$

Innerhalb des Schritt 1 bei der Komplexitätsanalyse (OSS_N^{S1}) beginnt die Verarbeitung mit

dem Aufbau von G_{sync} . Die Menge aller Systeme im Netzwerk N_C ist dabei ausschlaggebend. In der ersten Version entsteht ein vollständiger Graph, was zu einem quadratischen Wachstum mit N_C führt. Nach dem Aufbau des G_{sync} findet die syntaktische Vorverarbeitung statt. Die Implementierung geht hier durch jedes Element aus $z_i^l \in \mathfrak{Z}$ einmalig durch und führt die syntaktische Verarbeitung durch, wobei syntaktisch fehlerhafte Werte korrigiert werden und ein einheitliches Zielformat im Offline Synchronisierungssystem erreicht wird. Damit ist dieser Schritt direkt linear von der Anzahl der Zeitreihenpunkte in N_z abhängig. Der letzte Teilschritt ist die Ableitung von Eigenschaften der einzelnen Zeitdatenreihen. Die Berechnung der Eigenschaften umfasst mehrere einzelne Berechnungen. Die Berechnung von statistisch allgemeinen Informationen wie Frequenz, Minimum, Maximum, Varianz und Standardabweichung kann dabei jeweils linear in Abhängigkeit der jeweiligen Menge von Elementen der Zeitdatenreihe berechnet werden. Zur Berechnung von Schiefe (2.16) und Wölbung (2.17) existieren ebenfalls lineare Berechnungen. Im Detail ist die Komplexität bei diesem Teilschritt abhängig von dem Produkt aus Anzahl der Zeitdatenreihen und ihrer Länge, da alle Berechnungen auf den Zeitdatenreihen pro Zeitdatenreihe einmalig durchgeführt werden. Dies kann zusammengefasst werden zu einer Abhängigkeit von N_z . Daraus ergibt sich eine Komplexitätsbetrachtung im Schritt 1 zu

$$\begin{aligned} \mathcal{O}(OSS_N^{S1}) &= \mathcal{O}(N_C^2) + \mathcal{O}(N_z) + \mathcal{O}(N_3) \\ &\stackrel{Gl.5.3}{=} \mathcal{O}(N_C^2) + \mathcal{O}(N_z) \\ &= \mathcal{O}(N_C^2 + N_z). \end{aligned} \tag{5.4}$$

Im Hauptschritt 2, der Zerlegung, findet zuerst die Identifikation von Fixpunkten statt. Hierzu wählt der Fixpunktscanner in der Implementierung alle Paare von Zeitdatenreihen aus. Da die Reihenfolge nicht relevant ist ergeben sich $\binom{N}{2}$ Kombinationen. Für jede Kombination ist jede einzelne Regel der Ursache-Wirkzusammenhänge N_T zu betrachten. Alle Identifikationen durchlaufen dabei die Länge der Zeitdatenreihe entsprechend.

Der nächste Teilschritt befasst sich mit der Zerlegung in Sequenzen. Hierzu ergibt sich zunächst die Bestimmung N_{Ξ} aus dem Ergebnis der Fixpunkt-Identifikation. Die eigentliche Zerlegung einer Sequenz in mehrere Subsequenzen kann linear erfolgen in Abhängigkeit der Länge eben der Sequenzen. Die Ableitung von Eigenschaften der Subsequenzen geschieht analog zu den Berechnungen der Eigenschaften in Hauptschritt 1 und kann als linear betrachtet werden. Für den Hauptschritt 2 ergibt sich die Abschätzung der Laufzeit damit zu

$$\begin{aligned}
\mathcal{O}(OSS_N^{S2}) &= \mathcal{O}\left(\binom{N_3}{2} \cdot N_\Gamma \cdot N_z\right) + \mathcal{O}(N_\Xi \cdot N_z) + \mathcal{O}(N_\Xi \cdot N_z) \\
&= \mathcal{O}\left(\frac{N_3 \cdot (N_3 - 1)}{2} \cdot N_\Gamma \cdot N_z\right) + \mathcal{O}(N_\Xi \cdot N_z) + \mathcal{O}(N_\Xi \cdot N_z) \\
&= \mathcal{O}(N_3^2 \cdot N_\Gamma \cdot N_z) + \mathcal{O}(N_\Xi \cdot N_z) + \mathcal{O}(N_\Xi \cdot N_z) \\
&= \mathcal{O}(N_3^2 \cdot N_\Gamma \cdot N_z + N_\Xi) \\
&\stackrel{\text{Gl. 5.3}}{=} \mathcal{O}(N_3^2 \cdot N_\Gamma \cdot N_z).
\end{aligned} \tag{5.5}$$

Der dritte Hauptschritt im Synchronisierungssystem ist der Schritt der Entscheidung. Dieser startet mit der Bewertung aller Methoden. Diese Bewertung geschieht entlang aller Kanten G_{sync} , wobei für jede Kante erst die Eigenschaften im Vergleich der Zeitdatenreihen durchgeführt wird und dann die Bewertung pro Methode. In diesem Punkt dominiert die Berechnung der Metriken zwischen den Zeitdatenreihen. Bei den Korrelationsmaßen verhält sich die Pearson-Korrelation linear ($\mathcal{O}(N_z)$), während Spearman eine schlechtere Laufzeit aufweist ($\mathcal{O}(N_z \log N_z)$). Die Kreuzkorrelation kann direkt berechnet werden und geht mit einer quadratischen Laufzeit einher. In diesem Synchronisierungssystem ist stattdessen eine optimierte Implementierung gewählt, die mittels Fast Fourier Transformation eine vergleichbare Laufzeit wie Spearman erreicht. Ebenfalls kommt die Fast Fourier Transformation bei der spektralen Distanz zum Einsatz und führt zu der gleichen vergleichbaren Laufzeit. Die Berechnung der Fréchet-Distanz ist bei direkter Anwendung ebenfalls in quadratischer Laufzeit möglich, in der hier genutzten Implementierung kommt daher eine Approximationsvariante zum Einsatz, die die Laufzeitkomplexität quasilinear macht. Für die Berechnung der Normalized Mutual Information kann ebenfalls eine lineare Laufzeit angenommen werden. Bei direkter Anwendung des Pattern Matching Index kommt ein Schiebefenster-Ansatz zum Einsatz, was zu einer quadratischen Laufzeit führt. In der vorliegenden Implementierung findet erneut eine Optimierung mittels Fast Fourier Transformation statt, welche die Laufzeit quasilinear werden lässt. Insgesamt ergibt sich für die Berechnung der Eigenschaften damit zweimal eine lineare und fünfmal eine quasilineare Laufzeit in Bezug auf N_z .

Für den weiteren Verlauf kommen Graphalgorithmen auf G_{sync} zum Einsatz. Für alle mit der Senke verbundenen Knoten wird ein minimaler In-Tree berechnet. Die Berechnung eines In-Tree mit den Algorithmen von Edmonds lässt sich abschätzen zu einem quasilinearen Verhalten in Abhängigkeit von der Menge an Knoten in G_{sync} , also Systemen im

Produktionsnetzwerk N_C . Die Auswahl des In-Tree mit dem geringsten Gewicht ist im Anschluss trivial. Daraus ergibt sich für diesen Hauptschritt eine Laufzeit von

$$\begin{aligned}
 \mathcal{O}(OSS_N^{S3}) &= N_3 \cdot (2 \cdot \mathcal{O}(N_z) + 5 \cdot \mathcal{O}(N_z \log N_z)) + N_C \cdot \mathcal{O}(N_C \log N_C) \\
 &= \mathcal{O}(N_3 \cdot (N_z + N_z \log N_z)) + \mathcal{O}(N_C \cdot N_C \log N_C) \\
 &= \mathcal{O}(N_3 \cdot N_z \log N_z) + \mathcal{O}(N_C^2 \log N_C) \\
 &= \mathcal{O}(N_3 \cdot N_z \log N_z + N_C^2 \log N_C).
 \end{aligned} \tag{5.6}$$

Der letzte Hauptschritt ist die Anwendung. Die Laufzeit ist insbesondere von der Laufzeit der eingesetzten Synchronisierungsmethoden $\mathcal{O}(M(N_z))$ abhängig. Dabei ergibt sich aus der Empfehlung vom vorherigen Schritt die Strategie und Menge von Synchronisierungen. Da die Empfehlung der Rolle eines Spannbaums in G_{sync} entspricht, finden grundsätzlich maximal $N_C - 1$ Synchronisationen statt. Dies bedeutet eine Laufzeit ohne Zerlegungen von $\mathcal{O}(N_C \cdot M(N_z))$. Kommen Zerlegungen zum Tragen, so erhöht sich die Menge von Synchronisierungen um die Menge der Fixpunkte entsprechend, während die Größe des Parameters N_z für die Methoden verringert wird. Dies führt zu einer Laufzeitabschätzung

$$\mathcal{O}(OSS_N^{S4}) = \begin{cases} \mathcal{O}(N_C \cdot M(N_z)), & \text{falls } N_{\Xi} = 0 \\ \mathcal{O}\left(N_C \cdot \min\left(M(N_z), N_{\Xi} \cdot M\left(\frac{N_z}{N_{\Xi}}\right)\right)\right), & \text{falls } N_{\Xi} > 0 \end{cases} . \tag{5.7}$$

In dieser Darstellung ist die Auswirkung der Zerlegung erkennbar. Bei Betrachtung der Komplexität von M ergibt sich

$$\mathcal{O}(OSS_N^{S4}) = \begin{cases} \mathcal{O}(N_C \cdot N_z), & \text{falls } M(X) = \mathcal{O}(X) \\ \mathcal{O}\left(N_C \cdot N_z \log \frac{N_z}{N_{\Xi}}\right), & \text{falls } M(X) = \mathcal{O}(X \log X) \\ \mathcal{O}\left(N_C \cdot \frac{N_z^2}{N_{\Xi}}\right), & \text{falls } M(X) = \mathcal{O}(X^2) \end{cases} . \tag{5.8}$$

Aus dieser Betrachtung ergibt sich, dass die potenzielle Reduktion der Laufzeit bei linearen Methodenlaufzeiten vernachlässigbar ist. Bei Laufzeiten oberhalb der Linearität hingegen ist die Laufzeit in Abhängigkeit von den Fixpunkten verbessert. Auch wenn N_{Ξ} deutlich

kleiner als N_z ist, so ist das Wachstum gegenüber der Laufzeit ohne Fixpunkte signifikant verändert.

Die Gesamtlaufzeit des Synchronisierungssystems ergibt sich damit zu

$$\begin{aligned}
 \mathcal{O}(OSS_N) &= \max(\mathcal{O}(OSS_N^{S1}), \mathcal{O}(OSS_N^{S2}), \mathcal{O}(OSS_N^{S3}), \mathcal{O}(OSS_N^{S4})) \quad (5.9) \\
 &= \max \left(\mathcal{O}(N_C^2 + N_z), \mathcal{O}(N_3^2 \cdot N_\Gamma \cdot N_z), \right. \\
 &\quad \mathcal{O}(N_3 \cdot N_z \log N_z + N_C^2 \log N_C), \\
 &\quad \left. \mathcal{O} \left(N_C \cdot \min \left(M(N_z), N_\Xi \cdot M \left(\frac{N_z}{N_\Xi} \right) \right) \right) \right).
 \end{aligned}$$

Basierend auf den Erkenntnissen aus Abschnitt 5.1.1 und Literatur ergibt sich die Annahme, dass die von der Datenmenge abhängigen Terme dominieren. Da in der Praxis die Menge von Daten auch erwartbar stärker ansteigt als die Menge von Systemen, passt diese Annahme zu dem Ziel der Skalierung als Hauptfokus. Dies führt, ergänzend zu Formel 5.3, zu

$$\begin{aligned}
 N_z &\gg N_C^2, \quad N_z \gg N_\Gamma \quad (5.10) \\
 \Rightarrow \mathcal{O}(OSS_N) &= \max \left(\mathcal{O}(N_z), \mathcal{O}(N_3^2 \cdot N_\Gamma \cdot N_z), \right. \\
 &\quad \mathcal{O}(N_3 \cdot N_z \log N_z), \\
 &\quad \left. \mathcal{O} \left(\min \left(M(N_z), N_\Xi \cdot M \left(\frac{N_z}{N_\Xi} \right) \right) \right) \right) \\
 &= \max \left(\mathcal{O}(N_z \log N_z), \mathcal{O} \left(\min \left(M(N_z), N_\Xi \cdot M \left(\frac{N_z}{N_\Xi} \right) \right) \right) \right). \quad (5.11)
 \end{aligned}$$

Dies bedeutet, dass die Laufzeitkomplexität in erster Linie von der Menge der Datenpunkte abhängt und weiterhin von der Auswahl der Methoden und der Komplexität eben jener Methoden. Die zusätzliche Untersuchung und Zerlegung nach Fixpunkten selbst ist vernachlässigbar, allerdings kann bei erfolgreicher Zerlegung die Laufzeit der Methoden deutlich reduziert werden. Durch Vorauswahl der Methoden und Grenzwerte kann die Laufzeit in

Bezug auf die Methodenkomplexität darüber hinaus künstlich beschränkt werden auf den quasilinearen Anteil.

5.1.4 Evaluierung gegen Anforderungen

Evaluation Against the Requirements

Gemäß den Betrachtungen aus dem Aufbau des Synchronisierungssystems und den Ergebnissen der vorherigen Abschnitte lässt sich eine Evaluierung gegen die Anforderungen aus Abschnitt 3.2 durchführen. Diese Evaluierung findet, falls nicht anders angegeben, für die Variante der direkten Anwendung und die Variante mit Zerlegung gleichermaßen statt.

Für die **Effizienz** zeigt die Bewertung der Komplexität eine dominante Abhängigkeit von der Menge von Datenpunkten. Im Bereich der **kleinskaligen** Betrachtung lässt sich hier ein Effizienzgewinn gegenüber dem manuellen Ablauf feststellen. Die Komplexität selbst kann je nach Auswahl der Methoden allerdings zwischen quasilinearer und quadratischer Laufzeit schwanken. Da bei größeren Datenmengen automatisch Gegenmaßnahmen getroffen sind zur Eindämmung des Aufwuchses ist das Wachstumsverhalten bei großen Datenmengen künstlich beschränkt. Im Sinne der Bewertung ist dies als Stufe 2 gewertet. Analog dazu tritt die gleiche künstliche Beschränkung für **großskalige** Produktionen in Erscheinung. Durch die Auswahl der Reihenfolge der Synchronisationen ist zudem die Menge der durchzuführenden Synchronisierungen beschränkt und es findet keine paarweise Synchronisierung aller Zeitdatenreihen statt. Durch das dargestellte Integrationskonzept entfallen manuelle Eingriffe, sofern nicht explizit konfiguriert. Im Falle der Nicht-Nutzung der manuellen Eingriffe ergibt sich für diese Metrik damit eine Bewertung als Stufe 2.

Bezüglich der **Anwendbarkeit** bewertet das vorgestellte System anhand der vorliegenden Informationen und Wissen, welche Methoden anwendbar sind. Methoden, welche nicht anwendbar sind, werden nicht ausgewählt und mit der Methode der *Linearisierung* besteht stets eine Rückfalloption, bei der eine Anwendung grundsätzlich möglich ist, da sie nicht von Bedingungen der Produktion abhängig ist. Daraus ergibt sich für die Metrik zu den **In Situ Anforderungen** die Bewertungsstufe 3, da zumindest Informationen zum Aufbau der Produktion notwendig sind für die Aufstellung der Vorschläge. Im Falle einer nachträglich hinzugefügten Datenquelle kann auf die bestehenden Projektionen Φ zurückgegriffen werden. Dies bedeutet, dass für die neue Zeitdatenreihe eine Projektion Φ_X auf eine der bestehenden Zeitdatenreihen berechnet wird. Durch Komposition $\Phi \circ \Phi_X$ kann die Synchronisierung der neuen Datenquelle auf alle andere Daten erfolgen. Das bedeutet, dass lediglich ein einziges neues Modell berechnet wird und die weiteren Arbeiten der Anwendung der bestehenden Modelle entspricht. Für die Metrik zur **Erweiterung um**

Datenquellen entspricht dies der Stufe 3. Im Bereich der **Unabhängigkeit** ist die Bewertung erneut von der Auswahl der Methoden abhängig. Durch die Berechnung von Vergleichsmetriken und Nutzung von Wissen kann die Auswahl hierbei verbessert werden. Existieren keinerlei solche Möglichkeiten und die Daten haben keinerlei Zusammenhänge, kommt mit der *Linearisierung* erneut eine Rückfalloption zum Einsatz. Damit ist aber die Durchführung unabhängig von der Existenz der Zusammenhänge grundsätzlich im Synchronisierungssystem gegeben, was einer Bewertung als Stufe 2 entspricht.

Die finale **Qualität** der Synchronisierung hängt hier ebenfalls von den eingesetzten Methoden ab. Die **erwartete Qualität** richtet sich daher nach der schlechtesten erwarteten Qualität der eingesetzten Methoden. In der gegebenen Implementierung ist eine künstliche Obergrenze der Abweichungen und Sprungdistanzen festgeschrieben durch die Parametrierung, sodass eine grobe Grenze existiert. Da diese allerdings von einzelnen Effekten beeinflusst sein kann, ergibt sich eine Bewertung der Stufe 0. Für die **Nicht-Linearität** kann bei der Auswahl von entsprechenden Methoden diese adressiert werden. Auch im Falle der *Zerlegung* dienen Fixpunkte unabhängig als Synchronisierungen unabhängig von linearen Effekten. Daraus ergibt sich eine Bewertung für beide Varianten als Stufe 2. Zuletzt kann das Synchronisierungssystem Prüfungen der **Plausibilität** durchführen und entsprechende Regeln bei der Auswahl der Methoden einbeziehen. Allerdings gibt das System in der Form keine Garantien zur Einhaltung der Regeln oder Berücksichtigung innerhalb der einzelnen Methoden. Dies führt zu einer Bewertung als Stufe 1.

		Effizienz		Anwendung			Qualität		
		Klein-skalig	Groß-skalig	In situ	Erweiter-barkeit	Unabhängige Messungen	Erwartung	Nicht-Linearität	Plausibilitäts-prüfungen
	Direkt	●	●	◐	●	●	○	●	◐
	Zerlegung	●	●	◐	●	●	○	●	◐

Tabelle 5.1: Die Bewertung der neuen Methodik gegen die Anforderungen an die Offline Synchronisierung von Produktionsdaten aus der Batteriezellfertigung.

Insgesamt lässt sich feststellen, dass die Anforderungen durch die neue Methodik weitgehend erfüllt sind. Dies zeigt auch die tabellarische Darstellung 5.1. Abzüge gibt es einerseits bei der Notwendigkeit von Informationen zum Produktionsaufbau, welche eine notwendige Bedingung für die Durchführung darstellt. Andererseits gibt es stärkere

Einschränkungen bei der Qualität, da hier keine Garantien gegeben sind, insbesondere aufgrund der Abhängigkeit von letztlich eingesetzten Synchronisierungsmethoden.

5.2 Validierung des Synchronisierungssystems

Validation of the synchronization system

Die Validierung des Synchronisierungssystems findet durch die Nutzung dynamischer Methoden statt. Hierzu sind Ausführungstests mit kontrollierten Umgebungen, Stör- und Eingangsgrößen genutzt. Kern dieser Validierung ist daher die Nutzung von Szenarien und ihrer Auswertung. Diese Szenarien sind zunächst detailliert beschrieben inklusive der Testumgebung und der Variation der Größen. Im Anschluss findet die Auswertung der Ausführungstests inklusive der Beschreibung entsprechender Metriken statt.

5.2.1 Vorgehen für die Validierung

Approach for Validation

Für die Validierung des Synchronisierungssystems wurden mehrere Szenarien definiert. Die Szenarien sind dabei an historische Produktionsdaten angelehnt oder aus diesen extrahiert. Die Datensätze sind hierbei als Grundwahrheit angenommen und gelten als synchronisiert. Als Störparameter dienen daher synthetische Asynchronizitätseffekte, welche basierend auf den Mustern von Untersuchungen bestimmt sind. Mittels einer statistischen Versuchsplanung lassen sich so fundierte Aussagen für die Validierung zum Verhalten und Güte des untersuchten Systems treffen. [ERIK00]

Modellierung der Asynchronizitätseffekte

Da für die Szenarien künstlich Asynchronizitätseffekte eingebracht werden, welche realen Effekten entsprechend sollen, werden zunächst die entsprechenden Effekte untersucht. Dazu dient der Aufbau für die Charakterisierung von Systemen, dargestellt in Abbildung 4.4.

Durch Nutzung von mehreren Produktionsanlagen und Zeiträumen konnte so eine Analyse von Asynchronizitätseffekten durchgeführt werden. Eine solche exemplarische Auswertung ist in Abbildung 5.4 dargestellt. In dieser ist die Referenzzeit (Präzisionsuhr) als konstante Nulllinie dargestellt. Die Zeitangaben aus der Maschine (PLC) sind in rot dargestellt. Im verbundenen Edge Computer wurde die Zeit separat mit dem Standard API-Call für die

Uhrzeit (grün) sowie der Windows Kernel Zeit (blau) aufgenommen. In der Darstellung kann der Drift der Systeme erkannt werden. Dabei ist der Drift der beiden Geräte gegenläufig und beide Systeme enthalten eine nicht-lineare Komponente sowie Jitter und Ausreißer, insbesondere bei den Angaben der Zeiten aus der PLC. Derartige Darstellungen und Auswertungen wurden für mehrere Anlagen und Betrachtungszeiträume durchgeführt. Aus den Auswertungen konnte so die Menge von Effekten für die Szenarien bestimmt werden. [SCHM25]

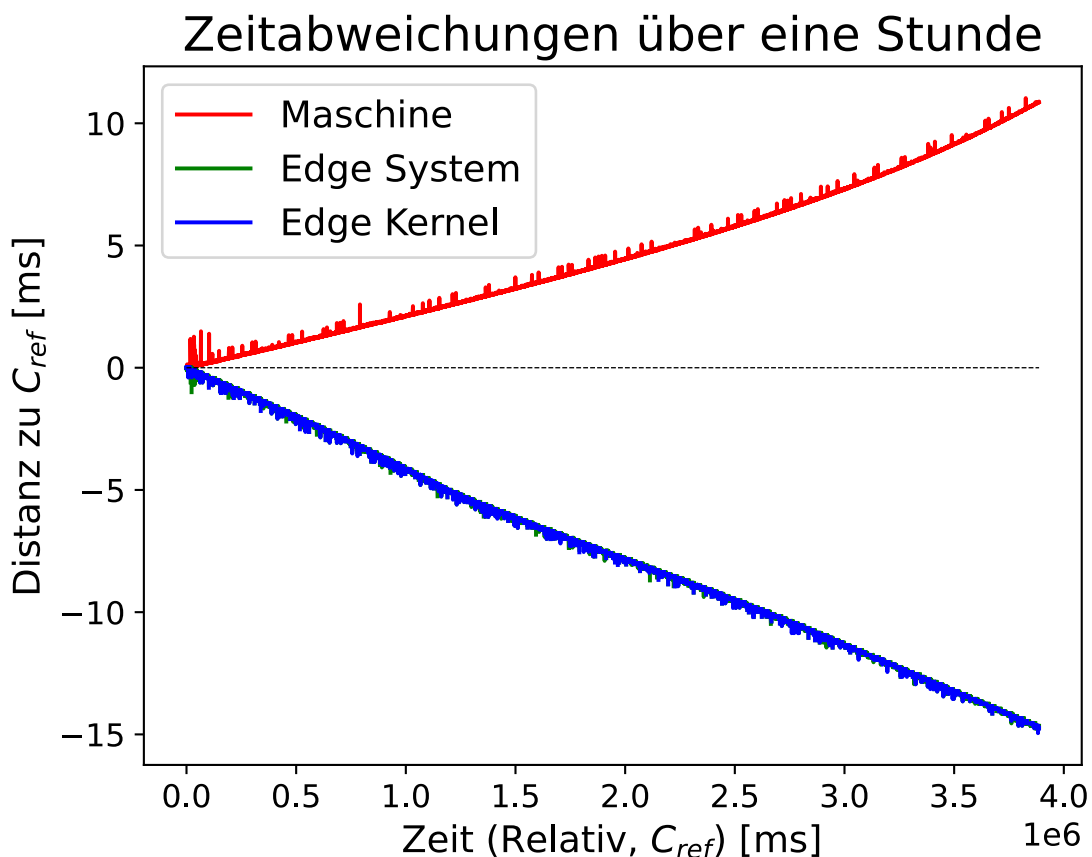


Abbildung 5.4: Darstellung der Systemzeitverläufe von Edge Computer (blau, grün) und Maschine (rot) in relativ zu einer Präzisionsuhr als Referenz (Nulllinie). (Übersetzt aus [SCHM25])

Insgesamt wurden fünf verschiedene Muster festgestellt, welche in Abbildung 5.5 exemplarisch dargestellt sind. Diese Muster lassen sich unterscheiden in Mikro Muster und Makro Muster. Mikro Muster beschreiben wiederkehrende Muster, welche in Zeitbereichen unterhalb einer Sekunde auftreten und einzelne Zeitwerte betreffen. So kommt es zu einer Treppenburg (Mikro Muster 1) bei den Zeitwerten und allgemeinem Jitter und

Wiederkehrende Muster der Abweichungen

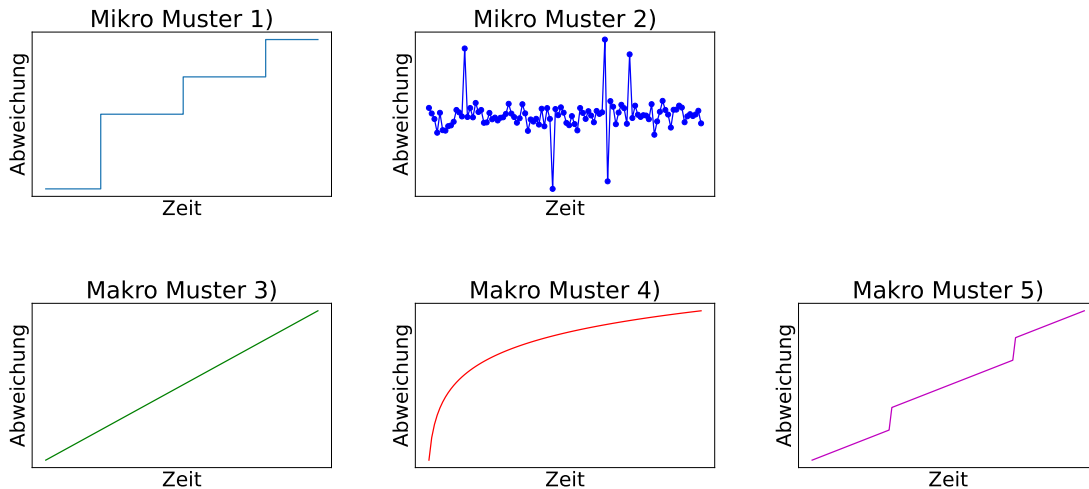


Abbildung 5.5: Abgeleitete wiederkehrende Muster der Asynchronizität in Zeitreihen. Unterteilt in Mikro Muster, welche einzelne Punkte oder Zeitbereiche unterhalb einer Sekunde betreffen, und Makro Muster, welche über längere Zeiträume über die gesamte Zeitreihe auswirken. (Übersetzt aus [SCHM25])

Ausreißern (Mikro Muster 2). In der Konsequenz gilt für Zeitreihen mit diesen Mustern die Annahme der strengen Monotonie und der Eindeutigkeit eines Zeitwerts nicht mehr. Makro Muster hingegen beschreiben langfristige Muster, welche über mehrere Sekunden bis Tage erkennbar sind. Das zentrale und dominierende Muster ist der lineare Drift (Makro Muster 3), welcher schnell auftritt und nachhaltig die gesamte Zeitreihe beeinflusst. Bei längeren Zeitreihen wurde neben dem linearen Drift zudem eine nicht-lineare Komponente (Makro Muster 4) identifiziert. Diese Komponente ist in der Regel schwächer als der lineare Drift, aber über die gesamte Zeitreihe erkennbar und verändert den Drift in den Zeitwerten. Zuletzt kann durch Ereignisse (z.B. eine versuchte Synchronisierung des Systems, ein manueller Eingriff oder Verzögerungen in der Pipeline) ein sprunghafter Anstieg der Zeitwerte entstehen (Makro Muster 5). Bei diesen Makro Mustern ist die Monotonie der Zeitwerte selbst eingehalten, die Abstände zwischen den einzelnen Zeitwerten kann allerdings deutlich variieren.

Für die Szenarien zur Validierung dienen diese Mikro und Makro Muster als Störgrößen

für die Variation. Stufen, Ausreißer, Jitter und Sprünge sind dabei dreistufig von *keinem Effekt* bis *starkem Effekt* implementiert. Für den Drift existiert eine zusammengefasste Störmodellierung. Hier sind lineare Drifts von *nicht vorhanden* bis *stark* ohne nicht-lineare Komponente (Stufen 0 bis 2) und mit nicht-linearer Komponente (Stufen 3 bis 5) verfügbar.

Versuchsplanung

Die Planung der Versuche basiert auf der statistischen Versuchsplanung [ERIK00]. Zunächst wird eine Reihe von Szenarien erstellt, die verschiedene realistische Umgebungen, Aufbauten und Daten abbilden. Innerhalb dieser Szenarien werden gezielt Störgrößen variiert, um das Verhalten des Untersuchungsgegenstands zu erfassen.

Innerhalb der Szenarien wird das Synchronisierungssystem in der Variante mit Zerlegung und ohne Zerlegung untersucht, ebenso zum Vergleich die weiteren vorgestellten Methoden aus Kapitel 3. Eine vollständige faktorielle Versuchsplanung, die alle Methoden mit sämtlichen Variationen der Störgrößen kombiniert, würde zu einer sehr hohen Anzahl an Versuchen führen. Zur Reduktion der Menge der benötigten Versuche kommt daher eine verschränkte Versuchsplanung zum Einsatz mittels einer Taguchi-Matrix. Dieser Ansatz reduziert die notwendigen Versuche deutlich, wobei die grundsätzliche Aussagekraft hoch bleibt [ISLA16].

Die detaillierte Versuchsmatrix für jedes der Szenarien basierend der statistischen Versuchsplanung ist in Tabelle 11 dargestellt. Die Verteilung der Variationen ist dabei ausgewogen, sodass eine bestmögliche Abdeckung der Störgrößen erreicht ist.

Zur Wahrung der Vergleichbarkeit fanden alle Versuche auf demselben Computersystem statt. Es handelt sich um eine Maschine mit Windows Server 2019, Intel Xeon (Skylake) mit acht Kernen zu je 2,20 GHz und 800 GB RAM. Die Experimente erstreckten sich über einen Zeitraum von zwei Monaten.

5.2.2 Beschreibung der Szenarien

Description of Scenarios

Die Szenarien sind an historischen Produktionsläufen entlehnt, sodass ein Praxisbezug sichergestellt ist bei gleichzeitiger Reduktion von unkontrollierbaren und unbekannten externen Störgrößen.

Jedes Szenario umfasst dabei stets ein zugrunde liegendes Produktionsnetzwerk (C_n), inklusive der Definition von Sensoren ($\mathbb{S}$), Gateways und Senken und der Datenflüsse ($C_n \succ \dots \succ_I Z$). Dazu gehört auch die Information, welches System welche Effekte und Daten aufnimmt ($\mathbb{E}, s \perp E$). Neben den Informationen sind auch die Daten und ihre Verläufe selbst ($I Z \in \mathfrak{Z}$) Teil der Szenarien. Optional kann auch das zur Verfügung stehende Wissen (Γ) für die Ursache-Wirkzusammenhänge angegeben sein.

Im Folgenden befindet sich eine Kurzbeschreibung zu jedem der Szenarien, welche in Abschnitt 5.2.4 gemäß der Versuchsplanung durchgeführten Versuche ausgewertet sind. Alle Darstellungen der Signalverläufe entsprechen dabei der idealen synchronisierten Situation, welche vor jedem Durchlauf für alle Sensoren verändert wird.

Die Auswahl der Szenarien ist dabei derart getroffen, dass verschiedene Eigenschaften, Eigenarten und Gegebenheiten abgedeckt sind. Dadurch kann ein breites Spektrum abgedeckt werden, wodurch die Vergleichbarkeit und Aussagekraft der Ergebnisse erhöht wird. Besonderheiten eines Szenarios, welche bei anderen Szenarien nicht auftreten sind in der Beschreibung am Ende zusammengefasst.

Szenario 1

Das Szenario 1 beschreibt einen kurzen Ablauf in einem Produktionsaufbau. Die Maschinensteuerung erfasst die Geschwindigkeit, was im Kontext des Produktionsnetzwerks als Sensor gilt ($C_1 \succ_{\text{Web_Speed}} Z$). Ein Sensorsystem C_2 nimmt die Dicke von zwei Seiten auf ($C_2 \succ_{\text{Thickness1}} Z, C_2 \succ_{\text{Thickness2}} Z$). Zuletzt existiert das Temperaturmesssystem für die Trocknung ($C_3 \succ_{\text{Drying_Temperature}} Z$). Alle Sensoren sind direkt mit der Senke C_{senke} verbunden, also existieren keine Gateways. Die Signalverläufe sind in Darstellung 5.6 aufgetragen. Die mittlere Frequenz der Zeitreihen liegt bei 10 Hz. Die integrierten bekannten Effekte beinhalten mehrere lineare Zusammenhänge, insbesondere zwischen der Geschwindigkeit und der Dicke. Das bereitgestellte Wissen umfasst Domänenwissen, dass es hier lineare Zusammenhänge gibt, allerdings keine detaillierten Angaben und keine Ursache-Wirkzusammenhänge. Insgesamt umfasst das Szenario achttausend Datenpunkte.

Besonders an diesem Szenario ist die Existenz von verzögerten Ausbrüchen in gegensätzliche Richtungen, während andere Zeitdatenreihen fast identisch verlaufen. Beim Aufbau ist ein sehr einfacher Aufbau genutzt, sodass Übertragungsverzögerungen weniger relevant sind.

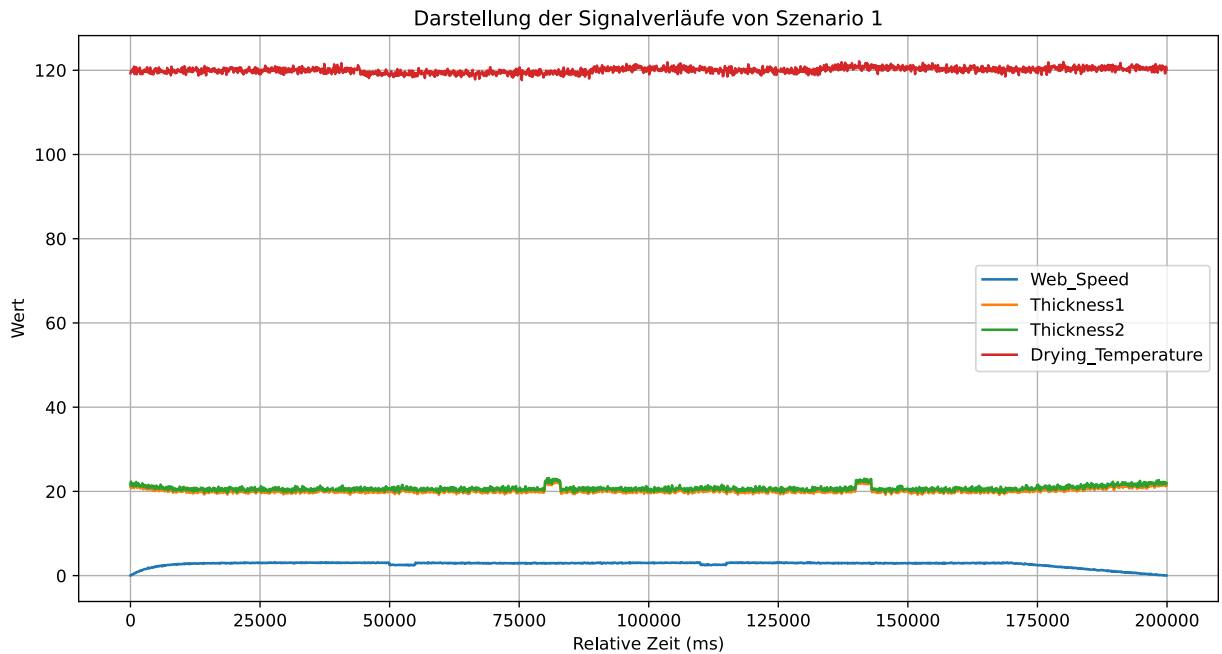


Abbildung 5.6: Darstellung der Signalverläufe im Szenario 1. Die Zeit ist relativ zum Startzeitpunkt (dem niedrigsten Zeitwert über alle Zeitreihen) aufgetragen.

Szenario 2 und Szenario 2b

Das Szenario 2 beschreibt einen Ablauf in einem Produktionsaufbau während eines Anlagentests. Die Maschinensteuerung erfasst die Prozessgeschwindigkeit, was im Kontext des Produktionsnetzwerks als Sensor gilt ($C_1 \succ_{\text{Process_Speed}} Z$). Sensorsysteme erfassen zudem innere und äußere Temperatur des Bearbeitungsbereichs ($C_4 \succ_{\text{Ambient_Temperature}} Z$, $C_2 \succ_{\text{Internal_Temperature}} Z$). Zuletzt erfasst ein Vibrationssensor die Vibrationen bei der Bearbeitung ($C_3 \succ_{\text{Machine_Vibration}} Z$). Neben der Senke C_{senke} existiert auch ein Gateway, welches zunächst die Werte aus C_1 und C_2 zusammenfasst und danach an C_{senke} weiterleitet. Die Signalverläufe sind in Darstellung 5.7 aufgetragen. Die mittlere Frequenz der Zeitreihen liegt bei 10 Hz. Die Informationen zum Aufbau des Systems umfassen die Existenz des Gateways und eine große Charakterisierung der Latenz von Gateway zur Senke. Weiteres Wissen steht als Reihe von Ursache-Wirkzusammenhängen zur Verfügung. Dies umfasst zeitlich versetzte Effekte der Umgebungstemperatur auf die interne Temperatur und das Vibrationsverhalten sowie Einflüsse von der Geschwindigkeit auf das Vibrationsverhalten. Das Szenario umfasst insgesamt 40.000 Datenpunkte.

Zusätzlich zu dem Szenario 2 wie oben beschrieben existiert eine Variation als Szenario 2b.

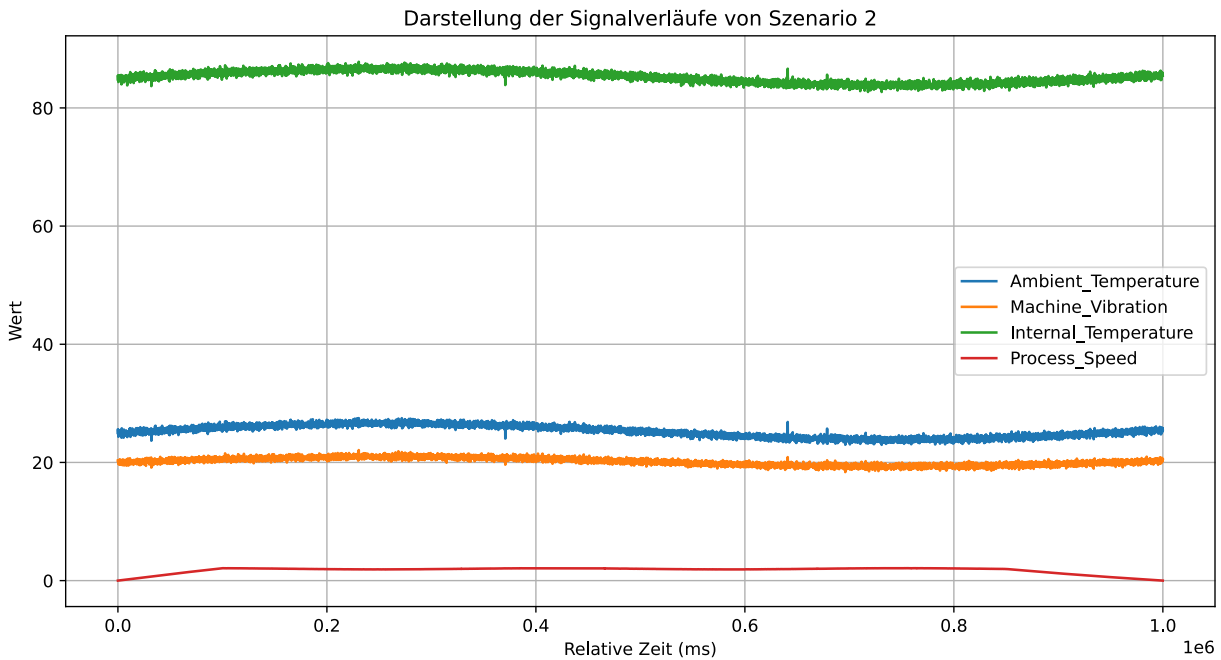


Abbildung 5.7: Darstellung der Signalverläufe im Szenario 3. Die Zeit ist relativ zum Startzeitpunkt (dem niedrigsten Zeitwert über alle Zeitreihen) aufgetragen.

Diese Variation übernimmt den Aufbau, das Wissen und die Informationen aus Szenario 2, verändert die Länge der Zeitreihen allerdings signifikant. Dadurch erhöht sich die Menge der Datenpunkte auf 240.000.

Besonderheit bei den Szenarien 2 und 2b ist hier in den Daten die Existenz von mehreren ähnlichen Mustern und gleichzeitigen Ausreißern. Weiterhin ist die Veränderung der Menge der zugrunde liegenden Daten als Unterschied von Szenario 2 zu 2b signifikant.

Szenario 3

Das Szenario 3 beschreibt einen Ablauf in einem Produktionsaufbau. Die Maschinensteuerung erfasst die Geschwindigkeit, was im Kontext des Produktionsnetzwerks als Sensor gilt ($C_4 \succ \text{Machine_Speed}Z$). Ein Sensorsystem erfasst den Druck ($C_1 \succ \text{Pressure}Z$), ein weiteres System die Durchflussrate ($C_2 \succ \text{Flow_Rate}Z$) und noch ein System die Temperatur des Bearbeitungsbereichs ($C_5 \succ \text{Temperature}Z$). Zusätzlich nimmt ein Energiezähler den Energieverbrauch der Maschine auf ($C_3 \succ \text{Energy_Consumption}Z$). Die Signalverläufe sind in Darstellung 5.8 aufgetragen.

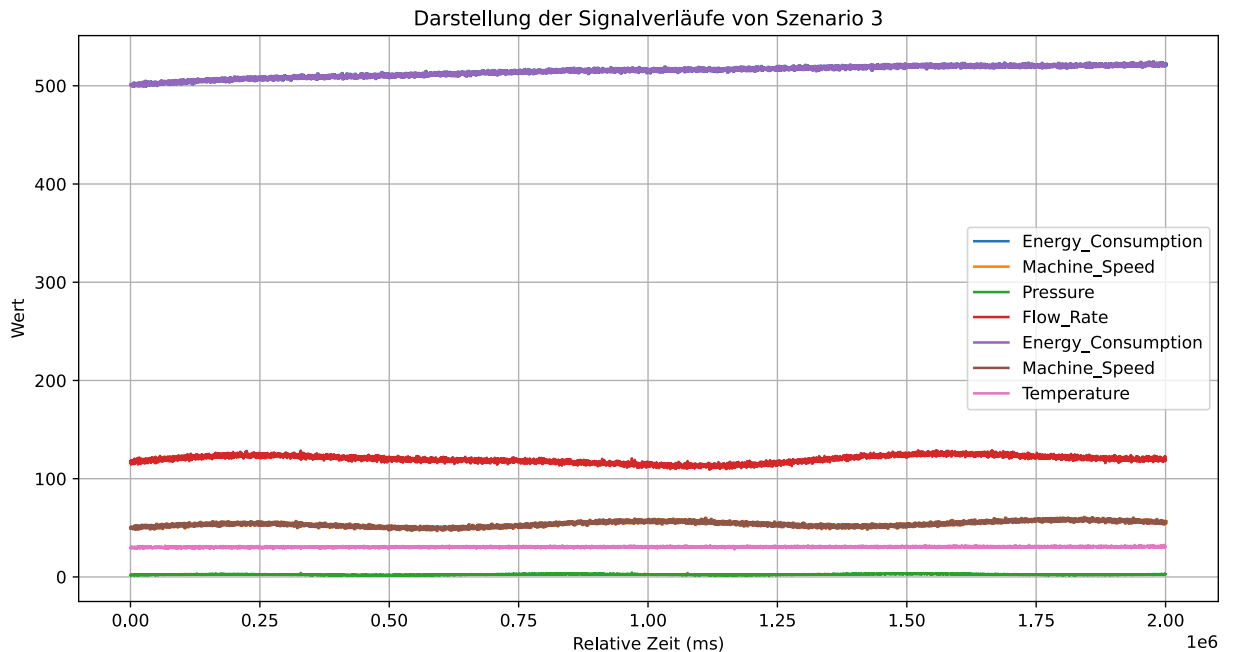


Abbildung 5.8: Darstellung der Signalverläufe im Szenario 3. Die Zeit ist relativ zum Startzeitpunkt (dem niedrigsten Zeitwert über alle Zeitreihen) aufgetragen.

Im Produktionsnetzwerk existieren zwei Gateways. Das erste Gateway (C_{G1}) leitet die Daten aus C_2 und C_1 an die Senke weiter. Hingegen führt das Gateway 2 (C_{G2}) die Daten aus C_4 und C_3 zusammen und fügt eigene Zeitangaben hinzu. Daher sind die entsprechenden Zeitreihen, welche mehrere Quellen (Sensor und Gateway) haben auch mehrfach dargestellt mit einem geringen zeitlichen Abstand in der Darstellung.

Neben den Informationen zum Aufbau inklusive einer Charakterisierung der Latenzen der Netzwerkverbindungen steht auch Wissen in Form von Ursache-Wirkzusammenhängen bereit. Dieses Wissen umfasst Verhaltensänderungen von Signalen bei Anstiegen, Ausbrüchen und Abfällen zueinander. Das Szenario umfasst insgesamt 70.000 Datenpunkte.

Besonders bei diesem Szenario ist in den Daten die Existenz von stark veränderlichen und kaum veränderlichen Zeitdatenreihen, sowie langfristige Trends. Bei dem Aufbau des Produktionsnetzwerks stehen mehrere gestaffelte Gateways, inklusive eigenen Zeitinformationen bereit.

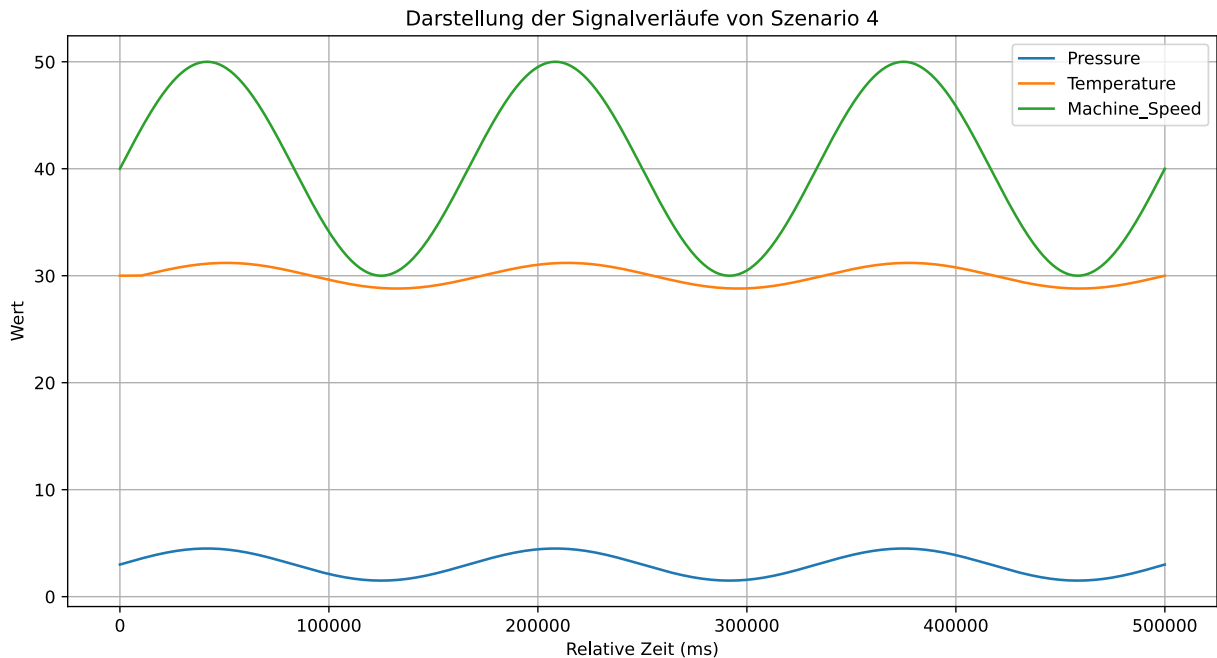


Abbildung 5.9: Darstellung der Signalverläufe im Szenario 4. Die Zeit ist relativ zum Startzeitpunkt (dem niedrigsten Zeitwert über alle Zeitreihen) aufgetragen.

Szenario 4

Das Szenario 4 beschreibt einen idealisierten gefilterten Ablauf in einem Produktionsaufbau. Die Maschinensteuerung erfasst die Geschwindigkeit, was im Kontext des Produktionsnetzwerks als Sensor gilt ($C_3 \succ_{\text{Machine_Speed}} Z$). Ein Sensorsystem C_3 nimmt den Druck in einer Komponente auf ($C_3 \succ_{\text{Pressure}} Z$). Zuletzt existiert das Temperaturmesssystem an der Seite dieser Komponente ($C_2 \succ_{\text{Temperature}} Z$). Alle Sensoren sind direkt mit der Senke C_{senke} verbunden, also existieren keine Gateways. Die Signalverläufe sind in Darstellung 5.9 aufgetragen. Abseits der Informationen zum Aufbau des Produktionsnetzwerks sind keine Informationen oder Wissen bereitgestellt an das Synchronisierungssystem. Insgesamt umfasst das Szenario 30.000 Datenpunkte.

Besonders beim Szenario 4 ist in den Daten die starke Glättung und Idealisierung, sodass kein Jitter vorhanden ist, sowie starke Unterschiede in den Frequenzen zwischen den Zeitdatenreihen. Weiterhin existieren einige Elemente im Bereich Wissen, welche hier nicht zum Einsatz kommen, da die betroffenen Sensoren nicht existieren und das System diese herausfiltern muss.

Szenario 5

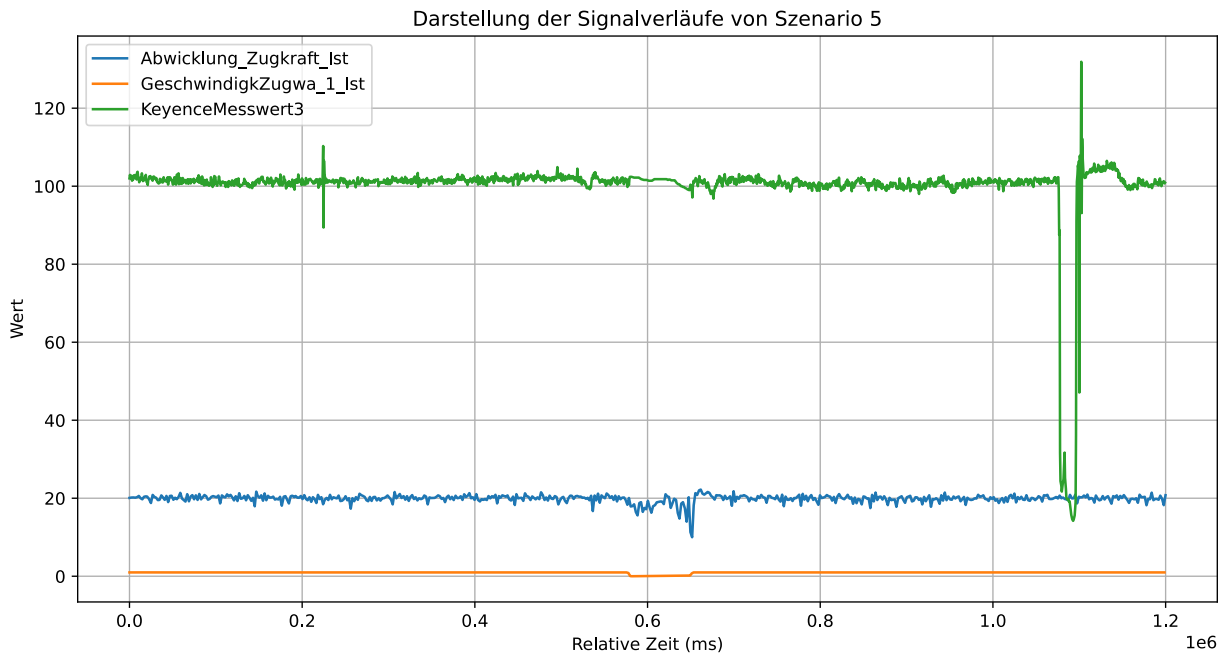


Abbildung 5.10: Darstellung der Signalverläufe im Szenario 5. Die Zeit ist relativ zum Startzeitpunkt (dem niedrigsten Zeitwert über alle Zeitreihen) aufgetragen.

Das Szenario 5 beschreibt Ablauf in einem Produktionsaufbau. Die Maschinensteuerung erfasst die Geschwindigkeit der Zugwalze, was im Kontext des Produktionsnetzwerks als Sensor gilt ($C_3 \succ \text{GeschwindigkZugwa_1_Ist}Z$). Ein Sensorsystem C_2 misst an eben jener Zugwalze zusätzlich die effektive Zugkraft ($C_2 \succ \text{Abwicklung_Zugkraft_Ist}Z$). Zuletzt existiert ein System zur Messung der Schichtdicke ($C_1 \succ \text{KeyenceMesswert3}Z$). Alle Sensoren sind direkt mit der Senke C_{senke} verbunden, also existieren keine Gateways. Die Signalverläufe sind in Darstellung 5.10 aufgetragen. Die Frequenz der Daten aus C_1 ist deutlich höher als in den anderen Systemen, während die Frequenz aus C_2 gegenüber C_3 leicht erhöht ist. Abseits der Informationen zum Aufbau des Produktionsnetzwerks steht Wissen in Form von Ursache-Wirkzusammenhängen bereit, welches sich um deutliche kurzfristige Abfälle der Werte für Geschwindigkeit und Zugkraft befasst und die Auswirkung auf die Schichtdicke. Insgesamt umfasst das Szenario 2.933 Datenpunkte.

Besonders beim Szenario 5 ist in den Daten insbesondere die Existenz von deutlichen Anomalien, welche sich teilweise über mehrere Zeitdatenreihen erstrecken. Weiterhin existieren zwischenzeitliche Veränderungen bei der Frequenz der Zeitdatenreihen.

Szenario 6

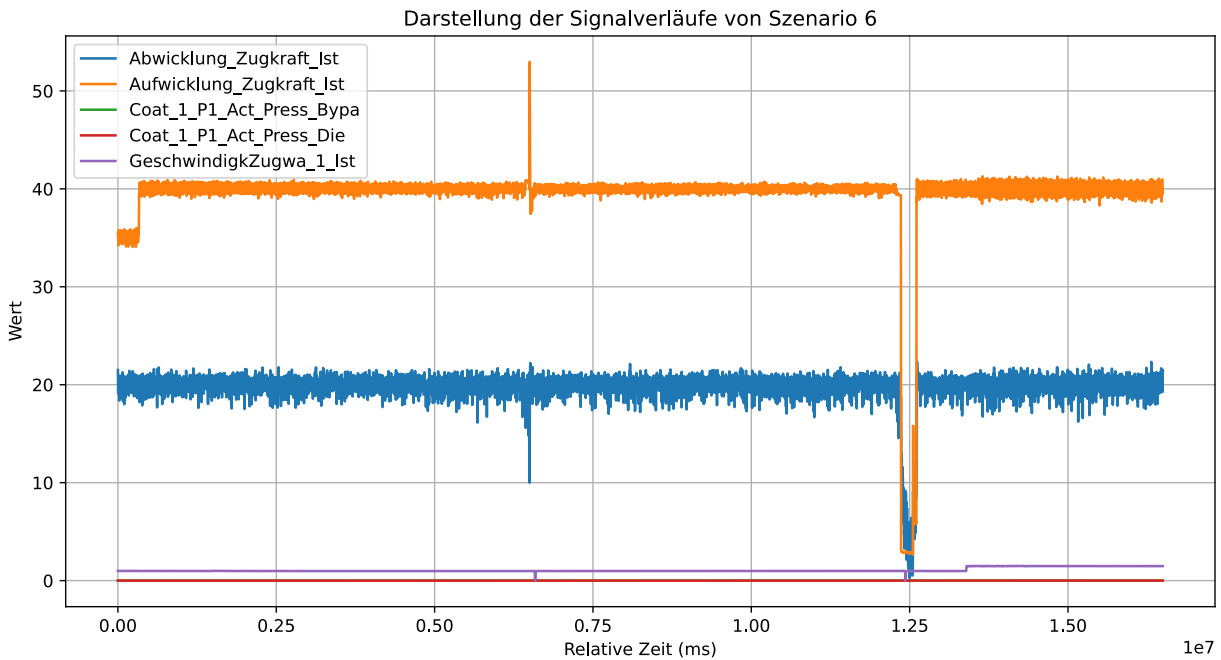


Abbildung 5.11: Darstellung der Signalverläufe im Szenario 6. Die Zeit ist relativ zum Startzeitpunkt (dem niedrigsten Zeitwert über alle Zeitreihen) aufgetragen.

Das Szenario 6 beschreibt Ablauf in einem Produktionsaufbau ähnlich zu Szenario 5. Neben der Geschwindigkeit ($C_5 \succ \text{GeschwindigkZugwa}_1\text{Ist}Z$) und Zugkraft der Abwicklung ($C_3 \succ \text{Abwicklung_Zugkraft_Ist}Z$) wird auch die Zugkraft der Aufwicklung ($C_4 \succ \text{Aufwicklung_Zugkraft_Ist}Z$) aufgenommen. Darüber hinaus kommen Sensoren im Beschichtungsbereich dazu. Ein Sensor nimmt den Druck an der Schlitzdüse ($C_2 \succ \text{Coat}_1\text{P1_Act_Press_Die}Z$) auf und ein weiterer Sensor analog dazu den Druck am Bypass ($C_1 \succ \text{Coat}_1\text{P1_Act_Press_Bypa}Z$). Die Signalverläufe sind in Darstellung 5.11 aufgetragen. Eine Sensorik für die Schichtdicke existiert nicht. Im Bereich Wissen stehen Ursache-Wirkzusammenhänge für die Wechselwirkung von den Walzen mit der Geschwindigkeit zur Verfügung, allerdings keinerlei Wissen im Bereich der Beschichtung und Schlitzdüse. Die Datenreihen haben eine mittlere Frequenz von 0,5 Hz., wobei die Abweichungen zwischen den Zeitreihen gering sind. Insgesamt umfasst das Szenario 39.867 Datenpunkte.

Besonders in dem Szenario 6 ist die Existenz mehrerer gleich. und gegenläufiger Ausreißer, welche teilweise zugleich eine Lücke in den Zeitdatenreihen mit sich bringen. Weiterhin ist das eingebrachte Wissen eingeschränkt auf einen geringen Teil der verfügbaren Sensoren.

Szenario 7

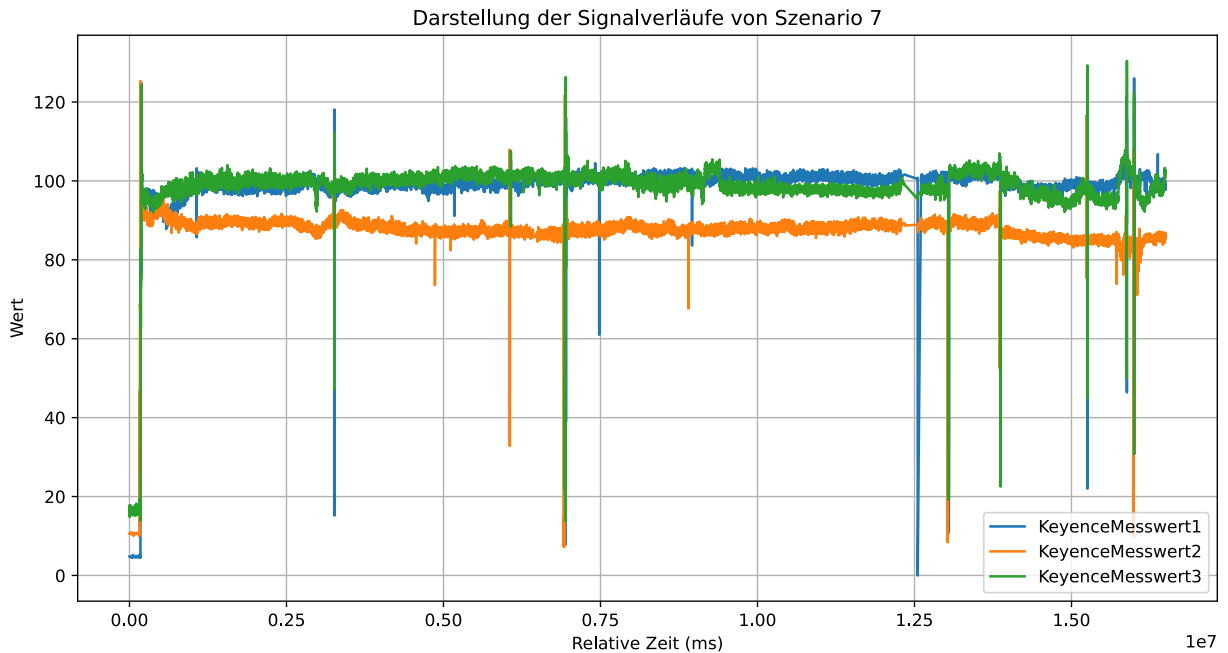


Abbildung 5.12: Darstellung der Signalverläufe im Szenario 7. Die Zeit ist relativ zum Startzeitpunkt (dem niedrigsten Zeitwert über alle Zeitreihen) aufgetragen.

Das Szenario 7 beschreibt Ablauf in einem Produktionsaufbau. Drei Sensoren nehmen dabei die Schichtdicken bei einer Beschichtungsanlage auf ($C_1 \succ_{\text{KeyenceMesswert1}} Z$, $C_2 \succ_{\text{KeyenceMesswert2}} Z$, $C_3 \succ_{\text{KeyenceMesswert3}} Z$). Alle Sensoren sind direkt mit der Senke C_{senke} verbunden, also existieren keine Gateways. Die Signalverläufe sind in Darstellung 5.12 aufgetragen. Die Frequenzen der Datenaufnahme der Sensoren im Mittel bei 2 Hz, wobei die Aufnahme nicht gleichmäßig ist und leichte Unterschiede zwischen den Systemen hat. Abseits der Informationen zum Aufbau des Netzwerks stehen keine weiteren Informationen oder Wissen zur Verfügung. Insgesamt umfasst das Szenario 76.143 Datenpunkte.

Besonders bei dem Szenario 7 ist das Fehlen jeglichen Wissens und Informationen abseits des Grundaufbaus. Darüber hinaus existieren mehrere Ausreißer in den Daten, welche teilweise gleichzeitig und teilweise unabhängig sind.

5.2.3 Metriken der Bewertungen

Metrics for Evaluations

Die Szenarien gemäß der Beschreibung in Abschnitt 5.2.2 und der Versuchsmatrix (Tabelle 11) wurden durchgeführt im Synchronisierungssystem. Ziel der Ausführung war die Bereitstellung einer bestmöglichen Projektion (Φ) um die Effekte der Asynchronizität auszugleichen. Um eine entsprechende Analyse und Bewertung der Ergebnisse durchzuführen, beschreibt dieser Abschnitt zunächst die Metriken und die Darstellungen für die Ergebnisanalyse.

Analog zu der Klassifikation der Bewertung in Abschnitt 3.2 stehen drei Hauptmetriken im Fokus der Analyse. Diese befassen sich mit der Qualität, der Effizienz und der Anwendbarkeit.

Für die Metrik der *Anwendbarkeit* gilt die einfache Betrachtung der formal erfolgreich durchgeführten Synchronisationen. Dies sind alle Synchronisierungsanfragen an das Synchronisierungssystem, welche nicht mit einem Fehler beendet wurden, sondern eine Synchronisierung erstellt haben, unabhängig von der Qualität. Diese Anwendbarkeit ist damit definiert als

$$B_{Anwendbarkeit} = 1 - \frac{|\text{Nicht-erfolgreiche Synchronisierungen}|}{|\text{Versuche}|}. \quad (5.12)$$

Während $B_{Anwendbarkeit}$ die formal erfolgreichen Synchronisierungen beschreibt, ist eine Metrik für die Beschreibung der inhaltlichen *Qualität* notwendig. Für die Betrachtung der Qualität dient grundsätzlich der Unterschied von dem idealen Zeitwert eines Elements Iz_i der Zeitdatenreihe zu dem Ergebnis der Synchronisierung dieses Punkts Φz_j . Für ein System mit mehreren synchronisierten Zeitdatenreihen $\Phi \mathfrak{Z}$ und den zugehörigen idealen Zeitdatenreihen $I \mathfrak{Z}$ lassen sich alle Abweichungen zusammenfassen als

$$B_{RMSE} = \{RMSE^{(j)} \mid j = 1, \dots, N\} \quad (5.13)$$

$$RMSE^{(j)} = \left\{ \sqrt{\frac{1}{|I \mathfrak{Z}^{(j)}|} \sum_{\forall I z_i \in I \mathfrak{Z}^{(j)}} d(I z_i, \Phi z_i)^2} \right\}, \quad (5.14)$$

wobei die $RMSE^{(j)}$ die Distanzberechnung für Versuch j aus der Versuchsplanung be-

schreibt, wodurch B_{RMSE} die Menge aller Distanzergebnisse aus den Versuchen ist. Entsprechend ist N die Anzahl der Versuche.

Zuletzt beschreibt die Metrik der benötigten *Zeit* die Dauer der vollständigen Durchführung der Synchronisierung für die Daten eines Produktionssystems. Analog zur Qualitätsmetrik ist diese Metrik eine Menge aller Einzelergebnisse der Versuche:

$$B_{Zeit} = \{ {}_{OSS}T_{ende}^{(j)} - {}_{OSS}T_{start}^{(j)} \mid j = 1, \dots, N \}, \quad (5.15)$$

wobei ${}_{OSS}T^{(j)}$ Start- bzw. Endzeitpunkt der Ausführung des Offline Synchronisierungssystems angeben. Hierzu dient die Systemuhr der eingesetzten virtuellen Maschine. Asynchronizitätseffekte auf der virtuellen Maschine sind für den Zweck und benötigte Genauigkeit der Auswertung vernachlässigbar.

Für die Auswertung der Versuche in den Szenarien kommen für B_{RMSE} und B_{Zeit} entsprechende Boxplots zum Einsatz. Bei B_{RMSE} wird zudem die mittlere Abweichung basierend auf den Asynchronizitätseffekten vor der Ausführung als rote gestrichelte Linie angegeben. Werte von B_{RMSE} unterhalb dieser Linie bedeuten dadurch, dass die Synchronisierung die Qualität verbessert, während Werte oberhalb dieser Linie einer Verschlechterung entsprechen. Für B_{RMSE} als auch B_{Zeit} sind nur erfolgreiche Durchführungen berücksichtigt und niedrigere Werte besser. Hingegen ist $B_{Anwendbarkeit}$ als einfaches Balkendiagramm angegeben, bei dem höhere Werte besser sind und per Definition alle Versuche Betrachtung finden.

5.2.4 Auswertung der Szenarien

Evaluation of the scenarios

Alle Szenarien, gemäß der Beschreibung in Abschnitt 5.2.2 wurden auf dem gleichen System durch das Offline Synchronisierungssystem mit allen Versuchen der Versuchsmatrix durchgeführt. In diesem Abschnitt sind die Ergebnisse dargestellt, wobei die Metriken aus Abschnitt 5.2.3 genutzt sind. Zusätzliche Eindrücke und Verhaltensweisen sind eventuell zusätzlich angegeben. Ebenfalls findet eine Einordnung der Ergebnisse innerhalb des jeweiligen Szenarios statt.

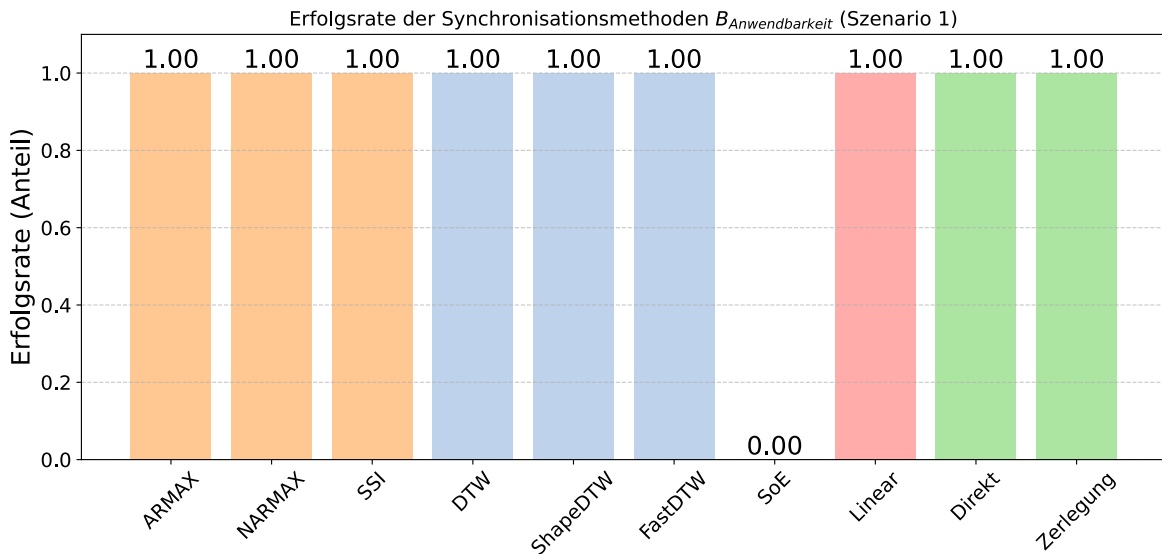


Abbildung 5.13: Darstellung der Erfolgsquoten ($B_{Anwendbarkeit}$) der einzelnen Methoden im Szenario 1. Ein Wert von 0 beschreibt eine Methode, welche nicht angewendet werden konnte, oder bei jeder Durchführung mit einem Fehler endete.

Szenario 1

In diesem Szenario existieren nicht für alle Systeme und Datenflüsse auf dem Weg durch das Produktionsnetzwerk eigenständige Zeitangaben, daher ist die Methode *Sequence of Events* nicht anwendbar und erhält eine entsprechend niedrige Bewertung für $B_{Anwendbarkeit}$. Für alle anderen Methoden ist der Aufbau und der Verlauf der Signale ausreichend, sodass die Durchführungen vollständig funktioniert haben, wie in Diagramm 5.13 dargestellt.

Im Bereich der Qualität zeigen die eingesetzten Methoden deutliche Unterschiede, dargestellt im Diagramm 5.14. Die Methoden *ARMAX* und *NARMAX* führen zu den gleichen Ergebnissen, während *SSI* schlechtere Ergebnisse liefert mit einzelnen Ausreißern, welche eine deutliche Verschlechterung der Datenqualität bedeuten. Noch weiter verschlechtert sich die Datenqualität bei der Nutzung von *DTW*. Während in den Daten ein ähnlicher Verlauf der Signale »Thickness1« und »Thickness2« erkennbar ist, führt die Anwendung mit den anderen Signalen zu einer Verschlechterung, da der Waring Pfad stärker auf kleine Veränderungen und dem Jitter in den Daten reagiert. Das schlechteste Ergebnis liefert die Methode *ShapeDTW*, bei der die Verschlechterung relativ konstant oberhalb einer Sekunde ist. Ebenfalls aus der Familie der Dynamic Time Warping Methoden, schafft

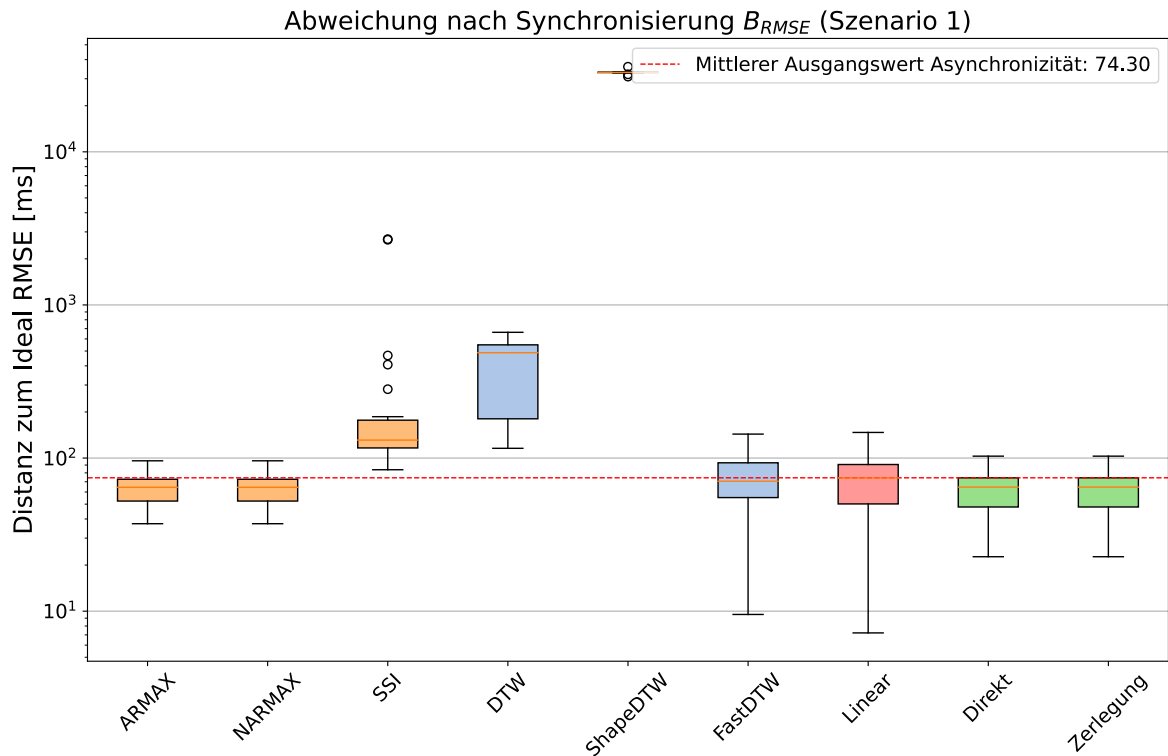


Abbildung 5.14: Darstellung der Qualität (B_{RMSE}) der einzelnen Methoden im Szenario 1. Die gestrichelte rote Linie zeigt die Abweichung von der Grundwahrheit nach der Anwendung von Asynchronizitätseffekten.

FastDTW deutlich bessere Ergebnisse als die zuvor genannten Methoden. Die Verteilung der Qualitätswerte bleibt dabei nahe an der originalen Asynchronizität, sodass eine deutliche Verbesserung nur in Einzelfällen zu beobachten ist. Ein ähnliches Bild zeichnet sich bei der *Linearisierung* ab. Eine leichte Verbesserung der Qualität ist in den meisten Fällen bei der Methode *Direkt* zu erkennen. In den meisten Fällen kommt es hier zur Auswahl der Methoden aus *ARMAX* und *NARMAX*, wobei die Reihenfolge angepasst ist. Diese Anpassung führt zu einer kleinen Verbesserung gegenüber der allgemeinen Anwendung der benannten Methoden. Die Qualität bei Nutzung der *Zerlegung* verändert sich im Vergleich nicht wesentlich.

Für die benötigte Zeit, dargestellt im Diagramm 5.15, zeigt sich eine deutliche Unterscheidung in mehrere Cluster. Die Methoden *ARMAX* und *NARMAX* sowie die neuen Methoden, welche auf diese zurückgreifen, benötigen signifikant mehr Zeit als die anderen Methoden. Die Methode der *Linearisierung* bleibt konstant unterhalb einer Sekunde bei der Anwendung und ist die schnellste Methode. Alle weiteren Methoden siedeln sich im

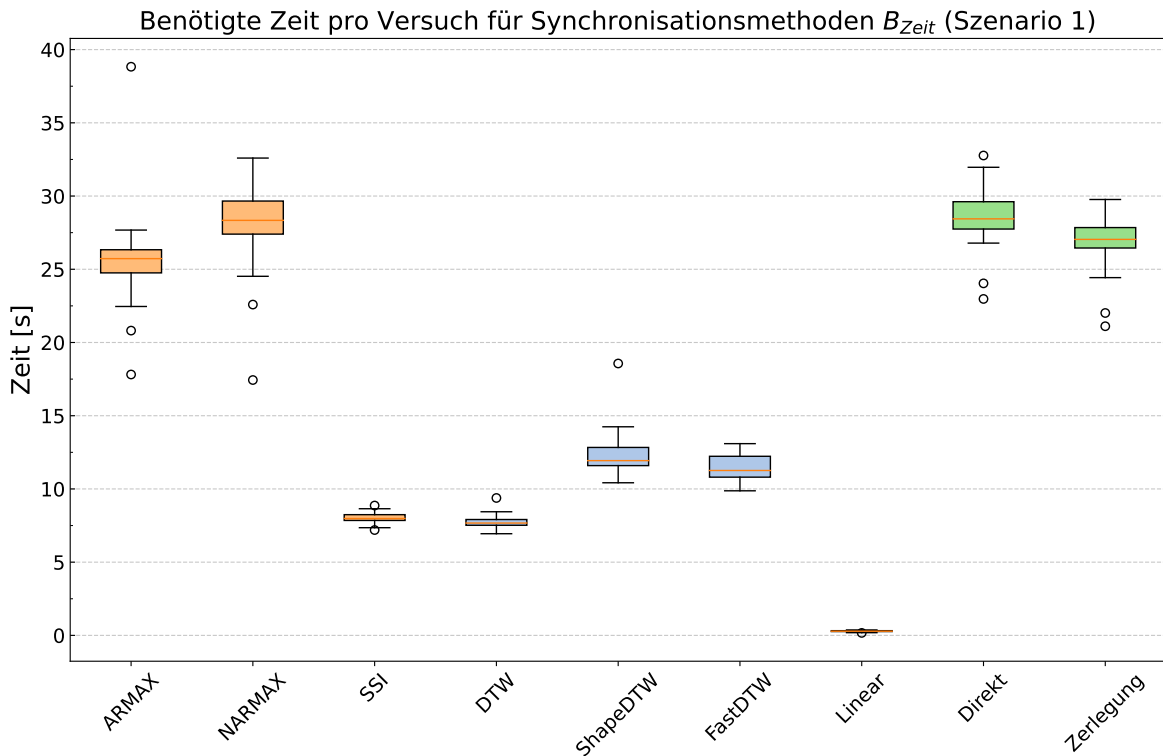


Abbildung 5.15: Darstellung der benötigten Zeit (B_{Zeit}) der einzelnen Methoden im Szenario 1.

Mittelfeld an, wo bis auf einzelne Ausreißer die Zeiten relativ konstant bleiben. Die Methode der *Zerlegung* nutzt einzelne Fixpunkte und unterteilt damit eine Synchronisierung in mehrere, wodurch ein leichter Vorteil gegenüber der direkten Anwendung existiert. Insgesamt bleiben die neuen Methoden in einem ähnlichen Zeitbereich wie *ARMAX* und *NARMAX*, allerdings tendenziell leicht schlechter.

Insgesamt sind in dem Szenario 1 mehrere Methoden anwendbar und führen zu leichten Verbesserungen der Qualität. Die Methoden mit der besten Qualität gehen dabei auch mit dem höchsten Zeitaufwand einher. Die neuen Methoden bieten hier geringe Vorteile bei der Qualität und geringe Nachteile bei der Zeit gegenüber anderen Methoden.

Szenario 2

In diesem Szenario existieren nicht für alle Systeme und Datenflüsse auf dem Weg durch das Produktionsnetzwerk eigenständige Zeitangaben, daher ist die Methode *Sequence of*

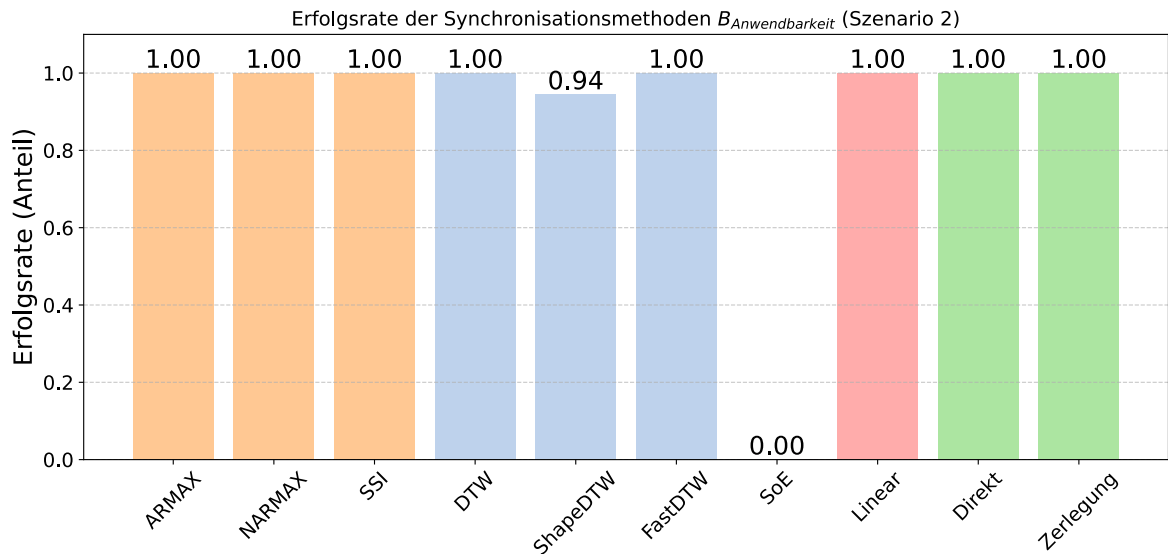


Abbildung 5.16: Darstellung der Erfolgsquoten ($B_{Anwendbarkeit}$) der einzelnen Methoden im Szenario 2. Ein Wert von 0 beschreibt eine Methode, welche nicht angewendet werden konnte, oder bei jeder Durchführung mit einem Fehler endete.

Events nicht anwendbar und erhält eine entsprechend niedrige Bewertung für $B_{Anwendbarkeit}$. Im Normalfall kommen die anderen Methoden zu einer erfolgreichen Durchführung der Synchronisierung wie in Diagramm 5.16 dargestellt, lediglich bei zwei Versuchen bricht *ShapeDTW* mit einem Fehler ab, dass kein Pfad gefunden werden kann.

Im Bereich der Qualität zeigen die meisten eingesetzten Methoden geringere Unterschiede. Lediglich die Methode *ShapeDTW* verschlechtert die Qualität erheblich in diesem Szenario, wie erkennbar im Diagramm 5.17. Die Anwendung von *SSI* zeigt nur in Einzelfällen eine Verbesserung, während in den meisten Fällen eine Verschlechterung sichtbar ist. Bei allen weiteren Methoden sind die Ergebnisse im vergleichbaren Wertebereich zu finden. Lediglich leichte Abweichung in Richtung einer Verbesserung sind hier erkennbar. Keine der Methoden schafft durchgehend eine deutliche Verbesserung der Qualität. Im Hinblick auf die Signalverläufe deuten diese Qualitätsergebnisse darauf hin, dass mehrere Methoden ähnliche Eigenschaften und Zusammenhänge erkennen und nutzen für die Synchronisierung.

Bei der benötigten Zeit, dargestellt im Diagramm 5.18, ist eine hohe Spannweite erkennbar. Analog zu Szenario 1 sind hier die neuen Methoden, *ARMAX* und *NARMAX* vergleichbar. Auffällig sind hier die Menge der Ausreißer, welche vereinzelt deutlich schnellere Durch-

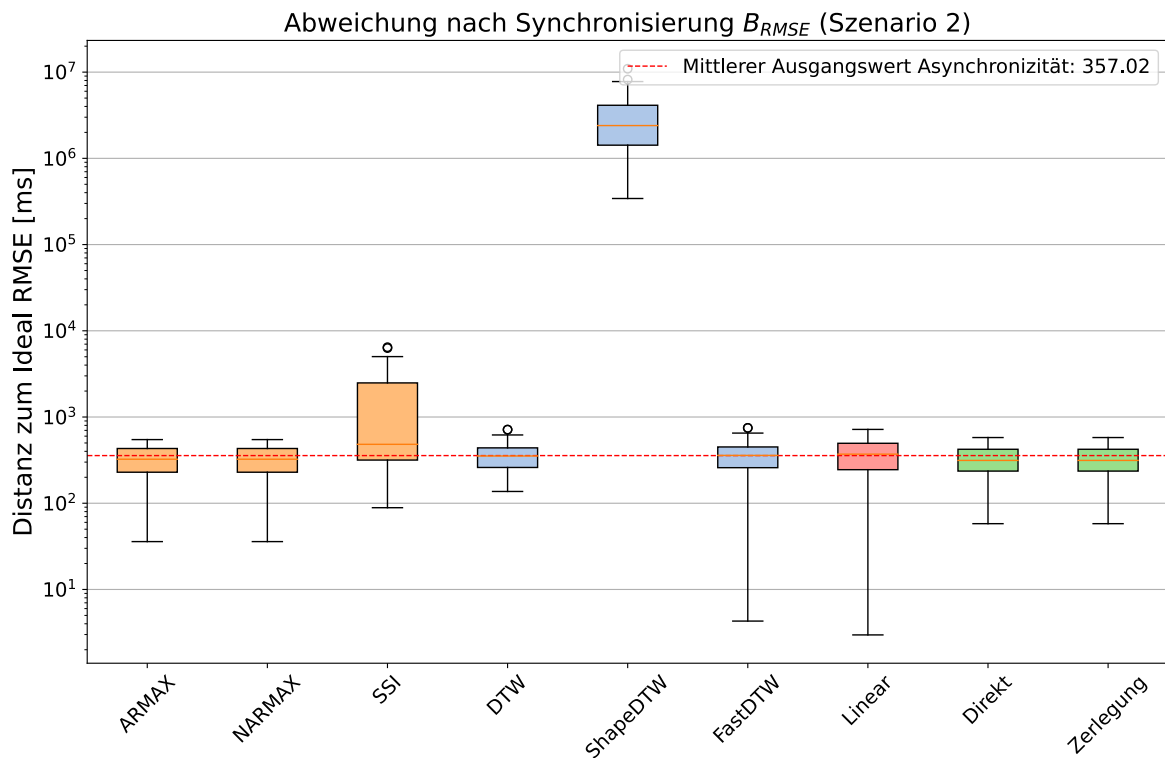


Abbildung 5.17: Darstellung der Qualität (B_{RMSE}) der einzelnen Methoden im Szenario 2. Die gestrichelte rote Linie zeigt die Abweichung von der Grundwahrheit nach der Anwendung von Asynchronizitätseffekten.

fürungen darstellen. Bei den anderen Methoden sind die Zeiten in Relation dazu stabil im Bereich zwischen 40 und 100 Sekunden. Lediglich die *Linearisierung* zeigt deutlich niedrigere Werte von wenigen Sekunden.

Insgesamt sind in dem Szenario 2 mehrere Methoden anwendbar und führen zu minimalen Verbesserungen der Qualität, allerdings nicht für jede Kombination von Störgrößen. Eine im Verhältnis geringfügig bessere mittlere Qualität der Synchronisierungsmethoden geht dabei mit der Vervielfachung der benötigten Zeit einher. Die neuen Methoden bieten hier keine signifikanten Vorteile gegenüber anderen Methoden.

Szenario 2b

Während sich die Signalverläufe zwischen dem Szenario 2 und dem Szenario 2b hauptsächlich durch ihre Länge und die Menge von Datenpunkten unterscheiden, zeigen die

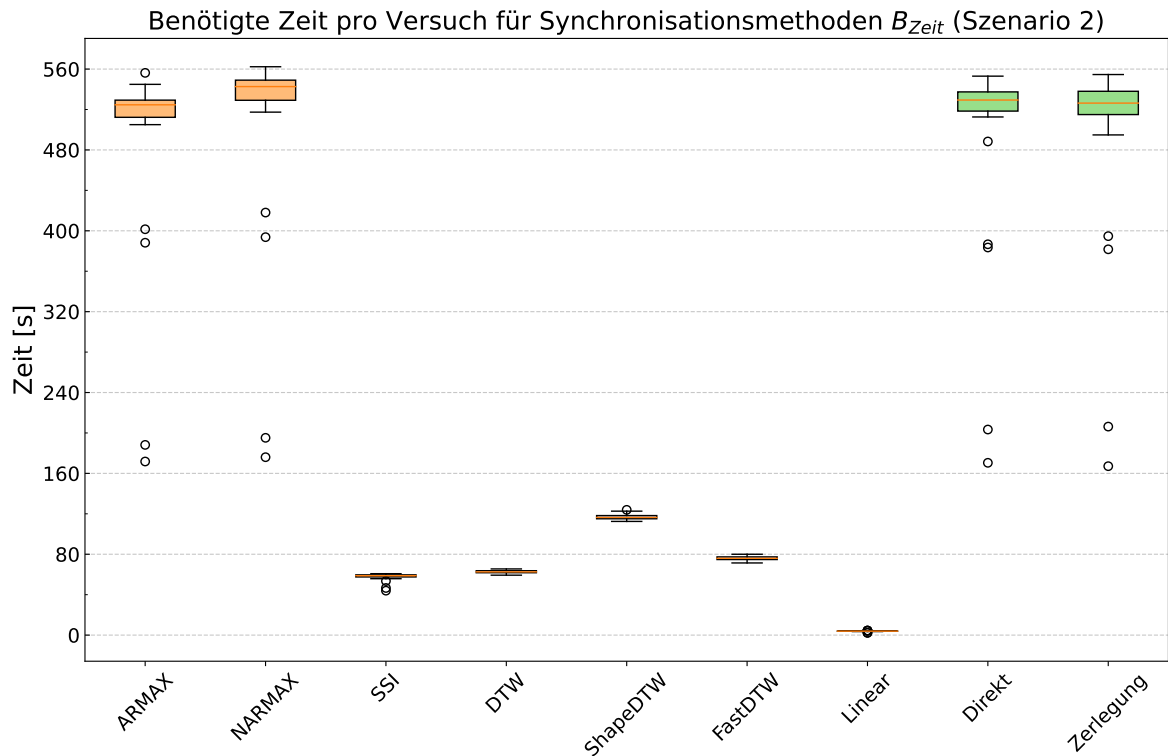


Abbildung 5.18: Darstellung der benötigten Zeit (B_{Zeit}) der einzelnen Methoden im Szenario 2.

Ergebnisse der Synchronisierungen deutliche Unterschiede. So sinkt die Menge der erfolgreichen Methoden der Synchronisierung nach $B_{Anwendbarkeit}$, sichtbar im Diagramm 5.19. Bei der Durchführung von *DTW* erstellt die Methode zunächst die Warping Matrix zusammen mit weiteren Datenstrukturen, was zu einer Allokation von über 300 GB RAM führt, bevor das Programm mit einem internen Speicherfehler abbricht. Ein ähnliches Bild zeigt sich bei *ShapeDTW*, auch wenn die belegte Menge des RAM vor dem Fehler niedriger ist. Bei den Methoden *ARMAX* und *NARMAX* fanden umfangreiche Berechnungen statt. Da diese Berechnungen allerdings nicht erkennbar zu einem Abschluss kamen, wurden Versuche nach 12 Stunden automatisch abgebrochen.

Aufseiten der Qualität sind die Ergebnisse der Synchronisierung in Diagramm 5.20 dargestellt und zeigen ein relativ gleichmäßiges Bild. Keine der erfolgreichen Methoden zeigt eine konsistente Verbesserung der Datenqualität. Bei *SSI* kann nur in Einzelfällen eine Verbesserung erreicht werden. Bei den anderen Methoden verteilt sich der Großteil der Ergebnisse um die ursprüngliche Asynchronizität. Das beste Ergebnis erreicht die *Zerlegung*, deren Median minimal unterhalb der ursprünglichen Asynchronizität liegt, insgesamt

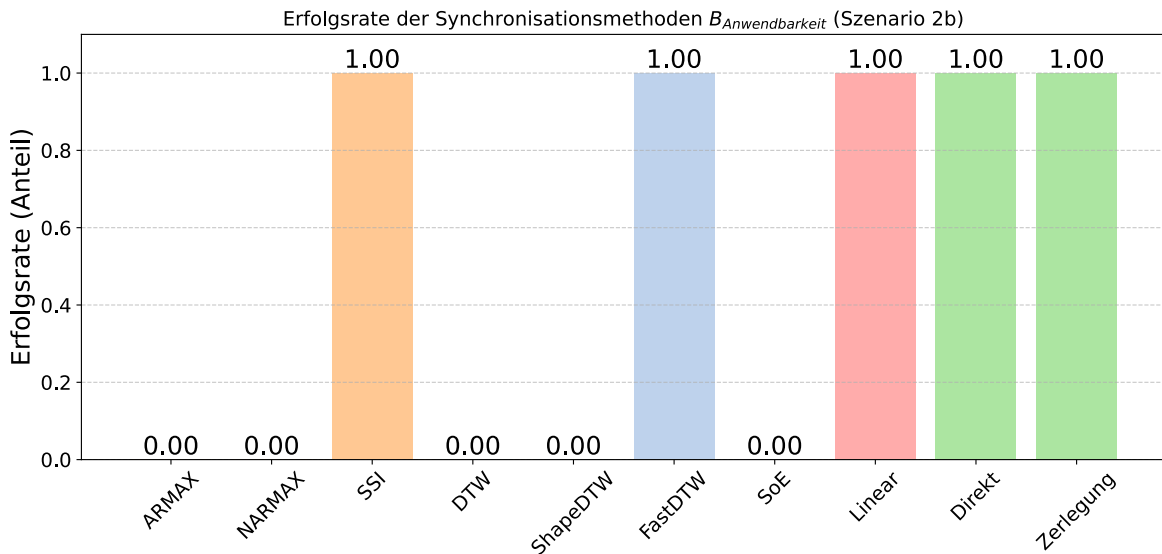


Abbildung 5.19: Darstellung der Erfolgsquoten ($B_{Anwendbarkeit}$) der einzelnen Methoden im Szenario 2b. Ein Wert von 0 beschreibt eine Methode, welche nicht angewendet werden konnte, oder bei jeder Durchführung mit einem Fehler endete.

aber ebenfalls viele Ergebnisse mit einer Verschlechterung beschreibt. Ohne die Zerlegung entlang von Fixpunkten, die Methode *Direkt*, sind die Ergebnisse vergleichbar, allerdings leicht schlechter, sodass der Median signifikant oberhalb der Referenz liegt.

Wie zuvor beschrieben, benötigen *ARMAX* und *NARMAX* eine derart umfassende Berechnungszeit, sodass diese als nicht anwendbar klassifiziert sind. Für *FastDTW* und *SSI* ergeben sich in diesem Szenario längere Berechnungszeiten, welche oberhalb der 20 Minuten angesiedelt sind, wie in Diagramm 5.21 erkennbar. Bei der *Linearisierung* und *Direkt* sind die Zeiten deutlich niedriger. Auffällig ist hier eine leichte Verringerung der benötigten Zeit gegenüber der *Linearisierung* im direkten Vergleich. Dies ist möglicherweise darauf zurückzuführen, dass bei der Reihenfolge der Synchronisierung die Menge der neu zu erstellenden, interpolierten Zeitwerte reduziert sein könnte. Der Unterschied selbst ist allerdings nicht als signifikante Unterscheidung bewertet. Bei der Nutzung der *Zerlegung* hingegen steigt die benötigte Zeit signifikant an, bleibt aber deutlich unterhalb der zeitaufwändigen Methoden.

Insgesamt sind in dem Szenario 2b weniger Methoden anwendbar und die Verbesserungen sind nur in wenigen Fällen erkennbar. Lediglich die *Zerlegung* führt im Median zu einer minimalen Verbesserung, allerdings ebenfalls nicht konsistent für alle Versuche. Die Erhöhung

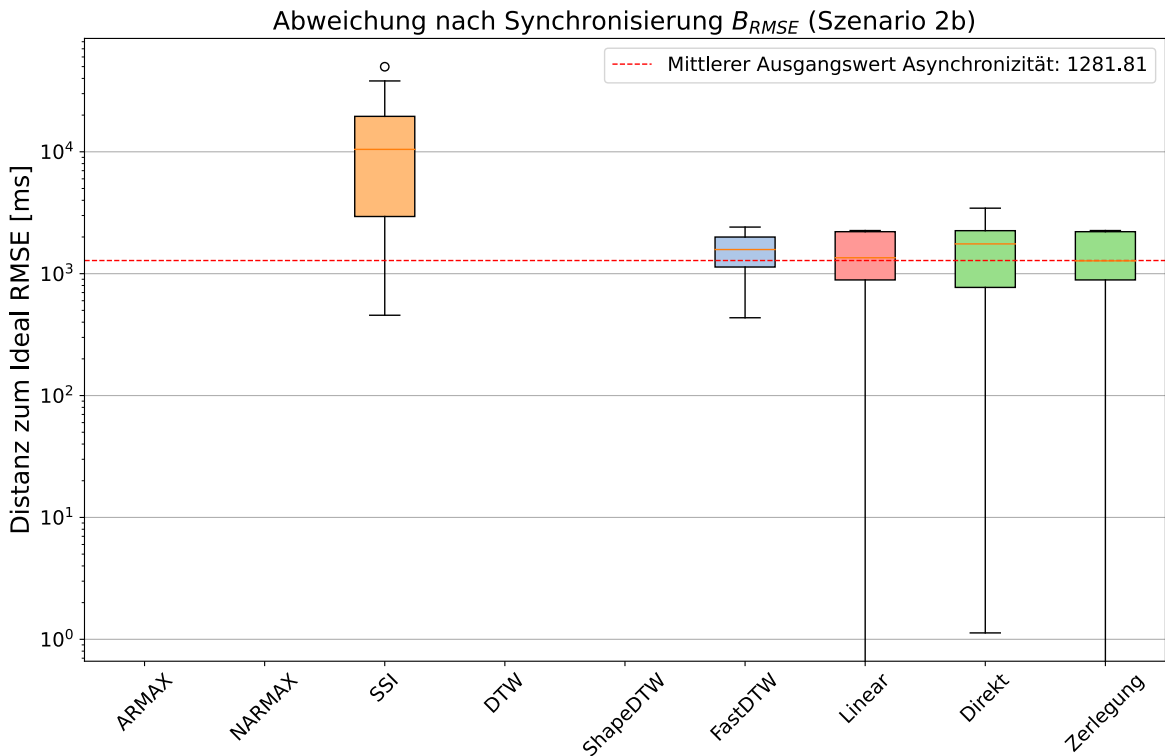


Abbildung 5.20: Darstellung der Qualität (B_{RMSE}) der einzelnen Methoden im Szenario 2b. Die gestrichelte rote Linie zeigt die Abweichung von der Grundwahrheit nach der Anwendung von Asynchronizitätseffekten.

der Datenmenge gegenüber Szenario 2 macht sich insbesondere bei der Anwendbarkeit und benötigten Zeit bemerkbar. In diesem Szenario ist der Mehrwert der Synchronisierung mit den vorgestellten Methoden fraglich.

Szenario 3

In diesem Szenario existieren nicht für alle Systeme und Datenflüsse auf dem Weg durch das Produktionsnetzwerk eigenständige Zeitangaben, daher ist die Methode *Sequence of Events* nicht anwendbar und erhält eine entsprechend niedrige Bewertung für $B_{Anwendbarkeit}$, auch wenn in einem Teilbereich des Produktionsnetzwerks die entsprechenden Bedingungen erfüllt sind. Die meisten Methoden kommen zu einer erfolgreichen Durchführung der Synchronisierung wie in Diagramm 5.22 dargestellt, lediglich bei drei Versuchen bricht *DTW* mit einem Fehler ab, dass kein Pfad gefunden werden kann und zugleich schafft *ShapeDTW* nur bei drei Versuchen überhaupt ein Ergebnis zu liefern.

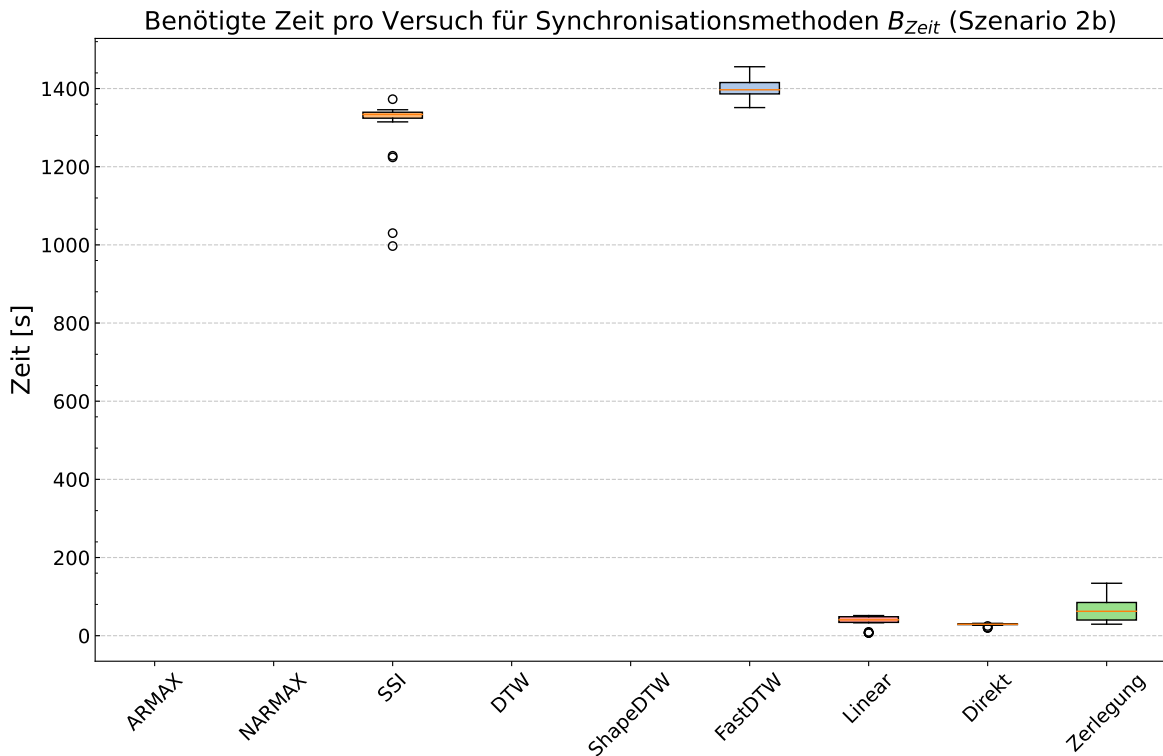


Abbildung 5.21: Darstellung der benötigten Zeit (B_{Zeit}) der einzelnen Methoden im Szenario 2b.

Im Aufbau des Produktionsnetzwerks existiert ein Teilbereich, welcher über Sensoren und ein Gateway mit Zeitangaben verfügt. Dies führt bei der Evaluierung in den neuen Methoden dazu, dass für diese im entsprechenden die Methode *SoE* als hoch und geeignet bewerten und nutzen. Da so ein Teil der Synchronisierung mit hoher Qualität durchgeführt werden kann, zeigen die Qualitätsergebnisse in Diagramm 5.23 ein deutliches Bild für diese neuen Methoden. Sie sorgen jeweils bis auf einen Versuch für eine deutliche Verbesserung der Datenqualität und unterscheiden sich dabei signifikant von den anderen Ergebnissen. Die Methoden *SSI* und *ShapeDTW* zeigen hingegen große Spannweiten der Qualität auf und verschlechtern in der Regel die Qualität der Daten. Auch bei *DTW* ist nur in Einzelfällen eine Verbesserung erkennbar, dafür eine auffallende Menge von Ausreißern nach oben. Bei den verbleibenden Methoden kommt es zu geringfügigen Verbesserungen der Datenqualität im Mittel, allerdings nicht in einem konsistenten Maße.

Bezüglich der benötigten Zeit lassen sich drei Cluster im Diagramm 5.24 erkennen. Die Methoden *ARMAX* und *NARMAX* benötigen die meiste Zeit und bewegen sich meistens im Bereich von 8 bis 11 Minuten. Das zweite Cluster umfasst *Direkt* und *Zerlegung*, welche

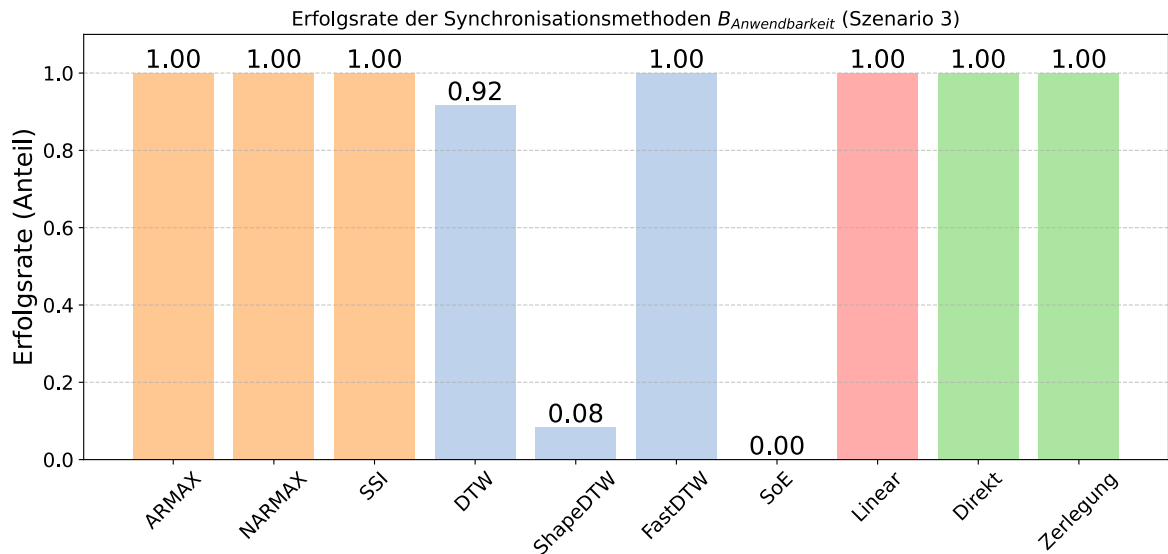


Abbildung 5.22: Darstellung der Erfolgsquoten ($B_{Anwendbarkeit}$) der einzelnen Methoden im Szenario 3. Ein Wert von 0 beschreibt eine Methode, welche nicht angewendet werden konnte, oder bei jeder Durchführung mit einem Fehler endete.

signifikant weniger Zeit benötigen als die zuvor genannten Alternativen. Die Streuung und Ausreißer orientieren sich dabei allerdings an eben jenen Methoden, da diese in der Bewertung teilweise ausgewählt sind. Das letzte Cluster umfasst die verbliebenen Methoden, welche eine geringe Streuung und wenige Minuten benötigen.

Szenario 3 zeigt deutliche Vorteile bei der Durchführung einer Synchronisierung mittels der neuen Methoden, welche auf die teilweise Anwendbarkeit hochqualitativer Methoden zurückgehen. Für die meisten Versuche ist die erwartete Verbesserung durch andere Methoden gering, sodass die Anwendung der Synchronisierung mittels langlaufender Methoden zu hinterfragen ist. Ebenfalls zeigt dieses Szenario den Grenzbereich der Anwendbarkeit für manche der Methoden auf.

Szenario 4

In diesem Szenario existieren nicht für alle Systeme und Datenflüsse auf dem Weg durch das Produktionsnetzwerk eigenständige Zeitangaben, daher ist die Methode *Sequence of Events* nicht anwendbar und erhält eine entsprechend niedrige Bewertung für $B_{Anwendbarkeit}$, wie in Diagramm 5.25 dargestellt. Für die meisten anderen Methoden ist der Aufbau und

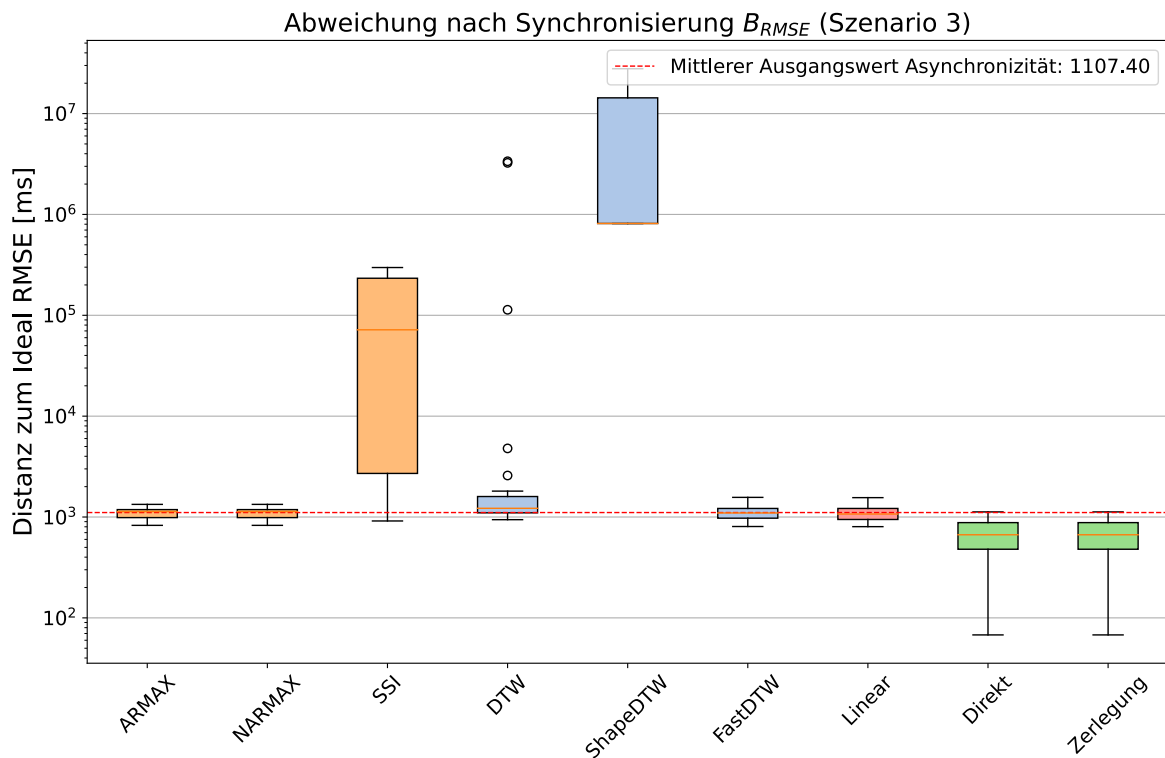


Abbildung 5.23: Darstellung der Qualität (B_{RMSE}) der einzelnen Methoden im Szenario 3. Die gestrichelte rote Linie zeigt die Abweichung von der Grundwahrheit nach der Anwendung von Asynchronizitätseffekten.

der Verlauf der Signale ausreichend, sodass die Synchronisierungen durchgeführt werden können. Allerdings kommt es bei der *Zerlegung* zu einzelnen Fehlern. Im Detail kommt es zu einzelnen falschen Fixpunkten. Diese führen zu einer deutlichen Verschiebung, was bei der Durchführung der eigentlichen Synchronisierungen dann zu Fehlern führt und im Gesamtablauf zu einem Abbruch führt.

Bei der Qualität sind mehrere Auffälligkeiten zu erkennen, dargestellt im Diagramm 5.26. Die neuen Methoden sowie *ARMAX* und *NARMAX* zeigen eine Verteilung um den Ausgangswert der Asynchronizität mit einem Median leicht unter jener Referenz. Bei *SSI* ist eine extreme Bandbreite zu erkennen, bei der einzelne Durchführungen zur Verbesserung der Qualität führen, aber der Großteil erhebliche Verschlechterungen mit sich bringt. Eine deutlich geringere Bandbreite, allerdings keinen Fall einer Verbesserung bringen die Varianten des Dynamic Time Warpings. Da die Signale im Verlauf bereits im Ideal leicht versetzt zueinander sind, ist ein Teil davon möglicherweise darauf zurückzuführen. Bei *ShapeDTW* und *FastDTW* allerdings kommt es am Anfang zu extremen Verschiebungen,

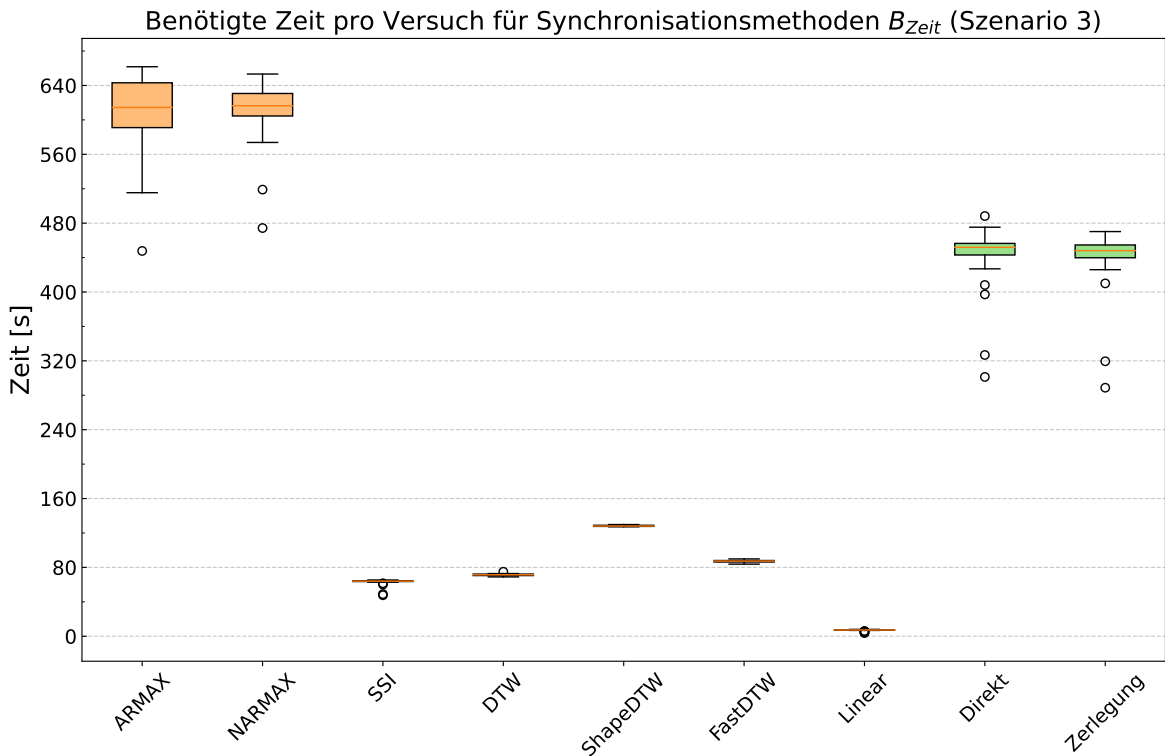


Abbildung 5.24: Darstellung der benötigten Zeit (B_{Zeit}) der einzelnen Methoden im Szenario 3.

die sich danach innerhalb des Mappings stabilisieren. Dies deutet darauf hin, dass die Suche nach dem ersten Verschiebungspunkt den Rest der Synchronisierung dominiert. Zuletzt zeigt die Methode der *Linearisierung* im Mittel eine deutliche Verbesserung der Qualität, wenn auch nicht bei allen Versuchen.

Bezogen auf die benötigte Zeit zeigen die Methoden *ARMAX* und *NARMAX* die höchsten Werte im Diagramm 5.26 mit Werten von mehreren Minuten. Die neuen Methoden benötigen im Normalfall etwas weniger Zeit bei gleicher Qualität, was auf die optimierte Reihenfolge zurückführbar ist. Die deutlich schnellste Methode ist auch hier die *Linearisierung*, während die anderen Methoden bei 20 bis 70 Sekunden liegen.

Das Szenario 4 umfasst idealisierte Verläufe ohne Jitter oder Sprünge in den Werten, allerdings mit einer leichten Verzögerung. Dies stellt beim Einziehen der Asynchronizitätseffekte mehrere Methoden vor schwere Herausforderungen. Da kein Wissen zu dieser Verzögerung existiert oder in den Methoden eingegeben werden kann, bleiben die meisten Methoden entsprechend begrenzt nutzbar für die Synchronisierung. Lediglich die *Linearisierung* geht

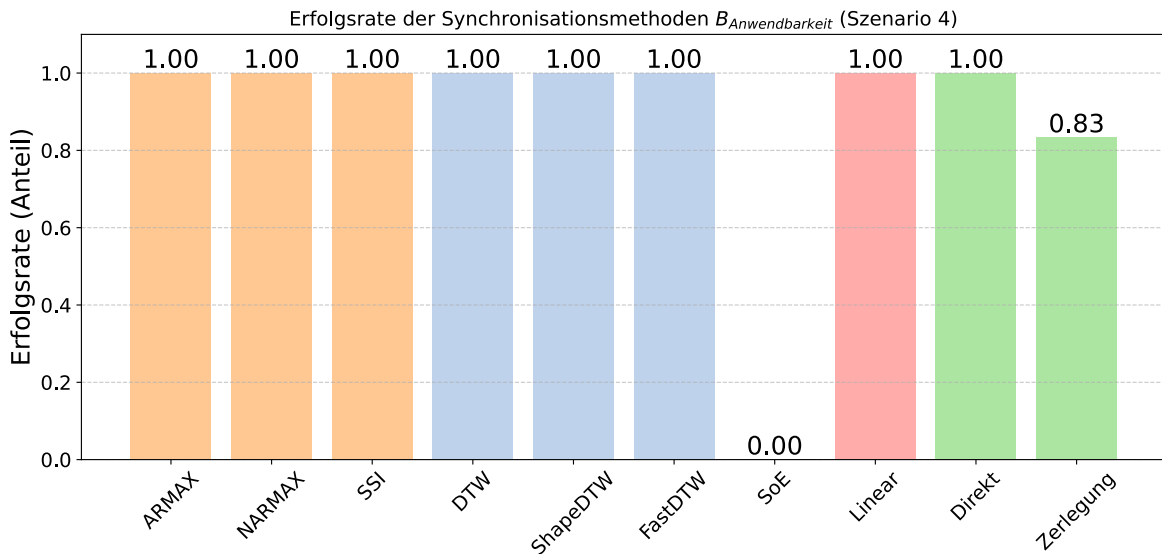


Abbildung 5.25: Darstellung der Erfolgsquoten ($B_{Anwendbarkeit}$) der einzelnen Methoden im Szenario 4. Ein Wert von 0 beschreibt eine Methode, welche nicht angewendet werden konnte, oder bei jeder Durchführung mit einem Fehler endete.

hier unabhängig vor und schafft in den meisten Fällen eine deutliche Verbesserung der Datenqualität und stellt bei dem Szenario die beste Option dar. Die neuen Methoden sind qualitativ im gleichen Bereich wie einige der anderen Methoden und können nicht konsistent bessere Werte erreichen. Aufgrund von falsch detektierten Fixpunkten kommt es bei der *Zerlegung* sogar vereinzelt zu Abbrüchen.

Szenario 5

In diesem Szenario existieren nicht für alle Systeme und Datenflüsse auf dem Weg durch das Produktionsnetzwerk eigenständige Zeitangaben, daher ist die Methode *Sequence of Events* nicht anwendbar und erhält eine entsprechend niedrige Bewertung für $B_{Anwendbarkeit}$, wie in Diagramm 5.28 dargestellt. Für die meisten anderen Methoden ist der Aufbau und der Verlauf der Signale ausreichend, sodass die Synchronisierungen durchgeführt werden können. Lediglich bei *DTW* kommt es stets zu einem Fehler, dass kein Warping Pfad gefunden wird.

Bezogen auf die Qualität können mehrere Methoden oftmals eine Verbesserung der Datenqualität erreichen, wie das Diagramm 5.29 zeigt. Bis auf einzelne Fälle schafft *SSI* eine

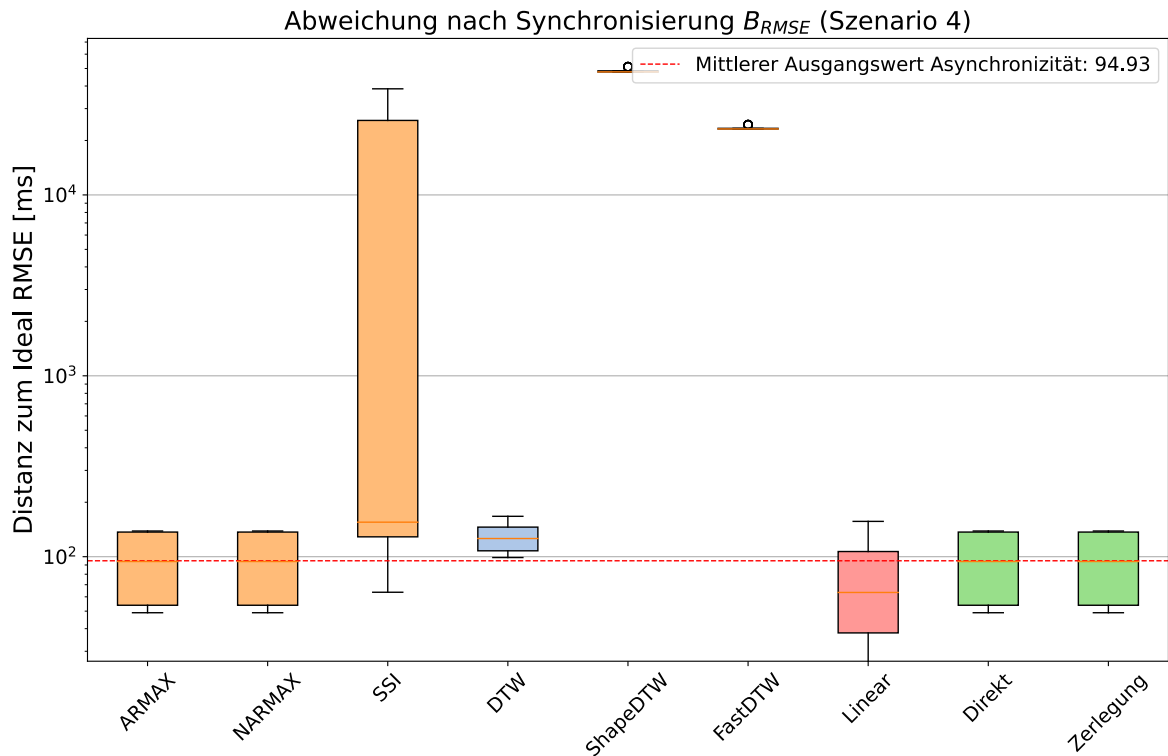


Abbildung 5.26: Darstellung der Qualität (B_{RMSE}) der einzelnen Methoden im Szenario 4. Die gestrichelte rote Linie zeigt die Abweichung von der Grundwahrheit nach der Anwendung von Asynchronizitätseffekten.

Verbesserung der Datenqualität. Die neuen Methoden sowie *ARMAX* und *NARMAX* erreichen vergleichbare Ergebnisse, wobei *Direkt* leicht besser und *Zerlegung* leicht schlechter ist. Die anderen Methoden sind in Relation stabiler, schaffen allerdings keine Verbesserungen der Datenqualität. Bei der Linearisierung kommt hier insbesondere die nicht-konstante Frequenz der Daten zu Tragen, da so nahezu jeder Datenpunkt mehrfach verschoben wird. Die konstant größten Verschlechterungen verursacht *ShapeDTW* im Bereich von mehreren Sekunden.

Die benötigte Zeit verteilen sich die Methoden stärker, wie Diagramm 5.30 zeigt. Erneut sind *ARMAX* und *NARMAX* am oberen Ende der Verteilung aufzufinden. Bei vergleichbaren Ergebnissen sind die neuen Methoden um mehrere Sekunden besser. Hier spart auch die *Zerlegung* weitere Sekunden im direkten Vergleich ein. Während die *Linearisierung* erneut im Wertbereich um eine Sekunde ist, finden sich die verbleibenden Methoden zwischen 6 und 10 Sekunden. Alle Methoden zeigen dabei auch eine geringe Bandbreite.

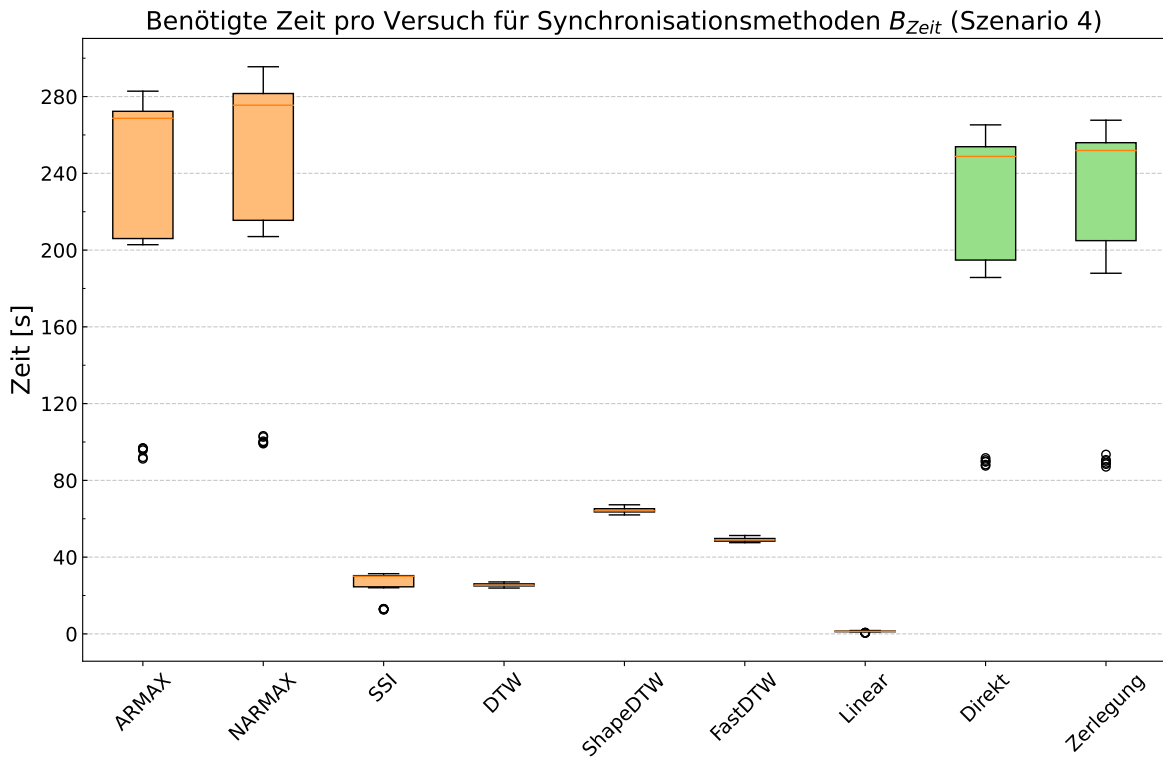


Abbildung 5.27: Darstellung der benötigten Zeit (B_{Zeit}) der einzelnen Methoden im Szenario 4.

Im Szenario 5 sind mit Ausreißern und nicht-konstanten Frequenzen Gegebenheiten anzutreffen, welche einige der Methoden vor Herausforderungen stellen. Die Methode *SSI* schafft die besten Ergebnisse und steckt bei der dafür benötigten Zeit lediglich hinter der *Linearisierung* zurück, welche im Gegenzug konsistent schlechte Ergebnisse liefert. Für dieses Szenario kann durch Nutzung von Synchronisierungsmethoden oftmals eine Verbesserung der Datenqualität erreicht werden. Bei den neuen Methoden fällt die Qualität der *Zerlegung* hinter der Alternative zurück, spart dafür etwas mehr Zeit ein. Beide Varianten reichen aber nicht an die Performanz von *SSI* heran.

Szenario 6

Im Szenario 6 ist eine Menge von Signalen betrachtet, welche teilweise nur begrenzte Veränderungen darstellen. Bei der Durchführung kommt es zu Abbrüchen und Fehlern für die Methoden *DTW*, *ShapeDTW* und *SoE*. Bei den verbleibenden Methoden, so zeigt das Diagramm 5.31, kann die Synchronisierung abgeschlossen werden.

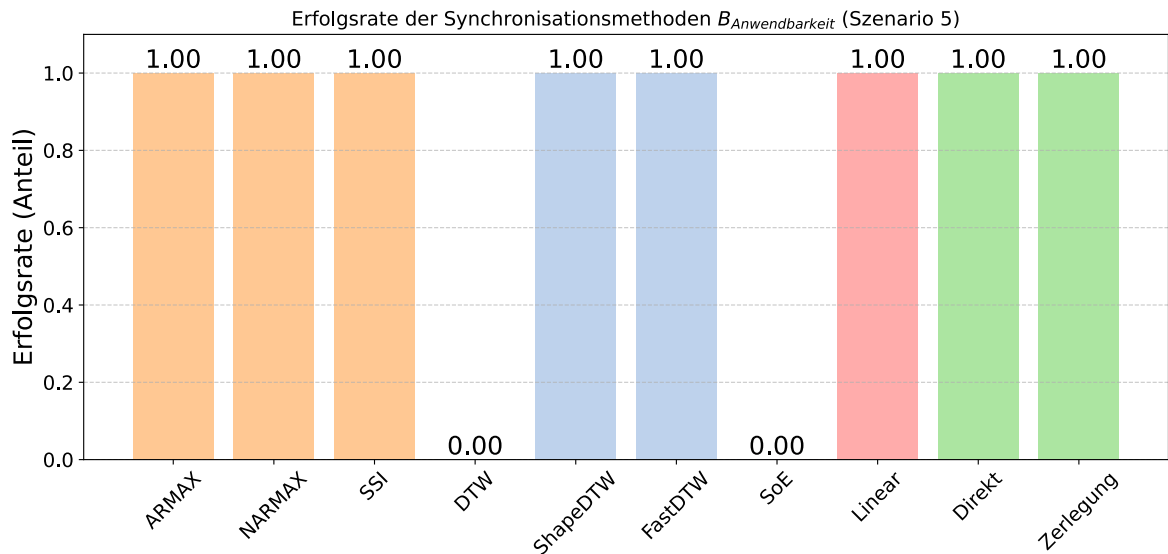


Abbildung 5.28: Darstellung der Erfolgsquoten ($B_{Anwendbarkeit}$) der einzelnen Methoden im Szenario 5. Ein Wert von 0 beschreibt eine Methode, welche nicht angewendet werden konnte, oder bei jeder Durchführung mit einem Fehler endete.

Die Durchführung mittels *FastDTW* zeigt konsistent schlechte Ergebnisse in Bezug auf die Qualität und beschreibt das obere Ende der Abweichungen im Diagramm 5.32. Bei allen anderen erfolgreich genutzten Methoden sind die Ergebnisse ähnlich zueinander. Bei all diesen Methoden verteilen sich die Werte nahe um die Ausgangswerte der Asynchronizität. Es gibt kleinere Abweichungen, sodass der Median bei *SSI* leicht besser ist, während die Spannweite bei der *Linearisierung* etwas geringer ist. Signifikante Unterschiede sind aber nicht erkennbar.

Erneut ist die benötigte Zeit, dargestellt im Diagramm 5.33 bei *ARMAX* und *NARMAX* höher als bei den anderen Methoden und liegt bei 7 bis 8 Minuten pro Versuch. Die neuen Methoden sparen demgegenüber wenige Minuten ein. Da mehrere Kandidaten für Fixpunkte überprüft, aber keine genutzt werden, erhöht sich die benötigte Zeit für *Zerlegung* etwas im Vergleich. Die weiteren Methoden bedürfen deutlich weniger Zeit und zeigen auch weniger Spannweite der Werte auf. Am schnellsten ist die *Linearisierung* mit wenigen Sekunden pro Versuch.

In diesem Szenario sind nur einige der Methoden anwendbar, allerdings bei den Ergebnissen sind die Unterschiede sehr gering. Gleichzeitig sind aber die Unterschiede bei der dafür benötigten Zeit deutlich verschieden. Bei den Ergebnissen ist die Verbesserung minimal,

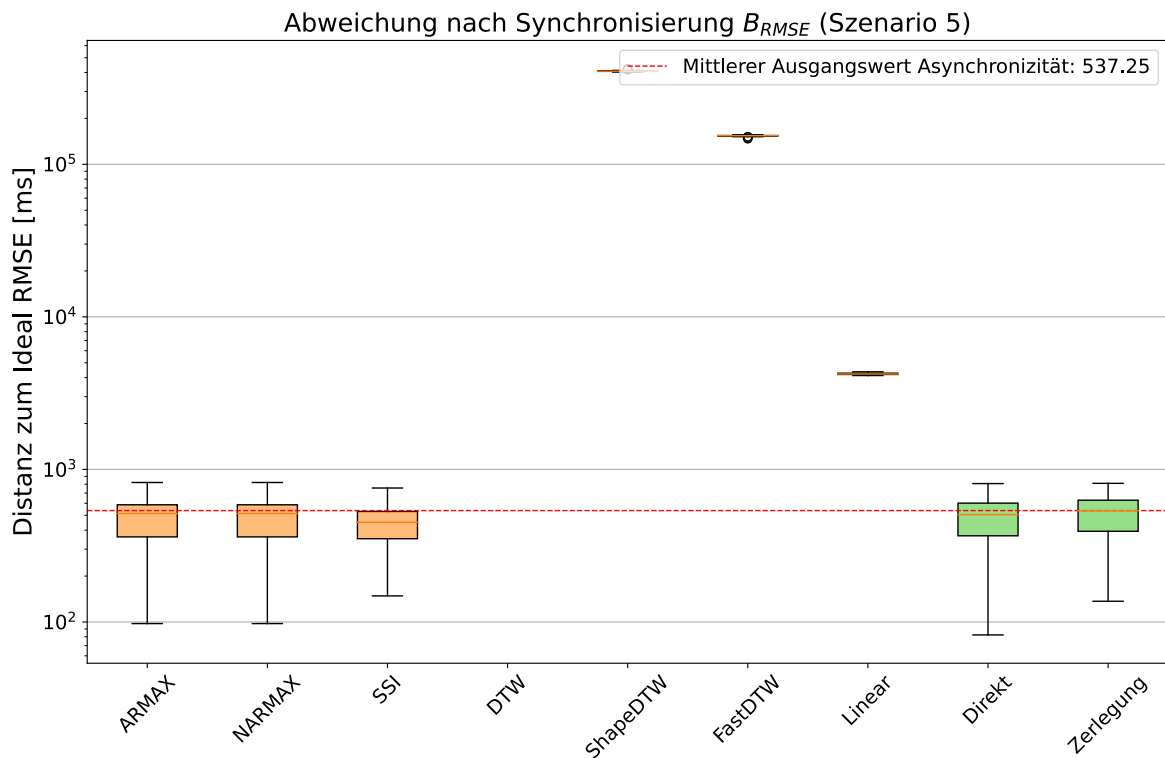


Abbildung 5.29: Darstellung der Qualität (B_{RMSE}) der einzelnen Methoden im Szenario 5. Die gestrichelte rote Linie zeigt die Abweichung von der Grundwahrheit nach der Anwendung von Asynchronizitätseffekten.

sodass der Aufwand der Synchronisierung für dieses Szenario zu hinterfragen ist. Die neuen Methoden schließen dabei vergleichbar zu den anderen Methoden ab und sind schneller als *ARMAX* und *NARMAX*, aber langsamer als die anderen Methoden bei gleichen Ergebnissen.

Szenario 7

Im Szenario 7 sind mehrere hochfrequente Zeitdatenreihen betrachtet, wobei die Ausreißer und Verläufe nur teilweise im Zusammenhang stehen. Da nicht alle Systeme und Gateways im Produktionsnetzwerk eigene Zeiten und Charakterisierungen haben, kann *SoE* nicht angewendet werden, was zu einer entsprechenden Bewertung von 0 führt. Bei der Nutzung von *DTW* bricht die Durchführung nach einer Weile ab mit einem Fehler, da kein Warping Pfad gefunden wird.

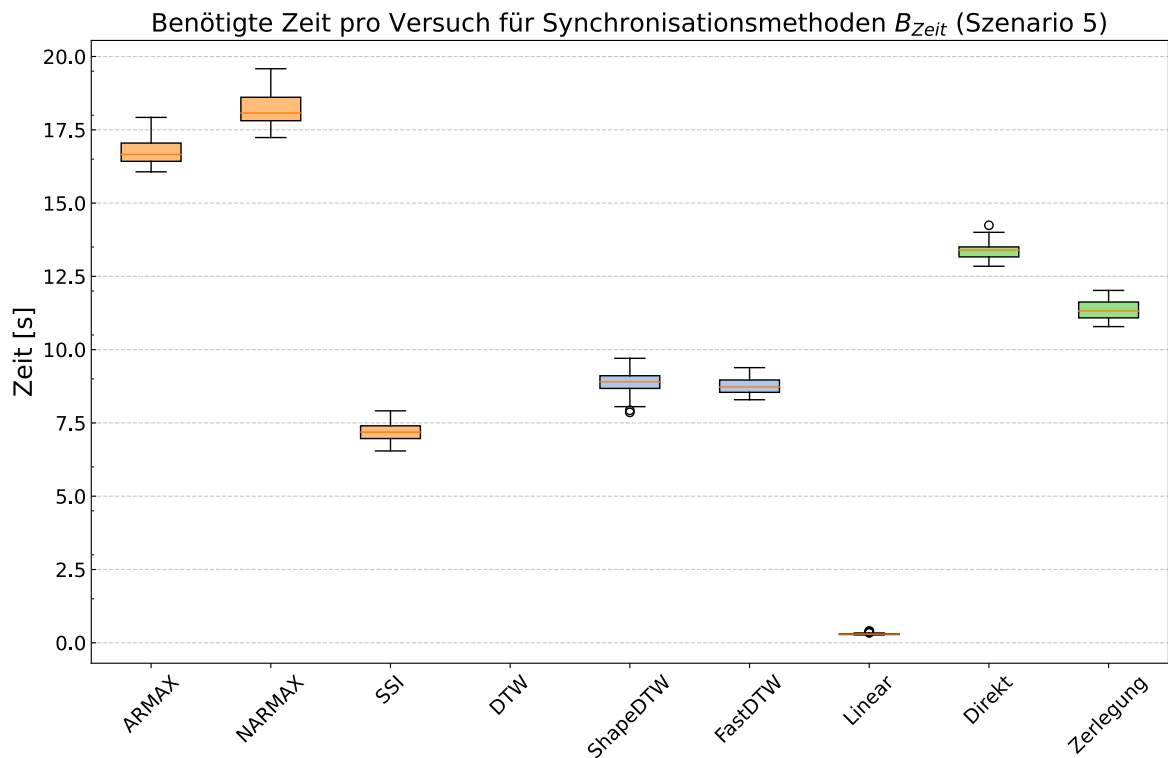


Abbildung 5.30: Darstellung der benötigten Zeit (B_{Zeit}) der einzelnen Methoden im Szenario 5.

Die resultierende Abweichung nach Synchronisierung ist in Diagramm 5.35 dargestellt und zeigt ein breites Spektrum. Die besten Ergebnisse liefern die Methoden aus dem Bereich System Identification, welche sich um den Ausgangswert der Asynchronizität verteilen und teilweise für Verbesserungen sorgen. Unter anderen aufgrund der nicht-konstanten Frequenz ist die *Linearisierung* konsistent eine Verschlechterung der Datenqualität. Alle weiteren Methoden zeigen noch weitergehende Verschlechterung der Datenqualität. Dabei liegen die Ergebnisse der neuen Methoden zwischen der *Linearisierung* und den Time Warping Ansätzen.

Mit Zeitbedarfen um die 35 Minuten zeigt das Diagramm 5.36 den weiten Abstand von *ARMAX* und *NARMAX* zu allen anderen Methoden. Am anderen Ende der Betrachtung sind die *Linearisierung* und die neuen Methoden mit wenigen Sekunden. Die verbleibenden Methoden liegen zwischen 3 und 6 Minuten. Die Spannweite ist bei allen Methoden gering.

Insgesamt ist die Mischung aus verschiedenen Eigenschaften und mangelnden Zusammenhängen eine Herausforderung für die Synchronisierung. Die Methoden der System

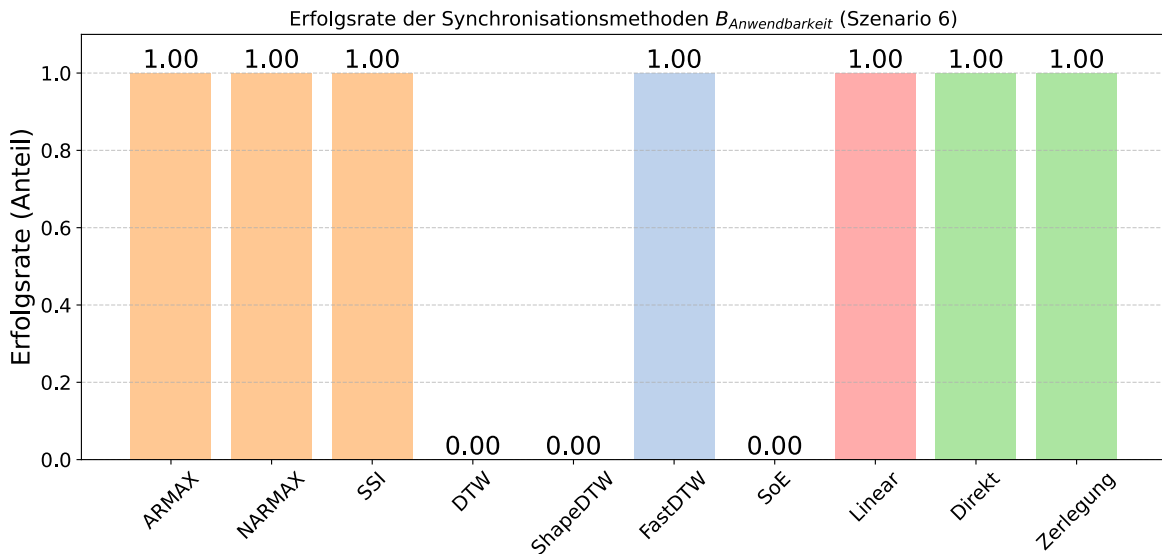


Abbildung 5.31: Darstellung der Erfolgsquoten ($B_{Anwendbarkeit}$) der einzelnen Methoden im Szenario 6. Ein Wert von 0 beschreibt eine Methode, welche nicht angewendet werden konnte, oder bei jeder Durchführung mit einem Fehler endete.

Identification zeigen die besten Ergebnisse, die allerdings auch nicht konsistent zu Verbesserungen führen. Die Methode *SSI* schafft diese Qualität dabei in einem Bruchteil der Zeit wie bei anderen. Beide neuen Methoden zeigen hohe Geschwindigkeit, aber mäßige Ergebnisse. Abschließend ist bei diesem Szenario eine Synchronisierung mittels *SSI* teilweise denkbar, aber bei den meisten Methoden eine Verschlechterung wahrscheinlich.

5.3 Kritische Diskussion der Ergebnisse

Critical Discussion of the Results

In diesem Abschnitt findet zunächst eine Diskussion der Ergebnisse statt. Im Anschluss werden die Forschungsfragen dieser Arbeit betrachtet und geprüft, wie weit auf diese Fragen Antworten gegeben werden können.

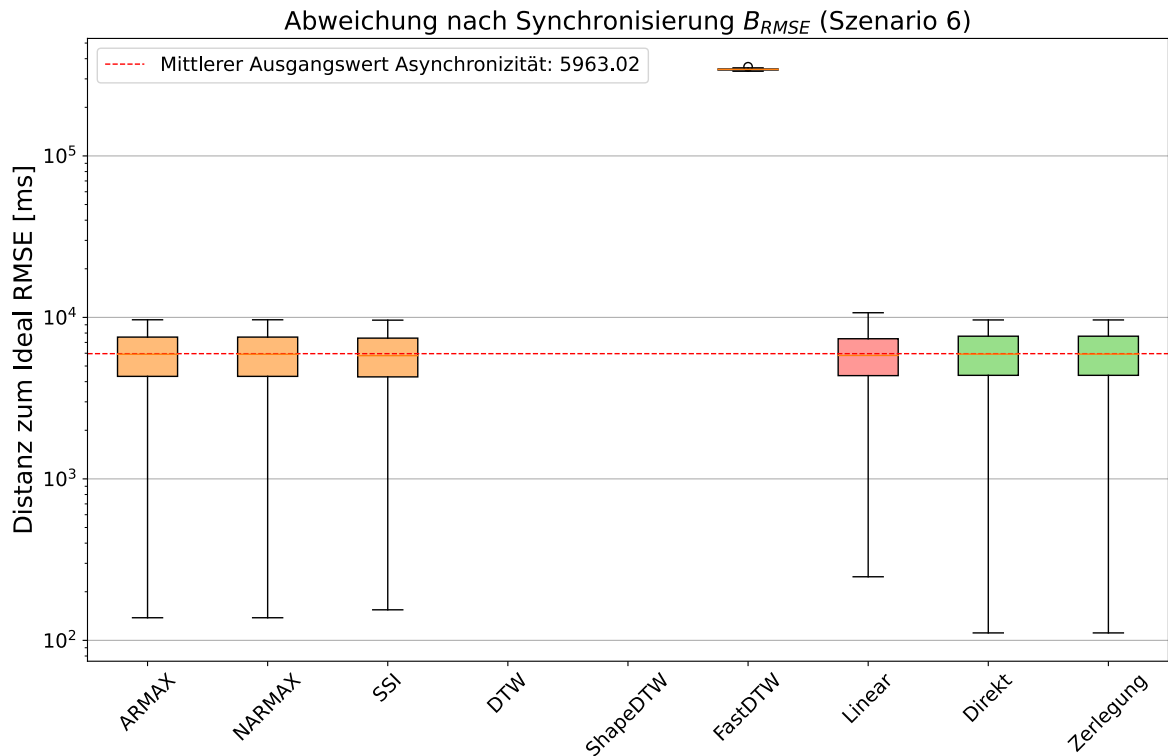


Abbildung 5.32: Darstellung der Qualität (B_{RMSE}) der einzelnen Methoden im Szenario 6. Die gestrichelte rote Linie zeigt die Abweichung von der Grundwahrheit nach der Anwendung von Asynchronizitätseffekten.

5.3.1 Ergebnisdiskussion

Result Discussion

Die Ergebnisse der Verifikation und der Validierung geben ein breites Bild der Eigenschaften des vorgestellten Offline Synchronisierungssystems und der Vergleichsmethoden. Bezogen auf einzelne Methoden zeigen sich einige auffällige Verhaltensweisen, welche das neue Synchronisierungssystem entsprechend aufgreift und selbst davon beeinflusst ist. Die Methoden *ARMAX* und *NARMAX* zeigen minimale bis keine Unterschiede in den Ergebnissen. Selbst im Falle von nicht-linearen Eigenschaften der Daten und Asynchronizitäten bleibt der Unterschied entgegen der Erwartung nicht signifikant. Gleichzeitig zeigen diese Methoden oftmals gute Ergebnisse, dominieren aber üblicherweise bei der benötigten Zeit. Dies führt einerseits zu einer häufigen Auswahl in den neuen Methoden von jenen Methoden auf Kosten einer erhöhten Rechenzeit. Im Ergebnis deutet dies darauf hin, dass nur eine der beiden Methoden verwendet werden kann anstatt beider Optionen im Synchronisierungssystem.

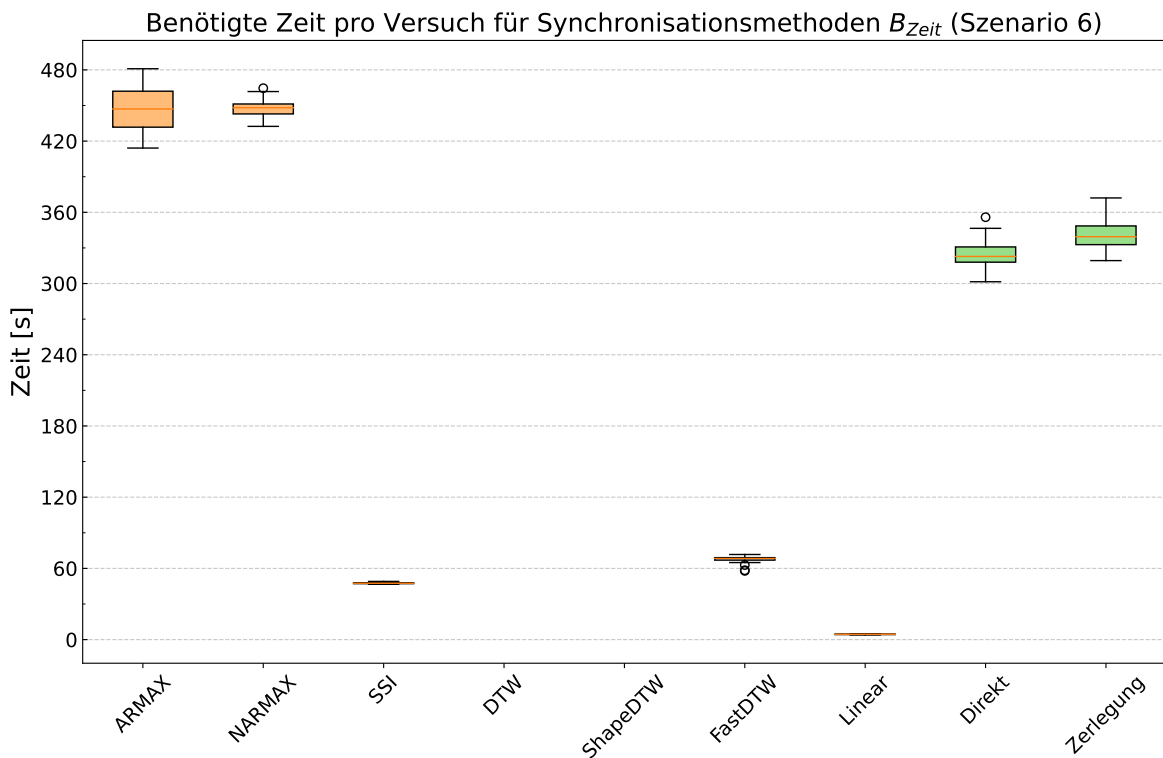


Abbildung 5.33: Darstellung der benötigten Zeit (B_{Zeit}) der einzelnen Methoden im Szenario 6.

Grundsätzlich zeigen die Variationen *Direkt* und *Zerlegung* bei den Szenarien geringe Abweichungen in der Qualität. Ist allerdings eine Zerlegung in Subsequenzen möglich, so wird die eigentliche Berechnungszeit signifikant reduziert. Auch die Auswahl der Reihenfolge der durchzuführenden Synchronisierungen zeigt einen Einfluss auf die Qualität und die Berechnungszeit, sodass die Variationen hierdurch Vorteile erreichen können gegenüber einer direkten Anwendung. Kann innerhalb eines Teilbereichs des Produktionsnetzwerks eine bessere Methode ausgewählt werden, so lässt sich die Qualität signifikant erhöhen, wie insbesondere das Szenario 3 zeigt.

Die Suche nach Fixpunkten führt in einigen Fällen zu Ergebnissen, bringt allerdings selbst eine starke Abhängigkeit von der Präzision des dafür genutzten Wissens mit sich. Ist der mittlere zeitliche Abstand zwischen Elementen gering, so müssen die entsprechenden Wissens Elemente im gleichen Rahmen präzise sein. Ist ein Wissen wesentlich unpräziser, so entsteht die Gefahr, dass ein Fixpunkt um mehrere Elemente versetzt identifiziert wird. Dies kann im Umkehrschluss auch zu einer Verschlechterung der Qualität führen. Insbesondere

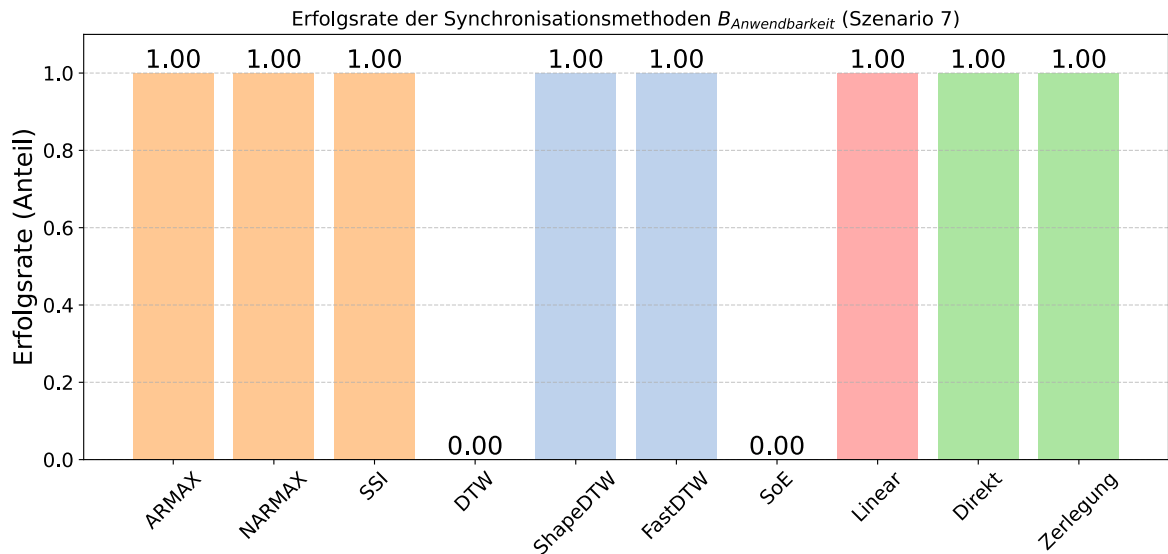


Abbildung 5.34: Darstellung der Erfolgsquoten ($B_{Anwendbarkeit}$) der einzelnen Methoden im Szenario 7. Ein Wert von 0 beschreibt eine Methode, welche nicht angewendet werden konnte, oder bei jeder Durchführung mit einem Fehler endete.

für hochfrequente Daten in Verbindung mit einer manuellen Markierung von Zusammenhängen ist dies ein zentrales Problem und offene Forschungsfrage. [KEOG05][WU21].

In der Theorie ist die Nutzung von *ShapeDTW* für die Beibehaltung von Strukturen in den Daten geeignet, zeigt allerdings bei den Szenarien teilweise erhebliche Probleme. In den Fällen, wo diese Methode zum Einsatz kommt, durch Auswahl im G_{sync} , ist die Qualität ebenfalls schlecht. Daraus ergibt sich die Empfehlung, diese Methode nicht weiter zu betrachten und aus dem Synchronisierungssystem zu entfernen.

Trotz Nutzung von Optimierungen wie dem Sakoe-Chiba-Band kann bei den Varianten des *Dynamic Time Warping* teilweise eine große Abweichung und resultierend schlechte Synchronisierungen beobachtet werden. Dieses Verhalten resultiert insbesondere beim Versuch, in Daten mit Jitter und Sprüngen einen minimalen Pfad zu finden. Dieser Effekt wird in der Literatur auch als »Pathological Warping« bezeichnet [ZHAN17]. Dieser Effekt ist insbesondere bei der Variation *FastDTW* in manchen Szenarien (z.B. Szenario 4) zu erkennen, da die Beschränkung hier anders funktioniert und weniger Stabilität gegen diese Effekte bietet. Gleichzeitig schafft *FastDTW* eine bessere Anwendbarkeit und wird daher bei ansteigender Datenmenge gegenüber *DTW* vom Synchronisierungssystem bevorzugt. Grundsätzlich ist die Vorbedingungen zur Anwendbarkeit in Form von ähnlichen Verläufen

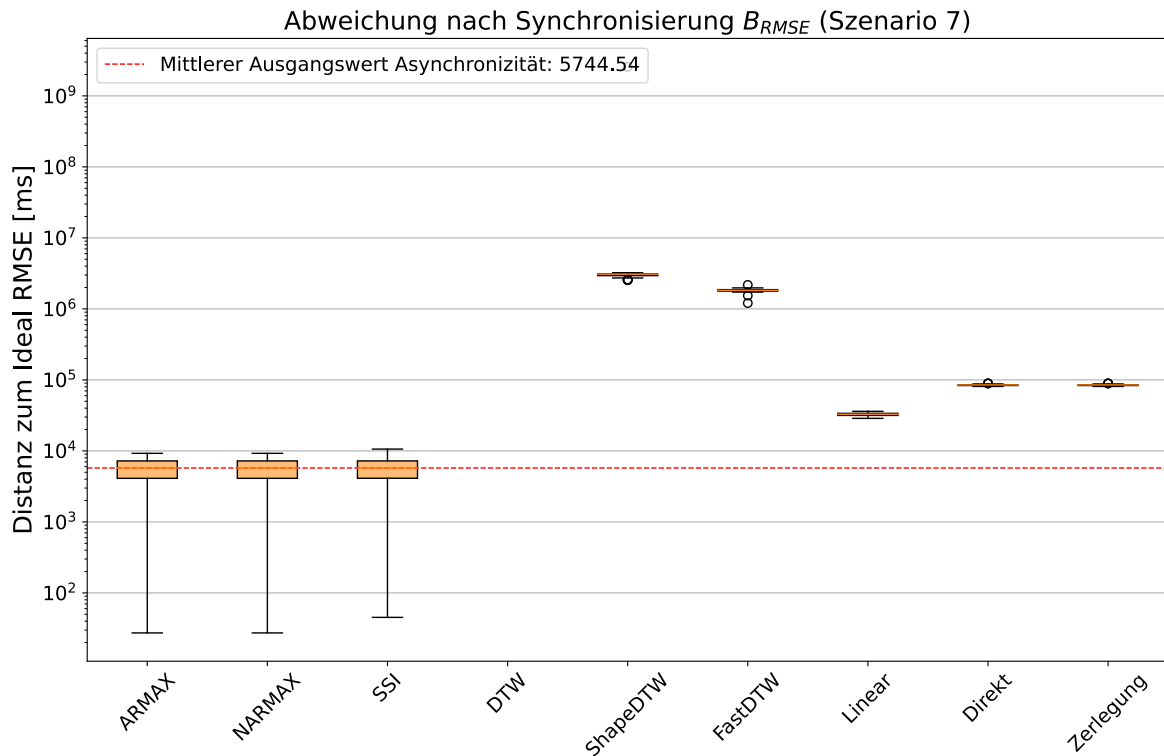


Abbildung 5.35: Darstellung der Qualität (B_{RMSE}) der einzelnen Methoden im Szenario 7. Die gestrichelte rote Linie zeigt die Abweichung von der Grundwahrheit nach der Anwendung von Asynchronizitätseffekten.

erkennbar, was die Auswahl im Synchronisierungssystem nur bei vorhandenem Wissen dazu empfiehlt.

Steigt die Menge von Zeitdatenreihen an, so finden mehr Synchronisierungen statt. Anstatt einer paarweisen Synchronisierung führt das Synchronisierungssystem hier eine gestaffelte Synchronisierung und damit eine Komposition der Projektionsfunktionen durch. Dies hat allerdings den Effekt, dass Fehler in den Synchronisierungen sich fortpflanzen. Je länger eine Komposition ist, desto anfälliger ist die finale Synchronisierung für diese Fortpflanzung sowie einzelne schlechte Synchronisierungen.

Mithilfe der statistischen Versuchsplanung konnten die verschiedenen Effekte der Asynchronizität angewendet und untersucht werden. Dabei kann beobachtet werden, dass bei einer linearen Asynchronizität (einfacher Drift) in der Regel eine Verbesserung durch die *Linearisierung* und andere Methoden erreicht wird bei kurzer Berechnungszeit. Ist also für einen Produktionsdatensatz abschätzbar, dass keine Sprünge oder Stufen vorliegen

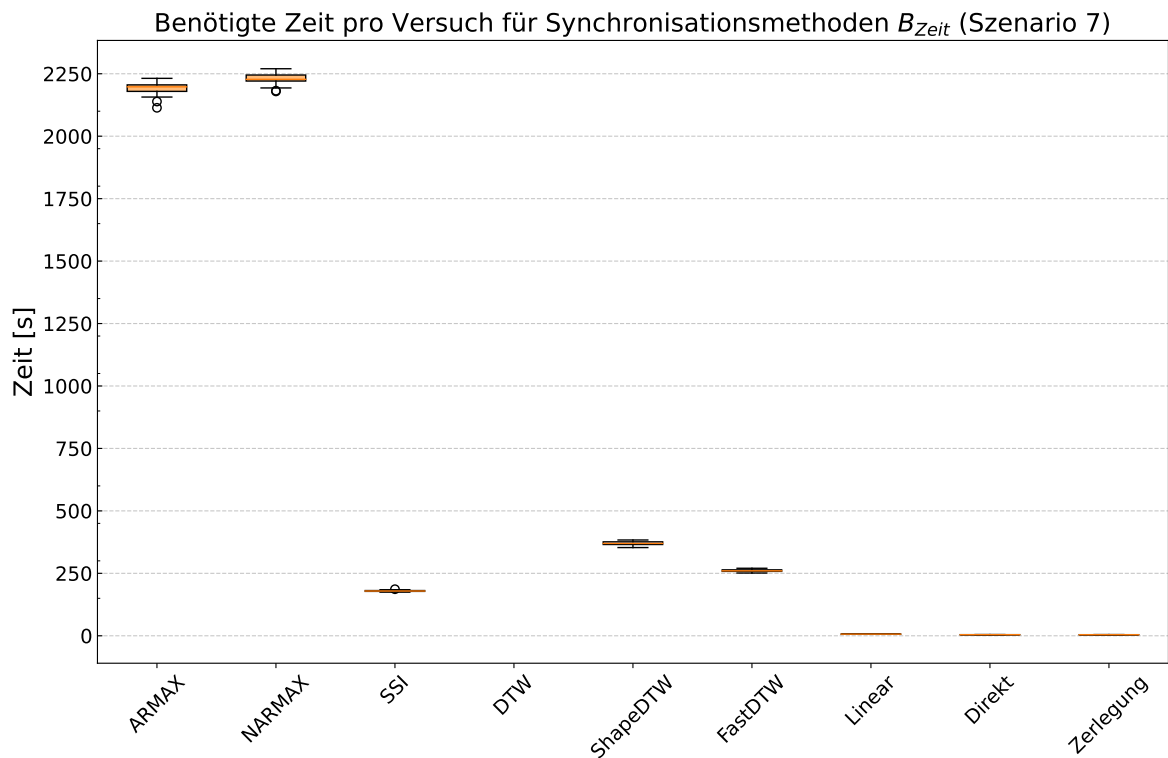


Abbildung 5.36: Darstellung der benötigten Zeit (B_{Zeit}) der einzelnen Methoden im Szenario 7.

und zugleich der lineare Drift als dominant angenommen werden kann, so bleibt die Linearisierung die favorisierte und geeignete Methode der Synchronisierung. Zumindest eine Prüfung auf die Sprünge und Stufen kann auch automatisch durchgeführt werden. Eine Prüfung auf Nicht-Linearität kann nicht im Allgemeinen durchgeführt werden, sodass das Problem weiterhin besteht und die Linearisierung nicht als allgemeines Mittel genutzt werden kann.

In der dargestellten Validierung kam ein System zum Einsatz, welches über große Rechen- und Speicherressourcen verfügte. Da die pure Rechenzeit in einigen Fällen auch die Betrachtungszeit des Datensatzes überstieg, zeigt sich die Notwendigkeit von Optimierungen und Grenzwerten, wie sie als Parameter in der neuen Methodik integriert sind. Die idealen Werte dieser Parameter sind hier allerdings von der Umgebung und dem Ziel abhängig, sodass die hier eingesetzten Parameter einer Einzelaufnahme entsprechen und eine Bestimmung dieser Parameter selbst eine Herausforderung darstellt.

Letztlich versuchen alle dargestellten Methoden eine nachträgliche Synchronisierung und

Verbesserung der Datenqualität. Aufgrund der Natur dieser Aufgabe kann keine Methode garantieren, dass das Ergebnis in jedem Fall besser als zuvor ist. Alle Methoden nutzen daher die Grundannahme, dass eine Asynchronizität vorliegt und versuchen eine Verbesserung zu finden. Auch die neue Methodik kann hier keine Garantien liefern, was in der Praxis auch zu Verschlechterungen führen kann.

Abschließend bedeuten diese Ergebnisse, dass die vorgestellte Methodik einige Vorteile mit sich bringt, aber die Grundprobleme der Synchronisierung wie mangelnde Garantien und Grundwahrheiten nicht lösen kann. Auch ist der Ansatz der Zerlegung theoretisch funktional, aber in der Praxis von dem Szenario und der möglichen Präzision des genutzten Wissens und Informationen abhängig. Insbesondere im Bereich der Batteriezellfertigung ist die Skalierung und Komplexität gegeben und kann hier teilweise adressiert werden. Die Tatsache, dass in der Batteriezellfertigung insbesondere die Ursache-Wirkzusammenhänge umfangreich und zugleich wenig erforscht sind, schränkt diese Methode hier zunächst stärker ein. Gleichzeitig führen neue Erkenntnisse in diesem Feld und der Formalisierung davon möglicherweise zu weiteren Verbesserungen der Synchronisierung. Dies bedeutet, dass beide Variationen der neuen Methodik anwendbar sind, aber alleine keine finale Lösung des Problems der Asynchronizität in der Batteriezellfertigung darstellen.

6 Fazit

Conclusion

In diesem Kapitel findet zunächst eine Betrachtung der Forschungsfragen und die Beantwortung auf Basis der dargestellten Ergebnisse statt. Anschließend findet der Ausblick für Forschungsfelder statt, die sich aus dieser Arbeit und den Erkenntnissen ableiten lassen. Abschließend gibt es eine kurze Zusammenfassung der Arbeit und der Ergebnisse.

6.1 Betrachtung Forschungsfragen

Research Questions Revisited

Mit den Ergebnissen der Verifikation und der Validierung lassen sich die Forschungsfragen aus Abschnitt 1.2 beantworten. Dazu werden zunächst die Unterforschungsfragen einzeln betrachtet und eine Einschätzung dargestellt. Abschließend wird die Hauptforschungsfrage dieser Arbeit betrachtet und eine Antwort gegeben.

UFF 1: Welche Bewertungsmetriken für eine Offline Synchronisierung können abgeleitet werden?

Für eine Offline Synchronisierung wurden mehrere Metriken in den Clustern Qualität, Anwendbarkeit und Effizienz hergeleitet. Diese sind in Abschnitt 3.2 detailliert beschrieben, ebenso die dafür genutzten Bewertungsstufen. Die Tatsache, dass alle bestehenden Methoden mit diesen Metriken bewertet werden konnten, ebenso wie die in dieser Arbeit neu entwickelte Methodik zeigt die Nutzbarkeit eben jener Metriken. Insgesamt kann diese Unterforschungsfrage damit als beantwortet angesehen werden.

UFF 2: Wie lässt sich die temporale Ungenauigkeit der Daten in der Batterieproduktion beschreiben und modellieren?

Für die Modellierung und Darstellung sind in dieser Arbeit zwei Abschnitte maßgeblich. Für die Modellierung auf einer mathematisch-theoretischen Seite kommen die Definitionen aus Kapitel 3 zum Einsatz. Dies betrifft insbesondere die Arten der Asynchronizität δ und Δ_{skew} sowie der daraus resultierenden Projektionsfunktionen Φ . Daneben kommt noch die experimentelle Auswertung von Asynchronizitätseffekten hinzu, welche in Abschnitt 4.2.2 methodisch beschrieben und zu den Mustern in Abbildung 5.5 geführt hat. Diese Modellierungen haben eine Implementierung im Versuchsaufbau geführt, wodurch die Validierung

erst stattfinden konnte. Insgesamt kann diese Unterforschungsfrage damit als beantwortet angesehen werden.

UFF 3: Wie kann Wissen zum Aufbau und Prozessverständnis bei der Offline Synchronisierung genutzt werden?

Die Unterscheidung nach Daten, Informationen und Wissen erlaubt die Aufteilung der benötigten Eingaben innerhalb des Synchronisierungssystems, dargestellt in Schaubild 4.1. Zunächst umfasst die entsprechende Nutzung die Modellierung, sodass diese automatisch genutzt werden können. Diese Modellierung ist in Abschnitt 4.2 detailliert beschrieben. Nach der Modellierung und Verfügbarmachung kommen diese Informationen und das Wissen insbesondere zum Einsatz, um eine Fixpunktzerlegung durchzuführen und um eine Empfehlung der Synchronisationsreihenfolge zusammen mit Bewertung der möglichen Methoden zu erstellen. Die Ergebnisse der Verifikation und der Validierung zeigen auf, dass diese Nutzung funktioniert und potenzielle Mehrwerte darstellt. Insgesamt kann diese Unterforschungsfrage damit als beantwortet angesehen werden.

UFF 4: Lässt sich die Offline Synchronisierung von Zeitdatenreihen für hochskalige Produktionen automatisieren?

Für die Automatisierung in der hochskaligen Produktion wurden zunächst alle benötigten Eingangsgrößen modelliert. Ein Zugriff auf diese Eingangsgrößen kann dadurch automatisch und ohne menschliche Interaktion vollzogen werden. Die eigentliche Automatisierung geschieht durch die Integration, die in Abschnitt 5.1.2 gezeigt wurde. Durch die Nutzung von technischen Schnittstellen und Integration in das Produktionsnetzwerk ist eine Automatisierung möglich. Das Gesamtsystem, dargestellt in Schaubild 4.1, umfasst auch zwei manuelle Eingaben. Da diese allerdings optional gehalten sind, können die auch abgeschaltet werden und damit eine vollständige Automatisierung erreicht werden. Im Rahmen der durchgeführten Untersuchungen ergaben sich keine Anhaltspunkte für eine Widerlegung dieser Unterforschungsfrage. Vielmehr zeigen die Ergebnisse, dass die angestrebte automatisierte Offline-Synchronisierung realisierbar ist. Insgesamt kann diese Unterforschungsfrage im Fokus dieser Arbeit damit als beantwortet angesehen werden.

HFF: Kann eine Offline Synchronisierung von Zeitdatenreihen aus der Batterieproduktion mit Wissen zum Produktionsaufbau automatisiert durchgeführt werden?

In dieser Arbeit konnte gezeigt werden, dass eine Modellierung von Informationen und Wissen für die Automatisierung möglich ist. Ebenso konnten Metriken erstellt werden, welche die Besonderheiten der Batterieproduktion berücksichtigen und zugleich eine Bewertung von Offline Synchronisierungsmethoden erlauben. Zuletzt konnte mit der Integration in das Produktionsnetzwerk der Fraunhofer FFB die Automatisierung gezeigt und Synchronisierungen entlang verschiedener Szenarien basierend auf echten Produktionsdaten durchgeführt werden. Alle Bestandteile der Hauptforschungsfrage sind damit erfolgreich beantwortet. Die Verifikation und die Validierung zeigen zudem das erfolgreiche Zusammenspiel ebenjener einzelner Komponenten. Im Rahmen der durchgeführten Untersuchungen und der Unterforschungsfragen ergaben sich keine Anhaltspunkte für negative Beantwortung der Forschungsfrage. Vielmehr zeigen die Ergebnisse, dass die angestrebte automatisierte Offline-Synchronisierung mit Wissen zum Aufbau und im Bereich der Batterieproduktion realisierbar ist. Dadurch kann insgesamt die Hauptforschungsfrage im Fokus dieser Arbeit positiv beantwortet werden.

6.2 Ausblick

Outlook

Im Rahmen dieser Arbeit und der Ergebnisse lassen sich weiterführende Forschungsfelder und Erweiterungen ableiten. Im Fokus stehen dabei zehn Themen, welche einen direkten Bezug zu dem Problem der Offline Synchronisierung oder der hier vorgestellten Methodik haben. Diese Themen sind in Darstellung 6.1 in zwei Dimensionen sortiert. Auf der Abszissenachse ist die Sortierung nach der Praxisnähe gegeben. Themen auf der linken Seite sind daher stärker grundlagenorientiert und allgemeiner gefasst, während die Themen auf der rechten Seite konkrete, spezifische und auf das Synchronisierungssystem bezogene Themen darstellen. Entlang der Ordinatenachse ist die Einschätzung des Innovationsgrads der Themen eingeordnet. Der untere Bereich beschreibt daher inkrementelle Erweiterungen und Entwicklungen auf Basis der existierenden Arbeiten. Demgegenüber entspricht der obere Bereich neuartiger Lösungen und deutliche Unterschiede zu dem aktuellen Entwicklungs- und Forschungsstand. Diese Themen sind im Folgenden kurz erläutert.

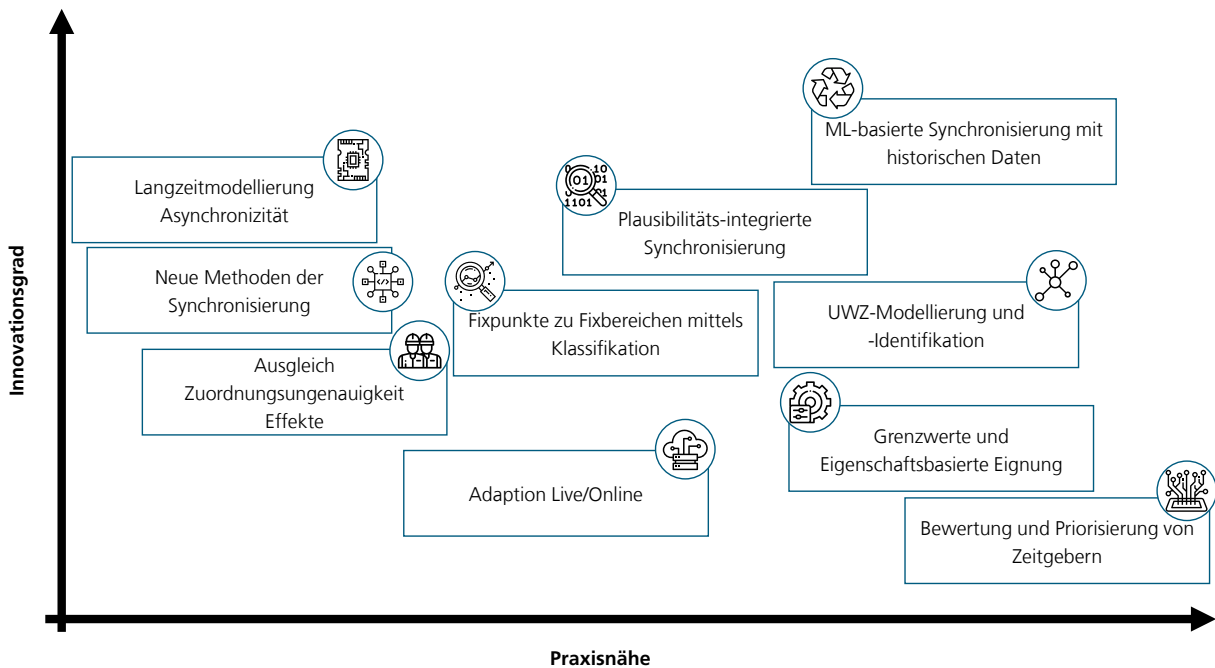


Abbildung 6.1: Darstellung der identifizierten Forschungsfelder im Ausblick dieser Arbeit. Die grobe Einordnung erfolgt nach Praxisnähe und erwartetem Innovationsgrad.

Bewertung und Priorisierung von Zeitgebern

In einem Produktionsnetzwerk mit mehreren Systemen und Zeitgebern können sich die Effekte der Asynchronizität verschieden stark ausprägen in den Systemen. Dies bedeutet, dass manche Systeme eine bessere Qualität bei den Zeitinformatoren aufweisen als andere. Eine direkte Erweiterung des aktuellen Synchronisierungssystems ist damit die Aufnahme von Informationen und Charakterisierungen der Zeitgeber der Systeme im Produktionsnetzwerk. Mit diesen Informationen kann im Rahmen der Bewertungen der Kanten in G_{sync} dann eine Bewertung der Zeitgeber erstellt und einbezogen werden. Dies ermöglicht Strategien, welche Synchronisierungen zwischen Systemen mit guten Zeitinformatoren bevorzugen und die Sprungdistanzen stärker einschränkt. Notwendig für diese Erweiterung ist die standardisierte Charakterisierung von Zeitgebern nach definierten Metriken, wie Temperaturstabilität, Art des Zeitgebers, oder Alter. Der Aufbau eines solchen Charakterisierungssystems kann die Qualität der Synchronisierung des Synchronisierungssystems erhöhen.

Grenzwerte und Eigenschaftsbasierte Eignung

In der aktuellen Implementierung kommen verschiedene Metriken zum Einsatz, welche die Zeitdatenreihe selbst oder den Vergleich von zwei Zeitdatenreihen als Grundlage nehmen. In dieser Arbeit konnte gezeigt werden, dass diese Metriken genutzt werden können, um eine Aussage zur Eignung von Synchronisierungsmethoden zu treffen. Dennoch kommen in dieser Implementierung nur eine Auswahl von solchen Metriken zum Einsatz. Auf einzelne Zeitdatenreihen kann eine Erweiterung neben den bisherigen deskriptiven Maßen und Momenten weitergehende Eigenschaften untersuchen. Infrage kommen hier Korrelationsmaße, Trendmaße (z.B. Dickey-Fuller-Test [MUSH11]), spektrale Leistungsdichten und Komplexitätsmaße wie der Hurst-Exponent [CARB04]. Beim Vergleich von Zeitdatenreihen sind hier multifraktale Korrelationsmaße denkbar, da sie skalierungsübergreifende Strukturähnlichkeiten identifizieren [WANG12]. Ebenfalls sind weitere Entropie-basierte Kennwerte [ROST16] denkbar. Einige der benannten Metriken sind in anderen Domänen wie der Psychologie oder dem Aktienmarkt verbreitete Metriken, jedoch für Produktionsdaten kaum betrachtet. Ebenfalls sind diese Metriken mit hohen Laufzeiten verbunden. Eine testweise Nutzung des Hurst-Exponenten bedurfte eine höhere Laufzeit der Berechnung als alle anderen Metriken zusammen, sodass dieser in der vorliegenden Implementierung verworfen wurde. Die Nutzung von weiteren Metriken und Untersuchung, welche Grenzwerte dabei jeweils geeignet sind, welche Synchronisierungsmethoden zu empfehlen sind, kann die Qualität und Stabilität der Synchronisierung erhöhen.

Adaption Live/Online

Untersuchungsgegenstand dieser Arbeit ist die Offline Synchronisierung, bei der die Produktionsdaten im Nachhinein synchronisiert werden. Die Erkenntnisse aus dem Offline Synchronisierungssystem umfassen neben den Projektionsfunktionen auch Einschätzungen zur Asynchronizität der Systeme im Produktionsnetzwerk. Dieses Wissen könnte genutzt werden, um im Betrieb eine Live-Einschätzung der aktuellen Datenqualität zur Verfügung zu stellen. Auf Basis dieser Einschätzungen kann so auch eine Empfehlung für manuelle Synchronisierungen gegeben werden, sofern keine automatische Synchronisierungen in dem Produktionsnetzwerk möglich sind. Neben einer Nutzung der Einsichten aus dem Synchronisierungssystem kann auch das Synchronisierungssystem selbst adaptiert werden für Online-Algorithmen. Dabei findet die Synchronisierung nicht auf einem festen Datensatz statt, sondern einem konstant anwachsenden Datensatz. Auf diese Art kann eine inkrementelle Synchronisierung von großen Datensätzen ermöglicht werden, welche die aktuelle Umsetzung an die Grenzen bringt. Auch zum Zwecke der zeitnahen automatischen

Synchronisierung kann eine solche Anpassung bei Anbindung an die Datenaufnahme zusätzliche Grenzwerte für die maximalen Abweichungen direkt einbeziehen. Am Ende kann eine solche Mischform die Skalierbarkeit und Effizienz deutlich erhöhen für Produktionen, wo keine Synchronisierung der Systeme im Betrieb möglich ist.

UWZ-Modellierung und -Identifikation

Die Nutzung von Ursache-Wirkzusammenhängen ist in dem vorgestellten Synchronisierungssystem ein zentraler Bestandteil. Innerhalb der Batterieproduktion sind diese Zusammenhänge allerdings weitgehend unbekannt und nicht quantifiziert [FITZ23]. Ebenfalls ist die maschinenlesbare Modellierung derartiger Zusammenhänge nicht gelöst. In dieser Arbeit ist die Modellierung selbst einfach gehalten und unterstützt nur einfache Zusammenhänge. Ein Vorgehen zur Identifizierung und Quantifizierung von derartigen Wirkzusammenhängen ist nicht nur für die Synchronisierung, sondern alle datengetriebenen Methoden in der Batterieproduktion von hoher Relevanz. Mit Blick auf die Notwendigkeit der Skalierung auch dieser Methoden ist auch eine maschinenlesbare Modellierung von einfachen und komplexen Zusammenhängen von entscheidender Bedeutung. Daraus ergibt sich dieses Themenfeld als kritisch und hoher Bedeutung für verschiedene Domänen. Hierbei können Technologien wie Ontologien (beispielsweise BattINFO [CLAR22]) und Anlagenstandards (beispielsweise OPC UA Companion Specifications) eine Grundlage liefern. Diese Technologien mit einer Methodik und explorativen Versuchen zusammenzubringen ist damit ein hochrelevantes Forschungsfeld.

Ausgleich Zuordnungsungenauigkeit Effekte

Soll das Wissen über Wirkzusammenhänge genutzt werden, so ist die entsprechende Zuordnung eines Zeitpunkts als Ursache und der Auswirkung in den Daten notwendig. Wie KEOGH zeigt, ist diese Zuordnung selbst aber nicht trivial und bei der notwendigen Genauigkeit letztlich nicht mit traditionellen Methoden nutzbar [KEOG05]. Dies ist ein grundsätzliches Problem und gewinnt an Komplexität bei Erhöhung der Frequenz der Datenaufnahme. Die Erhöhung der Genauigkeit und der Konsistenz bei der Zuordnung von Zeitpunkten der Ursachen und Auswirkungen in Zeitdatenreihen beeinflusst damit direkt andere Themenfelder wie die oben benannte Identifikation von Ursache-Wirkzusammenhängen, aber auch die Nutzbarkeit des entsprechenden Wissens in der Offline Synchronisierung. Die Nutzung von künstlichen Mustern und Hilfsmitteln könnte hier ein Lösungsansatz sein, befindet sich allerdings in einer grundlagenorientierten Phase, sodass hier umfangreiche Forschung

notwendig ist. Welche Genauigkeit für die benannten Themen notwendig ist, ist dabei selbst noch ein Forschungsgegenstand, der adressiert werden muss.

Fixpunkte zur Fixbereichen mittels Klassifikation

Eine Innovation dieser Arbeit ist die Identifikation von Fixpunkten in den Zeitdatenreihen, entlang dieser eine Zerlegung möglich ist, da diese Punkte sicher gleichzeitig stattgefunden haben. In der Validierung hat sich gezeigt, dass die punktgenaue Zuordnung nicht immer möglich ist und teilweise die eigentliche Gleichzeitigkeit zwischen zwei Elementen liegen kann. Ein aktives Forschungsfeld ist die Klassifikation von Zeitdatenreihen [FAOU22, IF19, LINE18], wo nicht einzelne Punkte, sondern komplette Sequenzen gesucht werden, welche eine temporale Verbindung zueinander haben. Das Konzept der Zerlegung kann mithilfe der Zeitdatenreihenklassifikation erweitert werden, sodass nicht einzelne Fixpunkte genutzt werden, sondern Fixbereiche. Wird ein solcher Bereich gefunden wird dann eine Zerlegung in drei Teile vollzogen: Die Sequenz vor dem Fixbereich, eine Sequenz repräsentiert durch den Fixbereich und eine Sequenz nach dem Fixbereich. In der vorliegenden Implementierung kann dies erreicht werden durch Austausch des Fixpunkt-Scanners durch einen Fixbereich-Scanner, der Anfangs- und Endpunkt des Fixbereichs analog zu den Fixpunkten erkennt. Die Nutzung der Zeitdatenreihenklassifikation bietet damit möglicherweise eine höhere Qualität und Stabilität bei der Synchronisierung als die Fixpunkte. Dieses Themenfeld ist daher als eines der vielversprechendsten Felder zu betrachten.

Plausibilitäts-integrierte Synchronisierung

Die aktuellen Synchronisierungsmethoden befassen sich mit den Signalverläufen, den Zeitangaben und weiteren direkten Eigenschaften. Abseits dieser Methoden existiert allerdings Wissen bezüglich Plausibilität, beispielsweise die Domänenregeln. Dies führt dazu, dass die aktuell existierenden Methoden Ergebnisse liefern, die nicht plausibel sind. Die Integration solcher Regeln in die eigentlichen Synchronisierungsmethoden ist ein innovatives Forschungsfeld, bei dem die bestehenden Methoden adaptiert und verändert werden. Beispielsweise könnte bei den Dynamic Time Warping Methoden eine Beschränkung, analog zu dem Sakoe-Chiba-Band, auf Basis der Domänenregeln nach jedem Sprung nachberechnet und angepasst werden. Auch bei der Methode SSI könnten die Domänenregeln mathematisch als Matrixbestandteile modelliert und eingegeben werden. Derartige Anpassungen sind methodenspezifisch und bedürfen tiefgreifender Veränderungen. Daraus

ergeben sich innovative Veränderungen und hohe Potenziale in Bezug auf Qualität der Ergebnisse. Der Aufwand ist allerdings ebenfalls als hoch einzuschätzen.

Neue Methoden der Synchronisierung

Während im Themenfeld zuvor eine umfassende Anpassung von bestehenden Methoden zur Berücksichtigung von Wissen im Fokus steht, ist auch Bedarf an neuen Methoden vorhanden. Die dargestellten Methodengruppen setzen dabei auf Verläufe von Daten, Zeitangaben oder Korrelationsanalysen. Gleichzeitig existieren weitergehende Eigenschaften der Zeitdatenreihen und neuartige Vorhersagemodelle. Die Nutzung dieser Metriken und Modelle für Synchronisierungen ist ein umfassendes Themenfeld und beschreibt ein aktives Forschungsfeld. Die Anwendung dieser Methoden ist derart umfangreich und von vielen Domänen geprägt, sodass der Bedarf hoch, aber auch generisch ist. Neue Methoden, die von Anfang an Produktionsdaten in den Vordergrund stellen, können hier eine höhere Praxisnähe erreichen.

ML-basierte Synchronisierung mit historischen Daten

Mit Aufkommen der Industrie 4.0 und weiteren Trends steigt die Verfügbarkeit von Daten in Unternehmen zunehmend. Dies führt auch zu einer großen Menge historischer Daten. Eine Möglichkeit ist nun, diese historischen Daten zu nutzen, um einerseits unternehmensspezifische Muster in den Daten zu finden und zugleich die Grundwahrheit einer synchronisierten Produktion zu analysieren. Aus diesen historischen Datensätzen können Modelle erstellt werden, welche die Synchronizität und Vorhersagen beschreiben. Derartige Modelle können dann bei einer Offline Synchronisierung zum Einsatz kommen und als Synchronisierungsmethode dienen, sofern eine Vergleichbarkeit grundsätzlich vorliegt. Das Problem der Klassifizierung der Asynchronizität historischer Daten bleibt dabei zwar vorhanden, allerdings könnte mit Methoden des maschinellen Lernens diese Ungenauigkeit mit in das Modell eingezogen werden. Ein solcher Ansatz wäre hochgradig innovativ, bedarf allerdings umfangreicher historischer Daten und Rechenleistung, was die Anwendung erschwert. Dennoch kann das bei vorhandenen Ressourcen eine vielversprechende Methodik darstellen, zugeschnitten auf die eigene Produktion. Mit Blick auf die Skalierung und Aufbau einer Batterieproduktion ist die Anwendbarkeit hier als hoch zu bewerten. Dieses Thema stellt damit eines der vielversprechendsten Themenfelder für weitere Forschung dar.

Langzeitmodellierung der Asynchronizität

Die Bewertung von Synchronisierungsmethoden und Erstellung von Testfällen hängt auch von der Grundwahrheit ab. Dazu kommt üblicherweise eine definierte Grundwahrheit zum Einsatz, welche dann künstlich mit Effekten der Asynchronizität beeinträchtigt wird. Das Langzeitverhalten verschiedener Systeme wird dabei auf Basis von Literatur, Erfahrung und kurzen Versuchsreihen nachgestellt. Für eine bessere Modellierung der Asynchronizität basierend auf den Eigenschaften des technischen Systems ist eine weitergehende Langzeituntersuchung und damit verbundene Modellierung der Effekte relevant.

Abschließend lässt sich feststellen, dass das Themenfeld der Offline Synchronisierung von Produktionsdaten viele weitere und offene Fragestellung beinhaltet. Die Bearbeitung dieser Themenfelder bedarf sowohl neuer Ansätze, Einsichten in Produktionen und Erkenntnisgewinne im Bereich der Batterieproduktion.

6.3 Zusammenfassung

Summary

Die Asynchronizität und Abweichungen von Zeitinformationen in technischen Systemen wirkt sich auf die Aufnahme von Daten in komplexen Produktionsnetzwerken aus. Können hierbei nicht im Betrieb (Online) Synchronisierungen der Systeme vorgenommen werden, so stellt die Offline Synchronisierung die letzte Chance dar, die Synchronizität der Produktionsdaten und die damit verbundene Datenqualität zu erhöhen. Während einige Methoden hierzu existieren, hat diese Arbeit einige neue Ansätze und Innovationen aufgezeigt. Die Nutzung von Wissen und Informationen zusätzlich zu den Daten selbst erlaubt eine Bewertung und Auswahl von Methoden entlang der konkreten Gegebenheiten und gestaffelte Synchronisierung. Auch lässt sich als weitere Innovation eine Suche nach gleichzeitigen Punkten in den Daten (Fixpunkte) vollziehen und bei Identifikation solcher Punkte eine Zerlegung der Zeitdatenreihen in mehrere Sequenzen vornehmen. Im Ergebnis hat die neue Methodik gezeigt, dass die notwendige Automatisierung mit Blick auf großskalige Batterieproduktionen erreicht werden kann und zugleich ein hohes Maß an Stabilität möglich ist, während einzelne Methoden höhere Schwankungen in Qualität und Anwendbarkeit mit sich bringen.

Die Hauptforschungsfrage und die damit verbundenen Unterforschungsfragen konnten beantwortet werden und bestätigen die Annahmen und Anwendbarkeit der Grundideen dieser Ausarbeitung im Rahmen der genutzten Szenarien und Betrachtungen. Die

Ergebnisse decken zudem Herausforderungen auf, wie die Modellierung und notwendige Genauigkeit der bereitgestellten Informationen und Wissenselementen. Daraus ergibt sich weiterer Forschungs- und Entwicklungsbedarf um eine vollständig tragfähige Lösung zu schaffen, welche konstant zu einer Verbesserung der Datenqualität führt. Das Hauptproblem der asynchronen Daten in einer Produktionsumgebung kann damit nicht zeitnah gelöst werden durch Offline Methoden, jedoch kann die Qualität der Daten in Zukunft weiter erhöht werden und damit möglicherweise auch historischen Datensätzen eine bessere Qualität und Nutzbarkeit gegeben werden. Damit leistet diese Arbeit einen wichtigen Beitrag und einen Schritt in die Richtung der Erhöhung der Datenqualität und Hebung von Potenzialen in historischen Datensätzen.

Literaturverzeichnis

Bibliography

- [ACKO89] Ackoff, R. L.: From data to wisdom. In: *Journal of applied systems analysis* 16 (1989), Nr. 1, S. 3–9
- [AD10] Andreas Dreher, D. M.: Precision Clock Synchronization - The Standard IEEE 1588 Rev. 1.2 / Hirschmann Automation and Control GmbH. 2010
- [ADEN23] Adenusi, H.; Chass, G. A.; Passerini, S.; Tian, K. V.; Chen, G.: Lithium batteries and the solid electrolyte interphase (SEI) – progress and outlook. In: *Advanced Energy Materials* 13 (2023), Nr. 10, S. 2203307
- [AG22] Arzac-Garmendia, I.; Vallerio, M.; Perez-Galvan, C.; Navarro-Brull, F. J.: Industrial Data Science for Batch Manufacturing Processes. In: *arXiv preprint arXiv:2209.09660* (2022)
- [AGUI09] Aguilera, M. K.; Walfish, M.: No Time for Asynchrony. In: *HotOS*, 2009
- [ALSI18] Alsius, A.; Paré, M.; Munhall, K. G.: Forty years after hearing lips and seeing voices: the McGurk effect revisited. In: *Multisensory Research* 31 (2018), Nr. 1-2, S. 111–144
- [ALWA20] Alwan, H. B.; Ku-Mahamud, K. R.: Big data: Definition, characteristics, life cycle, applications, and challenges. In: *IOP Conference Series: Materials Science and Engineering* Bd. 769 IOP Publishing, 2020, S. 012007
- [ANAC22] ANACONDA INC.: 2022 State of Data Science. Report. / Anaconda Inc. 2022
- [ANAC23] ANACONDA INC.: 2023 State of Data Science. Report. / Anaconda Inc. 2023
- [ANGE21] Angel, N. A.; Ravindran, D.; Vincent, P. D. R.; Srinivasan, K.; Hu, Y.-C.: Recent advances in evolving computing paradigms: Cloud, edge, and fog technologies. In: *Sensors* 22 (2021), Nr. 1, S. 196
- [ARD16] Ab Rashid Dar, R. D.: Survey On Scalability In Cloud Environment. In: *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 5 (2016), Nr. 7, S. 2124–2128
- [ASHJ21] Ashjaei, M.; Bello, L. L.; Daneshtalab, M.; Patti, G.; Saponara, S.; Mubeen, S.: Time-Sensitive Networking in automotive embedded systems: State of the art and research opportunities. In: *Journal of systems architecture* 117 (2021), S. 102137

- [ATLA18] Atlam, H. F.; Walters, R. J.; Wills, G. B.: Fog computing and the internet of things: A review. In: *big data and cognitive computing* 2 (2018), Nr. 2, S. 10
- [AYER22] Ayerbe, E.; Berecibar, M.; Clark, S.; Franco, A. A.; Ruhland, J.: Digitalization of battery manufacturing: current status, challenges, and opportunities. In: *Advanced Energy Materials* 12 (2022), Nr. 17, S. 2102696
- [BALA23] Balakrishnan, K.; Dhanalakshmi, R.; Gopalakrishnan, R. u. a.: Clock synchronization in industrial Internet of Things and potential works in precision time protocol: Review, challenges and future directions. In: *International Journal of Cognitive Computing in Engineering* 4 (2023), S. 205–219
- [BAND24] Bandi, T. N.: A comprehensive overview of atomic clocks and their applications. In: *Demo Journal* 1 (2024), Nr. 1, S. 40–50
- [BANK98] Banks, J.: *Handbook of simulation: principles, methodology, advances, applications, and practice*. John Wiley & Sons, 1998
- [BARB23] Barbulescu, C.: Microservices as a Key Enabler of a Cloud Native Architecture. In: *Education, Research and Business Technologies: Proceedings of 21st International Conference on Informatics in Economy (IE 2022)* Springer, 2023, S. 3–9
- [BAUC25] Baucks, M.; Mau, M.; Ruppelt, P.; Puchta, A.; Fleischer, J.: Lean Data – Anwendungsspezifische Reduktion großer Datenmengen im Produktionsumfeld. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 120 (2025), Nr. s1, 152–158. – DOI doi:10.1515/zwf–2024–0148
- [BEHR23] Behrendt, S.; Martin, M.; Puchta, A.; Ströbel, R.; Fisel, J.; May, M. C.; Gönzheimer, P.; Fleischer, J.; Lanza, G.: Software-Defined Manufacturing for the Entire Life Cycle at Different Levels of Production. In: Kiefl, N. (Hrsg.); Wulle, F. (Hrsg.); Ackermann, C. (Hrsg.); Holder, D. (Hrsg.): *Advances in Automotive Production Technology – Towards Software-Defined Manufacturing and Resilient Supply Chains*. Cham : Springer International Publishing, 2023. – ISBN 978–3–031–27933–1, S. 25–34
- [BELL00] Bellini, P.; Mattolini, R.; Nesi, P.: Temporal logics for real-time system specification. In: *ACM Computing Surveys (CSUR)* 32 (2000), Nr. 1, S. 12–42
- [BELL19] Bello, L. L.; Steiner, W.: A perspective on IEEE time-sensitive networking for industrial communication and automation systems. In: *Proceedings of the IEEE* 107 (2019), Nr. 6, S. 1094–1120

- [BG21] Blázquez-García, A.; Conde, A.; Mori, U.; Lozano, J. A.: A review on outlier/-anomaly detection in time series data. In: *ACM Computing Surveys (CSUR)* 54 (2021), Nr. 3, S. 1–33
- [BIDP19] BldPe, M.: SI Brochure: The International System of Units (SI). In: ed. <https://www.bipm.org> (2019)
- [BIER81] Biernacki, P.; Waldorf, D.: Snowball sampling: Problems and techniques of chain referral sampling. In: *Sociological methods & research* 10 (1981), Nr. 2, S. 141–163
- [BIGL22] Biglarbegan, M.: Formalization and Knowledge Representation in Advanced Engineering Informatics. In: *Journal of Computing and Natural Science* 2 (2022), Nr. 1, S. 008–014
- [BILL13] Billings, S. A.: *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013
- [BISW22] Biswas, S.; Wardat, M.; Rajan, H.: The art and practice of data science pipelines: A comprehensive study of data science pipelines in theory, in-the-small, and in-the-large. In: *Proceedings of the 44th International Conference on Software Engineering*, 2022, S. 2091–2103
- [BLUE22] BLUE SOLUTIONS: *SOLID-STATE BATTERIES FOR EVs – Strategic Roadmap Gen 4 SSB For Electric Vehicles*. Präsentation Battery Show 2022, 2022
- [BOCK23] Bockey, G.; Heimes, H.: *Europa Batterieproduktion*. Version: 2023. <https://battery-news.de/batterieproduktion/>. online. – Abgerufen: 07.05.2024
- [BONO12] Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S.: Fog computing and its role in the internet of things. In: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, S. 13–16
- [BOX15] Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; Ljung, G. M.: *Time series analysis: forecasting and control*. John Wiley & Sons, 2015
- [BREC17] Brecher, C.; Klocke, F.; Schmitt, R.; Schuh, G.: *Internet of Production für agile Unternehmen: AWK Aachener Werkzeugmaschinen-Kolloquium, 18. bis 19. Mai 2017*. Apprimus Verlag, 2017

- [BRET14] Brettel, M.; Friederichsen, N.; Keller, M.; Rosenberg, M.: How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 Perspective. In: *International journal of information and communication engineering* 8 (2014), Nr. 1, S. 37–44
- [BRIN18] Brincat, S. L.; Siegel, M.; Nicolai, C. von; Miller, E. K.: Gradual progression from sensory to task-related processing in cerebral cortex. In: *Proceedings of the National Academy of Sciences* 115 (2018), Nr. 30, S. E7202–E7211
- [BRIN24] Bringmann, K.; Fischer, N.; Hoog, I. van d.; Kipouridis, E.; Kociumaka, T.; Rotenberg, E.: Dynamic Dynamic Time Warping. In: *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* SIAM, 2024, S. 208–242
- [BROC91] Brockwell, P. J.; Davis, R. A.: *Time series: theory and methods*. Springer science & business media, 1991
- [BUND16a] BUNDESMINISTERIUM DER JUSTIZ BMI: Gesetz über die Einheiten im Messwesen und die Zeitbestimmung (Einheiten- und Zeitgesetz - EinhZeitG). 2016
- [BUND16b] BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG BMBF: *Industrie 4.0*. Version: 2016. <https://www.bmbf.de/bmbf/de/forschung/digitale-wirtschaft-und-gesellschaft/industrie-4-0/industrie-4-0>. online. – Abgerufen 01.05.2024
- [BURE22] BUREAU INTERNATIONAL DES POID ET MESURES: Resolution 5 of the 27th CGPM (2022). 2022
- [BÜRK22] Bürklin, B.; Latz, T.; Schenk, L.; Degen, F.; Diehl, M.; Krätzig, O.; Paulsen, T.; Wessel, S.; Kampker, A.; Lackner, N. u. a.: Umfeldbericht zum europäischen Innovationssystem Batterie 2022. (2022)
- [BUSH03] Bushara, K. O.; Hanakawa, T.; Immisch, I.; Toma, K.; Kansaku, K.; Hallett, M.: Neural correlates of cross-modal binding. In: *Nature neuroscience* 6 (2003), Nr. 2, S. 190–195
- [B.V.14] B.V., A.; contributors: *vis.js - A dynamic, browser-based visualization library*. Version: 2014. <https://visjs.org/>. – Abgerufen: 2024-09-23
- [BYRN18] Byrne, G.; Dimitrov, D.; Monostori, L.; Teti, R.; Houten, F. van; Wertheim, R.: Biologicalisation: Biological transformation in manufacturing. In: *CIRP Journal of Manufacturing Science and Technology* 21 (2018), S. 1–32

- [CAMP07] Camparo, J.: The rubidium atomic clock and basic research. In: *Physics today* 60 (2007), Nr. 11, S. 33–39
- [CAO19] Cao, Y.; Summerfield, C.; Park, H.; Giordano, B. L.; Kayser, C.: Causal inference in the multisensory brain. In: *Neuron* 102 (2019), Nr. 5, S. 1076–1087
- [CARB04] Carbone, A.; Castelli, G.; Stanley, H. E.: Time-dependent Hurst exponent in financial time series. In: *Physica A: Statistical Mechanics and its Applications* 344 (2004), Nr. 1-2, S. 267–271
- [CARB19] Carbery, C. M.; Woods, R.; Marshall, A. H.: A new data analytics framework emphasising preprocessing of data to generate insights into complex manufacturing systems. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 233 (2019), Nr. 19-20, S. 6713–6726
- [CHEN08] Chen, M.; Ebert, D.; Hagen, H.; Laramée, R. S.; Van Liere, R.; Ma, K.-L.; Ribarsky, W.; Scheuermann, G.; Silver, D.: Data, information, and knowledge in visualization. In: *IEEE computer graphics and applications* 29 (2008), Nr. 1, S. 12–19
- [CHER23] Chernov, G.: How We Identify Cause-Effect Relationships Given Evidence: An Exploratory Study. In: *Available at SSRN* 4657687 (2023)
- [CHO22] Cho, E.; Chang, T.-W.; Hwang, G.: Data preprocessing combination to improve the performance of quality classification in the manufacturing process. In: *Electronics* 11 (2022), Nr. 3, S. 477
- [CIOF20] Cioffi, R.; Travaglioni, M.; Piscitelli, G.; Petrillo, A.; De Felice, F.: Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. In: *Sustainability* 12 (2020), Nr. 2, S. 492
- [CLAR22] Clark, S.; Andersen, C. W.; Flores, E.; Bleken, F. L.; Friis, J.: (Digital presentation) a battery interface ontology for data interoperability and semantic knowledge representation. In: *Electrochemical Society Meeting Abstracts* 242 The Electrochemical Society, Inc., 2022, S. 2582–2582
- [CONR24] Conrad, F.; Mälzer, M.; Lange, F.; Wiemer, H.; Ihlenfeldt, S.: AutoML Applied to Time Series Analysis Tasks in Production Engineering. In: *Procedia Computer Science* 232 (2024), S. 849–860

- [CROW16] CrowdFlower: 2016 Data Science Report / Figure Eight (Formerly CrowdFlower). 2016
- [DAVE98] Davenport, T. H.; Prusak, L.: *Working knowledge: How organizations manage what they know*. Harvard Business Press, 1998
- [DD19] De Donno, M.; Tange, K.; Dragoni, N.: Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog. In: *Ieee Access* 7 (2019), S. 150936–150948
- [DEGE22] Degen, F.; Krätzig, O.: Future in battery production: An extensive benchmarking of novel production technologies as guidance for decision making in engineering. In: *IEEE Transactions on Engineering Management* (2022)
- [DEGE23] Degen, F.; Diehl, M.; Hettesheimer, T.; Kampker, A.; Krätzig, O.; Lackner, N.; Link, S.; Latz, T.; Paulsen, T.; Schaefer, M. u. a.: Umfeldbericht zu Natrium-Ionen-Batterien 2023: Status Quo und Perspektiven entlang einer zukünftigen Wertschöpfungskette. (2023)
- [DEUT22] DEUTSCHES INSTITUT FÜR NORMUNG DIN: DIN 8580:2022: Fertigungsverfahren / Deutsches Institut für Normung DIN. 2022
- [DONG23] Dong, A.; Starr, A.; Zhao, Y.: Neural network-based parametric system identification: a review. In: *International Journal of Systems Science* 54 (2023), Nr. 13, S. 2676–2688
- [DOWN13] Downie, N.; contributors: *Chart.js - Simple, flexible, interactive charts for designers & developers*. Version: 2013. <https://www.chartjs.org/>. – Abgerufen: 2024-09-23
- [DS19] Del Ser, J.; Osaba, E.; Molina, D.; Yang, X.-S.; Salcedo-Sanz, S.; Camacho, D.; Das, S.; Suganthan, P. N.; Coello, C. A. C.; Herrera, F.: Bio-inspired computation: Where we stand and what's next. In: *Swarm and Evolutionary Computation* 48 (2019), S. 220–250
- [EDMO67] Edmonds, J. u. a.: Optimum branchings. In: *Journal of Research of the national Bureau of Standards B* 71 (1967), Nr. 4, S. 233–240
- [EHRL22] Ehrlinger, L.; WöB, W.: A survey of data quality measurement and monitoring tools. In: *Frontiers in big data* 5 (2022), S. 850611

- [EINS05] Einstein, A.: Zur elektrodynamik bewegter körper. In: *Annalen der physik* 4 (1905)
- [ELLI73] Ellingson, C.; Kulpinski, R.: Dissemination of system time. In: *IEEE Transactions on Communications* 21 (1973), Nr. 5, S. 605–624
- [ERIK00] Eriksson, L.; Johansson, E.; Kettaneh-Wold, N.; Wikström, C.; Wold, S.: Design of experiments. In: *Principles and Applications, Learn ways AB, Stockholm* (2000)
- [ES18] Emmert-Streib, F.; Dehmer, M.: Defining data science by a data-driven quantification of the community. In: *Machine Learning and Knowledge Extraction* 1 (2018), Nr. 1, S. 235–251
- [ESLI12] Esling, P.; Agon, C.: Time-series data mining. In: *ACM Computing Surveys (CSUR)* 45 (2012), Nr. 1, S. 1–34
- [ETM 15] ETM Professional Control GmbH: Whitepaper – Discussion of WinCC OA Time Behavior / Siemens AG. 2015
- [FAOU22] Faouzi, J.: Time series classification: A review of algorithms and implementations. In: *Machine Learning (Emerging Trends and Applications)* (2022)
- [FIEL00] Fielding, R. T.: *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000
- [FITZ23] Fitzner, A.; Abramowski, J.-P.; Schmetz, A.; Krauß, J.; Borzutzki, K.; Eckstein, M.; Pouls, K.; Baum, C.; Schmitt, R.; Kampker, A.: Cause-Effect Relationships in Battery Cell Production-Data based validation of expert knowledge in electrode production. In: *Procedia CIRP* 120 (2023), S. 469–474
- [FRIE01] Friedman, E. G.: Clock distribution networks in synchronous digital integrated circuits. In: *Proceedings of the IEEE* 89 (2001), Nr. 5, S. 665–692
- [FRYE20] Frye, M.; Schmitt, R. H.: Structured data preparation pipeline for machine learning-applications in production. In: *17th IMEKO TC 10* (2020), S. 241–246
- [FRYE21] Frye, M.; Mohren, J.; Schmitt, R. H.: Benchmarking of data preprocessing methods for machine learning-applications in production. In: *Procedia CIRP* 104 (2021), S. 50–55

- [FRYE22] Frye, M.: *Recommending Data Preprocessing Pipelines for Machine Learning Applications in Production*. Diss. RWTH Aachen University, 09 2022
- [GA22] Govinda Atmaja, G. P. S.; Sambodo, N. P.; Muflikhun, M. A.: A Mini Review on The Recent Progress on The Method of Recycling Lithium-Ion Battery: Pros And Cons In Environmental and Economical Aspect. In: *Journal of Engineering Science & Technology Review* 15 (2022), Nr. 1
- [GEET17] Geetha, D.; Tabassum, N.: A survey on clock synchronization protocols in wireless sensor networks. In: *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)* IEEE, 2017, S. 504–509
- [GELP10] Gelper, S.; Fried, R.; Croux, C.: Robust forecasting with exponential and Holt–Winters smoothing. In: *Journal of forecasting* 29 (2010), Nr. 3, S. 285–300
- [GHAZ23] Ghazdali, A.; Hadri, A.; Laghrib, A.; Nachaoui, M.: A blind poisson–gaussian noise separation using learning copula densities. In: *Circuits, systems, and signal processing* 42 (2023), Nr. 11, S. 6564–6590
- [GIAN18] Giannoula, A.; Gutierrez-Sacristán, A.; Bravo, Á.; Sanz, F.; Furlong, L. I.: Identifying temporal patterns in patient disease trajectories using dynamic time warping: A population-based study. In: *Scientific reports* 8 (2018), Nr. 1, S. 4216
- [GIOR09] Giorgino, T.: Computing and visualizing dynamic time warping alignments in R: the dtw package. In: *Journal of statistical Software* 31 (2009), S. 1–24
- [GOLO23] Golovianko, M.; Terziyan, V.; Branytskyi, V.; Malyk, D.: Industry 4.0 vs. Industry 5.0: co-existence, Transition, or a Hybrid. In: *Procedia Computer Science* 217 (2023), S. 102–113
- [GONÇ22] Gonçalves, R.; Lanceros-Méndez, S.; Costa, C.: Electrode fabrication process and its influence in lithium-ion battery performance: State of the art and future trends. In: *Electrochemistry Communications* 135 (2022), S. 107210
- [GORE20] Gore, R. N.; Lisova, E.; Åkerberg, J.; Björkman, M.: Clock synchronization in future industrial networks: Applications, challenges, and directions. In: *2020 AEIT International Annual Conference (AEIT)* IEEE, 2020, S. 1–6
- [GREB69] Grebene, A.; Camenzind, H.: Phase locking as a new approach for tuned integrated circuits. In: *1969 IEEE International Solid-State Circuits Conference. Digest of Technical Papers* Bd. 12 IEEE, 1969, S. 100–101

- [GUO16] Guo, H.; Crossley, P.: Design of a time synchronization system based on GPS and IEEE 1588 for transmission substations. In: *IEEE Transactions on power delivery* 32 (2016), Nr. 4, S. 2091–2100
- [HAGH22] Haghi, S.; Summer, A.; Bauerschmidt, P.; Daub, R.: Tailored Digitalization in Electrode Manufacturing: The Backbone of Smart Lithium-Ion Battery Cell Production. In: *Energy Technology* 10 (2022), Nr. 10, S. 2200657
- [HAGH23] Haghi, S.; Hidalgo, M. F. V.; Niri, M. F.; Daub, R.; Marco, J.: Machine learning in lithium-ion battery cell production: a comprehensive mapping study. In: *Batteries & Supercaps* 6 (2023), Nr. 7, S. e202300046
- [HAMD20] Hamdan, S.; Ayyash, M.; Almajali, S.: Edge-computing architectures for internet of things applications: A survey. In: *Sensors* 20 (2020), Nr. 22, S. 6441
- [HAN22] Han, Y.; Li, Q.; Wang, Y.; Gong, K.; Liu, P.; Sun, Y.: Study on crystal oscillator frequency drift model and performance of low orbit satellite. In: *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* Bd. 10, 2022, S. 1622–1626
- [HANK15] Hankel, M.: Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0) / Verband der Elektro- und Digitalindustrie ZVEI e.V. 2015
- [HANN12] Hannan, E. J.; Deistler, M.: *The statistical theory of linear systems*. SIAM, 2012
- [HAXB91] Haxby, J. V.; Grady, C. L.; Ungerleider, L. G.; Horwitz, B.: Mapping the functional neuroanatomy of the intact human brain with brain work imaging. In: *Neuropsychologia* 29 (1991), Nr. 6, S. 539–555
- [HE21] He, X.; Zhao, K.; Chu, X.: AutoML: A survey of the state-of-the-art. In: *Knowledge-based systems* 212 (2021), S. 106622
- [HEID17] Heidel, R.; Döbrich, U.; Hankel, M.; Hoffmeister, M.: *Basiswissen RAMI 4.0: Referenzarchitekturmodell und Industrie 4.0-Komponente Industrie 4.0*. Beuth Verlag, 2017

- [HEID24] Heider, I.; Baumgärtner, J.; Bott, A.; Ströbel, R.; Puchta, A.; Fleischer, J.: Towards a Testing Framework for Machine Learning Model Deployment in Manufacturing Systems. In: *Procedia CIRP* 127 (2024), 122-128. – DOI <https://doi.org/10.1016/j.procir.2024.07.022>. – ISSN 2212–8271. – 10th CIRP Conference on Assembly Technology and Systems (CIRP CATS 2024)
- [HEIM18] Heimes, H. H.; Kampker, A.; Lienemann, C.; Locke, M.; Offermanns, C.; Michaelis, S.; Rahimzei, E.: *Lithium-ion battery cell production process*. PEM der RWTH Aachen University, 2018
- [HEIM22] Heimes, H.; Kampker, A.; Dorn, B.; Offermanns, C.; Bockey, G.; Wennemar, S.; Clever, H.; Sasse, K.; Vienenkötter, J.; Locke, M.; Soldan, N.; Kisseler, N.; Michaelis, S.; Schütrumpf, J.: BATTERY ATLAS 2022 – SHAPING THE EUROPEAN LITHIUM-ION BATTERY INDUSTRY / PEM of RWTH Aachen / VDMA Battery Production. 2022
- [HEVN04] Hevner, A. R.; March, S. T.; Park, J.; Ram, S.: Design science in information systems research. In: *MIS quarterly* (2004), S. 75–105
- [HORO89] Horowitz, P.; Hill, W.; Robinson, I.: *The art of electronics*. Bd. 2. Cambridge university press Cambridge, 1989
- [HSIE96] Hsieh, G.-C.; Hung, J. C.: Phase-locked loop techniques. A survey. In: *IEEE Transactions on industrial electronics* 43 (1996), Nr. 6, S. 609–615
- [HUAN85] Huang, H.-C.; Jansen, B.: EEG waveform analysis by means of dynamic time-warping. In: *International journal of bio-medical computing* 17 (1985), Nr. 2, S. 135–144
- [IEC 17] IEC TECHNICAL COMMITTEE 65: INDUSTRIAL-PROCESS MEASUREMENT, CONTROL AND AUTOMATION: IEC PAS 63088:2017: Smart manufacturing - Reference architecture model industry 4.0 (RAMI4.0) / International Electrotechnical Commission IEC. 2017
- [IEC 20] IEC TECHNICAL COMMITTEE 65: INDUSTRIAL-PROCESS MEASUREMENT, CONTROL AND AUTOMATION: IEC 62890:2020: Industrial-process measurement, control and automation - Life-cycle-management for systems and components / International Electrotechnical Commission IEC. 2020
- [IEEE20] IEEE: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. Version: 2020. <http://dx.doi.org/10.1109/IEEESTD.2020.9120376>. 2020. – 1–499 S.

- [IF19] Ismail Fawaz, H.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.-A.: Deep learning for time series classification: a review. In: *Data mining and knowledge discovery* 33 (2019), Nr. 4, S. 917–963
- [INTE10] INTERNATIONAL SOCIETY OF AUTOMATION: ANSI/ISA-95: Enterprise-Control System Integration / American National Standards Institute ANSI. 2010
- [INTE22] INTEL CORPORATION: *Intel Research Fuels Moore's Law and Paves the Way to a Trillion Transistors by 2030*. Version: 2022. <https://www.intel.com/content/www/us/en/newsroom/news/moores-law-paves-way-trillion-transistors-2030.html>. online. – Abgerufen: 02.05.2024
- [ISLA16] Islam, M. N.; Pramanik, A.: Comparison of design of experiments via traditional and Taguchi method. In: *Journal of Advanced Manufacturing Systems* 15 (2016), Nr. 03, S. 151–160
- [ITAK75] Itakura, F.: Minimum prediction residual principle applied to speech recognition. In: *IEEE Transactions on acoustics, speech, and signal processing* 23 (1975), Nr. 1, S. 67–72
- [ITSK16] Itskovich, E.: Basics of MES construction concept for production industries. In: *Automation and Remote Control* 77 (2016), S. 2035–2043
- [IXPD24] IXPDB: *IXPs*. Version: 2024. <https://ixpdb.euro-ix.net/en/explore/ixps/>. online. – Abgerufen: 06.05.2024
- [JEON11] Jeong, Y.-S.; Jeong, M. K.; Omitaomu, O. A.: Weighted dynamic time warping for time series classification. In: *Pattern recognition* 44 (2011), Nr. 9, S. 2231–2240
- [JIAN20] Jiang, W.; Wu, J.; Sun, G.; Ouyang, Y.; Li, J.; Zhou, S.: A survey of time series data visualization methods. In: *Journal of Quantum Computing* 2 (2020), Nr. 2, S. 105
- [JOÃO13] João, D. F.: *Bio-inspired self-organisation in evolvable production systems*. Diss. KTH Royal Institute of Technology, 2013
- [JUNI20] Junior, W. R. L.; Andrade, L. P. C.; Oliveira, S. C. P.; Martins, S. A. M.: SysIdentPy: A Python package for System Identification using NARMAX models. In: *Journal of Open Source Software* 5 (2020), Nr. 54, 2384. – DOI 10.21105/joss.02384

- [KAGE11] Kagermann, H.; Lukas, W.-D.; Wahlster, W.: Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution. In: *VDI nachrichten* 13 (2011), Nr. 1, S. 2–3
- [KAGG20] KAGGLE: State of Data Science and Machine Learning 2020. Report. / Kaggle. 2020
- [KAHN02] Kahn, B. K.; Strong, D. M.; Wang, R. Y.: Information quality benchmarks: product and service performance. In: *Communications of the ACM* 45 (2002), Nr. 4, S. 184–192
- [KAHV22] Kahveci, S.; Alkan, B.; Mus'ab H, A.; Ahmad, B.; Harrison, R.: An end-to-end big data analytics platform for IoT-enabled smart factories: A case study of battery module assembly system for electric vehicles. In: *Journal of Manufacturing Systems* 63 (2022), S. 214–223
- [KAMP13] Kampker, A.; Hohenthanner, C.-R.; Deutskens, C.; Heimes, H. H.; Sesterheim, C.: Fertigungsverfahren von Lithium-Ionen-Zellen und-Batterien. In: *Handbuch Lithium-Ionen-Batterien* (2013), S. 237–247
- [KAMP15] Kampker, A.; Heimes, H.; Deutskens, C.; Ordnung, M.; Maiser, E.; Michaelis, S.: Produktionsprozess einer Lithium-Ionen-Batteriezelle. In: *RWTH Aachen and VDMA Batterieproduktion, 2nd Edition, Aachen, Frankfurt* (2015), S. 11
- [KAMP23] Kampker, A.; Clever, H.: *Potentials and Implementation Strategies For Flexible Battery Cell Production*. Gottfried Wilhelm Leibniz Universität Hannover, 2023
- [KATE16] Kate, R. J.: Using dynamic time warping distances as features for improved time series classification. In: *Data mining and knowledge discovery* 30 (2016), S. 283–312
- [KELL02] Kelley, J.: *Knowledge nirvana: Achieving the competitive advantage through enterprise content management and optimizing team collaboration*. Xulon Press, 2002
- [KEOG99] Keogh, E. J.; Pazzani, M. J.: Scaling up dynamic time warping to massive datasets. In: *Principles of Data Mining and Knowledge Discovery: Third European Conference, PKDD'99, Prague, Czech Republic, September 15-18, 1999. Proceedings 3* Springer, 1999, S. 1–11

- [KEOG01] Keogh, E. J.; Pazzani, M. J.: Derivative Dynamic Time Warping. In: *Proceedings of the 2001 SIAM International Conference on Data Mining (SDM)*, 2001, 1–11
- [KEOG05] Keogh, E.; Lin, J.: Clustering of time-series subsequences is meaningless: implications for previous and future research. In: *Knowledge and information systems* 8 (2005), S. 154–177
- [KIES21] Kies, A.; Krauß, J.; Schmetz, A.; Baum, C.; Schmitt, R. H.; Brecher, C.: Der digitale Zwilling in der Batteriezellfertigung. Vom Datenmanagement über das Traceability-System zur zielgerichteten Nutzung. (2021)
- [KILT15] Kilteni, K.; Maselli, A.; Kording, K. P.; Slater, M.: Over my fake body: body ownership illusions for studying the multisensory basis of own-body perception. In: *Frontiers in human neuroscience* 9 (2015), S. 119452
- [KIM03] Kim, W.; Choi, B.-J.; Hong, E.-K.; Kim, S.-K.; Lee, D.: A taxonomy of dirty data. In: *Data mining and knowledge discovery* 7 (2003), S. 81–99
- [KIM20] Kim, S.; Kim, N. H.; Choi, J.-H.: Prediction of remaining useful life by data augmentation technique based on dynamic time warping. In: *Mechanical Systems and Signal Processing* 136 (2020), S. 106486
- [KRAU22] Krauß, J.: *Optimizing Hyperparameters for Machine Learning Algorithms in Production*. Diss. RWTH Aachen University, 2022
- [KRAU23] Krauß, J.; Schmetz, A.; Fitzner, A.; Ackermann, T.; Pouls, K. B.; Hülsmann, T.-H.; Roth, D.; Gehring, J.; Hamacher, N. C.; Jaspers, W. u. a.: Der Digitale Zwilling in der Batteriezellfertigung. (2023)
- [LATH10] Latha, C.; Shashidhara, H.: Clock synchronization in distributed systems. In: *2010 5th International Conference on Industrial and Information Systems* IEEE, 2010, S. 475–480
- [LÉVE16] Lévesque, M.; Tipper, D.: A survey of clock synchronization over packet-switched networks. In: *IEEE Communications Surveys & Tutorials* 18 (2016), Nr. 4, S. 2926–2947

- [LI20] Li, F.; Du, J.; He, Y.; Song, H.-Y.; Madkour, M.; Rao, G.; Xiang, Y.; Luo, Y.; Chen, H. W.; Liu, S. u. a.: Time event ontology (TEO): to support semantic representation and reasoning of complex temporal relations of clinical events. In: *Journal of the American Medical Informatics Association* 27 (2020), Nr. 7, S. 1046–1056
- [LI21] Li, H.: Time works well: Dynamic time warping based on time weighting for time series data mining. In: *Information Sciences* 547 (2021), S. 592–608
- [LIEW07] Liew, A.: Understanding data, information, knowledge and their inter-relationships. In: *Journal of knowledge management practice* 8 (2007), Nr. 2, S. 1–16
- [LIGG24] Liggesmeyer, P.; Schöning, H.; Schunk, H.: *Die vierte industrielle Revolution und Industrie 5.0- Eine Kritik*. online, 2024
- [LINE18] Lines, J.; Taylor, S.; Bagnall, A.: Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12 (2018), Nr. 5, S. 1–35
- [LINK23] Link, S.; Neef, C.; Wicke, T.: Trends in automotive battery cell design: A statistical analysis of empirical data. In: *Batteries* 9 (2023), Nr. 5, S. 261
- [LINU24] LINUX KERNEL ORGANIZATION: Timers / The Linux Kernel Organization. 2024
- [LIU10] Liu, K.; Yao, X.; Jiang, L.: Recent developments in bio-inspired special wettability. In: *Chemical Society Reviews* 39 (2010), Nr. 8, S. 3240–3255
- [LJUN10] Ljung, L.: Perspectives on system identification. In: *Annual Reviews in Control* 34 (2010), Nr. 1, S. 1–12
- [LOMB11] Lombardi, M. A.: The evolution of time measurement, Part 2: quartz clocks [Recalibration]. In: *IEEE Instrumentation & Measurement Magazine* 14 (2011), Nr. 5, S. 41–48
- [LU20] Lu, Y.; Xu, X.; Wang, L.: Smart manufacturing process and system automation—a critical review of the standards and envisioned scenarios. In: *Journal of Manufacturing Systems* 56 (2020), S. 312–325
- [LUDL15] Ludlow, A. D.; Boyd, M. M.; Ye, J.; Peik, E.; Schmidt, P. O.: Optical atomic clocks. In: *Reviews of Modern Physics* 87 (2015), Nr. 2, S. 637

- [LUO11] Luo, L.; Scheuermann, B.: Non-Linear Offline Time Synchronization. In: *Operations Research Proceedings 2010: Selected Papers of the Annual International Conference of the German Operations Research Society* Springer, 2011, S. 629–634
- [MAES16] Maes, K.; Reynders, E.; Rezayat, A.; De Roeck, G.; Lombaert, G.: Offline synchronization of data acquisition systems using system identification. In: *Journal of Sound and Vibration* 381 (2016), S. 264–272
- [MAGN04] Magnani, M.; Montesi, D.: A unified approach to structured, semistructured and unstructured data. In: *University of Bologna, Department of Computer Science, Tech. Rep* 9 (2004), S. 2004
- [MAHM16] Mahmood, A.; Exel, R.; Trsek, H.; Sauter, T.: Clock synchronization over IEEE 802.11—A survey of methodologies and protocols. In: *IEEE Transactions on Industrial Informatics* 13 (2016), Nr. 2, S. 907–922
- [MAIE11] Maier, A.; Niggemann, O.; Just, R.; Jäger, M.; Vodenčarević, A.: Anomaly detection in production plants using timed automata-automated learning of models from observations. In: *International Conference on Informatics in Control, Automation and Robotics* Bd. 2 SciTePress, 2011, S. 363–369
- [MARI14] Marinos, I.; Watson, R. N.; Handley, M.: Network stack specialization for performance. In: *ACM SIGCOMM Computer Communication Review* 44 (2014), Nr. 4, S. 175–186
- [MARK22] Marko, A.: Industrie 4.0 – so digital sind Deutschlands Fabriken – Ein Bitkom-Studienbericht / Bitkom e.V. 2022
- [MART72] Martin, R. M.: Piezoelectricity. In: *Physical Review B* 5 (1972), Nr. 4, S. 1607
- [MCGU76] McGurk, H.; MacDonald, J.: Hearing lips and seeing voices. In: *Nature* 264 (1976), Nr. 5588, S. 746–748
- [MEIN22] Meiners, J.; Fröhlich, A.; Dröder, K.: Potential of a machine learning based cross-process control in lithium-ion battery production. In: *Procedia CIRP* 112 (2022), S. 525–530
- [MEIN23a] MEINBERG FUNKUHREN GMBH & CO KG: *Meinberg Driver and API Concepts*. Version: 2023. https://kb.meinbergglobal.com/kb/driver_software/meinberg_sdks/meinberg_driver_and_api_concepts. online. – Abgerufen: 2024-07-13

- [MEIN23b] MEINBERG FUNKUHREN GMBH & CO KG: *Oszillatoren für Meinberg Empfänger*. Version: 2023. <https://www.meinberg.de/german/specs/gpsopt.htm>. online. – Abgerufen: 2024-07-13
- [MERZ23] Merz, D.; Gerlitz, E.; Kaiser, J.-P.; Fleischer, J.: Innovative Anomalieerkennung im Laserschweißen: Eine Pipeline basierend auf der Analyse von Strahlungsemissionen und maschinellem Lernen. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 118 (2023), Nr. 11, 790–794. – DOI doi:10.1515/zwf-2023-1146
- [MEUD17] Meudt, T.; Pohl, M.; Metternich, J.: Die automatisierungspyramide-ein literaturüberblick. (2017)
- [MICH23] Michaelis, S.; Schütrumpf, J.; Kampker, A.; Heimes, H.; Dorn, B.; Wenne-mar, S.; Scheibe, A.; Wolf, S.; Smulka, M.; Ingendoh, B.; Schmetz, A. u. a.: Roadmap Battery Production Equipment 2030. Update 2023. (2023)
- [MICR21] Microsoft: Windows Time Service (W32Time) / Microsoft. Version: 2021. <https://learn.microsoft.com/en-us/windows-server/networking/windows-time-service/windows-time-service-top>. 2021
- [MICR24] Microsoft: *FILETIME structure*. Version: 2024. <https://learn.microsoft.com/en-us/windows/win32/api/minwinbase/ns-minwinbase-filetime>. online. – Abgerufen: 03.05.2024
- [MIKO23] MikolajSzafraniecUPDS: *GitHub: shapedtw-python*. Version: 2023. <https://github.com/MikolajSzafraniecUPDS/shapedtw-python>. online. – Abgerufen: 2024-09-29
- [MILL10] Mills, D.; Martin, J.; Burbank, J.; Kasch, W.: Network time protocol version 4: Protocol and algorithms specification. 2010
- [MOHR25] Mohring, L.; Jaspers, W.; Schmetz, A.; Roth, D.; Hamacher, N. C.; Kampker, A.: Digital Battery Cell Manufacturing Systems: Approaching a scalable and versatile IT Architecture Design. In: *Procedia CIRP* 121 (2025). – ISSN 2212–8271. – 57th CIRP International Conference on Manufacturing Systems 2024
- [MUSH11] Mushtaq, R.: Augmented dickey fuller test. (2011)

- [NAAS17] Naas, M. I.; Parvedy, P. R.; Boukhobza, J.; Lemarchand, L.: iFogStor: an IoT data placement strategy for fog infrastructure. In: *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)* IEEE, 2017, S. 97–104
- [NADE22] Nadeem, F.: Evaluating and ranking cloud IaaS, PaaS and SaaS models based on functional and non-functional key performance indicators. In: *IEEE Access* 10 (2022), S. 63245–63257
- [NETZ18] NETZFREQUENZMESSUNG.DE: *Abweichung der Netzzeit überschreitet 5 Minuten*. Version: 2018. https://www.netzfrequenzmessung.de/aktuelles.htm#2018_04. online. – Abgerufen: 02.05.2024
- [NEUB97] Neubig, B.; Briese, W.: *Das grosse Quarz-Kochbuch: Quarze, Quarzoszillatoren, Quarz- und Oberflächenwellenfilter (SAW), Messtechnik; mit 23 Tabellen*. Franzis Feldkirchen, Austria, 1997
- [NEWT87] Newton, I.: *Philosophiae naturalis principia mathematica*. Bd. 1. G. Brookman, 1687
- [NIKO21] Nikolai, W.; SCHLEGL, T.; DEUSE, J.: Feature extraction for time series classification using univariate descriptive statistics and dynamic time warping in a manufacturing environment. In: *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)* IEEE, 2021, S. 762–768
- [NIRI22] Niri, M. F.; Liu, K.; Apachitei, G.; Román-Ramírez, L. A.; Lain, M.; Widanage, D.; Marco, J.: Quantifying key factors for optimised manufacturing of Li-ion battery anode and cathode via artificial intelligence. In: *Energy and AI* 7 (2022), S. 100129
- [NIST18] NIST: *Reference on Constants, Units, and Uncertainty – Fundamental Physic Constants*. online, 2018
- [OLIV05] Oliveira, P.; Rodrigues, F.; Henriques, P.; Galhardas, H.: A taxonomy of data quality problems. In: *2nd Int. Workshop on Data and Information Quality*, 2005, S. 219–233
- [OMET22] Ometov, A.; Molua, O. L.; Komarov, M.; Nurmi, J.: A survey of security in cloud, edge, and fog computing. In: *Sensors* 22 (2022), Nr. 3, S. 927
- [OPC 18] OPC Foundation: *UA-.NETStandard*. Version: 2018. <https://github.com/OPCFoundation/UA-.NETStandard>. online. – Abgerufen: 2024-07-13

- [OREG17] Oregi, I.; Pérez, A.; Del Ser, J.; Lozano, J. A.: On-line dynamic time warping for streaming time series. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II* 10 Springer, 2017, S. 591–605

- [OTTE24] Otte, S.; Sufian, N. N. A. M.; Schabel, S.; Fleischer Dr.-Ing, J.: Identification of Relevant Parameters for Traceability in the Continuous Mixing Process in Battery Cell Production. In: *Energy Technology* 12 (2024), Nr. 7, 2400493. – DOI <https://doi.org/10.1002/ente.202400493>

- [O'15] O'Donovan, P.; Leahy, K.; Bruton, K.; O'Sullivan, D. T.: An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities. In: *Journal of big data* 2 (2015), S. 1–26

- [PAHL15] Pahl, C.; Lee, B.: Containers and clusters for edge cloud architectures—a technology review. In: *2015 3rd international conference on future internet of things and cloud IEEE*, 2015, S. 379–386

- [PARZ62] Parzen, E.: On estimation of a probability density function and mode. In: *The annals of mathematical statistics* 33 (1962), Nr. 3, S. 1065–1076

- [PEGE24] Pegel, H.; Autenrieth, M.; Schaeffler, S.; Jossen, A.; Sauer, D. U.: Design guidelines to prevent thermal propagation and maximize packing density within battery systems with tabless cylindrical lithium-ion cells. In: *Journal of Energy Storage* 86 (2024), S. 111275

- [PETE07] Peterson, L. L.; Davie, B. S.: *Computer networks: a systems approach*. Elsevier, 2007

- [PI W17] PI WORKING GROUP PG6: Sequence of Events - Guideline for PROFINET / Profinet International PI. 2017

- [PIPI02] Pipino, L. L.; Lee, Y. W.; Wang, R. Y.: Data quality assessment. In: *Communications of the ACM* 45 (2002), Nr. 4, S. 211–218

- [PLAT19] PLATTFORM INDUSTRIE 4.0: Leitbild 2030 für Industrie 4.0 – Digitale Ökosysteme global gestalten / Bundesministerium für Wirtschaft und Energie (BMWi). 2019

- [PLUM23] Plumeyer, J. F.; Menebröcker, N.; Clever, H.; Kokozinski, L.; Wessel, S.; Heimes, H. H.; Kampker, A.: A Review of Process Innovations in the Cell Finishing of Lithium-Ion Batteries in Large-Scale Production. In: *ESSN: 2701-6277* (2023), S. 580–590
- [PRED15] Preden, J. S.; Tammemäe, K.; Jantsch, A.; Leier, M.; Riid, A.; Calis, E.: The benefits of self-awareness and attention in fog and mist computing. In: *Computer* 48 (2015), Nr. 7, S. 37–45
- [PUTT20] Puttnies, H.; Danielis, P.; Sharif, A. R.; Timmermann, D.: Estimators for time synchronization—survey, analysis, and outlook. In: *IoT* 1 (2020), Nr. 2, S. 398–435
- [QIN21] Qin, Z.; Lu, Y.: Self-organizing manufacturing network: A paradigm towards smart manufacturing in mass personalization. In: *Journal of Manufacturing Systems* 60 (2021), S. 35–47
- [RADA17] Radaideh, A.; Vaidya, U.; Ajjarapu, V.: Sequential set-point control for heterogeneous thermostatically controlled loads through an extended Markov chain abstraction. In: *IEEE Transactions on Smart Grid* 10 (2017), Nr. 1, S. 116–127
- [REDM18] Redman, T. C.: If your data is bad, your machine learning tools are useless. In: *Harvard Business Review* 2 (2018)
- [ROSS96] Rossum, G. Dohrn-van: *History of the hour: Clocks and modern temporal orders*. University of Chicago Press, 1996
- [ROST16] Rostaghi, M.; Azami, H.: Dispersion entropy: A measure for time-series analysis. In: *IEEE Signal Processing Letters* 23 (2016), Nr. 5, S. 610–614
- [RUBE25] Rube, N.; Schmetz, A.; Kampker, A.: Systematically integrating Industry 4.0 requirements into factory planning phases. In: *Procedia CIRP* 122 (2025). – 58th CIRP International Conference on Manufacturing Systems 2025 (PREPRINT)
- [SAKO78] Sakoe, H.; Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. In: *IEEE transactions on acoustics, speech, and signal processing* 26 (1978), Nr. 1, S. 43–49

- [SALV04] Salvador, S.; Chan, P.: FastDTW: Toward accurate dynamic time warping in linear time and space. In: *KDD workshop on mining temporal and sequential data* Bd. 6 Seattle, Washington, 2004, S. 70–80
- [SARA05] Saraiya, P.; Lee, P.; North, C.: Visualization of graphs with associated timeseries data. In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*. IEEE, 2005, S. 225–232
- [SCHE19] Scheibe, K.; Stamm, J.: *Uhr und Strom: Ein Handbuch über elektronische Uhren*. Walter de Gruyter GmbH & Co KG, 2019
- [SCHM19] Schmetz, A.; Siegburg, R.; Zontar, D.; Brecher, C.: Middleware for the IIoT. Study of the International Center for Networked, Adaptive Production (ICNAP) / International Center for Networked, Adaptive Production ICNAP. 2019
- [SCHM20] Schmitt, R. H.; Ellerich, M.; Schlegel, P.; Ngo, Q. H.; Emonts, D.; Montavon, B.; Buschmann, D.; Lauther, R.: Datenbasiertes qualitätsmanagement im internet of production. In: *Handbuch Industrie 4.0: Recht, Technik, Gesellschaft* (2020), S. 489–516
- [SCHM22] Schmetz, A.; Lee, T. H.; Zontar, D.; Brecher, C.: The Time Synchronization Problem in data-intense Manufacturing. In: *Procedia CIRP* 107 (2022), 827–832. – DOI <https://doi.org/10.1016/j.procir.2022.05.070>. – ISSN 2212–8271. – Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022
- [SCHM23a] Schmaltz, T.; Hartmann, F.; Wicke, T.; Weymann, L.; Neef, C.; Janek, J.: A Roadmap for Solid-State Batteries. In: *Advanced Energy Materials* 13 (2023), Nr. 43, S. 2301886
- [SCHM23b] Schmetz, A.; Ackermann, T.; Fitzner, A.; Steinhoff, V.; Kampker, A.: Bio-Sync: Offline-Synchronization of time-series data using bio-inspired semantic synchronization strategies. In: *Procedia CIRP* 120 (2023), 655–660. – DOI <https://doi.org/10.1016/j.procir.2023.09.054>. – ISSN 2212–8271. – 56th CIRP International Conference on Manufacturing Systems 2023

- [SCHM23c] Schmetz, A.; Roth, D.; Kampker, A.: Time Synchronization Uncertainty Estimation Methodology for data-centric Production Scenarios. In: *Procedia CIRP* 120 (2023), 637-642. – DOI <https://doi.org/10.1016/j.procir.2023.09.051>. – ISSN 2212–8271. – 56th CIRP International Conference on Manufacturing Systems 2023
- [SCHM24a] Schmetz, A.; Kampker, A.: Evaluation of Offline Data Synchronization Approaches in data-intense Manufacturing. In: *Procedia CIRP* (2024). – 57th CIRP International Conference on Manufacturing Systems 2024
- [SCHM24b] Schmetz, A.; Kampker, A.: Inside Production Data Science: Exploring the Main Tasks of Data Scientists in Production Environments. In: *AI* (2024)
- [SCHM25] Schmetz, A.; Kampker, A.: Towards Pattern Modeling for Asynchronicity in Production Data. In: *Procedia CIRP* 122 (2025). – 58th CIRP International Conference on Manufacturing Systems 2025 (PREPRINT)
- [SENI08] Senin, P.: Dynamic time warping algorithm review. In: *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855 (2008), Nr. 1-23, S. 40
- [SEST17] Sestito, G. S.; Turcato, A. C.; Dias, A. L.; Rocha, M. S.; Silva, M. M.; Ferrari, P.; Brandao, D.: A method for anomalies detection in real-time ethernet data traffic applied to PROFINET. In: *IEEE Transactions on Industrial Informatics* 14 (2017), Nr. 5, S. 2171–2180
- [SHAR10] Sharma, R.; Mathur, M.: Achieving Vertical Scalability: A hindrance to Cloud Computing. In: (Hrsg.): *Proceedings of the 4th National Conference; INDIA-Com*, 2010
- [SHI16] Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L.: Edge computing: Vision and challenges. In: *IEEE internet of things journal* 3 (2016), Nr. 5, S. 637–646
- [SILV16] Silva, D. F.; Batista, G. E.: Speeding up all-pairwise dynamic time warping matrix calculation. In: *Proceedings of the 2016 SIAM International Conference on Data Mining SIAM*, 2016, S. 837–845
- [SING12] Singh, A. V.; Rahman, A.; Kumar, N. S.; Aditi, A.; Galluzzi, M.; Bovio, S.; Barozzi, S.; Montani, E.; Parazzoli, D.: Bio-inspired approaches to design smart fabrics. In: *Materials & Design (1980-2015)* 36 (2012), S. 829–839

- [SLYP19] slypni: *GitHub: fastdtw*. Version: 2019. <https://github.com/slaypni/fastdtw>. online. – Abgerufen: 2024-09-29
- [Soft17] Software and Systems Engineering Standards Committee of the IEEE Computer Society: IEEE Standard for System, Software, and Hardware Verification and Validation. Version: 2017. <http://dx.doi.org/10.1109/IEEESTD.2017.8055462>. 2017. – 1–260 S.
- [STEP15] Stephen, M.; Gu, C.; Yang, H.: Visibility graph based time series analysis. In: *PloS one* 10 (2015), Nr. 11, S. e0143015
- [STOL95] Stolberg, L.; Stolberg, L. (Hrsg.): *Lexikon der Taschenuhr*. Carinthia, 1995
- [STRÖ23] Ströbel, R.; Probst, Y.; Deucker, S.; Fleischer, J.: Time Series Prediction for Energy Consumption of Computer Numerical Control Axes Using Hybrid Machine Learning Models. In: *Machines* 11 (2023), Nr. 11. – DOI 10.3390/machines11111015. – ISSN 2075–1702
- [TECH94] TECHNICAL COMMITTEE ISO/IEC JTC 1: Open Systems Interconnection (OSI) / International Organization for Standardization ISO. 1994
- [TECH13] TECHNICAL COMMITTEE ISO/TC 184/SC 5: IEC 62264: Enterprise-Control System Integration / International Organization for Standardization ISO. 2013
- [TECH17] TECHNICAL COMMITTEE ISO/IEC JTC 1/SC 7, Software and systems engineering: ISO/IEC/IEEE 24765:2017: Systems and software engineering–Vocabulary / International Organization for Standardization. 2017
- [TECH19a] TECHNICAL COMMITTEE ISO/IEC JTC 1/SC 7 SOFTWARE AND SYSTEMS ENGINEERING: ISO 25012:2008: Data Quality Model / International Organization for Standardization. 2019
- [TECH19b] TECHNICAL COMMITTEE ISO/TC 154, PROCESSES, DATA ELEMENTS AND DOCUMENTS IN COMMERCE, INDUSTRY AND ADMINISTRATION: ISO 8601-1:2019: Date and time — Representations for information interchange / International Organization for Standardization. 2019
- [TECH22a] TECHNICAL COMMITTEE TC 184, AUTOMATION SYSTEMS AND INTEGRATION, SUB-COMMITTEE SC 4, INDUSTRIAL DATA: ISO 8000:2022: Data Quality / International Organization for Standardization. 2022

- [TECH22b] TECHNICAL COMMITTEE ISO/IEC JTC 1/SC 27, INFORMATION SECURITY, CYBERSECURITY AND PRIVACY PROTECTION: ISO 27001:2022: Information security, cybersecurity and privacy protection — Information security management systems — Requirements / International Organization for Standardization. 2022
- [TELE24] TeleGeography: *Submarine Cable Map*. Version: 2024. <https://www.submarinecablemap.com/>. online. – Abgerufen: 06.05.2024
- [THOM20] Thomas, L.; Le Boudec, J.-Y.: On time synchronization issues in time-sensitive networks with regulators and nonideal clocks. In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4 (2020), Nr. 2, S. 1–41
- [TSIN14] Tsinaslanidis, P.; Alexandridis, A.; Zapranis, A.; Livanis, E.: Dynamic time warping as a similarity measure: applications in finance. (2014)
- [TURE20] Turetskyy, A.; Thiede, S.; Thomitzek, M.; Von Drachenfels, N.; Pape, T.; Herrmann, C.: Toward data-driven applications in lithium-ion battery cell manufacturing. In: *Energy Technology* 8 (2020), Nr. 2, S. 1900136
- [UBUN18] UBUNTU MANPAGE: *ntp.conf - Network Time Protocol (NTP) daemon configuration file format*. Version: 2018. <https://manpages.ubuntu.com/manpages/bionic/man5/ntp.conf.5.html>. – Abgerufen: 2024-05-30
- [ULRI76] Ulrich, P.; Hill, W.: Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre. In: *Wirtschaftswissenschaftliches Studium: Zeitschrift für Ausbildung und Hochschulkontakt* 5 (1976), Nr. 7+ 8, S. 304–309
- [ULRI81] Ulrich, H.: Die betriebswirtschaftslehre als anwendungsorientierte sozialwissenschaft. In: *Die Führung des Betriebes, Stuttgart* 1981 (1981), S. 1–25
- [VERH07] Verhaegen, M.; Verdult, V.: *Filtering and system identification: a least squares approach*. Cambridge university press, 2007
- [VIG92] Vig, J. R.: Introduction to quartz frequency standards. In: *Army Research Laboratory Electronics and Power Sources Directorate, Tech. Rep. SLCETT-R-92-1* (1992)
- [VO94] Van Overschee, P.; De Moor, B.: N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. In: *Automatica* 30 (1994), Nr. 1, S. 75–93

- [VS23] Ventura Silva, G.; Thomitzek, M.; Lippke, M.; Heckmann, T.; Karaki, H.; Lischka, C.; Möhlen, F.; Mayer, D.; Hagemeister, J.; Daub, R. u. a.: Digitalization Platform for Sustainable Battery Cell Production: Coupling of Process, Production, and Product Models. In: *Energy Technology* 11 (2023), Nr. 5, S. 2200801
- [WALH23] Walhse, R.; Farrell, S.; Guth-Orlowski, S.; Rosetti, M.; Schrundner, R.; Poggiali, F.; Roedding, T.; Heemskerk, G.; Olsson, S.; Clark, J.: Report of TWG Digital Product Passport: Landscape of Digital Product Passport Standards / StandICT.eu 2026. 2023
- [WANG05] Wang, X.; Yu, Q.: Unbiasedness of the Theil–Sen estimator. In: *Nonparametric Statistics* 17 (2005), Nr. 6, S. 685–695
- [WANG06] Wang, X.; Smith, K.; Hyndman, R.: Characteristic-based clustering for time series data. In: *Data mining and knowledge Discovery* 13 (2006), S. 335–364
- [WANG12] Wang, J.; Shang, P.; Ge, W.: Multifractal cross-correlation analysis based on statistical moments. In: *Fractals* 20 (2012), Nr. 03n04, S. 271–279
- [WANG22] Wang, J.; Xu, C.; Zhang, J.; Zhong, R.: Big data analytics for intelligent manufacturing systems: A review. In: *Journal of Manufacturing Systems* 62 (2022), S. 738–752
- [WANG24] Wang, C.; Chen, L.; Tang, C.; Wang, Y.; Xian, Y.; Zhao, Y.; Xue, H.; Huan, Z.: Enhanced time-sensitive networking configuration detection using optimized BPNN with feature selection for industry 4.0. In: *Cluster Computing* (2024), S. 1–16
- [WARD22] Ward, L.; Babinec, S.; Dufek, E. J.; Howey, D. A.; Viswanathan, V.; Aykol, M.; Beck, D. A.; Blaiszik, B.; Chen, B.-R.; Crabtree, G. u. a.: Principles of the battery data genome. In: *Joule* 6 (2022), Nr. 10, S. 2253–2271
- [WEN19] Wen, Q.; Gao, J.; Song, X.; Sun, L.; Xu, H.; Zhu, S.: RobustSTL: A robust seasonal-trend decomposition algorithm for long time series. In: *Proceedings of the AAAI conference on artificial intelligence* Bd. 33, 2019, S. 5409–5416
- [WESS23] Wessel, J.; Schoo, A.; Kwade, A.; Herrmann, C.: Traceability in Battery Cell Production. In: *Energy Technology* 11 (2023), Nr. 5, S. 2200911
- [WITT23] Witt, J.: *Zitat zu Asynchronizität*. 2023. – Head of Cell Production, Siemens; persönliche Kommunikation

- [WU21] Wu, R.; Keogh, E. J.: Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. In: *IEEE transactions on knowledge and data engineering* 35 (2021), Nr. 3, S. 2421–2429
- [WUND24] Wunderlich, P.; Ehteshami-Flammer, N.; Krauß, J.; Fitzner, A.; Mohring, L.; Dahmen, C.: The Power of Digitalization in Battery Cell Manufacturing. (2024)
- [XU23] Xu, L.; Ye, P.; Liao, S.; Chen, C.; Tan, F.: How does Ambient Temperature Fluctuation Influence the Short-Term Frequency Stability of OCXO? In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* (2023)
- [YOSH09] Yoshio, M.; Brodd, R. J.; Kozawa, A. u. a.: *Lithium-ion batteries*. Bd. 1. Springer, 2009
- [ZAMA20] Zaman, G.; Mahdin, H.; Hussain, K.; Rahman, A.: Information extraction from semi and unstructured data sources: A systematic literature review. In: *ICIC Express Letters* 14 (2020), Nr. 6, S. 593–603
- [ZARE23] Zarei, R.; Huang, G.; Wu, J.: GraphTS: Graph-represented time series for subsequence anomaly detection. In: *Plos one* 18 (2023), Nr. 8, S. e0290092
- [ZHAN16] Zhang, C.; Mcadams, D. A.; Grunlan, J. C.: Nano/micro-manufacturing of bioinspired materials: a review of methods to mimic natural structures. In: *Advanced Materials* 28 (2016), Nr. 30, S. 6292–6321
- [ZHAN17] Zhang, Z.; Tavenard, R.; Bailly, A.; Tang, X.; Tang, P.; Corpetti, T.: Dynamic Time Warping under limited warping path length. In: *Information Sciences* 393 (2017), S. 91–107. – DOI <https://doi.org/10.1016/j.ins.2017.02.018>. – ISSN 0020–0255
- [ZHAO16] Zhao, J.; Itti, L.: Decomposing time series with application to temporal segmentation. In: *2016 IEEE winter conference on applications of computer vision (WACV)* IEEE, 2016, S. 1–9
- [ZHAO18] Zhao, J.; Itti, L.: shapedtw: Shape dynamic time warping. In: *Pattern Recognition* 74 (2018), S. 171–184
- [ZHOU08] Zhou, H.; Nicholls, C.; Kunz, T.; Schwartz, H.: Frequency accuracy & stability dependencies of crystal oscillators. In: *Carleton University, Systems and Computer Engineering, Technical Report SCE-08-12* (2008)

Anhang

Appendix

A.1 Bildverweise

Die ikonografischen Darstellungen von Auge und Ohr in Abbildung 2.24 entstammt der »Bootstrap Icons« Bibliothek, lizenziert unter der MIT-Lizenz¹.

Darstellungen zu Dynamic Time Warping sind (sofern nicht anders angegeben) selbst erstellt unter Nutzung des *DTW-Python*-Pakets [GIOR09].

¹<https://github.com/twbs/icons/blob/main/LICENSE>

A.2 Aufgaben Data Scientists und Bereinigungsschritte

Aufgabe		Beschreibung
Data Acquisition		Aufnahme der Daten von Sensoren, Sensorintegration, Maschinenschnittstellen, Manuelle Datenaufnahme
Collecting Data Sets		Finden und Sammeln von Datenbasen, Personen nach Quellen fragen
Building Pipelines	Data	Entwicklung von Software für den Transfer von Daten zwischen Systemen, Konfiguration einer IIoT-Plattform, Aufsetzen von Middleware Systemen
Cleaning and Organizing Data		Untersuchung der Datenqualität, Schritte zur Bereinigung und Vorverarbeitung (Siehe Tabelle 2)
Pattern Searching		Data Mining, Datenexploration, explorative statistische Analyse, Mustererkennung
Creationg and Training Models		Auswahl von Modellen, Aufsetzen und Konfiguration von Modellen, (Aktive Zeit für) Modelltraining, Entwicklung statistischer Modelle
Refining and Retraining Models		Evaluierung der Modellgüte, (Aktive Zeit für) Retraining, Wechsel und Neuprogrammierung von Modellen, Anpassung Modellkonfigurationen
Implementing visualizations		Erstellung von Dashboards, Konfiguration der Visualisierungs-Software für live Daten, Erstellung von Präsentationen für User, Performance Monitoring Dashboards und Berichte

Tabelle 1: Die Aufgaben der Data Scientists und Beschreibung aus der Vorstudie. Da die Studie auf Englisch durchgeführt wurde, sind die Namen der Aufgaben auf Englisch beibehalten.

Bereinigungsschritt	Bezug Datenprobleme (Siehe Tabelle 2.1)
Matching data types	Datenformate
Data value conversions	Messgrößen
Filling missing data	Fehlende Werte
Removing outliers	Ausreißer
Synchronizing multiple data streams	Asynchronizität und Integration
Reduction of Data	
Binning values	Aggregationsebenen
Duplicate removals	Dopplungen
Text cleaning	Syntax (z.B. HTML Tags entfernen)
Data normalizationg	Skalierung
Setting up and managing databases	
Asking people for meaning of variables and data point meanings	Fehlender Kontext

Tabelle 2: Die Vorverarbeitungsschritte aus der Vorstudie und der Bezug zu den Datenproblemen, sofern vorhanden. Da die Studie auf Englisch durchgeführt wurde, sind die Namen der Schritte auf Englisch beibehalten.

A.3 Data Science Tasks Survey

Data Science Tasks Survey

* Erforderlich

Intro

Thank you for attending this survey.

This survey should take around 5 minutes and deals with the typical tasks, you as data scientist are facing during your work. It is focusing on the actual data science related work (meaning, business workflows and formalia of your company are not included)

By this, we want to get and publish a fair image of what a typical data scientist job consists of and also what components should be tackled to make the work easier and more joyful. By attending this survey, you agree to the (anonymous) publication.

If you have any questions or want to perform this survey as an interview instead, please feel free to contact me:
arno.schmetz@ffb.fraunhofer.de

Thank you!

Typical Tasks

First of all, think of your typical workload. Based on that, which tasks are you spending most/lest time on?

1. What amount of time are you spending on these tasks? (in percentage of your full time) *

	up to 5%	6%-10%	11%-15%	16%-25%	26%-40%	40%-70%	>70%
Data acquisition (sensor integration, machine interfaces)	☐	☐	☐	☐	☐	☐	☐
Collecting Data sets (databases, other people)	☐	☐	☐	☐	☐	☐	☐
Building Data Pipelines (Configuration of Middleware, etc.)	☐	☐	☐	☐	☐	☐	☐
Cleaning and organizing Data (preprocessing, data quality)	☐	☐	☐	☐	☐	☐	☐
Pattern Searching (Data Mining)	☐	☐	☐	☐	☐	☐	☐
Creating and Training models	☐	☐	☐	☐	☐	☐	☐
Refining and Retraining Models	☐	☐	☐	☐	☐	☐	☐
Implementing Visualizations	☐	☐	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐	☐	☐

2. If you selected ">70%" for one option, can you give an estimate on the actual share of your work regarding that task?

3. Revisiting those tasks - what do you enjoy (or not) to do? *

	least enjoyable			most enjoyable		
Data acquisition (sensor integration, machine interfaces)	☐	☐	☐	☐	☐	
Collecting Data sets (databases, other people)	☐	☐	☐	☐	☐	
Building Data Pipelines (Configuration of Middleware, etc.)	☐	☐	☐	☐	☐	
Cleaning and organizing Data (preprocessing, data quality)	☐	☐	☐	☐	☐	
Pattern Searching (Data Mining)	☐	☐	☐	☐	☐	
Creating and Training models	☐	☐	☐	☐	☐	
Refining and Retraining Models	☐	☐	☐	☐	☐	
Implementing Visualizations	☐	☐	☐	☐	☐	
Other	☐	☐	☐	☐	☐	

4. If you selected "Other" - please tell us more about this!

Deep Dive Preprocessing

Almost done! Let's have a short deep Dive into Data cleaning and organizing

5. Regarding data cleaning and organizing - what steps are you spending most/least time on? *

	Least / No Time			Most / Much Time	
Matching data types (e.g., string to double)	☐	☐	☐	☐	☐
Data Value Conversion (e.g., Fahrenheit to Celsius)	☐	☐	☐	☐	☐
Filling missing data	☐	☐	☐	☐	☐
Removing Outliers	☐	☐	☐	☐	☐
Synchronizing multiple data streams (e.g., matching data from multiple sensors)	☐	☐	☐	☐	☐
Reduction of Data	☐	☐	☐	☐	☐
Binning Values	☐	☐	☐	☐	☐
Duplicate Removals	☐	☐	☐	☐	☐
Text cleaning (e.g., remove HTML Tags or blanks)	☐	☐	☐	☐	☐
Data Normalization	☐	☐	☐	☐	☐
Setting up and Managing Databases	☐	☐	☐	☐	☐
Asking people for meaning of variables / data point naming	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

6. If you have chosen "Other" - please describe this

Final Rating

At last, we also want to hear your thoughts on the Data Science job and whether you are satisfied with being a data scientist

7. How satisfied are you, being a data scientist? *



8. What would you need to be more satisfied, being a data scientist?

9. Last One: Regarding your data sets - from which industries are you getting your data sets usually? *

- ☐ Manufacturing / Production
- ☐ Social Media / Social Networks
- ☐ Shopping / eCommerce
- ☐ Logistics
- ☐ Enterprise Processes
- ☐ Medical
- ☐ Construction
- ☐ Transportation
- ☐ Education
- ☐ Finance / Banking
- ☐ Energy
- ☐ Government
- ☐ Sonstiges

Dieser Inhalt wurde von Microsoft weder erstellt noch gebilligt. Die von Ihnen übermittelten Daten werden an den Formulareigentümer gesendet.

 Microsoft Forms

A.4 Ergebnisse Umfrage Data Scientists Aufgaben, Vergnügen, Vorverarbeitung

Aufgabe	<5%	6% - 10%	11% - 15%	16% - 25%	26% - 40%	40% - 70%	>70%
Data Acquisition	45,7	22,2	17,3	12,3	2,5	0,0	0,0
Collecting Data Sets	32,1	16,0	29,6	21,0	1,2	0,0	0,0
Building Data Pipelines	18,5	37,0	22,2	19,8	2,5	0,0	0,0
Cleaning and Organizing Data	0,0	0,0	4,9	22,2	28,4	35,8	8,6
Pattern Searching	6,2	28,4	33,3	18,5	11,1	2,5	0,0
Creating and Training Models	7,4	31,0	45,7	8,6	6,2	1,2	0,0
Refining and Retraining Models	16,0	32,1	28,4	17,3	4,9	1,2	0,0
Implementing Visualizations	25,9	42,0	19,8	8,6	2,5	1,2	0,0

Tabelle 3: Ergebnisse der Umfrage zur Zeit für die Aufgaben als Data Scientist. Alle Zahlen sind Prozentwerte der abgegebenen Stimmen. Die Option »Sonstige« ist hier ausgelassen.

Aufgabe	1	2	3	4	5	Mittel
Data Acquisition	13,60	18,50	44,40	18,50	4,90	2,83
Collecting Data Sets	7,40	23,50	49,40	18,50	1,20	2,83
Building Data Pipelines	2,50	29,60	44,40	21,00	2,50	2,91
Cleaning and Organizing Data	33,30	50,60	13,60	2,50	0,00	1,85
Pattern Searching	2,50	9,90	22,20	48,10	17,30	3,68
Creating and Training Models	1,20	4,90	22,20	48,10	23,50	3,88
Refining and Retraining Models	1,20	14,80	28,40	42,00	13,60	3,52
Implementing Visualizations	12,30	35,80	30,90	14,80	6,20	2,67

Tabelle 4: Ergebnisse der Umfrage zum Vergnügen für die Aufgaben als Data Scientist. Der Wert 1 entspricht dabei »Am Wenigsten« und Wert 5 »Am Meisten«. Alle Zahlen sind Prozentwerte der abgegebenen Stimmen. Die Option »Sons-tige« ist hier ausgelassen.

Schritt	1	2	3	4	5	Mittel
Matching Data Types	7,4	34,6	43,2	14,8	0	2,65
Data Value Conversion	32,1	43,2	22,2	2,5	0	1,95
Filling Missing Data	2,5	4,9	35,8	44,4	12,3	3,59
Removing Outliers	4,9	30,9	34,6	28,4	1,2	2,90
Synchronizing Multiple Data Streams	0	4,9	16	69,1	9,9	3,84
Reduction of Data	18,5	53,1	16	11,1	1,2	2,23
Binning Values	21	22,2	40,7	16	0	2,52
Duplicate Removals	18,5	37	29,6	13,6	1,2	2,42
Text Cleaning	58	32,1	8,6	1,2	0	1,53
Data Normalization	13,6	38,3	33,3	14,8	0	2,49
Setting up and Managing Databases	35,8	29,6	19,8	14,8	0	2,14
Asking People for Meaning	2,5	8,6	16	43,2	29,6	3,89

Tabelle 5: Ergebnisse der Umfrage zur relativen Arbeitslast der Vorverarbeitungsschritte als Data Scientist. Der Wert 1 entspricht dabei »Am Wenigsten« und Wert 5 »Am Meisten«. Alle Zahlen sind Prozentwerte der abgegebenen Stimmen. Die Option »Sonstige« ist hier ausgelassen.

A.5 Time Synchronization in Smart Manufacturing Survey

Time Synchronization in Smart Manufacturing

* Erforderlich

Intro

Thank you for attending this survey.

This survey should take ca. 5 minutes and deals with the topic of time synchronization in the smart manufacturing. As data from different sources are aggregated for data-based services in production, we want to investigate this topic.

The survey starts with a very short example of the effect. If you have any questions or want to perform this survey as an interview instead, please feel free to contact me: arno.schmetz@ffo.fraunhofer.de

Time drift/error effect of different clocks (Short example)

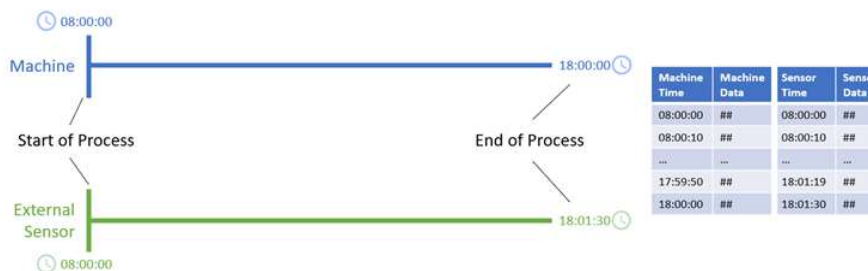
You have two independent systems - e.g., your machine PLC and an external sensor or measurement system.

Both of those systems have their own clocks, showing the same time when you start the process and data acquisition (e.g., 08:00:00)

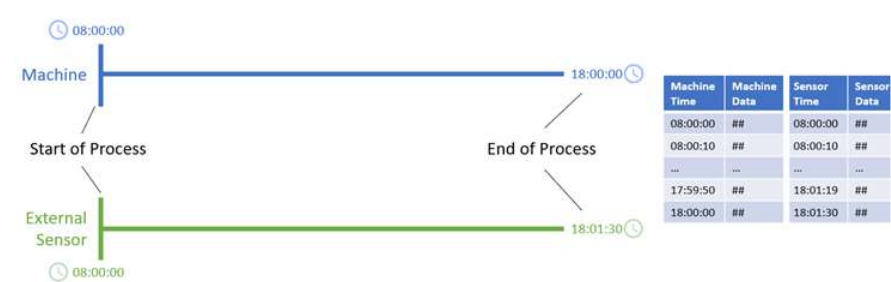
At the end of the process [at the same moment], both clocks show different time (e.g., 18:00:00 for the PLC and 18:01:30 for the external sensor).

Instead of looking at the clocks directly, you might see the last acquired timestamp in the data differing from each other accordingly. (Seen in tables from data acquisition on the right)

This effect results in time series data timestamps not mapping perfectly to each other.



Time Effect Occured?



1

Did you encounter the effect described before? (Your own production, at a customer or in a project)

*

- ☐ Yes, multiple times or regurlaly
- ☐ Yes, once or twice
- ☐ No
- ☐ Not sure

The Effect in Detail

As you encountered this behavior before, please share some of your insights with us

2

Do you recall the time differences in your clocks or data? ("The Error") *

- ☐ Less than a second
- ☐ In order of seconds
- ☐ In order of minutes
- ☐ Even longer
- ☐ Not Sure
- ☐ Sonstiges

3

With which frequency are you usually performing data acquisition? (Machines, sensors, etc.)

For different frequencies you can take the highest acquisition frequency in use

- ☐ One value per minute or less frequent ($<0.02\text{Hz}$)
- ☐ Some values per minute ($<1\text{Hz}$)
- ☐ Roughly one date per second (1Hz)
- ☐ Roughly one date per 100ms (10Hz)
- ☐ Roughly one date per 10ms (100Hz)
- ☐ Roughly one date per 1ms (1kHz)
- ☐ Even more / sub-ms range ($>1\text{kHz}$)
- ☐ Sonstiges

4

How many devices do you have usually in your production, project or customer site with clocks (roughly) *

☐ 2-3 devices

☐ 4-6 devices

☐ 7-10 devices

☐ More than 10

☐ Sonstiges

5

How do/did you tackle the described effect in your data?

6

Are there further information you can share about this effect arising on your side? (e.g., scenarios, implications, errors)

Your Background

Almost finished - please share some information about your company (optional)

7

In which industry/branch are you working in? *

- ☐ Automation
- ☐ Control Manufacturer (PLC, CNC...)
- ☐ Digitalization / Data Analytics
- ☐ Equipment Module Supplier
- ☐ Integrator
- ☐ Measurement / Sensoric Equipment
- ☐ Manufacturer of Goods
- ☐ Manufacturer of Machines
- ☐ Research
- ☐ Software Platform Developer (MES, IIoT, ERP...)
- ☐ Sonstiges

8

(Optional) Which company or institution are you working in?

Results will be published aggregated - no one can see the exact answers linked to you/your company/institution

9

Is there anything, you want to tell us in addition?

Remarks, Hints, Notes, etc.

Dieser Inhalt wurde von Microsoft weder erstellt noch gebilligt. Die von Ihnen übermittelten Daten werden an den Formulareigentümer gesendet.

 Microsoft Forms

A.6 Time Synchronization in Smart Manufacturing Survey – Ergebnisse

Antwort	Anteil
Yes, multiple times	67,3%
Yes, once or twice	28,6%
No	4,1%
Not Sure	0,0%

Tabelle 6: Ergebnisse zur Frage, ob Asynchronizität in den Produktionen aufgekommen ist. Da die Studie auf Englisch durchgeführt wurde, sind die Antwortmöglichkeiten Englisch beibehalten. (n = 49)

Antwort	Anteil
Less than a second	53,2%
Order of seconds	83,0%
Order of minutes	55,3%
Even longer	23,4%
Not Sure	0,0%

Tabelle 7: Ergebnisse zur Frage, in welcher Größenordnung Asynchronizität in den Produktionen aufgekommen ist. Da die Studie auf Englisch durchgeführt wurde, sind die Antwortmöglichkeiten Englisch beibehalten. Die Option »Sonstige« ist hier ausgelassen. (n = 47)

Antwort	Anteil
<0,02 Hz	0,0%
<1 Hz	8,5%
ca. 1 Hz	19,1%
ca. 10 Hz	25,5%
ca. 100 Hz	17,0%
ca. 1 kHz	19,1%
>1 kHz	6,4%

Tabelle 8: Ergebnisse zur Frage, in welcher Frequenz Daten aufgenommen wurden in den Produktionen. Die Option »Sonstige« ist hier ausgelassen. (n = 47)

Antwort	Anteil
2-3 Geräte	25,5%
4-6 Geräte	34,0%
7-10 Geräte	14,9%
Mehr als 10	21,3%

Tabelle 9: Ergebnisse zur Frage der Menge von Geräten in der Produktion. Die Option »Sonstige« ist hier ausgelassen. (n = 47)

A.7 Ablaufdiagramme System

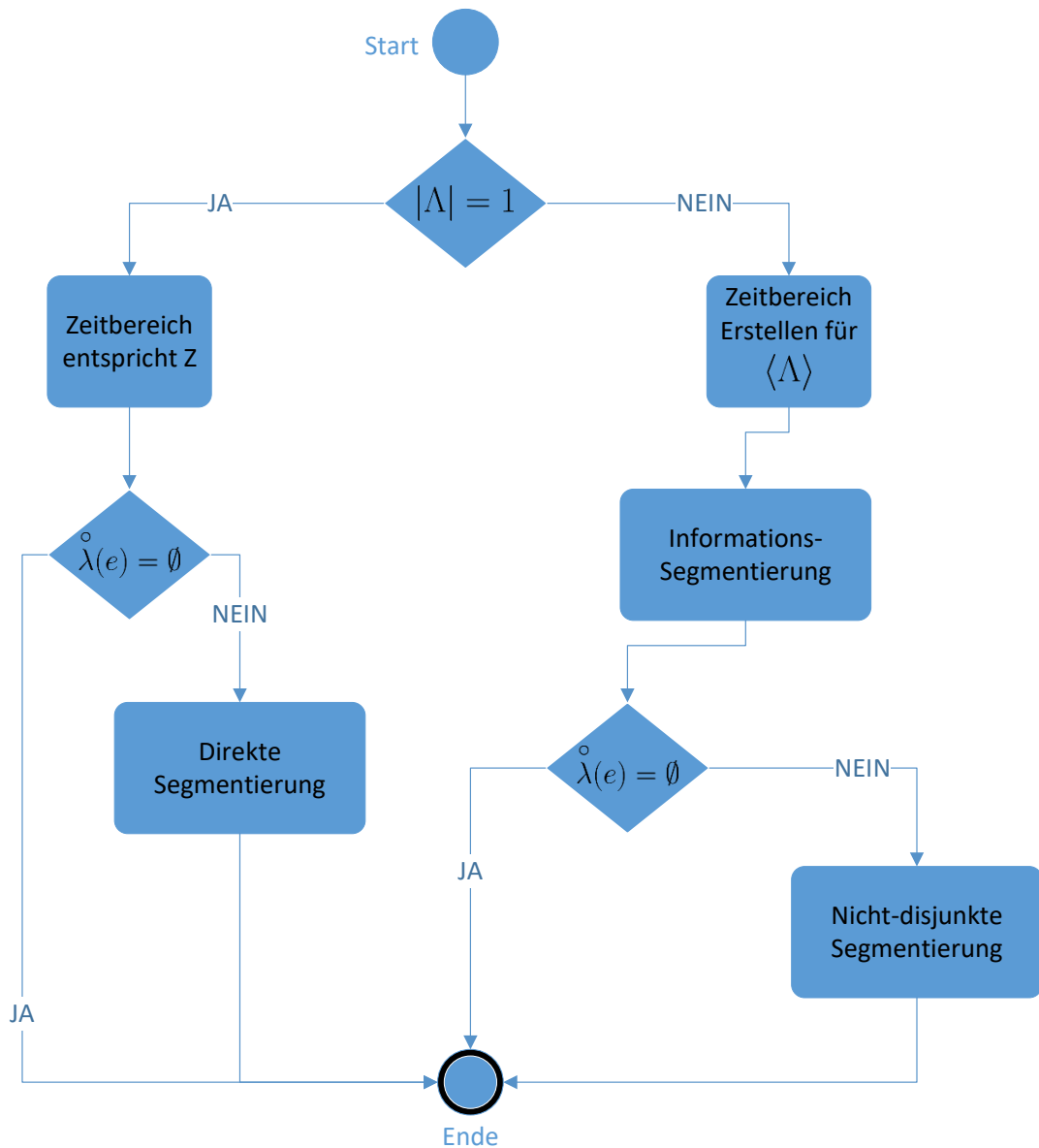


Abbildung A.1: Der Ablauf der Segmentierung in Phase 1 der Bewertung von Metriken für die Synchronisierung.

A.8 Abläufe Methodenbewertung

Die Methodenbewertungen für die Anwendbarkeit (siehe 4.3.3) wird unter Rückgriff auf Informationen und Wissen für jede Methode durchgeführt. In den folgenden Beschreibungen sind die Bewertungen $a_{Methode}$ als Formeln angegeben. Prüfungen von Eigenschaften werden zur Lesbarkeit mittels der Indikatorfunktion $\mathbf{1}_{Funktion}$ dargestellt. In den Formeln dient der Parameter l als Angabe der Länge der Zeitdatenreihen. Die Einordnung basiert dabei auf der Konfiguration des Synchronisierungssystems gemäß Abschnitt A.9.

Methode: Linear

$$a_{Linear} = a_{Linear}(l) + 0.25 \cdot (1 - \text{Spektrale Distanz}) \quad (1)$$

$$a_{Linear}(l) = \begin{cases} 0.5, & \text{falls } l = \text{kurz} \\ 0.6, & \text{falls } l = \text{mittel} \\ 0.4, & \text{falls } l = \text{lang} \\ 0, & \text{falls } l = \text{exzessiv} \end{cases} \quad (2)$$

Für diese Bewertung wird hauptsächlich die Länge der Zeitdatenreihen herangezogen. Zusätzlich kommt die Spektrale Distanz der Zeitdatenreihen im Vergleich zum Einsatz, sodass Reihen, welche eine ähnliche Verteilung der Zeiten umfassen einen Bonus bekommen.

Methode: ARMAX

$$a_{ARMAX} = a_{ARMAX}(l) + 0.1 \cdot \mathbf{1}_{Pearson > 0.7} + 0.1 \cdot XCor + \max(0.8 \cdot \mathbf{1}_{\exists linear \in \mathfrak{W}}, 0.05 \cdot \mathbf{1}_{\exists Lag \in \mathfrak{W}}) \quad (3)$$

$$a_{ARMAX}(l) = \begin{cases} 0.5, & \text{falls } l = \text{kurz} \\ 0.6, & \text{falls } l = \text{mittel} \\ 0.4, & \text{falls } l = \text{lang} \\ 0, & \text{falls } l = \text{exzessiv} \end{cases} \quad (4)$$

Für diese Bewertung kommt zunächst die Länge der Zeitdatenreihen zum Einsatz. Übersteigt die Pearson-Korrelation den definierten Grenzwert, steigt die Bewertung. Ebenfalls kommt

die Kreuzkorrelation in Abhängigkeit ihrer Stärke dazu. Aus dem Wissen wird danach geprüft, ob ein linearer Zusammenhang zwischen den Zeitdatenreihen vorhanden ist oder eine Verzögerung (Lag). In Abhängig dieses Wissens kann die Bewertung entsprechend erhöht werden.

Methode: NARMAX

$$a_{NARMAX} = a_{NARMAX}(l) + 0.1 \cdot \mathbf{1}_{Spearman > 0.7} + 0.1 \cdot XCor + \max(0.8 \cdot \mathbf{1}_{\exists nonlinear \in \mathfrak{W}}, 0.04 \cdot \mathbf{1}_{\exists Lag \in \mathfrak{W}}) \quad (5)$$

$$a_{NARMAX}(l) = \begin{cases} 0.5, & \text{falls } l = \text{kurz} \\ 0.5, & \text{falls } l = \text{mittel} \\ 0.3, & \text{falls } l = \text{lang} \\ 0, & \text{falls } l = \text{exzessiv} \end{cases} \quad (6)$$

Für diese Bewertung kommt zunächst die Länge der Zeitdatenreihen zum Einsatz. Übersteigt die Spearman-Korrelation den definierten Grenzwert, steigt die Bewertung. Ebenfalls kommt die Kreuzkorrelation in Abhängigkeit ihrer Stärke dazu. Aus dem Wissen wird danach geprüft, ob ein nicht-linearer Zusammenhang zwischen den Zeitdatenreihen vorhanden ist oder eine Verzögerung (Lag). In Abhängig dieses Wissens kann die Bewertung entsprechend erhöht werden.

Methode: SSI

$$a_{SSI} = a_{SSI}(l) + 0.1 \cdot NMI + 0.1 \cdot (1 - \text{SpektraleDistanz}) + \max(0.8 \cdot \mathbf{1}_{\exists unbekannt \in \mathfrak{W}}, 0.04 \cdot \mathbf{1}_{\exists Lag \in \mathfrak{W}}, 0.04 \cdot \mathbf{1}_{\exists nonlinear \in \mathfrak{W}}) \quad (7)$$

$$a_{SSI}(l) = \begin{cases} 0.2, & \text{falls } l = \text{kurz} \\ 0.6, & \text{falls } l = \text{mittel} \\ 0.4, & \text{falls } l = \text{lang} \\ 0, & \text{falls } l = \text{exzessiv} \end{cases} \quad (8)$$

Für diese Bewertung kommt zunächst die Länge der Zeitdatenreihen zum Einsatz. Erhöhungen kommen durch die Werte von Normalized Mutual Information und Spektraler Distanz

dazu. Aus dem Wissen wird danach geprüft, ob ein unbekannter Zusammenhang zwischen den Zeitdatenreihen vorhanden ist oder eine Verzögerung (Lag) oder nicht-linearer Zusammenhang. In Abhängig dieses Wissens kann die Bewertung entsprechend erhöht werden.

Methode: DTW

$$a_{DTW} = a_{DTW}(l) + 0.1 \cdot (1 - \text{SpektraleDistanz}) + 0.05 \cdot PMI + 0.05 \cdot \text{Fréchet} + \max(0.1 \cdot \mathbf{1}_{\exists \text{ident} \in \mathfrak{W}}, 0.05 \cdot \mathbf{1}_{\exists \text{Lag} \in \mathfrak{W}}) \quad (9)$$

$$a_{DTW}(l) = \begin{cases} 0.5, & \text{falls } l = \text{kurz} \\ 0.3, & \text{falls } l = \text{mittel} \\ 0.1, & \text{falls } l = \text{lang} \\ 0, & \text{falls } l = \text{exzessiv} \end{cases} \quad (10)$$

Für diese Bewertung kommt zunächst die Länge der Zeitdatenreihen zum Einsatz. Erhöhungen kommen durch die Werte von der Spektralen Distanz, dem Pattern Matching Index und der Fréchet-Distanz zustande. Aus dem Wissen wird danach geprüft, ob ein Zusammenhang zwischen den Zeitdatenreihen vorhanden ist, der eine enge Ähnlichkeit im Verlauf oder eine Verzögerung (Lag) benennt. In Abhängig dieses Wissens kann die Bewertung entsprechend erhöht werden.

Methode: FastDTW

$$a_{FastDTW} = a_{FastDTW}(l) + 0.05 \cdot (1 - \text{SpektraleDistanz}) + 0.1 \cdot PMI + \max(0.8 \cdot \mathbf{1}_{\exists \text{ident} \in \mathfrak{W}}, 0.04 \cdot \mathbf{1}_{\exists \text{Lag} \in \mathfrak{W}}) \quad (11)$$

$$a_{FastDTW}(l) = \begin{cases} 0.1, & \text{falls } l = \text{kurz} \\ 0.5, & \text{falls } l = \text{mittel} \\ 0.3, & \text{falls } l = \text{lang} \\ 0, & \text{falls } l = \text{exzessiv} \end{cases} \quad (12)$$

Für diese Bewertung kommt zunächst die Länge der Zeitdatenreihen zum Einsatz. Erhöhungen kommen durch die Werte von der Spektralen Distanz und dem Pattern Matching Index

zustande. Aus dem Wissen wird danach geprüft, ob ein Zusammenhang zwischen den Zeitdatenreihen vorhanden ist, der eine enge Ähnlichkeit im Verlauf oder eine Verzögerung (Lag) benennt. In Abhängig dieses Wissens kann die Bewertung entsprechend erhöht werden.

Methode: ShapeDTW

$$a_{ShapeDTW} = a_{ShapeDTW}(l) + 0.05 \cdot (1 - Fréchet) + 0.15 \cdot PMI + \max(0.07 \cdot \mathbf{1}_{\exists ident \in \mathfrak{W}}, 0.09 \cdot \mathbf{1}_{\exists Lag \in \mathfrak{W}}) \quad (13)$$

$$a_{ShapeDTW}(l) = \begin{cases} 0.4, & \text{falls } l = \text{kurz} \\ 0.6, & \text{falls } l = \text{mittel} \\ 0.1, & \text{falls } l = \text{lang} \\ 0, & \text{falls } l = \text{exzessiv} \end{cases} \quad (14)$$

Für diese Bewertung kommt zunächst die Länge der Zeitdatenreihen zum Einsatz. Erhöhungen kommen durch die Werte von der Fréchet-Distanz und dem Pattern Matching Index zustande. Aus dem Wissen wird danach geprüft, ob ein Zusammenhang zwischen den Zeitdatenreihen vorhanden ist, der eine enge Ähnlichkeit im Verlauf oder eine Verzögerung (Lag) benennt. In Abhängig dieses Wissens kann die Bewertung entsprechend erhöht werden.

Methode: SoE

Bei der Bewertung Geht es lediglich um die Frage, ob ein Pfad gefunden wird, sodass die beiden Systeme, welche die Zeitdatenreihen repräsentieren mit dem gleichen Knoten verbunden sind, wobei alle Knoten dazwischen Systeme mit eigener Zeitangabe repräsentieren. Die Bewertung entspricht damit exakt der Anwendbarkeitsdefinition. Ist die Methode hier anwendbar, erhält sie automatisch die bestmögliche Bewertung $a_{SoE} = 1$, ansonsten kann die Methode ausgeschlossen werden ($a_{SoE} = 0$).

A.9 Stellgrößen Synchronisierungssystem

Parameter	Beschreibung	Standard-Wert
Local System Settings		
LOC_PythonPath	Pfad zu Python auf dem System	python.exe
LOC_ExchangePath	Temp. Austauschverzeichnis für die Python-Calls	~
OSS General Settings		
OSS_General_ShowWebUi	Globaler Schalter, ob die manuelle Interaktion (FP/Zerlegung und Graph) als Web UI angezeigt werden soll oder nicht	False
Sync Methods Settings		
UseLinear	Soll Methode Linear genutzt werden	True
UseDTW	Soll Methode DTW genutzt werden	True
UseFastDTW	Soll Methode FastDTW genutzt werden	True
UseShapeDTW	Soll Methode ShapeDTW genutzt werden	True
UseARMAX	Soll Methode ARMAX genutzt werden	True
UseNARMAX	Soll Methode NARMAX genutzt werden	True
UseSSI	Soll Methode SSI genutzt werden	True
UseSoE	Soll Methode SoE genutzt werden?	True

Parameter	Beschreibung	Standard-Wert
UseManual	Soll als Backup die Angabe von manueller synch. erlaubt werden?	True
Addresses/Deployment Settings		
ADD_CauseEffectDb	Adresse der UWZ Datenbank	OSS.db
ADD_TimeSeriesDb	Adresse der TimeSeries Datenbank oder Filepath	OSS.db
ADD_SettingsDb	Adresse der Settings Datenbank (Konfiguration) oder Filepath	OSS.db
ADD_Information	Adresse der Informationsdatenbank oder Filepath	OSS.db
ADD_Domain	Adresse der Datenbank der Domänenregeln oder Filepath	OSS.db
ADD_Export	Adresse des Exports für Daten, Informationen und Projektionsfunktion	~
Synchro Call specific Settings (Extern)		
FixedStart	Falls externe Vorgabe existiert, was als Startzeit genutzt werden soll bei der Synchronisierung	DateTime.MinValue
FixedEnd	Falls externe Vorgabe existiert, was als Endzeit genutzt werden soll bei der Synchronisierung	DateTime.MaxValue
GUI/Manual steps and automation		

Parameter	Beschreibung	Standard-Wert
GUI_ShowFixPoints	Entscheidet, ob die Ausführung angehalten wird um die Fixpunkte (GUI) anzuzeigen und auf den menschlichen Input zu warten	True
GUI_ShowRecommendation	Entscheidet, ob die Ausführung angehalten wird um die Empfehlung (GUI) anzuzeigen und auf den menschlichen Input zu warten	True
Rating Config		
RC_Graph_Bonus	Der Bonus, der bei der Graph-Reduktion für existierende Links gegenüber nicht-existierender Links angewendet werden soll	0,05
RC_Seq_Strat	Strategie, wie der Score für einen Link mit mehreren Sequenzen berechnet wird	MAX
RC_Thresh_None_Count	Grenze, unterhalb dieser keine Synchronisierung durchgeführt werden soll	50
RC_Thresh_None_Time	Grenze, unterhalb dieser keine Synchronisierung durchgeführt werden soll	30s
RC_Thresh_Low	Grenze, bis wohin eine Reihe als kurz gilt in der Bewertung	10000
RC_Thresh_Mid	Grenze, bis wohin eine Reihe als mittellang gilt in der Bewertung	200000

Parameter	Beschreibung	Standard-Wert
RC_Thresh_Max	Grenze, aber der eine Zwangstrennung (Zwangs-Sequenzierung) durchgeführt wird	1000001
Fixpunkte		
FP_Speed_Name	Name des Signals, welches als Ortsreferenz/Geschwindigkeit herangezogen wird	Bahngeschwindigkeit_Ist
FP_Max_Window	Maximale Annahme (Suchbereich) für Fixpunkt-Umgebung	100000
Sequenzierung		
SEQ_Multi_Strategy	Die Strategie, mit der Multi-Reihen zerlegt werden, wenn für einen Repräsentanten Fixpunkte existieren	ONLY_FIT
Synchro-spezifische Einstellungen (Intern)		
MAP_MultiMapping_Strategy	Wenn bei einer Synchronisierung für einen Wert	RESAMPLING
LIN_Strategy	Strategie bei der Linearisierung:	HARD

Tabelle 10: Die Stellgrößen des Synchronisierungssystems. Die zweite Spalte beschreibt den entsprechenden Parameter, während die letzte Spalte den Standard-Wert der Implementierung beschreibt. Dies sind nicht die Informationen, Daten und Wissen (Eingangsgrößen), sondern direkte Stellparameter des Softwaresystems selbst.

A.10 Implementierungen und Software-Code

Alle Codes dieser Arbeit sind zu finden unter <https://gitlab.cc-asp.fraunhofer.de/sza/diss-oss-public> im GitLab, gehostet von Fraunhofer. Die Codes sind zur Nachvollziehbarkeit abgelegt, eine Nutzung unterliegt den dort beschriebenen Lizenzbedingungen.

A.11 Verifikation und Validierung – Taxonomie nach BANKS



Abbildung A.2: Die Taxonomie der Methoden für Verifikation, Validierung und Testing nach BANKS. (Entnommen aus [BANK98])

A.12 Verifikation und Validierung – Versuchsmatrix

Versuch	Jitter	Drift	Ausreißer	Stufen	Sprünge
0	0	3	2	0	0
1	1	0	2	0	0
2	1	0	0	1	0
3	2	1	0	2	2
4	2	1	0	0	1
5	0	4	2	1	2
6	1	0	0	2	1
7	0	1	0	0	0
8	0	5	2	2	1
9	2	3	0	2	1
10	0	3	0	2	0
11	0	0	1	2	0
12	2	4	2	2	0
13	1	5	2	1	2
14	2	2	1	0	1
15	2	2	2	2	0
16	0	1	2	0	1
17	1	4	1	0	1
18	1	5	2	0	2
19	2	2	0	0	2

Versuch	Jitter	Drift	Ausreißer	Stufen	Sprünge
20	1	5	0	1	2
21	0	0	2	1	2
22	0	4	1	0	0
23	2	4	1	1	2
24	0	2	2	0	1
25	2	3	0	1	0
26	1	2	0	1	1
27	2	5	1	0	0
28	1	3	1	2	2
29	1	4	1	2	0
30	1	3	1	1	2
31	1	1	1	1	1
32	0	1	0	2	2
33	2	0	1	1	1
34	2	5	1	1	2
35	0	2	2	2	1

Tabelle 11: Versuchsmatrix für die Validierung und ihre Störgrößen

