



OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

EIT

FAKULTÄT FÜR
ELEKTROTECHNIK UND
INFORMATIONSTECHNIK

Analytische Modellierung des Zeitverhaltens von verteilten industriellen Steuerungssystemen

Dissertation

zur Erlangung des akademischen Grades

**Doktoringenieur
(Dr.-Ing.)**

von Dipl.-Ing. Stephan Höme

geboren am 14.05.1985 in Weißenfels

genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik
der Otto-von-Guericke-Universität Magdeburg

Gutachter: Prof. Dr.-Ing. Christian Diedrich

Prof. Dr.-Ing. habil. Martin Wollschläger

Promotionskolloquium am 01.04.2016

Vorwort

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl Integrierte Automation der Otto-von-Guericke-Universität Magdeburg und am Institut für Automation und Kommunikation e.V. Dabei wurde ich von vielen Menschen unterstützt, bei denen ich mich hiermit recht herzlich bedanken möchte.

Mein besonderer Dank gilt Prof. Christian Diedrich, der mich bei dieser Arbeit wissenschaftlich begleitet hat. Die gute Zusammenarbeit und die vielen Diskussionen haben mir sehr geholfen.

Ich danke Prof. Martin Wollschläger, Technische Universität Dresden, für die Übernahme des Zweitgutachtens.

Weiterhin bedanke ich mich bei den Kollegen vom Lehrstuhl Integrierte Automation für die Unterstützung sowie das sehr gute Arbeitsklima. Dabei möchte ich besonders Dr. Thomas Hadlich und Jens Grützner für die vielen Gespräche, Diskussionen und ihr intensives Korrekturlesen der Arbeit danken. Sie haben mir viele Anregungen gegeben und mich bei der Lösung der Probleme unterstützt.

Besonderer Dank gilt meiner Frau Nadine für den Rückhalt und die Geduld während der Fertigstellung meiner Arbeit sowie meiner Mutter Antje und meinen Großeltern Irmgard und Bernd-Dieter Höme, die mir diesen Weg erst ermöglicht und mich immer unterstützt haben.

Danke!

Inhaltsverzeichnis

VORWORT	III
INHALTSVERZEICHNIS	V
ABBILDUNGSVERZEICHNIS	XI
TABELLENVERZEICHNIS	XV
KURZFASSUNG	XVII
ABSTRACT	XIX
1 EINLEITUNG UND MOTIVATION	1
1.1 ZIEL DER ARBEIT	2
1.2 GLIEDERUNG DER ARBEIT	3
2 GRUNDLAGEN UND STAND DER WISSENSCHAFT	5
2.1 BEGRIFFSDEFINITIONEN	5
2.1.1 <i>System</i>	5
2.1.2 <i>Signal und Signalpfad</i>	6
2.1.3 <i>Signaltypen</i>	7
2.1.4 <i>Echtzeit</i>	8
2.1.5 <i>Determinismus</i>	9
2.1.6 <i>Isochronität und Gleichzeitigkeit</i>	9
2.2 INDUSTRIELLE STEUERUNGSSYSTEME	11
2.2.1 <i>Klassifikation nach Architektur</i>	11
2.2.2 <i>Klassifikation nach Verhalten</i>	12
2.2.3 <i>Weitere Klassifikationen</i>	13
2.3 INDUSTRIELLE KOMMUNIKATIONSSYSTEME	14
2.3.1 <i>Klassifikation nach Zeitverhalten</i>	15
2.3.2 <i>Klassifikation nach Architektur</i>	16
2.3.3 <i>Zuordnung von Zeitverhalten zu Architekturen</i>	18
2.3.4 <i>Klassifikation nach minimaler Zykluszeit</i>	19
2.4 SYNCHRONISATION IN VERTEILTEN AUTOMATISIERUNGSSYSTEMEN	20
2.4.1 <i>Synchronisation</i>	20
2.4.2 <i>Verfahren zur Zeitsynchronisation in der Automatisierungstechnik</i>	22
2.4.2.1 <i>Verfahren zur Uhrzeitsynchronisation über Industrial Ethernet</i>	22
2.4.2.2 <i>Verfahren zur Maschinentaktsynchronisation über Industrial Ethernet</i>	24
2.5 MODELLIERUNG UND ZEITLICHE ANALYSE VON KOMMUNIKATIONSSYSTEMEN	26
2.5.1 <i>Analytische Verfahren</i>	26
2.5.2 <i>Simulative Verfahren</i>	27
2.5.2.1 <i>Modellierung mit Petrinetzen</i>	27
2.5.2.2 <i>Sonstige Simulationsverfahren und -werkzeuge</i>	27
2.5.3 <i>Messtechnische Analyse</i>	29
2.6 GRUNDLAGEN DER MODELLIERUNG VON DYNAMISCHEN SYSTEMEN	31
2.6.1 <i>LTI-Systeme</i>	31
2.6.2 <i>Beschreibung von linearen Systemen</i>	32
2.6.3 <i>Regelkreis</i>	33
2.6.4 <i>Stabilität</i>	34
2.6.5 <i>Regelgüte - Quality of Control</i>	34

2.7	NETWORKED CONTROL SYSTEMS	36
2.7.1	<i>Besonderheiten von NCS</i>	36
2.7.1.1	Bandbreitenbegrenzte Kanäle (Band-limited channels)	36
2.7.1.2	Systemarchitektur (Systems architecture)	37
2.7.1.3	Abtastung und Verzögerung (Sampling and Delay)	37
2.7.1.4	Paketverlust (Packet dropout)	38
2.7.2	<i>Kompensation von NCS</i>	38
2.8	HÄUFIGKEITSVERTEILUNGEN	39
2.8.1	<i>Rechteckverteilung</i>	40
2.8.2	<i>Dirac-Verteilung</i>	41
2.8.3	<i>Diskrete Gleichverteilung</i>	42
2.9	FAZIT	43
3	KENNGRÖßEN VON VERTEILTEN STEUERUNGSSYSTEMEN	45
3.1	KLEMME-KLEMME-REAKTIONSZEIT (IO RESPONSE TIME) T_{IORT}	46
3.2	FRAMELÄNGE ΔT_{FRAME}	46
3.3	WEITERLEITUNGSDAUER (BRIDGE DELAY) ΔT_{BD}	47
3.4	LINE DELAY ΔT_{LD}	48
3.5	SIGNALLAUFZEIT (PROPAGATION DELAY) ΔT_{PD}	48
3.6	ÜBERTRAGUNGSZEIT (TRANSMISSION DELAY) ΔT_{TD}	48
3.7	ÜBERGANGSZEIT (TRANSITION PERIOD) ΔT_{TP}	49
3.8	ZYKLUSZEIT	50
3.9	VERZÖGERUNGSZEIT DER STEUERUNG (PLC TIME) ΔT_{PLC}	50
3.10	ECHTZEIT-DATENDURCHSATZ	51
3.11	NRT-BANDBREITE	51
3.12	FAZIT	51
4	MODELLIERUNG VON DEZENTRALEN / VERTEILTEN STEUERUNGSSYSTEMEN	53
4.1	EINFÜHRUNG	53
4.1.1	<i>Asynchrone Zyklen</i>	53
4.1.2	<i>Verarbeitungs- und Wartezeiten</i>	55
4.1.3	<i>Einlesen von kontinuierlichen und diskreten Signalen</i>	56
4.2	MODELL DER ASYNCHRONEN PROZESSE	57
4.2.1	<i>Statischer Modellanteil (Strukturmodell)</i>	57
4.2.2	<i>Dynamischer Modellanteil (Signalpfade)</i>	61
4.2.3	<i>Beschreibung der Modellparameter</i>	63
4.2.3.1	Klasse Zeitdomäne	63
4.2.3.2	Klasse Teilprozess	63
4.2.3.3	Klasse Teilsystem	64
4.2.4	<i>Auswahl der Teilsysteme und Teilprozesse</i>	64
4.3	ABLEITUNG DES ZEITVERHALTENS AUS DEM MODELL	65
4.3.1	<i>Prinzip der Überführung</i>	65
4.3.2	<i>Over- und Undersampling in der Signalkette</i>	67
4.3.3	<i>Zeitverhalten am Übergang zwischen synchronen Prozessen</i>	69
4.3.3.1	Synchrone Teilprozesse mit gleicher Zykluszeit	69
4.3.3.2	Synchrone Teilprozesse mit unterschiedlichen Zykluszeiten	70
4.3.3.3	Synchrone Teilprozesse - Zusammenfassung	73
4.3.4	<i>Zeitverhalten am Übergang zwischen asynchronen Teilprozessen</i>	74
4.3.5	<i>Überführung von analytischer in stochastische Beschreibung</i>	74
4.3.6	<i>Anbindung an den technischen Prozess</i>	75

4.3.7	<i>Ablauf der Überführung</i>	75
4.4	BERECHNUNG AUS HÄUFIGKEITSVERTEILUNGEN	77
4.4.1	<i>Prinzip</i>	77
4.4.2	<i>Typische Häufigkeitsverteilungen</i>	78
4.4.3	<i>Faltungsoperation</i>	78
4.4.4	<i>Analytische Faltung von Häufigkeitsverteilungen</i>	79
4.4.4.1	Grundlagen	79
4.4.4.2	Faltungen mit Dirac-Verteilung	79
4.4.4.3	Faltung mit diskreter Gleichverteilung	81
4.4.4.4	Faltung von Rechteckverteilungen	85
4.4.4.5	Vorgehensweise bei der analytischen Faltung	90
4.4.5	<i>Numerische Faltung von Häufigkeitsverteilungen</i>	90
4.4.5.1	Grundlagen numerische Faltung	90
4.4.5.2	Vorgehensweise	91
4.4.5.3	Beispiel: Faltung von zwei Rechteckverteilungen	91
4.4.6	<i>Fazit</i>	93
4.5	ANALYTISCHE BERECHNUNG DES ZEITVERHALTENS MITTELS MODULO-ARITHMETIK	94
4.6	MÖGLICHE ERWEITERUNGEN DES MODELLS	94
4.7	FAZIT	95
5	ISOCHRONE STEUERUNGSSYSTEME	97
5.1	UMSETZUNGSKONZEPTE VON ISOCHRONEM VERHALTEN	97
5.2	MODELLIERUNG ISOCHRONER STEUERUNGSSYSTEME	98
5.3	SPEZIELLES MODELL FÜR ISOCHRONE STEUERUNGSSYSTEME	99
5.3.1	<i>Allgemeines Modell</i>	99
5.3.2	<i>Modell von PROFINET IRT</i>	101
5.3.3	<i>Modell von EtherCAT Distributed Clocks</i>	102
5.4	VERGLEICH ISOCHRONES MODELL UND MODELL ASYNCHRONER PROZESSE	102
5.5	FAZIT	104
6	PARAMETRIERUNG DER MODELLE	105
6.1	PARAMETER	105
6.1.1	<i>Zykluszeit</i>	105
6.1.2	<i>Verarbeitungszeit</i>	105
6.1.3	<i>Phase</i>	106
6.1.4	<i>Genauigkeit und Präzision der Zeitdomäne</i>	106
6.2	BESTIMMUNG DER VERARBEITUNGSZEIT	106
6.2.1	<i>Ein- und Ausgabebaugruppen</i>	106
6.2.2	<i>Kommunikationssysteme</i>	107
6.2.3	<i>Steuerungen</i>	107
6.2.4	<i>Granularität von Verarbeitungszeiten</i>	107
6.3	MESSTECHNISCHE BESTIMMUNG VON PARAMETERN	108
6.3.1	<i>Messprinzip Klemme-Klemme-Reaktionszeit</i>	108
6.3.1.1	Zustandsdiskretes Testsignal	108
6.3.1.2	Zustandskontinuierliches Testsignal	111
6.3.2	<i>Messung von Teilstrecken</i>	112
6.3.3	<i>Messung der Synchronisationsgenauigkeit</i>	113
6.4	ABLEITEN VON PARAMETERN UND EIGENSCHAFTEN AUS MESSDATEN	114
6.4.1	<i>Extremwerte der Verteilung</i>	114
6.4.2	<i>Analyse der Kurvenform</i>	114

6.4.3	<i>Numerische Entfaltung</i>	115
6.4.4	<i>Grenzen der Ableitung</i>	116
6.5	NUTZUNG VORHANDENER BESCHREIBUNGSSYSTEME	116
6.5.1	<i>Gerätebeschreibungssprachen</i>	116
6.5.2	<i>AutomationML</i>	117
6.6	FAZIT	118
7	MODELLIERUNGSBEISPIELE	119
7.1	MODELLIERUNG, BERECHNUNG UND MESSUNG VON BEISPIELSSYSTEMEN	119
7.1.1	<i>Szenarien</i>	119
7.1.1.1	Szenario A: PROFINET IRT mit taktsynchronen EAs	119
7.1.1.2	Szenario B: PROFINET IRT ohne taktsynchrone EAs	120
7.1.1.3	Szenario C: PROFINET RT	120
7.1.1.4	Szenario D: Verteiltes PROFINET-System.....	120
7.1.2	<i>Modellierung und Ableitung des Zeitverhaltens</i>	121
7.1.2.1	Szenario A: PROFINET IRT mit taktsynchronen EAs	121
7.1.2.2	Szenario B: PROFINET IRT ohne taktsynchrone EAs	123
7.1.2.3	Szenario C: PROFINET RT	125
7.1.2.4	Szenario D: Verteiltes PROFINET-System.....	127
7.1.3	<i>Messergebnisse und Vergleich zur Simulation</i>	129
7.1.3.1	Szenario A: PROFINET IRT mit taktsynchronen EAs	130
7.1.3.2	Szenario B: PROFINET IRT ohne taktsynchrone EAs	131
7.1.3.3	Szenario C: PROFINET RT	131
7.1.3.4	Szenario D: Verteiltes PROFINET-System.....	132
7.2	ANALYSE EINES UNBEKANNTEN SYSTEMS AUS MESSDATEN.....	133
7.2.1	<i>Analyse einer Trapezverteilung</i>	133
7.2.2	<i>Analyse einer Stufenverteilung</i>	135
7.2.2.1	Variante 1	136
7.2.2.2	Variante 2	139
7.2.3	<i>Numerische Entfaltung</i>	141
7.2.3.1	Entfaltung einer idealen Häufigkeitsverteilung	141
7.2.3.2	Entfaltung von realen Messwerten	142
7.3	FAZIT	142
8	ANWENDUNG DES MODELLS ZUR BESTIMMUNG DER REGELGÜTE	143
8.1	KENNGRÖßEN ZUR BEURTEILUNG DER REGELGÜTE	143
8.1.1	<i>Stabilität</i>	143
8.1.2	<i>Störkompensation und Sollwertfolge</i>	144
8.1.3	<i>Dynamikforderungen</i>	145
8.1.4	<i>Robustheitsforderungen</i>	146
8.1.5	<i>Fokus dieser Untersuchungen</i>	146
8.2	ANALYSEMÖGLICHKEITEN	147
8.3	BEISPIEL: REGELUNG EINES PORTALKRANS MIT PROFINET	147
8.3.1	<i>Portalkran</i>	147
8.3.2	<i>Steuerungssystem</i>	148
8.3.2.1	Zeitverhalten des Steuerungssystems - Szenario A	149
8.3.2.2	Zeitverhalten des Steuerungssystems - Szenario B	149
8.3.2.3	Zeitverhalten des Steuerungssystems - Szenario C	150
8.3.3	<i>Stabilität</i>	151
8.3.4	<i>Sprungantworten</i>	151
8.3.5	<i>Kenngrößen der Regelgüte</i>	154

9	ZUSAMMENFASSUNG UND AUSBLICK.....	157
	LITERATURVERZEICHNIS	159
ANHANG A	SIMULATION DES ZEITVERHALTENS AM ZYKLUSÜBERGANG	171
ANHANG A.1	ZUSAMMENFASSEN SYNCHRONER TEILPROZESSE.....	171
ANHANG A.2	SIMULATION VON EINEM ASYNCHRONEN ZYKLUSÜBERGANG	174
ANHANG A.3	SIMULATION VON ZWEI ASYNCHRONEN ZYKLUSÜBERGÄNGEN	175
ANHANG A.4	SIMULATION VON DREI ASYNCHRONEN ZYKLUSÜBERGÄNGEN.....	179
ANHANG B	SIMULATION DES EINFLUSS VON MESSRAUSCHEN.....	183
ANHANG C	VERWENDETE HARDWARE FÜR MESSUNG MIT PROFINET	186
ANHANG D	SIMULINK-MODELL DES PORTALKRANS	187
ANHANG E	HÄUFIGKEITSVERTEILUNGEN DER REGELGÜTE-KRITERIEN DES PORTALKRANS	188

Abbildungsverzeichnis

Abbildung 1: Druckverlauf beim Formen von Plastikbechern [Tre13, S. 19]	2
Abbildung 2: System [Had15, S. 9]	5
Abbildung 3: Anwendung der Definition „System“ auf ein industrielles Steuerungssystem.....	5
Abbildung 4: Beispielhafte Darstellung von zwei Signalpfaden durch ein Steuerungssystem	6
Abbildung 5: Einteilung von Signalen [Syr05, S. 36].....	7
Abbildung 6: Prinzip der Isochronität nach [HDD+13]	9
Abbildung 7: Prinzip der Gleichzeitigkeit von Ausgangssignalen	10
Abbildung 8: Prinzip der synchronen Abarbeitung des Steuerungsprogramms.....	10
Abbildung 9: Prinzip der Taktsynchronität.....	11
Abbildung 10: Beispiel eines netzwerkbasierten Automatisierungssystems (NAS) [Gre07].....	13
Abbildung 11: Klassifizierung des Entwicklungsaufwandes bei Industrial Ethernet [And06, S. 32]	16
Abbildung 12: ISO/OSI-Referenzmodell bei Anwendung im Industrial Ethernet [Run14, S. 19]	16
Abbildung 13: Einteilung der Herstellerkonzepte für Echtzeit-Ethernet nach [JSW07, S. 17].....	17
Abbildung 14: Gegenüberstellung von Architekturkonzepten für Industrial Ethernet [And06, S. 42].	18
Abbildung 15: Minimale Zykluszeiten als Funktion der Geräteanzahl [RGR+12].....	19
Abbildung 16: Beispielhafte Darstellung von Genauigkeit und Präzision [HKD14].....	21
Abbildung 17: Synchronisationsprinzip mit NTP nach [HKD14]	23
Abbildung 18: Synchronisationsprinzip von PTP [GM03, S. 88]	24
Abbildung 19: Timer-Modell von PTP bei PROFINET gemäß IEC 61158-6-10 nach [Hir14]	25
Abbildung 20: Messung der Synchronisationsgenauigkeit von PTP für 3 Devices [HKD14]	25
Abbildung 21: Vergleich der Antwortzeitverteilungen [GLF07]	26
Abbildung 22: Modell eines invertierten Pendels in NCLib [WLF08, S. 121].....	28
Abbildung 23: Jitterbug-Modell eines einfachen Regelungssystems [CHL+03, S. 19]	29
Abbildung 24: Beispiel einer Markow-Kette zur Beschreibung eines Netzwerks [Nil98, S. 32].....	30
Abbildung 25: Verzögerungen in den Zuständen der Markow-Kette [Nil98, S. 32].....	30
Abbildung 26: Strukturbild des Systems [Lun14b, S. 55]	31
Abbildung 27: Blockschaltbild eines Systems [Lun14b, S. 42].....	33
Abbildung 28: Grundstruktur des Regelkreises [Lun14b, S. 4].....	33
Abbildung 29: Einfluss der Verzögerung auf die Regelgüte [TC03, S. 1103]	35
Abbildung 30: Klemme-Klemme-Reaktionszeit im Regelkreis	35
Abbildung 31: Allgemeine NCS-Architektur [HNY07, S. 138]	37
Abbildung 32: Ein-Schleifen-NCS [HNY07, S. 139].....	37
Abbildung 33: NCS in direkter Struktur [TC03, S. 1100].....	37
Abbildung 34: NCS in hierarchischer Struktur [TC03, S. 1101].....	37
Abbildung 35: Vom Ein-Schleifen-NCS abgedeckte Architekturen [HNY07, S. 144]	37
Abbildung 36: NCS-Modell mit Netzwerk-induzierter Verzögerung [ZBP01, S. 88].....	38
Abbildung 37: Beispieldarstellung einer Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit ..	39
Abbildung 38: Beispiel für Rechteckverteilung ($\sigma=2$ ms; $T=5$ ms).....	40
Abbildung 39: Beispiel für Dirac-Verteilung ($t_0=2$ ms)	41
Abbildung 40: Beispiel für diskrete Gleichverteilung ($t_i \in \{2$ ms; 4 ms; 6 ms}).....	42
Abbildung 41: Modell eines dezentralen Steuerungssystems mit Kenngrößen nach [HDD+13].....	45
Abbildung 42: Klemme-Klemme-Reaktionszeit aus systemtheoretischer Sichtweise nach [HDD+13].	46

Abbildung 43: Aufbau eines Ethernet-Pakets [IEE12, S. 53]	47
Abbildung 44: Bridge Delay und Line Delay [DLK+14].....	47
Abbildung 45: Übergangszeit Δt_{TP} [DLK+14].....	50
Abbildung 46: Prinzipielle Darstellung der Zyklen eines Steuerungssystems	53
Abbildung 47: Entstehung von asynchronen Zyklen durch abweichende Taktgeber	54
Abbildung 48: Gliederung der entlang eines Signalpfades auftretenden Verzögerungszeiten	55
Abbildung 49: Zeitverhalten beim Einlesen zustandskontinuierlicher Signale (Prinzip-Skizze).....	56
Abbildung 50: Zeitverhalten beim Einlesen zustandsdiskreter Signale (Prinzip-Skizze) [HD12].....	56
Abbildung 51: Klassendiagramm des Strukturmodells	57
Abbildung 52: Statischer Modellanteil eines einfachen dezentralen Steuerungssystems	59
Abbildung 53: Zuordnung der Teilsysteme zum Steuerungssystem	59
Abbildung 54: Zuordnung der Teilprozesse zur Steuerungsanwendung	59
Abbildung 55: Vereinfachte Darstellung des statischen Modellanteils aus Abbildung 52	60
Abbildung 56: Dynamischer Modellanteil als Sequenzdiagramm.....	61
Abbildung 57: Dynamischer Modellanteil eines einfachen dezentralen Steuerungssystems	61
Abbildung 58: Beispielsystem mit 2 Signalpfaden	62
Abbildung 59: Dynamischer Modellanteil für Beispielsystem mit 2 Signalpfaden.....	62
Abbildung 60: Unterschied zwischen Zykluszeit und Verarbeitungszeit.....	63
Abbildung 61: Funktion der Phasenlage im Modell	64
Abbildung 62: Mögliche Wartezeiten (blau) und Verarbeitungszeiten (grün) im Modell	65
Abbildung 63: Zuordnung von Verarbeitungs- und Wartezeiten am Beispielsmodell	66
Abbildung 64: Aufsummierung von Verzögerungszeiten entlang eines Signalpfades.....	66
Abbildung 65: Prinzip des Oversampling beim Zyklusübergang	68
Abbildung 66: Prinzip des Undersampling beim Zyklusübergang	68
Abbildung 67: Übergang zwischen synchronen Prozessen ohne Zyklussprung.....	69
Abbildung 68: Übergang zwischen synchronen Prozessen mit Zyklussprung.....	69
Abbildung 69: Umrechnung der Phase bei synchronen Teilprozessen.....	71
Abbildung 70: Beispiel synchrone Teilprozesse mit gemeinsamen Basistakt (1).....	72
Abbildung 71: Beispiel synchrone Teilprozesse mit gemeinsamen Basistakt (2).....	73
Abbildung 72: Konstante Verarbeitungszeit $t_{process}$ als Dirac-Verteilung.....	75
Abbildung 73: Schritt 1: Zusammenfassen von synchronen Teilsystemen am Beispiel	76
Abbildung 74: Schritt 2: Wartezeiten zwischen asynchronen Teilprozessen bestimmen	76
Abbildung 75: Schritt 3: Stochastische Wartezeit am Eingang einfügen	76
Abbildung 76: Schritt 4: Verarbeitungszeiten als Häufigkeitsverteilungen angeben.....	77
Abbildung 77: Zwei diskrete Gleichverteilungen (Beispiel 1).....	82
Abbildung 78: Resultierende Verteilung (Beispiel 1)	82
Abbildung 79: Zwei diskrete Gleichverteilungen (Beispiel 2).....	83
Abbildung 80: Resultierende Verteilung (Beispiel 2)	83
Abbildung 81: Faltung einer diskreten Gleichverteilung mit einer Rechteckverteilung	85
Abbildung 82: Trapezverteilung ($T_1 \neq T_2$)	87
Abbildung 83: Dreiecksverteilung ($T_1 = T_2$).....	87
Abbildung 84: Beispielhafte Häufigkeitsverteilung für drei asynchrone Zyklusübergänge	88
Abbildung 85: Diskrete Rechteckverteilungen	92
Abbildung 86: Diskrete Faltung von zwei Rechteckverteilungen	92
Abbildung 87: Diskrete Faltung von zwei Rechteckverteilungen mit Schrittweite $\Delta t = 0,01$ ms.....	93

Abbildung 88: Modellierung der Paketverlustrate PLR durch die Verarbeitungszeit	95
Abbildung 89: Beispielmodell für isochrones Steuerungssystem	98
Abbildung 90: Modell für isochrone Steuerungssysteme nach [HDD+13].....	99
Abbildung 91: 3-T-Modell von PROFINET IRT nach [HDD+13]	101
Abbildung 92: IORT-Modell für EtherCAT [KDL13].....	102
Abbildung 93: Zuordnung des isochronen Modells zum Modell der asynchronen Prozesse	103
Abbildung 94: Zuordnung des 3-T-Modells zum Modell asynchroner Prozesse.....	104
Abbildung 95: Messaufbau zur Bestimmung der Klemme-Klemme-Reaktionszeit nach [HD12]	108
Abbildung 96: Bestimmung der Klemme-Klemme-Reaktionszeit mit Rechteck-Testsignal.....	109
Abbildung 97: Beispiel einer Klemme-Klemme-Reaktionszeit-Messung mittels Rechtecksignal	109
Abbildung 98: Häufigkeitsverteilung t_{IORT} für isochrones Steuerungssystem (diskretes Testsignal) ..	110
Abbildung 99: Messung von Klemme-Klemme-Reaktionszeit mittels Dreiecks-Testsignal	111
Abbildung 100: IORT für isochrones Steuerungssystem (zustandskontinuierliches Testsignal).....	112
Abbildung 101: Messaufbau für den Test von PTP [GM03, S. 93]	113
Abbildung 102: Gemessene Abweichung zwischen Master und Slave bei PTP [GM03, S. 94]	113
Abbildung 103: Physikalische und logische Sicht in AutomationML [LRD+13, S. 4].....	117
Abbildung 104: PROFINET-Demonstrator in AutomationML [Aut14, S. 68]	118
Abbildung 105: Architektur des zentralen Steuerungssystems mit PROFINET	119
Abbildung 106: Architektur des verteilten Steuerungssystems in Szenario D.....	120
Abbildung 107: Statischer Modellanteil für Szenario A	121
Abbildung 108: Dynamischer Modellanteil für Szenario A	121
Abbildung 109: Vereinfachter statischer Modellanteil für Szenario A.....	122
Abbildung 110: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit in Szenario A	123
Abbildung 111: Statischer Modellanteil für Szenario B.....	123
Abbildung 112: Zusammengefasster statischer Modellanteil für Szenario B	124
Abbildung 113: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit von Szenario B	125
Abbildung 114: Statischer Modellanteil für Szenario C.....	125
Abbildung 115: Zusammengefasster statischer Modellanteil für Szenario C	126
Abbildung 116: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit von Szenario C	127
Abbildung 117: Statischer Modellanteil für Szenario D	127
Abbildung 118: Parametrierung des Modells von Szenario D mittels 3-T-Modell.....	128
Abbildung 119: Zusammengefasster statischer Modellanteil für Szenario D.....	129
Abbildung 120: Gemessene Verteilung der Klemme-Klemme-Reaktionszeit für Szenario A	130
Abbildung 121: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit für Szenario A	130
Abbildung 122: Gemessene Verteilung der Klemme-Klemme-Reaktionszeit für Szenario B	131
Abbildung 123: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit für Szenario B.....	131
Abbildung 124: Gemessene Verteilung der Klemme-Klemme-Reaktionszeit für Szenario C.....	131
Abbildung 125: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit für Szenario C.....	131
Abbildung 126: Gemessene Verteilung der Klemme-Klemme-Reaktionszeit in Szenario D.....	132
Abbildung 127: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit von Szenario D	132
Abbildung 128: Klemme-Klemme-Reaktionszeit eines unbekanntes Systems.....	133
Abbildung 129: Geradeapproximation der Trapezverteilung des unbekanntes Systems	134
Abbildung 130: Mögliches Systemmodell des unbekanntes Systems	135
Abbildung 131: Klemme-Klemme-Reaktionszeit als Stufenverteilung.....	135
Abbildung 132: Stufenverteilung mit eingezeichneten Rechteckverteilungen – Variante 1	136

Abbildung 133: Stufenverteilung zerlegt in Rechteck- und Diskrete Verteilung – Variante 1	136
Abbildung 134: Stufenverteilung zerlegt in Rechteck-, diskrete Gleich- und Dirac-Verteilung	137
Abbildung 135: Mögliche Modell für Signalpfad mit Stufenverteilung – Variante 1	138
Abbildung 136: Stufenverteilung mit eingezeichneter Rechteckverteilung – Variante 2	139
Abbildung 137: Stufenverteilung zerlegt in Rechteck-, diskrete Gleich- und Dirac-Verteilung	139
Abbildung 138: Mögliche Modell für Signalpfad mit Stufenverteilung – Variante 2	140
Abbildung 139: Ideale Häufigkeitsverteilung als Ausgangspunkt der Entfaltung	141
Abbildung 140: Ergebnis der Entfaltung einer idealen Häufigkeitsverteilung	142
Abbildung 141: Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit mit $\max(t_{\text{IORT}}) = 27 \text{ ms}$...	144
Abbildung 142: Führungsübergangsfunktion des Regelkreises [Lun14b, S. 347]	145
Abbildung 143: Übergangsfunktion bei sprunghafter Störung h_d bzw. Führung h_w [Lun14b, S. 348]	146
Abbildung 144: Portalkran [Lun14b, S. 128].....	147
Abbildung 145: Regelkreis mit Zustandsrückführung K und Vorfilter V [Lun14a, S. 145].....	148
Abbildung 146: Architektur des Steuerungssystems zur Portalkran-Regelung.....	149
Abbildung 147: Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit für Szenario B.....	150
Abbildung 148: Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit für Szenario C.....	150
Abbildung 149: Sprungantwort für Szenario A.....	151
Abbildung 150: Kurvenschar der Sprungantworten für Szenario B	152
Abbildung 151: Kurvenschar der Sprungantworten für Szenario C	152
Abbildung 152: Einzelne Sprungantwort für Szenario B ($t=100 \text{ s}$)	153
Abbildung 153: Ausschnitt aus Abbildung 152.....	153
Abbildung 154: Simulink-Modell zur Untersuchung eines synchronen Zyklusübergangs	171
Abbildung 155: Simulierte Häufigkeitsverteilung der Wartezeit für Szenario 1	172
Abbildung 156: Simulierte Häufigkeitsverteilung der Wartezeit für Szenario 2	172
Abbildung 157: Simulierte Häufigkeitsverteilung der Wartezeit für Szenario 3	173
Abbildung 158: Simulierte Häufigkeitsverteilung der Wartezeit für Szenario 4	173
Abbildung 159: Simulink-Modell für einen asynchronen Zyklusübergang.....	174
Abbildung 160: Simulierte Verzögerung durch einen asynchronen Zyklusübergang	174
Abbildung 161: Simulink-Modell für zwei asynchrone Zyklusübergänge	175
Abbildung 162: Simulierte Verzögerung durch zwei asynchrone Zyklusübergänge – Szenario 1.....	176
Abbildung 163: Simulierte Verzögerung durch zwei asynchrone Zyklusübergänge – Szenario 2.....	177
Abbildung 164: Simulierte Verzögerung durch zwei asynchrone Zyklusübergänge	178
Abbildung 165: Simulink-Modell für drei asynchrone Zyklusübergänge	179
Abbildung 166: Simulierte Verzögerung durch drei asynchrone Zyklusübergänge (Szenario 1)	180
Abbildung 167: Simulierte Verzögerung durch drei asynchrone Zyklusübergänge (Szenario 2)	181
Abbildung 168: Simulierte Verzögerung durch drei asynchrone Zyklusübergänge (Szenario 2)	182
Abbildung 169: Häufigkeitsverteilung t_{IORT} für isochrones Steuerungssystem (diskretes Testsignal)	183
Abbildung 170: Simulink-Modell zur Simulation des Messaufbaus für t_{IORT}	183
Abbildung 171: Simulierter zeitlicher Verlauf von Ein- und Ausgangssignal (Ausschnitt)	184
Abbildung 172: Simulierte Klemme-Klemme-Reaktionszeit für Rauschen am Ausgang.....	185
Abbildung 173: Simulink-Modell des geschlossenen Regelkreises des Portalkrans	187
Abbildung 174: Struktur des Regelkreises im Simulink-Modell	187
Abbildung 175: Verteilung der Regelgüte-Kriterien für Szenario B	188
Abbildung 176: Verteilung der Regelgüte-Kriterien für Szenario C	188

Tabellenverzeichnis

Tabelle 1: Klassifizierung der Echtzeitanforderungen nach [Jas05]	15
Tabelle 2: Klassifizierung ausgewählter Verfahren nach der IAONA [And06, S. 43]	18
Tabelle 3: Vergleich von gemessener und berechneter IORT für Szenario A	130
Tabelle 4: Vergleich von gemessener und berechneter IORT für Szenario B.....	131
Tabelle 5: Vergleich von gemessener und berechneter IORT für Szenario C.....	132
Tabelle 6: Vergleich von gemessener und berechneter IORT für Szenario D	132
Tabelle 7: Kenngrößen der Regelgüte	154
Tabelle 8: Prozentuale Änderung der Kenngrößen im Vergleich zum Referenzsystem	154

Kurzfassung

Diese Arbeit beschreibt ein Modell zur analytischen Modellierung des Zeitverhaltens von verteilten industriellen Steuerungssystemen. Für viele technische Anwendungen, insbesondere in der Regelungstechnik, ist die Kenntnis des Zeitverhaltens eines Steuerungssystems sehr wichtig. Zum einen muss die grundsätzliche Eignung eines Systems (z. B. Stabilität), zum anderen die Güte der erreichbaren Ergebnisse (Regelgüte) untersucht werden. Die regelungstechnischen Aspekte werden durch die Disziplin der Networked Control Systems untersucht, setzen aber eine detaillierte Kenntnis des Zeitverhaltens des Steuerungssystems voraus.

Die meisten industriellen Steuerungssysteme arbeiten heute zyklisch, daher wird der Schwerpunkt dieser Arbeit auf diese gelegt. Für die Untersuchung der Einflüsse von Kommunikationssystemen wird von kabelgebundenen Industrial-Ethernet-Systemen ausgegangen.

Um Systeme vergleichbar zu machen, werden zunächst Kenngrößen eingeführt und definiert.

In der Arbeit wird das abstrakte Modell der asynchronen Teilprozesse zur Beschreibung von verteilten industriellen Steuerungssystemen vorgestellt. Das Modell sieht eine Trennung von statischem und dynamischem Modellanteil vor. Dazu werden die beiden Modellanteile Struktur (statischer Modellanteil) und Signalpfade (dynamischer Modellanteil) erläutert. Die Vorgehensweise zur Modellerstellung und die Grenzen des Modells werden diskutiert. Anschließend wird die Ableitung des Zeitverhaltens von Signalpfaden basierend auf dem zuvor erstellten Modell beschrieben. Das Ergebnis dieser Ableitung ist eine Summe von Häufigkeitsverteilungen von Verarbeitungs- und Wartezeiten. Diese werden durch geeignete mathematische Operationen zusammengefasst. Dafür werden die prinzipielle Vorgehensweise und für die relevanten Häufigkeitsverteilungen die konkreten Lösungsschritte erläutert. Im Gegensatz zu den gezeigten Vorarbeiten wird das Zeitverhalten eines konkreten Signalpfades vollständig analytisch bestimmt.

Ausgehend von diesem Modell wird die Untergruppe der isochronen Steuerungssysteme vertiefend untersucht. Dazu wird eine spezielle Ausprägung des Modells vorgestellt, welche die detaillierte Analyse solcher Systeme ermöglicht.

Neben der Erstellung des Modells ist die Parametrierung einer konkreten Instanz eine große Herausforderung. Es werden verschiedene Wege und Ansätze zur Bestimmung der Modellparameter aufgezeigt und verglichen.

Das entwickelte Modell wird an verschiedenen Beispielen erprobt. Dazu wird das berechnete Zeitverhalten mit konkreten Messdaten verglichen. Mit der analytischen Berechnung können sehr gute Ergebnisse erzielt werden. Abschließend wird das ermittelte Zeitverhalten zur Bestimmung der Gütekriterien eines geschlossenen Regelkreises genutzt. Am Beispiel eines Portalkrans werden verschiedene Konfigurationen eines Steuerungssystems hinsichtlich ihrer Eignung für die gestellte Regelungsaufgabe verglichen. An den Beispielen wird gezeigt, dass sich das Modell sowohl für die Analyse als auch als Planungsunterstützung einsetzen lässt.

Abstract

This thesis describes a model for the analytical modelling of the temporal behaviour of distributed industrial control systems. In many technical applications, especially in closed-loop control technology, knowledge of the temporal behaviour of a control system is very important. Both the fundamental suitability of a system (e.g. its stability) and the quality of the achievable results (quality of control) need to be investigated. The closed-loop control aspects are examined in the field of networked control systems but also require detailed knowledge of the temporal behaviour of the control system.

Most industrial control systems use cyclic processes, which is why this study focuses on cyclic control systems. To determine the influence of communication systems, wired industrial Ethernet systems are assumed.

Characteristic values are initially introduced and defined in order to enable a comparison of systems.

In this study, the abstract model of asynchronous sub-processes is introduced to describe distributed industrial control systems. The model can be separated into static and dynamic parts. Both the structure (static part) and signal path (dynamic part) of the model are therefore explained. The study also discusses the model creation procedure and the limits of the model. The approach used to derive the temporal behaviour of signal paths based on the model created is then described. The result of this derivation is a sum of probability distributions of processing and waiting times. These are summarised by means of suitable mathematical operations, which involve the illustration of the basic approach and the specific solutions for the relevant probability distributions. In contrast to the preliminary studies shown, the temporal behaviour of a specific signal path is determined fully analytically.

Based on this model, the subgroup of isochronous control systems is examined in detail. A special form of the model is introduced in order to facilitate the detailed analysis of such systems.

Alongside the creation of the model, the parameterisation of a specific entity is a major challenge. Different approaches and methods for determining the model parameters are illustrated and compared.

Several different examples are used to show the developed model and the calculated temporal behaviour is compared with the specific measurement results. The study reveals that the analytical calculation can generate very good results. Finally, the temporal behaviour recorded is used to determine the quality criteria of a closed-loop control system. A gantry crane is used as an example to compare different configurations of a control system regarding their suitability for the control task set. These examples show that the model can be used for both analysis purposes and planning support.

1 Einleitung und Motivation

Technische Anlagen in der Fertigungs- und Prozessindustrie werden von Steuerungssystemen kontrolliert, die dabei sowohl Regelungs- als auch Steuerungsaufgaben übernehmen. Während es teilweise noch dedizierte Kompaktregler für einzelne Aufgaben gibt, werden Regelkreise heute fast ausschließlich über sogenannte Speicherprogrammierbare Steuerungen (SPS) geschlossen. Die Ein- und Ausgabemodule zur Ankopplung an den Prozess sind dabei meist abgesetzt in Form von Feldgeräten organisiert. Diese tauschen mit der SPS über ein Kommunikationssystem Daten aus. Im Zuge der fortschreitenden Vernetzung von Anlagen wird die klassische hierarchische Struktur, in der eine Steuerung über ihr lokales Kommunikationssystem mit ihren zugehörigen Feldgeräten kommuniziert, zunehmend durch vernetzte Strukturen ersetzt. Dabei werden Eingangsdaten von einem Feldgerät nacheinander in mehreren Steuerungen verarbeitet, bevor eine Reaktion auf den Prozess in Form der Ausgabe von Signalen über ein Feldgerät erfolgt.

Am Markt findet sich eine große Anzahl von industriellen Kommunikationssystemen. Neben den klassischen Feldbussystemen haben in den letzten Jahren die Ethernet-basierten Systeme große Akzeptanz am Markt erhalten. Gleichzeitig werden die Anforderungen an Kommunikationssysteme hinsichtlich hoher Geschwindigkeiten, kurzer Zykluszeiten und einer hohen zeitlichen Präzision immer größer. Es wird ein hoher technischer Aufwand betrieben, um diese Anforderungen zu erfüllen. So werben Hersteller mit Zykluszeiten ihrer Kommunikationssysteme im unteren Mikrosekundenbereich (EtherCAT 12,5 μs [Bec13], PROFINET IRT 31,25 μs [Sch12]). Ein schneller Feldbus reicht jedoch in der Regel nicht aus, um kurze Reaktionszeiten zu erreichen [JB13, S. 6].

Das Erreichen dieser hohen Leistungsdaten geht meist mit Einschränkungen (z. B. hinsichtlich der maximalen Anzahl von Geräten oder der Netzwerktopologie) einher. Außerdem steigert der hohe Aufwand die Preise für derartige Komponenten und Systeme. Aus den regelungstechnischen Aufgaben und deren Gütekriterien ergeben sich zeitliche Anforderungen an ein Steuerungssystem. Daher stellt sich die Frage: Welche Systemkonfiguration braucht man, um eine bestimmte Aufgabe zu lösen?

Um diese Frage zu beantworten, muss berücksichtigt werden, dass das Kommunikationssystem nur ein Teil der Verarbeitungskette ist. Die beteiligten Geräte und deren Architektur haben ebenfalls einen großen, meist sogar den überwiegenden Einfluss auf das Gesamtzeitverhalten.

Tretter zeigt in [Tre13] ein Beispiel für das Einsparpotential durch den Einsatz schnellerer Steuerungskomponenten bei einer Maschine zur Herstellung von Plastikbechern. Durch die Senkung der Zykluszeit von 2 ms (rote Linie) auf 250 μs (grüne Linie) konnte die Abweichung vom Sollwert (blaue Linie) verringert und damit die Druckregelung verbessert (siehe Abbildung 1) werden. Pro Becher können 2 g Material eingespart werden. Bei 54,6 Mio. Becher pro Jahr ergibt das eine Ersparnis von über 200 000 Dollar. Dies zeigt, dass der Einsatz von schnelleren (und damit in der Regel teureren) Komponenten in Steuerungssystemen wirtschaftlich sinnvoll sein kann.

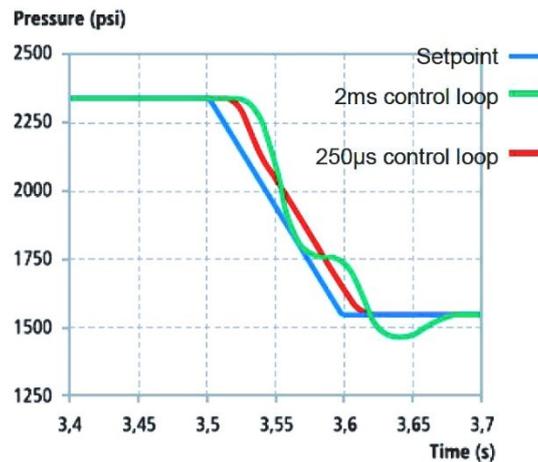


Abbildung 1: Druckverlauf beim Formen von Plastikbechern [Tre13, S. 19]

Moderne Steuerungssysteme mit komplexen Architekturen liefern im Normalfall keine konstante Antwortzeit. Die Kenntnis der Häufigkeitsverteilung dieser Antwortzeit ermöglicht die Bestimmung von Gütekriterien für Regelungsaufgaben. Durch die Ermittlung der Antwortzeit-Verteilungen für verschiedene Konfigurationen eines Steuerungssystems kann die Auswahl der passenden Lösung unterstützt werden.

Neben diesen Performance-/Leistungsanforderungen gibt es weitere Anforderungen an ein System, welche hier nicht weiter betrachtet werden sollen, aber unter Umständen eine wichtige Rolle bei der Auswahl eines geeigneten Systems spielen. So ist beispielsweise die Diagnose (siehe [DH13]) ein wesentlicher Aspekt, um Fehler in Kommunikationssystem, Steuerungssystem und Applikation schnell und gezielt finden zu können.

1.1 Ziel der Arbeit

Im Rahmen der vorliegenden Arbeit soll ein abstraktes Modell für verteilte industrielle Steuerungssysteme erstellt werden, das zur analytischen Ableitung und Analyse des Zeitverhaltens geeignet ist. Dieses Zeitverhalten soll genutzt werden, um die Eignung eines Systems für eine bestimmte regelungstechnische Aufgabe mittels Gütekriterien zu bestimmen. Dabei wird ein Ende-zu-Ende-Ansatz verfolgt.

Das Modell soll den Vergleich sowohl von verschiedenen Systemen als auch deren Konfigurationen ermöglichen. Weiterhin ist es so angelegt, dass nicht nur Aussagen für ein bestimmtes Szenario ermittelt werden können. Es soll auch bei der Analyse des Systems und dem Auffinden von Optimierungspotential unterstützen. Aufgrund des analytischen Ansatzes kann und wird das diskutierte Modell nicht den Detaillierungsgrad von simulativen Ansätzen erreichen, welche eine Vielzahl von Effekten und Einflüssen abbilden können. Die Arbeit hat jedoch gezeigt, dass für viele Szenarien die analytische Betrachtung zu den gewünschten Ergebnissen führt.

Während sich die Extremwerte des Zeitverhaltens meist recht einfach durch eine Worst-Case-Betrachtung bestimmen lassen, soll hier die Häufigkeitsverteilung ermittelt werden. „Zur Analyse von Performanz und Qualität ist es unerlässlich, die komplette Antwortzeitverteilung ermitteln zu können.“ [Gre07, S. 11]

Diese Arbeit legt den Schwerpunkt auf zyklisch arbeitende Steuerungssysteme, die auf drahtgebundener Industrial-Ethernet-Kommunikation beruhen. Damit kann der Umfang der Arbeit und die Zahl der Einflussfaktoren eingeschränkt werden. Es stellt jedoch keine grundsätzliche Einschränkung des Modells dar. Durch entsprechende Erweiterungen und Untersuchungen ist auch eine Nutzung für drahtlose Systeme denkbar.

1.2 Gliederung der Arbeit

Nach der Definition von Begriffen und einem Überblick über den aktuellen Stand der Technik werden in Kapitel 2 bestehende Ansätze für die Beschreibung und Analyse von Kommunikations- und Steuerungssystemen untersucht. Es folgen die für die späteren Betrachtungen notwendigen Grundlagen zur Modellierung von LTI-Systemen¹. Das Kapitel schließt mit der regelungstechnischen Betrachtung von Steuerungssystemen, welche in der Literatur unter dem Begriff Networked Control Systems zu finden ist. Im nächsten Kapitel werden Kenngrößen für die Bewertung von Steuerungssystemen benannt.

Im Kapitel 4 wird die Modellierung des Zeitverhaltens von Steuerungssystemen beschrieben. Ausgangspunkt ist die Betrachtung von synchronen und asynchronen Teilprozessen. Nach der Beschreibung der Modellstruktur und seiner Parameter folgen die Untersuchungen zur Ableitung des Zeitverhaltens eines Signalpfades aus einem bekannten Modell (Abschnitt 4.3). Durch diesen Schritt kann anschließend die eigentliche Berechnung des Zeitverhaltens als rein mathematisches Problem betrachtet werden. Im nächsten Kapitel wird eine Spezialisierung dieses Modells für isochrone Steuerungssysteme vorgestellt.

Zum Ableiten des Zeitverhaltens ist die Bestimmung der Parameter des Modells wesentlich. Dies wird in Kapitel 6 beschrieben. Abschließend findet eine Validierung des Modells durch Vergleichsmessungen (Kapitel 7) und eine beispielhafte Anwendung für eine Regelungsaufgabe (Kapitel 8) statt.

¹ LTI-System = lineares zeitinvariantes System (englisch linear time-invariant system)

2 Grundlagen und Stand der Wissenschaft

2.1 Begriffsdefinitionen

Zur Abgrenzung des Betrachtungsgebietes werden zunächst grundlegende Begriffe definiert.

2.1.1 System

Für ein System wird im Rahmen dieser Arbeit folgende Definition verwendet:

„Ein System besteht aus einer Menge von Elementen, welche durch Relationen miteinander verknüpft sind. Ein System hat eine Systemgrenze, die es von seiner Umwelt trennt, über die hinweg das System jedoch mit seiner Umwelt interagieren kann.“ [Had15, S. 9]

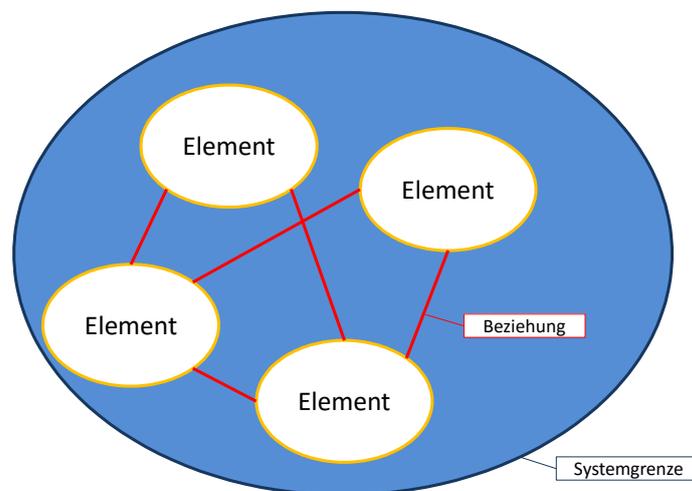


Abbildung 2: System [Had15, S. 9]

Die hier zu betrachtenden verteilten Steuerungssysteme lassen sich entsprechend dieser Definition als System bestehend aus Elementen (Steuerung, Feldgerät-IO, Kommunikationssystem, ...) betrachten, zwischen denen es Beziehungen zum Datenaustausch gibt. Neben dem Datenaustausch können weitere Beziehungen existieren, wie beispielsweise die Zugehörigkeit zu einer gemeinsamen Synchronisationsdomäne. Die Systemgrenze ist bei dieser Betrachtung die Schnittstelle zum zu steuernden Prozess bzw. der Anlage. Der Datenaustausch über die Systemgrenze erfolgt über eine physikalische Schnittstelle, über welche Signale ausgetauscht werden (z. B. analoge Spannung 0-10 V, digitales Signal 0 / 24 V).

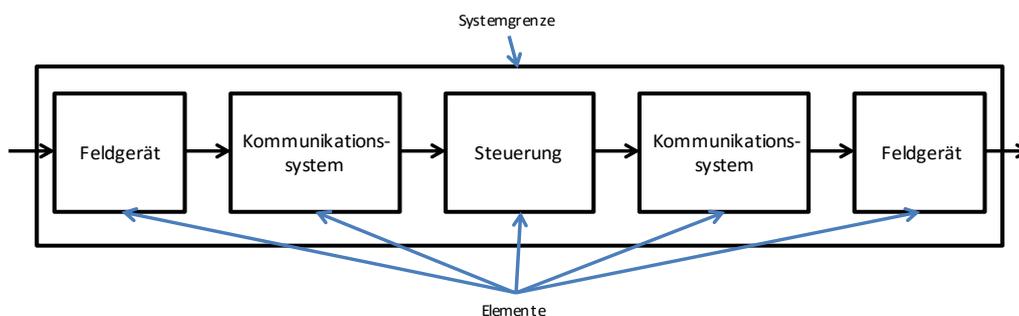


Abbildung 3: Anwendung der Definition „System“ auf ein industrielles Steuerungssystem

Jedes Element des Systems kann selbst wieder als System betrachtet werden, da dies keine monolithische Einheit darstellt, sondern seinerseits in Elemente zerlegt werden kann. Daher werden im Folgenden die Elemente des Steuerungssystems als Teilsysteme bezeichnet. Im Interesse der eingangs geforderten möglichst geringen Modellkomplexität wird versucht, auf eine feingranulare Beschreibung zu verzichten. Das bedeutet, die Teilsysteme werden möglich umfangreich gewählt, ohne jedoch das zeitliche Verhalten signifikant zu verändern.

Während in der allgemeinen Systemdefinition nach Hadlich die Beziehung zwischen den Teilsystemen ungerichtet ist, wird in Abbildung 3 bereits angedeutet, dass für die Modellierung von Steuerungs- und Kommunikationssystem die Beziehungen als gerichtet angenommen werden. Ursächlich dafür ist der Fluss von Informationen, welcher gerichtet vom Ein- zum Ausgang eines Teilsystems erfolgt.

2.1.2 Signal und Signalpfad

„Die Interaktion von Systemelementen erfolgt mittels Signalen.“ [Had15, S. 25] Dabei greift Hadlich im Speziellen eine Definition von Frey et al auf: „Unter **Signal** verstehen wir allgemein eine abstrakte Beschreibung einer veränderlichen Größe.“ [FB08, S. 1] Bezogen auf ein Steuerungssystem werden Signale überall dort im Modell sichtbar, wo Informationen zwischen Teilsystemen ausgetauscht werden müssen.

Für die Beschreibung des Zeitverhaltens eines Steuerungssystems ist der **Signalpfad** von großer Bedeutung. Ein Signalpfad beschreibt eine eindeutige Verbindung von Ein- und Ausgangsdaten und den Weg des Datums durch das Steuerungssystem. Er kann als gerichteter Graph verstanden werden, dessen Knoten die Teilsysteme und Kanten die Beziehungen zwischen den Teilsystemen sind. In [Had15] wird ein solcher Signalpfad als funktionaler Pfad bezeichnet. Der Begriff des Signalpfades wird auch in anderen Disziplinen wie Schaltungstechnik (siehe z. B. [RS15]) oder Mobilfunktechnik (siehe z. B. [Nus10]) ähnlich verwendet.

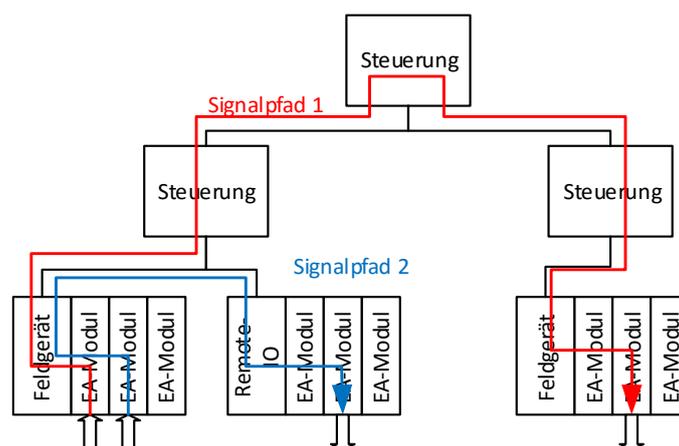


Abbildung 4: Beispielhafte Darstellung von zwei Signalpfaden durch ein Steuerungssystem

Das Zeitverhalten eines Steuerungssystems kann nicht als eine Gesamtheit beschrieben werden, sondern es muss eine separate Betrachtung für jeden Signalpfad durch das System erfolgen. In Abbildung 4 sind für ein dezentrales Steuerungssystem beispielhaft zwei Signalpfade gekennzeichnet, deren Zeitverhalten sich aufgrund verschiedener beteiligter Teilsysteme unterscheiden kann.

Während ein konkretes Steuerungssystem in der Regel durch eine Struktur beschrieben wird, kann eine Vielzahl von Signalpfaden durch das System existieren, welche beschrieben und später berechnet werden müssen.

Dieser Ansatz der Trennung wird unter anderem von Hadlich genutzt:

„Ein System hat zwei Strukturaspekte:

- *Kompositionelle Struktur (Patzak²: Aufbaustruktur) in der die Elemente durch Bestandsbeziehungen (besteht aus) verbunden sind.*
- *Funktionale Struktur (Patzak: Ablaufstruktur) in der die Elemente durch funktionale Verbindungen verbunden sind.“* [Had15, S. 10]

2.1.3 Signaltypen

Dynamische Signale können, wie in Abbildung 5 dargestellt, anhand ihres zeitlichen Verlaufes und dem Wertevorrat ihres Zustandes in zeitkontinuierliche und zeitdiskrete sowie in zustandskontinuierliche und zustandsdiskrete Signale unterschieden werden [WA05], [Syr05].

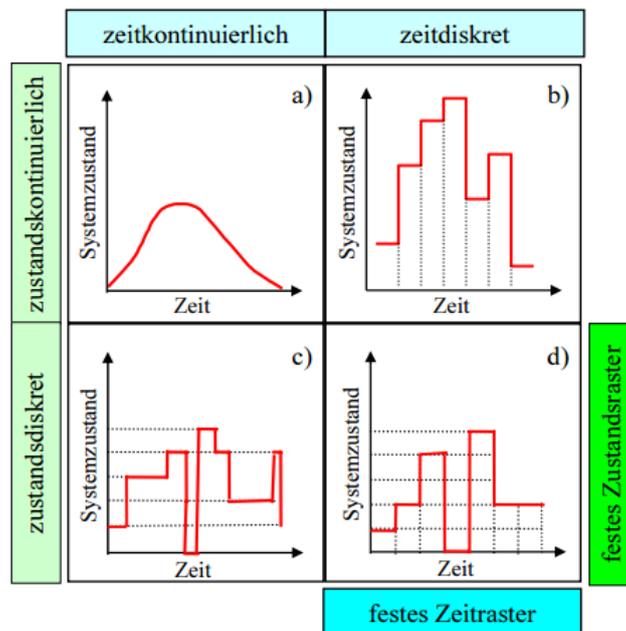


Abbildung 5: Einteilung von Signalen [Syr05, S. 36]

Bei der Ankopplung eines Steuerungssystems an den technischen Prozess (oder die Anlage) findet die Umwandlung von zeitkontinuierlichen in zeitdiskrete Signale statt. Dieser Vorgang findet in den Eingangsbaugruppen des Steuerungssystems statt und wird als Abtastung bezeichnet.

² Patzak bezieht sich auf die Veröffentlichung [Pat82]

2.1.4 Echtzeit

Die DIN 44300-9 [DIN88] definiert Echtzeit:

"Unter Echtzeit versteht man den Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind, derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind. Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu vorherbestimmten Zeitpunkten anfallen." [Sch05]

Damit sagt der Begriff Echtzeit nichts über die konkrete Geschwindigkeit eines Systems aus, d. h. wie groß oder klein die maximale Reaktionszeit ist. Es wird nur gefordert, dass es eine maximale Reaktionszeit gibt.

An Echtzeitsysteme werden verschiedene Anforderungen gestellt:

Eine Anforderung ist die **Rechtzeitigkeit**, das „heißt, die Ausgabedaten müssen rechtzeitig berechnet werden und zur Verfügung stehen. [...] Die einzuhaltenden Zeitbedingungen werden hierbei meist von einem technischen Prozess diktiert.“ [WB05, S. 318] Dies bedeutet, die Verarbeitung muss eine obere Zeitschranke einhalten.

Weiterhin wird von Echtzeitsystemen **Gleichzeitigkeit** gefordert. „Dies bedeutet, die Rechtzeitigkeit muss für mehrere Aktionen gleichzeitig gewährleistet sein.“ [WB05, S. 323] Daraus resultiert die Anforderung, dass der zeitliche Versatz zwischen dem Erfassen (bzw. der Ausgabe) von mehreren Signalen begrenzt sein muss.

Die dritte Anforderung ist die **Verfügbarkeit**, das bedeutet, sie „müssen unterbrechungsfrei betriebsbereit sein, da es sonst zu einer Verletzung der Zeitbedingung kommen kann.“ [WB05, S. 324]

Um die Bedingung der Rechtzeitigkeit einzuhalten ist eine zeitliche Vorhersagbarkeit notwendig. „Zeitliche Vorhersagbarkeit ist erforderlich, um die Einhaltung dieser Zeitbedingungen in jedem Fall zu garantieren.“ [WB05, S. 320]

Abhängig von der Strenge der einzuhaltenden Zeitbedingungen wird zwischen 3 Arten unterschieden:

- **Harte Echtzeitbedingungen**
„Hier müssen die Zeitbedingungen auf jedem Fall eingehalten werden, anderen Falls droht Schaden.“ [WB05, S. 321]
- **Feste Echtzeitbedingungen**
„Bei festen Echtzeitbedingungen wird die durchgeführte Aktion nach Überschreiten der Zeitbedingung wertlos und kann abgebrochen werden. Es droht jedoch kein unmittelbarer Schaden.“ [WB05, S. 321]
- **Weiche Echtzeitbedingungen**
„Hier sind die Zeitbedingungen als Richtlinien anzusehen, die durchaus in gewissem Rahmen überschritten werden dürfen.“ [WB05, S. 321]

Industrielle Steuerungssysteme können allgemein als Echtzeitsysteme betrachtet werden, die konkreten zeitlichen Anforderungen hängen von der Anwendung ab.

2.1.5 Determinismus

Determinismus ist in der Steuerungstechnik die „Eigenschaft von Systemen, die in ihrem Verhalten bzw. ihrer Reaktionszeit reproduzierbar bestimmbar sind.“ [WEK15]

In der Philosophie wird eine abstraktere Definition genutzt: „Ein abgeschlossenes System S ist dann bezüglich K als deterministisch zu bezeichnen, wenn aus einem für einen Zeitpunkt bezüglich K gegebenen Zustand ein späterer Zustand bezüglich K auf Grund der für S geltenden Gesetze deterministisch voraussagbar ist.“ [Ste83, S. 560f.]

Im Rahmen dieser Arbeit wird daraus der Begriff des **deterministischen Zeitverhaltens** abgeleitet. Dieser sagt aus, dass sich das Zeitverhalten eines Systems anhand der Modellbeschreibung und der bekannten Randbedingungen hinreichend genau beschreiben lässt.

2.1.6 Isochronität und Gleichzeitigkeit

In der Literatur finden sich im Zusammenhang mit synchronen und isochronen Steuerungssystemen diverse Begriffe, welche teilweise synonym verwendet werden. Zum besseren Verständnis sollen diese hier definiert und abgegrenzt werden.

Die Begriffe **Isochronität** oder **isochrone Kommunikation** werden oft im Zusammenhang mit Steuerungssystemen genutzt. Padeffke definiert: „Ein Signal ist isochron, wenn die durchschnittliche nominale Frequenz konstant ist.“ [Pad05, S. 9] Wikipedia erklärt Isochronität als „ein Signal mit konstanter Periodendauer“ [Wik12]. Nach [Sch06, S. 339] bedeutet „die Übertragungsart ‚isochron‘ [...], dass ein Synchronisationssignal ganze Datenblöcke (von z. B. 125 μ s) taktet“.

In realen Systemen kann die völlige Konstanz nicht garantiert werden. Hier bedeutet Isochronität, „dass die Zykluszeit (Zeitspanne zwischen zwei aufeinanderfolgenden Übertragungen) innerhalb eines vorgegebenen Toleranzfeldes konstant ist“ [HDD+13].

Das Toleranzfeld stellt den Jitter der Isochronität dar. Es muss beachtet werden, dass Isochronität eine Black-Box-Eigenschaft ist, d. h. es wird dadurch keine Aussage über die interne Funktionalität des Steuerungssystems gemacht. Lediglich die zeitliche Äquidistanz zwischen zwei Übertragungen wird definiert (Abbildung 6). Um Isochronität am Ein- und Ausgang eines Systems zu garantieren, ist eine konstante Verarbeitungszeit (im Rahmen des Toleranzfeldes) erforderlich.

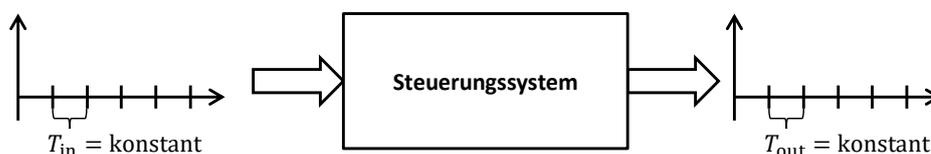


Abbildung 6: Prinzip der Isochronität nach [HDD+13]

Unter **Gleichzeitigkeit** versteht man das Nichtvorhandensein einer zeitlichen Abweichung zwischen der Ein- oder Ausgabe von verschiedenen Signalen eines Steuerungssystems. Wie bei der Isochronität kann auch Gleichzeitigkeit bei realen Systemen nur innerhalb eines Toleranzfeldes garantiert werden. Gleichzeitigkeit kann erforderlich sein, wenn beispielsweise mehrere sich zeitlich ändernde Größen zur Bestimmung des Systemzustandes verrechnet werden sollen oder verschiedene Aktoren zeitgleich geschaltet werden (siehe Abbildung 7). Gleichzeitigkeit geht oft mit Isochronität einher. Beide Eigenschaften sind jedoch unabhängig voneinander.

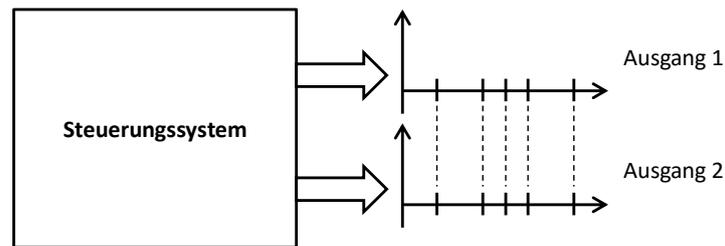


Abbildung 7: Prinzip der Gleichzeitigkeit von Ausgangssignalen

Während Isochronität und Gleichzeitigkeit Aussagen über von außen beobachtbares zeitliches Verhalten eines Systems treffen, beschreibt die **synchrone Abarbeitung des Steuerungsprogramms** eine interne Funktionsweise. Sie bedeutet, dass die Ausführung des Steuerungsprogramms starr an den Zyklus des Kommunikationssystems gekoppelt ist (siehe auch [HD12]). Die Abarbeitung kann dabei entweder direkt vom Kommunikationssystem (z. B. durch ein Ereignis, das den Empfang aller Telegramme eines Zyklus signalisiert) oder durch einen eigenen Zeitgeber angestoßen werden, der auf den des Kommunikationssystems synchronisiert ist. Im Sinne des später eingeführten Modells arbeiten die Zyklen von Kommunikationssystem und Steuerung synchron. Es treten keine stochastischen Wartezeiten bei der Übergabe auf (Abbildung 8).

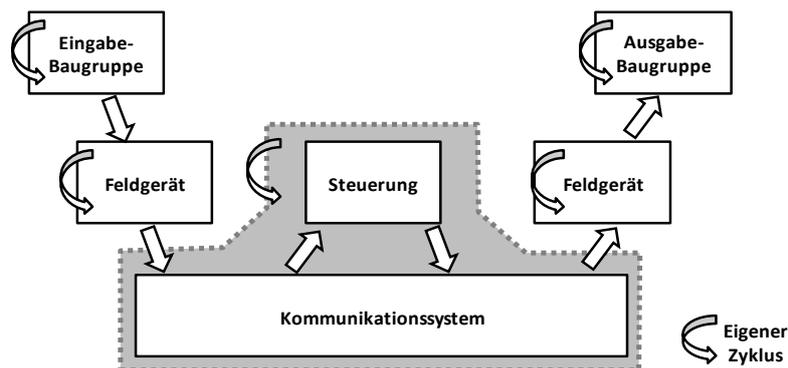


Abbildung 8: Prinzip der synchronen Abarbeitung des Steuerungsprogramms

Eine weitere Steigerung synchronen Abarbeitung und damit des Determinismus stellt die sogenannte **Taktsynchronität** dar (siehe Abbildung 9). Hierbei sind sämtliche Baugruppen des Steuerungssystems, welche an der Signalverarbeitung beteiligt sind, auf eine gemeinsame Zeitbasis synchronisiert (Steuerungsprogramm, Kommunikation, Ein- und Ausgabemodule, ...). Dadurch treten keine stochastischen Wartezeiten (im Rahmen der Synchronisationsgenauigkeit) innerhalb des Systems auf. Durch die Einführung von geplanten Wartezeiten zum Ausgleich von unterschiedlichen Verarbeitungszeiten können mehrere (oder alle) Einlese- und/oder Ausgabezeitpunkte eines Systems synchronisiert werden. Somit kann z. B. bei einer Gleichlaufregelung das Einlesen der Winkel der verschiedenen Achsen exakt gleichzeitig erfolgen.

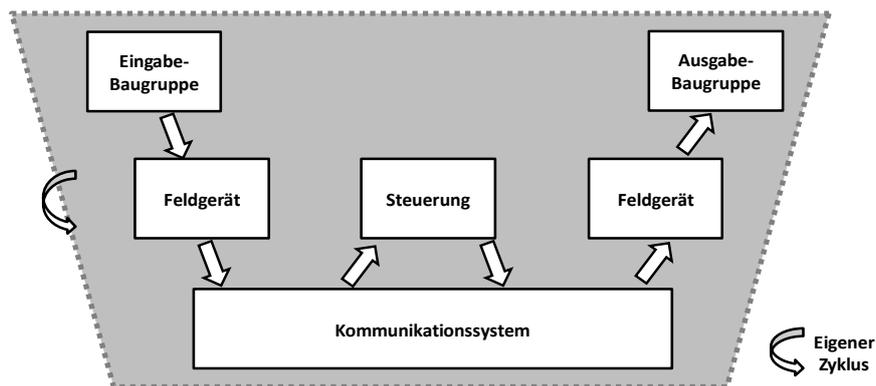


Abbildung 9: Prinzip der Taktsynchronität

Die Begriffe *synchrones Steuerungssystem* und *isochrones Steuerungssystem* werden synonym für ein Steuerungssystem mit Taktsynchronität verwendet. Das heißt, die Klemme-Klemme-Reaktionszeit (das Ende-zu-Ende-Zeitverhalten) ist konstant und Ein- und Ausgänge sind isochron.

2.2 Industrielle Steuerungssysteme

Industrielle Steuerungssysteme können nach verschiedenen Eigenschaften klassifiziert werden. Im Folgenden erfolgt eine Klassifizierung nach strukturellen und Verhaltenseigenschaften.

2.2.1 Klassifikation nach Architektur

Der Ursprung der heutigen industriellen Steuerungssysteme liegt in den festverdrahteten/verbindungsprogrammierten Steuerungen. Der Errichtungs- und Wartungsaufwand dieser Anlagen war sehr hoch, da bereits ein Verdrahtungsfehler oder Drahtbruch zum Ausfall oder Fehlverhalten der gesamten Steuerung führen konnte.

Im Jahre 1969 wurde mit der Modicon 084 die erste digitale, speicherprogrammierbare Steuerung (SPS) vorgestellt. Dieser Schritt wird von Drath auch als „dritte industrielle Revolution“ bezeichnet, da sie „das Fundament der gesamten heutigen Automatisierungspyramide und moderner Prozessleitsysteme“ [Dra14] ist.

Nach IEC61131 ist eine SPS „ein digital arbeitendes elektronisches System für den Einsatz in industriellen Umgebungen mit einem programmierbaren Speicher zur internen Speicherung der anwenderorientierten Steuerungsanweisungen zur Implementierung spezifischer Funktionen, wie z. B. Verknüpfungssteuerung, Ablaufsteuerung, Zeit-, Zählfunktion und arithmetische Funktionen, um durch digitale und analoge Eingangs- und Ausgangssignale verschiedene Arten von Maschinen und Prozessen zu steuern.“ [DIN03]

Steuerungssysteme lassen sich hinsichtlich ihres Aufbaus und ihrer Architektur in verschiedene Kategorien einteilen:

Kompakte oder Zentrale Steuerungen bilden die älteste Form der SPS. In einem Gerät sind Steuerungslogik und Ein- und Ausgangsmodule vereint. Sämtliche Sensoren und Aktoren einer Anlage müssen an die zentrale SPS angeschlossen werden, wodurch ein sehr hoher Verkabelungsaufwand entsteht.

Daher wurden die **dezentralen Steuerungen** entwickelt. Die Steuerungslogik (CPU) und Ein-Ausgabe-Baugruppen (sogenannte Feldgeräte oder Remote-IOs) sind hierbei getrennt. Die Kommunikation zwischen den Geräten erfolgt über industrielle Kommunikationssysteme. Diese Architektur erlaubt die Reduktion des Verdrahtungsaufwandes, da die Feldgeräte nahe an den Sensoren und Aktoren der Anlage angebracht werden können und nur noch über ein Buskabel mit der Steuerung verbunden werden müssen.

Durch weitere Dezentralisierung entstanden die **verteilten Steuerungen**. Hierbei sind nicht nur die Ein- und Ausgabefunktionen räumlich in der Anlage verteilt, auch Steuerungsfunktionen selbst werden „im Feld“ gelöst. Dadurch entstehen hierarchische Strukturen, welche von den Ein- und Ausgabebaugruppen über lokale Steuerungen bis hin zu den übergeordneten Steuerungen reichen. Dieser Architekturen orientieren sich häufig an der Anlagenstruktur.

Den zurzeit letzten Schritt in der Evolution der Steuerungsarchitekturen stellen die **kooperativ verteilten Steuerungen** dar. Auch hier sind mehrere Steuerungen in der Anlage verteilt, es gibt jedoch keinen zentralen Koordinator. Die einzelnen Teile kommunizieren direkt miteinander und steuern die Gesamtanlage.

Der zunehmende Grad an Dezentralisierung und Verteilung von Steuerungsfunktionen über die Anlage geht einher mit einer steigenden Anzahl von asynchron arbeitenden Geräten und Prozessen, wodurch ohne zusätzliche Maßnahmen der Determinismus des Zeitverhaltens abnimmt. In aktuellen Kommunikations- und Steuerungssystemen wurden daher Mechanismen integriert, welche mehr Determinismus durch Synchronisation erreichen sollen. Dies sind beispielweise die geplante, isochrone Kommunikation nach PROFINET IRT oder der Mechanismus der Taktsynchronität, bei dem sämtliche Ein- und Ausgangsbaugruppen sowie die Steuerung mit dem gleichen Takt arbeiten (siehe z. B. [SJ11]) Neben der Synchronisation wird auch Oversampling eingesetzt, d. h. einzelne Teilprozesse arbeiten zwar asynchron, aber mit kürzer Zykluszeit (siehe Abschnitt 4.3.2).

2.2.2 Klassifikation nach Verhalten

Entsprechend ihres zeitlichen Verhaltens können Steuerungssysteme in zwei Kategorien eingeteilt werden:

- Zeitgetriebene Systeme
- Ereignisgetriebene Systeme

Zeitgetriebene Systeme führen ihre Aufgaben zyklisch, d. h. in einem festen zeitlichen Raster³ aus, weshalb sie auch als **zyklische Systeme** bezeichnet werden. Demgegenüber arbeiten **ereignisgetriebene Systeme** reaktiv auf Ereignisse.

Diese Gliederung gilt allgemein für Echtzeitsysteme, Kopetz beschreibt sie in [Kop91] als zwei verschiedene Design-Philosophien für den Entwurf von Echtzeitsystemen.

In [Kop91], [Kop02] und [SS06] werden die Vor- und Nachteile von beiden Ansätzen diskutiert. Bei zyklischen Systemen wird das Zeitraster entsprechend der Anforderungen des Prozesses gewählt. Bei ereignisgetriebenen Systemen ist dies nicht notwendig, da die zeitliche Organisation automatisch

³ Der Begriff des festen zeitlichen Rasters stellt an dieser Stelle kein Merkmal hinsichtlich Determinismus oder Synchronität dar. Er wird in der Literatur zur Abgrenzung der beiden Systemklassen genutzt.

erfolgt. Dafür können durch vielfältige Änderungen (z. B. in Folge eines Anlagenfehlers) sogenannte Ereignisschwälle (event showers) auftreten, d. h. es wird eine große Menge von Ereignissen im Steuerungssystem generiert. Das System muss so ausgelegt sein, dass auch im Worst-Case-Fall sämtliche Ereignisse rechtzeitig bearbeitet werden können. Das muss durch intensive Tests im Vorfeld sichergestellt werden.

Neben den reinen zyklischen und ereignisgetriebenen Systemen sind diverse Mischformen möglich. Beispielsweise arbeitet die Kommunikation zyklisch, das Steuerungsprogramm wird jedoch ereignisgetrieben aufgerufen. Greifeneder und Frey diskutieren in [GF08] mögliche Formen und deren Einfluss auf die Reaktionszeit eines Systems.

Die überwiegende Mehrheit der Steuerungssysteme arbeitet heute zyklisch. Sie basieren auf dem Standard IEC 61131 [DIN03], welcher eine zyklische Arbeitsweise definiert. Die Steuerungen führen ihr Programm in einem festen zeitlichen Raster aus, gleiches gilt für die Übertragung der Kommunikationssysteme und die Datenein- und -ausgabe. Dies ist auch dem zyklischen Ansatz der klassischen Regelungstechnik geschuldet.

2.2.3 Weitere Klassifikationen

Neben den oben genannten Architekturen und Eigenschaften haben sich weiteren Klassifikationen etabliert.

Netzbasierende Automatisierungssysteme (NAS) sind nach [Gre07] Systeme bestehend aus mehreren SPSen, diversen Sensoren und Aktoren sowie einem Netzwerk, welches alle diese Komponenten miteinander verbindet, siehe Abbildung 10. „Das Besondere hieran ist, dass nicht nur das Netz von einer Anzahl unabhängiger Nutzer verwendet wird, sondern jede der SPSen von jedem Sensor Signale empfangen, an jede andere SPS Informationen senden und – zumindest theoretisch – an jedes Stellglied Anweisungen übermitteln kann.“ [Gre07, S. 1]

NAS können nach Ihrer Architektur den (kooperativ) verteilten Steuerungssystemen zugeordnet werden. Hinsichtlich des zeitlichen Verhaltens der Komponenten (zyklisch oder ereignisbasiert) wird keine Einschränkung getroffen.

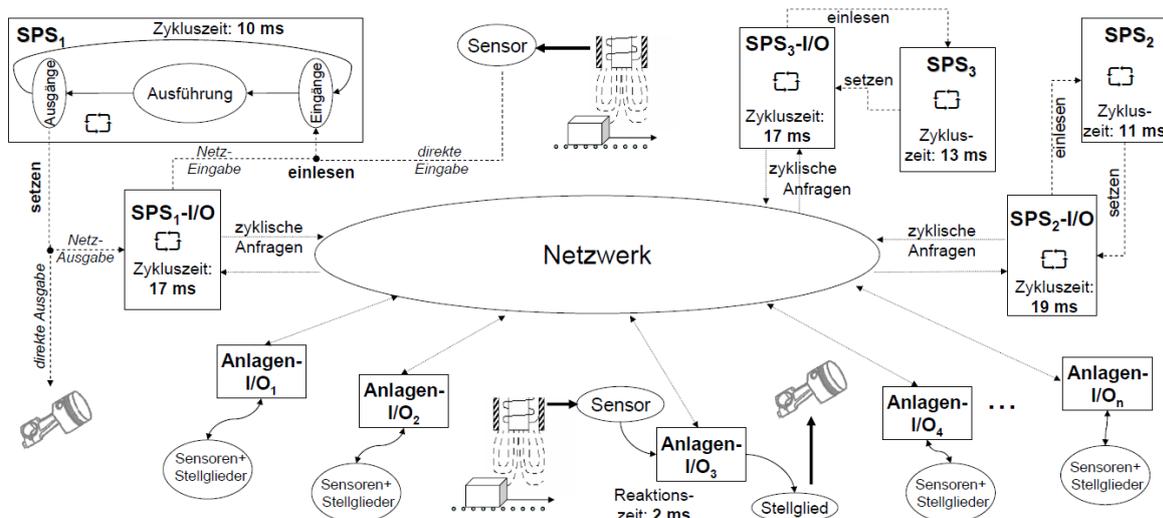


Abbildung 10: Beispiel eines netzwerkbasierten Automatisierungssystems (NAS) [Gre07]

„Moderne technische Systeme, die verschiedene Technologien zum Rechnen (**C**omputation) und zur Kommunikation (**C**ommunication) nutzen, um eine gegebene Regelungsaufgabe (**C**ontrol) zu erledigen, werden als **C³-Systeme** bezeichnet.“ [Liu14, Abstract] Liu verwendet den Begriff C³-System als Überbegriff für verschiedene, konkretere Beschreibungen wie bspw. NCS (Networked Control System), NAS oder DCS (Distributed Control Systems). Greifeneder bezeichnet C³-Systeme als „derzeit eine der wichtigsten Herausforderungen der Automatisierungstechnik.“ [Gre07, S. 2] Sie umfassen dabei verschiedene Teildisziplinen wie Regelungstechnik, Kommunikation, Signalverarbeitung und Steuerungstechnik.

Der Begriff der NCS stellt keine Klassifikation der Steuerungssysteme dar, sondern beschreibt eine Disziplin, die sich mit den Auswirkungen von Steuerungs- und Kommunikationssystemen auf den geschlossenen Regelkreis beschäftigt. Dies wird in Abschnitt 2.6.4 beschrieben.

2.3 Industrielle Kommunikationssysteme

Wie in den Zielen formuliert (siehe Abschnitt 1.1), fokussiert diese Arbeit auf kabelgebundene Industrial-Ethernet-Übertragungsverfahren. Während die ersten Versuche zur Einführung von Ethernet in der Automatisierungstechnik in den 80er-Jahren [And06, SS85] erfolglos blieben, ist in den letzten Jahren ein deutlicher Trend zur Nutzung von Ethernet-basierten Systemen zu erkennen. Diese bieten im Vergleich zu klassischen Feldbussen⁴ zahlreiche Vorteile, so sind die Übertragungsraten von 100 MBit/s oder 1 GBit/s deutlich höher als bspw. bei RS485-basierten Systemen (z. B. PROFIBUS) mit 12 MBit/s [Rei08, S. 63] oder CAN mit 1 MBit/s [Rei08, S. 85]. Neben diesen Leistungsdaten steht vor allem die leichtere Integration von Industrial Ethernet in bestehende IT-Infrastrukturen im Vordergrund. Viele Ethernet-basierte Kommunikationssysteme ermöglichen eine parallele Übertragung von Standard-IT-Daten über das Steuerungsnetzwerk, so dass z. B. eine IP-Kamera zur Anlagenüberwachung ohne zusätzlichen Verkabelungsaufwand betrieben werden kann. In Zeiten voranschreitender vertikaler Integration ist dies ein wichtiger Punkt. Jasperneite nennt folgende Gründe für den Einsatz von Ethernet in der Automatisierungstechnik:

- „Ethernet ist der Standard für die Vernetzung in der Bürokommunikation. Im Sinne der Vereinheitlichung soll diese Infrastruktur nun auch für die Vernetzung der Produktionssysteme bis in die Feldebene genutzt werden.
- Ethernet-basierte Produkte für den Einsatz im Büroumfeld werden in großer Stückzahl produziert, was zu entsprechend günstigen Preisen führt. Außerdem findet bei Ethernet eine kontinuierliche Weiterentwicklung statt.
- Ethernet ist Allgemeingut und wird von keinem Hersteller dominiert.“ [Jas05]

Ein weiterer entscheidender Vorteil von Ethernet liegt in der Möglichkeit, verschiedene Protokolle über das gleiche physikalische Medium und damit die gleichen Kabel zu betreiben. Dies ermöglicht die Nutzung von spezialisierten Protokollen für Teilaufgaben, ohne zusätzliche Installationskosten für die Verkabelung zu erzeugen. Die Möglichkeit der Nutzung verschiedener Kommunikationsprotokolle erhöht auch die Investitionssicherheit solcher Lösungen, da zukünftige (Ethernet-basierte) Protokolle auf der gleichen physikalischen Installation genutzt werden können.

⁴ „Unter Feldbus versteht man ein Bussystem, das in rauer Umgebung (Feld) eingesetzt wird. Neben besonderen Anforderungen an die mechanische Ausführung sind insbesondere auch Anforderungen an die Robustheit (Störempfindlichkeit) des Datenprotokolls gegeben.“ [Gut14, S. 706]

Aufgrund dieser Vorteile und der großen Akzeptanz Ethernet-basierter Systeme liegt der Schwerpunkt dieser Arbeit auf Steuerungssystemen, welche auf kabelgebundener Industrial-Ethernet-Kommunikation basieren. Diese Einschränkung gilt hauptsächlich für die gewählten Begriffe und Beispiele. Eine Übertragbarkeit der Ergebnisse auf Nicht-Ethernet-Systeme ist damit nicht ausgeschlossen, sie wird jedoch nicht detaillierter untersucht.

Trotz der Einschränkung auf Industrial-Ethernet-Systeme ist die Zahl der am Markt verfügbaren Systeme sehr groß⁵, weshalb im Folgenden eine Einteilung dieser Systeme nach verschiedenen Aspekten durchgeführt wird.

2.3.1 Klassifikation nach Zeitverhalten

Ethernet ist nach IEEE 802.3 [IEE12] definiert und bietet ein hohes Maß an Flexibilität, welche von Office-Netzwerken bis Wide Area Network (WAN) wie dem Internet in der IT-Welt reichen. Im industriellen Umfeld reichen die Einsatzfelder von Regelungs- und Steuerungsaufgaben im Feldbereich bis auf die MES-Ebene. Ethernet bietet jedoch kein deterministisches Zeitverhalten, bedingt durch das verwendete Buszugriffsverfahren CSMA/CD⁶ und die heute übliche Architektur als Switched Ethernet. Ein deterministisches Zeitverhalten ist jedoch eine wesentliche Voraussetzung für Regelungsaufgaben, weshalb dieses durch zusätzliche Maßnahmen erreicht werden muss. Nach [Neu04] können Ethernet-basierte Kommunikationssysteme anhand der erfüllten zeitlichen Anforderungen in 3 Klassen eingeteilt werden:

1. Weiche Echtzeit (Soft real-time)
2. Harte Echtzeit (Hard real-time)
3. Isochrone Echtzeit (Isochronous real-time)

Diese Einteilung entspricht der Einteilung von Echtzeit in 3 Arten anhand der Strenge der Zeitbedingung (siehe Abschnitt 2.1.4).

Während die meisten am Markt verfügbaren Ethernet-Protokolle Verhalten nach Klasse 2 anbieten, müssen spezielle Protokolle entwickelt werden, um isochrones Verhalten über eine Standard-Ethernet-Physik zu realisieren (z. B. PROFINET IRT, EtherCAT, Ethernet POWERLINK).

Eine ähnliche Einteilung nimmt Jasperneite in [Jas05] vor. Er definiert 3 Dienstgüte-Klassen (Quality-of-Service, QoS) und quantifiziert dabei die Echtzeitanforderungen an Netzwerktechnik in der Fertigungstechnik hinsichtlich Latenzzeit und Jitter, siehe Tabelle 1.

Tabelle 1: Klassifizierung der Echtzeitanforderungen nach [Jas05]

QoS-Klasse	Anwendung	QoS-Anforderung	
		Latenzzeit	Jitter
1	Steuerung-zu-Steuerung, Visualisierung	10-100 ms	- ⁷
2	Steuerung-zu-dezentraler Peripherie	1-10 ms	> 1 ms
3	Synchronisierte Bewegungsabläufe	< 1 ms	< 1 µs

⁵ [Sch13] listet 31 Lösungen für Real-Time-Ethernet (Stand Oktober 2013)

⁶ CSMA/CD = Carrier Sense Multiple Access/Collision Detection [IEE12, S. 62ff.]

⁷ „-“ bedeutet an dieser Stelle, dass keine Aussage zum Jitter gemacht wird

Anders nimmt in [And06] eine Einordnung des Implementierungsaufwandes für echtzeitfähige Ethernet-Systeme hinsichtlich Hardware, Software und Protokoll abhängig von den Anforderungen an die präzise Einhaltung des Zeitmanagements vor. Dabei werden Vorarbeiten der IAONA⁸ [IAO07] genutzt. Abbildung 11 stellt dar, wie mit zunehmenden Anforderungen an die zeitliche Präzision der Anteil von Standard-Komponenten in einem Industrial-Ethernet-System zugunsten von speziellen Lösungen abnimmt. Dieser höhere Anteil von speziellen Hard- und Software-Lösungen führt dazu, dass mit steigenden zeitlichen Anforderungen auch der Entwicklungsaufwand und damit die Kosten für ein System größer werden.

Klasse	1	2	3	4
Hardware	Standard	Standard	Standard	Speziell
Software	Standard	Speziell	Speziell	Speziell
Protokoll	Standard	Standard	Speziell	Speziell
Jitter	> 1 ms	100 µs - 3 ms	10 µs - 400 µs	0,5 µs - 15 µs

Abbildung 11: Klassifizierung des Entwicklungsaufwandes bei Industrial Ethernet [And06, S. 32]⁹

Nach [And06] sind die Klassen 1 und 2 mit den Leistungsparametern von klassischen Feldbussystemen (PROFIBUS, Interbus, CAN) vergleichbar und bei Industrial Ethernet „mit vergleichsweise vertretbarem Aufwand“ realisierbar. Anders geht davon aus, dass Jitter im Bereich von 10 µs bis 400 µs nur durch den Einsatz spezieller Protokolle erreicht werden können (Klasse 3). Für noch kleinere Jitter muss auch die Hardware angepasst werden (Klasse 4).

2.3.2 Klassifikation nach Architektur

Neben der Einteilung der Systeme anhand ihres Zeitverhaltens kann auch die verwendete Architektur als Kriterium genutzt werden. Die grundlegende Architektur von Industrial-Ethernet-Systemen basiert auf dem ISO/OSI-Referenzmodell [ISO98] (siehe Abbildung 12).

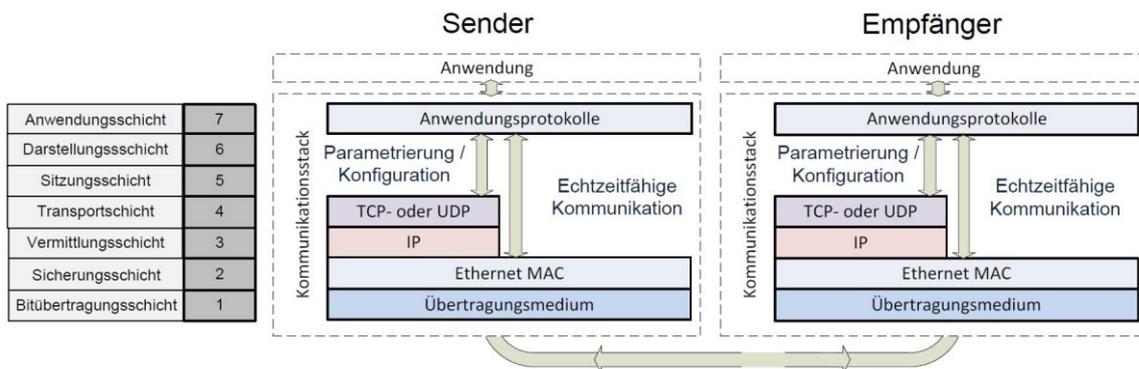


Abbildung 12: ISO/OSI-Referenzmodell bei Anwendung im Industrial Ethernet [Run14, S. 19]

⁸ IAONA = Industrial Automation Open Networking Alliance

⁹ Anders verweist bei diesem Bild auf die Webseite der IAONA [IAO07], jedoch konnte es dort nicht gefunden werden, daher wird [And06, S. 32] als Quellenangabe genutzt.

In der Automatisierungstechnik hat sich die Einteilung in 3 Klassen etabliert, welche mit verschiedenen individuellen Anpassungen z. B. in [Jas05] (siehe Abbildung 13) und [And06] (siehe Abbildung 14) beschrieben wird. In beiden ist der Übergang von Variante 1 bis Variante 3 durch abnehmenden Jitter der Kommunikation ausgezeichnet, analog zu den Einstufungen im vorigen Abschnitt.

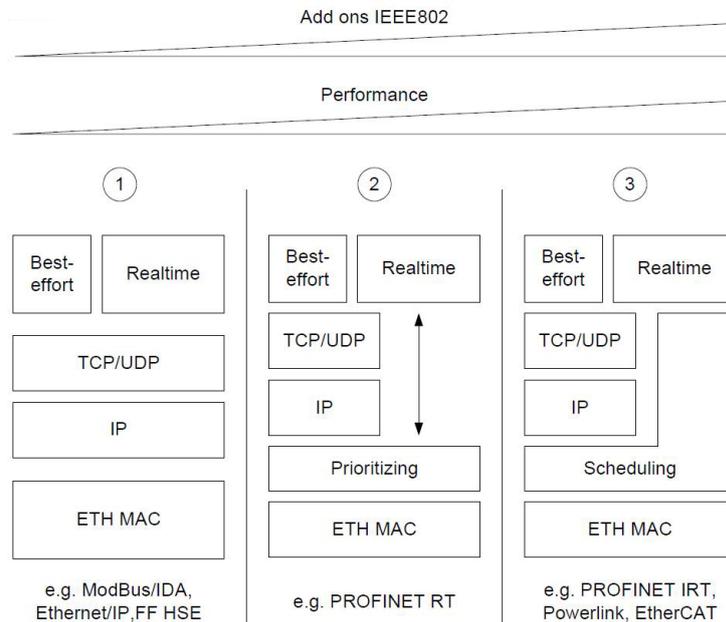


Abbildung 13: Einteilung der Herstellerkonzepte für Echtzeit-Ethernet nach [JSW07, S. 17]¹⁰

Abbildung 13 zeigt, dass steigende Performance (d. h. kürzere Zykluszeiten und deterministischeres Zeitverhalten) nur durch Erweiterungen oder Anpassungen des Ethernet-Standards IEEE802 erreicht werden können. Während für die erste Gruppe die Standard-Architektur nach ISO/OSI-Referenzmodell zur Datenübertragung von Echtzeitdaten genutzt wird, werden bei Gruppe 2 die Transport- und Vermittlungsschicht umgangen und eine Priorisierung eingeführt. Für die leistungsfähigste Gruppe 3 wird ein zusätzliches Scheduling eingesetzt, um die Echtzeitdaten entsprechend der hohen Zeitanforderungen zu übertragen.

¹⁰ Im originalen Bild war der Pfeil unter „Add ons IEEE802“ verdreht, dies wurde hier korrigiert.

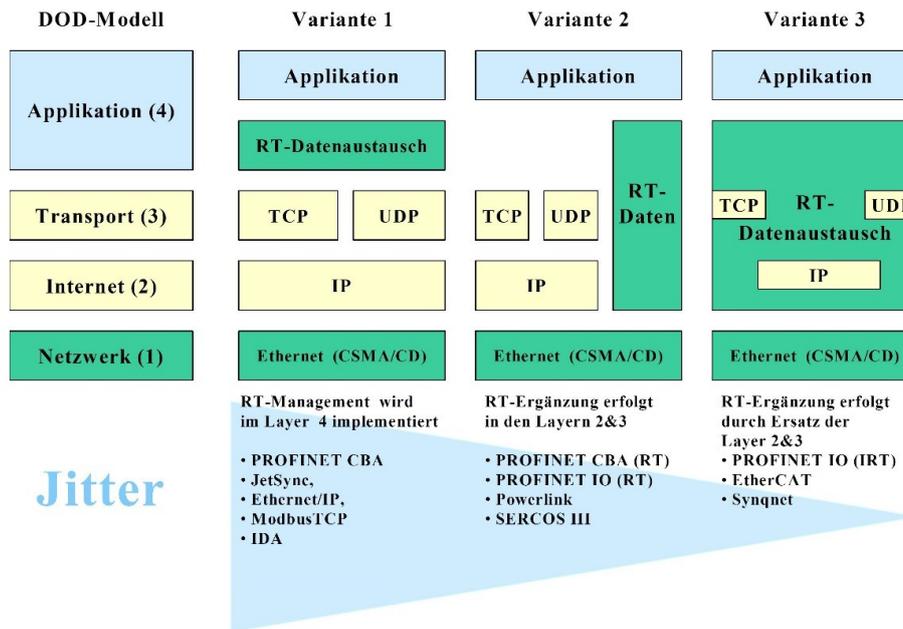


Abbildung 14: Gegenüberstellung von Architekturkonzepten für Industrial Ethernet [And06, S. 42]

Auch in Abbildung 14 werden in Variante 1 die bekannten Ethernet-Protokolle TCP, UDP und IP für die Übertragung der Steuerungsdaten genutzt. Variante 2 setzt auf den Parallelbetrieb von Standard-Kanal (TCP/UDP/IP) und einem Echtzeitkanal, welcher direkt auf das Ethernet zugreifen kann, unterstützt durch eine Priorisierung seiner Daten nach IEEE 802.1Q [IEE03]. Dadurch kann das Echtzeitverhalten für die Steuerungsdaten verbessert werden. In Variante 3 übernehmen die Echtzeiterweiterungen die komplette Kontrolle über das Stack-Verhalten, die Standard-Protokolle TCP/UDP/IP werden nur noch untergeordnet übertragen. Dadurch entsprechen diese Systeme nicht mehr dem Ethernet-Standard und die Kompatibilität ist eingeschränkt.

2.3.3 Zuordnung von Zeitverhalten zu Architekturen

In Tabelle 2 ordnet Anders den Klassifikationen nach Echtzeitklasse und Architektur typische Vertreter zu. Anhand seiner Einordnung kann man klar erkennen, dass eine höhere zeitliche Präzision mit einem erhöhten Aufwand bei der Geräteentwicklung (Kosten für Hard- und Software), aber auch bei der Integration dieser in die Gesamtanlage (durch abnehmende Kompatibilität zum Ethernet-Standard) einhergeht. Dabei werden die Echtzeitklassen aus Abbildung 11 den Architekturkonzepten aus Abbildung 14 gegenübergestellt.

Tabelle 2: Klassifizierung ausgewählter Verfahren nach der IAONA [And06, S. 43]¹¹

Echtzeitklasse	Architekturkonzept	Typische Vertreter
1	1	Ethernet/IP, Modbus/TCP, IDA
2	1	Ethernet/IP, Modbus/TCP
3	2	Powerlink, PROFINET V2
4	3	PROFINET V3, EtherCAT, SERCOS-III

¹¹ Der Autor verwendet in der Tabelle für PROFINET die Bezeichnungen „V2“ und „V3“. Nach heutiger Nomenklatur gilt: „PROFINET V2“ = „PROFINET RT“; „PROFINET V3“ = „PROFINET IRT“

Die Veränderungen des Ethernet-Standards können bis zu einer Änderung des Buszugriffverfahrens / Scheduling vom gleichberechtigten Zugriff per CSMA/CD auf ein Zeitschlitzverfahren, wie bei PROFINET IRT, gehen oder die Umstellung auf eine Token-Ring-artige Übertragung, wie bei EtherCAT.

2.3.4 Klassifikation nach minimaler Zykluszeit

Robert et al. bewerten in [RGR+12] verschiedene Industrial-Ethernet-Systeme anhand der minimal erreichbaren Zykluszeit. Dazu werden sowohl der Aufbau der Telegramme (Overhead) als auch die genutzten Sendemechanismen untersucht. Die minimale Zykluszeit ist dabei definiert als:

„The minimum cycle time was the communication time required by the controller to both collect and update the data memories of all sensors and actuators.“ [RGR+12, S. 744]

Der Vergleich berücksichtigt EtherCAT, PROFINET IRT, Modbus/TCP und Ethernet/IP, jeweils mit Fast Ethernet (100 MBit/s) als auch Gigabit-Ethernet. Abbildung 15 zeigt die minimale Zykluszeit als Funktion der Geräteanzahl, bei einer konstanten Payload von 100 Byte pro Gerät und den Bitraten 100 Mbit/s und 1 Gbit/s. Es wird gezeigt, dass von den vier Systemen Modbus/TCP die schlechteste Performance (d. h. die größte minimale Zykluszeit) erreicht. Bei Fast-Ethernet erreicht EtherCAT die beste Performance für wenige Geräte. Bei mehr Geräten ist PROFINET IRT schneller. Ethernet/IP liegt im Vergleich auf Platz 3. Ethernet/IP und PROFINET IRT profitieren stärker von der höheren Bandbreite bei Gigabit-Ethernet. Hier erreicht PROFINET IRT die kleinsten Zykluszeiten vor Ethernet/IP und EtherCAT.

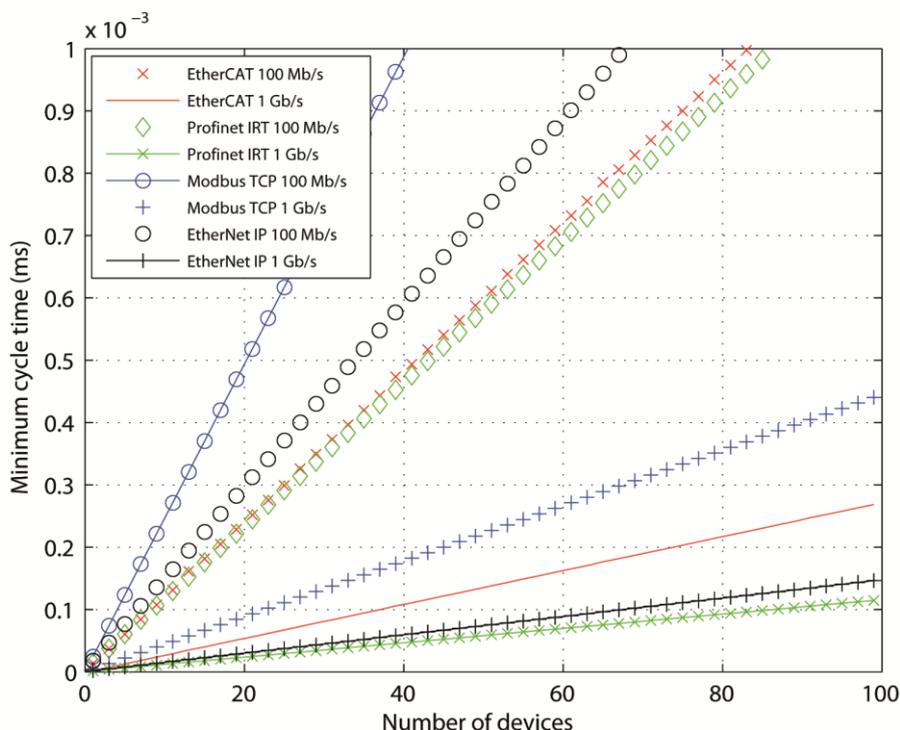


Abbildung 15: Minimale Zykluszeiten als Funktion der Geräteanzahl [RGR+12]

Die Autoren konzentrieren sich bei dem Vergleich ausschließlich auf Aspekte des Kommunikationssystems. Es werden keine architekturellen Unterschiede berücksichtigt. So wird bei modularen EtherCAT-Feldgeräten typischerweise der Rückwandbus als Teil des EtherCAT-Systems ausgeführt, wodurch jedes Modul ein einzelner Ethernet-Teilnehmer ist (Ring-Struktur), während

andere Systeme auf einen dedizierten Rückwandbus setzen (Baum-Struktur). Dies führt dazu, dass bei gleichem Aufbau (hinsichtlich der Feldgeräte- und IO-Anzahl) EtherCAT deutlich mehr Ethernet-Teilnehmer als beispielsweise Ethernet/IP oder PROFINET IRT besitzt. Dies beeinflusst die erreichbaren minimalen Zykluszeiten. Da EtherCAT mit Ringtelegrammen arbeitet, ist die logische Topologie unabhängig von der physikalischen immer ein Ring.

In der Literatur finden sich zahlreiche weitere Vergleiche oder Performance-Bewertungen von Industrial-Ethernet-Systemen. In [LJA07] wird die Performance von Ethernet/IP untersucht, Prytz vergleicht in [Pry08] die Leistungsfähigkeit von EtherCAT und PROFINET IRT. Jasperneite et al. untersuchen in [JSW07] den Leistungssteigerung durch den Wechsel von Fast Ethernet auf Gigabit-Ethernet bei EtherCAT und PROFINET IRT.

2.4 Synchronisation in verteilten Automatisierungssystemen

In Abschnitt 2.1.1 wird die Synchronisation von Komponenten auf eine gemeinsame Zeitbasis als Voraussetzung für isochrone Steuerungssysteme genannt. Daher soll im Folgenden ein Überblick über den Einsatz von Synchronisationsverfahren in verteilten Automatisierungssystemen gegeben werden.

2.4.1 Synchronisation

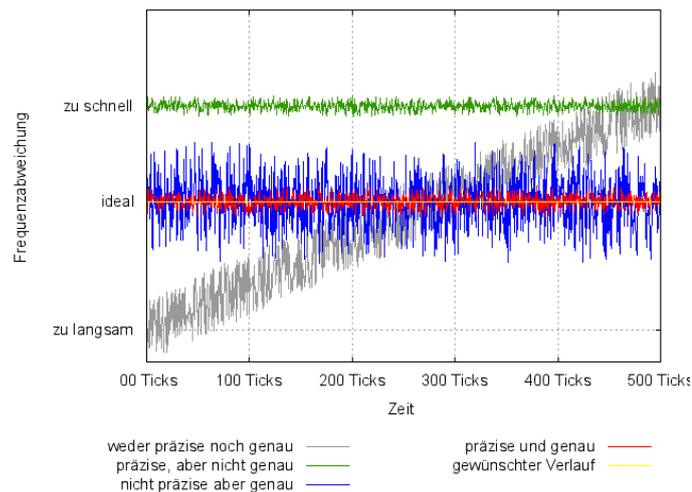
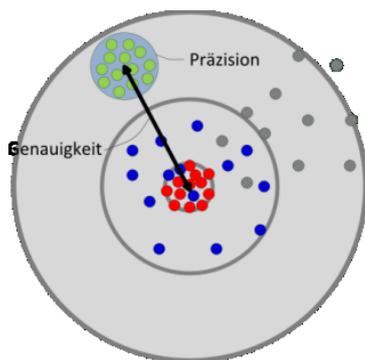
Der Duden definiert *synchronisieren* als „den Gleichlauf zwischen zwei Vorgängen, Maschinen oder Geräte[teile]n herstellen“ [Bib14]. Zur Beschreibung der Güte der Synchronisation von mehreren Vorgängen werden zwei wesentliche Merkmale verwendet: Genauigkeit und Präzision:

„Die **Genauigkeit** beschreibt die Abweichung des Mittelwertes aller Zustände zu einer definierten Vorgabe (Sollwert). Je geringer die Abweichung zum Sollwert ist, desto genauer ist das Synchronisationsverfahren.“ [HKD14, S. 3]

„Die **Präzision** beschreibt [...] die Abweichung der synchronisierten Ereignisse untereinander. Je geringer die Distanz der einzelnen Ereignisse zueinander ist, desto präziser ist die Synchronisation.“ [HKD14, S. 3f.]

Im Kontext von verteilten Steuerungssystemen ist die Präzision die Streubreite von allen synchronisierten Ereignissen um ihren Mittelwert, wohingegen die Genauigkeit der zeitliche Abstand dieses Mittelwerts vom Sollwert ist.

Der Unterschied zwischen Genauigkeit und Präzision ist in Abbildung 16 am Beispiel einer Zielscheibe und einer digitalen Uhr verdeutlicht.



(a) **Abbildung 16: Beispielhafte Darstellung von Genauigkeit und Präzision [HKD14]**

Im Zusammenhang mit Synchronisation wird häufig der Begriff **Drift** verwendet. Nach [ITS96] ist Drift die vergleichsweise langsame Änderung einer Eigenschaft, eines Wertes oder eines Betriebsparameters eines Gerätes¹². Für die Synchronisation wird die Drift der Zeit oder des Taktes betrachtet. Sie gibt an, dass sich das lokale Zeitverständnis langfristig von seinem Sollwert entfernt. Umgangssprachlich ausgedrückt geht die Uhr zu schnell oder zu langsam.

Um ein hohes Maß an Determinismus und kurze Ende-zu-Ende-Reaktionszeiten in verteilten Automatisierungssystemen zu erreichen, kann der Einsatz von Synchronisationsverfahren sinnvoll sein. Ziel der Synchronisation ist dabei, die lokalen Takt- / Zeitgeber der einzelnen Komponenten abzugleichen. In der Regel erzeugt jedes Gerät seinen eigenen Takt und damit auch sein Zeitverständnis basierend auf einer lokalen Baugruppe, dem sogenannten Taktgeber. Dieser wird zum Beispiel in Form eines Quarzes umgesetzt (siehe [BR14, S. 234]). Diese Bauteile unterliegen zum einen einer begrenzten Genauigkeit, d. h. die Frequenz eines konkreten Bauteiles ist nur mit einer Toleranz bekannt (diese wird meist in ppm¹³ als Maß der zulässigen Abweichung angegeben). Zum anderen ist die Frequenz dieser Bauteile nicht konstant. Sie kann sich über die Zeit, z. B. durch Alterungs- oder Temperatureffekte, ändern (siehe [BR14, S. 236]).

*„Für viele Anwendungen ist die **Kurzzeitstabilität** von großer Bedeutung. Hierunter versteht man die Frequenzstabilität innerhalb eines relativ kurzen Betrachtungszeitraums. Üblich sind hier Zeitfenster von 0,1 s bis 100 s; meist wird nur der Wert für eine Zeitdauer von 1 s angegeben.“ [BR14, S. 237]*

Die Auswirkungen dieser abweichenden Zeitgeber werden später im Abschnitt 4.1.1 untersucht. Im Folgenden sollen zunächst Mechanismen zur Synchronisation untersucht werden.

¹² Originaltext: „A comparatively long-term change in an attribute or value of a system or equipment operational parameter.“ [ITS96]

¹³ ppm – parts per million

2.4.2 Verfahren zur Zeitsynchronisation in der Automatisierungstechnik

In der Automatisierungstechnik wird zwischen (Uhr-)Zeitsynchronisation und (Maschinen-)Taktsynchronisation unterschieden werden.

Die **Uhrzeitsynchronisation** bezeichnet dabei eine globale Zeitdomäne. Das lokale Zeitverständnis einer Komponente (d. h. der lokale Zeitgeber) wird gegen eine globale Referenzzeit abgeglichen. Die Uhrzeitsynchronisation wird an vielen Stellen mit unterschiedlichsten Methoden verwendet. So werden Wanduhren mittels DCF77 [PHB04] synchronisiert (sogenannte Funkuhren) und in der IT-Welt werden die Uhren von Computern durch Verfahren wie NTP (siehe Abschnitt 2.4.2.1) synchronisiert.

Die **(Maschinen-)Taktsynchronisation** bezeichnet die lokale Domäne, d. h. die Taktgeber einer lokalen Gruppe von Komponenten werden abgeglichen, ohne Bezug zu einer globalen Referenz.

Beide Domänen unterscheiden sich hinsichtlich der Synchronisationsgüte. Die DIN EN 61784-2 [DIN15] fordert für die Maschinentaktsynchronisation eine Präzision von $\pm 1 \mu\text{s}$ für 64 Knoten, während für die Uhrzeitsynchronisation eine Genauigkeit von $\pm 100 \mu\text{s}$ (applikative Genauigkeit $\pm 1 \text{ms}$) ausreichend ist.

In [HKD14] werden Zeitbegriffe für die beiden Domänen beschrieben. „In der lokalen Domain (Taktsynchronisation) liegt in der Regel eine implizite Systemzeit vor, welche durch das Fehlen einer eigentlichen Uhr gekennzeichnet ist. [...] Eine implizite Systemzeit kann beispielsweise über regelmäßige Trigger-Ereignisse in Form eines Taktes realisiert werden, das den Beginn einer Zeiteinheit kennzeichnen und die entsprechende Aktionen in den Teilnehmern auslösen. [...] Wird hingegen die Systemzeit durch eine Uhr repräsentiert, liegt eine explizite Systemzeit vor.“ [HKD14, S. 4f.]

Die Zeitangaben in beiden Domänen können am Beispiel illustriert werden:

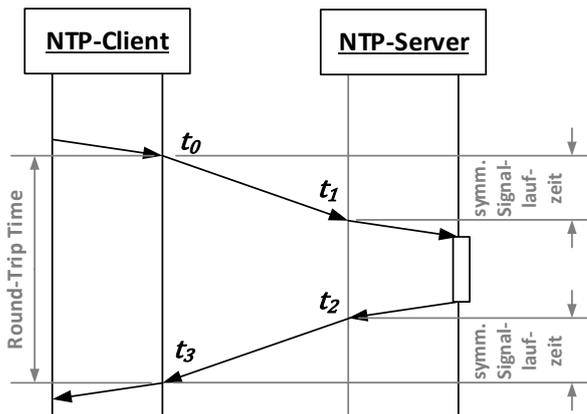
- globale Zeitdomäne „Es ist 02.11.2015 15:47:02 Uhr MEZ“
- lokale Zeitdomäne „Seit dem Start sind 234,5 s vergangen“

Entsprechend der unterschiedlichen Genauigkeitsanforderungen und Domänen haben beide Verfahren verschiedene Anwendungsfälle und Protokolle. Eine Uhrzeitsynchronisation kann in AT-Anlagen z. B. für die Erzeugung von Ereignisprotokollen in verschiedenen Komponenten genutzt werden, während die Taktsynchronisation zum Abgleich von Prozessen mit hohen zeitlichen Anforderungen genutzt wird, z. B. das Abstimmen mehrerer Kommunikationsteilnehmer für zeitbasiertes Senden und Empfangen oder das Abstimmen von Schaltvorgängen und Bewegungen.

2.4.2.1 Verfahren zur Uhrzeitsynchronisation über Industrial Ethernet

Einer der Vorteile vom Einsatz von Ethernet in der Automatisierungstechnik ist die Nutzung von vorhandenen Protokollen. Für die Uhrzeitsynchronisation werden im Wesentlichen zwei Protokolle genutzt: NTP und PTP.

Network Time Protocol (NTP - [Int10]) und das daraus abgeleitete einfachere **Simple Network Time Protocol** (SNTP - [Int06]) können zur Synchronisation von Uhren nach dem Client-Server-Prinzip in globalen Netzwerken (WANs – Wide Area Networks) eingesetzt werden.



- t_0 : Start der Anforderung am Client
- t_1 : Empfang der Anforderung am Server
- t_2 : Start der Antwort am Server
- t_3 : Empfang der Antwort am Client
- $t_3 - t_0$: Round-Trip Verzögerung
- $t_2 - t_1$: Wartezeit am Server

Abbildung 17: Synchronisationsprinzip mit NTP nach [HKD14]

Um die Uhrzeit des Servers auf den Client zu übertragen, muss die Signallaufzeit zwischen Server und Client ermittelt werden. Diese entspricht $t_3 - t_2$ in Abbildung 17. Da t_3 nur im Zeitsystem des NTP-Clients und t_2 im Zeitsystem des NTP-Servers bekannt ist, kann diese Differenz nicht direkt ermittelt werden. Daher wird zunächst die Zeit für eine Round-Trip-Übertragung auf dem Netzwerk (Client zu Server zu Client) ermittelt:

$$\sigma = (t_3 - t_0) - (t_2 - t_1) \tag{1}$$

Unter der Annahme, dass die Signallaufzeit vom Client zum Server und umgekehrt symmetrisch ist, lässt sich der Offset zwischen NTP-Client und -Server nach folgender Gleichung berechnen:

$$\theta = \frac{(t_1 + t_2) - (t_0 + t_3)}{2} \tag{2}$$

Wenn die Verbindung nicht symmetrisch ist, ergibt sich ein systematischer Fehler ϵ , dessen Maximalwert sich abschätzen lässt als:

$$\epsilon \leq \frac{\sigma}{2} \tag{3}$$

Die typische Genauigkeit in größeren Netzen oder über das Internet liegt im Bereich von 10 ms (siehe [KWK09, S. 16]).

In lokalen Netzen (LANs – Local Area Networks) kann mit dem **Precision Time Protocol** (PTP - [IEC09]) eine höhere Synchronisationsgenauigkeit erreicht werden. Dabei arbeitet PTP mit Punkt-zu-Punkt-Verbindungen, wodurch die Unsymmetrie bei der Laufzeitmessung deutlich reduziert werden kann. „Eine weitere Neuerung von PTP im Vergleich zu NTP ist der ‚Best Master Clock – Algorithmus‘ (BMC), welcher die Eigenschaften (Genauigkeit, Drift, Varianz, ...) der kommunizierenden Uhren für alle lokalen Ports des PTP-Elements vergleicht und automatisch eine Baumstruktur, ausgehend von der besten zur Verfügung stehenden Uhr (dem Grandmaster), konfiguriert.“ [HKD14]

In Abbildung 18 ist das Prinzip von PTP dargestellt. Die Synchronisation erfolgt in zwei Phasen, zunächst wird der Korrekturwert zum Master (Offset) ermittelt (gelb). Danach wird in einem zweiten Schritt die Signallaufzeit zwischen Master und Slave (Delay-Messung) bestimmt (grün).

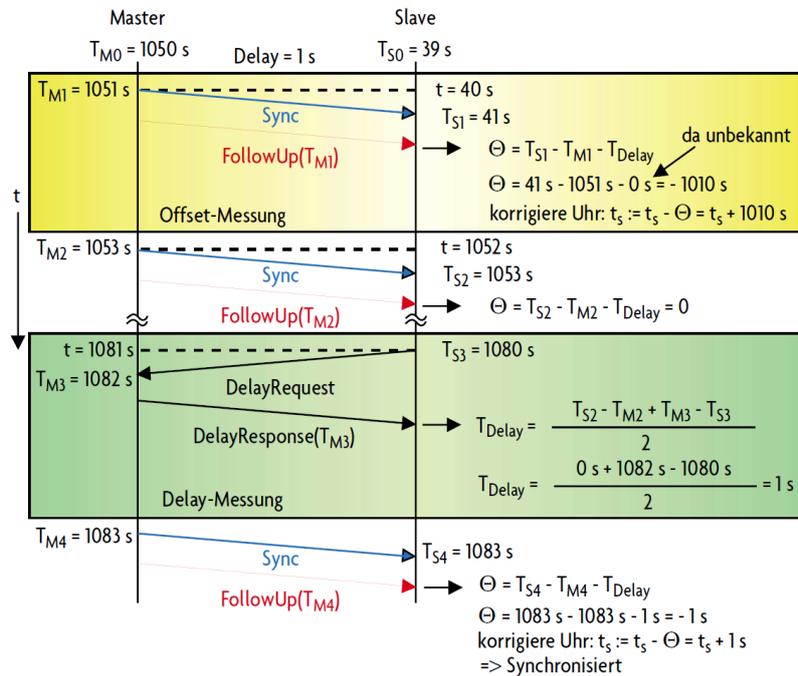


Abbildung 18: Synchronisationsprinzip von PTP [GM03, S. 88]

Gramann und Mohl haben in [GM03] eine Untersuchung zur Genauigkeit von PTP veröffentlicht. Dabei wird festgestellt, dass bei der Implementierung des zeitkritischen Teils (Erfassung von Zeitstempeln) in Hardware eine Genauigkeit von $\pm 100\text{ ns}$ erreichbar ist. Ohne diese Hardwareunterstützung beschreibt [Hir14] eine typische Synchronisationsgenauigkeit im Bereich von $10\text{--}200\text{ }\mu\text{s}$. In [GM03] erläutern Gramann und Mohl weiterhin, dass die Synchronisationsgenauigkeit bei PTP durch die Kurzzeitstabilität der eingesetzten Quarze bestimmt wird. PTP sieht nur eine Korrektur der momentanen zeitlichen Differenz zwischen Master- und Slave-Uhr vor.

Die Ursache des Fehlers, die unterschiedliche Ganggeschwindigkeit der Uhren, wird bei PTP und NTP nicht korrigiert.

2.4.2.2 Verfahren zur Maschinentaktsynchronisation über Industrial Ethernet

Zur weiteren Verbesserung der Synchronisationsgenauigkeit muss neben dem Offset auch die Änderungsrate der lokalen Uhren angepasst werden, eine sog. Driftkorrektur ist notwendig. Daher wurden die existierenden Verfahren erweitert.

Ein Verfahren zur Maschinentaktsynchronisation ist das **Precision Transparent Clock Protocol (PTCP** - [DIN13a]), welches für PROFINET IRT verwendet wird. Dieses Verfahren setzt zur Driftkorrektur auf eine Rate Compensation (siehe Abbildung 19). Dabei wird die Frequenz des Taktgebers über einen PLL-Schaltkreis variiert, so dass die lokalen Zeitgeber aller Komponenten mit der gleichen Ganggeschwindigkeit (im Rahmen der Synchronisationsgenauigkeit) arbeiten.

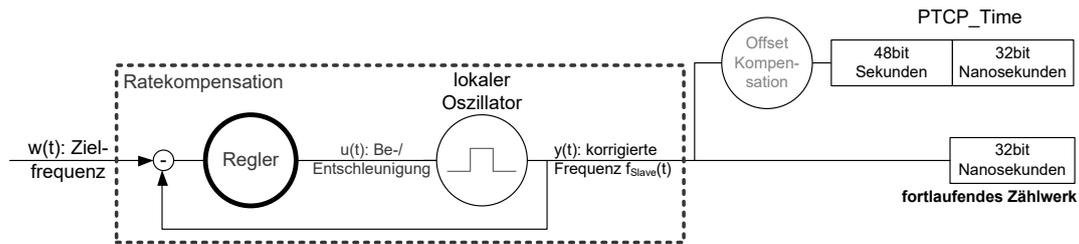


Abbildung 19: Timer-Modell von PTCP bei PROFINET gemäß IEC 61158-6-10 nach [Hir14]

Für PTCP wird eine Synchronisationsgenauigkeit von ± 10 ns angegeben, in Abbildung 20 wird bei einer Messung eine Genauigkeit von ± 14 ns erreicht.

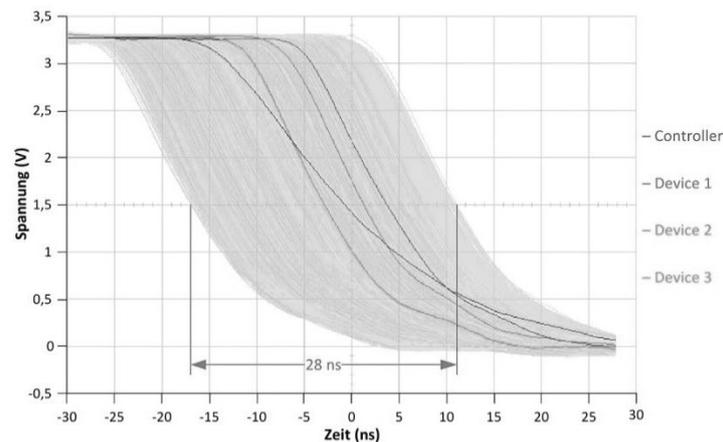


Abbildung 20: Messung der Synchronisationsgenauigkeit von PTCP für 3 Devices [HKD14]

Ein ähnliches Verfahren wird für EtherCAT unter dem Namen **Distributed Clocks** (DC - [DIN13b]) eingesetzt. Auch hier findet eine Driftkorrektur statt, im Gegensatz zu PTCP aber nicht in Hardware sondern durch Anpassung der Zählschritte des lokalen Zählwerkes. So wird im Normalfall mit jedem Takt des Taktgebers der Zähler um 10 erhöht, läuft die lokale Uhr zu schnell wird das Inkrement auf 9 reduziert, läuft sie zu langsam wird er auf 11 erhöht.

Cena et al. haben in [CBS+12] die erreichbare Genauigkeit und Präzision für dieses Verfahren untersucht und stellen fest, dass damit ebenfalls Abweichungen unterhalb von ± 10 ns erreicht werden können. Dabei zeigen sie, dass sich diese Werte mit größer werdenden Netzwerken deutlich verschlechtern. Durch die fehlende Anpassung der physikalischen Taktgeber besteht bei diesem Verfahren weiterhin die Abhängigkeit von der Kurzzeitstabilität der Quarze.

In [HKD14] wurde gezeigt, dass das Verhalten von PTCP bei der Synchronisation einem PI-Regler entspricht, während das von DC einem P-Regler entspricht und damit eine bleibende Regelabweichung behält.

2.5 Modellierung und zeitliche Analyse von Kommunikationssystemen

Es existiert eine Vielzahl von Ansätzen, um industrielle Kommunikationssysteme zu modellieren, deren Zeitverhalten und Leistungsfähigkeit zu analysieren und Verbesserungen vorzuschlagen. Viele dieser Untersuchungen konzentrieren sich auf spezielle Aspekte des Kommunikationssystems.

2.5.1 Analytische Verfahren

In [GLF07] und [Gre07] werden Methoden zur Antwortzeitanalyse von vernetzten Automatisierungssystemen untersucht. Der von den Autoren verwendete Begriff der „Antwortzeit“ ist dabei vergleichbar mit der hier genutzten Klemme-Klemme-Reaktionszeit (siehe Kapitel 3.1). Dabei ist die Ermittlung der Verteilungsfunktion der Antwortzeit von zentraler Bedeutung, da andere Kenngrößen (Minimal- und Maximalzeiten) daraus abgeleitet werden können. „Reine Extremwertanalysen [...] [sind] aufgrund des stochastischen Eigenverhaltens von NAS nicht geeignet“ [GLF07]. Die Autoren kommen zu dem Schluss, dass nur simulative und stochastisch formale Methoden zur Antwortzeitanalyse von NAS geeignet sind. Statische Analysemethoden (z. B. Network Calculus) entfallen, da hiermit nur Aussagen zu den Extremwerten, nicht jedoch zu den Verteilungen der Zeiten getroffen werden können. In dem Beitrag werden anschließend die beiden Methoden „Simulation mittels Modelica“ und „formale Analyse mittels PMC“ verglichen. Während die Ergebnisse beider Ansätze für das gewählte Beispiel vergleichbar sind (siehe Abbildung 21), zeigt der Methodenvergleich, dass mittels PMC trotz höherer Genauigkeit und geringerem Ressourcenaufwand kein geschlossener, kontinuierlicher Kreis (Regelkreis) untersucht werden kann. Für die Untersuchung der Auswirkungen des Steuerungssystems auf den geschlossenen Regelkreis ist der analytische Ansatz somit nicht geeignet.

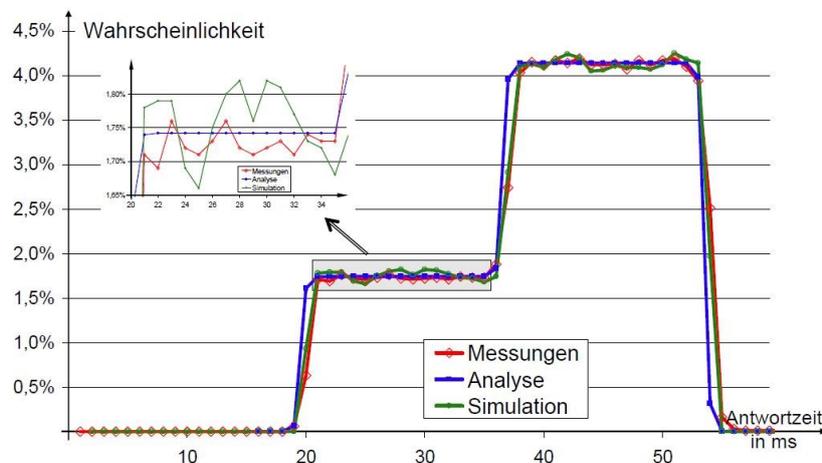


Abbildung 21: Vergleich der Antwortzeitverteilungen [GLF07]

In [SSF+07] beschreiben die Autoren unter anderem die Modellierung von verteilten industriellen Steuerungssystemen. Der Fokus liegt dabei auf der Beschreibung und Analyse von Kommunikationsfehlern. Dazu wird die Struktur mittels Klassendiagrammen beschrieben. Für die Beschreibung der Nachrichtenübermittlung werden Sequenzdiagramme genutzt. Es findet keine Betrachtung von zeitlichen Bedingungen statt. Daher ist das Verfahren nicht für das in dieser Arbeit gesetzte Ziel, der Beschreibung des Zeitverhaltens von verteilten Steuerungssystemen, geeignet.

2.5.2 Simulative Verfahren

2.5.2.1 Modellierung mit Petrinetzen

In [DS14] und [DS15] untersuchen Diekhake und Schnieder die Modellierung und Simulation von verteilten Automatisierungssystemen mit Hilfe von Petrinetzen. Sie beschreiben das System durch eine Attributhierarchie, aus welcher ein Petrinetz-Modell abgeleitet wird. Mittels Simulation werden so Informationen über das zeitliche Verhalten des verteilten Steuerungssystems basierend auf SmallCAN [SMS+11] gewonnen, welche zur Systemüberwachung im Online-Betrieb genutzt werden sollen. Die Ermittlung aller relevanten Informationen und Parameter des Systems wird hier als Schwierigkeit genannt, welche weiterer Untersuchung bedarf.

Dingle et al. zeigen in [DHK02] die Nutzung von generalisierten Stochastischen Petrinetz-Modellen (GSPNs) zur Berechnung von Häufigkeitsverteilung der Antwortzeiten eines Systems. Ausgehend vom Petrinetz-Modell eines Kommunikationssystems wird durch numerische Laplace-Transformation eine Häufigkeitsverteilung der Verzögerungszeit bestimmt. Die Betrachtung wird für ein Datenfluss-Kontroll-Protokoll in Computernetzen (Courier Protocol Software) durchgeführt.

Marsal nutzt in [Mar07] hierarchische, gefärbte Petrinetze zur Bewertung des Zeitverhaltens von Ethernet-basierten Automatisierungssystemen. Dabei steht die Modellierung des Kommunikationsmediums Switched Ethernet im Vordergrund. Es werden verschiedene Kommunikationsformen (Master/Slave, Producer/Consumer, Client/Server) und auch verschiedene Implementierungen (SPS, PC) hinsichtlich ihres Zeitverhaltens (Antwortzeit und Netzwerk-Zykluszeit) mittels numerischer Simulation verglichen.

Alle diese Verfahren nutzen Petrinetze zur Beschreibung des Verhaltens von Kommunikationssystemen. Das konkrete zeitliche Verhalten eines Signalpfades wird jedoch in jedem Fall durch Simulation bestimmt. Eine analytische Berechnung ist nicht möglich. Weiterhin wird in allen diesen Arbeiten nur das Zeitverhalten des Kommunikationssystems selbst untersucht, es erfolgt keine Betrachtung der restlichen Komponenten des Steuerungssystems (Ein- und Ausgabebaugruppen, Steuerungsprogramm, ...).

2.5.2.2 Sonstige Simulationsverfahren und -werkzeuge

Es existiert eine Vielzahl von Tools, welche die Simulation von Kommunikationsnetzwerken unterstützen. Hier sei z. B. OMNeT++ [Ope15] oder DIVAN [HJW10] zu nennen. Durch die sehr detaillierte Abbildung verschiedenster Aspekte der Kommunikationssysteme werden sehr genaue Ergebnissen hinsichtlich des Zeitverhaltens der Kommunikation erzielt. Dies geht jedoch mit einem hohen Aufwand bei der Modellerstellung (welche i.d.R. durch Bibliotheken unterstützt wird) und bei der Durchführung der Simulation einher. Um die gewünschten Verteilungen von zeitlichen Kenngrößen zu ermitteln, ist eine Vielzahl von Simulationen notwendig. Durch die ereignisbasierte Arbeitsweise der Tools ist es schwierig, diese mit einer kontinuierlichen Simulation des geschlossenen Regelkreises zu verbinden.

Andere Simulationswerkzeuge, wie z. B. Dymola (mit der Modellierungssprache Modelica) oder MATLAB/Simulink, bieten die Möglichkeit, ereignisbasierte und kontinuierliche Aspekte zu verknüpfen. In [LF07] zeigen Liu und Frey einen Ansatz zur Bestimmung der Reaktionszeit eines NAS mittels Simulation in Modelica/Dymola. Entsprechend des Bibliothekskonzeptes von Modelica werden Komponenten für die Beschreibung von Teilelementen des Steuerungssystems erstellt oder sind

bereits vorhanden. Es wird gezeigt, dass der Rechenaufwand für ein detailliertes Modell sehr hoch ist (10 000 Samples benötigen mehr als 120 min zur Berechnung), dieser aber durch geschickte Modellreduktion verringert werden kann. Konkret wurde durch die Vernachlässigung der Verzögerung im Switch die Rechenzeit auf 15 min reduziert. Abschließend wird durch die Integration in ein Prozessmodell die Verknüpfung von ereignisdiskreter Netzwerk- und kontinuierlicher Prozesssimulation dargestellt.

Liu stellt in [Liu14] die Bibliothek NCLib für Modelica/Dymola vor, welche eine simulative Analyse von C³-Systemen erlaubt. Durch die Integration des Zeitverhaltens von diskreten Steuerungskomponenten in kontinuierliche Systemmodelle zu hybriden Systemen erlaubt dieser Ansatz die Simulation des gesamten NCS-Regelkreises (siehe Beispiel in Abbildung 22). Liu unterscheidet in seiner Betrachtung zwischen interaktiven und nicht-interaktiven Verzögerungen. Letztere hängen in ihrem Wert nur von einer Komponente ab, während erstere von der Verknüpfung mehrerer Komponenten abhängen. Beide Verfahren zeichnen sich durch einen hohen Rechenaufwand aus, wie er für simulative Ansätze typisch ist.

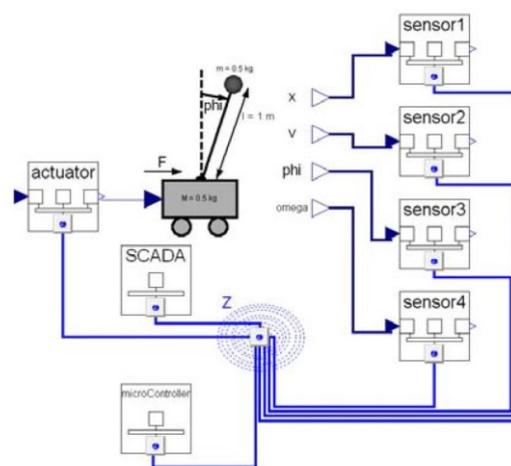


Abbildung 22: Modell eines invertierten Pendels in NCLib [WLF08, S. 121]

In [LC02] und [CHL+03] wird das Simulationswerkzeug Jitterbug vorgestellt, welches als MATLAB-basierte Toolbox quadratische Performance-Kriterien für lineare Regelungssysteme bestimmt. Dazu wird neben dem Signalmodell (signal model) ein Zeitmodell (timing model) eingeführt, welches die Verarbeitungszeit der einzelnen Teilsysteme durch Häufigkeitsverteilungen beschreibt.

In Abbildung 23 ist ein Jitterbug-Modell für ein einfaches Regelungssystem dargestellt. Dabei ist im linken Teil das Signalmodell (a) und im rechten das Zeitmodell (b) dargestellt. Der Prozess wird durch das kontinuierliche System $G(s)$ und der Regler durch drei zeitdiskrete Systeme $H_1(z)$, $H_2(z)$ und $H_3(z)$ beschrieben.

Jitterbug setzt die Kenntnis der Zeitverzögerungen und Zykluszeiten voraus, daher setzen die Autoren das Werkzeug TrueTime ein, um das Zeitverhalten zu bestimmen. TrueTime ist ein Simulator für das Zeitverhalten von Multitasking-Echtzeitsystemen und Netzwerk-Übertragungen, der ebenfalls an der Universität Lundt entwickelt wird. Durch die simulative Ermittlung des Zeitverhaltens eines Systems kann so die Qualität der Regelung bewertet werden. Schwerpunkt der Arbeiten liegt auf der Wechselwirkung zwischen Regler- und Kommunikationssystem-Entwurf.

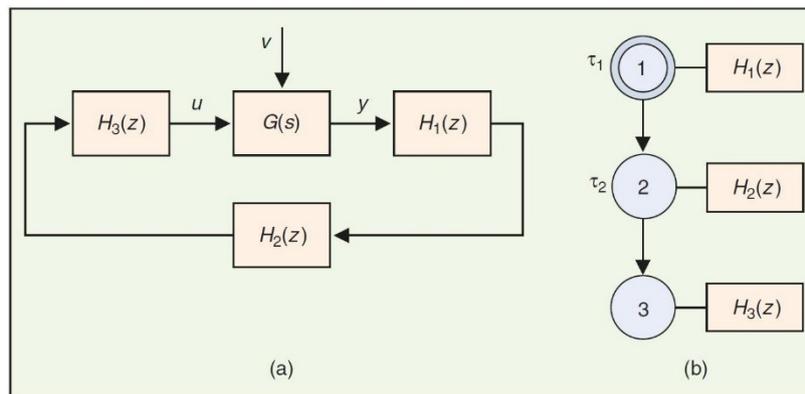


Abbildung 23: Jitterbug-Modell eines einfachen Regelungssystems [CHL+03, S. 19]

Neben der Übertragung von Prozessdaten können auch andere Aspekte des Kommunikationssystems simuliert werden. So stellen Ferrari et al. in [FFR+08] eine Simulationsumgebung zur Untersuchung der Genauigkeit von verschiedenen, koexistierenden Synchronisationsverfahren bei Real-Time Ethernet vor. Damit kann ein spezieller Aspekt des Kommunikationssystems untersucht werden, es erfolgt jedoch keine Beschreibung des Zeitverhaltens des Prozessdatentransports.

Jasperneite führt in [Jas02] eine Leistungsbewertung von Switched Ethernet für die Echtzeitkommunikation durch eine simulative Analyse durch. Dabei wird das Simulationswerkzeug OPNET Modeler¹⁴ genutzt. Der Autor kommt zu der Erkenntnis, dass sich Switched Ethernet auf der MAC-Ebene deterministisch verhält und dadurch für die Nutzung in der „prozessnahen Echtzeitkommunikation“ geeignet ist. Auch hier wird ein spezieller Aspekt von Industrial Ethernet modelliert und analysiert. Für die Analyse des Zeitverhaltens von verteilten Steuerungssystemen ist dieses Verfahren nicht ausreichend.

Das Ziel dieser Arbeit, eine analytische Modellierung des Zeitverhaltens von verteilten Steuerungssystemen lässt sich durch rein simulative Ansätze nicht erreichen. Simulationen sind in der Regel mit hohem Rechenaufwand verbunden. Dies ist für die Bewertung und den Vergleich von verschiedenen Steuerungssystemen ungünstig.

2.5.3 Messtechnische Analyse

Cena et al. beschreiben in [CBV+07] die messtechnische Bewertung der Antwortzeiten von industriellen WLAN-Systemen. Dabei arbeiten sie mit einem vereinfachten Systemmodell. Dieses stellt die Antwortzeit als Summe von Einspeisungsjitter (release jitter), Wartezeit und Übertragungszeit dar. Bei der theoretischen Analyse stehen die Besonderheiten von WLAN im Vordergrund. Zur Reduzierung der Einflüsse durch Steuerungsapplikationen werden ad-hoc-Netzwerke genutzt, bei denen die Daten direkt vom Sender zum Empfänger übertragen werden. Die experimentelle Untersuchung bestimmt die Wahrscheinlichkeitsdichtefunktion der Antwortzeiten und vergleicht die gemessenen mit berechneten Werten aus dem vereinfachten Systemmodell und einem Netzwerksimulator. Es werden die Auswirkungen von Interferenzen, Hintergrund-Verkehr und QoS-Mechanismen auf die Antwortzeiten untersucht und eine gute Übereinstimmung zwischen Simulation und Experiment gezeigt.

¹⁴ „OPNET Modeler“ ist mittlerweile als "SteelCentral NetModeler Suite" der Firma Riverbed erhältlich [Riv12]

Nilsson schlägt in [Nil98] eine Modellierung des Zeitverhaltens von Kommunikationssystemen mit Markow-Ketten vor. Dabei werden Zeitverzögerungen durch Kommunikationssysteme in 3 Arten untergliedert:

- Konstante Zeitverzögerung
- Zufällige Zeitverzögerung, die unabhängig von Übertragung zu Übertragung ist
- Zufällige Zeitverzögerung, deren Wahrscheinlichkeitsverteilung durch eine unterlagerte Markow-Kette bestimmt wird

Eine konstante Zeitverzögerung ist die einfachste Abbildungsform, sie eignet sich für Netzwerke mit konstanten Übertragungszeiten oder wenn die Verzögerung deutlich kleiner ist als die Zeitskala des Prozesses. In diesem Fall wird der Worst-Case-Wert angenommen. Aufgrund zufälliger Prozesse ist die Zeitverzögerung in einem Netzwerk oft zufällig in bestimmten Grenzen, d. h. sie lässt sich über eine Häufigkeitsverteilung abbilden. Dabei wird davon ausgegangen, dass die Zeitverzögerungen von aufeinanderfolgenden Übertragungen in keiner zeitlichen Relation stehen, d. h. die Zeitverzögerung ist jeweils zufällig entsprechend der Häufigkeitsverteilung. Für die Beschreibung von Effekten wie Warteschlangen und Netzlast, muss das Netzwerkmodell ein Gedächtnis bekommen. Durch die Verwendung einer Markow-Kette kann die Zeitverzögerung bei jeder Übertragung abhängig vom Vorzustand bestimmt werden. Jeder Zustand der Markow-Kette entspricht einer Häufigkeitsverteilung der Zeitverzögerung. In Abbildung 24 ist ein einfaches Modell zur Beschreibung von drei verschiedenen Netzlasten (L – niedrige Netzlast, M – mittlere Netzlast, H – hohe Netzlast) dargestellt. Dabei gibt q_{ik} die Übergangswahrscheinlichkeit vom Zustand i nach k an, mit $i \in \{L; M; H\}$ und $k \in \{L; M; H\}$. Die Verzögerungen der einzelnen Netzlasten sind wie in Abbildung 25 dargestellt als Häufigkeitsverteilungen angegeben. [Nil98]

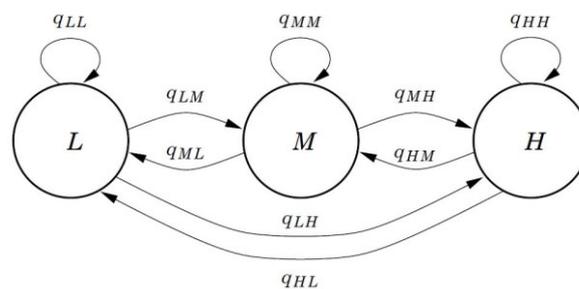


Abbildung 24: Beispiel einer Markow-Kette zur Beschreibung eines Netzwerks [Nil98, S. 32]

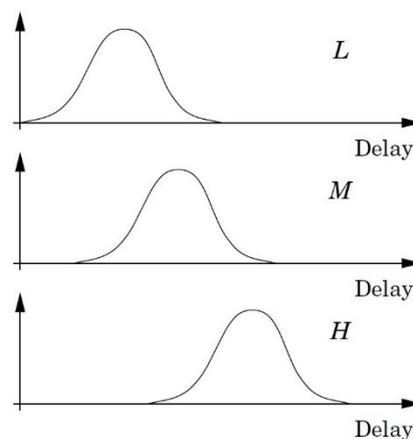


Abbildung 25: Verzögerungen in den Zuständen der Markow-Kette [Nil98, S. 32]

Die Bestimmung der Häufigkeitsverteilungen erfolgt in [Nil98] nur durch experimentelle Messung der Übertragungszeiten bei CAN und Ethernet. Die Arbeit konzentriert sich stärker auf die regelungstechnischen Aspekte.

Um die Auswirkungen von verschiedenen Arten und Konfigurationen von Steuerungssystemen auf eine reale Anlage zu untersuchen, sind messtechnische Verfahren schlecht geeignet. Der Aufwand zur Bestimmung des Zeitverhaltens für alle möglichen Konfigurationen ist sehr hoch, da diese aufgebaut und analysiert werden müssen.

2.6 Grundlagen der Modellierung von dynamischen Systemen

Mit dem im Rahmen dieser Arbeit bestimmten Zeitverhalten eines Steuerungssystems, soll die Verwendung in geschlossenen Regelkreisen bewertet werden. Dazu werden hier kurz die Grundlagen der Modellierung von dynamischen Systemen beschrieben.

2.6.1 LTI-Systeme

Ausgangspunkt der Betrachtungen ist die Modellierung eines Prozesses als dynamisches System Σ , welches auf ein Eingangssignal $u(t)$ mit einem Ausgangssignal $y(t)$ reagiert (Abbildung 26)



Abbildung 26: Strukturbild des Systems [Lun14b, S. 55]

Im Speziellen werden hier **lineare, zeitinvariante Systeme** (LTI-Systeme¹⁵) betrachtet:

„Ein dynamisches System Σ heißt linear, wenn sich die Wirkungen zweier linear überlagerter Eingangssignale am Ausgang des Systems in gleicher Weise linear überlagern (Superpositionsprinzip). Wird also für $u(t)$ die Linearkombination

$$u(t) = k u_1(t) + l u_2(t) \quad (4)$$

der beiden Funktionen $u_1(t)$ und $u_2(t)$ eingesetzt, so fordert das Superpositionsprinzip, dass sich die dabei erhaltene Ausgangsgröße $y(t)$ als Linearkombination

$$y(t) = k y_1(t) + l y_2(t) \quad (5)$$

darstellen lässt.“ [Lun14b, S. 65f.]

Ein zeitinvariantes System „reagiert auf eine Erregung (also einen vorgegebenen Verlauf $u(t)$) unabhängig davon, wann die Erregung eintrifft. Das heißt, dass eine Verschiebung der Eingangsgröße auf der Zeitachse nach rechts eine gleichgroße Verschiebung der Ausgangsgröße bewirkt. Man spricht hierbei auch vom Verschiebepinzip.“ [Lun14b, S. 70]

Die Betrachtung von LTI-Systemen stellt an dieser Stelle keine Einschränkung der Arbeiten zur Modellierung des Zeitverhaltens von verteilten industriellen Steuerungssystemen dar. Die Anwendung des Modells zur Bestimmung der Regelgüte in Kapitel 8 nutzt diese sehr häufige Beschreibungsform dynamischer Systeme.

¹⁵ LTI = lineares zeitinvariantes System (englisch linear time-invariant system)

2.6.2 Beschreibung von linearen Systemen

Der Zusammenhang zwischen den Ein- und Ausgangsgrößen von linearen Systemen kann durch eine **Differentialgleichung**¹⁶ beschrieben werden. Nach [Lun14b, S. 57] hat eine solche Beschreibung die allgemeine Form:

$$\frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1}}{dt^{n-1}} + \dots + a_1 \dot{y}(t) + a_0 y(t) = b_q \frac{d^q u}{dt^q} + \dots + b_1 \dot{u}(t) + b_0 u(t) \quad (6)$$

Anstelle einer Differentialgleichung der Ordnung n kann das System durch n Differentialgleichungen erster Ordnung beschrieben werden. Diese Darstellung nennt sich **Zustandsraummodell** und wird in der Regelungstechnik sehr häufig verwendet. Die typische Schreibweise ist in Matrizenform (nach [Lun14b, S. 72f.]):

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t), \quad \mathbf{x}(0) = \mathbf{x}_0 \\ y(t) &= \mathbf{c}^T \mathbf{x}(t) + du(t) \end{aligned} \quad (7)$$

mit

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}, \quad \dot{\mathbf{x}}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{pmatrix}, \quad (8)$$

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \quad (9)$$

$$\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad (10)$$

$$\mathbf{c}^T = (c_1 \quad c_2 \quad \dots \quad c_n) \quad (11)$$

Durch Laplace-Transformation und anschließende Umformung kann die Differentialgleichung in die sog. **Übertragungsfunktion** überführt werden. Dabei wird davon ausgegangen, dass der Anfangszustand des Systems verschwindet ($\mathbf{x}_0 = 0$). „Die Übertragungsfunktion wird definiert als Quotient der Laplacetransformierten der Ausgangsgröße und der Eingangsgröße des Systems.“ [Lun14b, S. 255]

$$G(s) = \frac{Y(s)}{U(s)} \quad (12)$$

¹⁶ „Differentialgleichung wird eine Gleichung genannt, in der neben einer oder mehreren unabhängigen Veränderlichen und einer oder mehreren Funktionen dieser Veränderlichen auch noch die Ableitungen dieser Funktionen nach den unabhängigen Veränderlichen auftreten. Die Ordnung einer Differentialgleichung ist gleich der Ordnung der höchsten in ihr auftretenden Ableitung.“ [BS05, S. 504]

Für die Beschreibung von Verzögerungen, wie sie durch das Zeitverhalten von Steuerungssystemen entstehen, muss die Beschreibung um ein **Totzeit-Element** oder Totzeit-Glied erweitert werden. „Das reine Totzeitglied verschiebt die Eingangsgröße, ohne sie zu verformen, so dass

$$y(t) = u(t - T_t) \quad (13)$$

gilt.“ [Lun14b, S. 113] Dabei ist T_t die sogenannte Totzeit. Insbesondere die Darstellung als Übertragungsfunktion hat sich hier als geeignet erwiesen. In der Literatur findet das Proportionale Totzeit-Element (PT_t-Element) Anwendung, es besteht aus einer Proportionalverstärkung K_P und einer Totzeit T_t (nach [LW10, S. 33ff.]):

$$G(s) = K_P \cdot e^{-sT_t} \quad (14)$$

Zur grafischen Darstellung von Systemen werden **Blockschaltbilder** verwendet. Dabei werden die Systeme als Blöcke und ihre Ein- und Ausgangsgrößen in Form von Pfeilen dargestellt, wie in Abbildung 27 dargestellt.

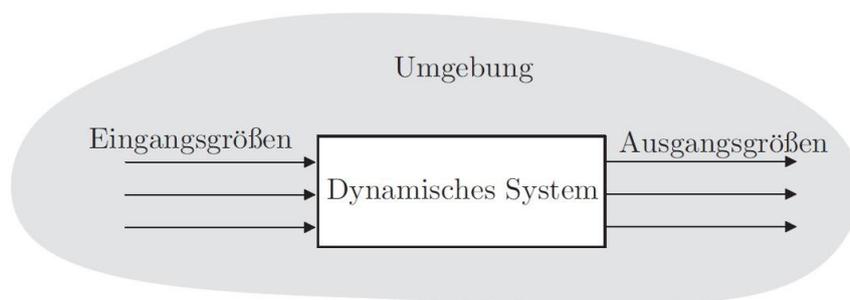


Abbildung 27: Blockschaltbild eines Systems [Lun14b, S. 42]

2.6.3 Regelkreis

„Unter einer Regelung versteht man einen Vorgang, bei dem eine Größe, die Regelgröße, fortlaufend gemessen wird und mit einer anderen Größe, der Führungsgröße, verglichen wird. Mit dem Vergleichsergebnis wird die Regelgröße so beeinflusst, dass sich die Regelgröße der Führungsgröße angleicht. Der sich ergebende Wirkungsablauf findet in einem geschlossenen Kreis, dem Regelkreis, statt.“ [LW10, S. 21]

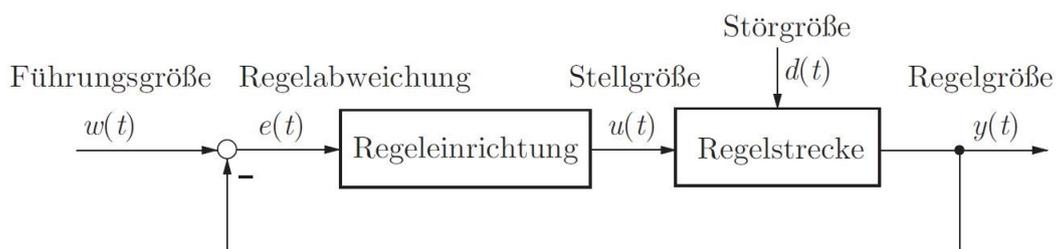


Abbildung 28: Grundstruktur des Regelkreises [Lun14b, S. 4]

In Abbildung 28 ist die Struktur des Regelkreises als Blockschaltbild dargestellt.

2.6.4 Stabilität

Stabilität ist eine Grundvoraussetzung für Regelkreise. „Unter Stabilität versteht man die Eigenschaft eines Systems, auf eine beschränkte Erregung mit einer beschränkten Bewegung zu reagieren.“ [Lun14b, S. 407]

In der Literatur finden sich zahlreiche Untersuchungen zur Stabilität von NCS und den Einflussfaktoren (z. B. [ZBP01], [WYB02], [DHB+12], [MEA07], [MEH+09], [MA02]). Stabilitätsbetrachtungen gehen in der Regel vom Worst-Case-Fall aus, d. h. von der größten zu erwartenden Zeitverzögerung durch ein System.

Sehr häufig wird des **MATI-Kriterium** (maximum allowable transfer interval – siehe [WYB02]) zur Bestimmung der maximalen Verzögerung, bei der ein gegebenes LTI-System (Strecke und Regler) stabil ist, verwendet.

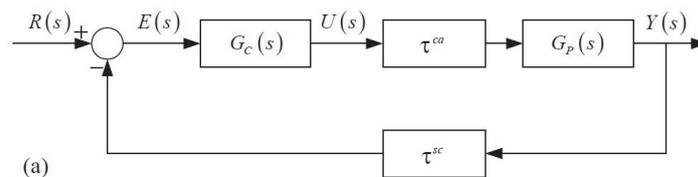
2.6.5 Regelgüte - Quality of Control

Während Stabilität eine notwendige Bedingung für die Durchführung einer Regelungsaufgabe ist, beschreibt die Regelgüte, wie „gut“ die Aufgabe erfüllt wird. In der Literatur findet sich vielfach der Begriff **Quality of Control** (QoC - Regelgüte). Greifeneder beschreibt ihn für die klassische kontinuierliche Regelungstechnik: „Dieses Maß basiert auf der Generalisierbarkeit der zugrunde liegenden Problembeschreibung und ist i. d. R. quadratisch über diversen Eigenschaften, wie z. B. der Stabilitätsreserve, der Konvergenzzeit oder dem Überschwingverhalten definiert.“ [Gre07, S. 8]

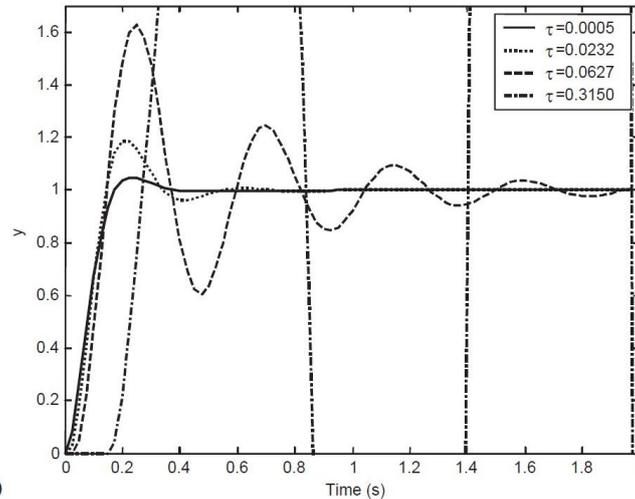
Koller und Sauter stellen in [KS04] eine Fallstudie zum Einfluss der Datenübertragung mittels Ethernet auf einen Regelkreis vor. Dabei wird das Verhalten des Kommunikationssystems auf eine statistische Verteilung der Übertragungszeit reduziert. Die Autoren benennen die Wahl der geeigneten Verteilungsform als Schwierigkeit und nehmen eine Poisson-Verteilung an. Sie simulieren dabei die Sprungantwort des Beispielsystems (magnetischer Schwebekörper) mit Kommunikationssystem unter verschiedenen Konfigurationen. Zur Bewertung wird die Überschwingstatistik verwendet.

Abbildung 29 zeigt beispielhaft den Einfluss der Zeitverzögerung durch das Steuerungs- und Kommunikationssystem auf die Sprungantwort eines linearen Systems. Im oberen Teil wird das Beispielsystem bestehend aus Regler $G_C(s)$, Strecke $G_P(s)$, Verzögerung vom Sensor zum Regler τ^{sc} und Verzögerung vom Regler zum Aktor τ^{ca} dargestellt. Im unteren Teil ist die Sprungantwort für verschiedene Verzögerungen bei gleichmäßiger Aufteilung $\tau^{ca} = \tau^{sc} = \tau/2$ dargestellt. Es wird deutlich, dass eine zunehmende Verzögerung durch das Kommunikationssystem die Regelgüte negativ beeinflusst.

Wie in Abschnitt 2.7.1.3 beschrieben hat die Lage der Verzögerung im einschleifigen Regelkreis keine Auswirkungen, so dass die beiden Verzögerungen zusammengefasst werden können. In Abbildung 29 werden nur die Verzögerung vom Sensor zum Regler und vom Regler zum Aktor betrachtet. Allgemein muss aber auch die Verzögerung des Reglers selbst betrachtet werden. Die Summe der Verzögerungen entspricht damit der Verzögerung vom Sensor über den Regler bis zum Aktor, welche als Klemme-Klemme-Reaktionszeit t_{IORT} bezeichnet wird. Für die Bewertung von Stabilität (und Regelgüte) eines geschlossenen Regelkreises mit einem technischen Steuerungssystem ist daher die genaue Kenntnis der Klemme-Klemme-Reaktionszeit notwendig.



(a)



(b)

Abbildung 29: Einfluss der Verzögerung auf die Regelgüte [TC03, S. 1103]

In Abbildung 30 ist die Klemme-Klemme-Reaktionszeit im Regelkreis von Abbildung 29a dargestellt.

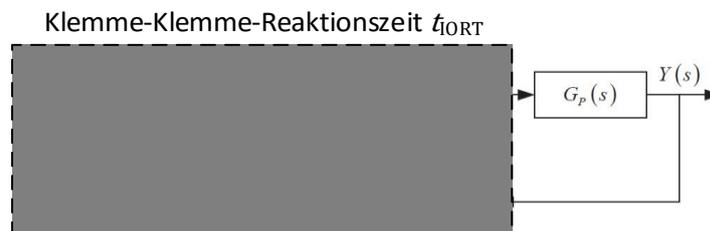


Abbildung 30: Klemme-Klemme-Reaktionszeit im Regelkreis

Zur Bewertung der Eigenschaften eines Regelkreises wurde das Arbeitsgebiet des **Control Performance Monitoring** (CPM) etabliert. In der Literatur wird anstelle von CPM auch der Begriff **Control Performance Assessment** (CPA) verwendet, wobei beide synonym genutzt werden (siehe [Jel06]). Dabei wurden „Methoden und Werkzeuge [...], die der fortlaufenden Überwachung, Bewertung und Diagnose von Regelungssystemen dienen“ [FA09, S. 142] entwickelt. In der Literatur wird dieses Verfahren überwiegend für prozesstechnische Anlagen genutzt.

Eine Herangehensweise ist der Vergleich von Soll- und Ist-Daten des realen Regelkreises mit den erwarteten Ergebnissen bei Verwendung eines optimalen Reglers. Dieses Vorgehen wird als Benchmarktest, die abgeleiteten Maßzahlen als Control-Performance-Index (Güteindex) bezeichnet [FA09, S. 145]. Unter dem Begriff Harris-Index $CPI(d)$ [Har89] wird ein Performance-Index verwendet, welcher die auftretende Varianz (Streuung) σ_{MV} eines Minimum-Varianz-Reglers (siehe [Ise88]) mit der tatsächlichen Varianz σ_y vergleicht:

$$CPI(d) = \frac{\sigma_{MV}^2}{\sigma_y^2} \quad (15)$$

Der Harris-Index ist jedoch in seiner Anwendbarkeit begrenzt. Horch und Isaksson [HI01] empfehlen ihn für Systeme, deren Performance durch die Totzeit des Prozesses bestimmt ist. Daher wurden verschiedene Erweiterungen und Modifikationen entwickelt, beispielsweise um den Index für Systeme, bei denen instabile Pole das Verhalten bestimmen ([TM95] zitiert nach [HI01]), oder für Systeme mit kleinen oder keinen Totzeiten [HI99] zu nutzen. In [HI01] wird basierend auf diesem Verfahren eine Methode zur Bewertung von Zykluszeiten der Regler präsentiert, um den Nutzen von schnellerer Verarbeitung zu quantifizieren.

2.7 Networked Control Systems

Ein **Networked Control System (NCS)** liegt vor, wenn ein Regelkreis über ein Echtzeit-Kommunikationssystem geschlossen wird (nach [GC08]).

„Während bei NCS (Network Control Systems [...]) die Stabilitätsanalyse des kontinuierlichen Reglers im Mittelpunkt steht, liegt bei NAS der Schwerpunkt in der Verlässlichkeits- und Qualitätsanalyse. Und somit [...] implizit in der Analyse von Antwortzeiten.“ [Gre07, S. 6]

Die Zuordnung zu NCS erfordert den Einsatz eines Kommunikationssystems, daher muss die Architektur des Steuerungssystems dezentral oder (kooperativ) verteilt sein. Auch hier wird hinsichtlich des zeitlichen Verhaltens der Komponenten (zyklisch oder ereignisbasiert) keine Einschränkung gemacht.

Das Forschungsgebiet NCS beschäftigt sich nicht mit der Modellierung und Analyse des Steuerungs- und Kommunikationssystems selbst, sondern mit den Auswirkungen selbiger auf den geschlossenen Regelkreis. In den letzten Jahrzehnten gab es sehr viele Arbeiten zum Thema NCS. Daher kann an dieser Stelle nur ein kurzer Überblick über den Stand der Technik gegeben werden.

2.7.1 Besonderheiten von NCS

Hespanha et al. geben in [HNY07] einen Überblick über aktuelle Entwicklungen im Bereich NCS. Sie benennen dabei vier Punkte, die ein NCS von einem klassischen Regelungssystem unterscheiden, welches von einem idealen Kanal zur Übertragung von Informationen ausgeht:

2.7.1.1 Bandbreitenbegrenzte Kanäle (*Band-limited channels*)

Das Kommunikationssystem in einem NCS stellt in der Regel keine unbegrenzte Bandbreite zur Verfügung. Speziell bei der Kommunikation über weite Entfernung oder unter schwierigen Bedingungen kann die Bandbreite stark begrenzt sein. Daher wurden zahlreiche Untersuchungen zur Ermittlung der minimalen Bitrate für die Stabilisierung eines Regelkreises durchgeführt (z. B. [HOV02], [TM04]). In [Mis14] wird die Verwendung von ereignisgetriebenen Regelungsstrategien zur Reduzierung der benötigten Bandbreite untersucht und verschiedene Regelungsverfahren vorgestellt und bewertet.

Im Umfeld von Industrial Ethernet ist dieser Punkt aufgrund der hohen Bandbreite von untergeordneter Bedeutung. Im Allgemeinen ist die zur Verfügung stehende Bandbreite ausreichend.

2.7.1.2 Systemarchitektur (Systems architecture)

Für die Untersuchung der regelungstechnischen Eigenschaften eines NCS wird anstelle der allgemeinen Architektur (Abbildung 31) häufig eine vereinfachte Architektur mit einem einfachen Regelkreis (Abbildung 32) genutzt, welcher nach [HNY07] für viele Aspekte hinreichende Aussagekraft besitzt.

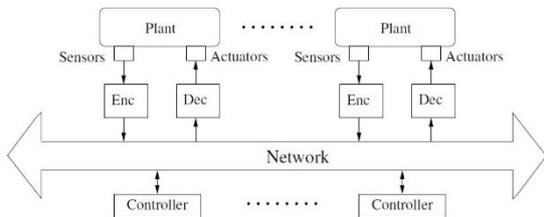


Abbildung 31: Allgemeine NCS-Architektur [HNY07, S. 138]

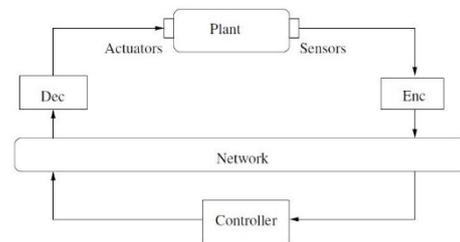


Abbildung 32: Ein-Schleifen-NCS [HNY07, S. 139]

Tipsuwan und Chow zeigen in [TC03] ebenfalls zwei verschiedene Strukturen für NCS, die direkte (Abbildung 33) und die hierarchische Struktur (Abbildung 34).

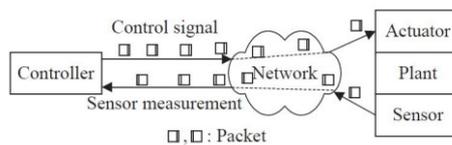


Abbildung 33: NCS in direkter Struktur [TC03, S. 1100]

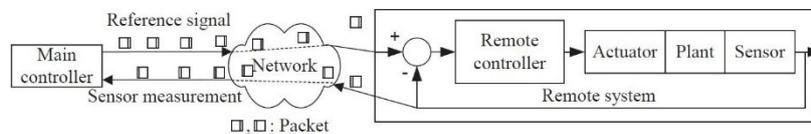


Abbildung 34: NCS in hierarchischer Struktur [TC03, S. 1101]

Für die Betrachtung der regelungstechnischen Eigenschaften ist die Nutzung der Ein-Schleifen-NCS von Vorteil, da zahlreiche Mittel der klassischen Regelungstechnik genutzt werden können.

2.7.1.3 Abtastung und Verzögerung (Sampling and Delay)

Während die klassische Regelungstechnik von kontinuierlichen Signalen ausgeht, werden die Informationen bei NCS nur zu bestimmten Zeitpunkten übertragen. Zusätzlich hat das Kommunikationssystem ein Totzeitverhalten mit (variablem) Jitter.

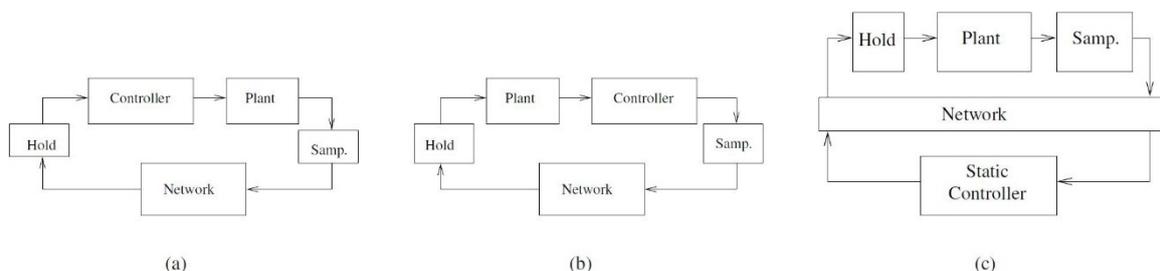


Abbildung 35: Vom Ein-Schleifen-NCS abgedeckte Architekturen [HNY07, S. 144]

In [HNY07] wird das **one-channel feedback NCS** (Ein-Schleifen-NCS) zur Untersuchung des Einflusses von Abtastung und Verzögerung auf einen Regelkreis genutzt. Wie in Abbildung 35 gezeigt, hat die Lage der Verzögerung innerhalb des Regelkreises (bestehend als LTI-Strecke und -Regler) im SISO¹⁷-Fall keinen Einfluss auf das Ergebnis, weshalb die allgemeine Betrachtung alle drei Varianten abdeckt.

Abbildung 36 zeigt ein typisches Modell für NCS (nach [ZBP01]), welches für den Regelkreis einen einzelnen Abtastvorgang vorsieht. Die digitalen Systeme werden als ereignisorientiert angenommen, weshalb die auftretenden Verzögerungen aufsummiert werden können (siehe auch [Liu14, S. 12]).

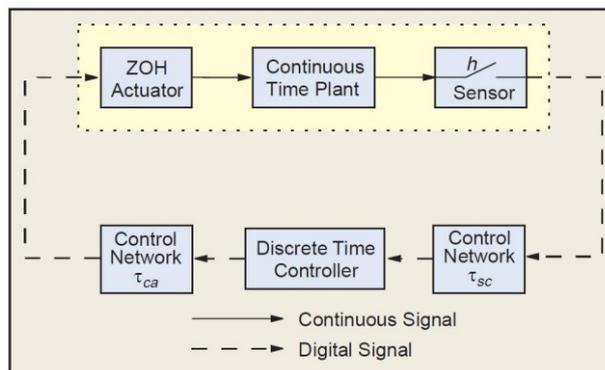


Abbildung 36: NCS-Modell mit Netzwerk-induzierter Verzögerung [ZBP01, S. 88]

2.7.1.4 Paketverlust (Packet dropout)

Bei der Datenübertragung können Informationen (Pakete) durch äußere Störungen oder Kollisionen auf dem Übertragungsmedium verloren gehen. Heute wird Industrial Ethernet fast ausschließlich als Switched Ethernet betrieben, wodurch Paketkollisionen auf dem Übertragungsmedium vermieden werden können und hier nicht weiter betrachtet werden. Daher werden Kollisionen und Störungen hauptsächlich bei drahtloser Kommunikation untersucht. Untersuchungen zum Einfluss von Paketverlusten auf NCS finden sich z. B. in [ZBP01]. In [BKS10] wird der Einfluss von Paketverlusten und Paketwiederholungen auf die Übertragungszeit von drahtlosen Kommunikationssystemen untersucht.

Neben dem Paketverlust kann auch eine Reihenfolgenvertauschung auftreten. Diese wird bei vielen Kommunikationssystemen durch Zähler detektiert. Untersuchungen zur Auswirkung von Reihenfolgenvertauschungen und deren Kompensation im Regler finden sich z. B. in [PH08] auf Basis von Bayes'schen Netzen.

2.7.2 Kompensation von NCS

Während die obigen Betrachtungen davon ausgehen, dass ein klassischer Regler für ein NCS genutzt wird, wurden neue Regelungsansätze entwickelt, deren Ziel es ist, die Nachteile der NCS auszugleichen.

In [HNY07] werden Verfahren zur Zustandsschätzung über verlustbehaftete Netzwerke beschrieben. Dabei wird ein Zustandsbeobachter im Regler genutzt, um basierend auf partiellen Messdaten und einem Systemmodell auf den aktuellen Zustand (und damit die aktuellen Sensorsignale) zu schlussfolgern. Ähnliche Ansätze finden sich in [NO08].

Durch den Einsatz von Warteschlangen (Queues) können starke Streuungen der Verzögerungen reduziert werden [CO94]. Dies führt jedoch zu einer Erhöhung der mittleren Verzögerungszeit und ist

¹⁷ SISO = "Single Input, Single Output"

daher nicht für jede Anwendung sinnvoll. Der Einsatz von Warteschlangen wird oftmals mit einer Zustandsschätzung verknüpft [LR94].

Nilsson schlägt in [Nil98, NBW98] die Nutzung eines LQG-Reglers (linear quadratic gaussian regulator) vor, um bei einer bekannten Häufigkeitsverteilung der Verzögerung einen optimalen Regler zu entwerfen (siehe auch [Lun14a, S. 368]).

Liu et al. stellen in [LXR+07] einen Ansatz zum Entwurf eines „network predictive control systems“ vor, für welches sie bei begrenzter zufälliger Netzwerkverzögerung Stabilität nachweisen können.

Während der optimierte Reglerentwurf für NCS nicht im Fokus dieser Arbeit steht, setzen auch diese Verfahren eine detaillierte Kenntnis der von einem Steuerungssystem verursachten tatsächlichen Zeitverzögerung voraus. Somit könnte das in dieser Arbeit entwickelte Modell als Datenlieferant für solche Verfahren dienen.

2.8 Häufigkeitsverteilungen

In den späteren Betrachtungen werden Häufigkeitsverteilungen von Zeiten genutzt. „Unter einer Häufigkeitsverteilung versteht man allgemein die Zuordnung von (absoluten oder relativen oder auch kumulierten) Häufigkeiten zu den Merkmalsausprägungen a_1, \dots, a_k .“ [BBK08, S. 12] Hier werden relative Häufigkeiten konkreten Werten für einen Zeitwert (z. B. Verarbeitungszeit) zugeordnet (siehe Abbildung 37).

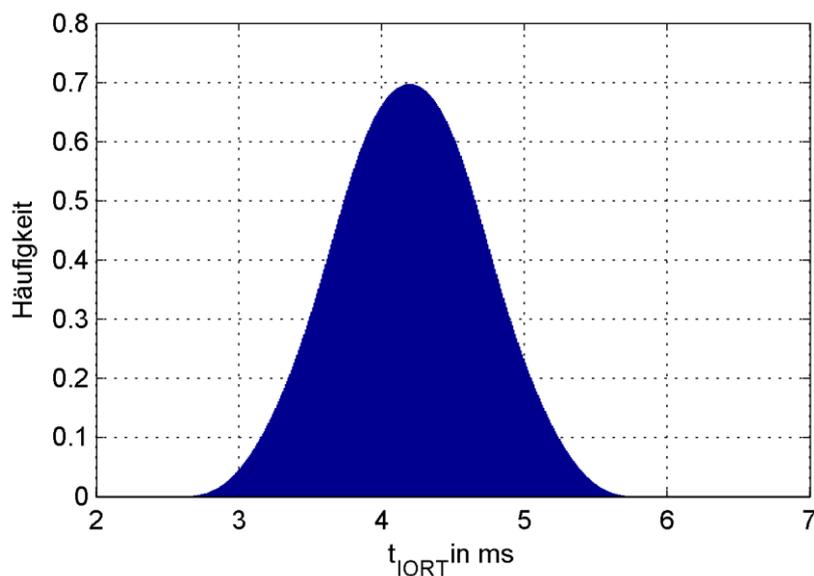


Abbildung 37: Beispieldarstellung einer Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit

Häufigkeitsverteilungen werden anhand ihrer Form klassifiziert. Folgende Verteilungsformen werden hier genauer betrachtet:

- Gleichverteilung, auch Rechteckverteilung genannt
- Dirac-Verteilung
- Diskrete Gleichverteilung

Für alle drei Verteilungsformen kann eine analytische Beschreibung der Wahrscheinlichkeitsdichtefunktion angegeben werden, diese wird für die spätere Berechnung zusätzlich Laplace-transformiert.

2.8.1 Rechteckverteilung

In dieser Verteilung haben alle Werte im Intervall $[a; a + T)$ die gleiche Häufigkeitsdichte $\frac{1}{T}$. In Abbildung 38 ist ein Beispiel dargestellt.

$$h(t) = \begin{cases} 1/T & \text{für } a \leq t < a + T \\ 0 & \text{sonst} \end{cases} \quad (16)$$

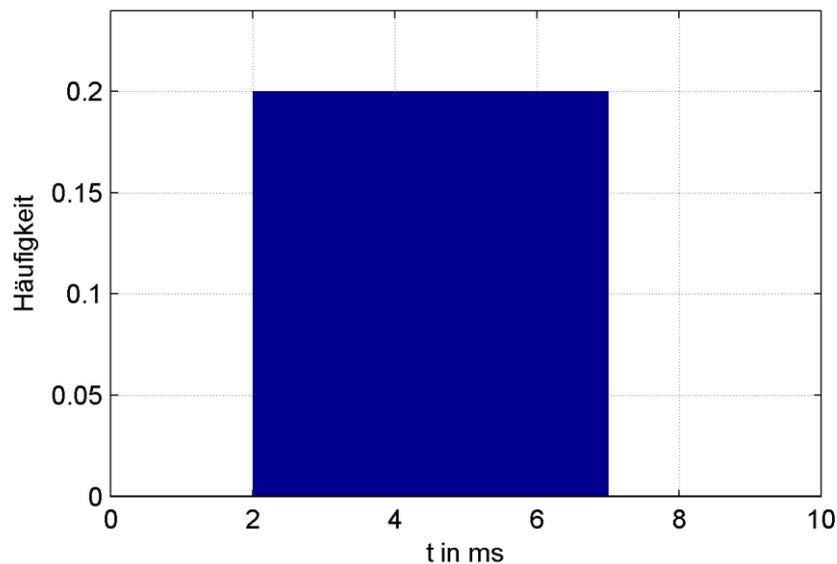


Abbildung 38: Beispiel für Rechteckverteilung ($a=2$ ms; $T=5$ ms)

Analytisch kann die Rechteckverteilung als Zusammensetzung aus zwei Heaviside-Sprungfunktionen beschrieben werden. Die Heaviside-Sprungfunktion ist nach [BS05, S. 737] definiert als¹⁸:

$$\Theta(t - t_0) = \begin{cases} 1 & \text{für } t \geq t_0 \\ 0 & \text{für } t < t_0 \end{cases} \quad (17)$$

Damit ergibt sich für die Wahrscheinlichkeitsdichtefunktion der Rechteckverteilung:

$$h(t) = \frac{1}{T} [\Theta(t - a) - \Theta(t - (a + T))] \quad (18)$$

Die Laplace-Transformation der Wahrscheinlichkeitsdichtefunktion der Rechteckverteilung liefert:

$$\mathcal{L}\{h(t)\} = \frac{e^{-as} - e^{-(a+T)s}}{Ts} \quad (19)$$

¹⁸ In der genannten Quelle ist die Funktion für $t > t_0$ als 1 definiert. Dies ergibt jedoch keine vollständige Definition, da die Funktion für $t = t_0$ undefiniert wäre. Daher wird hier die an verschiedenen Stellen in der Fachliteratur genutzte Definition mit $t \geq t_0$ verwendet.

2.8.2 Dirac-Verteilung

Die Dirac-Verteilung ist definiert als:

$$h(t) = \begin{cases} 1 & \text{für } t = t_0 \\ 0 & \text{sonst} \end{cases} \quad (20)$$

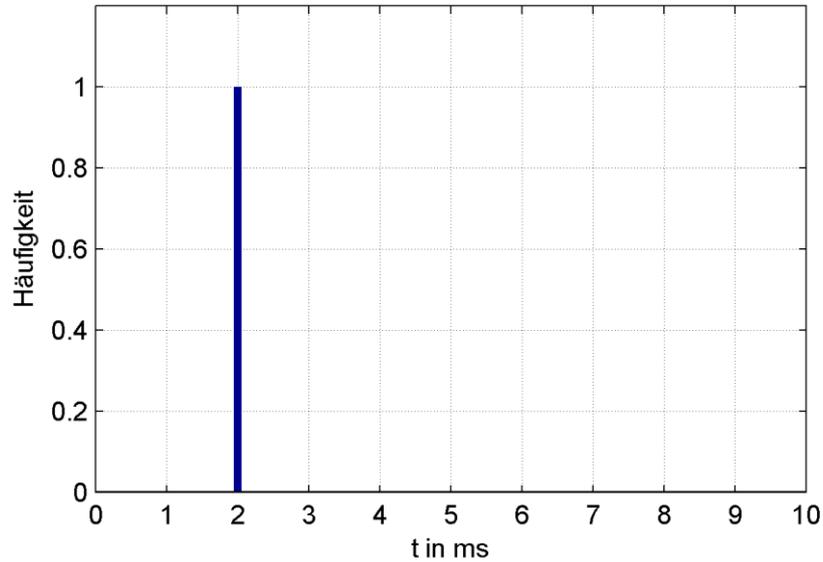


Abbildung 39: Beispiel für Dirac-Verteilung ($t_0=2$ ms)

Zur Beschreibung wird die Dirac'sche Delta-Funktion [BS05, S. 737] genutzt, welche wiederum von der Sprungfunktion abgeleitet ist:

$$\delta(t - t_0) = \lim_{T \rightarrow 0} \frac{1}{T} [\Theta(t - t_0) - u(t - t_0 - T)] \quad (21)$$

Damit ergibt sich die analytische Beschreibung der Wahrscheinlichkeitsdichtefunktion der Dirac-Verteilung:

$$h(t) = \delta(t - t_0) \quad (22)$$

Die Laplace-Transformation der Wahrscheinlichkeitsdichtefunktion der Dirac-Verteilung lautet:

$$\mathcal{L}\{h(t)\} = e^{-t_0 s} \quad (23)$$

2.8.3 Diskrete Gleichverteilung

Eine diskrete Gleichverteilung kann als Überlagerung von mehreren Dirac-Funktionen angesehen werden, deren Häufigkeit auf die Anzahl der Funktionen skaliert wird. Es wird davon ausgegangen, dass für die Verzögerungszeit die Zeiten t_i mit $i = 1 \dots n$ auftreten können.

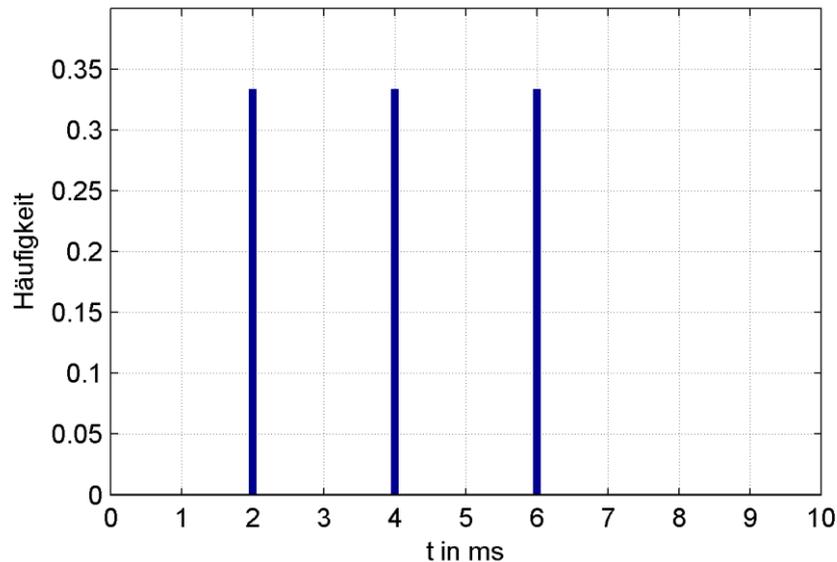


Abbildung 40: Beispiel für diskrete Gleichverteilung ($t_i \in \{2 \text{ ms}; 4 \text{ ms}; 6 \text{ ms}\}$)

Durch Überlagerung von n Funktionen nach Gleichung (22) ergibt sich die Wahrscheinlichkeitsdichtefunktion der Häufigkeitsverteilung zu:

$$h(t) = \frac{1}{n} \cdot \sum_{i=1}^n \delta(t - t_i) \quad (24)$$

Die Laplace-Transformation dieser Funktionen ergibt sich aus Gleichung (23) unter Nutzung des Additionssatzes¹⁹:

$$\mathcal{L}\{h(t)\} = \frac{1}{n} \cdot \sum_{i=1}^n e^{-t_i s} \quad (25)$$

¹⁹ „Die LAPLACE-Transformation einer Summe ist gleich der Summe der LAPLACE-Transformierten, wobei konstante Faktoren vor das LAPLACE-Integral gezogen werden können“ [BS05, S. 734]

2.9 Fazit

Wie in Kapitel 2.3 beschrieben, führen kürzere Zykluszeiten und höhere zeitliche Präzision von industriellen Kommunikationssystem zu erhöhtem Aufwand und damit Kosten, welche es zu minimieren gilt. Gleiches gilt für die Steuerungen, bei denen der steigende Rechenaufwand durch das häufigere Abarbeiten einer Steuerungsaufgabe bei kürzeren Zykluszeiten ein Kostenfaktor ist.

Daraus ergibt sich die Notwendigkeit, für eine gegebene regelungs- und steuerungstechnische Aufgabenstellung ein Steuerungssystem in einer Konfiguration auszuwählen, welches die Anforderungen erfüllt ohne zu viele Reserven bereitzustellen.

Für die Bewertung von Regelgüte und Stabilität mittels der Theorie der NCS ist die genaue Kenntnis der Verteilung der Klemme-Klemme-Reaktionszeit eines Steuerungssystems erforderlich, da neben den Verzögerungen durch das Kommunikationssystem auch die restlichen Komponenten (Steuerungsprogramm, EA-Module, ...) Einfluss auf das Zeitverhalten haben. Das typische, einschleifige NCS-Modell berücksichtigt nur einen Abtastvorgang, weshalb asynchron arbeitende Systeme nicht direkt modelliert werden können. Durch Bestimmung der Häufigkeitsverteilung der Gesamtverzögerung asynchroner Prozesse genügt das einschleifige NCS-Modell für die regelungstechnische Betrachtung.

In der Literatur finden sich vielfältige Performance-Vergleiche und Rechenmodelle für industrielle Kommunikationssysteme, jedoch muss für eine Aussage zur Eignung für einen spezifischen Aufbau neben dem Kommunikationssystem auch die Architektur des Steuerungssystems berücksichtigt werden, wie in Abschnitt 2.3.4 beispielhaft gezeigt. Viele Netzwerk-Simulationsmodelle liefern aufgrund ihrer detaillierten Abbildung des Kommunikationssystems sehr genaue Ergebnisse. Diese gehen jedoch mit einem hohen Rechenaufwand einher, welcher für die Betrachtung des geschlossenen Regelkreises ungünstig ist. Außerdem werden in diesen Modellen meist nur die Komponenten des Kommunikationssystems abgebildet, weitere Prozesse (z. B. Wandlung von Ein- und Ausgangssignalen) werden nicht betrachtet.

Der Fokus dieser Arbeit liegt daher in einem Modell zur Beschreibung des Zeitverhaltens eines industriellen Steuerungssystems, aus dem das Ende-zu-Ende-Zeitverhalten für verschiedene Signalpfade analytisch abgeleitet und berechnet werden kann. Der Schwerpunkt liegt dabei auf zyklischen Steuerungssystemen mit kabelgebundenen Industrial-Ethernet-basierten Kommunikationssystemen. Dieses Ende-zu-Ende-Zeitverhalten kann anschließend zur Bewertung oder zum Entwurf von Regelungsaufgaben genutzt werden.

3 Kenngrößen von verteilten Steuerungssystemen

Am Markt existiert eine Vielzahl von industriellen Kommunikationssystemen (siehe Kapitel 2.3). Um einen objektiven Vergleich von Eigenschaften und Leistungsdaten der Systeme zu ermöglichen, ist die Definition von Kenngrößen erforderlich, welche unabhängig von einer konkreten Produktrealisierung sind. Im Folgenden werden solche Kenngrößen basierend auf dem Modell eines dezentralen Steuerungssystems, bestehend aus Steuerung und Feldgerät (siehe Abbildung 41), definiert. In Abbildung 41 wird zur Vereinfachung der Darstellung für Dateneingabe und Datenausgabe das gleiche Feldgerät verwendet. Dies stellt keine Einschränkung dar und dient nur der Übersichtlichkeit. Die Definition ist für eine Architektur mit getrennten Ein- und Ausgabe-Feldgerät übertragbar. Die hier vorgestellten Kenngrößen basieren auf verschiedenen Vorarbeiten: [WG09], [RHG07a], [RHG07b], [ifa11] und wurden teilweise schon in [HDD+13] veröffentlicht.

Die Kenngrößen werden im Folgenden im Kontext industrieller, Ethernet-basierter Kommunikationssysteme erläutert, sind aber auf andere Systeme übertragbar. In der Literatur finden sich teilweise abweichende Definitionen, speziell in Hinsicht auf die Bemessungspunkte der Kenngrößen (z. B. Beginn eines Telegramms beim Rahmen oder bei den Nutzdaten). Aus den bekannten Größen wird hier eine Auswahl getroffen und diese für den Anwendungsfall präzisiert.

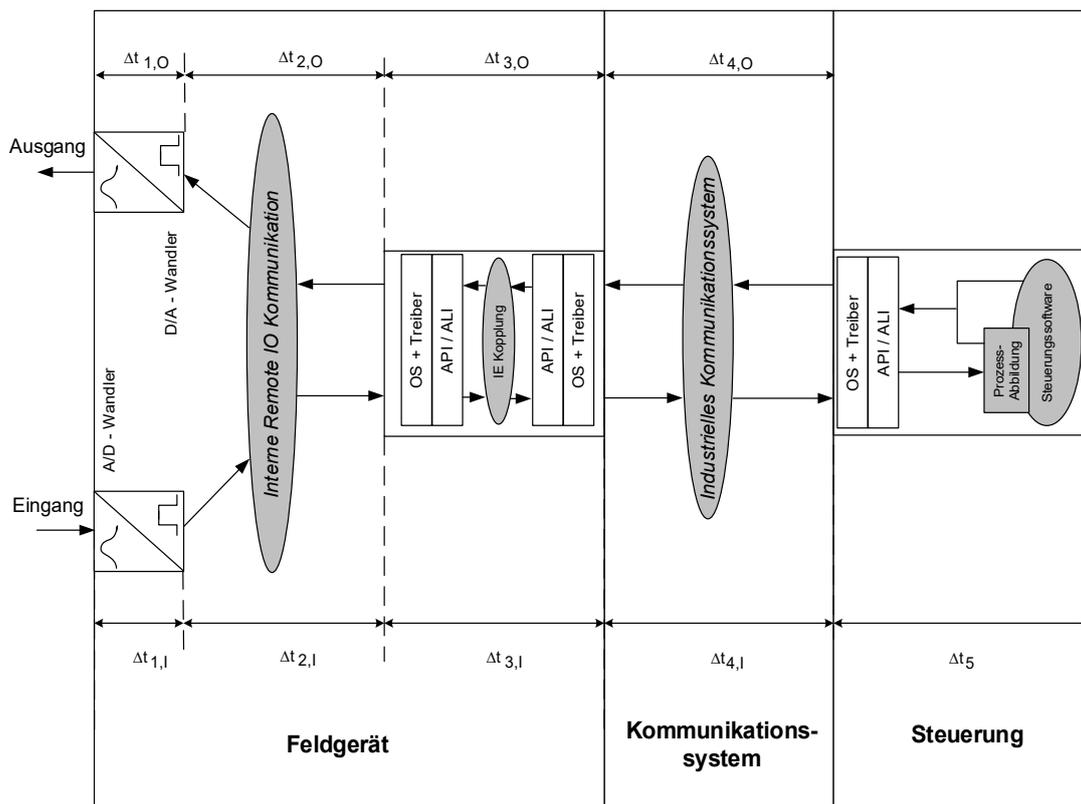


Abbildung 41: Modell eines dezentralen Steuerungssystems mit Kenngrößen nach [HDD+13]

In Abbildung 41 ist der Weg eines analogen Signals durch das Steuerungssystem dargestellt und mit entsprechenden Zeiten beschrieben. Am Eingang wird das physikalische Signal in ein digitales Datum umgewandelt, gefiltert und skaliert ($\Delta t_{1,1}$). Bei modularen Feldgeräten erfolgt anschließend ein interner Datentransport zur Kopfstation ($\Delta t_{2,1}$). Dieser Schritt entfällt bei kompakten Feldgeräten. In

der Kopfstation werden die Daten verarbeitet und auf dem Kommunikationsmedium bereitgestellt ($\Delta t_{3,1}$). Es folgt der Transport über das Kommunikationssystem zur Steuerung ($\Delta t_{4,1}$) und die Verarbeitung in der Steuerung (Δt_5). Anschließend wird dieser Weg umgekehrt durchlaufen, mit den Parametern $\Delta t_{4,0}$, $\Delta t_{3,0}$, $\Delta t_{2,0}$. Zuletzt erfolgt die Digital-Analog-Wandlung und Ausgabe als physikalisches Signal ($\Delta t_{1,0}$) an den Aktor.

3.1 Klemme-Klemme-Reaktionszeit (IO Response Time) t_{IORT}

Die Zeitdauer für einen kompletten Systemdurchlauf, vom Sensor zum Aktor, wird als Klemme-Klemme-Reaktionszeit definiert [VDI07, WG09] und ist die Summe der einzelnen Teilzeiten:

$$t_{\text{IORT}} = \sum_{i=1}^4 \Delta t_{i,I} + \Delta t_5 + \sum_{i=1}^4 \Delta t_{i,O} \quad (26)$$

Diese Größe eignet sich sehr gut, um das von außen beobachtbare zeitliche Verhalten eines Steuerungssystems ohne detaillierte Kenntnisse der inneren Abläufe zu beschreiben. Sie beschreibt aus Sicht des Anwendungsprozesses „die schnellstmögliche Reaktion eines Reglers / Steuerungsalgorithmus auf ein Ereignis“ [HD12].

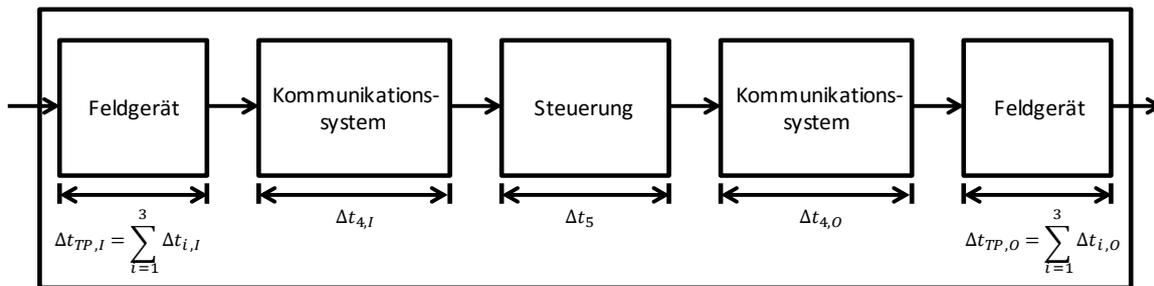


Abbildung 42: Klemme-Klemme-Reaktionszeit aus systemtheoretischer Sichtweise nach [HDD+13]

Wie in Abbildung 42 dargestellt, beschreibt die Klemme-Klemme-Reaktionszeit die Zeit entlang eines Signalpfades durch das Steuerungssystem und beinhaltet dabei die Zeiten für Signalwandlung an Ein- und Ausgängen, Übertragung der Daten über ein Kommunikationssystem und Verarbeitung in der Steuerung. Aus Sicht des Anwenders hat diese Kenngröße große Relevanz, denn sie bildet die Dynamik eines Signalpfades des Steuerungssystems und damit auch die Dynamik der damit zu steuernden bzw. regelnden Anlage ab.

3.2 Framelänge Δt_{Frame}

Ein Ethernet-Telegramm (auch Ethernet-Frame genannt) setzt sich zusammen aus den Nutzdaten und dem Ethernet-Rahmen. Dies ist in Abbildung 43 dargestellt. Zusätzlich muss nochmal zwischen dem Ethernet-Telegramm und dem Ethernet-Paket unterschieden werden. Letzteres enthält zusätzlich die Präambel und den Start Frame Delimiter (SFD).

Die Kenngröße Framelänge stellt die Zeitdauer für die Übertragung eines bestimmten Ethernet-Telegramms auf einem Übertragungsmedium dar. Sie wird durch die Anzahl der Bytes im Telegramm l_{Frame} und die Übertragungsrate des Mediums C bestimmt:

$$\Delta t_{\text{Frame}} = C \cdot l_{\text{Frame}} \quad (27)$$

Typische Werte sind $C = 80 \text{ ns/Byte}$ für Fast Ethernet (100 MBit/s) und $C = 8 \text{ ns/Byte}$ für Gigabit-Ethernet (1 GBit/s).

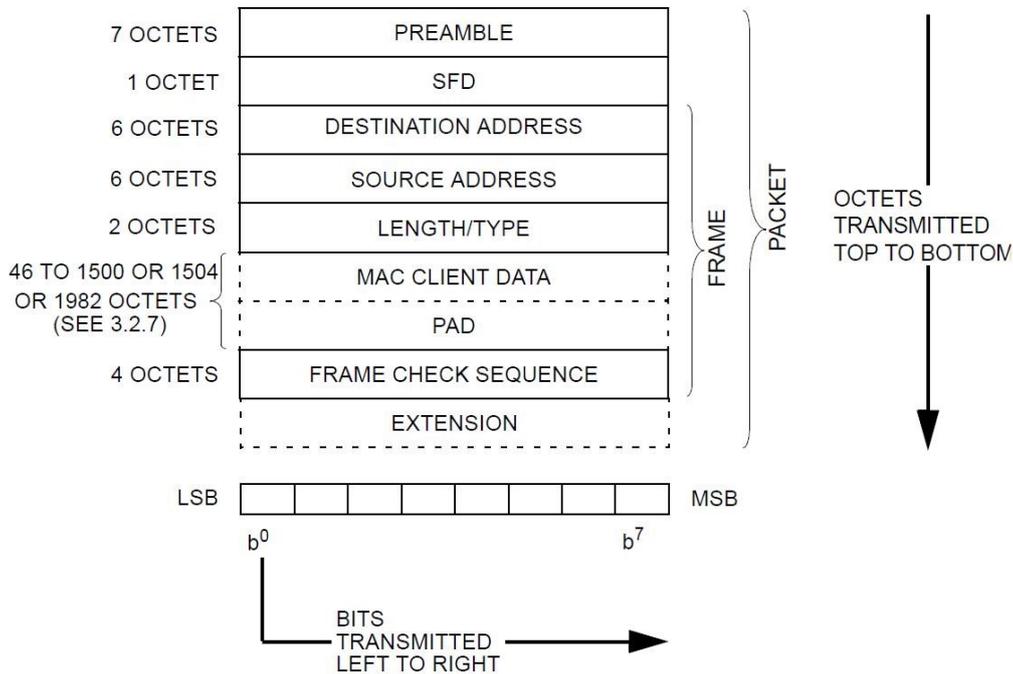


Abbildung 43: Aufbau eines Ethernet-Pakets [IEE12, S. 53]

3.3 Weiterleitungsdauer (Bridge Delay) Δt_{BD}

Die Weiterleitungsdauer (Bridge Delay) ist die Zeit, die ein Netzwerkteilnehmer benötigt, um ein ankommendes Telegramm von einem Port zu einem anderen weiterzuleiten. Die Weiterleitungsdauer wird gemessen vom Eintreffen des ersten Bytes eines Telegramms am Eingangsport bis zur Ausgabe des ersten Bytes am Ausgangsport (Telegrammbeginn bis Telegrammbeginn), die Grenze ist das Media Independent Interface (MII). Die Weiterleitungsdauer wird nur für Netzwerkteilnehmer angegeben, welche das Telegramm auf Kommunikationsebene von einem Port zu einem anderen übertragen (z. B. Switches). Findet dabei eine Verarbeitung der Nutzdaten statt, handelt es sich um die Verzögerungszeit der Steuerung (siehe Abschnitt 3.9).

$$\Delta t_{BD} = f(t, \text{Stack}) \quad (28)$$

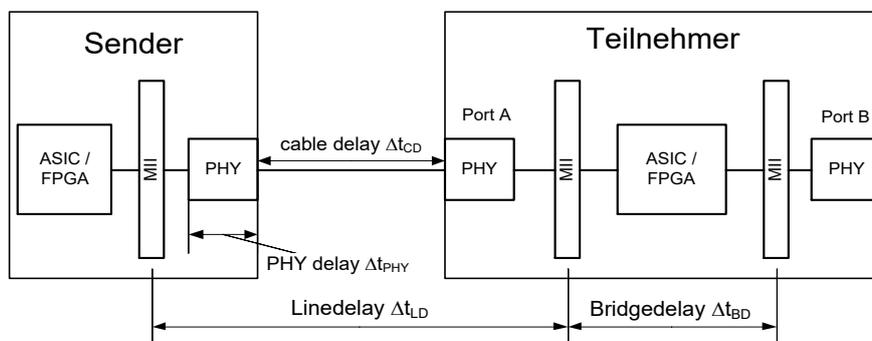


Abbildung 44: Bridge Delay und Line Delay [DLK+14]

Die Weiterleitungsdauer wird von der Hardwarerealisierung bestimmt. Es gibt aber einen prinzipbedingten unteren Grenzwert. Der Teilnehmer muss für die Entscheidung, auf welchem Port ein Telegramm ausgegeben wird, die Ziel-MAC-Adresse aus dem Telegramm lesen. Dazu müssen der Rahmen (Präambel und Start Frame Delimiter) sowie die Destination Address gelesen werden. Der Teilnehmer kann damit frühestens nach 14 Byte entscheiden, auf welchem Port das Telegramm ausgegeben wird. Abhängig vom Kommunikationsprotokoll müssen mehr Daten gelesen werden, z. B. die Frame ID bei PROFINET.

3.4 Line Delay Δt_{LD}

Die Line Delay gibt die Telegrammverzögerung durch das physikalische Übertragungsmedium wieder. Sie wird bestimmt von der Geschwindigkeit ϑ , mit der sich Informationen auf dem jeweiligen Medium ausbreiten können und der Länge des Übertragungsweges l . Neben der Signallaufzeit gehen auch die Verarbeitungszeit des PHY²⁰ beim Sender $\Delta t_{PHY,TX}$ und Empfänger $\Delta t_{PHY,RX}$ mit in die Line Delay ein.

$$\Delta t_{LD} = \vartheta \cdot l + \Delta t_{PHY,RX} + \Delta t_{PHY,TX} \quad (29)$$

Die Übertragungsgeschwindigkeit von Kupferleitern beträgt nach [IEE12, S. 350] :

$$\vartheta_{Cu} = 5,7 \text{ ns/m} \quad (30)$$

Damit ergibt sich für eine maximale Kabellänge von 100 m eine Verzögerung von 570 ns zzgl. der Verarbeitungsdauern der PHYs²¹.

3.5 Signallaufzeit (Propagation Delay) Δt_{PD}

Die Signallaufzeit ist die Summe der Weiterleitungsdauern und Line Delays eines Ethernet-Frames auf dem Weg vom Sender zum Empfänger über alle Kommunikationsteilnehmer.

$$\Delta t_{PD} = \sum (\Delta t_{BD} + \Delta t_{LD}) \quad (31)$$

Da die beiden Kenngrößen jeweils auf den Beginn des Telegramms definiert sind, hat die übertragene Datenmenge keinen Einfluss auf die Signallaufzeit. Die Kenngröße Signallaufzeit ist nur von den Geräten und der Kommunikationstopologie abhängig.

3.6 Übertragungszeit (Transmission Delay) Δt_{TD}

Die Übertragungszeit ist definiert als „Zeitspanne für die Übertragung eines Ethernet-Frames auf dem Medium zwischen 2 Geräten“ [WG09]. Die Zeit beginnt mit dem Senden des ersten Bytes des Frames am Sender und endet mit dem Empfangen des letzten Bytes am Empfänger. Sie beinhaltet sowohl die Signallaufzeit als auch die Framelänge Δt_{Frame} :

$$\Delta t_{TD} = \Delta t_{PD} + \Delta t_{Frame} = \sum \Delta t_{BD} + \sum \Delta t_{LD} + \Delta t_{Frame} \quad (32)$$

²⁰ PHY ist ein Schaltkreis zur Übersetzung von digitalen Daten auf das modulierte analoge Signal des Übertragungsmediums, d. h. zur Umsetzung der physikalischen Schicht (physical layer) des ISO/OSI-Referenzmodell.

²¹ Texas Instruments gibt für den Baustein TLK105 die Verarbeitungsdauer des PHY beim Senden mit 48ns und beim Empfangen mit 190 ns an [Tex14]

Die Übertragungszeit entspricht der Zeit $\Delta t_{4,I}$ bzw. $\Delta t_{4,O}$ in Abbildung 41. Sie wird oft zur Leitungsbewertung von industriellen Kommunikationssystemen genutzt, da sie alle Aspekte des Kommunikationssystems enthält (z. B. bei [Jas02]). Die Ermittlung der Übertragungszeit erfolgt nach dem Producer-Consumer-Modell, unabhängig vom Transportmodell des betrachteten Kommunikationssystems.

Die hier genutzte Definition setzt die Grenzen der Übertragungszeit im MII, d. h. die Verzögerungen durch die PHY im Sender und Empfänger werden eingeschlossen. Eine Messung an diesen Punkten ist anspruchsvoll, da die Schnittstelle bei realen Geräte nicht zugänglich ist. Alternativ wird häufig das Senden und Empfangen eines Telegramms auf dem Ethernet-Kabel mit Hilfe von TAPs²² gemessen. Für die praktische Nutzung der Kenngröße muss entweder die Verzögerung der PHYs anhand des Datenblattes addiert oder die Kenngröße wird explizit als „gemessen ohne PHY“ gekennzeichnet werden. In letzterem Fall ist die Vergleichbarkeit nur eingeschränkt gegeben.

Nach [RHG07a] fließen in die Ermittlung der Übertragungszeit nur fehlerfrei übermittelte Pakete ein, es erfolgt keine Betrachtung von Paketverlusten.

3.7 Übergangszeit (Transition Period) Δt_{TP}

Die Übertragungszeit ist für Eingabe- und Ausgaberrichtung unterschiedlich definiert. Bei Eingängen ist die Übergangszeit $\Delta t_{TP,I}$ die Zeitspanne, die benötigt wird, um einen Wert am Eingang des Feldgerätes zu erfassen, zu wandeln und für die Übertragung auf dem Industrial Ethernet bereitzustellen. Bei Ausgängen entspricht dies der Zeitspanne $\Delta t_{TP,O}$, um Daten von der Schnittstelle zum Industrial Ethernet (PHY) zu empfangen, für die Wandlung bereitzustellen und am Ausgang des Feldgerätes auszugeben. In Abbildung 41 entspricht die Übergangszeit der Summe der Zeiten im jeweiligen Feldgerät:

$$\Delta t_{TP,I} = \Delta t_{1,I} + \Delta t_{2,I} + \Delta t_{3,I} \quad (33)$$

$$\Delta t_{TP,O} = \Delta t_{1,O} + \Delta t_{2,O} + \Delta t_{3,O} \quad (34)$$

In Abbildung 45 ist die Übergangszeit beispielhaft für analoge Ein- und Ausgabe dargestellt.

Die Grenze zum Kommunikationssystem wird wie bei der Definition der Übertragungszeit beim MII gesetzt. Während sich die bisher betrachteten Kenngrößen auf das Kommunikationssystem bezogen, beschreibt diese Kenngröße die Eigenschaften der eingesetzten Feldgeräte. Der Einfluss des Kommunikationssystems ist nur untergeordnet in Form der Durchlaufzeit des Kommunikationsstacks enthalten.

²² TAP = Test Access Point

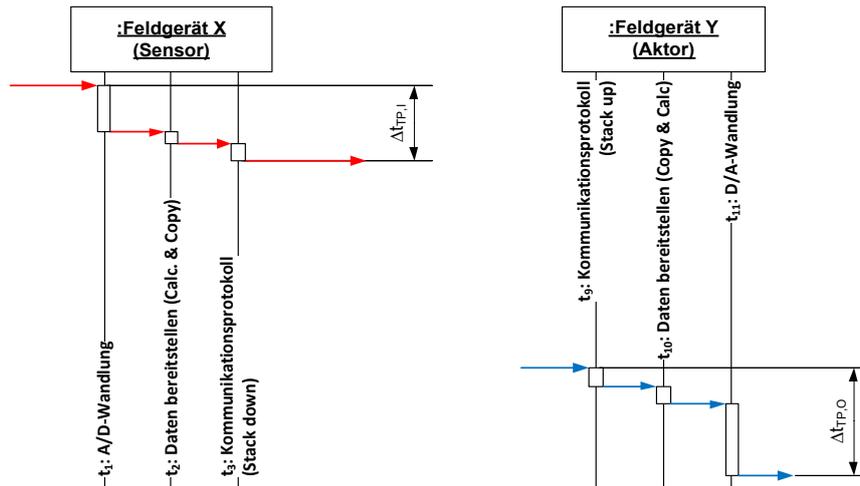


Abbildung 45: Übergangszeit Δt_{TP} [DLK+14]

3.8 Zykluszeit

Die IEC 61800-7-1 definiert den Begriff *cycle time* als „time span between two consecutive cyclically recurring events“. [IEC07] Übersetzt bedeutet der Begriff Zykluszeit die Zeit zwischen zwei aufeinanderfolgenden Anfängen eines zyklischen Vorgangs. Im Umfeld von industriellen Steuerungssystemen kann dieser Begriff an verschiedenen Stellen verwendet werden.

Die Zykluszeit eines Kommunikationssystems gibt die Dauer zwischen zwei aufeinanderfolgenden Paketen eines Übertragungsweges an, gemessen vom Paketanfang zum Paketanfang. Werner et al. definieren dabei die Ankunft der Pakete an der Ethernet-Schnittstelle des Endgerätes als relevanten Messpunkt. [WG09]

Die Zykluszeit einer Steuerung gibt die Zeitdauer zwischen zwei aufeinanderfolgenden Durchläufen des Steuerungsprogramms an, gemessen von Anfang zu Anfang. Sie kann sinnvollerweise nur für zyklisch arbeitende Steuerungen (z. B. SPS) angegeben werden.

3.9 Verzögerungszeit der Steuerung (PLC time) Δt_{PLC}

Die Verzögerungszeit der Steuerung gibt an, wie viel Zeit vom Empfangen von Daten (via Kommunikationssystem) bis zum Versenden der Reaktion vergeht. Bemessungsgrenze ist wieder das MII der Ethernet-Schnittstelle. In Abbildung 41 entspricht die Verzögerungszeit der Steuerung der Zeit Δt_5 .

In die Verzögerungszeit der Steuerung gehen die Durchlaufzeiten des Kommunikationsstacks und die Verarbeitungszeit des eigentlichen Steuerungsprogramms ein. Zusätzlich sind auch Wartezeiten zwischen den Prozessen enthalten. Abhängig vom Zeitmodell des Steuerungs- und Kommunikationssystems kann zwischen der Abarbeitung des Steuerungsprogramms und dem Versenden der Antwort eine geplante Wartezeit liegen. Dies ist z. B. bei PROFINET IRT der Fall, da hier die Kommunikation in Ein- und Ausgangsrichtung gleichzeitig abläuft, somit wird die Verarbeitungszeit des Steuerungsprogramms auf einen Zyklus aufgerundet. Ähnliches gilt auch für die meisten EtherCAT-basierten Systeme. Bei asynchron arbeitenden Prozessen können stochastische Wartezeiten auftreten, wodurch die Verzögerungszeit der Steuerung keinen konstanten Wert hat, sondern als Häufigkeitsverteilung angegeben wird.

3.10 Echtzeit-Datendurchsatz

Der Echtzeitdatendurchsatz gibt die Anzahl der Nutzbytes der Prozessdaten-Kommunikation (i. d. R. die zyklische Kommunikation) an, die in einer Zeiteinheit vom Kommunikationssystem übertragen werden. Er stellt die Netto-Nutzdatenrate dar.

3.11 NRT-Bandbreite

Die Non-Realtime-Bandbreite (NRT-Bandbreite) stellt die Datenrate dar, welche zur Übertragung von Nicht-Echtzeitdaten zur Verfügung steht.

Ethernet-basierte Kommunikationssysteme bieten die Möglichkeit, neben den Echtzeitdaten weitere Pakete zu übertragen. Dies macht den Einsatz von Webservern auf Feldgeräte oder die Übertragung von Videostreams möglich. Auch Diagnoseprotokolle wie SNMP²³ können eingesetzt werden.

Bei industriellen Kommunikationssystemen mit angepassten Buszugriffsverfahren (PROFINET IRT, EtherCAT, ...) können RT-Daten priorisiert werden und somit eine maximale verbleibende Bandbreite für NRT-Daten bereitgestellt werden. Die RFC2544 [Int99] beschreibt eine Methode zur Messung der erreichbaren Datenrate.

In industriellen Kommunikationssystemen, welche konform zum Ethernet-Standard übertragen, kann eine zu hohe Bandbreite von NRT-Daten zu Kommunikationsausfällen bei der Echtzeitübertragung führen. Daher ist die Messung nach RFC2544 nicht empfohlen. Sie sollte laut RFC6815 [Int12] nur unter Laborbedingungen durchgeführt werden. Industrial-Ethernet-Systeme, welche den Ethernet-Standard durch Scheduling-Mechanismen auf Layer 2 erweitern (vgl. Klasse 3 in Abbildung 13), sind prinzipiell immun gegen solche Ausfälle, da die Echtzeitdaten hier wirksam priorisiert werden können. Dies gilt z. B. für PROFINET IRT oder EtherCAT.

3.12 Fazit

In diesem Kapitel wurden Kenngrößen für verteilte industrielle Steuerungssysteme definiert. Diese ermöglichen einen objektiven Vergleich der verfügbaren Lösungen. Die eindeutige Definition der Kenngrößen ist wichtig, da gleiche Begriffe in verschiedenen Systemen unterschiedliche Bedeutung haben und daher die entsprechenden Werte nicht vergleichbar sind. Nur durch eindeutige Definition von Kenngrößen und deren Bemessungsgrenzen kann eine Vergleichbarkeit erzielt werden. Die Kenngrößen werden im Folgenden bei der Beschreibung von Teilprozessen in Steuerungssystemen genutzt.

Für die Bewertung eines Steuerungssystems hinsichtlich seiner Eignung für Steuerungs- und Regelungssicht ist die Kenngröße Klemme-Klemme-Reaktionszeit entscheidend. Alle anderen Kenngrößen beschreiben das zeitlichen Verhalten von Teilsystemen des Steuerungssystems und beeinflussen damit die Klemme-Klemme-Reaktionszeit.

²³ SNMP – Simple Network Management Protocol [Int02]

4 Modellierung von dezentralen / verteilten Steuerungssystemen

4.1 Einführung

Ausgangspunkt der Modellierung ist die Betrachtung von synchronen und asynchronen Zyklen sowie die Auswirkungen auf Verarbeitungs- und Wartezeiten von Teilprozessen eines Steuerungssystems.

Bei einem industriellen Steuerungssystem ist zunächst davon auszugehen, dass jedes beteiligte Teilsystem eine eigene Zeitbasis hat. In Abbildung 46 ist dies prinzipiell für ein dezentrales Steuerungssystem dargestellt. Das Steuerungssystem ist somit eine Verkettung von Teilsystemen mit eigenem Zyklus. Diese Darstellung zeigt, dass bereits kleine Systeme über eine Vielzahl von Zyklen verfügen können, deren Auswirkungen im Folgenden untersucht werden.

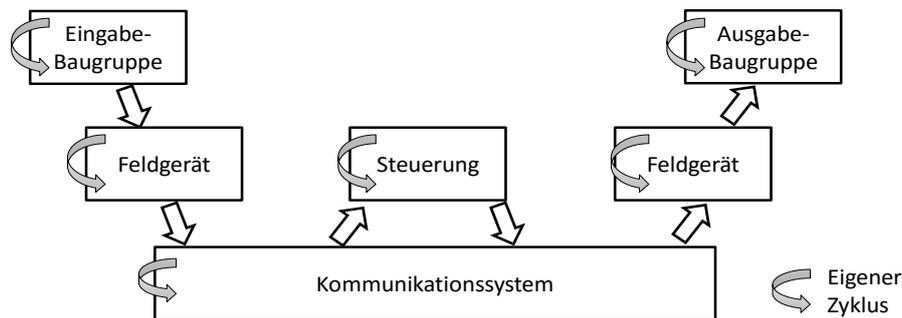


Abbildung 46: Prinzipielle Darstellung der Zyklen eines Steuerungssystems

4.1.1 Asynchrone Zyklen

Das Zeitverständnis eines technischen Systems basiert auf einem Taktgeber, der eine möglichst konstante Frequenz vorgibt. Durch das Zählen der Takte und die Kenntnis der Frequenz entsteht eine lokale Zeitbasis. Unter der Annahme, dass die Frequenzen aller Taktgeber konstant und exakt bekannt sind, können alle Zeitbasen als gleich angesehen werden. In der Realität ist die Frequenz der Taktgeber aber nicht genau bekannt. Die Bauteile werden mit einer Nennfrequenz und einer Genauigkeit verkauft. Zum anderen kann sie auch nicht als konstant angenommen werden, da durch äußere Einflüsse (Temperatur, Vibration, ...) und Alterung eine Änderung der Frequenz über die Lebenszeit des Bauteils erfolgen kann. Somit kann die Zeitbasis eines technischen Gerätes nicht als identisch zur globalen Zeit angenommen werden. Die Konsequenz daraus ist, dass die Zeitbasen von zwei verschiedenen Geräten (ohne zusätzliche technische Maßnahmen – siehe Abschnitt 2.4) nicht als gleich angenommen werden können.

Zyklisch arbeitende Geräte erzeugen ihren Zyklus basierend auf der eigenen Zeitbasis. Werden zwei Geräte auf die gleiche Zykluszeit parametrisiert, so ist aufgrund der Tatsache, dass ihre Zeitbasen einer Streuung unterliegen, davon auszugehen, dass die Zyklen nicht identisch sind (siehe Abbildung 47). Allgemein betrachtet entsteht dieser Effekt nicht nur zwischen Geräten, sondern zwischen Teilsystemen mit verschiedenen Zeitbasen. Er kann z. B. auch innerhalb eines Gerätes auftreten, wenn mehrere Prozessoren mit eigenen Taktgebern verbaut werden. Wenn zwei Zyklen auf unterschiedlichen, nichtsynchronisierten Zeitbasen basieren, werden sie als **asynchrone Zyklen** bezeichnet.

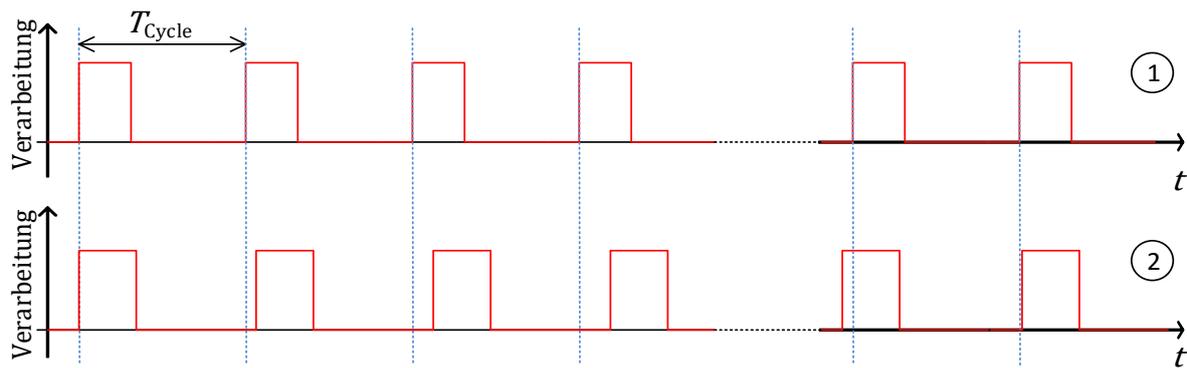


Abbildung 47: Entstehung von asynchronen Zyklen durch abweichende Taktgeber

Die Ursache der asynchronen Zyklen liegt in der Abweichung der lokalen Zeitbasen, verursacht durch Streuungen der Frequenz der Taktgeber. Es gibt zwei Möglichkeiten, diese zu verhindern:

1. Nutzung des gleichen Taktgebers für mehrere Teilsysteme
2. Synchronisation der Taktgeber der Teilsysteme

Die Nutzung des gleichen Taktgebers ist innerhalb eines Gerätes durch das Betreiben von mehreren Bauelementen mit dem gleichen Taktgeber möglich. Da jedoch verschiedene Bauelemente häufig verschiedene Frequenzen benötigen, geht hiermit ein erhöhter Schaltungsaufwand einher (Takteiler, Routing, ...). Wenn mehrere Geräte den gleichen Taktgeber nutzen sollen, können Taktleitungen eingesetzt werden. Verschiedene Kommunikationssysteme sehen eine separate Taktleitung vor, um die Asynchronität der Kommunikationspartner zu kompensieren (I²C, SPI, ...). Diese werden jedoch kaum als Taktgeber für das Gerät selbst verwendet. Ethernet sieht keine Taktleitung vor.

Die zweite Möglichkeit der Synchronisation der Teilsysteme mittels Synchronisationsverfahren wird vielfältig eingesetzt, die möglichen Verfahren wurden in Abschnitt 2.4 vorgestellt.

Die Abweichung der Zeitbasen und damit der Zyklen muss unterschieden werden in eine zeitlich begrenzte Abweichung und eine globale Drift. Eine **globale Drift** bedeutet, dass der Mittelwert der Zykluszeiten über einen längeren Zeitraum unterschiedlich ist, wodurch der Effekt der asynchronen Zyklen entsteht.

Eine **kurzzeitige Abweichung** der Zykluszeiten führt demgegenüber nicht zwingend zu einer Asynchronität. Die Abweichung kann durch das Vorhalten einer Sicherheitsreserve bei der Datenübergabe zwischen den Systemen ausgeglichen werden. Wenn technische Systeme mit Synchronisationsverfahren zu einer (lokalen) Zeitdomäne zusammengefasst werden, so ist diese Synchronisation nur mit einer bestimmten Präzision möglich. Dieser Fehler muss bei der Wahl der Parametrierung des konkreten Systems durch das Einfügen von Wartezeiten ausgeglichen werden. Die Dauer der Wartezeit muss größer als der maximale Fehler zwischen den lokalen Uhren der Teilsysteme (deren Präzision) sein.

Werden Informationen zwischen asynchron arbeitenden Teilsystemen ausgetauscht, wird im Folgenden von einem **asynchronen Zyklusübergang** gesprochen. Dabei entstehen stochastische Wartezeiten. Im Interesse einer Optimierung der Signallaufzeiten gilt es, diese Wartezeiten zu minimieren. Aber auch bei synchron arbeitenden Teilsystemen können Wartezeiten auftreten, um eine zeitliche Unsicherheit durch kurzfristige Abweichungen auszugleichen.

4.1.2 Verarbeitungs- und Wartezeiten

Die einzelnen Zeitelemente, welche beim Durchlauf des Steuerungssystems auftreten, lassen sich in zwei Arten unterteilen:

- Abarbeitungszeiten
- Wartezeiten

Als Abarbeitungszeiten sind in diesem Zusammenhang Anteile zu sehen, in welchen die Daten aktiv von einem Teilsystem verarbeitet werden. Dies kann beispielsweise die Analog-Digital-Wandlung in einer Eingangsbaugruppe, die Übertragungszeit auf einem Ethernet-Kabel oder die Abarbeitungszeit des SPS-Programmes sein. Diesen Anteilen ist gemeinsam, dass Sie in der Regel für eine vorhandene Hardware nicht verändert werden können.

Wartezeiten entstehen im System immer dann, wenn ein Teilsystem Daten an das nächste Teilsystem übergibt, dessen Abarbeitung aber erst später beginnt. Die Wartezeiten können nochmal untergliedert werden in geplante und stochastische Wartezeiten. Geplante Wartezeiten werden bewusst ins System eingebaut, um ein gewünschtes Sollverhalten zu erreichen oder zeitliche Unsicherheiten zu kompensieren. Ein Beispiel für den gezielten Einsatz von Wartezeiten ist das Ausgeben von mehreren physikalischen Signalen, deren Wandlung unterschiedliche Zeitdauern in Anspruch nimmt (z. B. ein analoges und ein digitales Spannungssignal). Damit deren Wandlung gleichzeitig endet (Gleichzeitigkeit), muss vor dem schnelleren Vorgang eine geplante Wartezeit eingefügt werden. Arbeiten zwei Systeme synchronisiert mit einer bekannten Präzision, so muss bei der Datenübergabe zwischen den Systemen eine Wartezeit in Höhe der Präzision eingeplant werden, um eine korrekte Übergabe zu gewährleisten. Stochastische Wartezeiten treten im System wie in Abschnitt 4.1.1 beschrieben zwischen asynchronen Zyklen auf.

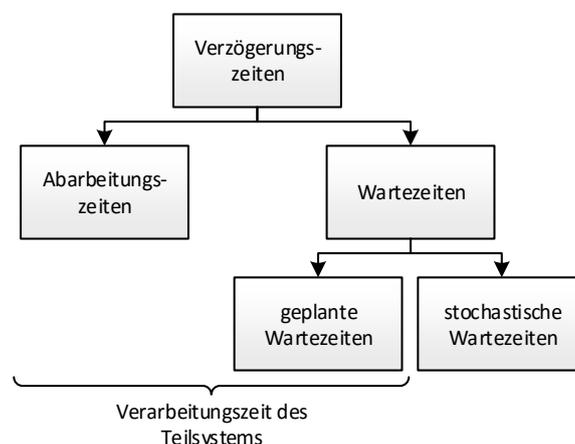


Abbildung 48: Gliederung der entlang eines Signalpfades auftretenden Verzögerungszeiten

Der im Rahmen dieser Arbeit genutzte Begriff der Verarbeitungszeit eines Teilsystems setzt sich aus der Abarbeitungszeit und eventuell eingefügten, geplanten Wartezeiten zusammen (Abbildung 48). Die geplanten Wartezeiten sind für das korrekte Systemverhalten notwendig und daher ebenso wie die Abarbeitungszeit zu behandeln.

In [LF07] schlagen die Autoren eine ähnliche Gliederung vor. Sie beschreiben noch eine dritte Art der Verzögerung (Warten auf die Verfügbarkeit einer Ressource). Diese ist auf den Schwerpunkt der Arbeit zur Wechselwirkung in NAS zurückzuführen.

4.1.3 Einlesen von kontinuierlichen und diskreten Signalen

Das Zeitverhalten der Abtastung von Signalen ist abhängig von der Art des Prozesssignals. Wie in [HD12] veröffentlicht, muss an einem solchen Übergang zwischen zustandskontinuierlichen oder zustandsdiskreten Signalen unterschieden werden.

Zustandskontinuierliche Signale (Abbildung 49) haben zu jedem Zeitpunkt einen aktuellen Wert, welcher erfasst werden kann. Wird das Signal zeitlich diskretisiert, so ist die auftretende Zeitverzögerung nur durch die interne Verarbeitungsdauer t_{process} (Wandlerzeiten, ...) des umwandelnden Systems bestimmt und kann damit als konstant angenommen werden. Entsprechend Abschnitt 4.1.1 kann der Übergang von einem zeitkontinuierlichen in ein zeitdiskretes Signal für zustandskontinuierliche Signale als synchroner Zyklusübergang betrachtet werden.

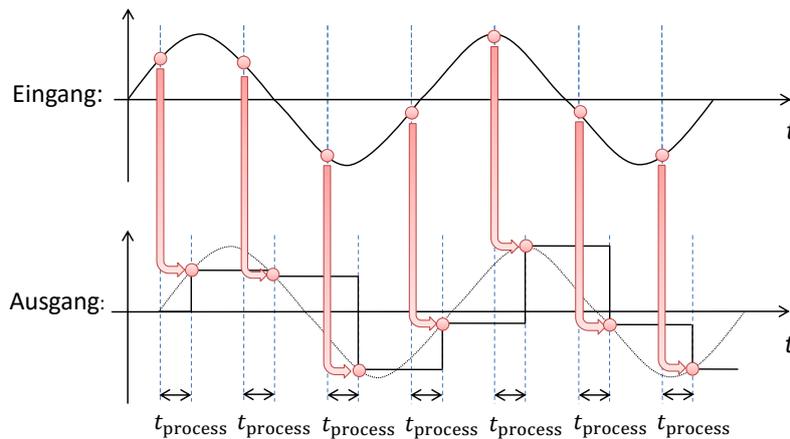


Abbildung 49: Zeitverhalten beim Einlesen zustandskontinuierlicher Signale (Prinzip-Skizze)

Bei einem zustandsdiskreten Signal (Abbildung 50) kann ein Zustandswechsel (Ereignis) zu jedem beliebigen Zeitpunkt zwischen zwei Abtastungen des Signals stattfinden. Er wird aber erst zum nächsten Abtastzeitpunkt erfasst. Dies bedeutet, dass die Zeitdauer für die Umwandlung des Signals nicht nur von der internen Verarbeitungsdauer t_{process} des umwandelnden Teilsystems abhängig ist. Zusätzlich tritt noch eine stochastische Wartezeit t_n auf, deren Wert zwischen Null (Ereignis tritt genau zum Zeitpunkt des Wandels auf) und der Zykluszeit T_{Cycle} des wandelnden Systems (Ereignis tritt exakt nach dem Beginn des Wandlungsvorgangs auf) liegt: $0 \leq t_n < T_{\text{Cycle}}$. Entsprechend Abschnitt 4.1.1 kann der Übergang von einem zeitkontinuierlichen in ein zeitdiskretes Signal für zustandsdiskrete Signale als asynchroner Zyklusübergang betrachtet werden.

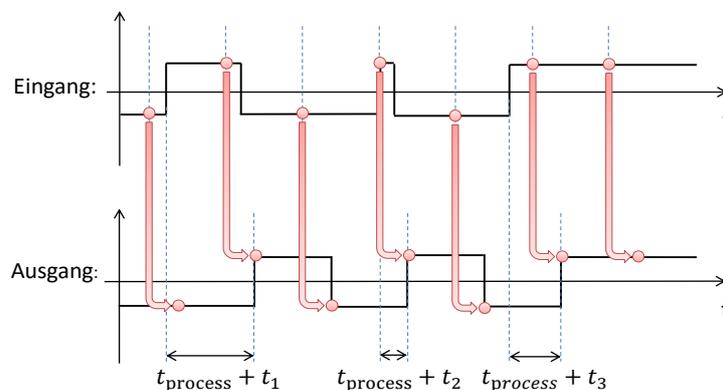


Abbildung 50: Zeitverhalten beim Einlesen zustandsdiskreter Signale (Prinzip-Skizze) [HD12]

4.2 Modell der asynchronen Prozesse

Im Rahmen dieser Arbeit soll ein mathematisches Modell entworfen und verifiziert werden, welches das zeitliche Verhalten (hier im Speziellen die Klemme-Klemme-Reaktionszeit) von zyklisch arbeitenden Steuerungssystemen abbildet. Es kann somit als Physik-zu-Physik-Modell verstanden werden, da die gesamte Verarbeitung zwischen den physikalischen Schnittstellen zum zu steuernden Prozess als Black-Box betrachtet werden kann²⁴. Die Komplexität des Modells soll dabei möglichst gering gehalten werden, um eine Ableitung des Zeitverhaltens ohne großen Rechen- und damit Zeitaufwand zu ermöglichen.

Das hier vorgeschlagene Modell bildet zwei verschiedene Aspekte des Systems ab, zum einen die Struktur des Steuerungssystems, welche als statischer Anteil betrachtet werden kann. Die Struktur umfasst neben den Informationen zu den einzelnen technischen Komponenten auch deren Verbindung untereinander. Ein zweiter Aspekt ist die Modellierung konkreter Signalpfade im System.

Zur Beschreibung des Modells werden ausgewählte Elemente der Beschreibungssprache UML²⁵ benutzt, welche über verschiedene Diagrammtypen zur Modellierung von statischen und dynamischen Aspekten eines Systems verfügt.

4.2.1 Statischer Modellanteil (Strukturmodell)

Die Abbildung der Modellstruktur erfolgt in einem Klassendiagramm, siehe Abbildung 51. Das Modell trennt die ausgeführten Funktionen und die ausführende Hardware. Dabei wird der Begriff des Prozesses nach Hadlich verwendet:

„Ein Prozess beschreibt die Ausführung von einer oder mehreren Funktionen eines Systems mit zeitlichem und räumlichem Bezug, mit Anfangs- und Endzustand der Systemumgebung.“ [Had15, S. 17]

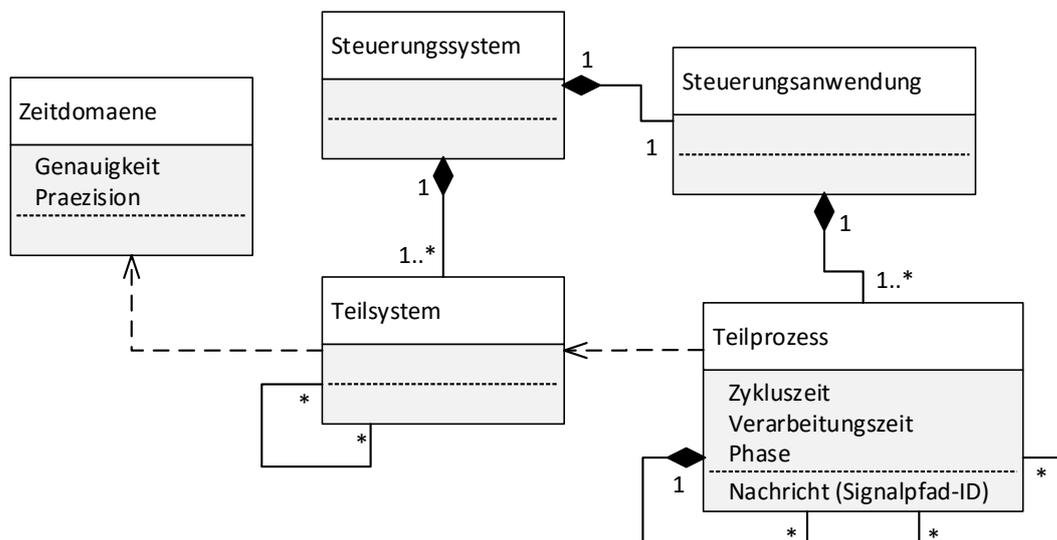


Abbildung 51: Klassendiagramm des Strukturmodells

²⁴ Die physikalische Schnittstelle zum Prozess ist nicht mit der physikalischen Schnittstelle des Kommunikationssystems (PHY) gleichzusetzen. Die physikalischen Schnittstellen zum Prozess sind die Ein- und Ausgänge des Steuerungssystems. Das Kommunikationssystem ist hier Teil der Black-Box.

²⁵ UML = Unified Modeling Language [Obj15] – siehe z. B. [RHQ+05]

Ausgangspunkt ist die Klasse *Steuerungssystem*, welche ein Gesamtsystem repräsentiert. Das Steuerungssystem enthält eine Klasse *Steuerungsanwendung*, welche die Gesamtheit aller Prozesse und Vorgänge in dem Steuerungssystem repräsentiert. Die Steuerungsanwendung ist dabei nicht mit der Applikation einer SPS zu verwechseln, diese ist nur ein Teil dieser Steuerungsanwendung.

Dieses Steuerungssystem besteht aus einer beliebigen Anzahl von Teilsystemen (mindestens eins). Ein *Teilsystem* repräsentiert eine Hardwareeinheit des Steuerungssystems, welche wiederum eine Aufgabe (*Teilprozess*) ausführen kann. Je nach gewünschter Modellierungstiefe kann ein Teilsystem ein Feldgerät, ein Kommunikationssystem oder auch ein Analog-Digital-Wandler in einem modularen Feldgerät sein.

Die Steuerungsanwendung besteht aus einer beliebigen Anzahl von Teilprozessen (mindestens eins). Diese Teilprozesse stellen die Abarbeitung von Funktionen innerhalb des Systems dar, jeder Teilprozess ist ein Vorgang. Ein Teilprozess wird einem Teilsystem zugeordnet, auf dem diese Aufgabe ausgeführt wird. Ein Teilprozess kann wiederum rekursiv in weitere Teilprozesse zerlegt werden, dies ist im Modell durch die Kompositionsbeziehung beschrieben. Somit lassen sich Hierarchien von Prozessen abbilden und bei komplexen Systemen die Verständlichkeit und Lesbarkeit des Modells erhöhen.

Zwischen Teilsystemen kann eine funktionale Beziehung existieren, im Modell beschrieben durch eine Assoziation. Diese Beziehung repräsentiert die Möglichkeit der Datenübergabe zwischen den Teilsystemen, d. h. wenn zwischen zwei Teilsystemen Daten übergeben werden sollen, muss eine Assoziation zwischen diesen bestehen.

Analog zur funktionalen Beziehung zwischen Teilsystemen kann diese funktionale Beziehung auch zwischen Teilprozessen existieren. Ein Teilprozess kann Informationen an einen anderen übergeben, dies ist im Modell durch die Operation *Nachricht* dargestellt. Mit der Nachricht wird eine ID übergeben, diese identifiziert den Signalpfad, zu dem diese Nachricht gehört. Ein Teilprozess übergibt seine Ergebnisse an den nächsten Teilprozess in der Signalkette als Nachricht.

Um den Synchronitätsaspekt zu beschreiben, ist im Modell eine Klasse *Zeitdomaene* eingeführt worden. Mehrere Teilsysteme können einer Zeitdomaene zugeordnet werden, um eine Synchronisation ihrer internen Zeitbasen (siehe Abschnitt 2.4.1) zu beschreiben. Teilsysteme, welche zu einer Zeitdomäne gehören, arbeiten zeitlich synchron. Technisch kann die Zuordnung von mehreren Teilsystemen zu einer Zeitdomäne entweder durch aktive Synchronisationsmechanismen (siehe Abschnitt 2.4) erfolgen oder aufgrund der hardwaretechnischen Gegebenheiten bedingt sein (mehrere Teilsysteme in einer Hardware mit einem Taktgeber realisiert).

Teilsysteme und Teilprozesse, die der gleichen Zeitdomäne angehören, werden im Folgenden als synchrone Teilsysteme bzw. synchrone Teilprozesse bezeichnet. Umgekehrt werden Teilsysteme und Teilprozesse in unterschiedlichen Zeitdomänen als asynchrone Teilsysteme bzw. asynchrone Teilprozesse bezeichnet. Verschiedene synchrone Teilprozesse können mit unterschiedlichen Zykluszeiten ausgeführt werden.

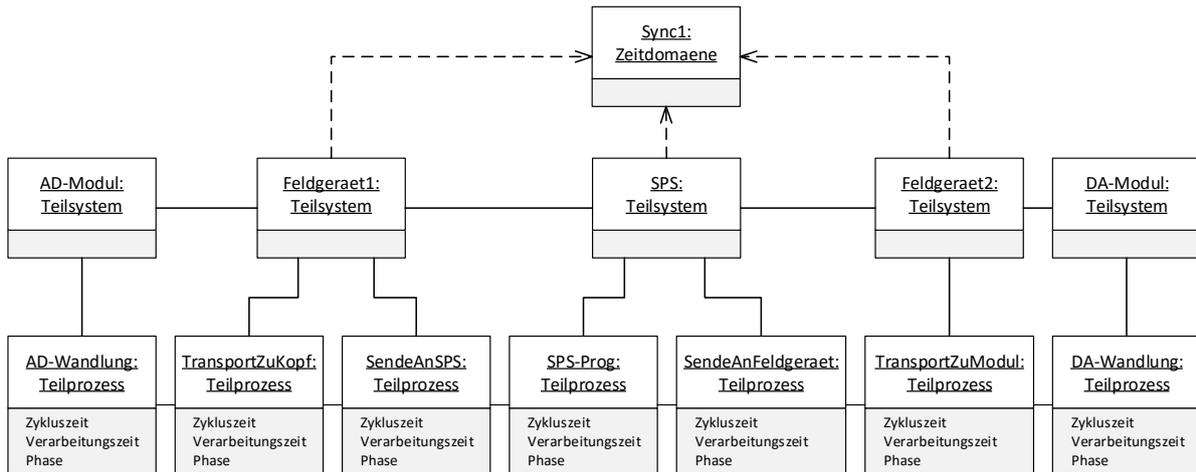


Abbildung 52: Statischer Modellanteil eines einfachen dezentralen Steuerungssystems

Die Beschreibung eines konkreten Systems erfolgt als Objektdiagramm, dieses ist beispielhaft in Abbildung 52 für ein dezentrales Steuerungssystem bestehend aus einer Steuerung und zwei Feldgeräten dargestellt (wie in Abbildung 46 eingeführt). Die Zuordnung zu den übergeordneten Objekten für das Steuerungssystem (*EinfachesSteuerungssystem* – siehe Abbildung 53) und die Steuerungsanwendung (*EinfacheSteuerungsanwendung* – siehe Abbildung 54) wird zur besseren Übersicht in separaten Diagrammen dargestellt.

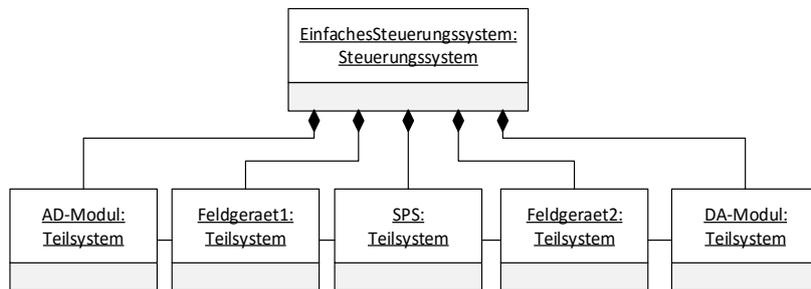


Abbildung 53: Zuordnung der Teilsysteme zum Steuerungssystem

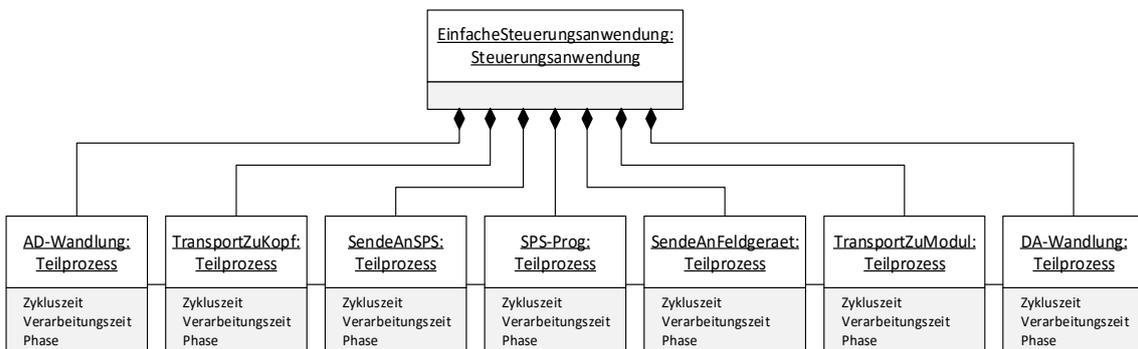


Abbildung 54: Zuordnung der Teilprozesse zur Steuerungsanwendung

Das Beispielsystem besteht aus den Teilsystemen SPS (die Steuerung selbst), den beiden Feldgeräten und jeweils einem Analog-Digital-Modul (AD-Modul) und Digital-Analog-Modul (DA-Modul) als Ein- bzw. Ausgang in den Feldgeräten. Diesen 5 Teilsystemen können 7 Teilprozesse zugeordnet werden:

- *AD-Wandlung* umfasst das Einlesen des physikalischen Signals, dessen Skalierung und die Bereitstellung als digitale Daten.
- *TransportZuKopf* umfasst die Übertragung der digitalen Daten vom Eingangsmodul über den Rückwandbus des modularen Feldgerätes zur Kopfstation.
- *SendeAnSPS* umfasst Senden der Daten über das industrielle Kommunikationssystem zur SPS.
- *SPS-Prog* repräsentiert die Abarbeitung des Steuerungsprogramms.
- *SendeAnFeldgeraet* umfasst den Transport über das Kommunikationssystem von der SPS zum Feldgerät.
- *TransportZuModul* umfasst die Verarbeitung im Feldgerätekopf und die Übertragung zum Ausgangsmodul.
- *DA-Wandlung* umfasst die Umsetzung der digitalen Daten in ein physikalisches Signal.

Die Teilprozesse *AD-Wandlung* und *DA-Wandlung* können dabei komplett in Hardware ausgeführt werden. Der Begriff des Teilprozesses setzt keine Softwareanteile voraus.

Unter der Annahme, dass das Kommunikationssystem nach dem Publisher-Subscriber-Verfahren arbeitet, sind die Teilprozesse für das industrielle Kommunikationssystem jeweils dem sendenden Teilsystem zugeordnet. Jedes Teilsystem im Modell wird einer Zeitdomäne zugeordnet. In dem Beispielsystem arbeiten SPS und Feldgeräte synchronisiert, d. h. sie gehören einer Zeitdomäne an. Dies ist beispielsweise bei einem PROFINET IRT-System ohne takt synchrone EAs der Fall.

Ist einer Zeitdomäne nur ein Teilsystem zugeordnet, d.h. das Teilsystem arbeitet asynchron zu den anderen Teilsystemen, so kann diese Zeitdomäne in der Darstellung des Modells entfallen

Für weitere Betrachtungen wird teilweise eine vereinfachte Darstellung des Modells verwendet. Im Modell werden Teilsysteme eindeutig zu einer Zeitdomäne zugeordnet. Diesen Teilsystemen werden wiederum Teilprozesse zugeordnet. Zur Vereinfachung von Formulierungen und Darstellungen werden die Teilprozesse direkt der zugehörigen Zeitdomäne zugeordnet. Aufgrund der eindeutigen Zuordnung eines Teilprozesses zu einem Teilsystem und der ebenfalls eindeutigen Zuordnung eines Teilsystems zu einer Zeitdomäne ist diese Vereinfachung zulässig. Das Modell aus Abbildung 52 ist in Abbildung 55 vereinfacht dargestellt.

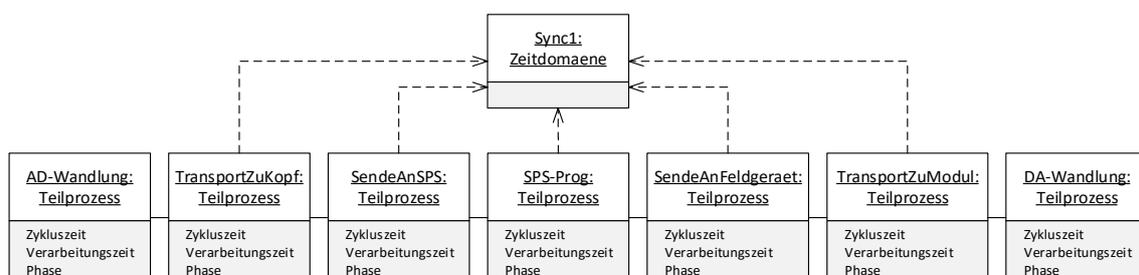


Abbildung 55: Vereinfachte Darstellung des statischen Modellanteils aus Abbildung 52

4.2.2 Dynamischer Modellanteil (Signalpfade)

Im statischen Modellanteil werden alle Möglichkeiten zum Datenaustausch zwischen Teilsystemen und Teilprozessen durch Assoziationen beschrieben. Damit bildet das statische Modell die Gesamtheit aller möglichen Signalpfade in einem Steuerungssystem ab. Sinnvollerweise werden nicht alle möglichen Signalpfade durch ein Steuerungssystem untersucht, sondern nur die konkret umgesetzten. Diese werden im dynamischen Modellanteil beschrieben.

Die konkret genutzten Signalpfade werden als Nachrichtenaustausch zwischen Teilprozessen modelliert. Dies kann als UML-Sequenzdiagramm beschrieben werden. Die Beschreibung erfolgt in Form von Nachrichten, die von einem Teilprozess an einen anderen übergeben werden. Diese Nachrichten sind im Klassendiagramm durch die Operation *Nachricht* der Teilprozesse enthalten. Um eine Zuordnung der Nachrichten bei mehreren Signalpfaden zu ermöglichen, wird jedem Signalpfad eine ID zugeordnet, welche wiederum mit der Nachricht übergeben wird. Der prinzipielle Aufbau ist in Abbildung 56 dargestellt.

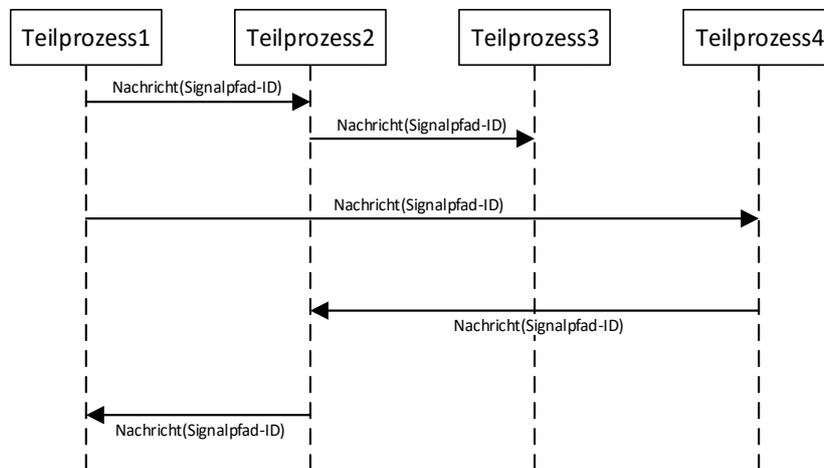


Abbildung 56: Dynamischer Modellanteil als Sequenzdiagramm

Der dynamische Modellanteil ist in Abbildung 57 exemplarisch für das Beispielsystem (Abbildung 52) dargestellt, zunächst mit nur einer Nachricht.

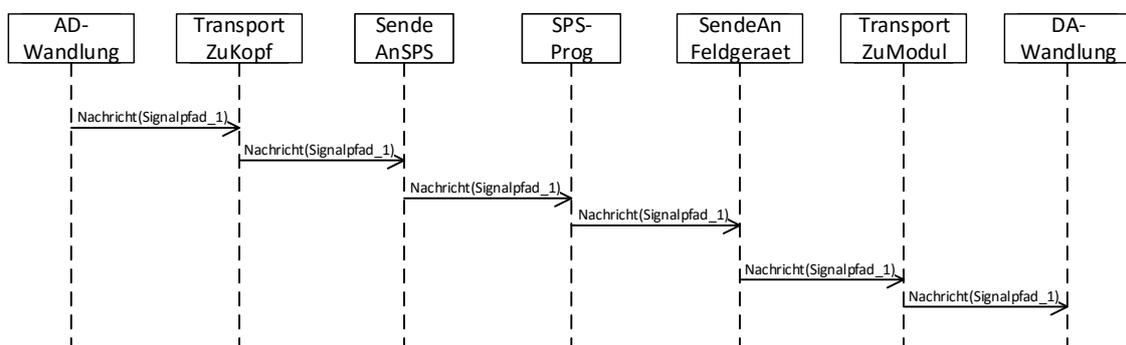


Abbildung 57: Dynamischer Modellanteil eines einfachen dezentralen Steuerungssystems

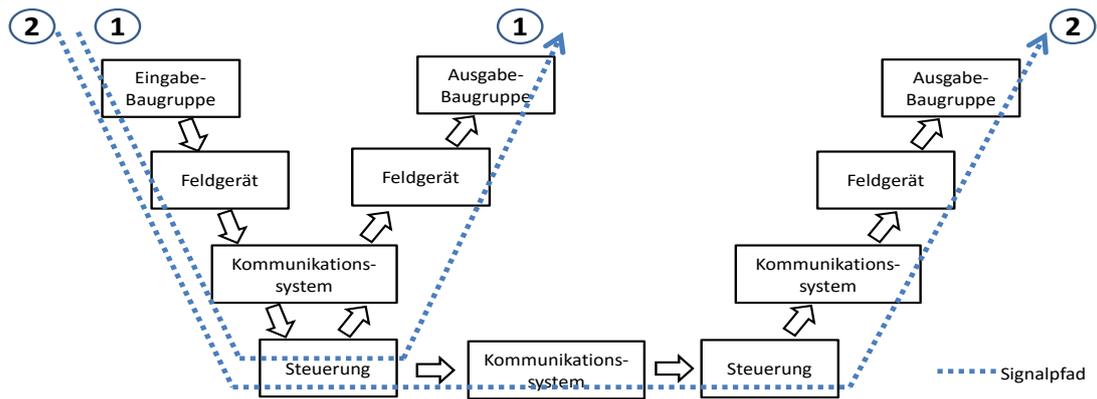


Abbildung 58: Beispielsystem mit 2 Signalpfaden

Für komplexere Topologien (Abbildung 58) ergibt sich ein dynamischer Modellanteil, wie in Abbildung 59 dargestellt. Die zeitliche Abfolge der Nachrichten innerhalb eines Signalpfades entspricht dem tatsächlichen Verlauf, während die beiden unterschiedlichen Signalpfade nur zur besseren Lesbarkeit untereinander dargestellt sind. Dies ist keine Beschreibung ihrer zeitlichen Relation.

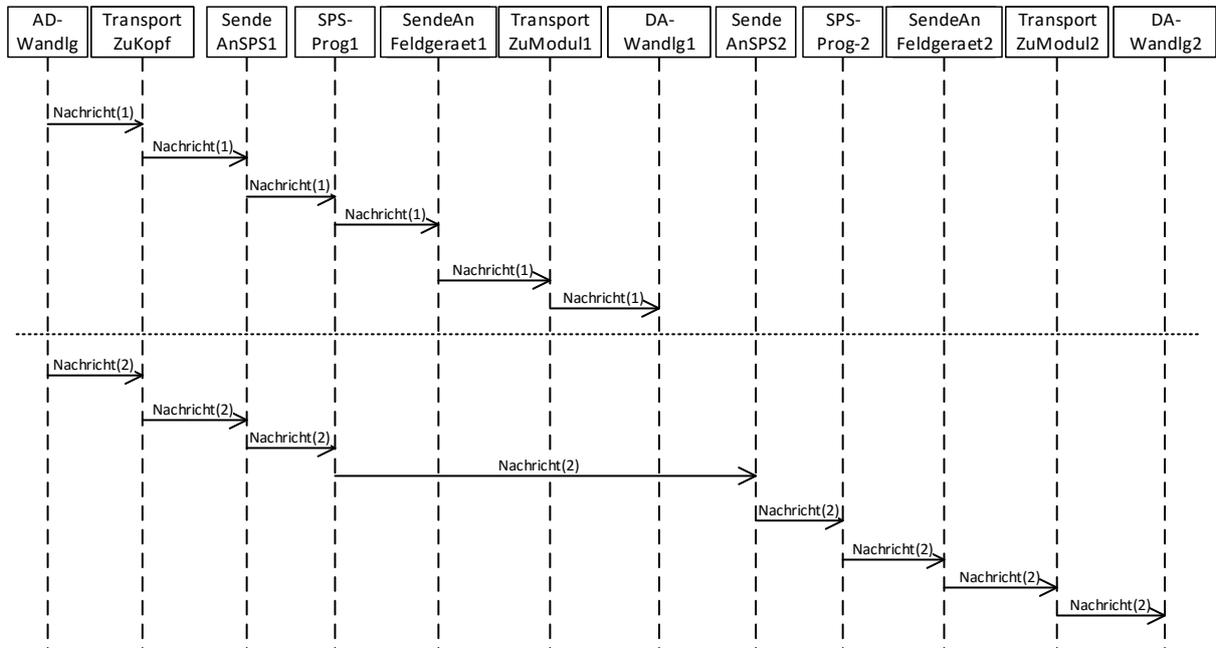


Abbildung 59: Dynamischer Modellanteil für Beispielsystem mit 2 Signalpfaden

In der Darstellung des dynamischen Modellanteils in Abbildung 59 wurde aus Gründen der Übersichtlichkeit die Signalpfad-ID als Ziffer angegeben.

4.2.3 Beschreibung der Modellparameter

Die Parametrierung des Modells erfolgt über die Attribute der Klassen und Objekte. Deren Bedeutung soll nachfolgend erläutert werden.

4.2.3.1 Klasse Zeitdomaene

Die Klasse *Zeitdomaene* wird durch zwei Attribute beschrieben: *Genauigkeit* und *Praezision*. Wie in Abschnitt 2.4.1 definiert, beschreibt die *Praezision* die maximale Abweichung $t_{\text{Praezision}}$ zwischen allen Uhren innerhalb einer Zeitdomäne. Es gilt für alle Zeitbasen t_i und t_j innerhalb der Domäne:

$$|t_i - t_j| \leq t_{\text{Praezision}} \quad (35)$$

Der Parameter *Genauigkeit* beschreibt im Gegensatz dazu als Zeit $t_{\text{Genauigkeit}}$ die maximale Abweichung der Uhren einer Zeitdomäne von einer globalen Referenzzeit, es gilt für alle Zeitbasen t_i innerhalb der Domäne:

$$|t_i - t_{\text{Referenz}}| \leq t_{\text{Genauigkeit}} \quad (36)$$

4.2.3.2 Klasse Teilprozess

Ein Teilprozess hat zeitliche Eigenschaften. Ausgehend von zyklisch arbeitenden Systemen ist er durch eine Zykluszeit T_{Cycle} und eine Verarbeitungszeit t_{process} gekennzeichnet, optional kommt noch der Parameter Phase φ hinzu. Wie in Abbildung 60 dargestellt, bezeichnet die Zykluszeit die Zeitdauer zwischen 2 Verarbeitungszyklen eines Systems, während die Verarbeitungszeit, wie oben beschrieben, der Dauer eines Prozesses entspricht.

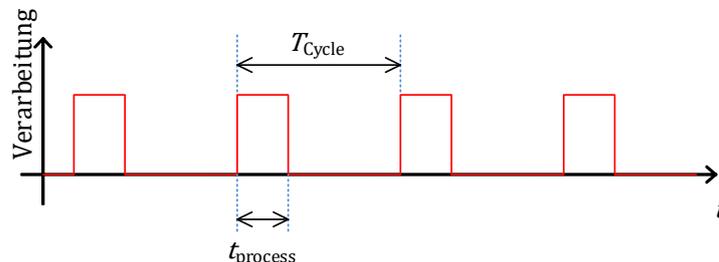


Abbildung 60: Unterschied zwischen Zykluszeit und Verarbeitungszeit

Der Parameter *Zykluszeit* wird als numerischer Wert mit der Einheit Sekunde angegeben. Die Umsetzung der *Verarbeitungszeit* ist unterschiedlich, es gibt eine analytische und eine stochastische Angabe für dieses Attribut. Bei der analytischen Variante ist die Verarbeitungszeit ein Attribut, welches durch einen numerischen Wert (Einheit Sekunde) oder alternativ eine analytische Funktion der Häufigkeitsverteilung repräsentiert wird. In der stochastischen Ausprägung wird das Attribut Verarbeitungszeit durch eine Häufigkeitsverteilung der zu erwartenden Verarbeitungszeit darstellt. Diese Art der Beschreibung der Verarbeitungszeit entspricht dem Ansatz der zufälligen Zeitverzögerung, die unabhängig von Übertragung zu Übertragung ist nach [Nil98] (siehe Abschnitt 2.5.3). Die Modellierung als analytische oder stochastische Variante ist abhängig von der Verfügbarkeit des Parameters Verarbeitungszeit. Ist eine analytische Beschreibung möglich, so ist diese zu bevorzugen.

Ein weiteres Attribut des Teilprozesses ist die Phasenlage, welche im Modell als *Phase* bezeichnet. Sie gibt den Startzeitpunkt der Abarbeitung dieses Teilprozesses bezogen auf den Zyklusstart an. Dieser

Parameter kann genutzt werden, wenn mehrere synchrone Teilprozesse mit gleicher Zykluszeit Aufgaben zu unterschiedlichen Startzeitpunkten ausführen. Die Angabe der Phasenlage erfolgt als numerischer Wert in der Einheit Sekunden. Die Angabe der Phasenlage ist optional, da sie nur beim Zusammenfassen von synchronen Teilprozessen genutzt wird.

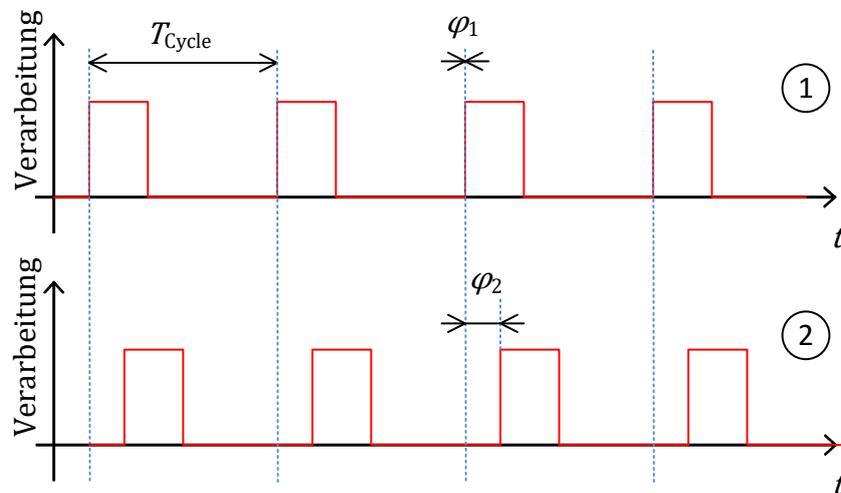


Abbildung 61: Funktion der Phasenlage im Modell

4.2.3.3 Klasse Teilsystem

Die Klasse Teilsystem besitzt in diesem Modell keine Attribute, da für die weiteren Untersuchungen im Rahmen dieser Arbeit keine Parameter vom Teilsystem betrachtet werden. Es wäre möglich weitere Aspekte wie z. B. Auslastung darzustellen.

4.2.4 Auswahl der Teilsysteme und Teilprozesse

Die Gliederung eines Steuerungssystems in Teilsysteme und Teilprozesse bietet sehr viele Freiheitsgrade, zum Beispiel in der Festlegung, wie feingranular die Gliederung erfolgt. Neben grundsätzlichen Bedingungen, wann Teilsysteme/-prozesse getrennt modelliert werden müssen, sollen im Folgenden Empfehlungen zur Gliederung gegeben werden.

Grundsätzlich gilt, dass ein Teilprozess eine eindeutige Zykluszeit und eine Verarbeitungszeit hat und in genau einer Zeitdomäne liegen kann. Zwei asynchrone Teilprozesse dürfen im Modell nicht als gemeinsamer Teilprozess modelliert werden. Wenn zwei Teilprozesse synchron und mit gleicher Zykluszeit arbeiten, so können diese zu einem Teilprozess zusammengefasst werden. Eine Modellierung als zwei getrennte Teilprozesse kann dennoch sinnvoll sein, wenn damit die Verständlichkeit des Modells verbessert wird.

Durch die Trennung von Teilsystemen als Repräsentation der physischen Komponenten des Steuerungssystems und der Teilprozesse als Vorgänge, die auf diesen Komponenten ausgeführt werden, kann die Nutzung von Teilsystemen durch verschiedene Signalpfade abgebildet werden. Ist ein Teilprozess auf einem Teilsystem (z. B. das Senden von Daten von einem Feldgerät an eine SPS) in mehreren Signalpfaden enthalten, so können in den verschiedenen Signalpfaden unterschiedliche Zykluszeiten oder Verarbeitungszeiten für diesen Teilprozess vorliegen. Dies wird im Modell durch verschiedene Teilprozesse abgebildet, welche auf dem gleichen Teilsystem ausgeführt werden. Im System in Abbildung 58 könnte dies auf die Steuerung unten links zutreffen, wenn sie für die beiden Signalpfade unterschiedliche Verarbeitungszeiten hat.

Aufgrund der Ausführung auf der gleichen Hardware (dem gleichen Teilsystem) können diese Teilprozesse als synchronisiert betrachtet werden. Durch das Attribut *Phase* der Teilprozesse kann die Phasenlage der Ausführung der Teilprozesse auf dem Teilsystem definiert werden, um z. B. eine sequentielle Abarbeitung zu beschreiben.

Soll ein System in ein Modell überführt werden, empfiehlt sich folgende Herangehensweise zur Gliederung in Teilsysteme:

1. Identifikation der Teilsysteme mit unterschiedlichen Zeitbasen und Modellierung als Teilsystem.
2. Modellierung eines Teilprozesses für die Verarbeitung in jedem Teilsystem.
3. Prüfen, ob innerhalb eines Teilsystems Vorgänge mit verschiedenen Zykluszeiten vorliegen. Wenn ja, erfolgt eine Zergliederung in mehrere Teilprozesse.
4. Wenn ein Teilprozess in mehreren Signalpfaden enthalten ist, muss geprüft werden, ob die Attribute (Zykluszeit, Verarbeitungszeit, Phase) für die verschiedenen Signalpfade unterschiedlich sind. Wenn ja, werden verschiedene Teilprozesse für die verschiedenen Signalpfade modelliert.

Wenn es der Lesbarkeit des Modells zuträgt, können Teilprozesse weiter untergliedert werden, dies ist jedoch nicht notwendig.

4.3 Ableitung des Zeitverhaltens aus dem Modell

Aus dem oben beschriebenen Modell kann das Zeitverhalten für jeden Signalpfad abgeleitet werden.

4.3.1 Prinzip der Überführung

Grundsätzlich müssen für die Ermittlung des Zeitverhaltens eines Signalpfades alle Verzögerungen (Verarbeitungszeiten, Wartezeiten) entlang des Pfades aufsummiert werden. Enthält das Modell eine Hierarchie, so werden für die Ableitung und Berechnung stets die Teilprozesse auf unterster Hierarchieebene, d. h. mit der feinsten Granularität, genutzt. Entsprechend der Gliederung in Abschnitt 4.1.2 muss zwischen den Verarbeitungszeiten des Teilprozesses selbst und den Wartezeiten, welche am Übergang zwischen Teilprozessen entstehen, unterschieden werden. In Abbildung 62 sind die möglichen Wartezeiten und die Verarbeitungszeiten für das oben genutzte Beispiel dargestellt.

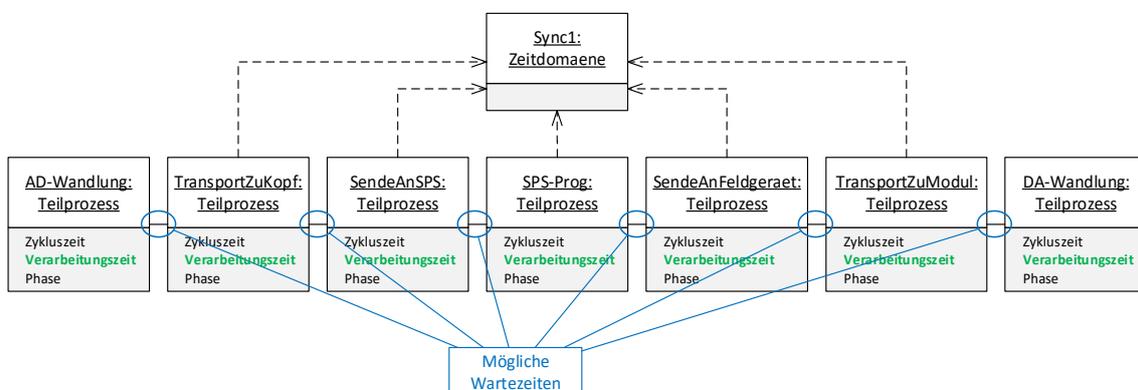


Abbildung 62: Mögliche Wartezeiten (blau) und Verarbeitungszeiten (grün) im Modell

Um aus dem Modell ein Zeitverhalten des Signalpfades abzuleiten, werden die Modellelemente entlang des Signalpfades in eine Kette von Verzögerungszeiten (wahlweise konstante Werte oder Häufigkeitsverteilungen) überführt, welche anschließend aufsummiert werden. In Abbildung 63 sind den Teilprozessen konstante Verzögerungszeiten (dargestellt durch eine Dirac-Verteilung) zugeordnet. Die Verzögerungen zwischen asynchronen Teilprozessen werden durch Rechteckverteilungen beschrieben. Zwischen Teilprozessen in der gleichen Zeitdomäne wurde keine Wartezeit eingezeichnet, diese werden als synchron betrachtet (siehe Abschnitt 4.3.3). Somit ergibt sich aus den beteiligten Teilprozessen eines Signalpfades und den Strukturinformationen des statischen Modells eine mathematische Beschreibung einer Kette von Verzögerungszeiten.

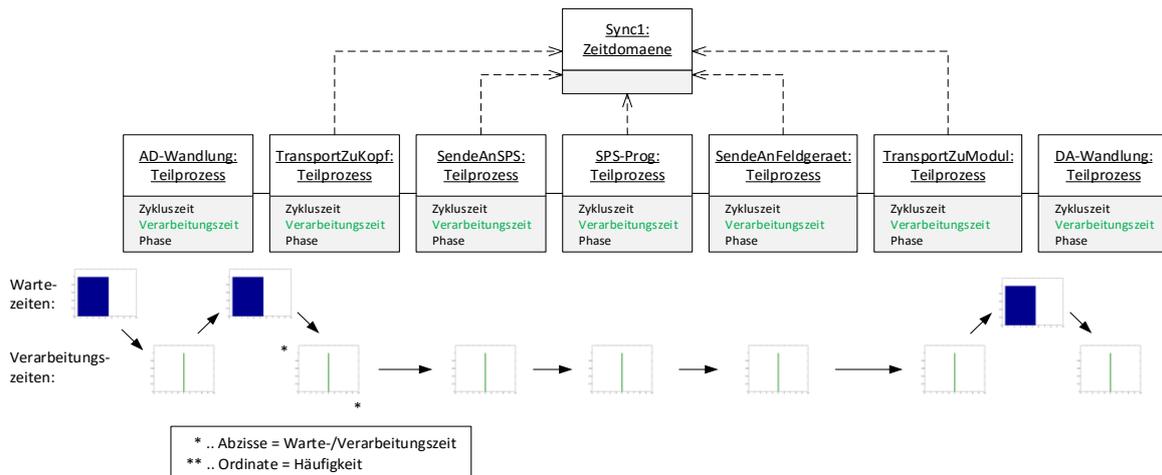


Abbildung 63: Zuordnung von Verarbeitungs- und Wartezeiten am Beispielmodell

Durch diese Vorgehensweise wird die Berechnung der Verzögerungszeit eines Signalpfades aus einem Modell in zwei Teilprobleme zerlegt:

1. Die Überführung des Modells in eine Kette von Verzögerungszeiten
2. Die Aufsummierung der Verzögerungszeiten

Durch diese Trennung kann der zweite Schritt ohne spezielle Anpassungen an Steuerungssysteme oder Wissen über Struktur des Modells ausgeführt werden. Es ist ein rein mathematisches Problem (siehe Abbildung 64). Die notwendigen Schritte für die Überführung werden in diesem Kapitel gezeigt, die mathematische Berechnung des Zeitverhaltens im Kapitel 4.4.



Abbildung 64: Aufsummierung von Verzögerungszeiten entlang eines Signalpfades

Unter der Annahme, dass jeder Teilprozess i eine Verarbeitungszeit $t_{process,i}$ und eine Wartezeit durch Asynchronität zum vorherigen Prozess $t_{wait,i}$ besitzt, ergibt sich die Klemme-Klemme-Reaktionszeit t_{IORT} prinzipiell als:

$$t_{IORT} = \sum_{i=1}^n (t_{process,i} + t_{wait,i}) \tag{37}$$

Die Aufsummierung kann bei konstanten Werten für $t_{\text{process},i}$ und $t_{\text{wait},i}$ als mathematische Addition ausgeführt werden. Liegt einer der Zeitwerte als Häufigkeitsverteilungen vor, findet die Faltungsoperation Anwendung. Eine detaillierte Darstellung erfolgt in Kapitel 4.4.

Während die Verarbeitungszeit t_{process} eines Systems als Parameter bekannt ist, ergibt sich die Wartezeit zwischen den Teilprozessen t_{wait} aus der Parameterkonstellation der beiden Teilprozesse am Übergang und muss daher im Folgenden näher untersucht werden.

4.3.2 Over- und Undersampling in der Signalkette

Die Begriffe der Überabtastung (Oversampling) und Unterabtastung (Undersampling) stammen aus der digitalen Signalverarbeitung und werden dort häufig zur Beschreibung von Analog-Digital- oder Digital-Analog-Umsetzer verwendet. Das Abtasttheorem²⁶ besagt (siehe auch [Sha98]):

„Ein analoges Signal sei bandbegrenzt, d. h. seine Fouriertransformierte verschwindet identisch für $|f| > f_{\text{max}}$. Dann lässt sich aus dem abgetasteten Signal das ursprüngliche Signal komplett und eindeutig wiedergewinnen. Dazu ist die Abtastfrequenz f_a mindest doppelt so groß zu wählen wie f_{max} .“ [WA05, S. 97]

Basierend darauf ist Oversampling die Abtastung mit einer größeren Frequenz als nach Abtasttheorem notwendig [Sch11, S. 111]. Undersampling ist entsprechend die Abtastung mit einer geringeren Frequenz als notwendig. Beim Undersampling können Informationen verloren gehen, beim Oversampling werden mehr Daten erfasst als erforderlich.

Diese beiden Begriffe werden im Folgenden für die Zykluszeiten von Teilprozessen einer Signalkette adaptiert. Entlang eines Signalpfades können unterschiedliche Zykluszeiten auftreten, so wird häufig das Kommunikationssystem mit einer deutlich kleineren Zykluszeit als das Steuerungsprogramm betrieben, um eine „quasikontinuierliche“ Übertragung zu erreichen. Die Begriffe werden nachfolgend für diesen Kontext definiert und die Auswirkung auf die Verzögerungszeiten untersucht.

Im **Kontext der Signalpfade** muss bei der Definition angegeben werden, welcher Teilprozess schneller ist, der vorherige oder der nachfolgende. Basierend auf der Definition Over- und Undersampling für Abtastung ist dies nicht eindeutig. Oversampling beim Einlesen (AD-Wandlung) bedeutet, dass der nachfolgende Teilprozess schneller arbeitet als der vorherige. Bei der Ausgabe (DA-Wandlung) ist es genau anders herum. Im nachfolgenden wird die Sichtweise der Eingangsseite verwendet (siehe [Sto11, S. 55]). Diese entspricht auch der Verwendung des Begriffes Oversampling in [KS04].

Beim **Oversampling**, auch Überabtastung genannt, liefert der vorherige Teilprozess i die Daten mit einer niedrigeren Frequenz, als der nachfolgende Teilprozess k sie verarbeitet. Dies führt dazu, dass im nachfolgenden Teilprozess mehrmals die gleichen Eingangsdaten verarbeitet werden (siehe Abbildung 65).

²⁶ Das Abtasttheorem wird in der Literatur auch Nyquist-Shannon-Theorem, Whittaker-Shannon-Theorem oder in anderen Kombinationen der Namen von Whittaker, Wladimir Kotelnikow und Shannon bezeichnet, welche es formulierten und bewiesen.

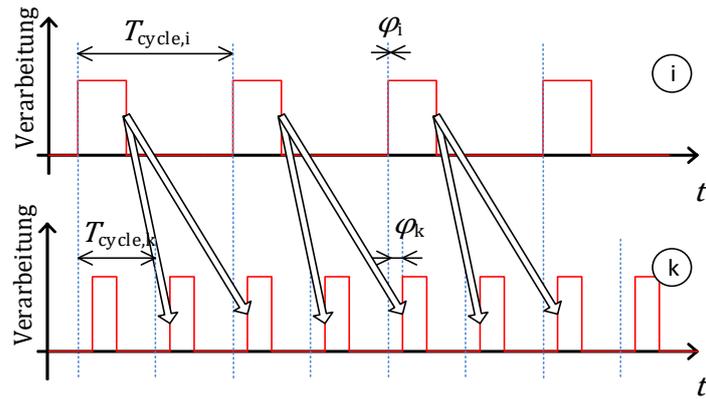


Abbildung 65: Prinzip des Oversampling beim Zyklusübergang

Für die Berechnung der Klemme-Klemme-Reaktionszeit bedeutet Oversampling, dass die gleichen Eingangsdaten nochmals zeitverzögert verarbeitet werden. Da keine neuen Informationen übermittelt werden, kann die mehrfache Datenübertragung für dieses Modell vernachlässigt werden.

Beim **Undersampling**, auch Unterabtastung genannt, werden die Daten vom vorherigen Teilprozess i mit einer höheren Frequenz bereitgestellt, als sie der nachfolgende Teilprozess k verarbeitet (siehe Abbildung 66). Dies führt dazu, dass nicht alle vom Teilprozess i gelieferten Daten weiterverarbeitet werden, es kann zu Informationsverlusten kommen.

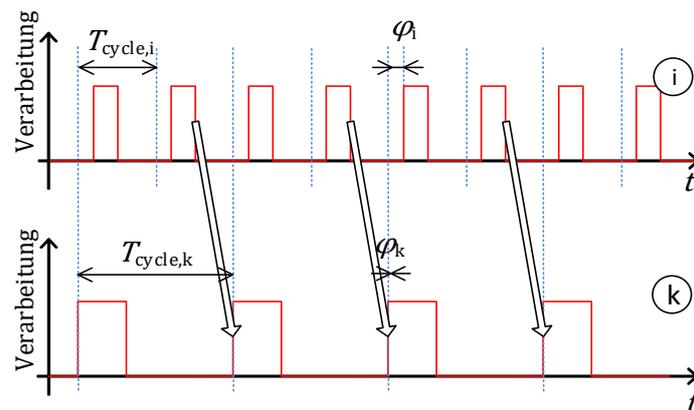


Abbildung 66: Prinzip des Undersampling beim Zyklusübergang

Es ist davon auszugehen, dass, wie in Abbildung 66 dargestellt, stets die letzten Daten vom Teilprozess i für die Weiterverarbeitung im Teilprozess k genutzt werden. Für die Berechnung der Gesamtverzögerung muss jedoch berücksichtigt werden, dass die Daten von einem Teilprozess $i-1$ eventuell nicht an den letztmöglichen Zyklus von i übergeben wurden, sondern an einen vorherigen. Für die Bestimmung der Klemme-Klemme-Reaktionszeit muss daher davon ausgegangen werden, dass die relevante Eingangsinformation zu einem beliebigen Zeitpunkt innerhalb von $T_{Cycle,k}$ eintrifft.

Nach diesen prinzipiellen Untersuchungen soll das Zeitverhalten beim Übergang zwischen zwei zyklisch arbeitenden Teilprozessen genauer untersucht werden. Dazu muss unterschieden werden, ob diese Teilprozesse zur gleichen Zeitdomäne gehören, d. h. ob sie synchron oder asynchron arbeiten.

4.3.3 Zeitverhalten am Übergang zwischen synchronen Prozessen

Sind zwei aufeinanderfolgende Teilprozesse i und k einer Zeitdomäne zugeordnet, so treten zwischen den Prozessen keine nicht-deterministischen Wartezeiten auf. Daher lassen sich zwei synchron arbeitende, aufeinanderfolgende Teilprozesse i und k zu einem Teilprozess m zusammenfassen.

4.3.3.1 Synchrone Teilprozesse mit gleicher Zykluszeit

Arbeiten beide Teilprozesse mit der gleichen Zykluszeit, hat Teilprozess m die gleiche Zykluszeit:

$$T_{\text{Cycle},m} = T_{\text{Cycle},i} = T_{\text{Cycle},k} \quad (38)$$

Die Verarbeitungszeiten müssen aufsummiert werden. Abhängig von der Phasenlage und der Verarbeitungszeit kann die Verarbeitung von k im gleichen Zyklus wie auch i stattfinden (Abbildung 67) oder erst im darauffolgenden (Abbildung 68).

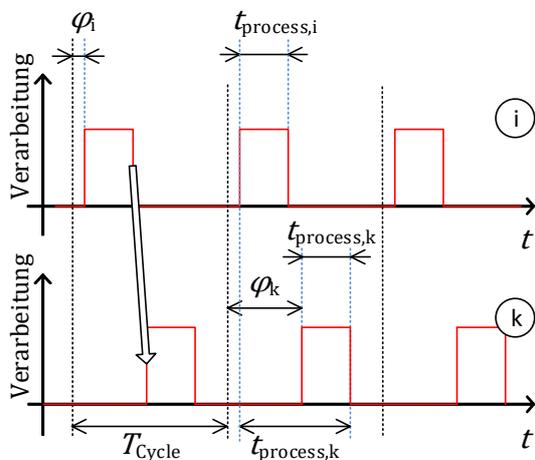


Abbildung 67: Übergang zwischen synchronen Prozessen ohne Zyklusprung

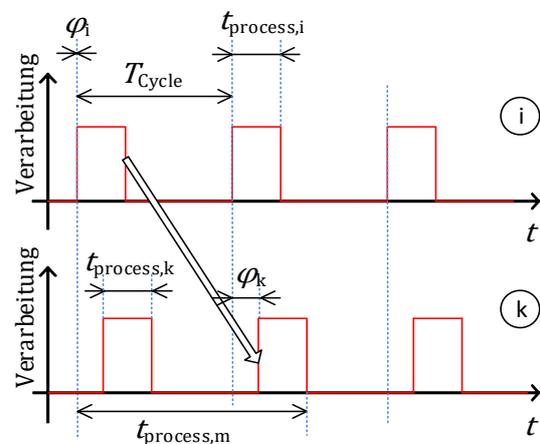


Abbildung 68: Übergang zwischen synchronen Prozessen mit Zyklusprung

Wenn die Phase von k so groß ist, dass die Verarbeitung immer nach der Fertigstellung von i beginnt, kann k die Daten im gleichen Zyklus verarbeiten, es findet kein Zyklusprung statt. Die Verarbeitungszeit des zusammengefassten Teilprozesses m ergibt sich dann als:

$$t_{\text{process},m} = \varphi_k - \varphi_i + t_{\text{process},k} \quad (39)$$

Beginnt die Verarbeitung von k vor dem Abschluss der Verarbeitung von i , so kann k die Daten erst im nächsten Zyklus verarbeiten. Die Verarbeitungszeit des zusammengefassten Teilprozesses m ergibt sich als:

$$t_{\text{process},m} = \begin{cases} \varphi_k - \varphi_i + t_{\text{process},k} & \text{für } \varphi_k \geq \varphi_i + t_{\text{process},i} \\ T_{\text{Cycle}} + \varphi_k - \varphi_i + t_{\text{process},k} & \text{für } \varphi_k < \varphi_i + t_{\text{process},i} \end{cases} \quad (40)$$

Durch Nutzung der Heaviside-Funktion²⁷ kann die Beschreibung vereinfacht werden:

$$t_{\text{process},m} = \Theta_1(\varphi_i + t_{\text{process},i} - \varphi_k) \cdot T_{\text{Cycle}} + \varphi_k - \varphi_i + t_{\text{process},k} \quad (41)$$

²⁷ Die Heaviside-Funktion wurde in Abschnitt 2.8.1 eingeführt. Hier wird die alternative Variante $\Theta_c(x)$ genutzt, bei der $\Theta_c(0) = c$ gilt.

Die Phasenlage des resultierenden Teilprozesses entspricht der Phasenlage von i :

$$\varphi_m = \varphi_i \quad (42)$$

Im Interesse einer minimalen Klemme-Klemme-Reaktionszeit sollten die Phasenlagen für synchrone Teilprozesse mit gleichen Zykluszeiten so gewählt werden, dass kein Zyklussprung auftritt.

4.3.3.2 *Synchrone Teilprozesse mit unterschiedlichen Zykluszeiten*

Wenn die Teilprozesse i und k mit unterschiedlichen Zykluszeiten arbeiten, ist das Verhältnis der beiden Zykluszeiten zu untersuchen. Es können folgende Fälle auftreten:

1. Eine Zykluszeit ist eine ganzzahlige Vielfache der anderen.
2. Beide Zykluszeiten sind ganzzahlige Vielfache einer gemeinsamen Basis.
3. Weder 1 noch 2 treffen zu, dies wird hier als teilerfremd bezeichnet.

4.3.3.2.1 Fall 1: Ganzzahlige Vielfache

Zwei Teilprozesse i und k , welche zur gleichen Zeitdomäne gehören und deren Zykluszeiten ganzzahlige Vielfache voneinander sind, werden zu einem Teilprozess m zusammengefasst. Die Zykluszeit des resultierenden Teilprozesses entspricht der größeren Zykluszeit, da nur mit dieser die Informationen übertragen werden:

$$T_{\text{Cycle},m} = \max(T_{\text{Cycle},i}; T_{\text{Cycle},k}) \quad (43)$$

Aufgrund der Annahme des ganzzahligen Vielfachen für die Ermittlung der resultierenden Verarbeitungszeit gelten die gleichen Betrachtungen wie in Abschnitt 4.3.3.1 hinsichtlich des Einflusses der Phase. Ergibt sich ein Zyklussprung, d. h. die Verarbeitung von Teilprozess i endet nach dem Beginn der Verarbeitung in Teilprozess k , so verlängert sich die Verarbeitungszeit des zusammengefassten Teilprozesses um die kleinere der beiden Zykluszeiten.

$$t_{\text{process},m} = \begin{cases} \varphi_k - \varphi_i + t_{\text{process},k} & \text{für } \varphi_k \geq \varphi_i + t_{\text{process},i} \\ \min(T_{\text{Cycle},i}; T_{\text{Cycle},k}) + \varphi_k - \varphi_i + t_{\text{process},k} & \text{für } \varphi_k < \varphi_i + t_{\text{process},i} \end{cases} \quad (44)$$

$$t_{\text{process},m} = \Theta_1(\varphi_i + t_{\text{process},i} - \varphi_k) \cdot \min(T_{\text{Cycle},i}; T_{\text{Cycle},k}) + \varphi_k - \varphi_i + t_{\text{process},k} \quad (45)$$

Die Phasenlage des zusammengefassten Teilprozesses beschreibt den Beginn der Verarbeitung von Teilprozess i im neuen Zyklus $T_{\text{Cycle},m}$. Hat Teilprozess i die größere Zykluszeit (Oversampling), d. h. $T_{\text{Cycle},m} = T_{\text{Cycle},i}$, so kann die Phasenlage von i übernommen werden.

Ist die Zykluszeit von Teilprozess k größer (Undersampling), so muss die Phasenlage φ_i in die Zykluszeit $T_{\text{Cycle},k}$ entsprechend Abbildung 69 umgerechnet werden: $\varphi_m = T_{\text{Cycle},k} - (T_{\text{Cycle},i} - \varphi_i)$

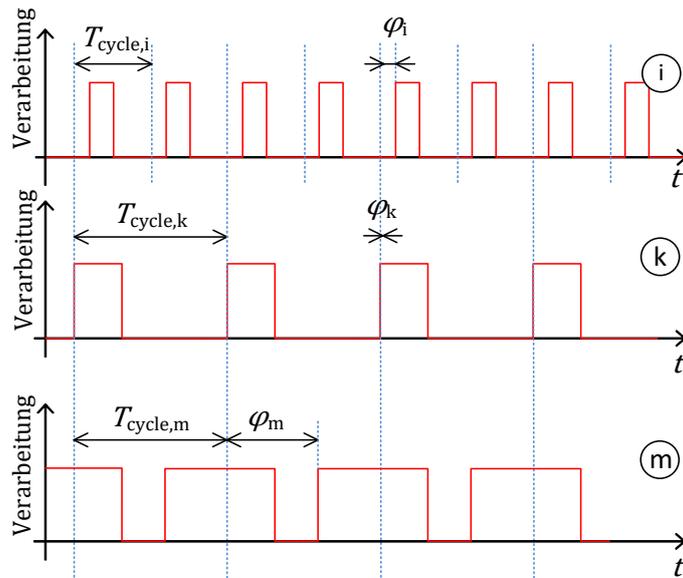


Abbildung 69: Umrechnung der Phase bei synchronen Teilprozessen

Allgemein ergibt sich die resultierende Phase zu:

$$\varphi_m = \begin{cases} \varphi_i & \text{für } T_{\text{Cycle},m} = T_{\text{Cycle},i} \\ T_{\text{Cycle},k} - T_{\text{Cycle},i} + \varphi_i & \text{für } T_{\text{Cycle},m} = T_{\text{Cycle},k} \end{cases} \quad (46)$$

4.3.3.2.2 Fall 2: Ganzzahlige Vielfache einer gemeinsamen Basis

Haben zwei synchrone Teilprozesse unterschiedliche Zykluszeiten und keine ist ein ganzzahliges Vielfaches der anderen, so muss untersucht werden, ob beide einen gemeinsamen Basistakt T_{Base} haben. Wenn mehrere Basistakte T_{Base} diese Anforderung erfüllen, so ist der größtmögliche zu wählen. Der Basistakt T_{Base} ist damit der größte Takt, der sich aus den beiden Zykluszeiten $T_{\text{Cycle},i}$ und $T_{\text{Cycle},k}$ mittels Teilung durch eine natürliche Zahl berechnen lässt.

$$T_{\text{Base}} = \max \left\{ T_{\text{Teiler}} : T_{\text{Teiler}} = \frac{T_{\text{Cycle},i}}{a} = \frac{T_{\text{Cycle},k}}{b} \text{ mit } a, b \in \mathbb{N} \right\} \quad (47)$$

Für die Zykluszeit des zusammengefassten Teilprozesses gilt analog zum Fall 1:

$$T_{\text{Cycle},m} = \max(T_{\text{Cycle},i}; T_{\text{Cycle},k}) \quad (48)$$

Die Betrachtungen aus Abschnitt 4.3.3.1 und 4.3.3.2.1 hinsichtlich Phasenlage und Verarbeitungszeiten gelten auch hier, sie müssen aber um einen zusätzlichen Einflussfaktor erweitert werden. Aufgrund der Taktteilung gibt es eine zusätzliche Wartezeit t_{wait} , welche nicht als konstant angenommen werden kann. Diese zusätzliche Wartezeit hat mehrere mögliche Werte, welche alle mit gleicher Wahrscheinlichkeit auftreten können.

Für die Ermittlung der Werte dieser zusätzlichen Verzögerung muss der Teiler aus kleinerer Zykluszeit und Basistakt bestimmt werden:

$$c = \frac{\min(T_{\text{Cycle},i}; T_{\text{Cycle},k})}{T_{\text{Base}}} = \min(a; b) \quad (49)$$

Die Verzögerungszeit t_{wait} erhält mit einer Häufigkeit von $1/c$ jeden der Werte der Menge:

$$t_{\text{wait}} = \{d \cdot T_{\text{Base}} \mid d \in \mathbb{N}_0 \mid 0 \leq d < c; \} \quad (50)$$

$$P(t_{\text{wait}}) = \frac{1}{c} \quad (51)$$

Die Verzögerungszeit des zusammengefassten Teilprozesses ergibt sich damit zu:

$$t_{\text{process},m} = \begin{cases} \varphi_k - \varphi_i + t_{\text{process},k} + t_{\text{wait}} & \varphi_k \geq \varphi_i + t_{\text{process},i} \\ \min(T_{\text{Cycle},i}; T_{\text{Cycle},k}) + \varphi_k - \varphi_i + t_{\text{process},k} + t_{\text{wait}} & \varphi_k < \varphi_i + t_{\text{process},i} \end{cases} \quad (52)$$

$$t_{\text{process},m} = \theta_1(\varphi_i + t_{\text{process},i} - \varphi_k) \cdot \min(T_{\text{Cycle},i}; T_{\text{Cycle},k}) + \varphi_k - \varphi_i + t_{\text{process},k} + t_{\text{wait}} \quad (53)$$

Die Verzögerungszeit ist damit eine **diskrete Gleichverteilung**.

Die resultierende Phase ergibt sich analog zu Fall 1:

$$\varphi_m = \begin{cases} \varphi_i & \text{für } T_{\text{Cycle},m} = T_{\text{Cycle},i} \\ T_{\text{Cycle},k} - T_{\text{Cycle},i} + \varphi_i & \text{für } T_{\text{Cycle},m} = T_{\text{Cycle},k} \end{cases} \quad (54)$$

Zum besseren Verständnis soll dieser Zusammenhang nachfolgend an 2 Beispielen erläutert werden.

4.3.3.2.1 Beispiel 1

Es werden zwei synchrone Teilprozesse angenommen, deren Zykluszeiten $T_{\text{Cycle},i} = 2 \text{ ms}$ und $T_{\text{Cycle},k} = 3 \text{ ms}$ betragen. Daraus leitet sich der Basistakt $T_{\text{Base}} = 1 \text{ ms}$ ab. In Abbildung 70 sind die Wartezeiten durch den Taktübergang dargestellt. Aus Gründen der Übersichtlichkeit wurde auf Verarbeitungszeit und Phase verzichtet und nur der Beginn der jeweiligen Zyklen dargestellt.

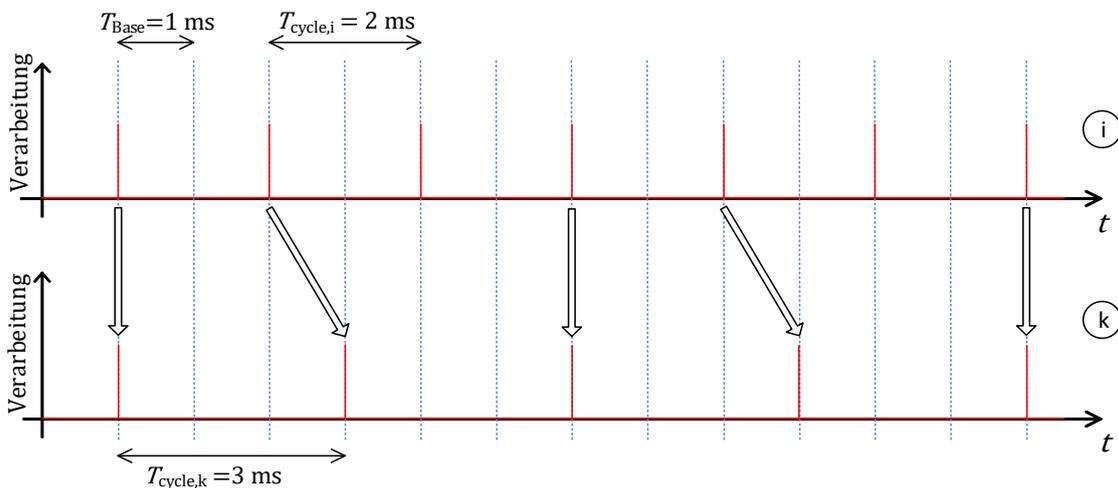


Abbildung 70: Beispiel synchrone Teilprozesse mit gemeinsamen Basistakt (1)

Anhand der oben genannten Berechnungsvorschrift ergibt sich der Faktor $c = 2$. Die Verzögerungen durch den Taktübergang sind damit zu gleicher Wahrscheinlichkeit (je 50 %) die Elemente der Menge $t_{\text{wait}} \in \{0; 1 \text{ ms}\}$. Dies lässt sich in Abbildung 70 ablesen, beim ersten Übergang entsteht keine Wartezeit, beim zweiten 1 ms, bei dritten keine usw.

4.3.3.2.2 Beispiel 2

In Abbildung 71 ist der Übergang für $T_{\text{Cycle},i} = 1,5 \text{ ms}$ und $T_{\text{Cycle},k} = 2,5 \text{ ms}$ dargestellt. Es ergibt sich ein Basistakt $T_{\text{Base}} = 0,5 \text{ ms}$. Die Verzögerungen durch den Taktübergang ergeben sich mit gleicher Wahrscheinlichkeit (33 %) als: $t_{\text{wait}} \in \{0; 0,5 \text{ ms}; 1 \text{ ms}\}$.

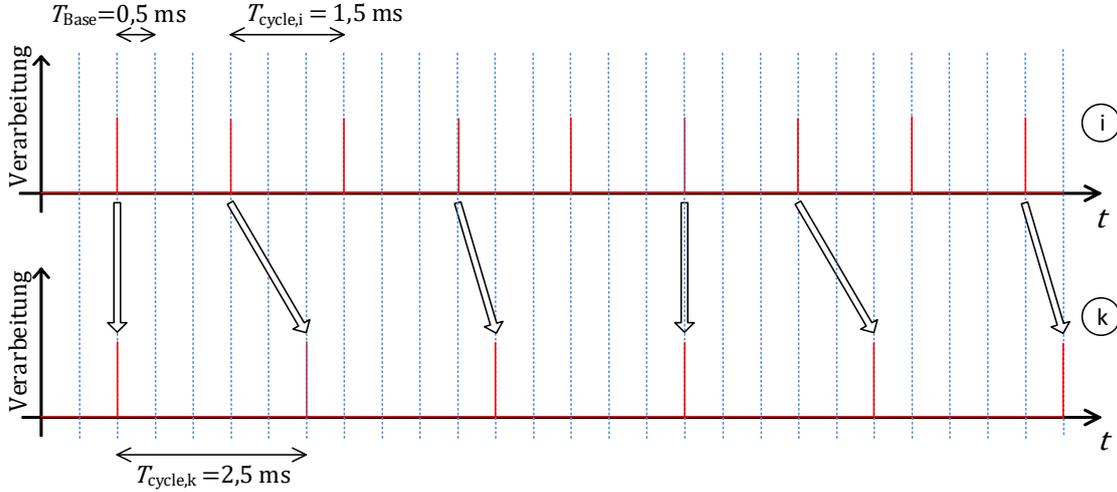


Abbildung 71: Beispiel synchrone Teilprozesse mit gemeinsamen Basistakt (2)

4.3.3.2.3 Fall 3: Teilerfremde Zykluszeiten

Ein gemeinsamer Basistakt lässt sich immer bestimmen, wenn beide Zykluszeiten rationale Zahlen sind. Ist eine (oder beide) Zykluszeit von zwei synchronen Teilprozessen eine irrationale Zahl, so kann kein gemeinsamer Basistakt bestimmt werden. In diesem Fall schwankt die zusätzliche Wartezeit durch den Taktübergang zwischen Null und der kleineren der beiden Zykluszeiten. Alle Werte haben die gleiche Häufigkeit. Da in digitalen Systemen irrationale Zahlen nur durch eine (rationale) Näherung repräsentiert werden können, wird diesem Fall keine praktische Relevanz beigemessen.

4.3.3.3 Synchrone Teilprozesse - Zusammenfassung

Arbeiten zwei Teilprozesse synchron, so können sie in jedem Fall zu einem Teilprozess zusammengefasst werden. Es gelten dabei allgemein folgende Regeln für die Zusammenfassung von Teilprozessen i und k zu einem Teilprozess m :

$$T_{\text{Cycle},m} = \max(T_{\text{Cycle},i}; T_{\text{Cycle},k}) \quad (55)$$

$$t_{\text{process},m} = \varphi_k - \varphi_i + t_{\text{process},k} + t_{\text{wait}} + \begin{cases} 0 & \varphi_k \geq \varphi_i + t_{\text{process},i} \\ \min(T_{\text{Cycle},i}; T_{\text{Cycle},k}) & \varphi_k < \varphi_i + t_{\text{process},i} \end{cases} \quad (56)$$

$$t_{\text{wait}} \in \{d \cdot T_{\text{Base}} \mid d \in \mathbb{N}_0 \mid 0 \leq d < c\} \quad (57)$$

$$c = \frac{\min(T_{\text{Cycle},i}; T_{\text{Cycle},k})}{T_{\text{Base}}} \quad (58)$$

$$\varphi_m = \begin{cases} \varphi_i & \text{für } T_{\text{Cycle},m} = T_{\text{Cycle},i} \\ T_{\text{Cycle},k} - T_{\text{Cycle},i} + \varphi_i & \text{für } T_{\text{Cycle},m} = T_{\text{Cycle},k} \end{cases} \quad (59)$$

$$T_{\text{Base}} = \max \left\{ T_{\text{Teiler}} : T_{\text{Teiler}} = \frac{T_{\text{Cycle},i}}{a} = \frac{T_{\text{Cycle},k}}{b} \text{ mit } a, b \in \mathbb{N} \right\} \quad (60)$$

Im Anhang A.1 werden Simulationen für synchrone Zyklen durchgeführt. Die Thesen wurden bestätigt.

4.3.4 Zeitverhalten am Übergang zwischen asynchronen Teilprozessen

Gehören zwei aufeinanderfolgende Teilprozesse verschiedenen Zeitdomänen an, so arbeiten sie asynchron. Bei der Übergabe von Daten zwischen diesen Teilprozessen entstehen nicht-deterministische Wartezeiten. Im Rahmen dieses Modells können die Teilprozesse daher nicht zusammengefasst werden. Für die Berechnung des Zeitverhaltens eines Signalpfades ist die Analyse der entstehenden Wartezeiten aufgrund der Asynchronität wichtig. Ausgangspunkt der nachfolgenden Betrachtung ist die Datenübergabe von Teilprozess i zu Teilprozess k , welche durch die Parameter Zykluszeit $T_{\text{Cycle},i}$ bzw. $T_{\text{Cycle},k}$ beschrieben werden.

Mangels Synchronisation kann ein Teilprozess k zu jedem Zeitpunkt innerhalb seines Zyklus Daten vom vorherigen System übergeben bekommen. Dementsprechend ist die Wartezeit bei der Übergabe gleichverteilt um Bereich von Null bis zur Zykluszeit des Nachfolgeprozesses²⁸.

$$t_{\text{wait},k} \in [0; T_{\text{Cycle},k}) \quad (61)$$

Diese Betrachtung wurde im Anhang A.2, Anhang A.3 und Anhang A.4 für einen und mehrere Zyklusübergänge per Simulation überprüft. Dabei wurden speziell auch der Einfluss von Over- und Undersampling untersucht, in allen Fällen wurde Formel (61) bestätigt.

Eine Betrachtung der Phasen der Teilprozesse ist bei einem asynchronen Übergang nicht notwendig, da die Phasenlage den Startzeitpunkt der Verarbeitung bezogen auf den eigenen Zyklusbeginn angibt. Da die Zyklen der beiden Teilprozesse in keinem festen zeitlichen Bezug zueinanderstehen, ist die Phasenlage ohne Bedeutung für die Berechnung der Wartezeit. Gleiches gilt für die Verarbeitungszeit, welche (in Zusammenhang mit der Phase) das Ende des Teilprozesses bezogen auf das lokale Zeitsystem beschreibt.

4.3.5 Überführung von analytischer in stochastische Beschreibung

In Abschnitt 4.2.3.2 wurden für die Beschreibung der Verarbeitungszeit eines Teilprozesses im vorliegenden Modell zwei Varianten eingeführt, die analytische und die stochastische. Für die spätere Berechnung kann es erforderlich sein, eine analytisch beschriebene Verarbeitungszeit in eine stochastische Beschreibung zu überführen.

In der stochastischen Beschreibung kann die Phase nicht berücksichtigt werden. Daher findet nur eine Transformation des Parameters Verarbeitungszeit $t_{\text{process,analytisch}}$ in die Verteilung $h(t_{\text{process}})$ statt. Da die Verarbeitungszeit ein konstanter Wert ist, wird sie in einer Häufigkeitsverteilung als einzelner Peak mit der Häufigkeit 1 abgebildet. Dies wird als Dirac-Verteilung bezeichnet (siehe Abschnitt 2.8.2). In Abbildung 72 ist sie grafisch dargestellt.

$$h(t_{\text{process}}) = \begin{cases} 1 & \text{für } t_{\text{process}} = t_{\text{process,analytisch}} \\ 0 & \text{sonst} \end{cases} \quad (62)$$

²⁸ Die Rechteckverteilung wird hier durch ein rechtsseitig offenes Intervall beschrieben, denn für die Wartezeit gilt: $0 \leq t_{\text{wait},k} < T_{\text{Cycle},k}$

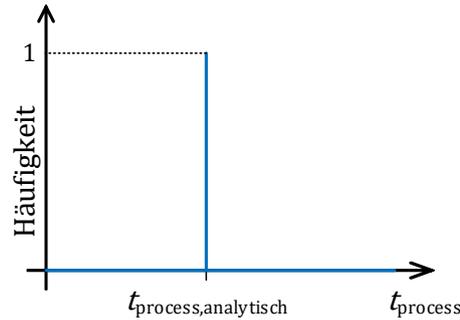


Abbildung 72: Konstante Verarbeitungszeit t_{process} als Dirac-Verteilung

4.3.6 Anbindung an den technischen Prozess

Entsprechend Abschnitt 2.1.3 muss der Übergang vom technischen Prozess zum ersten Teilprozess des Steuerungssystems, d. h. das Einlesen von Prozessdaten, als asynchroner Übergang betrachtet werden, wenn es sich um Zustandsdiskrete Signale handelt.

Entsprechend der Gleichung (61) entspricht die maximal auftretende Wartezeit an diesem Übergang der Zykluszeit des Folgeprozesses. Daher kann die auftretende Wartezeit zwischen dem technischen Prozess und dem ersten Teilprozess des Signalpfades beim Einlesen von Zustandsdiskreten Messwerten als Rechteckverteilung mit den Parametern

$$t_{\text{wait},1} \in [0; T_{\text{Cycle},1}) \quad (63)$$

angenommen werden.

4.3.7 Ablauf der Überführung

Wenn für ein System und seine Signalpfade ein Modell nach Abschnitt 4.2 erstellt wurde, sind mehrere Schritte notwendig, um daraus das Zeitverhalten abzuleiten. Die Betrachtung erfolgt dabei separat für jeden Signalpfad.

Zunächst werden innerhalb des Signalpfades aufeinanderfolgende synchrone Teilprozesse gesucht. Diese werden wie in 4.3.3.3 beschrieben zusammengefasst. Bei mehreren aufeinanderfolgenden synchronen Teilprozessen erfolgt das Zusammenfassen schrittweise. In Abbildung 73 ist dies für das Beispielmodell aus Kapitel 4.2.1 durchgeführt.

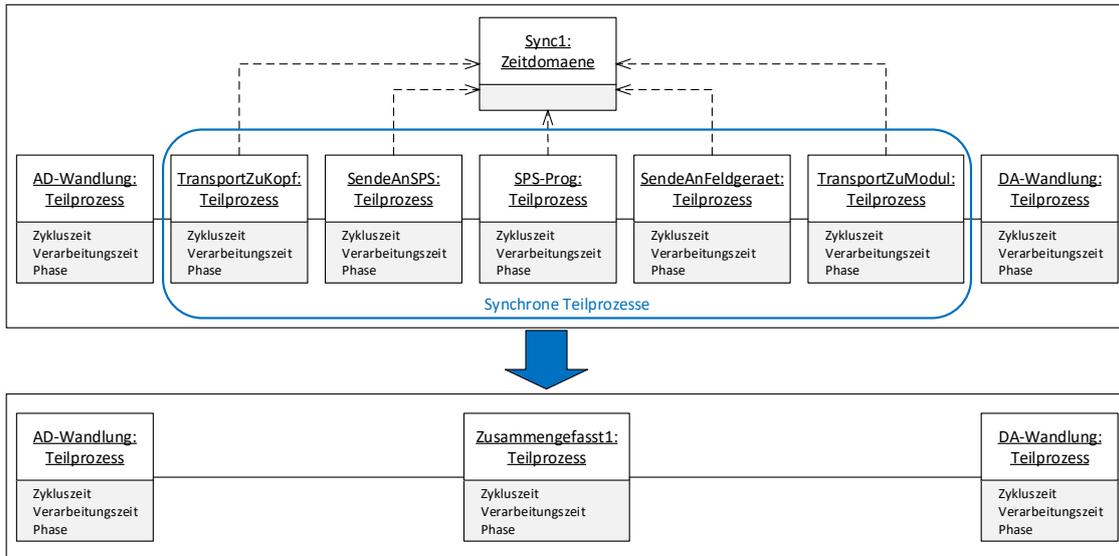


Abbildung 73: Schritt 1: Zusammenfassen von synchronen Teilsystemen am Beispiel

Nach dem Zusammenfassen der synchronen Teilprozesse besteht der Signalpfad nur noch aus asynchron arbeitenden Teilprozessen. Die zwischen diesen Teilprozessen auftretenden Wartezeiten werden im zweiten Schritt entsprechend Abschnitt 4.3.4 beschrieben. Die Häufigkeitsverteilung dieser Wartezeiten entspricht einer Rechteckverteilung, siehe Abbildung 74.



Abbildung 74: Schritt 2: Wartezeiten zwischen asynchronen Teilprozessen bestimmen

Abhängig von der Art des übertragenen Signals kann am Eingang des Signalpfades eine stochastische Wartezeit entstehen (siehe Abschnitt 4.3.6), diese wird im dritten Schritt bestimmt. Für das Beispiel ist dies in Abbildung 75 dargestellt.



Abbildung 75: Schritt 3: Stochastische Wartezeit am Eingang einfügen

Im vierten Schritt werden die Verarbeitungszeiten der Teilprozesse als Häufigkeitsverteilungen beschrieben. Abhängig von der Beschreibung im Modell kann die Verarbeitungszeit jeweils als konstante Zahl oder Häufigkeitsverteilung angegeben sein. In ersterem Fall muss eine Überführung nach Abschnitt 4.3.5 erfolgen.

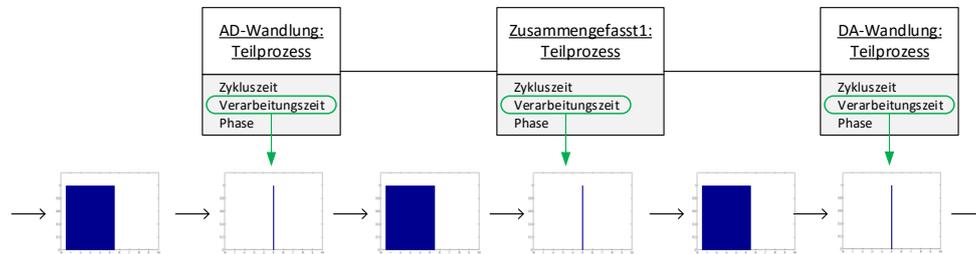


Abbildung 76: Schritt 4: Verarbeitungszeiten als Häufigkeitsverteilungen angeben

Nach diesem Schritt ist die Überführung des Modells abgeschlossen. Der Signalpfad ist in eine Kette von Verzögerungszeiten in Form von Häufigkeitsverteilungen überführt, welche aufsummiert werden müssen. Die Aufsummierung erfolgt durch Faltung und wird im nächsten Abschnitt beschrieben.

4.4 Berechnung aus Häufigkeitsverteilungen

Im Folgenden wird die Berechnung des resultierenden Zeitverhaltens eines Signalpfades durch Aufsummierung der Häufigkeitsverteilungen der Teilverzögerungen beschrieben. Durch diese Vorgehensweise kann das Zeitverhalten des Signalpfades direkt berechnet werden, es muss nicht durch Simulationen mit Variation der Ausgangslage ermittelt werden (siehe auch Abschnitt 4.5).

Diese Berechnung kann analytisch oder numerisch erfolgen, abhängig davon, wie die Verzögerungszeiten, im Speziellen die Verarbeitungszeit der Teilprozesse, beschrieben ist. Liegt für alle Häufigkeitsverteilungen der Teilverzögerungen eine analytische Beschreibung vor, so kann die Berechnung analytisch erfolgen. Ist mindestens eine Teilverzögerung durch eine nichtanalytische Beschreibung der Häufigkeitsverteilung (z. B. durch Messwerte) angegeben, so muss die Berechnung numerisch erfolgen. In bestimmten Fällen kann es trotz vollständig analytisch beschriebener Teilverzögerungen sinnvoll sein, das Ergebnis numerisch zu bestimmen, da der Rechenaufwand für eine analytische Lösung zu hoch wäre.

4.4.1 Prinzip

Als Ergebnis der Modellierung (Abschnitt 4.2) und Ableitung des Zeitverhaltens (Abschnitt 4.3) ergibt sich eine Abfolge von Verarbeitungs- und Verzögerungszeiten. Diese müssen aufsummiert werden.

Nach [Pes03, S. 99] und [Kry12, S. 188] ergibt sich die Häufigkeitsverteilung der Verarbeitungszeit von zwei aufeinanderfolgenden Prozessen als Faltung der Häufigkeitsverteilungen der beiden Verarbeitungszeiten. Liegen die Beschreibungen der Häufigkeitsverteilungen für alle Teilprozesse als Funktionen vor, kann die Berechnung analytisch erfolgen, anderenfalls bleibt die Option der numerischen Ermittlung der resultierenden Häufigkeitsverteilung.

Die Begriffe **Häufigkeitsverteilung**, **Verteilung** und **Funktion der Häufigkeitsverteilung** beziehen sich in dieser Arbeit immer auf die **Wahrscheinlichkeitsdichtefunktion** der Verteilung. Diese ist dadurch gekennzeichnet, dass die Fläche unter der Funktion den Wert 1 hat:

$$\int_{-\infty}^{\infty} h(x)dx = 1 \quad (64)$$

Des Weiteren gilt für die Zeit t (wie in technischen Systemen üblich) die Einschränkung, dass diese nur positive Werte annehmen kann:

$$t \geq 0 \quad (65)$$

4.4.2 Typische Häufigkeitsverteilungen

In den bisherigen Betrachtungen sind folgende Formen von Häufigkeitsverteilungen aufgetreten:

- Gleichverteilung / Rechteckverteilung
- Dirac-Verteilung
- Diskrete Gleichverteilung

Eine Rechteckverteilung tritt z. B. bei der stochastischen Wartezeit am Übergang zwischen zwei asynchronen Teilprozessen auf.

Die Dirac-Verteilung entspricht einer konstanten Verzögerung und tritt damit z. B. als konstante Verarbeitungszeit eines Teilprozesses auf.

Eine diskrete Gleichverteilung tritt z. B. für die Wartezeit am Übergang zwischen synchronen Teilprozessen mit gemeinsamem Basistakt auf.

Die drei Verteilungsformen wurden in Abschnitt 2.8 eingeführt und eine mathematische Beschreibung der Wahrscheinlichkeitsdichtefunktion dargestellt.

4.4.3 Faltungsoption

Die einseitige Faltung²⁹ wird entsprechend folgender Formel berechnet, dem sogenannten Faltungsintegral:

$$f_1(t) * f_2(t) = \int_0^{+\infty} f_1(\tau)f_2(t - \tau)d\tau \quad (66)$$

Die Faltungsoption ist kommutativ und assoziativ:

$$f_1 * f_2 = f_2 * f_1 \quad (67)$$

$$f_1 * (f_2 * f_3) = (f_1 * f_2) * f_3 \quad (68)$$

Daraus folgt, dass die Reihenfolge der auftretenden Verzögerungen durch oder zwischen Teilprozessen zum Zwecke der Berechnung frei wählbar ist. Ein Vorschlag zur effektiven Berechnung ist in Abschnitt 4.4.4.5 beschrieben.

²⁹ Einseitige Faltung ist zulässig da $h_n(t)$ für $t < 0$ verschwindet [BS05, S. 752]

4.4.4 Analytische Faltung von Häufigkeitsverteilungen

Im Folgenden soll die analytische Faltung der vorgestellten Verteilungen untersucht werden. Dies hat jedoch keine ausschließende Wirkung. Alle Häufigkeitsverteilungen, die analytisch beschreibbar sind, können mittels dieser Vorgehensweise berechnet werden. Aus Umfangsgründen findet hier eine Beschränkung auf diese drei Varianten statt.

4.4.4.1 Grundlagen

Die Berechnung der einseitigen Faltung kann durch Nutzung der Laplace-Transformation vereinfacht werden. Dabei gilt: "Im Bildbereich entspricht der Faltung die gewöhnliche Multiplikation" [BS05, S. 736]:

$$\mathcal{L}\{f_1 * f_2\} = F_1(p) \cdot F_2(p) \quad (69)$$

Die Laplace-Transformation ist nach [BS05, S. 733] definiert als³⁰:

$$\mathcal{L}\{f(t)\} = \int_0^{+\infty} e^{-st} f(t) dt = F(s) \quad (70)$$

Die Berechnung der Häufigkeitsverteilung der Verzögerungen entlang eines Signalpfades kann damit komplett im Bildbereich erfolgen. Abschließend ist die Rücktransformation in den Zeitbereich nötig.

Die Laplace-Rücktransformation, auch inverse Laplace-Transformation genannt, ist nach [BS05, S. 733] definiert als³¹:

$$\mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j} \int_{c-i\infty}^{c+i\infty} e^{-st} F(s) ds = \begin{cases} f(t) & \text{für } t > 0 \\ 0 & \text{für } t < 0 \end{cases} \quad (71)$$

Es stehen Transformationsregeln und Tabellen für Hin- und Rücktransformation zur Verfügung (z. B. [BS05, S. 734ff.], [BS05, S. 1097ff.]), welche für zahlreiche Funktionen die Berechnung deutlich vereinfachen. Die Nutzung der Laplace-Transformation verringert somit den Aufwand der Faltungsberechnung, insbesondere beim Zusammenfassen von mehreren Verteilungen, dafür muss aber Aufwand für Hin- und Rücktransformation betrieben werden. Während die Hin-Transformation mit Hilfe der Tabellen und Regeln meist recht einfach durchführbar ist, kann die Rücktransformation aufwendig oder unter Umständen analytisch nicht lösbar werden.

4.4.4.2 Faltungen mit Dirac-Verteilung

4.4.4.2.1 Faltung von zwei Dirac-Verteilungen

Für die Untersuchung der Faltung von zwei Dirac-Verteilungen werden folgende Ausgangsverteilungen angenommen:

$$h_1(t) = \delta(t - a) \quad (72)$$

$$h_2(t) = \delta(t - b) \quad (73)$$

³⁰ Während Bronstein für den komplexen Frequenzparameter das Zeichen p verwendet, wird hier das in der Fachliteratur häufiger genutzte Zeichen s verwendet.

³¹ Als imaginäre Einheit wird in dieser Arbeit, wie in der Elektrotechnik üblich, j anstelle des in der Mathematik genutzten i verwendet.

Die Laplace-Transformationen ergibt:

$$\mathcal{L}\{h_1(t)\} = e^{-as} \quad (74)$$

$$\mathcal{L}\{h_2(t)\} = e^{-bs} \quad (75)$$

Die Faltung im Bildbereich ergibt:

$$\mathcal{L}\{h_1 * h_2\} = e^{-as} \cdot e^{-bs} = e^{-(a+b)s} \quad (76)$$

Für die Rücktransformation kann die Rechenregel der Hin-Transformation angewandt werden:

$$h_1 * h_2 = \delta(t - (a + b)) \quad (77)$$

Bei der Faltung von zwei Dirac-Verteilungen entsteht also eine neue Dirac-Verteilung, deren Verzögerung der Summe der beiden ursprünglichen Verzögerungen entspricht.

4.4.4.2.2 Faltung von beliebiger Verteilung mit Dirac-Verteilung

Für die Faltung von einer beliebigen Verteilung $h_1(t)$ mit einer Dirac-Verteilung $h_2(t)$ kann eine Rechenregel für Laplace-Transformationen genutzt werden.

$$h_2(t) = \delta(t - a) \quad (78)$$

Es resultieren im Bildbereich:

$$\mathcal{L}\{h_1(t)\} = H_1(s) \quad (79)$$

$$\mathcal{L}\{h_2(t)\} = e^{-as} \quad (80)$$

Faltung im Bildbereich:

$$\mathcal{L}\{h_1 * h_2\} = e^{-as} \cdot H_1(s) \quad (81)$$

Nach dem Verschiebungssatz der Laplace-Transformation [BS05, S. 734ff.] entspricht eine Multiplikation mit e^{-as} im Bildbereich einer Verschiebung der Funktion um a nach rechts im Zeitbereich:

$$\mathcal{L}\{f(t - a)\} = e^{-as} \cdot F(s) \quad (82)$$

Das bedeutet, wenn eine beliebige Häufigkeitsverteilung $h_1(t)$ mit einer Dirac-Verteilung zusammengefasst werden soll, so reicht es aus, diese Verteilung um die Verzögerungszeit der Dirac-Verteilung nach rechts zu verschieben. Dies entspricht einer Verzögerung der Verteilung $h_1(t)$.

4.4.4.2.3 Fazit Dirac-Verteilungen

Es wurde gezeigt, dass sich zwei Dirac-Verteilungen zu einer zusammenfassen lassen. Durch rekursive Anwendung gilt daher, dass mehrere konstante Verzögerungen in einem System zu einer zusammengefasst werden können, indem ihre Verzögerungen aufsummiert werden.

Eine Dirac-Verteilung kann als zusätzliche Verzögerung (Verschiebung nach rechts) in eine andere Häufigkeitsverteilung integriert werden.

4.4.4.3 Faltung mit diskreter Gleichverteilung

Diskrete Gleichverteilung treten auf, wenn synchrone Teilprozesse, deren Zykluszeiten ganzzahlige Vielfache einer gemeinsamen Basis sind, zusammengefasst werden, siehe Abschnitt 4.3.3.2.2. Entstehen nach der Zusammenfassung der synchronen Teilprozesse zwei asynchrone Teilprozesse, deren Verarbeitungszeiten einer diskreten Gleichverteilung entsprechen, so müssen diese durch Faltung zusammengefasst werden.

4.4.4.3.1 Faltung von zwei diskreten Gleichverteilungen

Für die Faltung von zwei diskreten Gleichverteilungen werden folgende Ausgangsverteilungen angenommen:

$$h_1(t) = \frac{1}{n_1} \cdot \sum_{i=1}^{n_1} \delta(t - t_{1,i}) \quad (83)$$

$$h_2(t) = \frac{1}{n_2} \cdot \sum_{k=1}^{n_2} \delta(t - t_{2,k}) \quad (84)$$

Die Laplace-Transformation liefert:

$$\mathcal{L}\{h_1(t)\} = \frac{1}{n_1} \cdot \sum_{i=1}^{n_1} e^{-t_{1,i}s} \quad (85)$$

$$\mathcal{L}\{h_2(t)\} = \frac{1}{n_2} \cdot \sum_{k=1}^{n_2} e^{-t_{2,k}s} \quad (86)$$

Die Faltung im Bildbereich ergibt:

$$\mathcal{L}\{h_1 * h_2\} = \frac{1}{n_1 n_2} \cdot \sum_{i=1}^{n_1} e^{-t_{1,i}s} \cdot \sum_{k=1}^{n_2} e^{-t_{2,k}s} \quad (87)$$

Durch Ausmultiplikation der Summen erhält man:

$$\mathcal{L}\{h_1 * h_2\} = \frac{1}{n_1 n_2} \cdot (e^{-(t_{1,1}+t_{2,1})s} + \dots + e^{-(t_{1,1}+t_{2,k})s} + \dots + e^{-(t_{1,i}+t_{2,k})s} + \dots + e^{-(t_{1,n_1}+t_{2,n_2})s}) \quad (88)$$

Die Rücktransformation liefert:

$$h_1(t) * h_2(t) = \frac{1}{n_1 n_2} \cdot [\delta(t - t_{1,1} - t_{2,1}) + \dots + \delta(t - t_{1,1} - t_{2,k}) + \dots + \delta(t - t_{1,i} - t_{2,k})] \quad (89)$$

Die Faltung von 2 diskreten Gleichverteilungen, deren mögliche Ausprägungen durch die Mengen $\{t_{1,1} \dots t_{1,n_1}\}$ und $\{t_{2,1} \dots t_{2,n_2}\}$ beschrieben sind, ergibt eine diskrete Verteilung, in der jede mögliche Summenbildung der einzelnen Verteilungspunkte der Eingangsverteilungen auftritt. Diese Summenbildung entspricht der Bildung des kartesischen Produktes der beiden Mengen, dessen Elemente jeweils zu einem Skalar summiert werden. Da bei der Summenbildung doppelte Elemente in der Ergebnismenge entstehen können, muss die resultierende Verteilungsfunktion keine diskrete Gleichverteilung mehr sein.

4.4.4.3.1.1 Beispiel 1

Beispielhaft werden zwei diskrete Gleichverteilungen gefaltet, welche die folgenden möglichen Verzögerungen repräsentieren (siehe Abbildung 77):

$$t_1 \in \{1; 3; 5\} \tag{90}$$

$$t_2 \in \{2; 5\} \tag{91}$$

Es ergeben sich folgende Häufigkeitsverteilungen:

$$h_1(t) = \frac{1}{3} \cdot [\delta(t - 1) + \delta(t - 3) + \delta(t - 5)] \tag{92}$$

$$h_2(t) = \frac{1}{2} \cdot [\delta(t - 2) + \delta(t - 5)] \tag{93}$$

Die Faltung ergibt:

$$h_3 = h_1 * h_2 = \frac{1}{3 \cdot 2} \cdot [\delta(t - 1 - 2) + \delta(t - 1 - 5) + \delta(t - 3 - 2) + \delta(t - 3 - 5) + \delta(t - 5 - 2) + \delta(t - 5 - 5)] \tag{94}$$

Die resultierende Verteilung erhält also folgende mögliche Verzögerungen (siehe Abbildung 78):

$$t_3 \in \{3; 5; 6; 7; 8; 10\} \tag{95}$$

$$h_3(t) = \begin{cases} 1/6 & \text{für } t \in \{3; 5; 6; 7; 8; 10\} \\ 0 & \text{sonst} \end{cases} \tag{96}$$

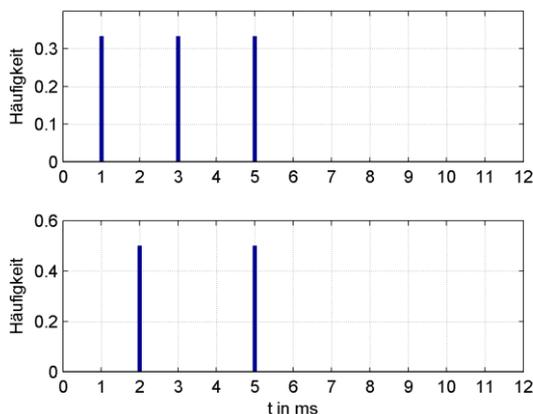


Abbildung 77: Zwei diskrete Gleichverteilungen (Beispiel 1)

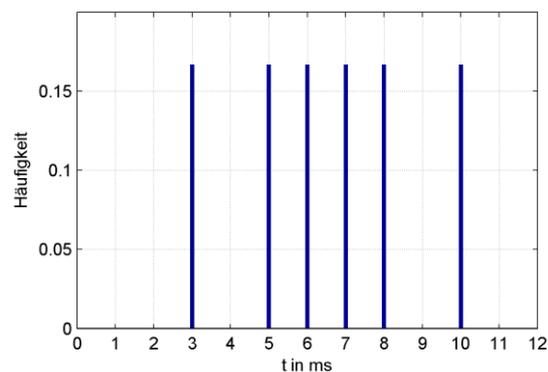


Abbildung 78: Resultierende Verteilung (Beispiel 1)

4.4.4.3.1.2 Beispiel 2

Beispielhaft werden zwei diskrete Gleichverteilungen gefaltet, welche die folgenden möglichen Verzögerungen repräsentieren (siehe Abbildung 79):

$$t_1 \in \{0; 1\} \quad (97)$$

$$t_2 \in \{0; 1; 2\} \quad (98)$$

Es ergeben sich folgende Häufigkeitsverteilungen:

$$h_1(t) = \frac{1}{2} \cdot [\delta(t - 0) + \delta(t - 1)] \quad (99)$$

$$h_2(t) = \frac{1}{3} \cdot [\delta(t - 0) + \delta(t - 1) + \delta(t - 2)] \quad (100)$$

Die Faltung ergibt:

$$h_3 = h_1 * h_2 = \frac{1}{2 \cdot 3} \cdot [\delta(t - 0 - 0) + \delta(t - 0 - 1) + \delta(t - 0 - 2) + \delta(t - 1 - 0) + \delta(t - 1 - 1) + \delta(t - 1 - 2)] \quad (101)$$

Die resultierende Verteilung erhält also folgende mögliche Verzögerungen:

$$t_3 \in \{0; 1; 2; 1; 2; 3\} \quad (102)$$

Die möglichen Zeiten werden additiv überlagert, da die Werte 1 und 2 doppelt vorkommen, erhalten sie in der resultierenden Verteilung die doppelte Häufigkeit (siehe Abbildung 80):

$$t_3 \in \{0; 1; 2; 3\} \quad (103)$$

$$h_3(t) = \begin{cases} 1/6 & \text{für } t \in \{0; 3\} \\ 1/3 & \text{für } t \in \{1; 2\} \\ 0 & \text{sonst} \end{cases} \quad (104)$$

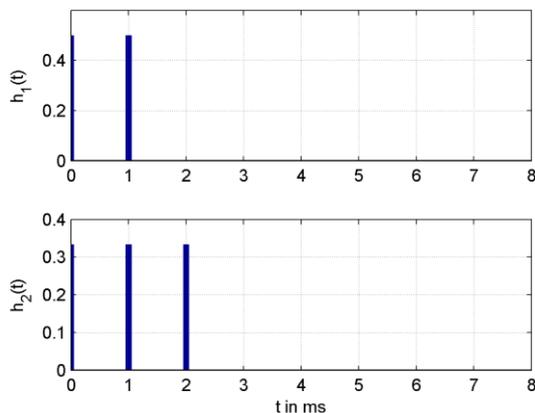


Abbildung 79: Zwei diskrete Gleichverteilungen (Beispiel 2)

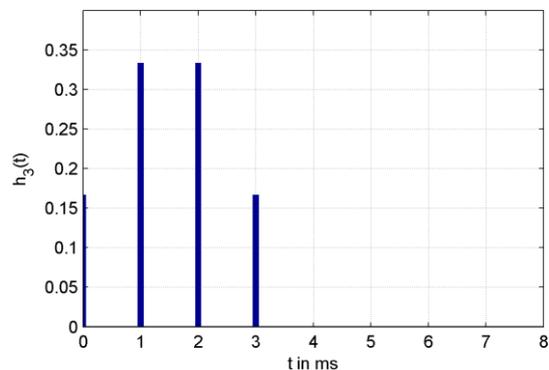


Abbildung 80: Resultierende Verteilung (Beispiel 2)

4.4.4.3.2 Faltung von beliebiger Verteilung mit diskreter Gleichverteilung

Für die Faltung von einer beliebigen Verteilung $h_1(t)$ mit einer diskreten Gleichverteilung $h_2(t)$ kann die Erkenntnis aus der Faltung von Dirac- und beliebiger Verteilung sowie der Additionssatz der Laplace-Transformation genutzt werden.

$$h_2(t) = \frac{1}{n} \cdot \sum_{i=1}^n \delta(t - t_i) \quad (105)$$

Es resultieren im Bildbereich:

$$\mathcal{L}\{h_1(t)\} = H_1(s) \quad (106)$$

$$\mathcal{L}\{h_2(t)\} = \frac{1}{n} \cdot \sum_{i=1}^n e^{-t_i s} \quad (107)$$

Faltung im Bildbereich:

$$\mathcal{L}\{h_1 * h_2\} = \frac{1}{n} \cdot \sum_{i=1}^n e^{-t_i s} \cdot H_1(s) \quad (108)$$

Dies bedeutet, die resultierende Verteilung entsteht durch Überlagerung von n Verteilungen $h_1(t)$, deren Häufigkeit jeweils um den Faktor $1/n$ skaliert und um die Verzögerung t_i nach rechts verschoben ist. Praktisch bedeutet dies, dass die Verteilung $h_1(t)$ zu gleicher Wahrscheinlichkeit um jede der möglichen Zeiten von Verteilung $h_2(t)$ verzögert wird.

4.4.4.3.2.1 Beispiel

Dies soll im Folgenden an einem Beispiel illustriert werden.

Ausgangspunkt ist eine Rechteckverteilung mit dem Intervall $[1; 2)$ und eine diskrete Gleichverteilung mit den möglichen Verzögerungen $t_2 \in \{2; 5\}$.

$$h_1(t) = \frac{1}{1} [\Theta(t - 1) - \Theta(t - 2)] \quad (109)$$

$$h_2(t) = \frac{1}{2} \cdot [\delta(t - 2) + \delta(t - 5)] \quad (110)$$

Laplace-Transformation:

$$\mathcal{L}\{h_1(t)\} = \frac{1}{s} \cdot (e^{-1s} - e^{-2s}) \quad (111)$$

$$\mathcal{L}\{h_2(t)\} = \frac{1}{2} \cdot (e^{-2s} + e^{-5s}) \quad (112)$$

Faltung im Bildbereich:

$$\mathcal{L}\{h_1 * h_2\} = \frac{1}{2s} \cdot [(e^{-1s} - e^{-2s}) \cdot e^{-2s} + (e^{-1s} - e^{-2s}) \cdot e^{-5s}] \quad (113)$$

$$\mathcal{L}\{h_1 * h_2\} = \frac{1}{2s} \cdot (e^{-3s} - e^{-4s} + e^{-6s} - e^{-7s}) \quad (114)$$

$$\mathcal{L}\{h_1 * h_2\} = \frac{e^{-3s} - e^{-4s}}{2s} + \frac{e^{-6s} - e^{-7s}}{2s} \quad (115)$$

Dies entspricht der Summe von zwei Rechteckverteilungen, die Rücktransformation liefert:

$$h_1 * h_2 = \frac{1}{2} [\Theta(t - 3) - \Theta(t - 4) + \Theta(t - 6) - \Theta(t - 7)] \quad (116)$$

Wie oben beschrieben, entspricht die resultierende Verteilung der Summe von zwei Ursprungsverteilungen $h_1(t)$, jeweils um den Faktor $1/2$ skaliert und um 2 bzw. 5 nach rechts verschoben. Die Ursprungs- und resultierende Verteilung sind in Abbildung 81 dargestellt.

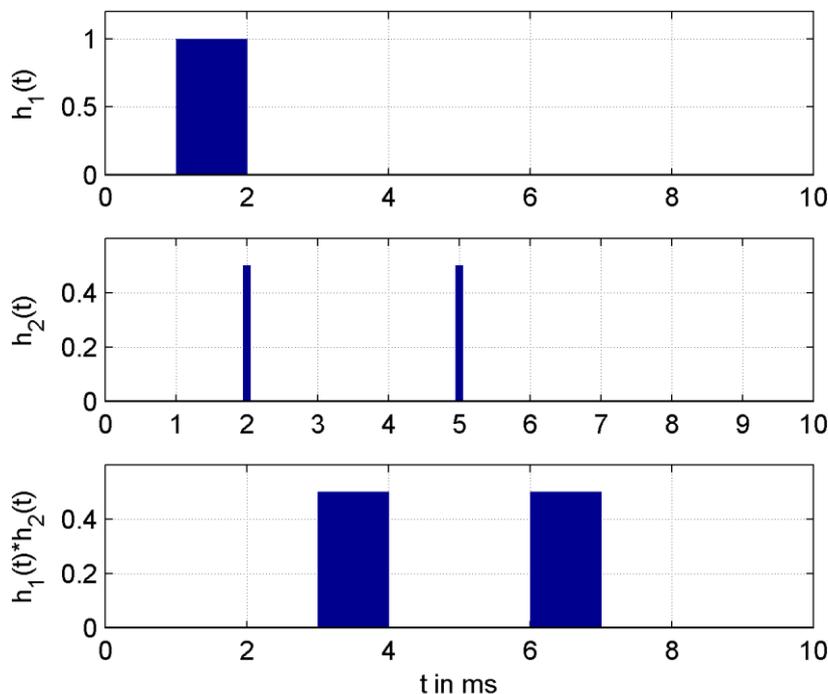


Abbildung 81: Faltung einer diskreten Gleichverteilung mit einer Rechteckverteilung

4.4.4.4 Faltung von Rechteckverteilungen

Bei Übergang zwischen asynchronen Teilprozessen entsteht eine Wartezeit, deren Häufigkeitsverteilung einer Rechteckverteilung entspricht. Daher hat die Verknüpfung von zwei oder mehr dieser Verteilungen eine zentrale Bedeutung für die Ableitung des Zeitverhaltens entlang eines Signalpfades.

4.4.4.4.1 Faltung von zwei Rechteckverteilung

Zunächst soll hier die Faltung von zwei Rechteckverteilungen betrachtet werden. Mathematisch betrachtet ist die Aufgabe die Faltung von zwei nicht-nullsymmetrischen Rechteckverteilungen.

Ausgangspunkt der Betrachtung sind zwei Rechteckverteilungen h_1 und h_2 , deren Verzögerungen im Intervall $[a_1; a_1 + T_1)$ bzw. $[a_2; a_2 + T_2)$ liegen, mit nachfolgenden Dichtefunktionen:

$$h_1(t) = \frac{1}{T} [\Theta(t - a_1) - \Theta(t - (a_1 + T_2))] \quad (117)$$

$$h_2(t) = \frac{1}{T} [\Theta(t - a_2) - \Theta(t - (a_2 + T_2))] \quad (118)$$

Rechteckverteilungen in Laplace-Darstellung:

$$\mathcal{L}\{h_1(t)\} = \frac{(e^{-a_1s} - e^{-(a_1+T_1)s})}{T_1s} \quad (119)$$

$$\mathcal{L}\{h_2(t)\} = \frac{(e^{-a_2s} - e^{-(a_2+T_2)s})}{T_2s} \quad (120)$$

Faltung im Bildbereich:

$$\mathcal{L}\{h_1 * h_2\} = \frac{e^{-(a_1+a_2)s} - e^{-(a_1+a_2+T_1)s} - e^{-(a_1+a_2+T_2)s} + e^{-(a_1+a_2+T_1+T_2)s}}{T_1T_2s^2} \quad (121)$$

Konstanten zusammenfassen:

$$\mathcal{L}\{h_1 * h_2\} = \frac{e^{-b_1s} - e^{-b_2s} - e^{-b_3s} + e^{-b_4s}}{T_1T_2s^2} \quad (122)$$

Rücktransformation in Zeitbereich³²:

$$h_1 * h_2 = \frac{1}{T_1T_2} \cdot [(t - b_1) \cdot \Theta(t - b_1) - (t - b_2) \cdot \Theta(t - b_2) - (t - b_3) \cdot \Theta(t - b_3) + (t - b_4) \cdot \Theta(t - b_4)] \quad (123)$$

Die resultierende Dichtefunktion ist die Summe von 4 linearen Funktionen mit dem Anstieg $\frac{1}{T_1T_2}$, welche durch die Verwendung der Heaviside-Sprungfunktion eine abschnittsweise definierte Funktion ergibt.

Für die weiteren Betrachtungen wird zur Vereinfachung der Darstellung angenommen, dass $b_2 \leq b_3$ und damit $T_1 \leq T_2$. Aufgrund der kommutativen Eigenschaft der Faltungsoperation stellt dies keine Einschränkung der Gültigkeit dar.

$$h_1 * h_2 = \frac{1}{T_1T_2} \cdot \begin{cases} 0 & \text{für } t \leq b_1 \\ (t - b_1) & \text{für } b_1 < t \leq b_2 \\ b_2 - b_1 & \text{für } b_2 < t \leq b_3 \\ -t - b_1 + b_2 + b_3 & \text{für } b_3 < t \leq b_4 \\ -b_1 + b_2 + b_3 - b_4 & \text{für } b_4 < t \end{cases} \quad (124)$$

³² Nach Rechenregeln zur Laplace-Transformation [BS05, S. 734ff.] und Tabellen zur Laplace-Transformation [BS05, S. 1097ff.]

Einsetzen der Konstanten liefert:

$$h_1 * h_2 = \frac{1}{T_1 T_2} \cdot \begin{cases} 0 & \text{für } t \leq b_1 \\ (t - b_1) & \text{für } b_1 < t \leq b_2 \\ T_1 & \text{für } b_2 < t \leq b_3 \\ -t + b_4 & \text{für } b_3 < t \leq b_4 \\ 0 & \text{für } b_4 < t \end{cases} \quad (125)$$

Die resultierende Häufigkeitsverteilung besteht aus 5 Abschnitten mit den folgenden 4 Knickpunkten:

- $b_1 = a_1 + a_2$ (positiver Knick)
- $b_2 = a_1 + a_2 + T_1$ (negativer Knick)
- $b_3 = a_1 + a_2 + T_2$ (negativer Knick)
- $b_4 = a_1 + a_2 + T_1 + T_2$ (positiver Knick)

Wenn $T_1 \neq T_2$ ist, ergibt sich damit eine Trapezverteilung (siehe Abbildung 82), für den Sonderfall $T_1 = T_2$ erhält man eine Dreiecksverteilung (Abbildung 83).

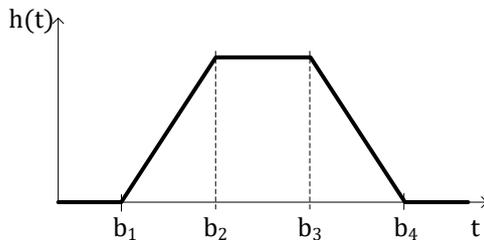


Abbildung 82: Trapezverteilung ($T_1 \neq T_2$)

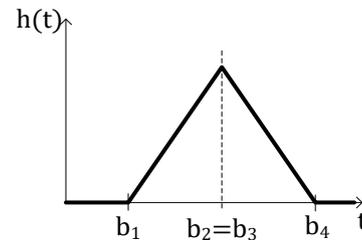


Abbildung 83: Dreiecksverteilung ($T_1 = T_2$)

4.4.4.4.2 Faltung von drei Rechteckverteilung

Das gleiche Prinzip lässt sich auch auf die Faltung von drei Rechteckverteilungen, d. h. auf drei asynchrone Zyklusübergänge entlang eines Signalpfades, anwenden.

Es werden Rechteckverteilungen gefaltet, deren Verzögerungen in den Intervallen $[a_1; a_1 + T_1)$, $[a_2; a_2 + T_2)$ und $[a_3; a_3 + T_3)$ liegen. Die Dichtefunktionen dieser Verteilungen im Bildbereich lauten:

$$\mathcal{L}\{h_1(t)\} = \frac{(e^{-a_1 s} - e^{-(a_1 + T_1)s})}{T_1 s} \quad (126)$$

$$\mathcal{L}\{h_2(t)\} = \frac{(e^{-a_2 s} - e^{-(a_2 + T_2)s})}{T_2 s} \quad (127)$$

$$\mathcal{L}\{h_3(t)\} = \frac{(e^{-a_3 s} - e^{-(a_3 + T_3)s})}{T_3 s} \quad (128)$$

Die Faltung im Bildbereich mit anschließendem Zusammenfassen der Konstanten liefert:

$$\mathcal{L}\{h_1 * h_2 * h_3\} = \frac{e^{-b_1 s} - e^{-b_2 s} - e^{-b_3 s} - e^{-b_4 s} + e^{-b_5 s} + e^{-b_6 s} + e^{-b_7 s} - e^{-b_8 s}}{T_1 T_2 T_3 s^3} \quad (129)$$

Bei der Rücktransformation wird zur übersichtlichen Darstellung $f(t) = t^2 \cdot \Theta(t)$ als Hilfsfunktion eingeführt:

$$h_1 * h_2 * h_3 = \frac{1}{2 \cdot T_1 T_2 T_3} \cdot [f(t - b_1) - f(t - b_2) - f(t - b_3) - f(t - b_4) + f(t - b_5) + f(t - b_6) + f(t - b_7) - f(t - b_8)] \quad (130)$$

Das Ergebnis ist eine Verteilung mit einer abschnittsweise definierten Dichtefunktion, die durch Aufsummierung von quadratischen Funktionen mit der Streckung $\frac{1}{2 \cdot T_1 T_2 T_3}$ entsteht. Diese Funktion besteht aus 9 Abschnitten mit bis zu 8 Knickpunkten:

- $b_1 = a_1 + a_2 + a_3$ (positiver Knick)
- $b_2 = a_1 + a_2 + a_3 + T_1$ (negativer Knick)
- $b_3 = a_1 + a_2 + a_3 + T_2$ (negativer Knick)
- $b_4 = a_1 + a_2 + a_3 + T_3$ (negativer Knick)
- $b_5 = a_1 + a_2 + a_3 + T_1 + T_2$ (positiver Knick)
- $b_6 = a_1 + a_2 + a_3 + T_1 + T_3$ (positiver Knick)
- $b_7 = a_1 + a_2 + a_3 + T_2 + T_3$ (positiver Knick)
- $b_8 = a_1 + a_2 + a_3 + T_1 + T_2 + T_3$ (negativer Knick)

Die Form der Verteilung ist abhängig von den Werten der Parameter T_1, T_2 und T_3 , in Abbildung 84 ist sie beispielhaft dargestellt.

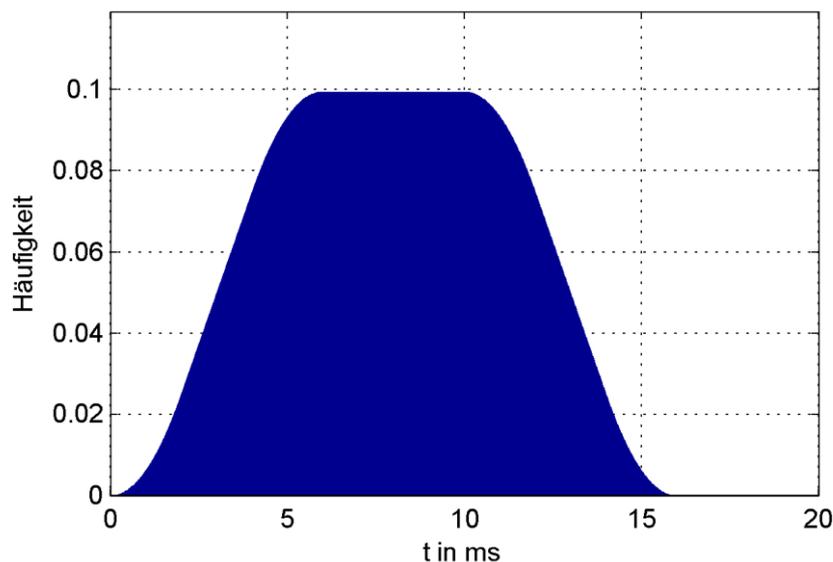


Abbildung 84: Beispielhafte Häufigkeitsverteilung für drei asynchrone Zyklusübergänge

4.4.4.4.3 Faltung von n Rechteckverteilungen

Verallgemeinert kann die Faltung auch für n Rechteckverteilungen mit den Intervallen $[a_i; a_i + T_i)$ ($i = 1 \dots n$) durchgeführt werden. Dies erfolgt analog zur Vorgehensweise bei 2 oder 3 Zyklusübergänge, weshalb hier auf eine ausführliche Betrachtung verzichtet wird.

Für jede Verteilung h_i gilt:

$$\mathcal{L}\{h_i(t_{\text{IORT}})\} = \frac{(e^{-a_i s} - e^{-(a_i+T_i)s})}{T_i s} \quad (131)$$

Die Faltung liefert folgende Dichtefunktion der Häufigkeitsverteilung im Bildbereich:

$$\mathcal{L}\{h_1 * h_2 * \dots * h_n\} = \prod_{i=1}^m \frac{e^{-a_i s} - e^{-(a_i+T_i)s}}{T_i s} = \frac{1}{s^n \prod T_i} \cdot \prod_{i=1}^m (e^{-a_i s} - e^{-(a_i+T_i)s}) \quad (132)$$

Für diese Funktion kann keine allgemeine Rücktransformation angegeben werden, jedoch kann sie für eine bekannte Anzahl von Verteilungen n durchgeführt werden. Aus Gleichung (132) und den Erkenntnissen in Abschnitt 4.4.4.2 lassen sich folgende Aussagen treffen:

- Die resultierende Dichtefunktion der Verteilung ist abschnittsweise definiert mit 2^n Knickpunkten.
- Die Abschnitte werden durch die Summe von Teilfunktionen des Grades $(n - 1)$ bestimmt.

Bei der Multiplikation von Exponentialfunktionen werden deren Exponenten addiert. Die Knickpunkte der resultierenden Funktionen entstehen durch Summenbildung der Exponenten, d. h. der Konstanten a_i und $a_i + T_i$. Der kleinstmögliche Wert dieser Summe und damit der erste Knickpunkt ergibt sich zu:

$$b_1 = \sum_{i=1}^n a_i \quad (133)$$

Der größtmögliche Wert und damit der letzte Knickpunkt liegt bei:

$$b_n = \sum_{i=1}^n a_i + \sum_{i=1}^n T_i \quad (134)$$

Diese Knickpunkte stellen die minimalen und maximalen Werte der Häufigkeitsverteilungen dar:

$$\min(t_{\text{IORT}}) = \sum_{i=1}^n a_i \quad (135)$$

$$\max(t_{\text{IORT}}) = \sum_{i=1}^n a_i + \sum_{i=1}^n T_i \quad (136)$$

Messungen und Berechnungen haben gezeigt, dass mit zunehmender Anzahl von Rechteckverteilungen, welche gefaltet werden, die resultierende Verteilung sich einer **beschränkten Normalverteilung** [JKB94] annähert.

4.4.4.5 Vorgehensweise bei der analytischen Faltung

Um das Zeitverhalten eines Signalpfades bei bekanntem Zeitverhalten aller Teilprozesse und der Zyklusübergänge durch analytische Faltung zu bestimmen, empfiehlt sich aufgrund der Erkenntnisse in Abschnitt 4.4.4 folgende Vorgehensweise:

1. Alle Dirac-Verteilungen zusammenfassen und ans Ende sortieren.
2. Alle Rechteckverteilungen zusammenfassen.
3. Die resultierende Verteilung mit eventuellen diskreten Gleichverteilungen zusammenfassen.
4. Die resultierende Verteilung entsprechend der zusammengefassten Dirac-Verteilung aus Schritt 1 verzögern.
5. Das Ergebnis mit eventuellen Verteilungen, die nicht den Schritten 1-4 entsprechen, zusammenfassen.

Da die meisten Systeme aus einer Menge von Dirac- und Rechteckverteilungen bestehen, ist der Rechenaufwand mit den hier gezeigten Formeln gering. Des Weiteren ergibt sich aufgrund der Erkenntnisse zur Häufigkeitsverteilung der Zusammenfassung von n Rechteckverteilungen ein Rückwärtsansatz zur Analyse eines unbekanntes Systems anhand des Zeitverhaltens eines oder mehrere Signalpfade. Hierauf wird in Kapitel 6.4 näher eingegangen.

4.4.5 Numerische Faltung von Häufigkeitsverteilungen

Der Ansatz, die Häufigkeitsverteilung der Zeitverzögerung entlang eines Signalpfades mittels Faltungsoperation zu berechnen, ist allgemeingültig. Jedoch kann der im Abschnitt 4.4.4 beschriebene analytische Weg nur genutzt werden, wenn die Dichtefunktionen aller Häufigkeitsverteilungen analytisch beschrieben sind. Es kann jedoch vorkommen, dass für die Häufigkeitsverteilung der Verzögerung eines Teilprozesses Messwerte vorliegen, für die keine analytische Approximation möglich ist. Um trotzdem das Zeitverhalten des Signalpfades zu berechnen, wird im Folgenden die numerische Faltung genutzt.

4.4.5.1 Grundlagen numerische Faltung

Bei der numerischen (diskreten³³) Faltung wird das Faltungsintegral aus Gleichung (66) durch eine Summe ersetzt (nach [MH04, S. 88]):

$$(f_1 * f_2)(n) = \sum_{m=-\infty}^{\infty} f_1(m) \cdot f_2(n - m) \quad (137)$$

Da hier nur die einseitige Faltung genutzt wird (Verzögerungszeiten sind immer positiv), kann folgende Vereinfachung getroffen werden:

$$(f_1 * f_2)(n) = \sum_{m=0}^{\infty} f_1(m) \cdot f_2(n - m) \quad (138)$$

³³ Die Begriffe „numerische Faltung“ und „diskrete Faltung“ werden synonym verwendet. Während in der Fachliteratur zur Signalverarbeitung häufiger von „diskreter Faltung“ gesprochen wird, wird hier zur Betonung des Unterschiedes zum analytischen Berechnungsweg der Begriff der „numerischen Faltung“ genutzt.

Die numerische Faltung kann auch auf analytisch beschriebene Häufigkeitsverteilungen angewendet werden, dazu müssen diese zunächst zeitlich diskretisiert werden. Dabei wird aus der kontinuierlichen Größe $t \in [0; \infty)$ eine Menge von Zeitpunkten t_i , für die jeweils eine Häufigkeit $h(t_i)$ angegeben wird.

Die numerische Faltung ist in gängigen Mathematik-Softwarepaketen implementiert, hier wurde die Funktion `conv` aus MATLAB³⁴ genutzt.

4.4.5.2 Vorgehensweise

Beim numerischen Ansatz können die Verzögerungen entlang des Signalpfades schrittweise durch Faltung aufsummiert werden.

Um den Rechenaufwand zu reduzieren, können entsprechend der Vorgehensweise der analytischen Berechnung (Kapitel 4.4.4.5) konstante Verzögerungen (Dirac-Verteilungen) zusammengefasst werden. Diese werden nicht numerisch, sondern am Ende durch Verschiebung der Verteilung nach rechts verrechnet.

4.4.5.3 Beispiel: Faltung von zwei Rechteckverteilungen

Die Faltung von zwei Rechteckverteilungen wurde im Abschnitt 4.4.4.4 analytisch gelöst und wird hier beispielhaft numerisch umgesetzt. Ausgangspunkt sind die beiden Rechteck-Verteilungen h_1 und h_2 , deren mögliche Verzögerungen in den Intervallen $[2 \text{ ms}; 6 \text{ ms})$ und $[3 \text{ ms}; 5 \text{ ms})$ liegen.

Entsprechend Gleichung (16) ergeben sich folgende kontinuierliche Wahrscheinlichkeitsdichtefunktionen:

$$h_1(t) = \begin{cases} 1/4 & \text{für } 2 \text{ ms} \leq t < 6 \text{ ms} \\ 0 & \text{sonst} \end{cases} \quad (139)$$

$$h_2(t) = \begin{cases} 1/2 & \text{für } 3 \text{ ms} \leq t < 5 \text{ ms} \\ 0 & \text{sonst} \end{cases} \quad (140)$$

Diese Funktionen müssen zunächst diskretisiert werden, dabei wird eine zeitliche Schrittweite von $\Delta t = 0,1 \text{ ms}$ und ein Definitionsbereich von $t \in [0; 15]$ gewählt. Zwischen der diskreten Veränderlichen k und der kontinuierlichen Veränderlichen t gilt daher folgender Zusammenhang:

$$k = \frac{t}{\Delta t} \quad (141)$$

$$k \in \{0; 1; \dots; 150\} \quad (142)$$

Es ergeben sich folgende diskrete Verteilungen, welche in Abbildung 85 dargestellt sind:

$$h_1(k) = \begin{cases} 1/4 & \text{für } 20 \leq k < 60 \\ 0 & \text{sonst} \end{cases} \quad (143)$$

$$h_2(k) = \begin{cases} 1/2 & \text{für } 30 \leq k < 50 \\ 0 & \text{sonst} \end{cases} \quad (144)$$

³⁴ The MathWorks: MATLAB [The15]

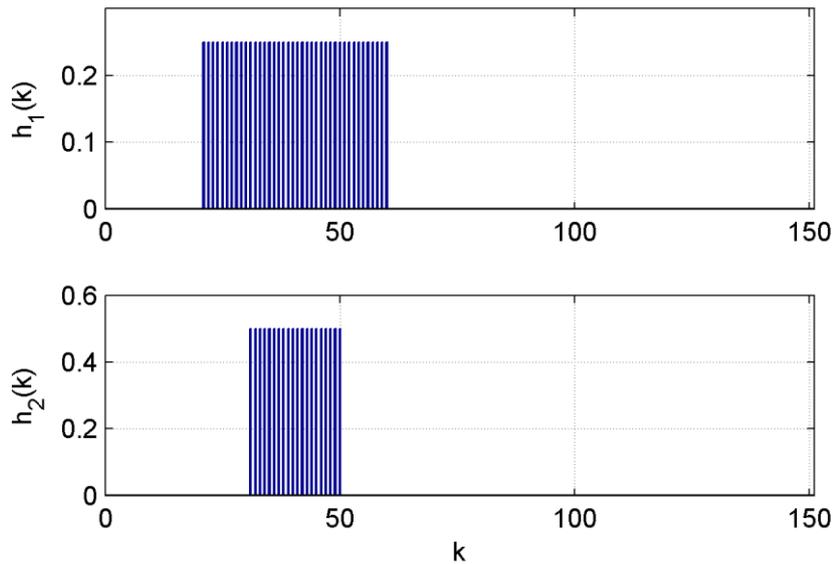


Abbildung 85: Diskrete Rechteckverteilungen

Die diskrete Faltungsoperation nach Gleichung (138) liefert eine Verteilung, wie in Abbildung 86 dargestellt.

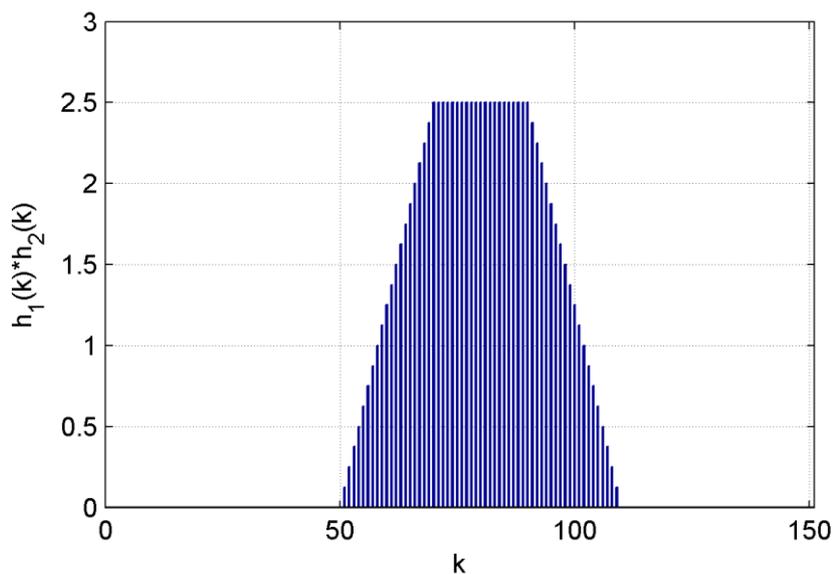


Abbildung 86: Diskrete Faltung von zwei Rechteckverteilungen

Zum besseren Vergleich mit der analytischen Berechnung wird die Faltung nochmals mit einer Schrittweite von $\Delta t = 0,01 \text{ ms}$ durchgeführt. In Abbildung 87 sind Eingangs- und Ergebnisverteilungen der diskreten Faltung mit der Zeit als Abszisse (nach Gleichung (141)) dargestellt.

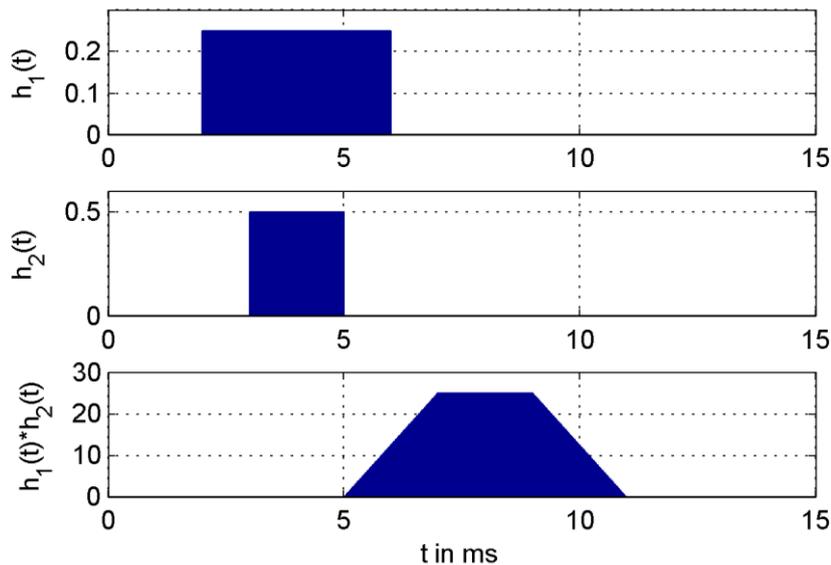


Abbildung 87: Diskrete Faltung von zwei Rechteckverteilungen mit Schrittweite $\Delta t=0,01$ ms

Nach der analytischen Herleitung in Kapitel 4.4.4.4 entsteht als Ergebnis eine Trapezverteilung mit den Knickpunkten:

- $b_1 = a_1 + a_2 = 5$ (positiver Knick)
- $b_2 = a_1 + a_2 + T_1 = 7$ (negativer Knick)
- $b_3 = a_1 + a_2 + T_2 = 9$ (negativer Knick)
- $b_4 = a_1 + a_2 + T_1 + T_2 = 11$ (positiver Knick)

Die numerische Faltung liefert bei ausreichend geringer Schrittweite eine gute Approximation dieser Häufigkeitsverteilung.

4.4.6 Fazit

Aufbauend auf der Überführung des Modells des Steuerungssystems und seiner Signalpfade in eine Verkettung von Verzögerungszeiten, wurde ein Weg zur direkten Berechnung der resultierenden Häufigkeitsverteilung aus den Häufigkeitsverteilungen der Teilverzögerungen gezeigt. Dabei wurden zunächst drei typische Verteilungen für Verzögerungszeiten vorgestellt. Diese Verteilungen können durch Faltung aufsummiert werden. Hierfür kann der analytische Weg mittels Laplace-Transformation angewendet werden. Es wurden Berechnungen für die Faltungen der typischen Häufigkeitsverteilungen gezeigt. Des Weiteren wurde die Möglichkeit der numerischen Berechnung für die Nutzung von gemessenen Häufigkeitsverteilungen aufgezeigt, welche auch für analytisch schwer lösbare Berechnungen verwendet werden kann.

Der hier vorgestellte Weg ermöglicht eine vollständig analytische Berechnung des Zeitverhaltens von Signalpfaden durch ein verteiltes Steuerungssystem, welches durch das in Abschnitt 4.2 vorgestellte Modell der asynchronen Prozesse beschrieben ist. Durch den analytischen Ansatz kann das Zeitverhalten für einen bestimmten Pfad und eine Parameterkonfiguration sehr schnell berechnet werden. Dies ermöglicht den Vergleich verschiedener Konfigurationen (z. B. unterschiedliche Zykluszeiten) ohne den Zeitaufwand einer Vielzahl von Simulationen. Für die überwiegende Mehrheit der Fälle reichen die hier beschriebenen analytischen Faltungsoperationen aus. Da es sich um mathematische Operationen ohne Beziehung zum Steuerungs- oder Kommunikationssystem handelt, können sie problemlos erweitert werden.

4.5 Analytische Berechnung des Zeitverhaltens mittels Modulo-Arithmetik

Der Ansatz der analytischen Berechnung des Zeitverhaltens eines Signalpfades beruht auf dem Prinzip, die Übertragungszeit eines Datums durch das System zu bestimmten Zeitpunkten zu berechnen. Dazu wurde vom Autor in [HD12, HPD14, HD14] ein Verfahren vorgeschlagen, das mittels Modulo-Arithmetik³⁵ und Rekursion die Klemme-Klemme-Reaktionszeit t_{IORT} zum Zeitpunkt t_0 berechnet.

Da die Anfangslage der asynchronen Teilprozesse zueinander nicht bekannt ist, kann somit keine deterministische Aussage über die Verzögerung in einem realen System zu einem bestimmten Zeitpunkt getroffen werden. Daher muss die Berechnung in einer Monte-Carlo-Simulation³⁶ sehr oft durchgeführt werden um daraus eine Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit zu ermitteln. Im Laufe der Arbeiten hat sich gezeigt, dass dieses Verfahren für alle hier durchgeführten Betrachtungen keine relevanten Vorteile gegenüber der Methode in Kapitel 4.4 bietet. Daher wird auf eine detaillierte Wiedergabe verzichtet und auf die genannten Veröffentlichungen verwiesen.

4.6 Mögliche Erweiterungen des Modells

Das im Rahmen dieser Arbeit entwickelte Modell stellt eine abstrakte Abbildung eines Steuerungssystems dar, welches auf die explizite Modellierung von spezifischen Besonderheiten der Teilsysteme zu Gunsten einer **geringeren Komplexität** verzichtet. Diese Abstraktion ist die Grundlage für die nachfolgend gezeigte analytische Ableitung des Zeitverhaltens. Die im Modell eingeführte Trennung von Prozessen und ausführender Hardware ermöglicht die effiziente Beschreibung des Zeitverhaltens verschiedener Signalpfade durch ein komplexes Steuerungssystem.

Im Rahmen dieser Arbeit wurde der Schwerpunkt auf zyklisch arbeitende Systeme und Teilprozesse gelegt. Das Modell kann auch für die Beschreibung des Zeitverhaltens von ereignisbasierten oder hybriden Systemen genutzt werden. Grundlage dieser Betrachtungen ist die Annahme, dass ereignisbasierte Teilprozesse synchron zum vorherigen Teilprozess arbeiten. Die Warte- und Verarbeitungszeit kann variieren.

Trotz der Vereinfachung lassen sich viele Effekte durch eine geeignete Wahl der Parameter des Modells hinreichend genau abbilden. Im Kapitel 2.7 wurden verschiedene Aspekte genannt, welche im Zusammenhang mit NCS untersucht werden. Bei der Untersuchung von NCS und bei der Modellierung von industriellen Kommunikationssystemen ist der Aspekt **Paketverlust** ein sehr häufiges Thema (siehe Abschnitt 2.7.1.4). Der Fokus dieser Arbeit und des Modells liegt auf kabelgebundenen Industrial-Ethernet-Systemen, in welchen Paketverluste eine untergeordnete Rolle spielen. Paketverlustraten lassen sich nicht direkt im hier gezeigten Modell abbilden, dennoch können diese im Modell betrachtet werden.

Paketverlust kann verschiedene Auswirkungen haben. Wenn ein bestimmter Signalwert nur für einen Kommunikationszyklus vorliegt, ist diese Information durch einen Paketverlust verloren. Bei kontinuierlichen Signalen kann ein Paketverlust jedoch durch das verzögerte Eintreffen der Informationen beim Empfänger angenähert werden. Dies kann durch die Verarbeitungszeit des Teilprozesses Kommunikation abgebildet werden. In Abbildung 88 ist für ein Kommunikationssystem

³⁵ Der Modulo-Operator *mod* liefert den Rest bei der ganzzahligen Division.

³⁶ „Verfahren der stochastischen Simulation zur näherungsweise Bestimmung von mathematischen Größen, die abhängig vom Zufall (Verteilungsfunktionen) sind. Die Zufallszahlen aus dem Zufallsgenerator gehen direkt in die mathematischen Ausdrücke ein.“ [Spr15], siehe auch [Beu07, S. 139ff.] und [BH10]

mit der Verarbeitungszeit t_{process} , der Zykluszeit T_{Cycle} und der Paketverlustrate PLR (packet loss rate) eine Häufigkeitsverteilung der modellierten Verarbeitungszeit dargestellt.

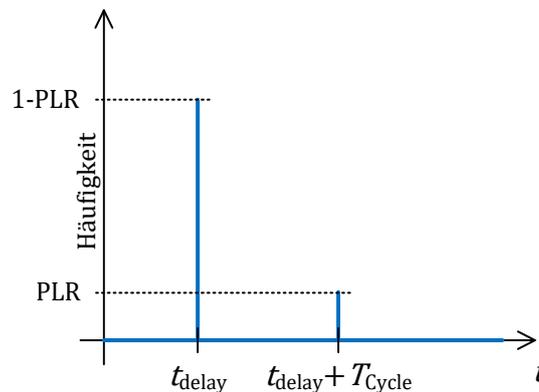


Abbildung 88: Modellierung der Paketverlustrate PLR durch die Verarbeitungszeit

Ein weiterer, im vorliegenden Modell nicht direkt beschriebener Aspekt ist die Wechselwirkung zwischen verschiedenen Signalpfaden oder allgemein die Auswirkung der **Auslastung von Ressourcen**. Die Auslastung einer Ressource bedeutet, dass ein Teilsystem mit der Abarbeitung eines Teilprozesses beschäftigt ist und daher keinen weiteren Teilprozess ausführen kann. Beispielsweise kann ein Kommunikationssystem auf einer Übertragungstrecke in eine Richtung nur ein Telegramm gleichzeitig übertragen. Während der Übertragung ist es ausgelastet.

Die Auslastung der Ressourcen kann näherungsweise durch entsprechende Parametrierung der Verarbeitungszeit von einem oder mehreren Teilprozessen abgebildet werden. Als Erweiterung ist an dieser Stelle die Nutzung von Markow-Ketten zur Beschreibung von verschiedenen Häufigkeitsverteilungen der Verarbeitungszeit, abhängig von der Lastsituation, denkbar (siehe Ansatz von Nilsson in Abschnitt 2.5.3). Durch die Zuordnung von Teilprozessen (und damit Signalpfaden) zu Teilsystemen, sind die notwendigen Informationen über die Auslastung im Modell bereits enthalten (siehe Abschnitt 4.2.4).

4.7 Fazit

In diesem Kapitel wurde ausgehend vom Ansatz der (a)synchron arbeitenden zyklischen Teilprozesse ein Modell zur Beschreibung der Struktur eines Systems und der darin enthaltenen Signalpfade vorgestellt. Dieses Modell ist parametrierbar mit dem Ziel, eine Ableitung des Zeitverhaltens der Signalpfade zu ermöglichen.

Anschließend wurde das Zeitverhalten bei der Übergabe von Daten zwischen verschiedenen Teilprozessen intensiv untersucht, um die auftretenden Verzögerungen in und zwischen den Teilprozessen zu quantifizieren. Mit Hilfe dieser Informationen kann das Zeitverhalten des Systems entlang der Signalpfade mittels Faltung der Häufigkeitsverteilungen berechnet werden. Dafür steht neben dem analytischen Ansatz auch ein numerischer Berechnungsweg zur Verfügung. Dieser erlaubt die Verwendung von Messdaten über das Zeitverhalten von Teilprozessen.

Durch diesen Ansatz kann der qualitative und quantitative Verlauf der Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit eines Signalpfades eines Steuerungssystems direkt berechnet werden.

5 Isochrone Steuerungssysteme

Isochrone Steuerungssysteme wurden in Kapitel 2.1.1 definiert als Systeme, deren Klemme-Klemme-Reaktionszeit konstant ist und deren Ein- und Ausgabe isochron sind. Im Kontext der bisherigen Betrachtungen gilt damit für diese Systeme die Anforderung, dass die Ein- und Ausgabeprozesse zeitlich synchronisiert arbeiten. Um dies zu erreichen, müssen alle zeitlichen Unsicherheiten innerhalb eines Signalpfades durch geplante Wartezeiten ausgeglichen werden.

Während die Synchronisation von Ein- und Ausgabebaugruppen hinreichend für Isochronität sein kann, werden die Klemme-Klemme-Reaktionszeiten durch die zum Ausgleich der asynchronen Übergänge zwischen den anderen Teilprozessen notwendigen Wartezeiten sehr groß. Dies widerspricht den Anwendungsfeldern solcher Systeme (Regelung und Steuerung mit hohen dynamischen Anforderungen). Daher wird neben der Forderung nach Isochronität auch eine möglichst kurze Klemme-Klemme-Reaktionszeit angestrebt. Allgemein lässt sich feststellen, je mehr Teilprozesse des Signalpfades synchronisiert sind (und je höher die Synchronisationsgenauigkeit dabei ist), desto kürzere Klemme-Klemme-Reaktionszeiten können erreicht werden.

5.1 Umsetzungskonzepte von isochronem Verhalten

Für die Betrachtung der verschiedenen Umsetzungskonzepte von isochronem Verhalten soll wie in [HKD14] die Kopplung zwischen den Teilprozessen genauer untersucht werden. Es werden drei Arten der Kopplung von Teilprozessen unterschieden:

- Keine Kopplung
- Lose Kopplung
- Starre Kopplung

Keine zeitliche Kopplung wurde als asynchroner Zyklusübergang ausführlich in Abschnitt 4.1.1 betrachtet, hierbei gibt es keine zeitliche Kopplung zwischen den Teilprozessen. Dies ist aufgrund der entstehenden stochastischen Wartezeiten am Übergang zwischen den Teilprozessen ungeeignet für den Aufbau schneller isochroner Steuerungssysteme.

Bei der **starren zeitlichen Kopplung** von zwei Teilprozessen gibt es eine feste zeitliche Beziehung zwischen diesen. Diese Kopplung setzt bei zyklisch arbeitenden Prozessen voraus, dass die Zeitbasen der beiden Teilsysteme synchronisiert sind. Die Zeit $t_{\text{transition}}$ zwischen dem Ende des vorigen Teilprozesses und dem Beginn des nachfolgenden Teilprozesses ist konstant. Im Sinne des im vorigen Kapitel vorgestellten Modells stellt die starre Kopplung eine Synchronisation mit hoher Präzision dar.

Ein Mittelweg stellt die sog. **lose zeitliche Kopplung** dar. Die Zeitbasen der beiden Teilprozesse sind in diesem Fall synchronisiert, aber mit geringerer Präzision. Entsprechend der Betrachtungen in Abschnitt 4.1.1 ist eine kurzzeitige Abweichung der Zykluszeiten erlaubt, es darf aber keine globale Drift geben. Dies hat zur Folge, dass kein Zyklussprung stattfindet, d. h. in einem Zeitzyklus werden immer alle beteiligten Teilprozesse einmal ausgeführt.

Die Zeitdauer für den Übergang (stochastische Wartezeit) zwischen den Teilprozessen ist damit nicht konstant, aber auf einen Wert kleiner als die Zykluszeit begrenzt: $t_{\text{transition}} < T_{\text{Cycle}}$. Für isochrones Verhalten muss eine geplante Wartezeit eingefügt werden, um die stochastische Wartezeit zwischen den Teilprozessen auf den größtmöglichen Wert von $t_{\text{transition}}$ aufzurunden.

Ein Vorteil der losen Kopplung ist, dass auf eine sehr exakte Zeitsynchronisation verzichtet werden kann und sich damit der Aufwand reduziert. Nachteilig ist, dass die minimal erreichbare Klemme-Klemme-Reaktionszeit steigt, da die zeitliche Unsicherheit durch Wartezeiten ausgeglichen werden muss. Im Sinne des im vorigen Kapitel vorgestellten Modells stellt die lose Kopplung eine Synchronisation mit geringer Präzision dar.

Die kleinsten Klemme-Klemme-Reaktionszeiten bei isochronen Steuerungssystemen können erreicht werden können, wenn alle beteiligten Teilprozesse starr gekoppelt sind, da in diesem Fall keine zusätzlichen geplanten Wartezeiten notwendig sind. Dies setzt die präzise Synchronisation aller beteiligten Teilsysteme voraus. Dieses Konzept wird bei PROFINET IRT verwendet.

Eine weitere Umsetzungsmöglichkeit sieht die starre Kopplung von Ein- und Ausgabebaugruppen und der Steuerungsapplikation vor. Die Teilprozesse für die Kommunikation und die Verarbeitung in den Feldgeräten sind nur lose angekoppelt. Diese Umsetzung sieht beispielsweise EtherCAT Distributed Clocks vor.

5.2 Modellierung isochroner Steuerungssysteme

Im Kontext dieser Arbeit stellen isochrone Steuerungssysteme somit den Sonderfall dar, dass alle an einem Signalpfad beteiligten Teilsysteme und deren Teilprozesse synchronisiert arbeiten. Es treten keine Wartezeiten durch asynchrone Zykluszeiten innerhalb des Signalpfades auf (Ausnahme ist die asynchrone Kopplung an den technischen Prozess, siehe Kapitel 2.1.3).

Während in den bisherigen Betrachtungen und Berechnungen die Präzision und die Genauigkeit aufgrund der Tatsache, dass beide um Größenordnungen kleiner sind als die Zykluszeit der Teilprozesse, weitestgehend vernachlässigt wurden, müssen sie hier genauer untersucht werden.

Zur Abbildung dieser isochronen Steuerungssysteme auf das hier verwendete Modell werden alle Teilsysteme (und damit alle Teilprozesse) einer Zeitdomäne zugeordnet, siehe Beispiel in Abbildung 89.

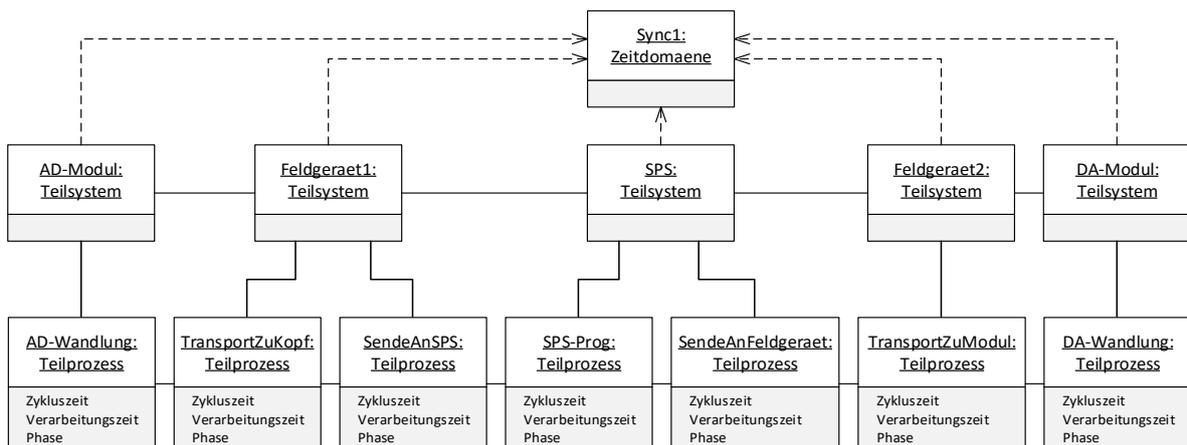


Abbildung 89: Beispielmodell für isochrones Steuerungssystem

Bei der Parametrierung des Modells erhält neben den Parametern Zykluszeit (welche in der Regel für alle Teilprozesse gleich ist) und Verarbeitungszeit vor allem die Phase eine große Bedeutung. Sie bildet die zeitliche Lage der Teilprozesse zueinander ab. Entsprechend der hier getroffenen Konventionen werden geplante Wartezeiten im Modell der Verarbeitungszeit des jeweiligen Teilprozesses

zugeordnet. Da die lose Kopplung von Teilprozessen durch Wartezeiten ausgeglichen wird, wird dieser Aspekt im Modell berücksichtigt.

Zur Ableitung des Zeitverhaltens eines Signalpfades werden die synchronen Teilsysteme entsprechend der Regeln in Kapitel 4.3 zusammengefasst. Während diese Modellierung für die Berechnung des Zeitverhaltens hinreichend ist (Anwendersicht), kann für die konkrete Auslegung und Parametrierung synchroner Steuerungssysteme eine genauere Betrachtung sinnvoll sein (Herstellersicht).

5.3 Spezielles Modell für isochrone Steuerungssysteme

In [HDD+13] und [DLK+14] wurde zu diesem Zweck ein sogenanntes „IORT-Modell“ für die Analyse von synchronen Steuerungssystemen vorgeschlagen, welches hier erläutert werden soll.

5.3.1 Allgemeines Modell

Ausgangspunkt der Betrachtungen sind die Teilprozesse entlang eines Signalpfades und deren zeitliche Anordnung. Die Signalkette lässt sich in fünf wesentliche Teilprozesse aufteilen: Datenerfassung, Kommunikation des Prozesseingangsabbildes (PEA), Steuerungsprogramm, Kommunikation des Prozessausgangsabbildes (PAA) und Datenausgabe.

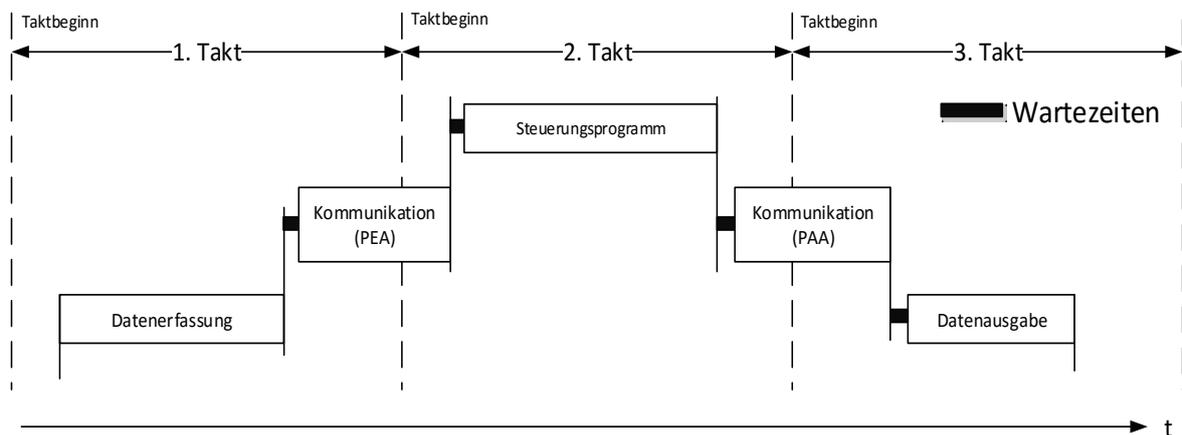


Abbildung 90: Modell für isochrone Steuerungssysteme nach [HDD+13]

Wie beschrieben, wird bei isochronen Steuerungssystemen davon ausgegangen, dass alle Teilprozesse die gleiche Zykluszeit haben, es gibt eine Gesamtzykluszeit. Diese ist im Modell in Abbildung 90 in Form der Takte dargestellt. Die Darstellung des Modells geht von der allgemeinen Annahme aus, dass die Kommunikation des PEA und des PAA unabhängig voneinander erfolgen. In konkreten Umsetzungen können diese aneinander gekoppelt sein. In der Darstellung erfolgt die Bearbeitung des Steuerungsprogramms im 2. Takt. Dies bedeutet nicht, wie das Bild suggerieren könnte, dass sie nur in einem von 3 Takten ausgeführt wird. Alle gezeigten Prozesse werden in jedem Takt ausgeführt und sind nur aus Gründen der Übersichtlichkeit für einen Durchlauf des Signalpfades dargestellt.

Der Beginn eines Taktes wurde bewusst nicht an den Start eines Teilprozesses gelegt, da die am Markt verfügbaren Umsetzungen unterschiedliche Zeitpunkte für den Taktbeginn definieren.

Zwischen diesen Teilprozessen treten geplante Wartezeiten auf, die verschiedene Funktionen erfüllen:

- Ausgleich von Unsicherheiten der Zeitsynchronisation
- Ausgleich von Unsicherheiten der Verarbeitungszeiten (z. B. variierende Abarbeitungszeit des Steuerungsprogramms)
- Ausgleich von verschiedenen Verarbeitungszeiten einzelner Teilprozesse in verschiedenen Signalpfaden (z. B. unterschiedliche Analog-Digital-Wandlungszeiten)

Die Unsicherheiten der Zeitsynchronisation (konkret die Präzision) zwischen den Teilsystemen müssen zum Erreichen von isochronem Verhalten durch Wartezeiten ausgeglichen werden, deren Dauer größer ist als die maximal mögliche Abweichung zwischen den beiden Taktgebern. Gleiches gilt für zeitliche Unsicherheiten der Verarbeitungszeiten der Teilprozesse, diese müssen durch das Anhängen einer Wartezeit mindestens auf den Maximalwert der Verarbeitungszeit verlängert werden. Ist neben Isochronität auch eine Gleichzeitigkeit der Ein- bzw. Ausgänge gefordert, müssen eventuell unterschiedliche Verarbeitungszeiten der verschiedenen Ein- und Ausgangsbaugruppen durch Wartezeiten so verlängert werden, dass sie der größten Verarbeitungszeit aller beteiligten Ein- bzw. Ausgabebaugruppen entsprechen³⁷.

Durch die Darstellung im Taktkontext können neben den Aussagen zu den Wartezeiten weitere Randbedingungen für die Auslegung von synchronen Steuerungssystemen abgeleitet werden. Aufgrund der zyklischen Abarbeitung der Teilprozesse müssen die Verarbeitungszeiten jeweils kleiner sein als die Zykluszeit. Diese Einschränkung tritt in vorhandenen Systemen vor allem hinsichtlich der Verarbeitungszeit des Steuerungsprogramms und der Wandlungszeit der Ein- und Ausgangsbaugruppen auf. Durch die Konkretisierung des Modells auf ein bestimmtes Kommunikationssystem können weitere Bedingungen abgeleitet werden.

³⁷ Die Verarbeitungszeiten der Ein- bzw. Ausgangsbaugruppen müssen bei manchen Systemen nicht auf gleiche Länge gebracht werden, da die unterschiedlichen Laufzeiten der Kommunikationstelegramme bei der Bestimmung der Wartezeiten berücksichtigt werden. So muss eine Baugruppe mit schneller Wandlung, die am Ende des Kommunikationssystems positioniert ist (und damit die höchste Verarbeitungszeit für die Kommunikation hat), nicht auf die langsamste Baugruppe verlängert werden.

5.3.2 Modell von PROFINET IRT

Mit PROFINET IRT existiert ein Kommunikationssystem, welches das geforderte isochrone Verhalten unterstützt. Durch die Integration in ein entsprechendes Steuerungssystem kann damit ein isochrones Steuerungssystem aufgebaut werden. Dieses Verhalten ist in der Spezifikation für das Profil PROFIdrive [PRO06] beschrieben, welches auch ein ähnliches Modell unter dem Begriff 3-T-Modell spezifiziert. Für PROFINET IRT kann das allgemeine Modell konkretisiert werden (siehe Abbildung 91).

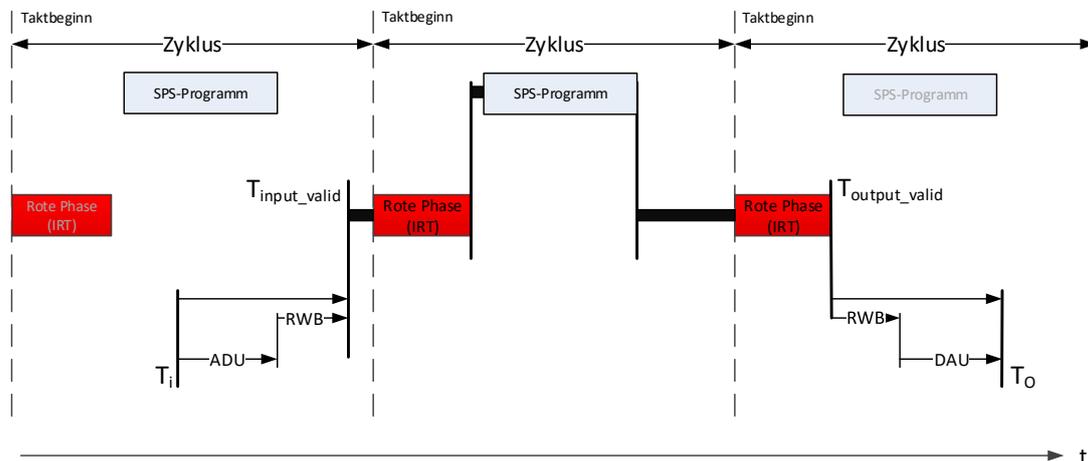


Abbildung 91: 3-T-Modell von PROFINET IRT nach [HDD+13]

Zunächst sieht PROFINET IRT vor, dass die Kommunikation des PEA und des PAA gleichzeitig stattfinden, dies wird als die „Rote Phase“ bezeichnet. Der Referenzpunkt für den Taktbeginn ist der Beginn der roten Phase, d. h. der Beginn der Kommunikation. Bei der Verarbeitungszeit der Ein- und Ausgangsbaugruppen geht das 3-T-Modell von modularen Feldgeräten aus und teilt diese daher in eine Verarbeitungszeit des EA-Moduls (im Bild ADU – Analog-Digital-Umsetzung – bzw. DAU – Digital-Analog-Umsetzung – bezeichnet) und die Zeit für die Kommunikation innerhalb des Feldgerätes (im Bild mit RWB – Rückwandbus – bezeichnet).

Für PROFINET IRT sind die Zeitpunkte T_{input_valid} (alle Eingangsdaten wurden eingelesen und stehen zur Kommunikation bereit) und T_{output_valid} (die Ausgangsdaten liegen in allen Feldgeräten vor) definiert. Daraus resultiert, dass zum Erreichen der Gleichzeitigkeit der Eingänge die Summe aus Wandlungszeit und Verarbeitungszeit im Feldgerät (RWB) durch eine Wartezeit auf den größten auftretenden Wert verlängert werden muss. Gleiches gilt für die Ausgänge.

Aus dem 3-T-Modell lassen sich neben den allgemeinen Bedingungen (Wartezeiten zum Ausgleich der Synchronisationsfehler, alle Verarbeitungszeiten sind kleiner als die Zykluszeit) weitere Randbedingungen ableiten.

- Die Zeit für das Einlesen und Bereitstellen der Eingänge (T_i) muss kleiner sein als die Zykluszeit
- Die Zeit für das Bereitstellen und Ausgeben der Ausgänge (T_o) muss kleiner sein als die Zykluszeit
- Die Summe aus Verarbeitungszeit des Kommunikationssystems (Rote Phase) und des Steuerungsprogramms muss kleiner sein als die Zykluszeit

Die starre Kopplung aller beteiligten Teilsysteme wird bei PROFINET IRT durch Taktsynchronisation mit einer Synchronisationsgenauigkeit kleiner als eine Mikrosekunde erreicht [DIN15, S. 111].

5.3.3 Modell von EtherCAT Distributed Clocks

EtherCAT Distributed Clocks setzt das isochrone Verhalten durch Zeitsynchronisation von Datenerfassung, Steuerungsprogramm und Datenausgabe um. Die Kommunikation ist lose gekoppelt. Der zeitliche Referenzpunkt in dieser Realisierung ist der Start der Datenausgabe. Abbildung 92 zeigt die Konkretisierung des Modells für EtherCAT nach [KDL13].

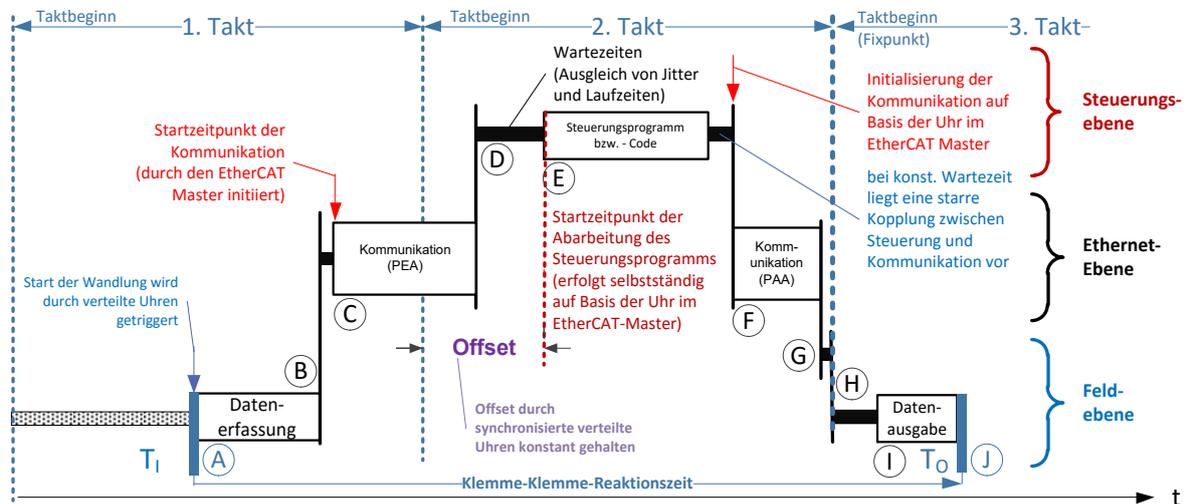


Abbildung 92: IORT-Modell für EtherCAT [KDL13]

Der Verzicht auf die starre Kopplung aller Teilsysteme vereinfacht den Aufbau des Systems. Er setzt aber, wie oben betrachtet, das Einfügen entsprechender Wartezeiten zur Kompensation der zeitlichen Unsicherheiten voraus. Für eine detailliertere Analyse von EtherCAT sei auf [KDL13] verwiesen.

5.4 Vergleich isochrones Modell und Modell asynchroner Prozesse

Das im vorigen Abschnitt gezeigte spezielle Modell für isochrone Steuerungssysteme stellt keinen Widerspruch zu dem in dieser Arbeit entwickelten Modell der asynchronen Prozesse (siehe Abschnitt 4.2) dar.

In Abbildung 93 wird das allgemeine isochrone Modell (Abbildung 90) mit dem Beispiel für das Modell der asynchronen Prozesse (Abbildung 89) verglichen.

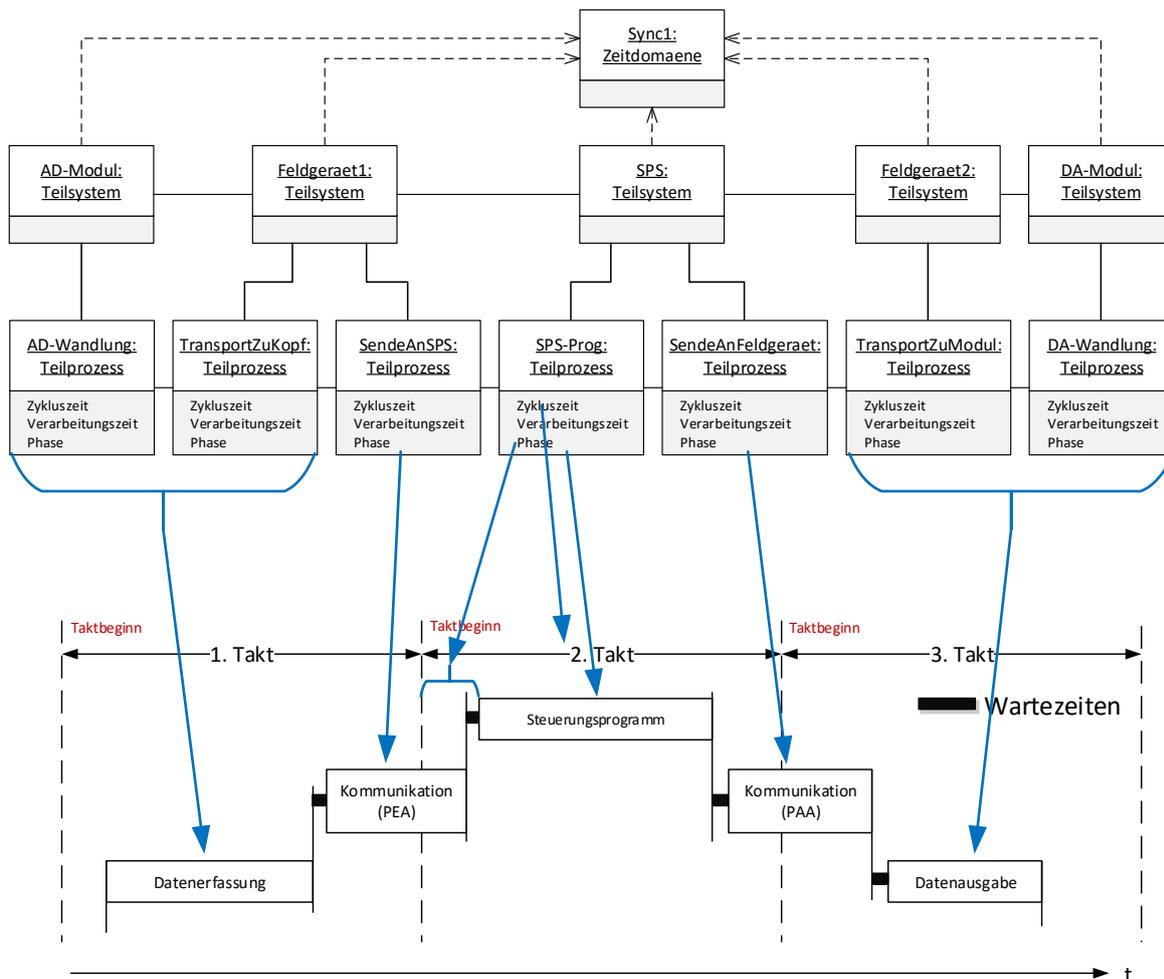


Abbildung 93: Zuordnung des isochronen Modells zum Modell der asynchronen Prozesse

Im isochronen Modell werden Teilprozesse der Signalkette abgebildet. Die Zykluszeit der Teilprozesse entspricht der Taktdauer im isochronen Modell. Die Phasenlage jedes Teilprozesses wird auf der Zeitachse abgebildet. Diese Zuordnung wurde aus Übersichtlichkeitsgründen nur für das SPS-Programm eingezeichnet, gilt aber für alle Teilprozesse.

Da das allgemeine isochrone Modell nicht von modularen Feldgeräten ausgeht, sind die EA-Wandlung und der Datentransport im Gerät nicht separat ausgewiesen.

Die Zuordnung kann auf für die speziellen Ausprägungen des Modells durchgeführt werden. Abbildung 94 zeigt sie für das 3-T-Modell von PROFINET IRT aus Abschnitt 5.3.2.

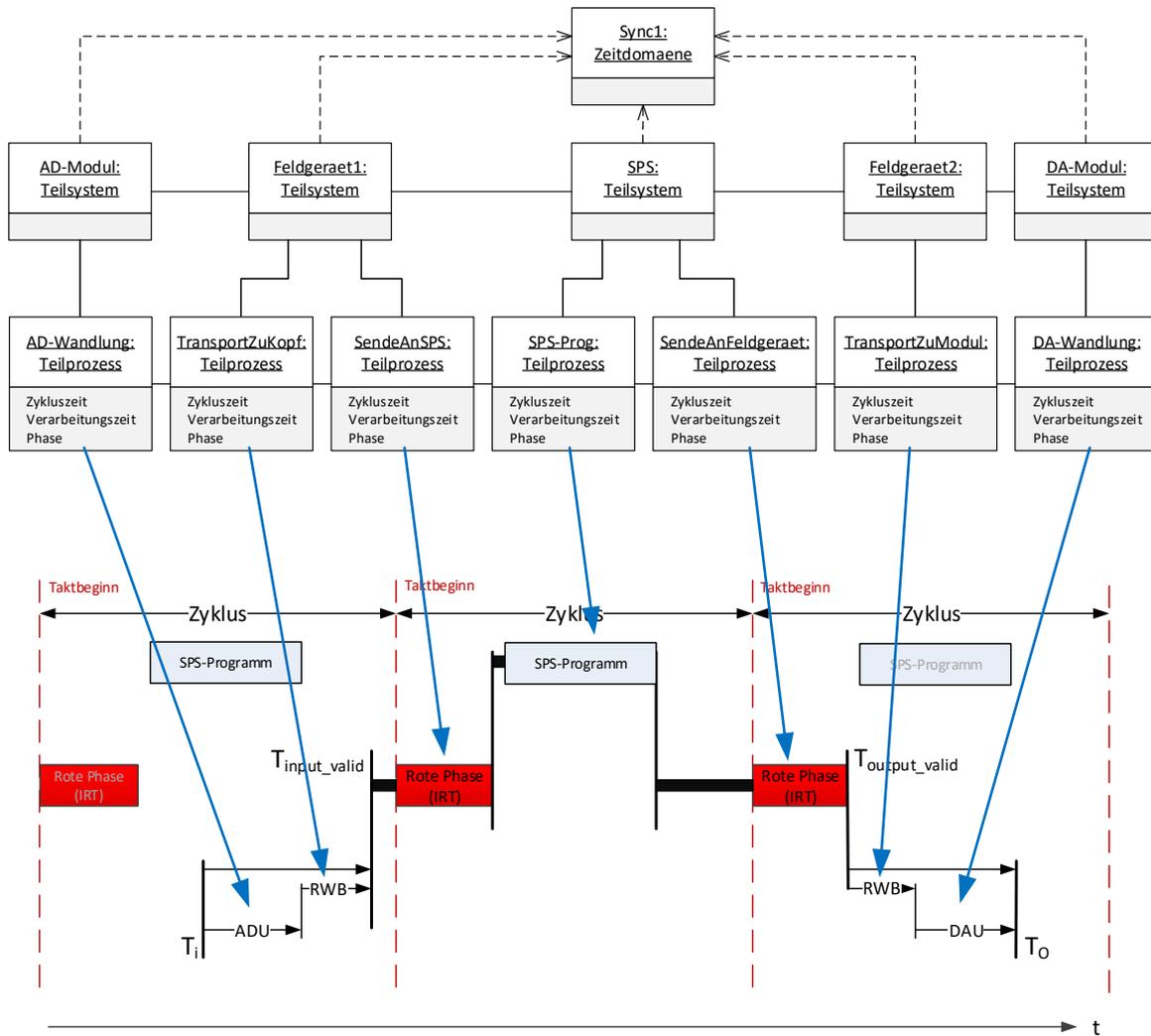


Abbildung 94: Zuordnung des 3-T-Modells zum Modell asynchroner Prozesse

In der Darstellung wurde auf die Zuordnung von Zykluszeit und Phasenlage verzichtet. Es gelten jedoch die gleichen Aussagen wie beim allgemeinen Modell.

5.5 Fazit

Es konnte gezeigt werden, dass isochrone Steuerungssysteme im Modell der asynchronen Prozesse abgebildet und mit diesem berechnet werden können. Zur genauen Analyse der Umsetzungen und damit der Auswirkungen von starrer und loser zeitlicher Kopplung (d. h. unterschiedlicher Präzision der Synchronisation) wurde ein spezielles IORT-Modell eingeführt.

6 Parametrierung der Modelle

In Kapitel 4 wurde das Modell der asynchronen Prozesse und die darin enthaltenen Parameter beschrieben. Alle Parameter in dem Modell müssen für eine konkrete Modellierung bestimmt werden, d. h. ihnen müssen Werte zugewiesen werden. In diesem Kapitel sollen Wege zur Ermittlung der Parameter erläutert werden.

6.1 Parameter

Zunächst sollen die im Modell enthaltenen Parameter (Definition der Parameter siehe 4.2.3) und mögliche Quellen der Werte gezeigt werden.

6.1.1 Zykluszeit

Die Zykluszeit eines jeden Teilprozesses ist ein Wert, der im Zug der Entwicklung der Komponente (feste Zykluszeit) oder während des Engineerings eines Systems bestimmt wird. Die Wahl des Parameters obliegt, im Rahmen definierter Grenzen, dem Entwickler des Systems oder der Anlage.

Ein hartes Kriterium für die Wahl einer Zykluszeit ist die Tatsache, dass die Abarbeitung der Aufgabe vor dem Beginn eines neuen Zyklus abgeschlossen sein muss. Es gilt also:

$$T_{\text{Cycle},i} \geq t_{\text{process},i} \quad (145)$$

Die Bestimmung des Parameters Zykluszeit für die jeweiligen Elemente des Modells muss anhand von Projektierungsdaten (z. B. Zykluszeit des Kommunikationssystems oder des Steuerungsprogramms) oder auch aus Datenblättern (z. B. Zykluszeit von Wandler-Baugruppen) erfolgen.

Die Zykluszeiten von Teilprozessen sind Parameter, die zur Optimierung des Zeitverhaltens eines Gesamtsystems verändert werden können. Dabei bestimmt der Gerätehersteller durch Auswahl und Dimensionierung die Grenzwerte für die Zykluszeit eines Gerätes. Der Anwender kann sie innerhalb dieses Rahmens passend für seine Anforderungen wählen oder muss ein anderes Gerät einsetzen, falls die Anforderungen nicht erfüllt werden.

6.1.2 Verarbeitungszeit

Die Verarbeitungszeit beschreibt, wie lange ein Teilprozess für die Abarbeitung seiner Funktion benötigt. Dieser Parameter ist im Gegensatz zur Zykluszeit nicht direkt vom Anwender bestimmbar (mit Ausnahme des SPS-Programms). Er ergibt sich aus der Hard- und Software eines Teilprozesses und des ausführenden Teilsystems.

Trotzdem hat der Anwender eventuell Einflussmöglichkeiten auf die Größe der Verarbeitungszeit. Zum einen kann die Verarbeitungszeit bewusst um geplante Wartezeiten verlängert werden, um damit eventuelle zeitliche Unsicherheiten auszugleichen. Zum anderen erlauben beispielsweise Analog-Digital-Wandler teilweise eine Verringerung der Verarbeitungszeit zu Lasten einer geringeren Wandlungsgenauigkeit. Somit kann die Präzision des Signalwertes gegen die Verarbeitungszeit abgewogen werden.

Abgesehen von solchen Sonderfällen kann die Verarbeitungszeit in der Regel nur durch Veränderung der Systemarchitektur (z. B. Lastreduktion in Abschnitten des Kommunikationssystems) oder den

Einsatz leistungsfähigerer Komponenten verringert werden. Da der Preis für leistungsfähigere Komponenten in der Regel höher ist, muss hier Aufwand und Nutzen abgewogen werden. Dabei soll das in dieser Arbeit entwickelte Modell unterstützen.

6.1.3 Phase

Die Phase gibt den Start der Verarbeitung eines Teilprozesses bezogen auf den Zyklusbeginn an. Der Parameter Phase ist nur bei der Berechnung des Zeitverhaltens von synchronen Teilprozessen von Bedeutung. Da er ähnlich wie die Zykluszeit durch den Entwickler des Systems bestimmt wird, kann sein Wert ebenfalls nur aus Projektierungsdaten oder Datenblättern bestimmt werden.

Da eine ungünstig gewählte Phase zu einer Verschlechterung des Zeitverhaltens von synchronen Systemen führen kann (Zyklussprung, siehe 4.3.3.1), kann die Phase zur Optimierung des Zeitverhaltens eines Gesamtsystems genutzt werden.

6.1.4 Genauigkeit und Präzision der Zeitdomäne

Die Parameter Genauigkeit und Präzision hängen von der erreichbaren Synchronisationsgüte des eingesetzten Verfahrens und der Hardware ab. Im Kapitel 2.4 und in [HKD14] wurden verschiedene Verfahren und die damit typischerweise erreichbaren Genauigkeiten beschrieben. Die Werte der Parameter Genauigkeit und Präzision hängen neben dem Verfahren auch von dessen konkreter Umsetzung ab.

Beide Parameter stellen „Worst-Case“-Werte dar und sind somit theoretisch durch Betrachtungen zur Fehlerfortpflanzung ausgehend von den Ungenauigkeiten der Taktgeber (Quarze) sowie der beteiligten Hard- und Softwarekomponenten bestimmbar. Für die Modellparametrierung wird hier, wenn möglich, auf Aussagen der Datenblätter und Handbücher von Komponenten zurückgegriffen, in denen Hersteller die erreichbare Präzision und Genauigkeit angeben.

6.2 Bestimmung der Verarbeitungszeit

Während die Parameter Zykluszeit und Phase vom Entwickler oder Anwender definierte Werte sind, hängt die Verarbeitungszeit von der eingesetzten Hard- und Software ab. Die Bestimmung der Verarbeitungszeit eines Teilprozesses kann abhängig vom Typ des Teilsystems schwierig sein.

Die erste Schwierigkeit liegt in der genauen Abgrenzung der Teilsysteme und -prozesse. Beispielsweise kann der Zeitpunkt der Übergabe von Daten vom Steuerungsprogramm zum Kommunikationssystem bei Aufruf entsprechender Senderoutinen im Anwenderprogramm, bei der Datenübergabe an den Kommunikationsstack, bei der Übergabe an die Kommunikationshardware (ISO/OSI-Schicht 1 oder bei Ethernet z. B. MII) oder beim Versenden eines Paketes auf dem Kommunikationsmedium liegen. Wichtig ist hierbei eine einheitliche Definition innerhalb der Systembeschreibung.

6.2.1 Ein- und Ausgabebaugruppen

Die Verarbeitungszeit von Ein- und Ausgabebaugruppen lässt sich oftmals aus entsprechenden Datenblättern ermitteln. Dabei ist zu beachten, dass zu den reinen Wandlerzeiten häufig noch Laufzeiten für Filterung u. ä. addiert werden müssen. Im Interesse einer einheitlichen Definition bietet sich die Nutzung der Kenngröße Übergangszeit (siehe Kapitel 3.7) an.

6.2.2 Kommunikationssysteme

Die Verarbeitungszeit eines Kommunikationssystems entspricht in der Regel der Laufzeit eines Telegramms auf dem Medium zuzüglich der benötigten Zeiten für Senden und Empfangen. In Kapitel 3.6 wurde zu diesem Zweck die Kenngröße Übertragungszeit definiert, welche sich als Summe einer feingranularen Betrachtung herleiten lässt.

Für die Bestimmung des Parameters muss zwischen geplanten und ungeplanten Kommunikationssystemen unterschieden werden. Deterministische, geplante Kommunikationssysteme (z. B. PROFINET IRT) haben ein vorgegebenes Zeitverhalten. Dieses wird durch Offline-Scheduling der Telegramme (siehe z. B. [BHS10] [WSJ+12]) vorgegeben. Die Verarbeitungszeiten können in diesem Fall aus den Scheduling-Ergebnissen extrahiert werden. Eine mögliche Quelle ist hier das Engineering-System.

Bei ungeplanter Kommunikation (z. B. PROFINET RT) ist eine detaillierte Analyse notwendig. Derartige Zeiten werden mit Modellen zur Berechnung von industriellen Kommunikationssystemen ermittelt, siehe Abschnitt 2.5. Teilweise können Abschätzungen auch von den Engineering-Systemen der Steuerungshersteller vorgenommen werden.

6.2.3 Steuerungen

Die Verarbeitungszeiten von Steuerungen werden durch das Lesen des Prozesseingangsabbildes, die Ausführungsdauer des Steuerungsprogramms und das Schreiben des Prozessausgangsabbildes bestimmt, siehe auch Kenngröße in Kapitel 3.9.

Die Bestimmung der Ausführungsdauer ist bei umfangreichen Programmen schwierig, es kann jedoch eine Abschätzung anhand der Anzahl der Operationen und der Herstellerangaben zur Dauer der Operationen auf der vorliegenden Hardware erfolgen. Je nach Arbeitsweise des Steuerungssystems ist die genaue Kenntnis der Laufzeit des Programms nicht notwendig, da das Schreiben der Ausgangsdaten mit dem Beginn des nächsten Zyklus erfolgt. In diesem Fall muss nur die Bedingung erfüllt sein, dass die Ausführungszeit des Steuerungsprogramms inkl. Lesen und Schreiben des Prozessabbildes kleiner ist als die Zykluszeit der Kommunikation. Die Verarbeitungszeit des Steuerungssystems entspricht dann genau einem solchen Zyklus.

6.2.4 Granularität von Verarbeitungszeiten

Die Bestimmung der Verarbeitungszeit jedes Teilprozesses eines Signalpfades kann sehr aufwendig sein. Wie in Abschnitt 4.4.3 beschrieben, ist die Berechnung der resultierenden Zeitverzögerung kommutativ. Aufgrund dieser Eigenschaft können konstante Verarbeitungszeiten von mehreren Teilsystemen zusammengefasst werden. Für die Parametrierung des Modells bedeutet dies, dass die konstanten Verzögerungen von mehreren Teilprozessen nicht separat bestimmt werden müssen, sondern als Summenparameter vorliegen können. So können z. B. für eine Eingabebaugruppe die Laufzeit der Signalwandlung und die des Kommunikationsprozesses (Versenden der Messdaten) als ein Parameter vorliegen. Speziell bei der messtechnischen Bestimmung von Parametern ist dieses Vorgehen von Bedeutung.

6.3 Messtechnische Bestimmung von Parametern

Die Messung von Zeiten an realen Systemen kann für die Parametrierung des Modells notwendig sein, wenn nicht alle Parameter aus Datenblätter oder Planungswerkzeugen bestimmbar sind. Dabei muss zwischen der Messung von einzelnen Teilsystemen, d. h. der direkten Bestimmung von Parametern, und der Messung von einer Kette von Teilsystemen oder dem Gesamtsystem unterschieden werden. In letzterem Fall müssen die einzelnen Parameter aus den Messdaten abgeleitet werden.

6.3.1 Messprinzip Klemme-Klemme-Reaktionszeit

Wird das Steuerungssystem als Black-Box betrachtet, so kann eine Messung der Klemme-Klemme-Reaktionszeit erfolgen. Das System wird dazu als SISO-System mit einem Eingang und einem Ausgang abstrahiert. Für diese Messung muss das System so programmiert sein, dass es den Eingangswert direkt auf den Ausgang kopiert.

Gibt man jetzt ein Testsignal auf den Eingang des Systems, kann man die Verarbeitungszeit durch Vergleich des Ein- und Ausgangssignals ermitteln, siehe Abbildung 95.

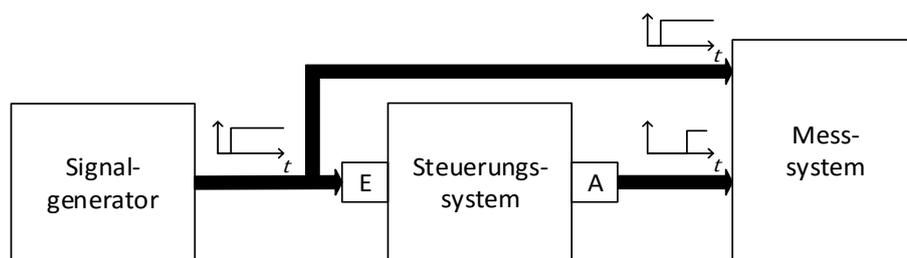


Abbildung 95: Messaufbau zur Bestimmung der Klemme-Klemme-Reaktionszeit nach [HD12]

Die Wahl des Testsignals beeinflusst das Ergebnis der Messung. Nach Abschnitt 2.1.3 und 4.3.6 entsteht bei der Nutzung eines Zustandsdiskreten Testsignals ein Zyklusübergang zwischen Testsignal und Steuerungssystem. Dadurch beinhaltet die Klemme-Klemme-Reaktionszeit bei einem Zustandsdiskreten Testsignal eine gleichverteilte Verzögerungszeit, die zwischen Null und der Zykluszeit des ersten Teilsystems liegt.

Dies kann durch die Nutzung eines Zustandskontinuierlichen Testsignals verhindert werden, hier wird die Klemme-Klemme-Reaktionszeit des Systems ohne den asynchronen Zyklusübergang bei der Prozessanbindung gemessen. Jedoch ist die praktische Umsetzung einer solchen Messung deutlich aufwendiger.

6.3.1.1 Zustandsdiskretes Testsignal

Als Zustandsdiskretes Testsignal eignet sich am besten ein Rechtecksignal, dessen Zykluszeit deutlich größer ist als die größte zu erwartende Zykluszeit des Systems. Entsprechend dem Aufbau in Abbildung 95 wird das Testsignal auf einen Eingang des zu betrachtenden Steuerungssystems gegeben und die Kopie des Eingangssignals wird vom Steuerungssystem ausgegeben. Beide Signale werden mit einem Messsystem aufgenommen.

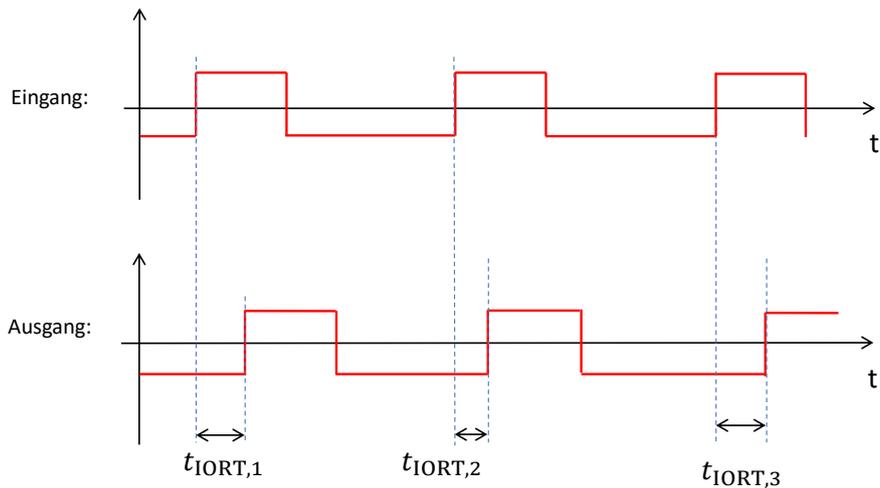


Abbildung 96: Bestimmung der Klemme-Klemme-Reaktionszeit mit Rechteck-Testsignal

Das Messsystem detektiert wahlweise steigende oder fallende Flanken³⁸ in den beiden Signalen und speichert diese mit einem Zeitstempel. Die Genauigkeit dieser Zeitstempelung bestimmt die erreichbare Genauigkeit des Messvorgangs. Zur Bestimmung der Klemme-Klemme-Reaktionszeit t_{IORT} des Systems wird der zeitliche Abstand zwischen einer steigenden Flanke am Eingang und einer steigenden Flanke am Ausgang. Das Ergebnis ist eine Menge von Werten $t_{IORT,i}$, welche anschließend in eine Häufigkeitsverteilung übertragen werden können.

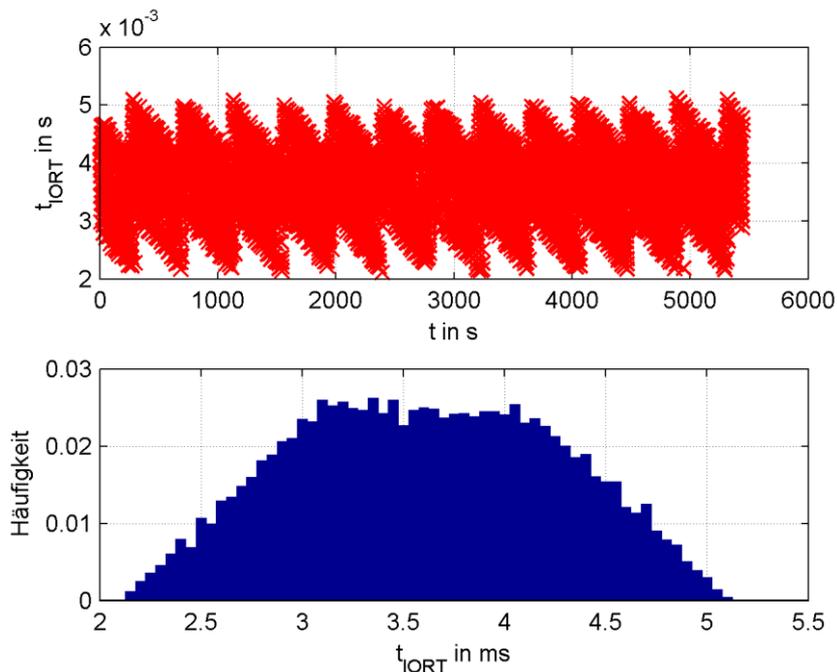


Abbildung 97: Beispiel einer Klemme-Klemme-Reaktionszeit-Messung mittels Rechtecksignal

In Abbildung 97 ist das Ergebnis einer solchen Messung beispielhaft als zeitlicher Verlauf (oben – rot) und als Häufigkeitsverteilung (unten – blau) dargestellt.

Der Vorteil dieses Messaufbaus liegt zum einen darin, dass das Messsystem lediglich steigende bzw. fallende Flanken detektieren und diese zeitstempeln muss. Es ist keine aufwändige Analog-Digital-

³⁸ Im Folgenden wird nur noch die steigende Flanke erwähnt, alle Betrachtungen gelten auch für fallende.

Wandlung mit hoher Geschwindigkeit erforderlich. Des Weiteren kann diese Art der Messung sowohl mit analogen als auch mit digitalen Ein- und Ausgangsmodulen am Steuerungssystem durchgeführt werden.

Die Nachteile dieses Messaufbaus liegen in dem Prinzip bedingten Jitter am Eingang des Steuerungssystems. Da dieser die Breite von einem Zyklus der Eingangsbaugruppe hat, kann er, soweit bekannt, rechnerisch entfernt werden. Es wäre denkbar, die Zykluszeit des Testsignals auf die Zykluszeit der Eingangsbaugruppe zu synchronisieren, wodurch der Jitter vermieden würde. Dies ist jedoch mangels Zugang zu diesem Signal in der Eingangsbaugruppe selten möglich.

Ein weiteres Problem tritt bei der Nutzung von Analogbaugruppen auf. Das Rechteck-Testsignal hat nur eine begrenzte Anstiegsgeschwindigkeit, es ist genau betrachtet ein Trapezsignal. Daher kann die Eingangsbaugruppe das Signal während des Anstiegs abtasten, so dass das Ausgangssignal eine Stufe erhält. Das Messsystem erfasst nur steigende Flanken, dazu wird das Signal in ein digitales Signal gewandelt. Ab einem definierten Pegel wird das Signal als 1 erfasst. Wenn im Ausgangssignal eine Stufe knapp unter dem Schwellwert auftritt, kann der Schwellwert durch das Messrauschen überschritten werden. Gleiches kann bei der Wandlung des Eingangssignales auftreten, auch hier kann das Messrauschen eine „verfrühte“ Detektion der steigenden Flanke bewirken.

Dies führt zu einer Verfälschung des Messergebnisses. Dies ist bei der Messung von isochronen Systemen in der Häufigkeitsverteilung sichtbar. Entsprechend der Theorie sollte diese einer Rechteckverteilung entsprechen. Der Messaufbau liefert aber eine Verteilung wie in Abbildung 98, sie hat Überhöhungen an den Rändern.

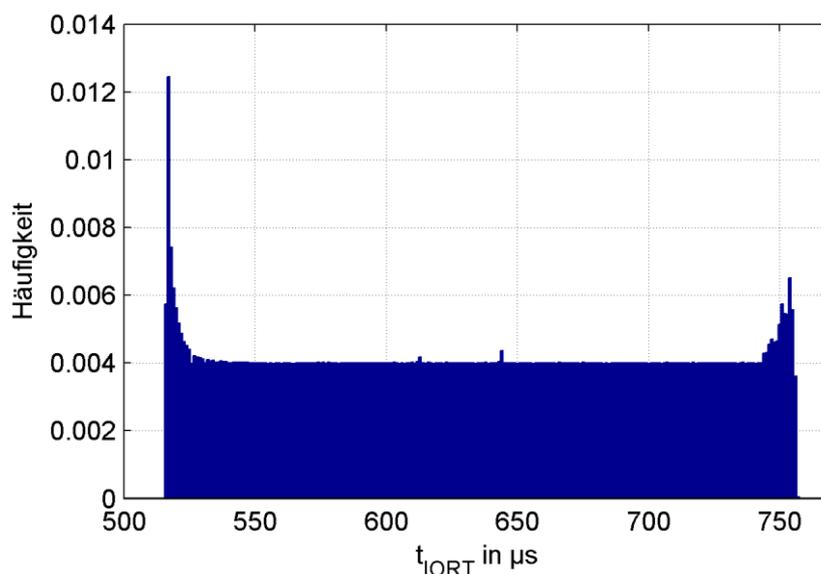


Abbildung 98: Häufigkeitsverteilung t_{IORT} für isochrones Steuerungssystem (diskretes Testsignal)

Im Anhang B wurden die Auswirkungen von Messrauschen und endlicher Anstiegsgeschwindigkeit des Eingangssignals simulativ analysiert. Es kann gezeigt werden, dass die Überhöhungen an den Rändern als Störeffekt des Messaufbaus entstehen und keine Eigenschaften des vermessenen Systems sind.

Im Rahmen dieser Arbeit wurden diese Messungen mit dem netANALYZER der Firma Hilscher [Hil15] durchgeführt. Dieser kann die Pakete von 2 Ethernet-Verbindungen und 4 Digitalsignale mit einer zeitlichen Auflösung von 10 ns zeitstempeln. Die anschließende Auswertung erfolgte in MATLAB.

6.3.1.2 Zustandskontinuierliches Testsignal

Um die Entstehung des Jitters zwischen Testsignal und Steuerungssystem zu vermeiden, muss ein zustandskontinuierliches Testsignal verwendet werden. Hierzu bieten sich Signale wie ein Sägezahn oder Dreieckssignal an. Der Messaufbau entspricht auch hier Abbildung 95, es ändert sich jedoch das Messsystem. Dieses muss das Ein- und Ausgangssignal analog abtasten und die Messkurve speichern. In Abbildung 99 ist beispielhaft der zeitliche Verlauf von Eingangs- (schwarz) und Ausgangssignal (rot) bei der Messung mit einem Dreieckssignal dargestellt. Zur Bestimmung der Klemme-Klemme-Reaktionszeit muss zu jedem neuen Wert des Ausgangssignals (nach jeder Stufe) der gleiche Wert auf der Messkurve des Eingangssignals gesucht werden. Die zeitliche Differenz zwischen beiden Werten entspricht der Klemme-Klemme-Reaktionszeit $t_{IORT,i}$ für diesen Signaldurchlauf. Auch hier wird eine Menge von Messwerten für t_{IORT} bestimmt, welche anschließend als Häufigkeitsverteilung dargestellt werden kann.

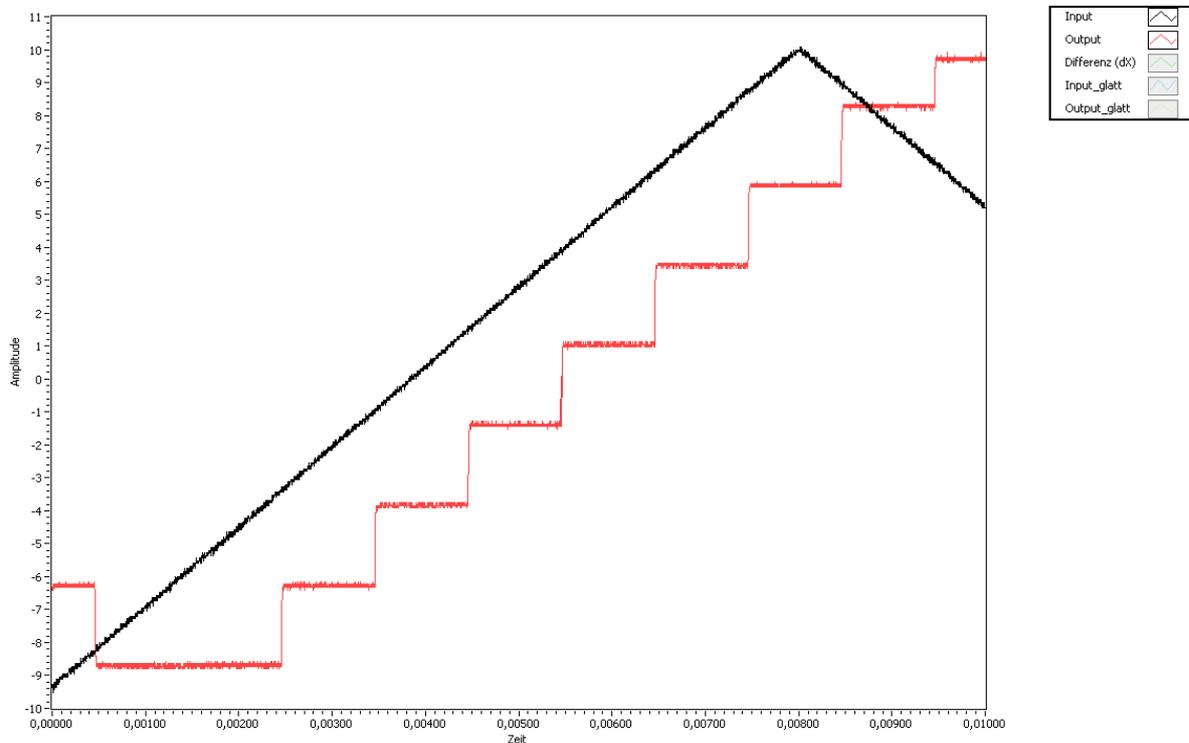


Abbildung 99: Messung von Klemme-Klemme-Reaktionszeit mittels Dreiecks-Testsignal

Der Vorteil dieses Messaufbaus ist die Vermeidung des Jitters zwischen Testsignal und Steuerungssystem.

Der Nachteil liegt jedoch darin, dass die zeitliche Auflösung der gemessenen Klemme-Klemme-Reaktionszeit von der AD-Wandlungsrate des Messsystems abhängt. Für eine zeitliche Auflösung von 100 ns ist ein Messsystem erforderlich, welches kontinuierlich mit einer Abtastrate von mindestens 10 MHz zwei Analogsignale einliest. Neben der Abtastrate muss auch die Auflösung der Messkanäle ausreichend hoch sein, da sich der Quantisierungsfehler der Amplitude auf die zeitliche Genauigkeit auswirkt.

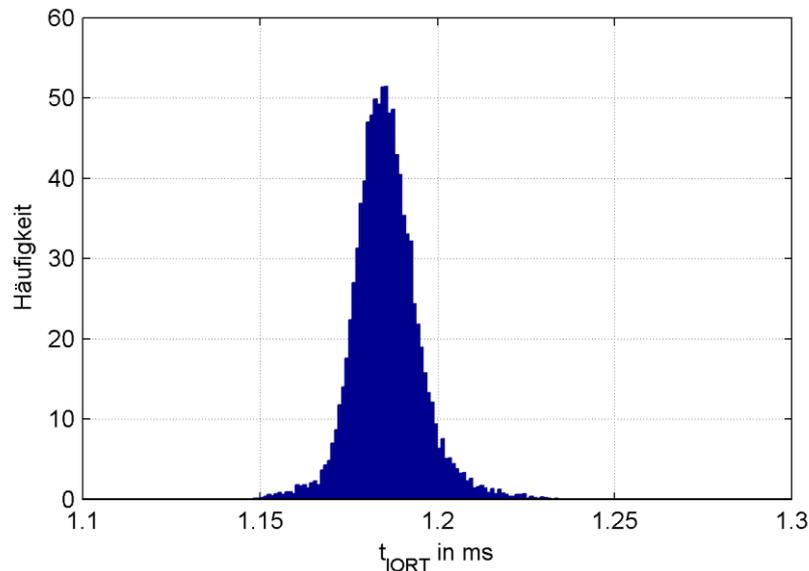


Abbildung 100: IORT für isochrones Steuerungssystem (zustandskontinuierliches Testsignal)

Auch bei diesem Messaufbau behindert das Signal- und Messrauschen die Auswertung, da es eine Zuordnung von Eingangs- zu Ausgangswerten erschwert. Im Rahmen dieser Arbeit wurden Messungen mit kontinuierlichem Testsignal durchgeführt, siehe Abbildung 100. Da kein Messsystem mit ausreichender Geschwindigkeit und Auflösung zur Verfügung stand, konnten keine zufriedenstellenden Messergebnisse erzielt werden. Daher wurden alle folgenden Messungen mit einem Rechtecksignal durchgeführt.

6.3.2 Messung von Teilstrecken

Für die Bestimmung von Parameter des Systems kann die Messung von Verarbeitungszeiten von Teilprozessen oder eine Teilstrecke (Kette von Teilprozessen) erforderlich sein. Dazu müssen geeignete Messpunkte im Signalpfad identifiziert werden. Typische Messpunkte, welche mit geringem Aufwand genutzt werden können, sind Ein- und Ausgangssignale, Kommunikationstelegramme und (soweit vorhanden und zugänglich) Trigger-Signale von Baugruppen.

Die Messung von physikalischen Signalen wurde im vorigen Abschnitt beschrieben. Für die Erfassung von Telegrammen im Kommunikationsmedium ist ein sogenannter Test Access Point (TAP) erforderlich, welcher die Aufzeichnung von Telegrammen erlaubt, ohne den eigentlichen Kommunikationsprozess zu stören. Die Telegramme müssen gefiltert und das passende Telegramm mit einem Zeitstempel versehen werden. So kann beispielsweise die Zeit vom Anlegen eines Signals an der Eingangsbaugruppe bis zum Übertragen auf dem Kommunikationssystem oder die Zeit vom Senden eines Datums an eine Ausgabebaugruppe bis zur physikalischen Ausgabe gemessen werden.

Die Bereitstellung von Messpunkten innerhalb von Baugruppen (z. B. bei modularen Feldgeräten zur Bestimmung der Verarbeitungszeit in einem Modul) oder der Steuerung (Messung der Laufzeit des Steuerungsprogramms) ist sehr schwierig und in der Regel nur mit Unterstützung des Herstellers möglich. Im Rahmen dieser Arbeit wurden speziell vom Hersteller präparierte Feldgeräte verwendet, welche den Beginn eines Verarbeitungszyklus als digitales Signal ausgeben. Dies erlaubte die Messung von internen Verarbeitungszeiten.

Entsprechend der Betrachtungen zur Granularität ist eine Kenntnis der Zeiten jedes Teilprozesses selten erforderlich. Daher reicht es beispielsweise bei einem modularen Feldgerät aus, die Verarbeitungszeit des Gerätes vom physikalischen Signal bis zum Kommunikationssystem zu bestimmen.

6.3.3 Messung der Synchronisationsgenauigkeit

Eine messtechnische Bestimmung der Güte einer Synchronisation wurde von Gramann und Mohl in [GM03] vorgestellt. Der entsprechende Messaufbau ist in Abbildung 101 dargestellt. Abbildung 102 stellt das Messergebnis für einen Aufbau mit PTP dar.

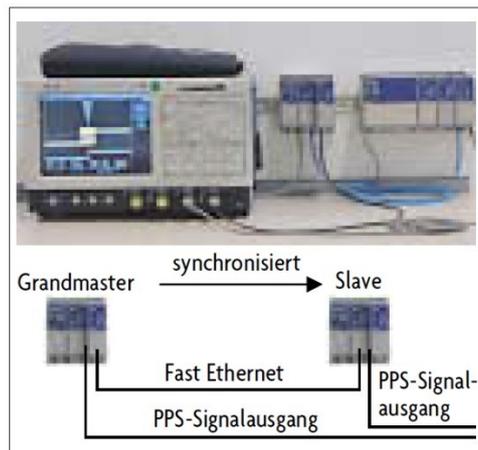


Abbildung 101: Messaufbau für den Test von PTP [GM03, S. 93]

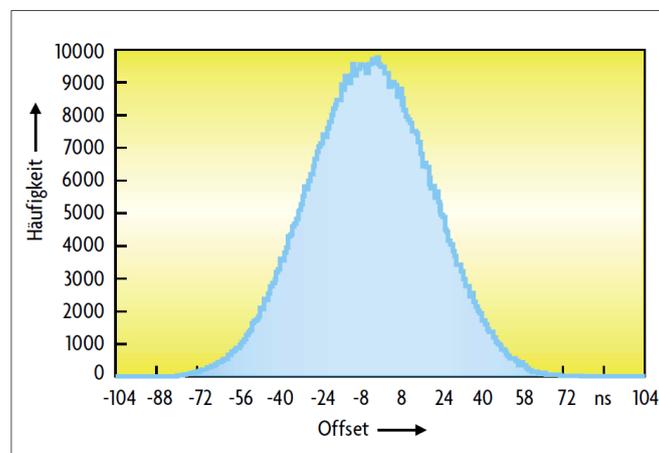


Abbildung 102: Gemessene Abweichung zwischen Master und Slave bei PTP [GM03, S. 94]

Bei der messtechnischen Bestimmung der Synchronisationsgenauigkeit muss beachtet werden, dass eine entsprechende Messung in der Regel einen Gut-Fall unter Laborbedingungen darstellt. Die Parameter für Präzision oder Genauigkeit sollen aber den Worst-Case-Fall abdecken. Störeinflüsse in realen Anlage (Kabellängen, Temperatur, Alterung, ...) müssen dazu berücksichtigt werden.

Die Messung zwischen verschiedenen Geräten dient zur Bestimmung der Präzision, d. h. der Abweichung zwischen den Geräten. Wird eine solche Messung bezogen auf eine Referenzzeit durchgeführt (z. B. GPS-Zeit), so kann auch die Genauigkeit ermittelt werden.

6.4 Ableiten von Parametern und Eigenschaften aus Messdaten

Messtechnisch können Verzögerungszeiten einer Kette von Teilprozessen oder des Gesamtsystems bestimmt werden. Aus den Häufigkeitsverteilungen dieser Verzögerungen können Rückschlüsse auf die Parameter der beteiligten Prozesse gezogen werden. So können bei Systemen, deren Parameter unvollständig bekannt sind, fehlende Parameter ergänzt werden. Im Folgenden wird die Betrachtung für die Klemme-Klemme-Reaktionszeit t_{IORT} durchgeführt, die Informationen gelten aber auch für die Verzögerung einer Kette von Teilprozessen.

6.4.1 Extremwerte der Verteilung

Aus den Extremwerten der Häufigkeitsverteilung von t_{IORT} lassen sich zwei Eigenschaften des Systems ableiten (Herleitung siehe Abschnitt 4.4.4). Das Minimum von t_{IORT} entspricht der Summe der minimalen Verarbeitungszeiten der Teilprozesse. Sinnbildlich ist dies der „Gut-Fall“, d. h. es treten keine stochastischen Wartezeiten auf.

$$\min(t_{\text{IORT}}) = \sum \min(t_{\text{process}}) \quad (146)$$

Das Maximum von t_{IORT} stellt im Gegenzug den „Schlecht-Fall“ dar (alle stochastischen Wartezeiten nehmen ihren größtmöglichen Wert an). Es entspricht der Summe aus den Maximalwerten der Verarbeitungszeiten und der stochastischen Wartezeiten:

$$\max(t_{\text{IORT}}) = \sum \max(t_{\text{process}}) + \sum \max(t_{\text{wait}}) \quad (147)$$

Wenn die Verarbeitungszeiten der Teilprozesse als konstant angenommen werden können, so entspricht die Jitter-Breite der Verteilung von t_{IORT} der Summe der stochastischen Wartezeiten durch asynchrone Zyklusübergänge:

$$\max(t_{\text{IORT}}) - \min(t_{\text{IORT}}) = \sum \max(t_{\text{wait}}) \quad (148)$$

Wenn für ein System eine Zykluszeit unbekannt ist, kann aus den bekannten Zykluszeiten und dem Strukturwissen die fehlende Zykluszeit abgeleitet werden.

6.4.2 Analyse der Kurvenform

Aus der Kurvenform der Verteilung lassen sich Rückschlüsse auf die Zahl der asynchronen Teilprozesse und deren Zykluszeiten treffen. Dies funktioniert am besten, wenn der Signalpfad nur aus konstanten Verzögerungen und Wartezeiten durch asynchrone Zyklusübergänge besteht. In diesem Fall entspricht die resultierende Verteilung einer Faltung von n Rechteckverteilung, die in Kapitel 4.4.4.4 analytisch berechnet wurde.

Die Faltung von n Rechteckverteilungen ergibt eine Verteilung mit bis zu 2^n Knickpunkten, die abschnittsweise aus einer Summe von Funktionen des Grades $(n - 1)$ besteht. Für eine geringe Anzahl Teilprozesse lassen sich aus der Kurvenform Knickpunkte und Parameter der Teilfunktionen ableiten. Aus diesen Daten lassen sich mit den analytisch berechneten Gleichungen und bekanntem Systemwissen fehlende Parameter bestimmen.

Die Schwierigkeit dieses Verfahrens liegt im Finden der Knickpunkte. Da Messergebnisse fehlerbehaftet sind (siehe z. B. Abschnitt 6.3.1.1 zum Einfluss des Messrauschens), wird die Bestimmung der Knickpunkte ab fünf asynchronen Teilprozessen nahezu unmöglich. Der Einfluss des Messrauschens kann durch eine Hüllkurvenapproximation reduziert werden. Dabei werden verschiedene bekannte Kurvenformen (verschiedene Anzahl von Zyklen) iterativ an die Messwerte angepasst und die Kurvenform mit der geringsten Abweichung ausgewählt. Im Abschnitt 7.2.1 wird der Ansatz an Beispielen gezeigt.

Neben der Faltung von Rechteckverteilungen kann dieses Verfahren auch für andere Verteilungen genutzt werden, die durch eine analytische Herleitung bestimmt werden können. Beispielhaft sei hier das Ergebnis der Faltung einer diskreten Gleichverteilung (synchrone Teilsysteme wurden zusammengefasst) mit einer Rechteckverteilung genannt. Dies kann beim Zusammenfassen von synchronen und asynchronen Teilprozessen entstehen und wird in Abschnitt 7.2.2 gezeigt.

6.4.3 Numerische Entfaltung

Ein allgemeiner theoretischer Ansatz zur Analyse der Messdaten ist die Entfaltung (auch Dekonvolution genannt) der Wahrscheinlichkeitsdichtefunktion. Diese stellt die Umkehrung der Faltungsoperation dar. Sie ist definiert als die Berechnung der Funktion f_1 aus bekannten Funktionen f_2 und h für die gilt:

$$h = f_1 * f_2 \quad (149)$$

Die Entfaltung ist nicht immer möglich, da bei der Faltung Informationen verloren gehen können.

Die Entfaltung findet in verschiedenen Gebieten Anwendung z. B. in der digitalen Bilderverarbeitung [BL11] und Signalverarbeitung in der Medizintechnik [Joh06]. Es existieren verschiedene numerische Verfahren, die versuchen, aus den bekannten Daten die wahrscheinlichste Entfaltung zu bestimmen. Beispielhaft seien hier genannt:

- Entfaltung mittels Fourier-Transformation [BL11, S. 113ff.]
- Entfaltung mittels Variationsmethoden [BL11, S. 239ff.]
- Wiener-Dekonvolution [GWE04, S. 170ff.]

Neben dem Informationsverlust bei der Faltung stellt das Messrauschen ein Problem bei der Entfaltung dar, da es durch Entfaltung verstärkt wird. Daher wird meist der Ansatz der Glättung mit anschließender Entfaltung verfolgt (siehe z. B. [BL11] oder [Tec03]). Dies ist für den hiesigen Anwendungsfall problematisch, da die Wahrscheinlichkeitsdichtefunktion oftmals keine glatte Funktion ist und Sprungstellen (infolge von Rechteckverteilungen) enthält, welche durch die Glättung verfälscht werden.

Die Mathematik-Software MATLAB implementiert als Umkehrung der numerischen Faltung eine Funktion *deconv*, welche die Entfaltung per Polynomdivision durchführt. Der theoretische Ansatz der Entfaltung ist nur in Sonderfällen einsetzbar, einen allgemeinen Weg zu Parameterrekonstruktion aus Messwerten stellt er nicht dar. Im Abschnitt 7.2.3 wird die Umsetzung am Beispiel gezeigt.

6.4.4 Grenzen der Ableitung

Für die Ableitung von Parametern der Teilprozesse aus Messwerten von einem Signalpfad sind Prinzip bedingt Grenzen gesetzt (Informationsverlust bei der Faltung).

Die Reihenfolge der Teilverzögerungen entlang des Signalpfades kann nicht rekonstruiert werden, da das Zusammenfassen der Teilverteilungen kommutativ ist. Jedoch kann aufgrund der strukturellen Abhängigkeiten bei der Entstehung von Wartezeiten an asynchronen Zyklusübergängen teilweise auf die mögliche Reihenfolge von Teilprozessen geschlossen werden, siehe Beispiel in Abschnitt 7.2.2.2.

Weiterhin ist es nicht möglich, aus der Summe der konstanten Verzögerungen auf die konstanten Verzögerungen von Teilprozessen zu schlussfolgern. Die konstanten Verzögerungen werden beim Verketteten von Teilprozessen addiert, die Reihenfolge hat keinen Einfluss.

Trotz der Einschränkungen können mit Hilfe von Messdaten und Wissen über Struktur und einen Teil der Parameter eines Systems, fehlende Parameter bestimmt werden. Das genutzte Verfahren ist abhängig davon, welche Parameter fehlen, welche Messdaten zur Verfügung stehen oder welche Messpunkte für eine Messung zugänglich sind. Wesentliche Verfahren zur Ermittlung von Parametern wurden gezeigt.

6.5 Nutzung vorhandener Beschreibungssysteme

Für den Austausch von Daten zwischen verschiedenen Werkzeugen in der Anlagenplanung wurden verschiedene Beschreibungssysteme entwickelt, welche teilweise auch Informationen über das Steuerungs- und Kommunikationssystem enthalten.

6.5.1 Gerätebeschreibungssprachen

Es existieren verschiedene Austauschformate für die Beschreibung von Geräten in Kommunikationssystemen. Diese liefern in der Regel gerätespezifische Eigenschaften (mögliche Werte), aber keine Informationen über die reale Konfiguration eines Gerätes.

Die Nutzerorganisationen der Feldbusse haben Beschreibungssprachen für die jeweiligen Systeme entwickelt, beispielsweise GSDML³⁹ für PROFINET, ESI⁴⁰ für EtherCAT oder EDS⁴¹ für EtherNet/IP. Neben den systemspezifischen Beschreibungssprachen existieren auch systemneutrale Formate, hier ist beispielsweise FDCML⁴² zu nennen. Diese Beschreibungsdateien für ein Gerät beschreiben seine Möglichkeiten, d. h. es werden beispielsweise sämtliche mögliche Buszykluszeiten des Gerätes angegeben. Für die Parametrierung des hier vorgestellten Modells ist jedoch die konkrete Parametrierung in einer Anlage erforderlich.

Für die Ermittlung von Verarbeitungszeiten von Feldgeräten kann die Beschreibungsdatei eines Gerätes hilfreich sein. So wird zum Zeitpunkt dieser Arbeit für PROFINET eine Isochronous Mode-Guideline erstellt, welche die Ablage von Isochron-Parameter eines Feldgerätes in der Gerätebeschreibungsdatei (GSDML-Datei) beschreibt. Dadurch ließe sich die Zeit für Datenein- und -ausgabe isochroner Baugruppen anhand der Beschreibungsdatei ermitteln.

³⁹ GSDML = Generic Station Description Markup Language

⁴⁰ ESI = EtherCAT Slave Information

⁴¹ EDS = Electronic Data Sheets

⁴² FDCML = Field Device Configuration Markup Language

6.5.2 AutomationML

„AutomationML ist ein Datenaustauschformat für die konsistente Weitergabe von Planungsdaten im Engineering von Produktionssystemen.“ [LRD+13, S. 4] Die Beschreibung erfolgt dabei mittels hierarchischer Objektstrukturen, welche im Format CAEX nach IEC 62424 [IEC08] abgelegt werden. AutomationML wird im Rahmen der Standardserie IEC 62714 genormt [IEC14].

Das AutomationML Consortium hat in [Aut14] eine Methode zur Beschreibung eines Kommunikationssystems in industriellen Steuerungen unabhängig von Hersteller und dem eingesetzten Kommunikationssystem beschrieben. Dazu werden zwei verschiedene Sichtweisen auf das Steuerungssystem eingeführt, eine physikalische und eine logische (siehe Abbildung 103). Es wird ein Mapping auf das ISO/OSI-Referenzmodell durchgeführt, die physikalische Schicht entspricht den Schichten 1 und 2, die logische Sicht bildet die Schichten 3 bis 7 ab.

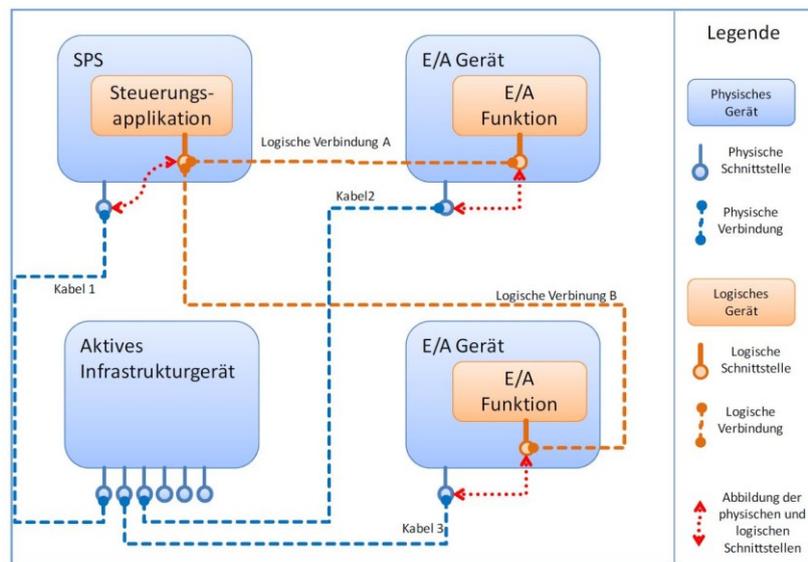


Abbildung 103: Physikalische und logische Sicht in AutomationML [LRD+13, S. 4]

Diese Trennung in verschiedene Sichten ähnelt dem Ansatz im hier gezeigten Modell, welches im Strukturmodell eine Trennung von Funktionen (logische Sicht) und ausführender Hardware (physikalische Sicht) vorsieht. Daher kann die AutomationML-Beschreibung (wie im Beispiel in Abbildung 104 dargestellt) die Erstellung des Modells eines Steuerungssystems unterstützen. Es werden die logischen Pfade der Daten dargestellt. Nicht dargestellt werden die Teilprozesse innerhalb der Komponenten. Dies ist in der Fokussierung auf die Beschreibung von Kommunikationssystemen begründet.

Die Parameter als Attribute der Objekte sind in AutomationML nicht festgelegt, sondern werden abhängig vom Anwendungsfall definiert. In [Aut14] wird beispielsweise die Zykluszeit des Kommunikationssystems erwähnt, in [LRD+13] die Abarbeitungszeit von Steuerungsapplikationsteilen als mögliches Attribut genannt. Diese Attribute können für die Parametrierung des Modells genutzt werden. Parameter der Teilprozesse, wie Verarbeitungszeiten von EA-Baugruppen, sind in den untersuchten Veröffentlichungen zu AutomationML nicht enthalten, ließen sich aber aufgrund des flexiblen Ansatzes hinzufügen.

Durch eindeutige Identifier stellt AutomationML eine Beziehung zwischen Variablen im Steuerungsprogramm, übertragenen Daten in der PDU und physischen Signalen (Ein- und Ausgabekanäle) her [RLH+14]. Es liefert damit einen Teil der Beschreibung des Signalpfades innerhalb des hier vorgestellten Modells.

Die detaillierte Beschreibung des Kommunikationssystems mit Topologie und PDU-Beschreibung kann nicht direkt verwendet werden. Sie kann jedoch als Eingangsdaten für ein Werkzeug zur Berechnung von Laufzeiten im Kommunikationssystem genutzt werden, welches wiederum die Verarbeitungszeiten des Kommunikationssystems für bestimmte Signalpfade liefert.

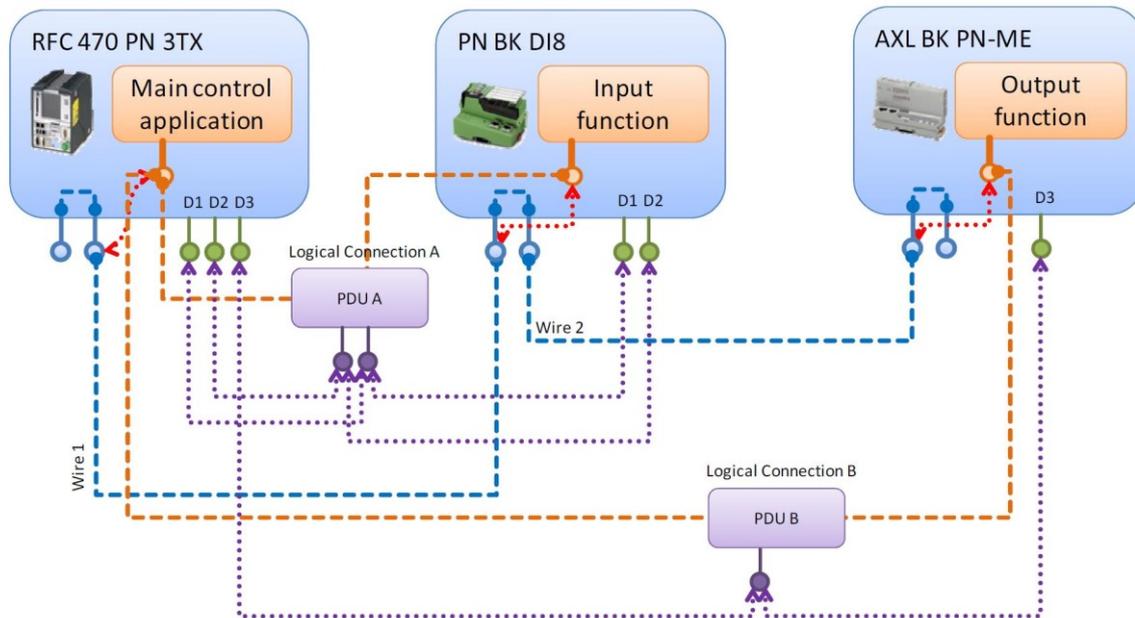


Abbildung 104: PROFINET-Demonstrator in AutomationML [Aut14, S. 68]

Zusammenfassend kann man feststellen, dass die Beschreibung eines Kommunikationssystems in AutomationML für die Erstellung und Parametrierung eines Modells zur Zeitberechnung genutzt werden kann. Wie viele strukturelle und parametrische Details im Modell enthalten sind, hängt von der konkreten Umsetzung ab. Bei entsprechendem Modellumfang wäre eine Ableitung des Modells aus der AutomationML-Beschreibung möglich.

6.6 Fazit

Wie auch von anderen Autoren benannt (z. B. Diekhake und Schnieder in [DS14] und [DS15]), stellt die Parametrierung des Modells eine große Schwierigkeit dar. Selten sind alle benötigten Parameter direkt zugänglich. In diesem Kapitel wurden verschiedene Ansätze zur Bestimmung der Werte gezeigt. Durch geschickte Kombination von Informationen aus Datenblättern, Planungswerkzeugen sowie Messungen und Analysen von Teilstrecken kann das hier vorgestellt abstrakte Modell vollständig parametriert werden.

Durch die Nutzung von Beschreibungssprachen wie AutomationML kann die Erstellung und Parametrierung eines solchen Modells deutlich vereinfacht werden. Voraussetzung ist jedoch auch hier, dass die notwendigen Informationen in der konkreten Systembeschreibung enthalten sind.

7 Modellierungsbeispiele

7.1 Modellierung, Berechnung und Messung von Beispielsystemen

Die in Kapitel 4 beschriebene Modellierung von Steuerungssystemen und die Berechnung des Zeitverhaltens soll, ähnlich wie in [HD12] veröffentlicht, anhand von Versuchsaufbauten mit dem Kommunikationssystem PROFINET verifiziert werden. Dabei wird durch unterschiedliche Konfigurationen die Zahl der asynchronen Teilprozesse variiert.

7.1.1 Szenarien

Es wird ein Steuerungssystem bestehend aus einer oder zwei Steuerungen und zwei modularen Feldgeräten, welche über PROFINET RT bzw. PROFINET IRT kommunizieren, gewählt. Die modularen Feldgeräte sind mit analogen Ein- und Ausgabebaugruppen bestückt⁴³.

Das Anwendungsprogramm kopiert die analogen Eingänge auf die Ausgänge, wie im Versuchsaufbau zur Bestimmung der Klemme-Klemme-Reaktionszeit vorgesehen. Die Messungen werden mit einem zustandsdiskreten Testsignal (Rechtecksignal), welches von einem Signalgenerator erzeugt wird, durchgeführt. Die Messwerterfassung erfolgt mittels netANALYZER.

Die genauen Bezeichnungen der eingesetzten Hardware finden sich in Anhang C .

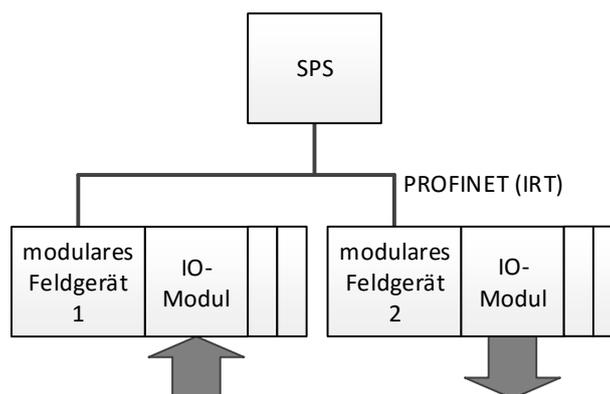


Abbildung 105: Architektur des zentralen Steuerungssystems mit PROFINET

7.1.1.1 Szenario A: PROFINET IRT mit taktsynchronen EAs

Im ersten Versuchsaufbau kommuniziert eine SPS über PROFINET IRT (d. h. synchronisiert) mit beiden Feldgeräten. Durch Nutzung der sogenannten Taktsynchronität werden außerdem die Ein- und Ausgangsmodule sowie das Steuerungsprogramm auf den Kommunikationszyklus synchronisiert. Es liegt damit ein isochrones Steuerungssystem nach Kapitel 5 vor. Die Zykluszeit des Systems wurde mit 1 ms parametrisiert.

Durch die Nutzung des zustandsdiskreten Testsignals liegt **1 asynchroner Zyklusübergang** am Eingang des Steuerungssystems vor.

⁴³ Es wurden analoge Ein- und Ausgabemodule genutzt, obwohl später mit einem digitalen Testsignal gemessen wurde, da keine digitalen Module mit der Fähigkeit Taktsynchronität zur Verfügung standen.

7.1.1.2 Szenario B: PROFINET IRT ohne taktische EAs

Die Konfiguration des Systems wird im Vergleich zu Szenario A geändert, die Kommunikation läuft weiterhin über PROFINET IRT, die Synchronisation der Ein- und Ausgangsmodule wird jedoch deaktiviert. Da es sich um modulare Feldgeräte handelt, entstehen asynchrone Übergänge zwischen dem Ein- bzw. Ausgangsmodul und dem jeweiligen Feldgerät.

Zusammen mit dem asynchronen Übergang am Eingang ergeben sich somit **3 asynchrone Zyklusübergänge** entlang des Signalpfades.

7.1.1.3 Szenario C: PROFINET RT

Zu Vergleichszwecken werden bei diesem Szenario alle Synchronisationsmechanismen (IRT, Taktsynchronität) deaktiviert. Es ergeben sich **5 asynchrone Zyklusübergänge**.

7.1.1.4 Szenario D: Verteiltes PROFINET-System

Als Beispiel für ein einfaches verteiltes Steuerungssystem wurden in diesem Szenario zwei Steuerungen eingesetzt. Jede Steuerung kommuniziert mit einem Feldgerät über PROFINET IRT mit Taktsynchronität. Die Kommunikation zwischen den Steuerungen findet über PROFINET RT statt, sie sind nicht synchronisiert. Die Daten werden vom SPS 1 aus dem zugeordneten Feldgerät eingelesen, an SPS 2 geschickt und von dieser über das zweite Feldgerät ausgegeben.

Dies entspricht dem realen Szenario, dass innerhalb eines Anlagenteiles (z. B. einer Fertigungszelle) isochrone, geplante Kommunikation stattfindet, zwischen ihnen jedoch nur RT-Kommunikation.

Wie in den vorherigen Szenarien findet auch hier ein asynchroner Zyklusübergang am Eingang des Feldgerätes statt. Weiterhin liegt zwischen den Steuerungen mangels Synchronisation der Zeitdomänen ein asynchroner Übergang vor. Das Szenario besteht damit aus **2 asynchronen Zyklusübergängen**.

Die Zykluszeiten der beiden synchronen Verbünde aus Steuerung und Feldgerät sind auf 1 ms parametrisiert, die Kommunikation zwischen den Steuerungen arbeitet mit einem Zyklus von 2 ms.

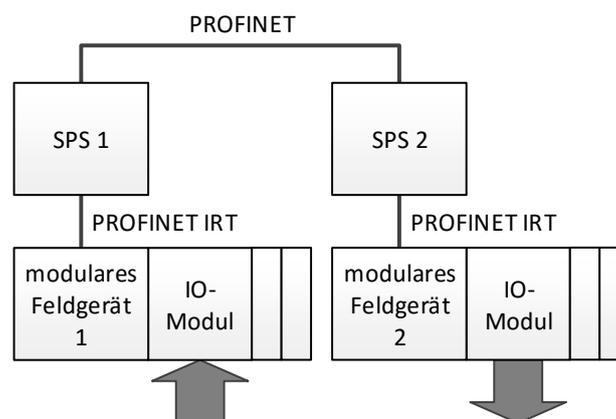


Abbildung 106: Architektur des verteilten Steuerungssystems in Szenario D

7.1.2 Modellierung und Ableitung des Zeitverhaltens

Im Folgenden wird für jedes Szenario ein Modell erstellt (nach Kapitel 4.2), das Zeitverhalten entlang des Signalpfades abgeleitet (nach Kapitel 4.3) und mittels Faltung der Häufigkeitsverteilungen berechnet (nach Kapitel 4.4).

7.1.2.1 Szenario A: PROFINET IRT mit taktsynchronen EAs

Das Modell für dieses Szenario besteht aus einem statischen Anteil (Abbildung 107) und einem dynamischen Anteil (Abbildung 108).

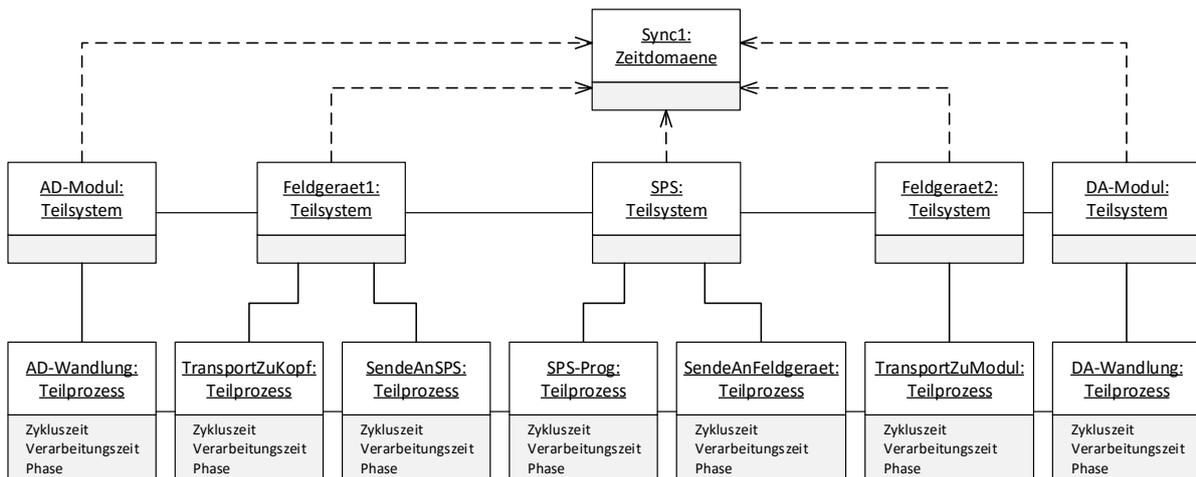


Abbildung 107: Statischer Modellanteil für Szenario A

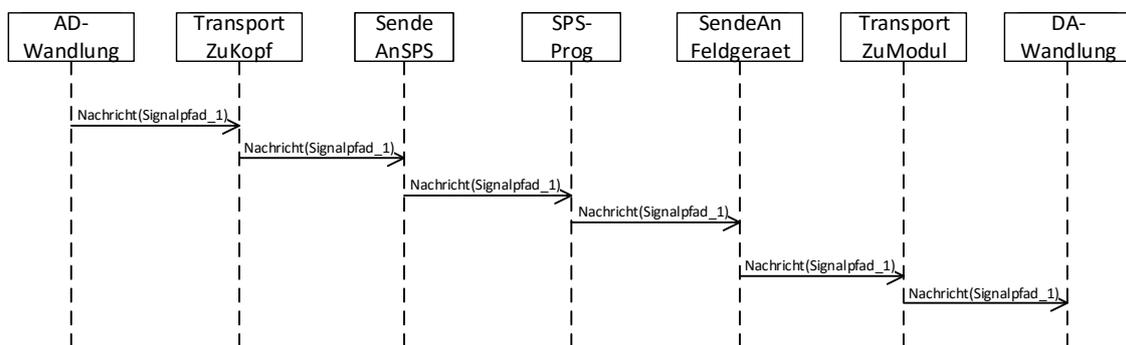


Abbildung 108: Dynamischer Modellanteil für Szenario A

Entsprechend der obigen Beschreibung sind alle Teilprozesse auf eine Zykluszeit von 1 ms parametrisiert.

Für die Parametrierung der Zeitdomäne werden die spezifizierten Parameter für PROFINET IRT verwendet, d. h. Genauigkeit und Präzision werden mit 1 μ s angenommen. Eine genauere Untersuchung ist nicht erforderlich, da dieser Wert deutlich kleiner ist als die Zykluszeiten und damit keinen (wahrnehmbaren) Einfluss auf das Ergebnis hat.

Eine Bestimmung der Verarbeitungszeit jedes Teilprozesses ist mit den vorliegenden technischen Mitteln nicht möglich. Daher werden entsprechend Kapitel 6.2.4 Parameter aus dem Engineering-System zur Bestimmung der Verarbeitungszeiten von Teilstrecken genutzt. Entsprechend des 3-T-Modells kann die Verarbeitungszeit von *AD-Wandlung* und *TransportZuKopf* als Zeitdauer T_1 , die Zeitdauer für *SendeAnFeldgeraet*, *TransportZuModul* und *DA-Wandlung* als Zeitdauer T_0 aus dem

Engineering-System bestimmt werden. Die Zeitdauer für *SendeAnSPS* und *SPS-Prog* entspricht nach dem 3-T-Modell einer Zykluszeit.

$$T_{\text{Cycle}} = 1 \text{ ms} \quad (150)$$

$$T_I = 0,14 \text{ ms} \quad (151)$$

$$T_O = 0,15 \text{ ms} \quad (152)$$

Für die Ableitung des Zeitverhaltens werden zunächst die synchronen Teilprozesse zusammengefasst. Entsprechend der Verfügbarkeit von Parametern wird das Modell auf drei synchrone Teilprozesse vereinfacht, siehe Abbildung 109. Die Phasen der Teilprozesse wurden entsprechend dem 3-T-Modell bestimmt.

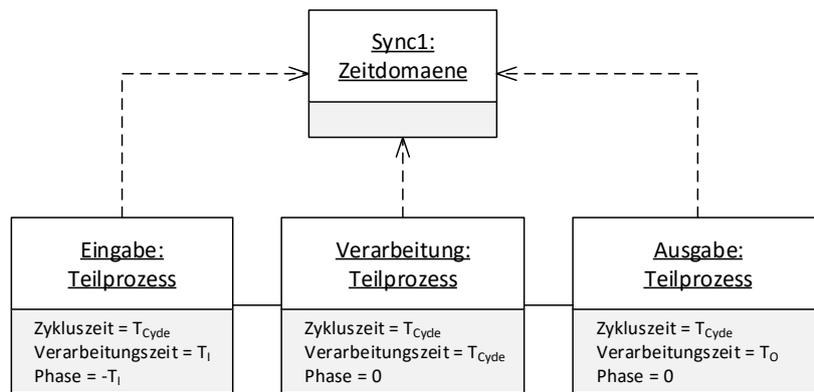


Abbildung 109: Vereinfachter statischer Modellanteil für Szenario A

Im nächsten Schritt werden die drei verbliebenen synchronen Teilprozesse zusammengefasst, dies geschieht durch Addition der Verarbeitungszeiten entsprechend Gleichung (40). Die resultierende Verarbeitungszeit der zusammengefassten Teilprozesse ergibt sich damit zu:

$$t_{\text{process, sync}} = T_I + T_{\text{Cycle}} + T_O = 1,29 \text{ ms} \quad (153)$$

Am Eingang des AD-Moduls findet ein asynchroner Zyklusübergang statt. Nach Abschnitt 4.3.6 ergibt sich daraus eine Wartezeit mit einer Rechteckverteilung in folgendem Intervall:

$$t_{\text{wait,1}} \in [0; T_{\text{Cycle}}) \quad (154)$$

Damit ergibt sich für diesen Signalpfad eine mathematische Beschreibung der Klemme-Klemme-Reaktionszeit, die aus der Summe der Rechteckverteilung am Eingang und der konstanten Verarbeitungszeit besteht. Die Klemme-Klemme-Reaktionszeit ist eine Rechteckverteilung im Intervall:

$$t_{\text{IORT}} \in [t_{\text{process, sync}}; t_{\text{process, sync}} + T_{\text{Cycle}}) \quad (155)$$

$$t_{\text{IORT}} \in [1,29 \text{ ms}; 2,29 \text{ ms}) \quad (156)$$

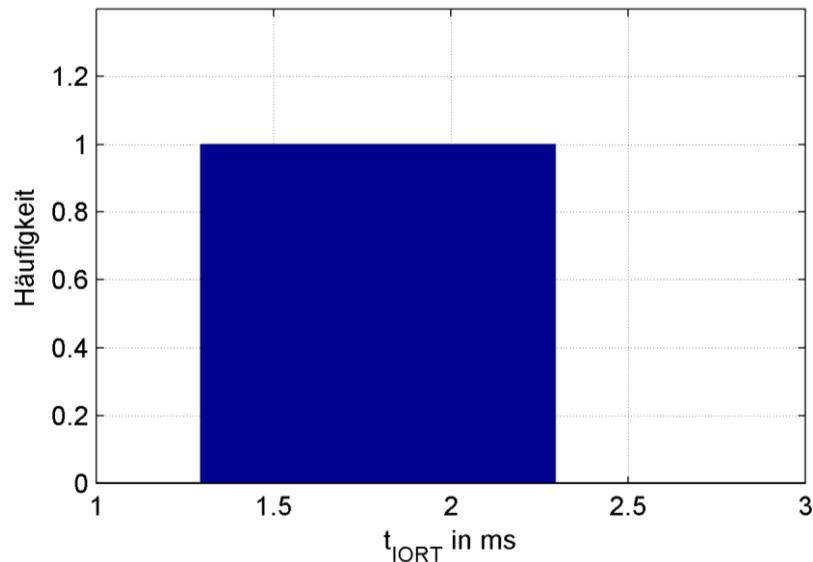


Abbildung 110: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit in Szenario A

7.1.2.2 Szenario B: PROFINET IRT ohne takttsynchrone EAs

Das statische Modell für Szenario B ist in Abbildung 111 dargestellt. Die Teilsysteme *Feldgeraet1*, *SPS* und *Feldgeraet2* arbeiten synchronisiert, während *AD-* und *DA-Modul* unsynchronisiert arbeiten. Auf eine Darstellung des dynamischen Modells soll verzichtet werden, da es ein Signalpfad von links nach rechts über alle Teilprozesse ist.

Die Zykluszeiten der synchronen Teilprozesse *TransportZuKopf*, *SendeAnSPS*, *SPS-Prog*, *SendeAnFeldgeraet* und *TransportZuModul* sind auf $T_{Cycle,1} = 1$ ms parametrisiert. Die Teilprozesse *AD-Wandlung* und *DA-Wandlung* arbeiten entsprechend der Datenblätter der Module mit einer Zykluszeit von $T_{Cycle,2} = 0,25$ ms.

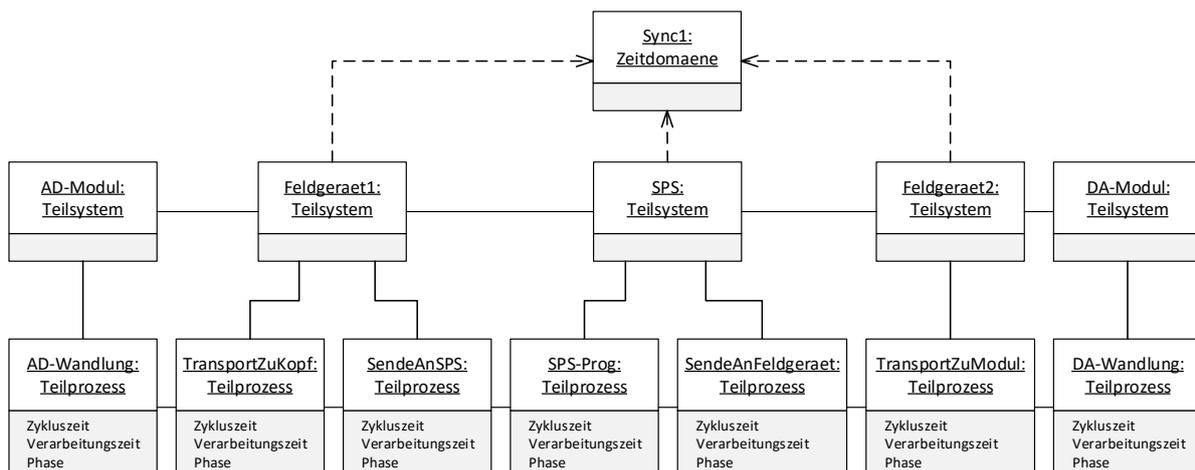


Abbildung 111: Statischer Modellanteil für Szenario B

Für die Ableitung des Zeitverhaltens werden zunächst die synchronen, aufeinanderfolgenden Teilprozesse zusammengefasst. Die Zykluszeit des Teilprozesses *Zusammengefasst* entspricht denen der Teilprozesse ($T_{Cycle,1}$), das vereinfachte Modell ist in Abbildung 112 dargestellt.

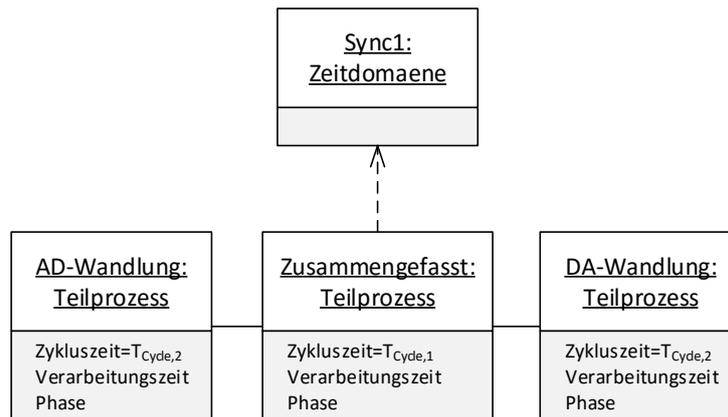


Abbildung 112: Zusammengefasster statischer Modellanteil für Szenario B

Die Bestimmung der Verarbeitungszeiten der Teilprozesse wurde messtechnisch durchgeführt. Bei der Messung der Klemme-Klemme-Reaktionszeit entspricht die Summe der konstanten Verarbeitungszeiten der minimal auftretenden Klemme-Klemme-Reaktionszeit.

$$\sum t_{\text{process}} = 1,19 \text{ ms} \quad (157)$$

Die stochastischen Wartezeiten durch asynchronen Zyklusübergang treten am Eingang und zwischen den Teilprozessen in Abbildung 112 auf. Nach Abschnitt 4.3.4 ergeben sie sich als Rechteckverteilungen mit den Intervallen:

$$t_{\text{wait},1} \in [0; T_{\text{Cycle},2}) \quad (158)$$

$$t_{\text{wait},2} \in [0; T_{\text{Cycle},1}) \quad (159)$$

$$t_{\text{wait},3} \in [0; T_{\text{Cycle},2}) \quad (160)$$

Entsprechend der Berechnungsvorschrift für das Zusammenfassen von drei Rechteckverteilungen in Abschnitt 4.4.4.4.2 ergibt sich eine Häufigkeitsverteilung im Intervall

$$t_{\text{IORT}} \in \left[\sum t_{\text{process}} ; \sum t_{\text{process}} + T_{\text{Cycle},1} + 2 \cdot T_{\text{Cycle},2} \right) \quad (161)$$

$$t_{\text{IORT}} \in [1,19 \text{ ms}; 2,69 \text{ ms}) \quad (162)$$

Der qualitative Verlauf ist in Abbildung 113 dargestellt.

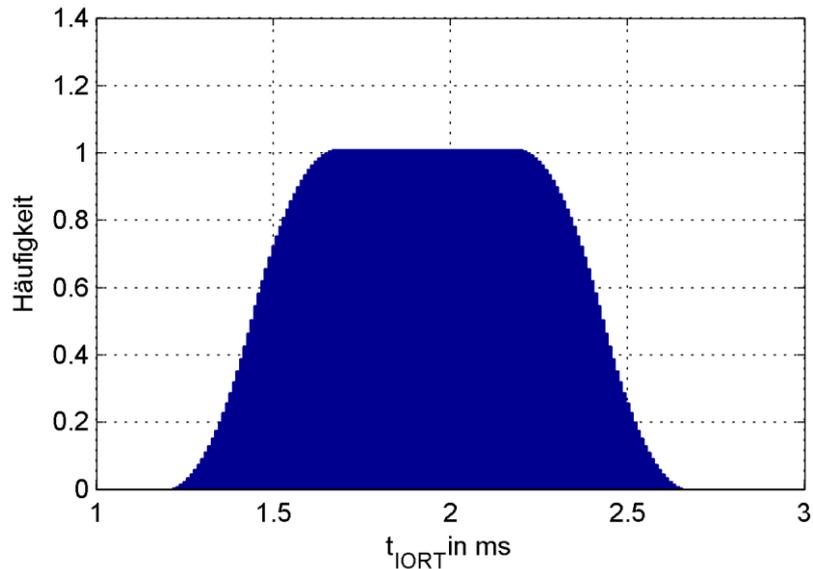


Abbildung 113: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit von Szenario B

7.1.2.3 Szenario C: PROFINET RT

Das statische Modell für Szenario C ist in Abbildung 114 dargestellt, auf eine Darstellung des Signalpfades wurde auch hier verzichtet.

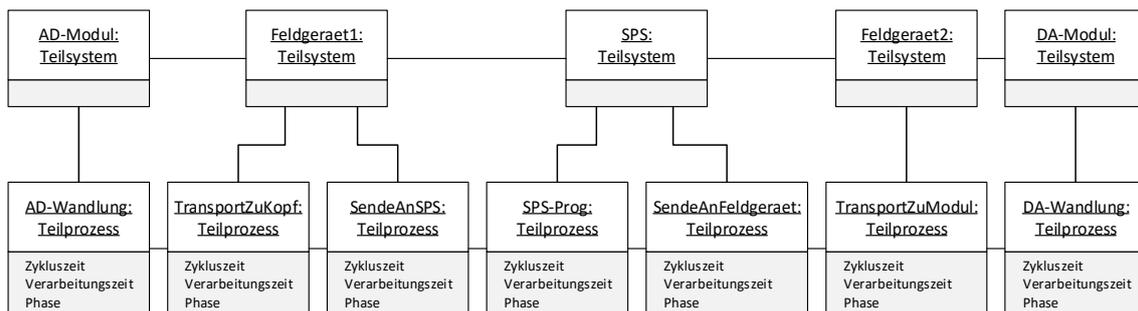


Abbildung 114: Statischer Modellanteil für Szenario C

Die Zykluszeiten der Teilprozesse liegen entsprechend der Parametrierung im Engineering-System und im Falle der EA-Module anhand der Datenblätter bei $T_{\text{Cycle},1} = 0,25$ ms für *AD-Wandlung* und *DA-Wandlung*, $T_{\text{Cycle},2} = 1$ ms für die Teilprozesse *TransportZuKopf*, *SendeAnSPS*, *SendeAnFeldgeraet* und *TransportZuModul* sowie bei $T_{\text{Cycle},3} = 0,5$ ms für das SPS-Programm.

Obwohl es keine dedizierten Synchronisationsmechanismen in diesem Szenario gibt, arbeiten Teilprozesse, welche dem gleichen Teilsystem zugeordnet sind, synchron, da sie vom gleichen Taktgeber abhängen.

Daher ergibt sich durch Zusammenfassen der synchronen Teilprozesse das vereinfachte Modell in Abbildung 115.

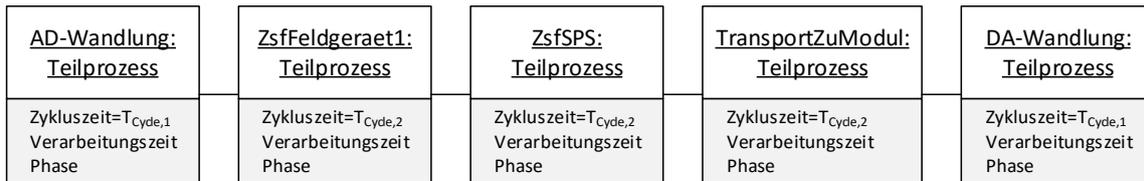


Abbildung 115: Zusammengefasster statischer Modellanteil für Szenario C

Die Zykluszeit von *ZsfFeldgeraet1* entspricht der identischen Zykluszeit der beiden Teilprozesse: $T_{\text{Cycle},2}$. Beim Zusammenfassen von *SPS-Prog* und *SendeAnFeldgeraet* wird die größere Zykluszeit von *SendeAnFeldgeraet* übernommen⁴⁴: $T_{\text{Cycle},2}$.

Die Bestimmung der Verarbeitungszeiten, welche als konstant angenommen werden, wurde wie bei Szenario B durch Messung der geringsten Klemme-Klemme-Klemme-Reaktionszeit durchgeführt.

$$\sum t_{\text{process}} = 2,4 \text{ ms} \quad (163)$$

Die Wartezeiten am Eingang und zwischen den Teilprozessen ergeben sich als Rechteckverteilungen in den Intervallen:

$$t_{\text{wait},1} \in [0; T_{\text{Cycle},1}) \quad (164)$$

$$t_{\text{wait},2} \in [0; T_{\text{Cycle},2}) \quad (165)$$

$$t_{\text{wait},3} \in [0; T_{\text{Cycle},2}) \quad (166)$$

$$t_{\text{wait},4} \in [0; T_{\text{Cycle},2}) \quad (167)$$

$$t_{\text{wait},5} \in [0; T_{\text{Cycle},1}) \quad (168)$$

Durch Aufsummieren dieser Verteilungen und der konstanten Verarbeitungszeiten ergibt sich eine Verteilung (siehe Abbildung 116) die in folgendem Intervall liegt:

$$t_{\text{IORT}} \in \left[\sum t_{\text{process}} ; \sum t_{\text{process}} + 2 \cdot T_{\text{Cycle},1} + 3 \cdot T_{\text{Cycle},2} \right) \quad (169)$$

$$t_{\text{IORT}} \in [2,4 \text{ ms}; 5,9 \text{ ms}) \quad (170)$$

⁴⁴ Hier zeigt sich, dass die schnellere Abarbeitung des SPS-Programms bei langsamerer Kommunikation ohne Vorteil bleibt.

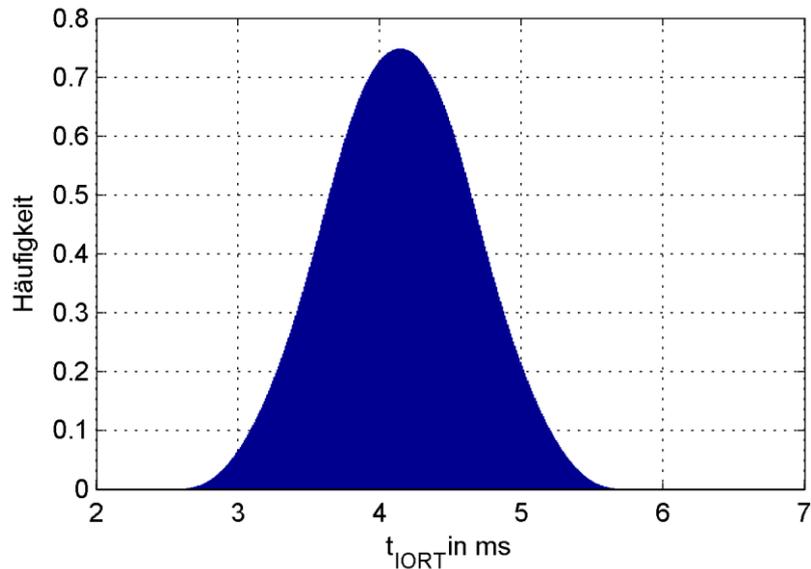


Abbildung 116: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit von Szenario C

7.1.2.4 Szenario D: Verteiltes PROFINET-System

Das statische Modell für Szenario D ist in Abbildung 117 dargestellt. Die Teilsysteme gehören entsprechend der Beschreibung zwei verschiedenen Zeitdomänen an.

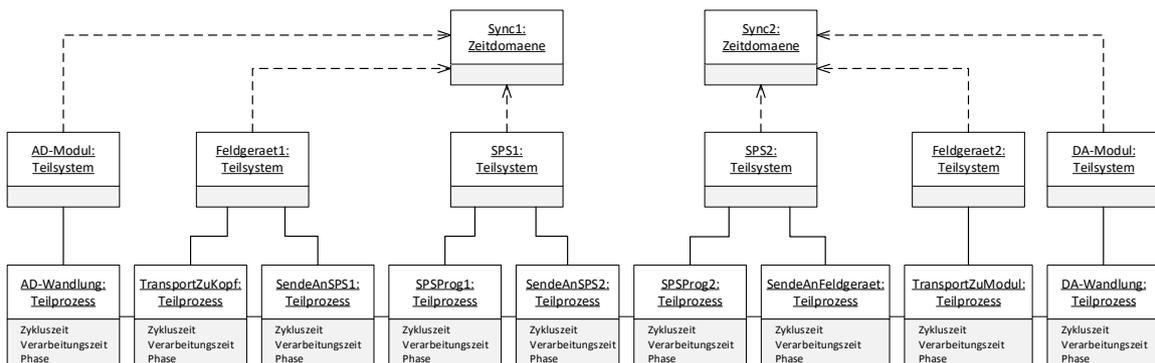


Abbildung 117: Statischer Modellanteil für Szenario D

Die Zykluszeit des Teilprozesses *SendeAnSPS2* ist mit $T_{\text{Cycle},1} = 2$ ms parametrisiert, alle anderen Teilprozesse arbeiten mit $T_{\text{Cycle},2} = 1$ ms. Zur Bestimmung der Parameter Verarbeitungszeit wird wie bei Szenario A das 3-T-Modell und das Engineering-System genutzt, um die Verarbeitungszeiten von Gruppen von Teilprozessen zu quantifizieren.

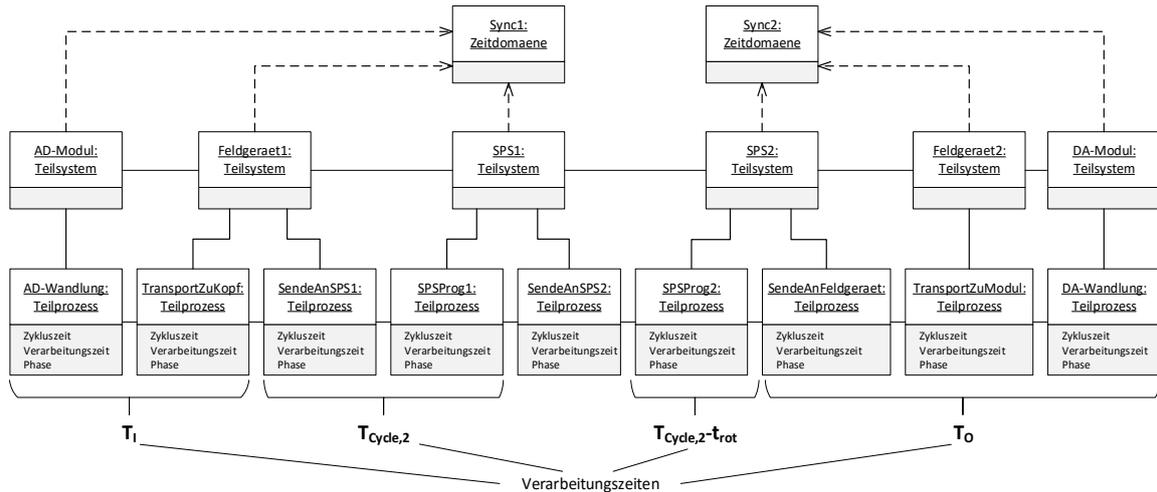


Abbildung 118: Parametrierung des Modells von Szenario D mittels 3-T-Modell

Die Verarbeitungszeit von *AD-Wandlung* und *TransportZuKopf* ergibt sich als Zeitdauer T_1 (von SPS1), die Zeitdauer für *SendeAnFeldgeraet*, *TransportZuModul* und *DA-Wandlung* als Zeitdauer T_0 (von SPS2). Die Summe der Verarbeitungszeit von *SendeAnSPS1* und *SPSProg1* entsprechen der Zykluszeit von SPS1. Die Zeitdauer von *SPSProg2* entspricht der Zykluszeit von SPS2 abzüglich der Zeitdauer für die rote Phase t_{rot} (dies ergibt sich aus dem 3-T-Modell für PROFINET IRT - siehe Abschnitt 5.3.2). Das Engineering-System liefert folgende Parameter:

$$T_{Cycle,1} = 2 \text{ ms} \quad (171)$$

$$T_{Cycle,2} = 1 \text{ ms} \quad (172)$$

$$T_1 = 0,14 \text{ ms} \quad (173)$$

$$T_0 = 0,15 \text{ ms} \quad (174)$$

$$t_{rot} = 0,016 \text{ ms} \quad (175)$$

Die Bestimmung der Verarbeitungszeit von *SendeAnSPS2* ist ohne eine Netzwerksimulation nicht möglich. Da das Kommunikationssystem unbelastet ist, wird hier näherungsweise davon ausgegangen, dass die Zeitdauer für die Übertragungszeit zwischen SPS1 und SPS2 etwa der Übertragungszeit zwischen SPS und Feldgeräten (t_{rot}) entspricht, welche im Engineering-System durch die Dauer der roten Phase angegeben ist.

Damit ergibt sich die Summe der konstanten Verarbeitungszeiten im System zu:

$$\sum t_{process} = T_1 + 2 \cdot T_{Cycle,2} + T_0 = 2,29 \text{ ms} \quad (176)$$

Entsprechend der bisherigen Vorgehensweise werden die synchronen Teilprozesse zusammengefasst. Damit ergibt sich das vereinfachte Modell in Abbildung 119. Die Zykluszeit des Teilprozesses *ZusammenEingabe* ergibt sich aus der größten Zykluszeit eines Teilprozesses, dies ist *SendeAnSPS2* mit $T_{Cycle,1} = 2 \text{ ms}$. *ZusammenAusgabe* hat dementsprechend die Zykluszeit $T_{Cycle,2} = 1 \text{ ms}$.

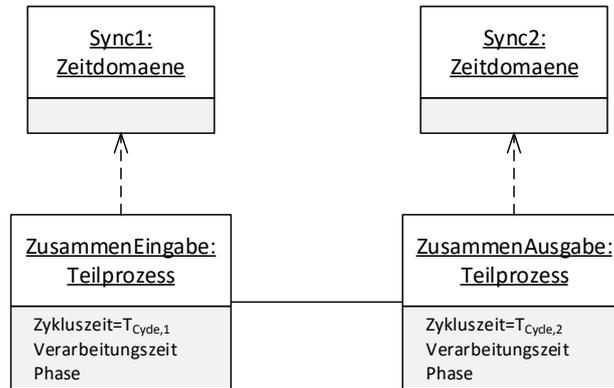


Abbildung 119: Zusammengefasster statischer Modellanteil für Szenario D

Im nächsten Schritt werden die Wartezeiten durch asynchrone Zyklusübergänge abgeleitet. Am Eingang ergibt sich eine Wartezeit im Intervall:

$$t_{\text{wait},1} \in [0; T_{\text{Cycle},1}) \quad (177)$$

Die Wartezeit am Übergang zwischen *ZusammenEingabe* und *ZusammenAusgabe* ergibt sich nach Gleichung (61) als Intervall:

$$t_{\text{wait},2} \in [0; T_{\text{Cycle},2}) \quad (178)$$

Die Klemme-Klemme-Reaktionszeit konnte damit auf die Summe der beiden rechteckverteilten Wartezeiten $t_{\text{wait},1}$ und $t_{\text{wait},2}$ und der konstanten Verarbeitungszeit $\sum t_{\text{process}}$ vereinfacht werden.

Da die Verteilungen analytisch beschrieben sind, erfolgt die Berechnung des Ergebnisses durch analytische Faltung. Nach Kapitel 4.4.4.4.1 und 4.4.4.2.2 ergibt sich eine Trapezverteilung, die um $\sum t_{\text{process}}$ nach rechts verschoben wurde (siehe Abbildung 127). Anhand der Parameter liegt die Verteilung der Klemme-Klemme-Reaktionszeit damit im Intervall:

$$t_{\text{IORT}} \in \left[\sum t_{\text{process}} ; \sum t_{\text{process}} + T_{\text{Cycle},1} + T_{\text{Cycle},2} \right) \quad (179)$$

$$t_{\text{IORT}} \in [2,29 \text{ ms}; 5,29 \text{ ms}) \quad (180)$$

7.1.3 Messergebnisse und Vergleich zur Simulation

Zur Verifikation der berechneten Klemme-Klemme-Reaktionszeiten wurden Messungen durchgeführt. Um eine statistisch verlässliche Aussage treffen zu können, wurden sämtliche Messungen mit mindestens 200 000 Messpunkten ausgeführt. Ein Messpunkt ist dabei ein Durchlauf des Testsignals durch das Steuerungssystem.

Um einen Vergleich zwischen den gemessenen und den berechneten Häufigkeitsverteilungen vorzunehmen, werden diese nebeneinander dargestellt.

Neben dem optischen Vergleich und den Vergleich der Eckpunkte wird die quadratische Abweichung zwischen gemessener und berechneter Häufigkeitsverteilung berechnet. Die quadratische

Abweichung wird auf die gemessene Verteilung normiert. Das Integral der Abweichung liefert eine Kenngröße. Diese wird auch als Chi-Quadrat-Größe bezeichnet (siehe [GHK94, S. 73]).

$$\chi^2 = \int \frac{(h_{\text{berechnet}}(t) - h_{\text{gemessen}}(t))^2}{h_{\text{gemessen}}(t)} dt \quad (181)$$

Für diskrete Verteilungen mit der Schrittweite Δt ergibt sich:

$$\chi^2 = \Delta t \cdot \sum_k \frac{(h_{\text{berechnet}}(k) - h_{\text{gemessen}}(k))^2}{h_{\text{gemessen}}(k)} \quad (182)$$

7.1.3.1 Szenario A: PROFINET IRT mit taktsynchronen EAs

Die gemessene Häufigkeitsverteilung (Abbildung 120) entspricht annähernd der Rechteckverteilung. Dies entspricht der Erwartung. Die Ausprägung der Überhöhungen am linken und rechten Rand ist auf das Messprinzip zurückzuführen und wurde in 6.3.1.1 untersucht.

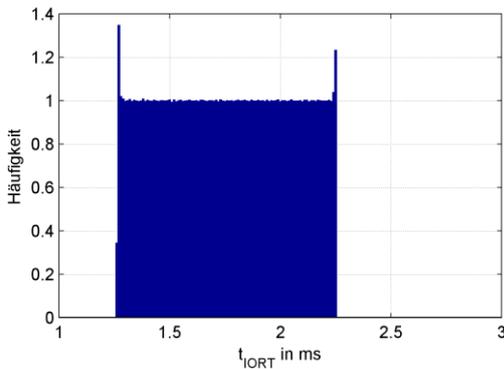


Abbildung 120: Gemessene Verteilung der Klemme-Klemme-Reaktionszeit für Szenario A

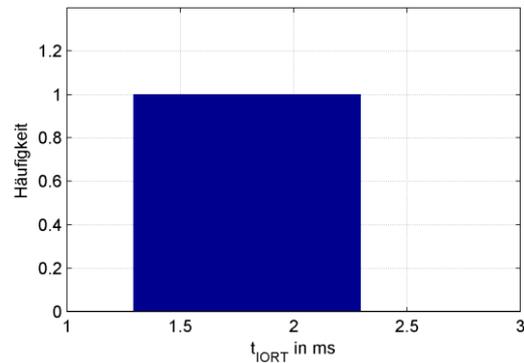


Abbildung 121: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit für Szenario A

In Tabelle 3 sind zum Vergleich Messwerte und berechnete Erwartungswerte dargestellt. Abgesehen von den Überhöhungen ist eine sehr gute Übereinstimmung zu erkennen.

Tabelle 3: Vergleich von gemessener und berechneter IORT für Szenario A

Gemessen	Berechnet
$\min(t_{\text{IORT}}) = 1,26 \text{ ms}$	$\min(t_{\text{IORT}}) = 1,29 \text{ ms}$
$\max(t_{\text{IORT}}) = 2,25 \text{ ms}$	$\max(t_{\text{IORT}}) = 2,29 \text{ ms}$
$\max(t_{\text{IORT}}) - \min(t_{\text{IORT}}) = 0,99 \text{ ms}$	$\max(t_{\text{IORT}}) - \min(t_{\text{IORT}}) = 1,00 \text{ ms}$

Die Summe der quadratischen Abweichungen liefert:

$$\chi^2 = 0,0377 \quad (183)$$

7.1.3.2 Szenario B: PROFINET IRT ohne takttsynchrone EAs

Die gemessene Häufigkeitsverteilung ist in Abbildung 122 dargestellt.

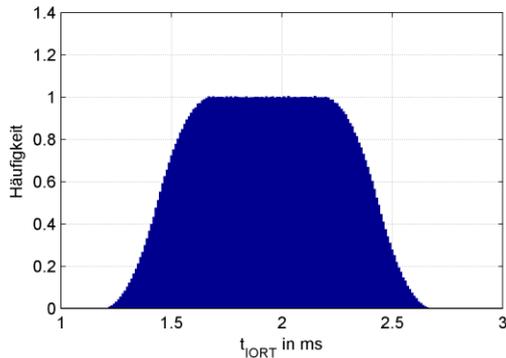


Abbildung 122: Gemessene Verteilung der Klemme-Klemme-Reaktionszeit für Szenario B

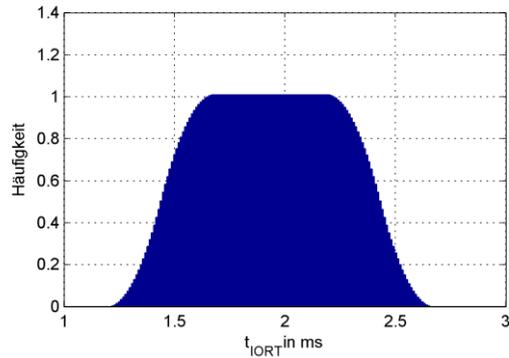


Abbildung 123: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit für Szenario B

Der Vergleich zeigt deutlich die Ähnlichkeit von gemessener und berechneter Häufigkeitsverteilung. Während der Minimalwert der Berechnung durch Messung bestimmt wurde, sind die für die Form und Breite der Verteilung notwendigen Parameter aus dem Engineering-System bestimmt. Der Erwartungswert und damit das Modell für dieses Szenario wurden bestätigt.

Tabelle 4: Vergleich von gemessener und berechneter IORT für Szenario B

Gemessen	Berechnet
$\min(t_{IORT}) = 1,19 \text{ ms}$	$\min(t_{IORT}) = 1,19 \text{ ms}$
$\max(t_{IORT}) = 2,68 \text{ ms}$	$\max(t_{IORT}) = 2,69 \text{ ms}$
$\max(t_{IORT}) - \min(t_{IORT}) = 1,49 \text{ ms}$	$\max(t_{IORT}) - \min(t_{IORT}) = 1,50 \text{ ms}$

Die Summe der quadratischen Abweichungen liefert:

$$\chi^2 = 7,91 \cdot 10^{-4} \quad (184)$$

7.1.3.3 Szenario C: PROFINET RT

Die gemessene Häufigkeitsverteilung ist in Abbildung 124 dargestellt.

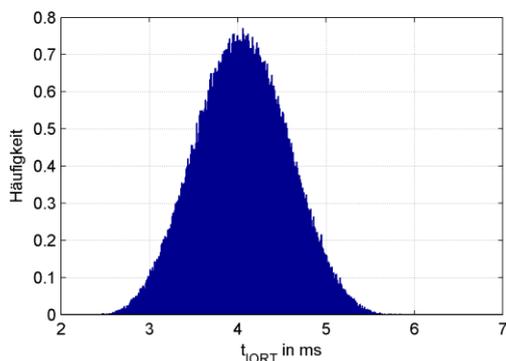


Abbildung 124: Gemessene Verteilung der Klemme-Klemme-Reaktionszeit für Szenario C

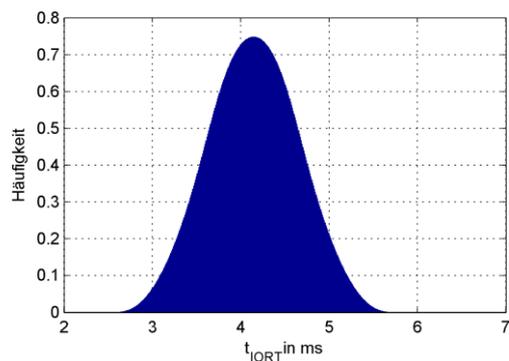


Abbildung 125: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit für Szenario C

In Tabelle 5 sind zum Vergleich Messwerte und berechnete Erwartungswerte dargestellt.

Tabelle 5: Vergleich von gemessener und berechneter IORT für Szenario C

Gemessen	Berechnet
$\min(t_{\text{IORT}}) = 2,45 \text{ ms}$	$\min(t_{\text{IORT}}) = 2,40 \text{ ms}$
$\max(t_{\text{IORT}}) = 6,29 \text{ ms}$	$\max(t_{\text{IORT}}) = 5,90 \text{ ms}$
$\max(t_{\text{IORT}}) - \min(t_{\text{IORT}}) = 3,84 \text{ ms}$	$\max(t_{\text{IORT}}) - \min(t_{\text{IORT}}) = 3,50 \text{ ms}$

Die Summe der quadratischen Abweichungen liefert:

$$\chi^2 = 0,0329 \quad (185)$$

Auch in diesem Szenario zeigen die Häufigkeitsverteilungen von gemessener und berechneter Klemme-Klemme-Reaktionszeit hohe Ähnlichkeit. Bei den Messwerten zeigen sich einzelne Ausreißer, die größer sind als die Erwartung und welche damit auch den gemessenen Jitter erhöhen. Diese lassen sich zum einen durch nicht-modellierte stochastische Effekte im System, zum anderen aber auch durch die Problematik des Rauschens im Messaufbau mit Zustandsdiskretem Testsignal erklären (siehe Abschnitt 6.3.1.1).

7.1.3.4 Szenario D: Verteiltes PROFINET-System

Die gemessene Häufigkeitsverteilung ist in Abbildung 126 dargestellt.

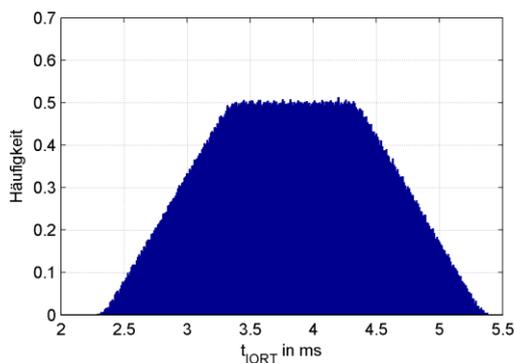


Abbildung 126: Gemessene Verteilung der Klemme-Klemme-Reaktionszeit in Szenario D

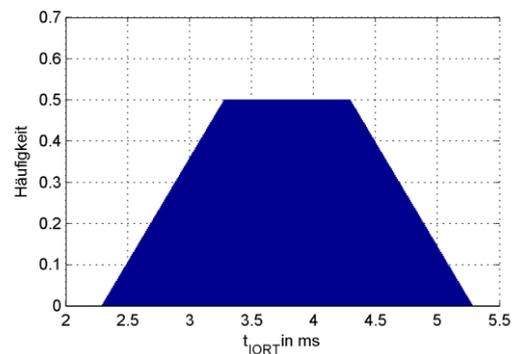


Abbildung 127: Berechnete Verteilung der Klemme-Klemme-Reaktionszeit von Szenario D

In Tabelle 6 sind zum Vergleich Messwerte und berechnete Werte dargestellt.

Tabelle 6: Vergleich von gemessener und berechneter IORT für Szenario D

Gemessen	Berechnet
$\min(t_{\text{IORT}}) = 2,28 \text{ ms}$	$\min(t_{\text{IORT}}) = 2,29 \text{ ms}$
$\max(t_{\text{IORT}}) = 5,39 \text{ ms}$	$\max(t_{\text{IORT}}) = 5,29 \text{ ms}$
$\max(t_{\text{IORT}}) - \min(t_{\text{IORT}}) = 3,11 \text{ ms}$	$\max(t_{\text{IORT}}) - \min(t_{\text{IORT}}) = 3,00 \text{ ms}$

Die Summe der quadratischen Abweichungen liefert:

$$\chi^2 = 0,0144 \quad (186)$$

Man erkennt deutlich die Trapezform, welche an den Knickpunkten abgerundet ist. Diese Abrundung ist zum einen auf nicht betrachtete stochastische Effekte zurückzuführen (die Verarbeitungszeiten wurden als konstant angenommen). Zum anderen wirkt sich auch hier der bei der Rechteckverteilung gezeigte Effekt aus, dass die Messung mit einem Rechtecksignal von endlichem Anstieg und

Signalrauschen ausgeführt wird. Dies führt zu einer Abweichung von der Idealform. Die Breite der Verteilung ist in der Messung mit 3,11 ms etwas größer als die berechneten 3 ms. Bei genauerer Betrachtung zeigt sich, dass die Breite der Trapezverteilung bei ca. 3 ms liegt. Der größere Wert entsteht durch einzelne Ausreißer am rechten Rand der Verteilung.

Trotz der Abweichungen durch stochastische Effekte können die Messungen als Bestätigung von Modell und Berechnung angesehen werden.

7.2 Analyse eines unbekanntes Systems aus Messdaten

In Abschnitt 6.4 wurden verschiedene Ansätze zur Bestimmung von Systemparametern aus gemessenen Klemme-Klemme-Reaktionszeiten vorgeschlagen. Das Vorgehen soll im Folgenden anhand von verschiedenen Verteilungen, welche ein reales System repräsentieren, gezeigt werden.

7.2.1 Analyse einer Trapezverteilung

An einem Steuerungssystem wurde mit einem Zustandsdiskreten Testsignal für die Klemme-Klemme-Reaktionszeit die Häufigkeitsverteilung in Abbildung 128 gemessen.

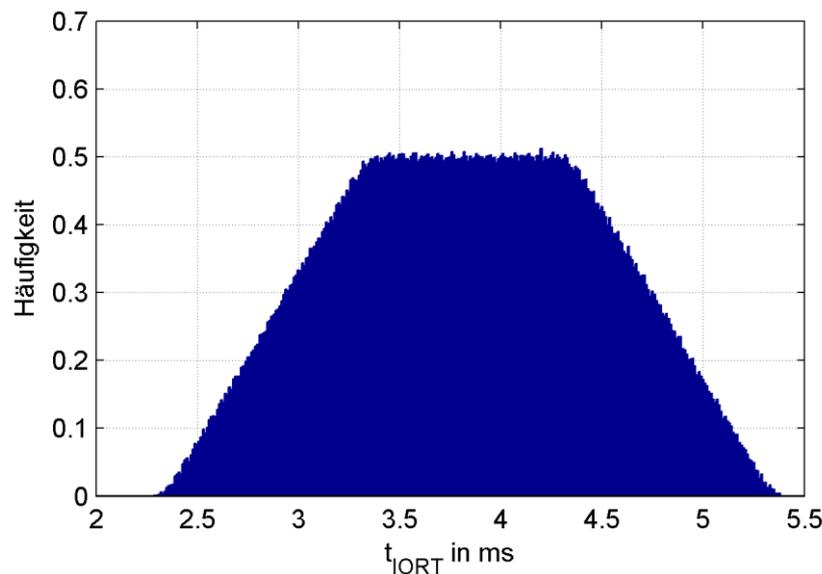


Abbildung 128: Klemme-Klemme-Reaktionszeit eines unbekanntes Systems

Man erkennt deutlich die Form einer Trapezverteilung, welche nach Abschnitt 4.4.4.4.1 das Ergebnis der Faltung von zwei Rechteckverteilungen $h_1(t)$ und $h_2(t)$ ist.

Zur Ermittlung der Parameter der Ausgangsverteilung werden zunächst die Knickpunkte der Häufigkeitsverteilung durch Geradenapproximation bestimmt (siehe Abbildung 129):

$$b_1 = 2,35 \text{ ms} \quad (187)$$

$$b_2 = 3,35 \text{ ms} \quad (188)$$

$$b_3 = 4,35 \text{ ms} \quad (189)$$

$$b_4 = 5,35 \text{ ms} \quad (190)$$

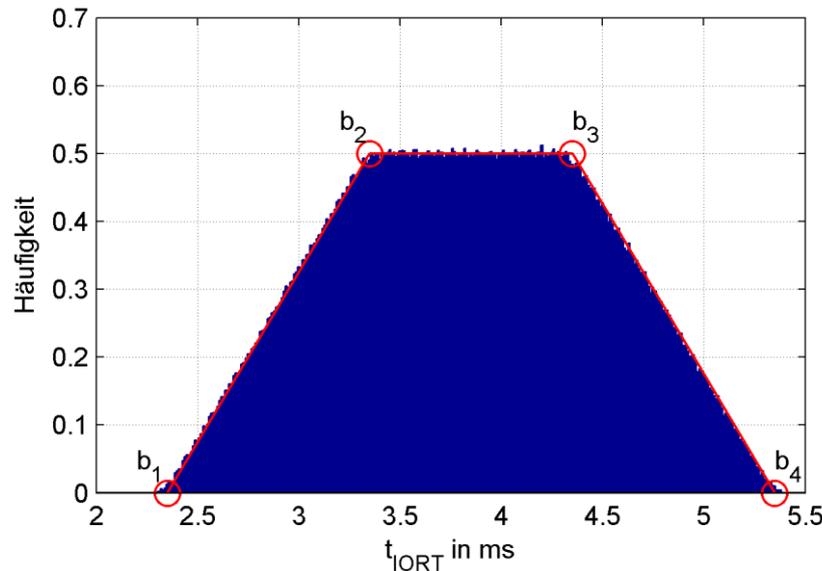


Abbildung 129: Geradeapproximation der Trapezverteilung des unbekanntes Systems

Anhand der Knickpunkte ergibt sich folgendes lineares Gleichungssystem (Herleitung in 4.4.4.4.1):

$$b_1 = a_1 + a_2 \quad (191)$$

$$b_2 = a_1 + a_2 + T_1 \quad (192)$$

$$b_3 = a_1 + a_2 + T_2 \quad (193)$$

$$b_4 = a_1 + a_2 + T_1 + T_2 \quad (194)$$

Die Trapezverteilung entsteht durch Faltung von zwei Rechteckverteilungen, deren Verzögerungen im Intervall $[a_1; a_1 + T_1)$ bzw. $[a_2; a_2 + T_2)$ liegen. Durch Lösen ergeben sich die Parameter der Rechteckverteilungen zu:

$$a_1 + a_2 = 2,35 \text{ ms} \quad (195)$$

$$T_1 = 1 \text{ ms} \quad (196)$$

$$T_2 = 2 \text{ ms} \quad (197)$$

Wie schon im Kapitel 4.3 beschrieben, gibt es keine Möglichkeit, die konstante Verarbeitungszeit auf die beiden Teilsysteme aufzuteilen. Des Weiteren kann die Reihenfolge der beiden Verzögerungen aufgrund der Kommutativität der Faltung nicht aus den Messergebnissen eines Signalpfades bestimmt werden.

Entsprechend der Messdaten kann der untersuchte Signalpfad aus zwei asynchron arbeitenden Teilprozessen bestehen, die mit einer Zykluszeit von 1 ms und 2 ms arbeiten. Die Summe der beiden Verarbeitungszeiten ist 2,35 ms. Für die beispielhafte Darstellung in Abbildung 130 wurde sie gleichmäßig aufgeteilt. Die Phasenlage kann bei asynchronen Teilprozessen nicht bestimmt werden.

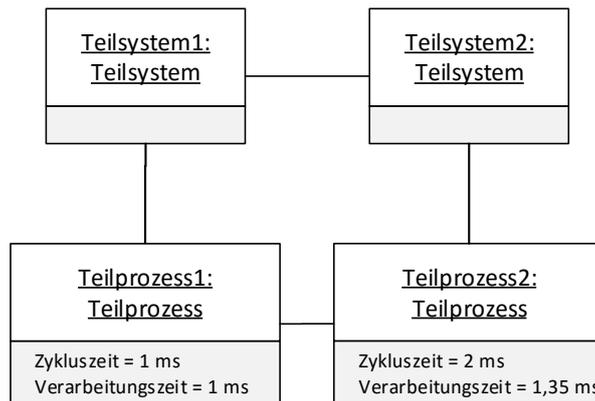


Abbildung 130: Mögliches Systemmodell des unbekanntes Systems

Man erkennt, dass die Approximation der Hüllkurve der gemessenen Verteilung die Schwierigkeit bei dieser Analyse ist. Je stärker verrauscht die Verteilung ist, d. h. je mehr stochastische Effekte (aus der Messung oder dem System) überlagern, desto schwieriger wird eine exakte Bestimmung.

7.2.2 Analyse einer Stufenverteilung

Ein zweites Beispiel für die Parameterbestimmung aus Messdaten wird anhand einer Stufenverteilung dargestellt. Diese Verteilung ist in Abbildung 131 dargestellt und bildet eine Vereinfachung der Messergebnisse aus [The13]. Die Häufigkeitsverteilung wurde durch Messung der Klemme-Klemme-Reaktionszeit mit zustandsdiskretem Testsignal (Rechtecksignal) bestimmt.

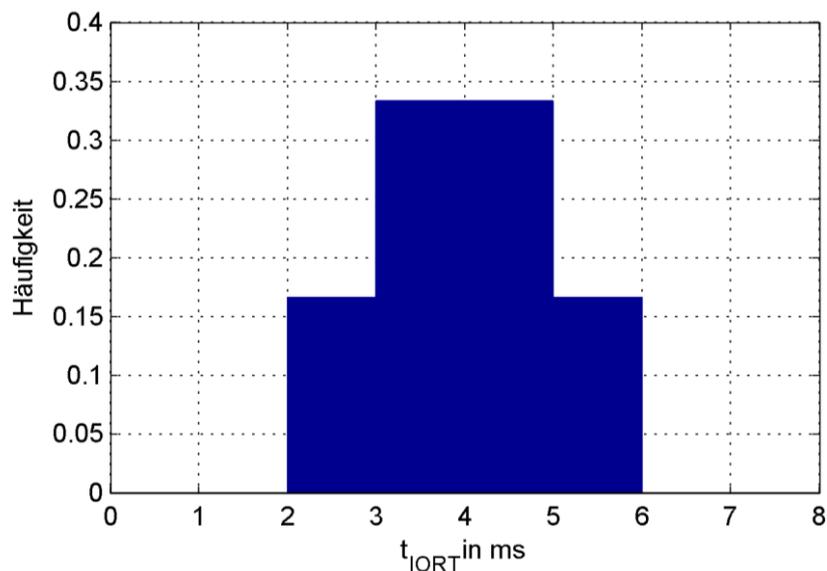


Abbildung 131: Klemme-Klemme-Reaktionszeit als Stufenverteilung

Diese Verteilung besteht aus mehreren Rechteckverteilungen, die additiv überlagert sind. Für die Wahl der Teilprozesse gibt es verschiedene Möglichkeiten.

7.2.2.1 Variante 1

In Abbildung 132 ist eine Variante der Summierung aus 4 Rechteckverteilungen gezeigt.

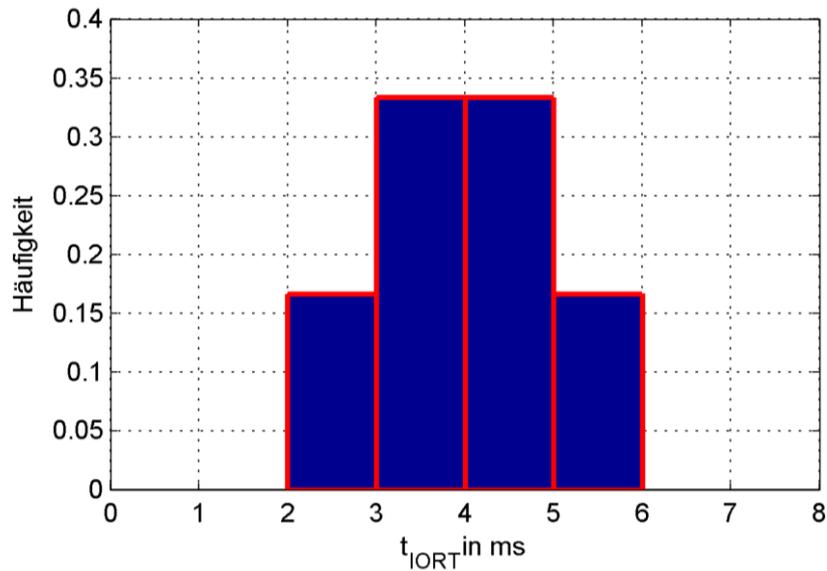


Abbildung 132: Stufenverteilung mit eingezeichneten Rechteckverteilungen – Variante 1

Der Ansatz für die Analyse sind die theoretischen Betrachtungen zur Faltung von zwei diskreter Gleichverteilungen (Abschnitt 4.4.4.3.1) und zur Faltung einer diskreten Gleichverteilung mit einer beliebigen Verteilung (4.4.4.3.2). Wird eine diskrete Gleichverteilung mit einer beliebigen Verteilung (z. B. einer Rechteckverteilung) gefaltet, so wird die Rechteckverteilung um jeden Wert der diskreten Gleichverteilung verschoben und skaliert, danach werden die verschobenen Verteilungen überlagert (addiert).

Damit könnte die Stufenverteilung als Faltung einer Rechteckverteilung der Breite 1 ms und einer diskreten Verteilung bestehend aus 4 Dirac-Verteilungen angesehen werden, siehe Abbildung 133.

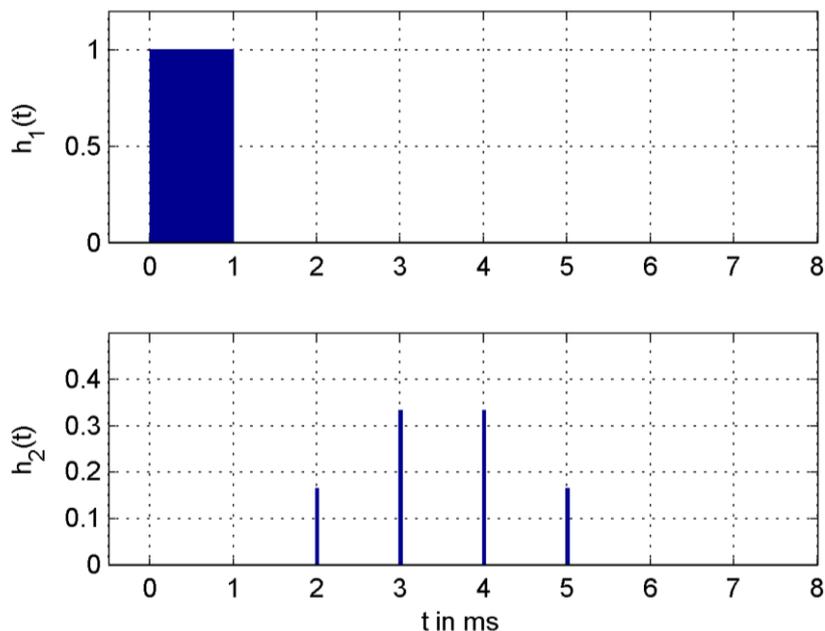


Abbildung 133: Stufenverteilung zerlegt in Rechteck- und Diskrete Verteilung – Variante 1

Die diskrete Verteilung kann wiederum als Faltung von zwei diskreten Gleichverteilungen angesehen werden, siehe Abschnitt 4.4.4.3.1.2. Wie im dortigen Beispiel gezeigt, ergibt sich diese Verteilung durch Faltung von zwei diskreten Gleichverteilungen mit den Parametern:

$$t_1 \in \{0; 1 \text{ ms}\} \quad (198)$$

$$t_2 \in \{2 \text{ ms}; 3 \text{ ms}; 4 \text{ ms}\} \quad (199)$$

Aus der zweiten diskreten Gleichverteilung wird die konstante Verzögerung (Dirac-Verteilung) extrahiert, so dass im Ergebnis die Stufenverteilung als Faltung von 4 Ausgangsverteilungen angesehen werden kann (siehe Abbildung 134):

- Rechteckverteilung $h_1(t)$
- Zwei diskrete Gleichverteilungen $h_3(t)$ und $h_4(t)$
- Dirac-Verteilung $h_5(t)$

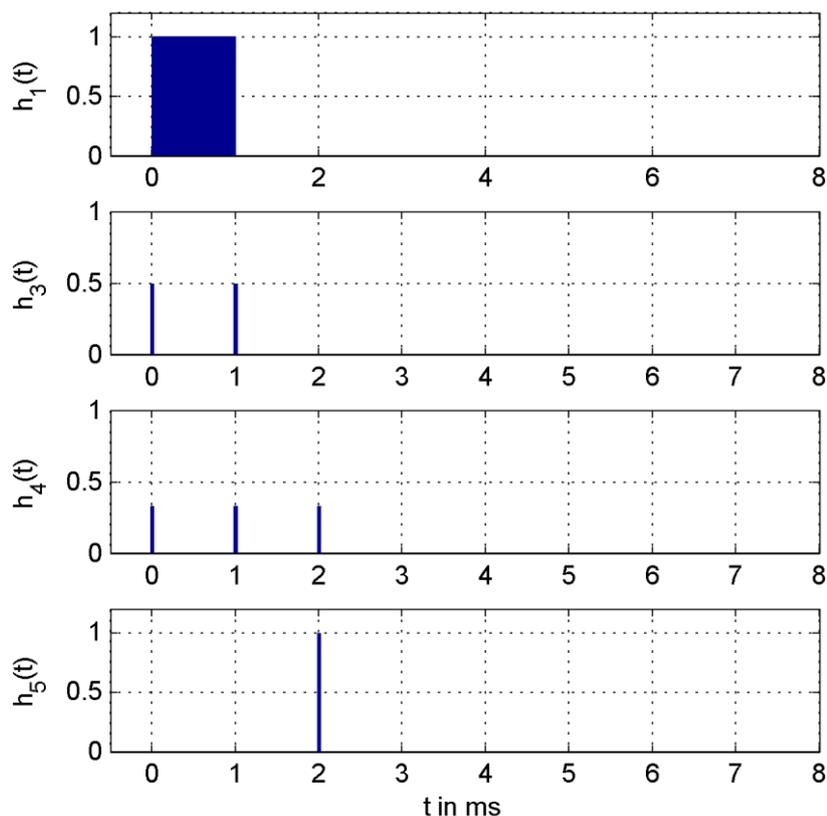


Abbildung 134: Stufenverteilung zerlegt in Rechteck-, diskrete Gleich- und Dirac-Verteilung

Im nächsten Schritt werden den Verzögerungen Elemente im Signalpfad zugeordnet:

Die Rechteckverteilung $h_1(t)$ entspricht einem asynchronen Zyklusübergang, der aufgrund des zustandsdiskreten Testsignals am Eingang des Signalpfades liegen muss. Aus der Breite folgt die Zykluszeit des ersten Teilprozesses: $T_{\text{Cycle},1} = 1 \text{ ms}$.

Die diskreten Gleichverteilungen entstehen als Wartezeiten beim Zusammenfassen von synchronen Teilprozessen mit gemeinsamen Basistakt, siehe Kapitel 4.3.3.2.2.

Die Gleichverteilung $h_3(t)$ entspricht der Wartezeit zwischen zwei synchronen Teilprozessen mit einem Basistakt von $T_{Base} = 1$ ms. Die kleinere der Zykluszeiten der beiden synchronen Teilprozesse beträgt $T_{Cycle,1} = 2$ ms, die zweite Zykluszeit $T_{Cycle,2}$ ist ein Vielfaches von T_{Base} , wobei $T_{Cycle,2} > T_{Cycle,1}$.

Die Gleichverteilung $h_4(t)$ entspricht ebenfalls der Wartezeit zwischen 2 synchronen Teilprozessen mit einem Basistakt von $T_{Base} = 1$ ms. Die kleinere der Zykluszeiten beträgt hier $T_{Cycle,3} = 3$ ms, für die zweite Zykluszeit gilt: $T_{Cycle,4} > T_{Cycle,3}$

Die Dirac-Verteilung $h_5(t)$ repräsentiert eine konstante Verarbeitungszeit, in der Regel die Summe der konstanten Verarbeitungszeiten entlang des Signalpfades.

Im letzten Schritt wird ein mögliches Modell erstellt (siehe Abbildung 135). Die Teilprozesse 1 und 2 bzw. 3 und 4 arbeiten synchron entsprechend der diskreten Gleichverteilungen. Die Rechteckverteilung muss am Eingang des Signalpfades liegen, bestimmt damit also die Zykluszeit der Zusammenfassung aus Teilprozess1 und Teilprozess2.

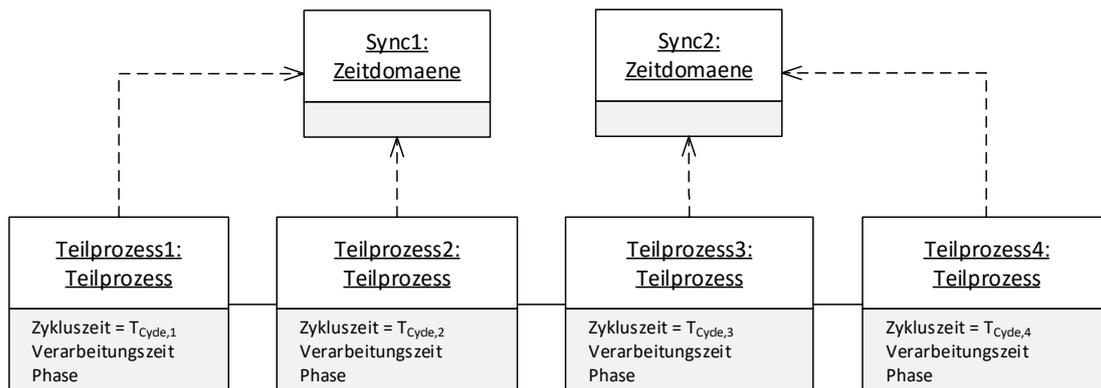


Abbildung 135: Mögliches Modell für Signalpfad mit Stufenverteilung – Variante 1

Das Ableiten des Zeitverhaltens aus dem in Abbildung 135 gezeigten Modell ergibt verschiedene **Widersprüche**:

Zwischen den beiden synchronen Teilprozessgruppen findet ein asynchroner Übergang statt, daher muss hier eine stochastische Wartezeit mit Rechteckverteilung eingefügt werden. Diese ist in den analysierten Messergebnissen nicht enthalten.

Die beiden synchronen Teilprozesse 1 und 2 werden zusammengefasst, der resultierende Teilprozess erhält die größere der beiden Zykluszeiten. Entsprechend der obigen Betrachtungen muss diese Zykluszeit mindestens 3 ms betragen (kleinster möglicher Wert für $T_{Cycle,2}$). Die Rechteckverteilung hat aber nur eine Breite von 1 ms.

Aufgrund dieser Widersprüche liefert diese Analyse der Klemme-Klemme-Reaktionszeit **keine gültige Systemkonfiguration**, nachfolgend wird eine andere Zerlegung der Stufenverteilung gezeigt.

7.2.2.2 Variante 2

Für die Bildung der Stufenverteilung in Abbildung 131 aus überlagerten Rechteckverteilungen gibt es eine weitere Möglichkeit. Sie kann durch Aufsummierung von zwei identischen Rechteckverteilungen mit einer Breite von 3 ms entstanden sein. In Abbildung 136 ist eine solche Rechteckverteilung rot eingezeichnet.

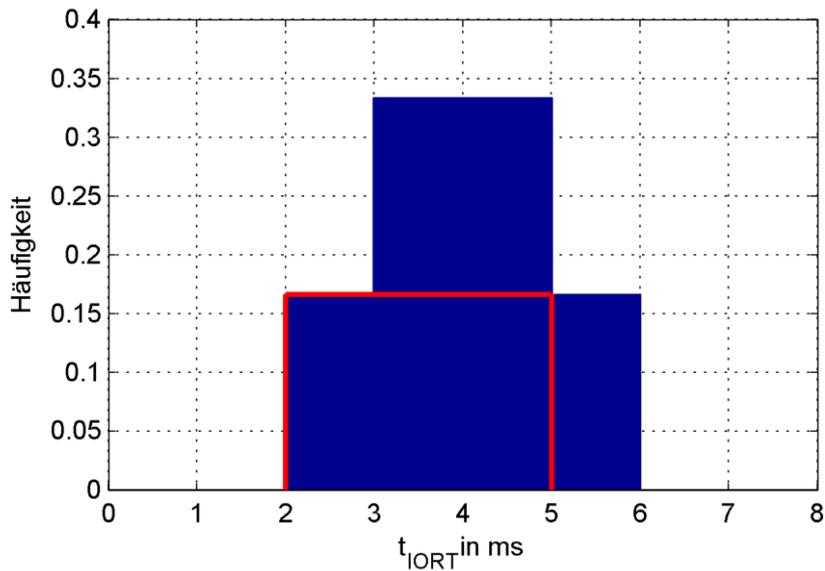


Abbildung 136: Stufenverteilung mit eingezeichneter Rechteckverteilung – Variante 2

Eine Zerlegung wie im vorigen Abschnitt liefert die in Abbildung 137 dargestellten Verteilungen.

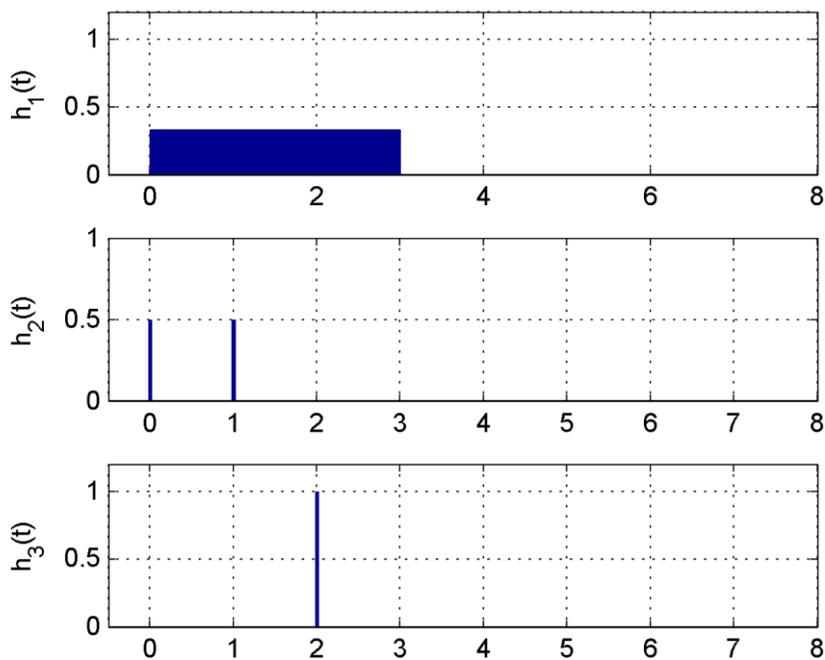


Abbildung 137: Stufenverteilung zerlegt in Rechteck-, diskrete Gleich- und Dirac-Verteilung

Im nächsten Schritt werden den Verzögerungen Elementen im Signalpfad zugeordnet:

Die Rechteckverteilung $h_1(t)$ entspricht einem asynchronen Zyklusübergang am Eingang des Signalpfades. Aus der Breite folgt die Zykluszeit des ersten Teilprozesses (nach der Zusammenfassung synchroner Teilprozesse): 3 ms.

Die diskreten Gleichverteilung $h_2(t)$ entspricht der Wartezeit zwischen zwei synchronen Teilprozessen mit einem Basistakt von $T_{\text{Base}} = 1$ ms. Die kleinere der Zykluszeiten der beiden synchronen Teilprozesse beträgt $T_{\text{Cycle},1} = 2$ ms, die zweite Zykluszeit $T_{\text{Cycle},2}$ ist ein Vielfaches von T_{Base} , wobei $T_{\text{Cycle},2} > T_{\text{Cycle},1}$.

Die Dirac-Verteilung $h_3(t)$ repräsentiert eine konstante Verarbeitungszeit von 2 ms. Dies ist die Summe der konstanten Verarbeitungszeiten entlang des Signalpfades.

Aus diesen Randbedingungen wird erneut ein mögliches Modell abgeleitet, siehe Abbildung 138. Das System besteht aus 2 synchronen Teilprozessen, die kleinere Zykluszeit beträgt $T_{\text{Cycle},1} = 2$ ms. Die Zykluszeit des langsameren Teilprozesses kann aus der Breite der Rechteckverteilung bestimmt werden: $T_{\text{Cycle},2} = 3$ ms. Die Reihenfolge der Teilprozesse 1 und 2 kann nicht ermittelt werden.

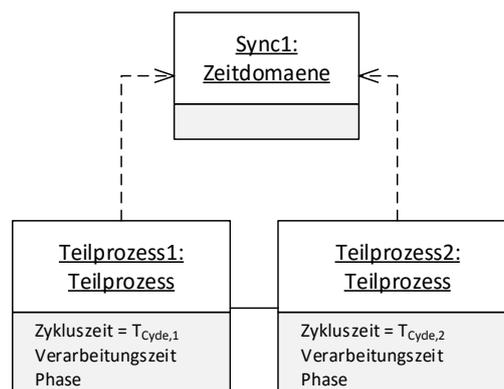


Abbildung 138: Mögliches Modell für Signalpfad mit Stufenverteilung – Variante 2

In dieser Variante tritt **kein Widerspruch** auf, daher ist es **eine mögliche Systemstruktur** für die gemessene Häufigkeitsverteilung der Verzögerung entlang des Signalpfades.

Es konnte gezeigt werden, dass die mathematische Zerlegung einer gemessenen Häufigkeitsverteilung nicht eindeutig ist. Bei der Faltung tritt Informationsverlust auf. Durch die Abbildung auf das Systemmodell können Varianten aufgrund von Widersprüchen ausgeschlossen werden.

Für größere Systeme kann die Bestimmung anhand der Messdaten eines Signalpfades uneindeutig sein. So ist besonders die Reihenfolge der Teilverzögerungen nicht aus einer Messung zu ermitteln. Hierfür müssen Messungen an verschiedenen Signalpfaden durchgeführt werden, die jeweils nur eine Untermenge der Teilprozesse enthalten.

7.2.3 Numerische Entfaltung

In Abschnitt 6.4.3 wurde die Nutzung der numerischen Entfaltung zur Analyse der Häufigkeitsverteilung von Messdaten vorgeschlagen. Dies soll im Folgenden exemplarisch durchgeführt werden.

Die hier gezeigte numerische Entfaltung wird mit Hilfe der Software MATLAB durchgeführt, diese bietet mit der Funktion *deconv* eine Implementierung der Entfaltung als Polynomdivision.

7.2.3.1 Entfaltung einer idealen Häufigkeitsverteilung

Als erstes Beispiel soll eine ideale Trapezverteilung entfaltet werden. Ausgangspunkt ist die in Abbildung 139 dargestellte Häufigkeitsverteilung $h_3(t)$, welche mit einer zeitlichen Schrittweite von $\Delta t = 0,01$ ms aufgelöst ist. Zunächst muss ein Faltungskern (Begriff nach [BL11, S. 113]) definiert werden, um den diese Verteilung entfaltet wird.

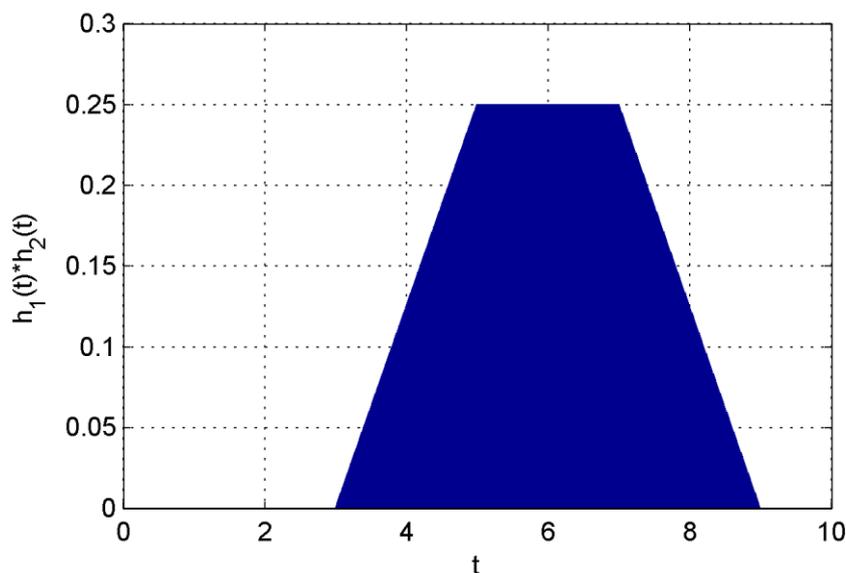


Abbildung 139: Ideale Häufigkeitsverteilung als Ausgangspunkt der Entfaltung

In Kapitel 4.4.4.4.1 wurde die Trapezverteilung als Faltung von zwei Rechteckverteilungen hergeleitet. Dabei wurde gezeigt, dass die Breite des ansteigenden Abschnittes der kleineren Zykluszeit entspricht. Daher wird als Faltungskern eine Rechteckverteilung $h_2(t)$ der Breite 2 ms genutzt.

Das Ergebnis der Entfaltung $h_1(t)$ ist in Abbildung 140 zusammen mit der Ausgangsverteilung $h_3(t)$ und des Faltungskerns $h_2(t)$ dargestellt.

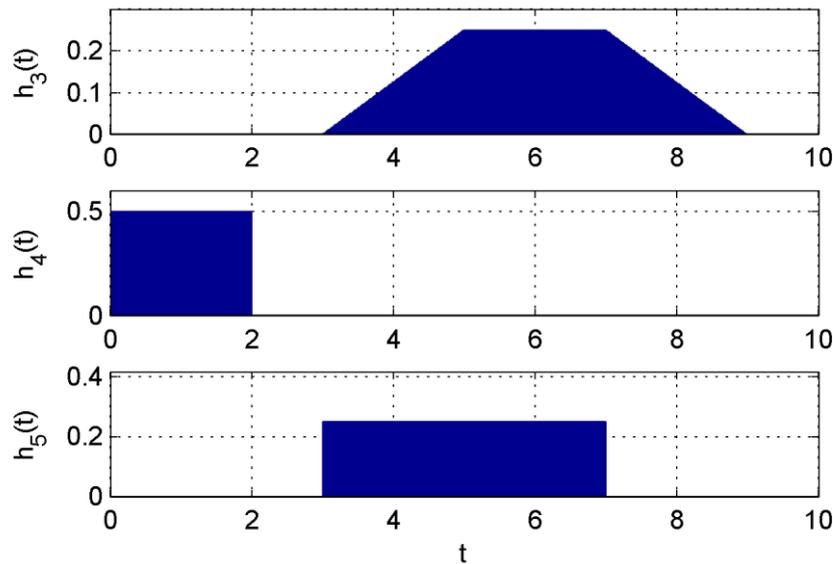


Abbildung 140: Ergebnis der Entfaltung einer idealen Häufigkeitsverteilung

Man erkennt, dass die numerische Entfaltung mittels Polynomdivision bei idealisierten Messwerten korrekte Ergebnisse liefert.

7.2.3.2 Entfaltung von realen Messwerten

Für die Nutzung mit (verrauschten) Messwerten ist diese Methode nicht geeignet. Sie liefert für die gemessene Trapezverteilung in Abbildung 128 kein Ergebnis. Um trotzdem eine Lösung zu erhalten, müsste zunächst eine Approximation der Verteilung erstellt werden, wie in Kapitel 7.2.1 gezeigt. Dann kann die numerische Entfaltung, wie im vorigen Abschnitt für ideale Werte gezeigt, funktionieren.

Alternativ können verschiedene Lösungsansätze aus der Literatur für die Entfaltung von verrauschten Messwerten genutzt werden. Dazu wurden in Abschnitt 6.4.3 verschiedene Verfahren genannt, für die genaue Umsetzung sei auf die Fachliteratur verwiesen.

7.3 Fazit

In diesem Kapitel wurde die Anwendung des vorgeschlagenen Modells zur Modellierung von Systemen und zur Berechnung von deren Zeitverhalten anhand von verschiedenen Beispielen gezeigt und verifiziert. Es wurde auch gezeigt, dass die empirische Bestimmung von Modellparametern nach den Methoden in Kapitel 6 möglich ist.

Je mehr asynchrone Teilprozesse in einem Signalpfad enthalten werden, desto größer wird aufgrund von nicht-modellierten Effekten die Abweichung zwischen berechneter und gemessener Häufigkeitsverteilung. Durch die Verwendung eines besseren Messaufbaus (weniger Rauschen, steilerer Flankenanstieg beim Signalgenerator) können die negativen Einflüsse des Messsystems (Überhöhungen, verschliffene Kanten) minimiert werden.

Außerdem wurden Beispiele für die Analyse eines (Teil-)Signalpfades aus Messwerten und die Erstellung eines Strukturmodells aus diesen Daten geliefert.

8 Anwendung des Modells zur Bestimmung der Regelgüte

Das in Kapitel 4 vorgestellte Modell und das daraus abgeleitete Zeitverhalten eines industriellen Steuerungssystems können für die Bewertung der Güte einer konkreten Regler-Implementierung genutzt werden. In Abschnitt 2.6 ist ein kurzer Überblick über die Modellierung von dynamischen Systemen und Regelkreisen gegeben. Daraus werden hier Kenngrößen für die Bewertung der Regelgüte abgeleitet. Danach wird anhand eines Beispiels die Nutzung des in dieser Arbeit vorgeschlagenen Modells zur Bewertung der mit verschiedenen Steuerungssystem-Konfigurationen erreichbaren Regelgüte gezeigt.

8.1 Kenngrößen zur Beurteilung der Regelgüte

Zur Bewertung der Güte einer Regelung existiert eine Vielzahl von Kenngrößen und Kriterien. Die Auswahl und Nutzung dieser ist von der betrachteten regelungstechnischen Aufgabe und deren Anforderungen abhängig. Je nach System und dessen Funktion können einzelne Kenngrößen eine hohe, geringe oder keine Bedeutung besitzen. Lunze beschreibt in [Lun14b] vier allgemeine Güteforderungen an einen Regelkreis.

8.1.1 Stabilität

Stabilität als Grundforderung an einen geschlossenen Regelkreis wurde in Abschnitt 2.6.4 definiert. Dabei wurde das in Zusammenhang mit dem Totzeitverhalten von technischen Systemen häufig verwendete **MATI-Kriterium** eingeführt, welches auch hier zur Bewertung der Stabilität herangezogen werden soll.

Bei der Nutzung des MATI-Kriteriums muss beachtet werden, dass dies ein sehr konservativer Ansatz ist. Bei der Bewertung wird nur die größte zu erwartende Zeitverzögerung eines Systems betrachtet. Wird das MATI-Kriterium erfüllt, kann das System dauerhaft mit dieser maximalen Zeitverzögerung stabilisiert werden.

In der Realität ist die Zeitverzögerung durch das Steuerungssystem jedoch selten konstant, sondern folgt einer Häufigkeitsverteilung. Wenn die Zeitverzögerung eines Steuerungssystems eine Häufigkeitsverteilung wie in Abbildung 141 mit einem Maximalwert von $\max(T_{IORT}) = 27 \text{ ms}$ hat und das MATI-Kriterium eine Stabilitätsgrenze von $\tau_{\max} = 25 \text{ ms}$ liefert, so ist das Stabilitätskriterium formal verletzt. Praktisch treten Zeitverzögerungen von mehr als 25 ms jedoch nur mit einer Häufigkeit von weniger als $10^{-4} \%$ auf. Um instabil zu werden, muss die Zeitverzögerung länger oberhalb der MATI-Grenze verbleiben. Das ist sehr unwahrscheinlich.

Ein solches System könnte somit als „praktisch stabil“ bezeichnet werden, d. h. es ist theoretisch instabil, praktisch ist dieser Fall aber zu vernachlässigen. Ob der Betrieb eines solchen Systems möglich ist, hängt letztendlich vom Prozess und dessen Sicherheitsanforderungen ab.

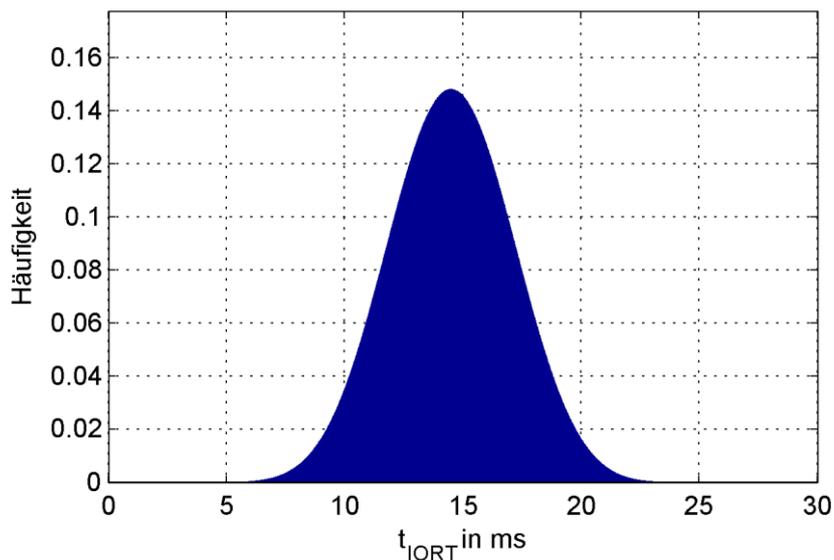


Abbildung 141: Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit mit $\max(t_{\text{IORT}}) = 27 \text{ ms}$

Greifeneder führt eine ähnliche Argumentation:

„Würde man von einem solchen System erwarten, dass es obige Spezifikationen grundsätzlich erfüllt (Worst-Case-Analyse), dann würde dies zu kaum erfüllbaren Anforderungen an die Hardware-Struktur führen, und somit die Verwendung von NAS ad absurdum führen. Allerdings ist es im Normalfall gar nicht notwendig, diese Spezifikationen grundsätzlich zu erfüllen, sondern es genügt, wenn sie mit großer Sicherheit erfüllt werden.“ [Gre07, S. 9]

Allgemein sind Stabilitätsbetrachtungen für Systeme mit Zeitverzögerungen Worst-Case-Analysen, so dass eine Kenntnis der Häufigkeitsverteilung keinen Mehrwert erzeugt. Die Kenntnis der Häufigkeitsverteilung gibt dem Entwurfsingenieur die Möglichkeit zur Bewertung des Risikos eines instabilen Systemverhaltens. Dadurch kann der sehr unwahrscheinliche Fall eines instabilen Verhaltens akzeptiert werden.

8.1.2 Störkompensation und Sollwertfolge

Ein Regelkreis muss so ausgelegt sein, dass die Regelgröße einer Änderung der Führungsgröße folgt und den Einfluss von Störgrößen ausgleicht.

„Für vorgegebene Klassen von Führungs- und Störsignalen soll die Regelgröße der Führungsgröße asymptotisch folgen.“ [Lun14b, S. 346]

$$\lim_{t \rightarrow \infty} (w(t) - y(t)) = 0 \quad (200)$$

Ist diese und die Forderung nach Stabilität erfüllt, „so hat der Regelkreis für große Zeit t keine bleibende Regelabweichung $e(\infty)$ “. [Lun14b, S. 347]

8.1.3 Dynamikforderungen

Während die Forderung nach Störkompensation und Sollwertfolge nur beschreibt, **dass** ein Regelkreis diese Aufgaben erfüllt, beschreiben die Dynamikforderungen **wie gut** diese Aufgaben erfüllt werden.

Die Reaktion der Regelgröße auf eine sprunghafte Änderung der Führungsgröße wird als **Sprungantwort** bezeichnet und kann zur Beurteilung der Güte des **Führungsverhaltens** einer Regelung verwendet werden. In Abbildung 142 ist eine Sprungantwort mit verschiedenen Kennwerten dargestellt.

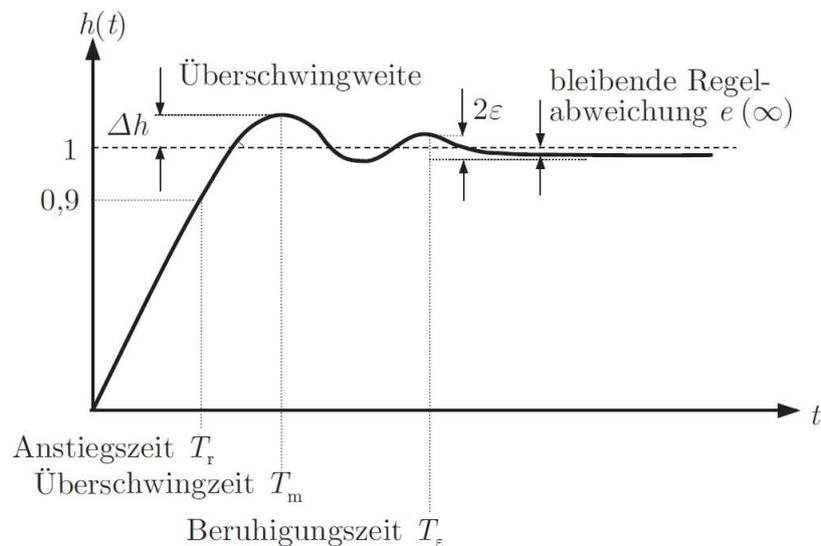


Abbildung 142: Führungsübergangsfunktion des Regelkreises [Lun14b, S. 347]

Das erste Überschwingen der Sprungantwort kann in Form der Zeit (Überschwingzeit T_m) und des Betrages (Überschwingweite Δh) als Gütekriterium genutzt werden. Weitere zeitliche Kriterien sind die Anstiegszeit T_r und die Einschwing- oder Beruhigungszeit T_e . Die Anstiegszeit gibt die Zeit an, welche die Regelgröße benötigt, um 90 % des statischen Endwertes zu erreichen. Die Einschwingzeit ist die Zeit, bis die Abweichung der Regelgröße vom statischen Endwert dauerhaft kleiner als ϵ ist. Entsprechend der Forderung im vorigen Abschnitt ist die bleibende Regelabweichung $e(\infty)$ charakterisiert als:

$$e(\infty) = \lim_{t \rightarrow \infty} (w(t) - y(t)) \quad (201)$$

Diese Kenngrößen für das Führungsverhalten können auch für die Reaktion des geschlossenen Regelkreises auf eine Störgröße, das sogenannte **Störungsverhalten**, angewendet werden (siehe Abbildung 143).

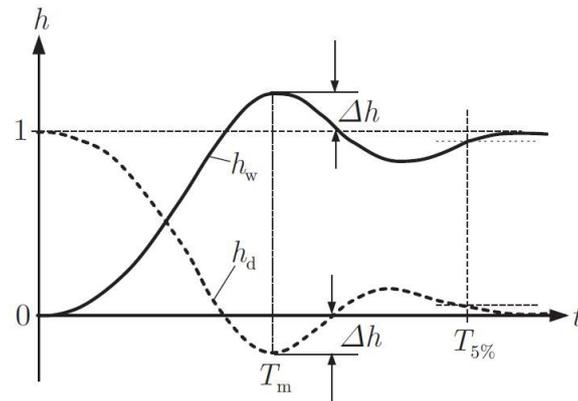


Abbildung 143: Übergangsfunktion bei sprunghafter Störung h_d bzw. Führung h_w [Lun14b, S. 348]

Für diese und weitere Kenngrößen des Regelkreises können abhängig von der konkreten Regelungsaufgabe Anforderungen aufgestellt werden.

8.1.4 Robustheitsforderungen

„Die Forderungen (I) – (III) sollen trotz Unsicherheiten im Regelstreckenmodell erfüllt werden.“ [Lun14b, S. 348] Das bedeutet, dass sowohl Modellunsicherheiten bei der Regelstrecke als auch beim Regler (als technisches System) nicht dazu führen dürfen, dass die Forderung nach Stabilität, Störkompensation, Sollwertfolge und Dynamik verletzt werden. Dies wird durch eine sogenannte Reserve erreicht, d. h. die Anforderungen werden für das Modell übererfüllt.

8.1.5 Fokus dieser Untersuchungen

Im Fokus dieser Arbeit liegen die Dynamikforderungen. Die Untersuchung der Stabilität als Worst-Case-Betrachtung ist ebenfalls mit dem vorliegenden Modell möglich. Untersuchungen zur Robustheit liegen nicht im Fokus dieser Arbeit. Zur Bewertung der Regelgüte werden folgende, in der Literatur häufig zu findende⁴⁵ Kenngrößen verwendet:

Die **Überschwingweite** Δh bezeichnet den größten Wert der Regelabweichung $e(t)$ (siehe auch Abbildung 142):

$$\Delta h = \max(e(t)) \quad (202)$$

Der **Integral Square Error** (ISE) gibt das Integral über dem Quadrat der Regelabweichung an:

$$ISE = \int_0^{\infty} e^2(t) dt \quad (203)$$

Um die zeitliche Dynamik stärker zu wichten, wird beim **Integral Time-weighted Absolute Error** (ITAE) der Betrag der Regelabweichung mit der Zeit multipliziert:

$$ITAE = \int_0^{\infty} t |e(t)| dt \quad (204)$$

Für alle drei Kenngrößen gilt, dass größere Werte eine schlechtere Güte bedeuten.

⁴⁵ siehe z. B. [MAC+08], [Pos09]

8.2 Analysemöglichkeiten

Der Ansatz zur Bewertung des Einflusses vom Zeitverhalten des Steuerungssystems auf die Güte des Regelkreises entspricht dem von Koller und Sauter in [KS04]. Der geschlossene Regelkreis wird mit dem Zeitverhalten des Steuerungssystems, beschrieben als Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit, simuliert und die Gütekriterien mit den optimalen Ergebnissen verglichen.

Der geschlossene Regelkreis wird dabei im Vergleich zur Grundstruktur in Abbildung 28 um ein Totzeit-Glied erweitert, dessen Wert zufällig entsprechend der Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit des Steuerungssystems variiert. So kann die Reaktion des geschlossenen Regelkreises auf Sollgrößen-Änderung oder Störgrößen-Einfluss simulativ untersucht werden.

Die Simulation wird nach dem Prinzip der Monte-Carlo-Simulation sehr oft wiederholt. Dabei wird der Wert der Totzeit entsprechend der gegebenen Häufigkeitsverteilung variiert. Durch eine genügend große Anzahl von Simulationen kann so eine Aussage über die möglichen Reaktionen des Systems getroffen werden.

Eine analytische Berechnung der Regelgüte ist aufgrund der unterschiedlichen Formen der Häufigkeitsverteilungen nicht oder nur mit sehr großem Aufwand möglich, weshalb hier der simulative Ansatz gewählt wurde.

8.3 Beispiel: Regelung eines Portalkrans mit PROFINET

Die Nutzung des vorgestellten Modells und des daraus ermittelten Zeitverhaltens zur Bewertung der erreichbaren Regelgüte soll beispielhaft an der Positions- und Pendelwinkel-Regelung eines Portalkrans gezeigt werden. Die hier gezeigten Ergebnisse wurden in [HPD14] veröffentlicht.

8.3.1 Portalkran

Pendelwinkeldämpfung bei einem Portalkran ist eine klassische regelungstechnische Aufgabe. In Abbildung 144 ist ein Portalkran mit einem Greifer als Last dargestellt. Die Position der Laufkatze ist s_K und der Pendelwinkel der Last ϑ . Das Ziel der Regelung ist es, beim Verfahren der Laufkatze den Pendelwinkel klein zu halten und somit das Schwingen der Last zu minimieren.

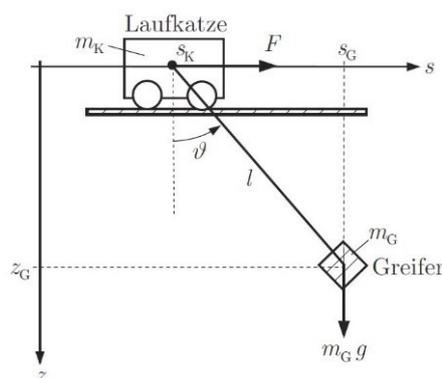


Abbildung 144: Portalkran [Lun14b, S. 128]

Der Portalkran wird, wie in [HPD14], durch folgendes Zustandsraummodell beschrieben:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 28 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -37.8 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0.286 \\ 0 \\ -0.286 \end{pmatrix} u, \quad (205)$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \quad (206)$$

Dabei geben die Zustandsgrößen x_1 und x_2 die Position und Geschwindigkeit der Laufkatze, die Größen x_3 und x_4 den Pendelwinkel und die Winkelgeschwindigkeit der Last an. Ausgangsgrößen des Systems sind die Katzposition y_1 und der Pendelwinkel y_2 .

Für dieses System wurde jeweils ein PD-Regler für die Position der Katze $R_p(s)$ und die Dämpfung des Pendelwinkels $R_s(s)$ entworfen (siehe [HPD14]):

$$R_p(s) = 31.6 + 41.8 s \quad (207)$$

$$R_s(s) = -241.7 - 59.7 s \quad (208)$$

Dies entspricht einer Zustandsrückführung:

$$K = (-31.6 \quad -41.8 \quad 241.7 \quad 59.7) \quad (209)$$

Die Struktur des Regelkreises mit der äquivalenten Zustandsrückführung ist in Abbildung 145 dargestellt.

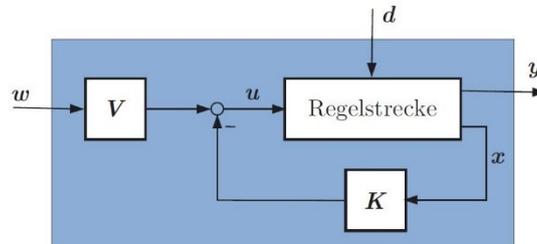


Abbildung 145: Regelkreis mit Zustandsrückführung K und Vorfilter V [Lun14a, S. 145]

Nach den Betrachtungen zur Architektur von NCS in Abschnitt 2.7.1.2 ist die Position der Zeitverzögerungen durch das Steuerungssystem im Regelkreis, der als Einschleifen-NCS betrachtet wird, frei wählbar.

8.3.2 Steuerungssystem

Die Regelung des Portalkrans erfolgt in diesem Beispiel mit einem Steuerungssystem bestehend aus einer SPS und zwei modularen Feldgeräten für die Ein- und Ausgabe der Daten (siehe Abbildung 146). Als Kommunikationssystem wird PROFINET eingesetzt, wobei die Konfiguration des Kommunikations- und Steuerungssystems in den verschiedenen Szenarien variiert.

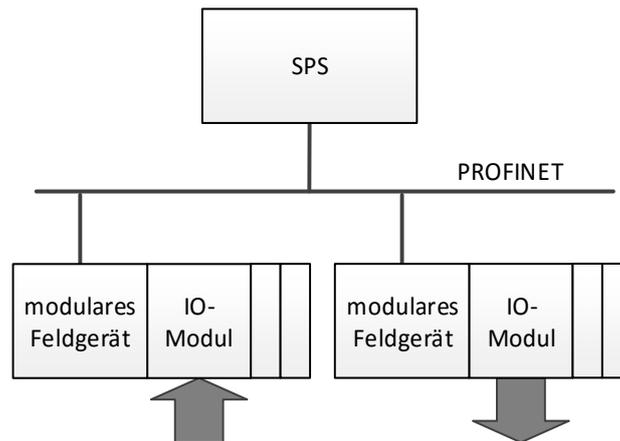


Abbildung 146: Architektur des Steuerungssystems zur Portalkran-Regelung

Die Szenarien entsprechen den in Kapitel 7.1 untersuchten Konfigurationen eines dezentralen Steuerungssystems. Da die Eingangssignale (Katzposition und Pendelwinkel) zustandskontinuierlich sind, tritt nach Abschnitt 4.3.6 kein asynchroner Übergang am Eingang auf.

Szenario A nutzt PROFINET IRT mit taktsynchronen EAs. Das bedeutet, dass alle Teilsysteme und -prozesse synchronisiert arbeiten und keine asynchronen Zyklusübergänge innerhalb des Steuerungssystems auftreten.

Szenario B nutzt PROFINET IRT ohne taktsynchrone EAs. Das bedeutet, dass zwischen den EA-Modul und der Kopfstation des Feldgeräts jeweils ein asynchroner Zyklusübergang stattfindet.

Szenario C nutzt PROFINET RT ohne isochrone Erweiterungen. Entsprechend Abschnitt 7.1.2.3 werden hier vier asynchrone Zyklusübergänge beobachtet.

Das Zeitverhalten des Steuerungssystems wird durch Berechnungen wie in Kapitel 7.1 (ohne den asynchronen Übergang am Eingang) für die drei Konfigurationen bestimmt. Die Parameter des Systems wurden entsprechend [HPD14] und [HD12] gewählt.

8.3.2.1 Zeitverhalten des Steuerungssystems - Szenario A

Da es bei der Nutzung von PROFINET IRT mit taktsynchronen EAs keine asynchronen Übergänge zwischen Teilprozessen gibt, ist die Klemme-Klemme-Reaktionszeit konstant. Die konstante Verarbeitungszeit des Systems beträgt $t_{\text{process}} = 1,26 \text{ ms}$.

8.3.2.2 Zeitverhalten des Steuerungssystems - Szenario B

Das Modell für Szenario B entspricht dem aus Abschnitt 7.1.2.2 mit folgenden Parametern:

$$T_{\text{Cycle},1} = 1 \text{ ms} \quad (210)$$

$$T_{\text{Cycle},2} = 0,25 \text{ ms} \quad (211)$$

$$t_{\text{process},1} = 1,9 \text{ ms} \quad (212)$$

$$t_{\text{process},2} = 1,15 \text{ ms} \quad (213)$$

Entsprechend der Berechnung in Abschnitt 4.4.4.1 ergibt sich die Klemme-Klemme-Reaktionszeit als Trapezverteilung (siehe Abbildung 147) mit folgenden Eckpunkten:

$$b_1 = t_{\text{process},1} + t_{\text{process},2} = 3,05 \text{ ms} \quad (214)$$

$$b_2 = t_{\text{process},1} + t_{\text{process},2} + T_{\text{Cycle},1} = 4,05 \text{ ms} \quad (215)$$

$$b_3 = t_{\text{process},1} + t_{\text{process},2} + T_{\text{Cycle},2} = 3,3 \text{ ms} \quad (216)$$

$$b_4 = t_{\text{process},1} + t_{\text{process},2} + T_{\text{Cycle},1} + T_{\text{Cycle},2} = 4,3 \text{ ms} \quad (217)$$

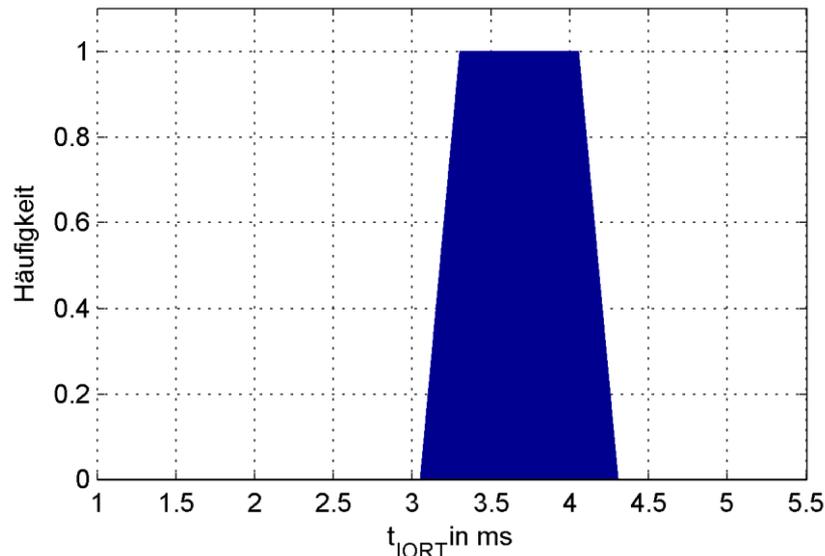


Abbildung 147: Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit für Szenario B

8.3.2.3 Zeitverhalten des Steuerungssystems - Szenario C

Dieses Szenario entspricht dem Modell in Abschnitt 7.1.2.3. Mit den Parametern aus [HD12] ergibt sich eine Klemme-Klemme-Reaktionszeit im Intervall $t_{IORT} \in [1,6 \text{ ms}; 5,35 \text{ ms}]$ (siehe Abbildung 148).

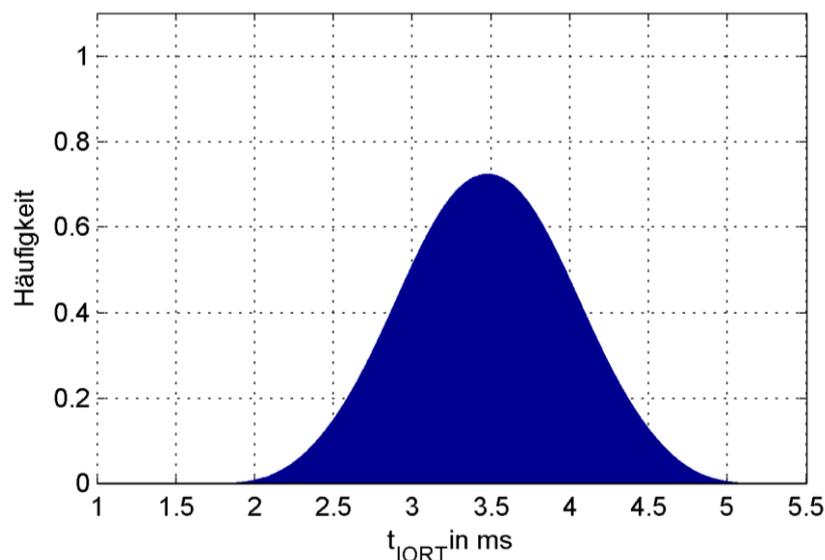


Abbildung 148: Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit für Szenario C

Die resultierende Zykluszeit des Steuerungssystems beträgt in allen Konfigurationen $T_{\text{Cycle}} = 1 \text{ ms}$.

8.3.3 Stabilität

Zur Untersuchung der Stabilität wird, wie in Abschnitt 8.1.1 beschrieben, das MATI-Kriterium verwendet. Wie in [HPD14] gezeigt, liegt die maximale Klemme-Klemme-Reaktionszeit des Steuerungssystems, bei der Stabilität garantiert werden kann, bei $\tau_{\max} = 4,9$ ms.

Dieses Kriterium ist für die Konfigurationen A und B, welche beide auf PROFINET IRT basieren, erfüllt. Szenario C ($\max(t_{IORT}) = 5,35$ ms) verletzt das MATI-Kriterium, jedoch nur mit sehr geringer Wahrscheinlichkeit. Entsprechend der Argumentation in Abschnitt 8.1.1 ist eine Verletzung des MATI-Kriteriums für mehrere aufeinanderfolgende Zyklen extrem unwahrscheinlich, weshalb diese Konfiguration weiter betrachtet wird.

8.3.4 Sprungantworten

Zur Bewertung der Regelgüte wird der geschlossene Regelkreis mit der Zeitverzögerung durch das Steuerungssystem simuliert. Die Simulation erfolgt mit MATLAB/Simulink, das entsprechende Modell ist in Anhang D dargestellt. Das System wird zum Zeitpunkt $t = 1$ s mit einem Eingangssprung der Amplitude $w(t = 1 \text{ s}) = 0,1$ beaufschlagt.

Abbildung 149 zeigt die Sprungantwort des Portalkrans für Szenario A (konstante Klemme-Klemme-Reaktionszeit). Die Abbildung 150 und Abbildung 151 zeigen die Kurvenschar der Sprungantworten für die beiden Szenarien B und C, jeweils für 100 Simulationen des Regelkreises mit zufällig (entsprechend der gegebenen Häufigkeitsverteilungen) variierten Zeitverzögerungen.

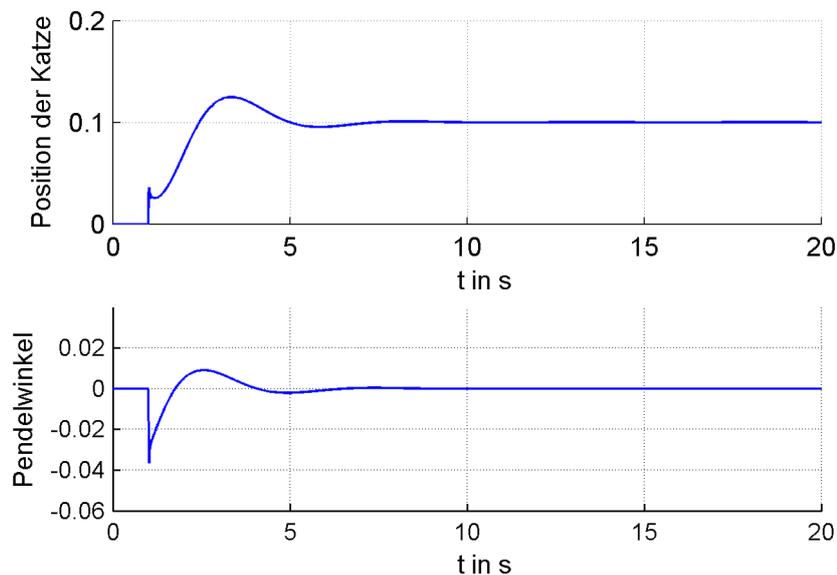


Abbildung 149: Sprungantwort für Szenario A

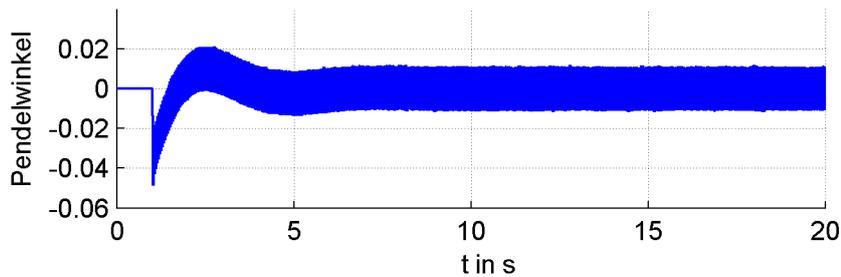
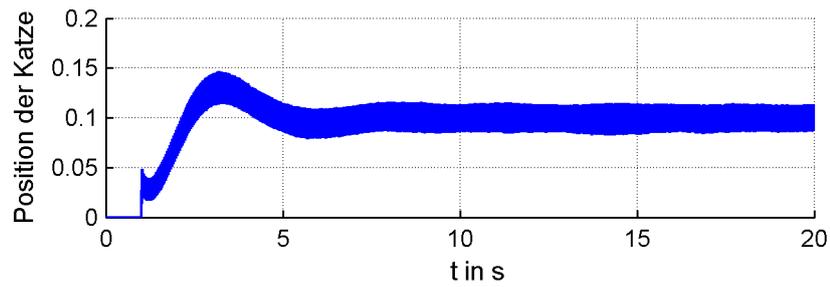


Abbildung 150: Kurvenschar der Sprungantworten für Szenario B

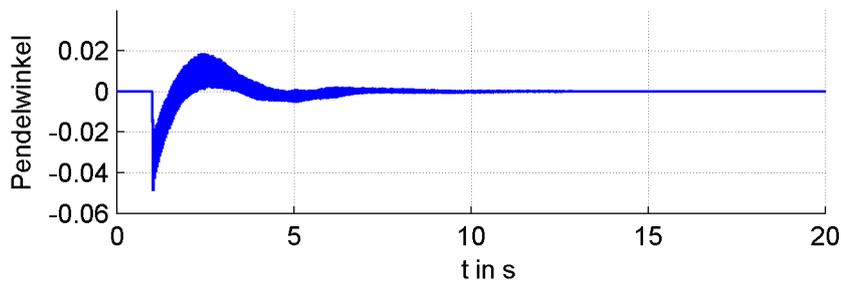
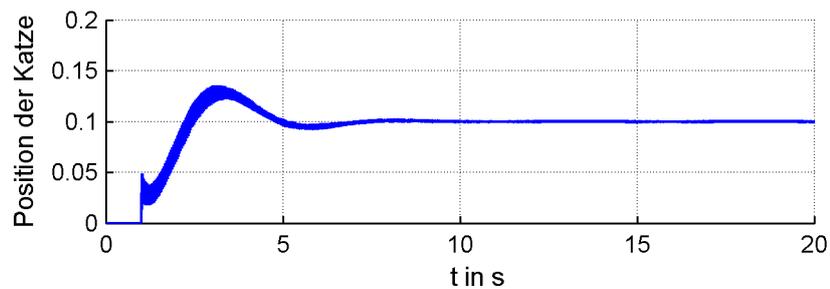


Abbildung 151: Kurvenschar der Sprungantworten für Szenario C

Man erkennt in Abbildung 150 dass die Sprungantworten für Szenario B schwingen. Die Darstellung einer einzelnen Sprungantwort für Szenario B in Abbildung 152 und der Ausschnitt in Abbildung 153 zeigen, dass sich eine bleibende, oszillierende Regelabweichung einstellt, welche auch für große Zeiten nicht verschwindet.

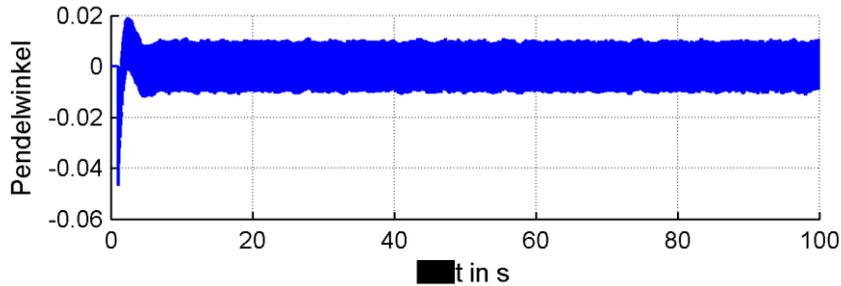
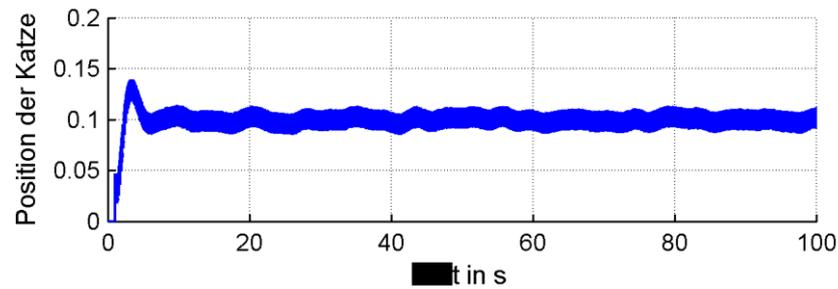


Abbildung 152: Einzelne Sprungantwort für Szenario B ($t=100$ s)

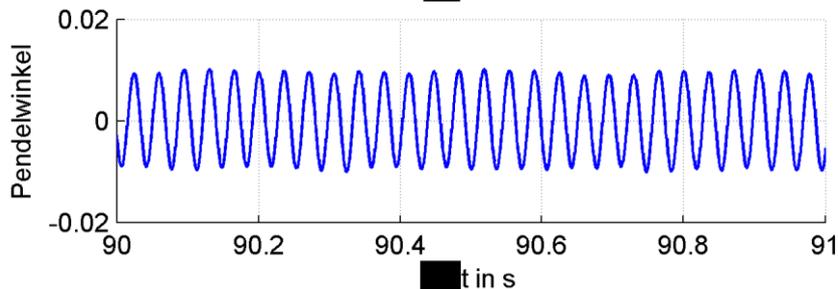
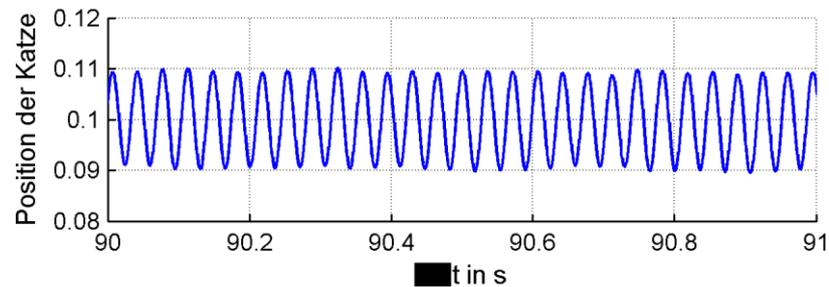


Abbildung 153: Ausschnitt aus Abbildung 152

Obwohl Szenario B mehr Synchronisationsmechanismen verwendet als Szenario C und der Jitter der Klemme-Klemme-Reaktionszeit geringer ist, zeigt die Sprungantwort ein schlechteres Verhalten des geschlossenen Regelkreises. Während der Jitter geringer ist (mehr Determinismus), ist der Mittelwert der Klemme-Klemme-Reaktionszeit bei Szenario B (3,675 ms) größer als bei Szenario C (3,475 ms). Die Synchronisation von Teilsystemen für mehr Determinismus kann dazu führen, dass Unsicherheiten durch Wartezeiten ausgeglichen werden, wie im vorliegenden Fall. Der größere Mittelwert der Klemme-Klemme-Reaktionszeit führt damit offensichtlich zu einem schlechteren Verhalten (Schwingungen) des geschlossenen Regelkreises.

Die Simulationen der Sprungantworten haben gezeigt, dass alle drei Szenarien stabil sind. Während Szenario C das MATI-Kriterium verletzt, treten bei Szenario B dauerhafte Schwingungen auf. Dies bestätigt die Aussagen zum konservativen Ansatz des MATI-Kriteriums.

8.3.5 Kenngrößen der Regelgüte

Für die Position der Katze und den Pendelwinkel werden die Kenngrößen Überschwingweite, ISE und ITAE für jedes Szenario bestimmt. Als Referenz wird eine Simulation ohne Zeitverzögerung durchgeführt. Für die Szenarien B und C ergeben sich keine konstanten Kenngrößen, da die Verzögerungen nicht konstant sind. Anhand von 1000 Simulationen der Sprungantworten wurden Häufigkeitsverteilungen der Kenngrößen von Szenario B und C bestimmt. Diese sind in Anhang E dargestellt.

Tabelle 7 zeigt für Referenz und Szenario A die Werte der Kenngrößen, für Szenario B und C sind Minimum, Maximum und Mittelwert angegeben. In Tabelle 8 sind die prozentualen Änderungen der Kenngrößen im Vergleich zur Referenz angegeben. Sie spiegelt die Verschlechterung der Regelgüte durch das Zeitverhalten des Steuerungssystems wider. Bei den verteilten Kenngrößen wird hier der schlechteste (größte) Wert verwendet.

Die Bestimmung der Integrale für ISE und ITAE wurde nicht bis $t \rightarrow \infty$ sondern entsprechend der Simulationszeit bis $t = 20$ s durchgeführt. Da Szenario B eine bleibende Regelabweichung aufweist, konvergiert das Integral des quadratischen Fehlers (ISE) nicht und wird daher nicht angegeben.

Tabelle 7: Kenngrößen der Regelgüte

Kenngröße		Referenz	Sz. A	Sz. B	Sz. C
$\Delta h_{\text{position}} \cdot 10^{-2}$	min	2,361	2,482	3,395	2,760
	mean			3,964	3,197
	max			4,664	3,746
$\Delta h_{\text{angle}} \cdot 10^{-2}$	min	-3,185	-3,644	-4,848	-4,882
	mean			-4,667	-4,554
	max			-4,501	-4,249
$ISE_{\text{position}} \cdot 10^{-3}$	min	4,827	4,774	-	4,619
	mean			-	4,699
	max			-	4,834
$ISE_{\text{angle}} \cdot 10^{-4}$	min	2,693	2,863	-	3,319
	mean			-	3,855
	max			-	4,609
$ITAE_{\text{position}}$	min	0,171	0,176	1,214	0,1796
	mean			1,247	0,1929
	max			1,328	0,2146
$ITAE_{\text{angle}}$	min	0,040	0,042	1,125	0,0436
	mean			1,132	0,0496
	max			1,138	0,1031

Tabelle 8: Prozentuale Änderung der Kenngrößen im Vergleich zum Referenzsystem

	Szenario A	Szenario B	Szenario C
$\Delta h_{\text{position}}$	5,1 %	97,5 %	58,6 %
Δh_{angle}	14,4 %	52,2 %	53,3 %
ISE_{position}	-1,1 %	-	0,1 %
ISE_{angle}	6,3 %	-	71,2 %
$ITAE_{\text{position}}$	2,6 %	674,7 %	25,2 %
$ITAE_{\text{angle}}$	2,7 %	2703,1 %	154,0 %

Tabelle 8 zeigt, dass die Regelgüte des vollständig synchronisierten Steuerungssystems (Szenario A) annähernd so gut ist wie die des Referenzsystems, welches keine Zeitverzögerungen durch das Steuerungssystem beinhaltet. Szenario B zeigt eine deutlich schlechtere Regelgüte. Ohne Synchronisationsmechanismen (Szenario C) ist die Güte schlechter als im Referenzsystem, jedoch besser als beim Szenario B. Entsprechend der größeren Streuung der Klemme-Klemme-Reaktionszeit streuen auch die Kenngrößen für Szenario C deutlich mehr.

Die Auswahl der geeigneten Konfiguration des Steuerungssystems für diese Regelungsaufgabe hängt von verschiedenen Randbedingungen ab. Hinsichtlich der Regelgüte ist eindeutig Szenario A mit vollständiger Synchronisation zu bevorzugen. Da sich der zusätzliche Aufwand in der Regel in deutlich höheren Investitionskosten widerspiegelt, können die anderen Szenarien die bessere Wahl sein. Bei geringeren Anforderungen hinsichtlich Regelgüte ist Szenario C mit den geringsten Hardwarekosten bei akzeptabler Regelgüte eine mögliche Alternative.

9 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Modell zur Beschreibung des Zeitverhaltens von verteilten industriellen Steuerungssystemen entwickelt. Die Untersuchungen wurden anhand von zyklisch arbeitenden Systemen durchgeführt, welche heute die überwiegende Mehrheit der industriellen Steuerungssysteme darstellen. Das Modell kann auch für die Beschreibung von ereignisbasierten Teilsystemen genutzt werden, eventuelle Erweiterungen wären zu prüfen. Aus dem entwickelten Modell wird das Zeitverhalten von Signalpfaden in Steuerungssystemen analytisch abgeleitet und berechnet. Basierend auf dieser Berechnung wird die Auswirkung verschiedener Konfigurationen eines Steuerungssystems auf die Güte von geschlossenen Regelkreisen untersucht.

Im Kapitel 2 findet eine Einordnung der Arbeit in den aktuellen Stand der Technik statt. Neben den Grundlagen zu industriellen Steuerungs- und Kommunikationssystemen werden bestehende Ansätze zur Modellierung von Kommunikations- und Steuerungssystemen sowie zur Berechnung von deren Zeitverhalten dargestellt. Dabei werden die verschiedenen Ansätze (Simulationen, Analytische Verfahren, ...) diskutiert. Es kann gezeigt werden, dass keiner der bisherigen Ansätze die Zielstellung dieser Arbeit, das Zeitverhalten verteilter Steuerungssysteme analytisch zu beschreiben, erfüllt. Im Anschluss daran werden grundlegende Aspekte zur Modellierung dynamischer Systeme und zur Disziplin der Networked Control Systems erklärt.

Um verschiedene Systeme und Konfigurationen vergleichbar zu machen, sind Kenngrößen erforderlich. Da die Namensgebung und Bedeutung der Kenngrößen über die Vielzahl der am Markt verfügbaren Systeme unterschiedlich ist, werden in Kapitel 3 Kenngrößen von Steuerungssystemen definiert. Dies ermöglicht bei der späteren Betrachtung eine eindeutige Verwendung der Begriffe.

Im Kapitel 4 wird ein Modell vorgestellt, welches ein Steuerungssystem in Teilsysteme und Teilprozesse mit zyklischer Arbeitsweise untergliedert. Dieses wird in statische und dynamische Modellanteile gegliedert und mit den Mitteln der UML beschrieben. Aus dieser Beschreibung wird das Zeitverhalten für beliebige Signalpfade durch das Steuerungssystem mittels Untersuchung der zeitlichen Effekte am Übergang zwischen den Teilprozessen abgeleitet. Ein Signalpfad schließt dabei ausdrücklich sowohl Kommunikationsteilsysteme als auch die Hardware- und Softwareanteile der Steuerungskomponenten ein. Die Berechnung des konkreten Zeitverhaltens eines Signalpfades geschieht durch das mathematische Mittel der Faltung. Damit ist ein durchgehend analytischer Modellierungs- und Berechnungsweg realisiert. Dieser ermöglicht den formalen Vergleich verschiedener Umsetzungskonzepte und Konfigurationen von verteilten industriellen Steuerungssystemen. Ergebnis ist ein Modell, das sowohl strukturelle als auch zeitliche Aspekte von industriellen Steuerungssystemen konsistent und formal beschreibt. Im Gegensatz zu den gezeigten Arbeiten in Kapitel 2.5 kann aus diesem Modell das Zeitverhalten eines konkreten Signalpfades vollständig analytisch bestimmt werden. Dies ermöglicht es, verschiedene Konfigurationen ohne großen Rechenaufwand (in Form von Simulationen) zu vergleichen.

Einen Sonderfall stellen die sogenannten isochronen Steuerungssysteme dar. Im Kapitel 5 wird auf die besonderen Aspekte dieser Systeme eingegangen und ein spezielles Modell für Bewertung und Analyse solcher Systeme vorgestellt. Dieses Modell kann aus dem in Kapitel 4 eingeführte Modell abgeleitet. Es ermöglicht jedoch die Betrachtung von spezifischen Besonderheiten dieser Systeme.

Eine wesentliche Herausforderung bei der Erstellung von Modellen ist die Parametrierung. Im Kapitel 6 werden die für das vorgeschlagene Modell notwendigen Parameter zusammengefasst und die verschiedenen Wege zur Bestimmung der Parameter evaluiert. Mögliche Datenquellen sind Datenblätter der Komponenten, Engineering-Systemen und Beschreibungssprachen. Wenn dadurch nicht alle Informationen verfügbar sind, können messtechnische Analysen durchgeführt werden.

In Kapitel 7 wird die Nutzung des Modells zur Beschreibung des Zeitverhaltens von verteilten Steuerungssystemen an verschiedenen Beispielen evaluiert. Dafür wird für mehrere Szenarien zunächst das Modell aufgestellt und parametrierung. Das abgeleitete Zeitverhalten wird anschließend mit Messungen an entsprechenden Beispielanlagen verglichen. Es konnte eine gute Übereinstimmung von berechneten und gemessenen Häufigkeitsverteilungen der Klemme-Klemme-Reaktionszeiten festgestellt werden. Außerdem werden die Ansätze zur Identifikation von Systemen und Teilsystemen anhand von Messdaten exemplarisch evaluiert.

Abschließend wird in Kapitel 8 die Nutzung des Modells zur Bewertung der Regelgüte mit verschiedenen Steuerungssystem-Konfigurationen untersucht. Dabei wird auf die verschiedenen Kenngrößen von Regelkreisen (Stabilität, Robustheit, Güte, ...) eingegangen. Am Beispiel eines Portalkrans wird für mehrere Szenarien von Steuerungssystemen die Regelgüte bestimmt. Diese Untersuchung kann einen Anlagenplaner bei der Auswahl geeigneter Steuerungssystem-Komponenten unterstützen. Während die Wahl des leistungsfähigsten Systems oft die beste Regelgüte erreicht, muss im Sinne der Kosteneffizienz ein Kompromiss aus Regelgüte und technischem Aufwand gefunden werden. Dies wird durch das vorgeschlagene Modell in besonderem Maße unterstützt.

Das gezeigte Modell und die Ableitung des Zeitverhaltens kann somit sehr gut für die Beschreibung und Analyse von verteilten industriellen Steuerungssystemen genutzt werden.

Oftmals wird die Klemme-Klemme-Reaktionszeit in der Literatur als (beschränkte) Gaußverteilung oder konstanter Wert angenommen. Dies führt zu abweichenden Ergebnissen bei der Ermittlung der erreichbaren Güte des geschlossenen Regelkreises. Es ist zu vermuten, dass diese Annahmen zu konservativen Ergebnissen bei der Auslegung von Steuerungssystemen führen. Damit wären die Kosten für solche Anlagen höher als nötig. Diese These sollte in Zukunft untersucht werden, um den zusätzlichen Aufwand zur Erstellung eines Modells zu rechtfertigen.

Im Rahmen der aktuellen Entwicklungen zum Thema Industrie 4.0 wird allgemein von einer steigenden Vernetzung der Komponenten innerhalb einer Produktionsanlage und auch zwischen diesen ausgegangen (siehe z. B. [Dra14]). Durch die zunehmende Verteilung von Funktionen wird der Signalpfad eines Datums durch solche Anlagen über viele verschiedene, nichtsynchronisierte Knoten laufen. Dies macht Untersuchungen zum Zeitverhalten, wie sie in dieser Arbeit durchgeführt wurden, umso wichtiger.

Literaturverzeichnis

- [And06] Anders, G.: Beitrag zur Verhaltensanalyse und Synchronisation von steuerungstechnischen Prozessen durch verteilte echtzeitfähige Kommunikationssysteme. Dissertation, Freiberg, 2006.
- [Aut14] AutomationML consortium: Whitepaper AutomationML. Part 5 - Communication, 2014.
- [BBK08] Bamberg, G.; Baur, F.; Krapp, M.: Statistik. Oldenbourg, München, 2008.
- [Bec13] Beckhoff: ETHERCAT-PLC mit 12,5 μ s Zykluszeit. In PC-Control, 2013; S. 22–23.
- [Beu07] Beucher, O.: Wahrscheinlichkeitsrechnung und Statistik mit MATLAB. Anwendungsorientierte Einführung für Ingenieure und Naturwissenschaftler. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [BH10] Binder, K.; Heermann, D. W.: Monte Carlo Simulation in Statistical Physics. An Introduction. Springer Berlin Heidelberg, Berlin/Heidelberg, 2010.
- [BHS10] Burget, P.; Hanzalek, Z.; Sucha, P.: Profinet IO IRT Message Scheduling With Temporal Constraints. In IEEE Transactions on Industrial Informatics, 2010, 6; S. 369–380.
- [Bib14] Bibliographisches Institut GmbH: Duden: synchronisieren. <http://www.duden.de/node/718110/revisions/1299005/view>, Geprüft am: 24.09.2014.
- [BKS10] Bester, N.; Krätzig, M.; Schimschar, A.: Bewertung von Funklösungen unter dem Aspekt der Anwendung in geschlossenen Regelkreisen: Komma 2010. 1. Jahreskolloquium „Kommunikation in der Automation“, Lemgo, Germany, 11.11.2010, 2010.
- [BL11] Bredies, K.; Lorenz, D.: Mathematische Bildverarbeitung. Einführung in Grundlagen und moderne Theorie. Vieweg + Teubner, Wiesbaden, 2011.
- [BR14] Bressler, K.; Rudolph, H.: Transistoren. In (Hering, E.; Bressler, K.; Gutekunst, J. Hrsg.): Elektronik für Ingenieure und Naturwissenschaftler, Springer Vieweg, Berlin, 2014; S. 163–254.
- [BS05] Bronstejn, I. N.; Semendjaev, K. A.: Taschenbuch der Mathematik. [mit CD-ROM DeskTop Bronstein]. Harri Deutsch, Frankfurt am Main, 2005.
- [CBS+12] Cena, G.; Bertolotti, I. C.; Scanzio, S.; Valenzano, A.; Zunino, C.: Evaluation of EtherCAT Distributed Clock Performance. In IEEE Transactions on Industrial Informatics, 2012, 8; S. 20–29.
- [CBV+07] Cena, G.; Bertolotti, I. C.; Valenzano, A.; Zunino, C.: Evaluation of Response Times in Industrial WLANs. In IEEE Transactions on Industrial Informatics, 2007, 3; S. 191–201.
- [CHL+03] Cervin, A.; Henriksson, D.; Lincoln, B.; Eker, J.; Årzén, K.-E.: How Does Control Timing Affect Performance? Analysis and Simulation of Timing Using Jitterbug and TrueTime. In IEEE Control Systems Magazine, 2003, 23; S. 16–30.
- [CO94] Chan, H.; Ozguner, U.: Closed-loop control of systems over a communication network with queues: ACC 1994. American Control Conference, Baltimore, USA, 29 June-1 July 1994, 1994; S. 811–815.

- [DH13] Diedrich, C.; Höme, S.: Diagnose für Industrial Ethernet. In Industrial ethernet journal. - Marburg: Technik-Dokumentations-Verl., 2013.
- [DHB+12] Donkers, M. C. F.; Heemels, W. P. M. H.; Bernardini, D.; Bemporad, A.; Shneer, V.: Stability analysis of stochastic networked control systems. In Automatica, 2012, 48; S. 917–925.
- [DHK02] Dingle, N. J.; Harrison, P. G.; Knottenbelt, W. J.: Response Time Densities in Generalised Stochastic Petri Net Models: WOSP 2002. 3rd International Workshop on Software and Performance, Rome, Italy, 24.-26.07.2002, ACM, New York, NY, USA, 2002; S. 46–54.
- [DIN03] DIN EN 61131-1: Speicherprogrammierbare Steuerungen - Teil 1: Allgemeine Informationen, DIN, 2003.
- [DIN13a] DIN EN 61158-5-10: Industrielle Kommunikationsnetze – Feldbusse – Teil 5-10: Dienstfestlegungen des Application Layer (Anwendungsschicht) – Typ 10-Elemente, DIN, 2013.
- [DIN13b] DIN EN 61158-4-12: Industrielle Kommunikationsnetze – Feldbusse – Teil 4-12: Protokollspezifikation des Data Link Layer (Sicherheitsschicht) – Typ 12-Elemente, DIN, 2013.
- [DIN15] DIN EN 61784-2: Industrielle Kommunikationsnetze – Profile – Teil 2: Zusätzliche Feldbusprofile für Echtzeitnetzwerke basierend auf ISO/IEC 8802-3, DIN, 2015.
- [DIN88] DIN 44300-9: Informationsverarbeitung - Begriffe - Verarbeitungsabläufe, DIN, 1988.
- [DLK+14] Diedrich, C.; Löber, P.; Keller, B.; Dach, S.; Höme, S.; Werner, T.; Damm, M.: Projektabschlussbericht Benchmark PROFINET IRT, 2014.
- [Dra14] Drath, R.: Industrie 4.0 – eine Einführung. In open automation, 2014, 2014; S. 2–7.
- [DS14] Diekhake, P.; Schnieder, E.: Analyse und Überwachung des Zeitverhaltens von Funktionsabläufen in einem verteilten Automatisierungssystem. In (Jumar, U.; Diedrich, C. Hrsg.): EKA 2014. Entwurf komplexer Automatisierungssysteme, Magdeburg, 14.-15.05.2014, ifak, Magdeburg, 2014.
- [DS15] Diekhake, P.; Schnieder, E.: Strukturierte Modellierung, Simulation und Überwachung verteilter Automatisierungssysteme. In at - Automatisierungstechnik, 2015, 63.
- [FA09] Früh, K. F.; Ahrens, W.: Handbuch der Prozessautomatisierung. Prozessleittechnik für verfahrenstechnische Anlagen. Oldenbourg, München, 2009.
- [FB08] Frey, T.; Bossert, M.: Signal- und Systemtheorie. Mit 26 Tabellen, 64 Aufgaben mit Lösungen und 84 Beispielen. Vieweg + Teubner, Wiesbaden, 2008.
- [FFR+08] Ferrari, P.; Flammini, A.; Rinaldi, S.; Gaderer, G.: Evaluation of clock synchronization accuracy of coexistent Real-Time Ethernet protocols: ISPCS 2008. IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, Ann Arbor, Michigan, USA, 22.-26.09.2008, IEEE, Piscataway, NJ, USA, 2008.
- [GC08] Gupta, R. A.; Chow, M.-Y.: Overview of Networked Control Systems. In (Wang, F.-Y.; Liu, D. Hrsg.): Networked Control Systems, Springer London, London, 2008; S. 1–23.

-
- [GF08] Greifeneder, J.; Frey, G.: Analyse der Reaktivität unterschiedlicher Architekturen Netzbasierter Automatisierungssysteme: Proceedings of the SPS/IPC/DRIVES, Nürnberg, November 2008, 2008; S. 517–525.
- [GHK94] Gabler, S.; Hoffmeyer-Zlotnik, J.H.P.; Krebs, D. Hrsg.: Gewichtung in der Umfragepraxis. VS Verlag für Sozialwissenschaften, Wiesbaden, s.l., 1994.
- [GLF07] Greifeneder, J.; Liu, L.; Frey, G.: Methoden zur Antwortzeitanalyse in vernetzten Automatisierungssystemen: Proceedings of the SPS/IPC/DRIVES, Nürnberg, November 2007, 2007; S. 517–525.
- [GM03] Gramann, T.; Mohl, D. S.: Precision Time Protocol IEEE 1588 in der Praxis. Zeitsynchronisation im Submikrosekundenbereich. In *Elektronik*, 2003, 24/2003; S. 86–94.
- [Gre07] Greifeneder, J.: Formale Analyse des Zeitverhaltens netzbasierter Automatisierungssysteme. Shaker, Aachen, 2007.
- [Gut14] Gutekunst, J.: Schnittstellen, Bussysteme und Netze. In (Hering, E.; Bressler, K.; Gutekunst, J. Hrsg.): *Elektronik für Ingenieure und Naturwissenschaftler*, Springer Vieweg, Berlin, 2014; S. 683–727.
- [GWE04] Gonzalez, R. C.; Woods, R. E.; Eddins, S. L.: *Digital image processing using MATLAB*. Prentice Hall, Upper Saddle River, NJ, 2004.
- [Had15] Hadlich, T.: Verwendung von Merkmalen im Engineering von Systemen. Dissertation, Magdeburg, 2015.
- [Har89] Harris, T. J.: Assessment of control loop performance. In *The Canadian Journal of Chemical Engineering*, 1989, 67; S. 856–861.
- [HD12] Höme, S.; Diedrich, C.: Analyse des Zeitverhaltens von Steuerungssystemen mit zyklischer Kommunikation: KomMA 2012. 3. Jahreskolloquium "Kommunikation in der Automation", Lemgo, 14.11.2012, 2012; S. 7.
- [HD14] Höme, S.; Diedrich, C.: Analytisch bestimmtes Zeitverhalten für verteilte Steuerungssysteme: KomMA 2014. 5. Jahreskolloquium "Kommunikation in der Automation", Lemgo, 18.11.2014, 2014.
- [HDD+13] Höme, S.; Diedrich, C.; Damm, M.; Werner, T.: Performancebenchmark für Steuerungen mit synchroner Kommunikation: Automation 2013. 14. Branchentreff der Mess- und Automatisierungstechnik, Baden-Baden, 25.-26.06.2013, VDI Verlag, Düsseldorf, 2013.
- [HI01] Horch, A.; Isaksson, A. J.: Assessment of the sampling rate in control systems. In *Control Engineering Practice*, 2001, 9; S. 533–544.
- [HI99] Horch, A.; Isaksson, A. J.: A modified index for control performance assessment. In *Journal of Process Control*, 1999, 9; S. 475–483.
- [Hil15] Hilscher: Ethernet-Analyzer Real-Time-Ethernet. NANL-B500E-RE. <http://www.hilscher.com/de/produkte/produktgruppen/analyse-und-datenerfassung/ethernet-analyse/nanl-b500e-re/?cats=>, Geprüft am: 27.01.2015.

- [Hir14] Hirschmann: White Paper – Präzise Uhrzeitsynchronisation. http://www.sgconnect.com/cms/documents/pdfs/Whitepapers/White_Paper_IEEE1588_de_v11.pdf, Geprüft am: 29.04.2014.
- [HJW10] Han, L.; Jasperneite, J.; Werner, T.: DIVAN: A network calculator for the off-line performance analysis of Virtual Automation Networks: WFCS 2010. 8th IEEE International Workshop on Factory Communication Systems, Nancy, France, 18.-21.05.2010, IEEE, Piscataway, NJ, USA, 2010; S. 293–302.
- [HKD14] Höme, S.; Keller, B.; Diedrich, C.: Präzision und Genauigkeit verschiedener Synchronisationsmechanismen und -verfahren von dezentralen AT-Systemen: Automation 2014. 14. Branchentreff der Mess- und Automatisierungstechnik, Baden-Baden, 01.-02.07.2014, VDI Verlag, Düsseldorf, 2014.
- [HNY07] Hespanha, J. P.; Naghshtabrizi, P.; Yonggang Xu: A Survey of Recent Results in Networked Control Systems. In Proceedings of the IEEE, 2007, 95; S. 138–162.
- [HOV02] Hespanha, J. P.; Ortega, A.; Vasudevan, L.: Towards the Control of Linear Systems with Minimum Bit-Rate: MTNS 2002. 15th International Symposium on the Mathematical Theory of Networks and Systems, Notre Dame, IN, USA, August 12-16, 2002, 2002.
- [HPD14] Höme, S.; Palis, S.; Diedrich, C.: Design of communication systems for networked control system running on PROFINET: WFCS 2014. 10th IEEE International Workshop on Factory Communication Systems, Toulouse, France, 05.-07. Mai 2014, IEEE, Piscataway, NJ, USA, 2014.
- [IAO07] IAONA: IAONA | Competence in Industrial Ethernet. <http://www.iaona.org/>, Geprüft am: 19.08.2014.
- [IEC07] IEC 61800-7-1: Adjustable speed electrical power drive systems - Part 7-1: Generic interface and use of profiles for power drive systems - Interface definition, IEC, 2007.
- [IEC08] IEC 62424: Representation of process control engineering - Request in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools, IEC, 2008.
- [IEC09] IEC 61588: Precision clock synchronization protocol for networked measurement and control systems, IEC, 2009.
- [IEC14] IEC 62714: Engineering data exchange format for use in industrial automation systems engineering - Automation markup language, IEC, 2014.
- [IEE03] IEEE 802.1Q: Standard for Local and metropolitan area networks--Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks, IEEE, 2003.
- [IEE12] IEEE 802.3: Standard for Ethernet, IEEE, 2012.
- [ifa11] ifak e.V. Magdeburg: Entwicklung von Standardtests zur einheitlichen Bewertung industrieller Funklösungen. Definition von Standardtestfällen, Magdeburg, 2011.
- [Int02] RTC 3411: An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, Internet Engineering Task Force, 2002.
- [Int06] RFC 4330: Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI, Internet Engineering Task Force, 2006.

-
- [Int10] RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification, Internet Engineering Task Force, 2010.
- [Int12] RFC 6815: Applicability Statement for RFC 2544: Use on Production Networks Considered Harmful, Internet Engineering Task Force, 2012.
- [Int99] RFC 2544: Benchmarking Methodology for Network Interconnect Devices, Internet Engineering Task Force, 1999.
- [Ise88] Isermann, R.: Digitale Regelsysteme. Springer, Berlin, 1988.
- [ISO98] ISO / IEC 7498: Information technology -- Open Systems Interconnection - Basic Reference Model: The Basic Model, ISO/IEC, 1998.
- [ITS96] Federal Standard 1037C: Federal Standard 1037C - Glossary of Telecommunication Terms, Institute for Telecommunication Sciences, 1996.
- [Jas02] Jasperneite, J.: Leistungsbewertung eines lokalen Netzwerkes mit Class-of-Service Unterstützung für die prozessnahe Echtzeitkommunikation. Dissertation, Magdeburg, 2002.
- [Jas05] Jasperneite, J.: Echtzeit-Ethernet im Überblick. In atp, 2005; S. 29–34.
- [JB13] Janssen, D.; Barth, R.: Steuerungstechnik im Sub-100- μ s-Bereich. In Industrial ethernet journal. - Marburg: Technik-Dokumentations-Verl., 2013; S. 6–9.
- [Jel06] Jelali, M.: An overview of control performance assessment technology and industrial applications. In IFAC International Conference on Intelligent Control Systems and Signal Processing, 2006, 14; S. 441–466.
- [JKB94] Johnson, N. L.; Kotz, S.; Balakrishnan, N.: Continuous univariate distributions. Wiley, New York, 1994.
- [Joh06] Johann, U.: Entfaltung von Hochfrequenzsignalen in der Ultraschall-Computertomographie. Diplomarbeit, Koblenz, 2006.
- [JSW07] Jasperneite, J.; Schumacher, M.; Weber, K.: Limits of increasing the performance of Industrial Ethernet protocols: ETFA 2007. IEEE 12th International Conference on Emerging Technologies and Factory Automation, Patras, Greece, 25 - 28 Sept. 2007, IEEE, Piscataway, NJ, USA, 2007; S. 17–24.
- [KDL13] Keller, B.; Dach, S.; Löber, P.: Konzept zur virtuellen Planung von RTE-Systemen mit integrierter Analyse des zeitlichen Sensor/Aktor-Verhaltens. In (Frey, G.; Schumacher, W.; Verl, A. Hrsg.): sps ipc drives 2013. Elektrische Automatisierung - Systeme und Komponenten, Internationale Fachmesse und Kongress, Nürnberg, 26. - 28.11.2013, VDE Verlag, Berlin, 2013.
- [Kop02] Kopetz, H.: Time-triggered real-time computing: Proceedings of the 15th IFAC World Congress 2002, Barcelona, Spain, 21.-26.07.2002, IFAC, Elsevier, 2002; S. 1638.
- [Kop91] Kopetz, H.: Event-Triggered Versus Time-Triggered Real-Time Systems. In (Karshmer, A.; Nehmer, J. Hrsg.): Proceedings of the International Workshop on Operating Systems of the 90s and Beyond, Springer, London, UK, UK, 1991; S. 87–101.

- [Kry12] Krystek, M.: Berechnung der Messunsicherheit. Grundlagen und Anleitung für die praktische Anwendung. Beuth Verlag, Berlin, 2012.
- [KS04] Koller, G.; Sauter, T.: Das Internet in der Automation — eine Herausforderung für die Regelungstechnik. In e&i Elektrotechnik und Informationstechnik, 2004, 121; S. 31–36.
- [KWK09] Kolesnikov, A.; Wolfinger, B. E.; Kulas, M.: UniLoG - ein System zur verteilten Lastgenerierung in Netzen. In (Halang, W. A.; Holleccek, P. Hrsg.): Software-intensive verteilte Echtzeitsysteme. Fachtagung des GI/GMA-Fachausschusses Echtzeitsysteme (real-time) Boppard, 19. und 20. November 2009, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2009; S. 11–20.
- [LC02] Lincoln, B.; Cervin, A.: Jitterbug: A Tool for Analysis of Real-Time Control Performance: CDC 2002. 41st IEEE Conference on Decision and Control, Las Vegas, USA, 10.-13.12.2002, IEEE, Piscataway, NJ, USA, 2002.
- [LF07] Liu, L.; Frey, G.: Simulation approach for evaluating response times in networked automation systems: ETFA 2007. IEEE 12th International Conference on Emerging Technologies and Factory Automation, Patras, Greece, 25 - 28 Sept. 2007, IEEE, Piscataway, NJ, USA, 2007.
- [Liu14] Liu, L.: Object-oriented modeling and efficient simulation of C3 systems. Objektorientierte Modellierung und effiziente Simulation von C3-Systemen. Shaker, Aachen, 2014.
- [LJA07] Leon, M.; Juanole, G.; Alessandria, E.: Performance analysis of Ethernet/IP networks. In (Leon, M. Hrsg.): Fieldbuses and Networks in Industrial and Embedded Systems 2007, Toulouse, France, Nov. 07, 2007, IFAC, Elsevier, 2007; S. 391–398.
- [LR94] Luck, R.; Ray, A.: Experimental verification of a delay compensation algorithm for integrated communication and control systems. In International Journal of Control, 1994, 59; S. 1357–1372.
- [LRD+13] Lüder, A.; Riedl, M.; Drath, R.; Heines, B.; Niggemann, O.: Austausch von Entwurfsdaten für Kommunikationssysteme mit Hilfe von AutomationML: Automation 2013. 14. Branchentreff der Mess- und Automatisierungstechnik, Baden-Baden, 25.-26.06.2013, VDI Verlag, Düsseldorf, 2013.
- [Lun14a] Lunze, J.: Regelungstechnik 2. Mehrgrößensysteme, digitale Regelung. Springer Berlin Heidelberg, Berlin, 2014a.
- [Lun14b] Lunze, J.: Regelungstechnik 1. Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen. Springer Berlin Heidelberg, Berlin, 2014b.
- [LW10] Lutz, H.; Wendt, W.: Taschenbuch der Regelungstechnik. Mit MATLAB und Simulink. Harri Deutsch, Frankfurt am Main, 2010.
- [LXR+07] Liu, G.-P.; Xia, Y.; Rees, D.; Hu, W.: Design and Stability Criteria of Networked Predictive Control Systems With Random Network Delay in the Feedback Channel. In IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2007, 37; S. 173–184.

-
- [MA02] Montestruque, L. A.; Antsaklis, P. J.: Model-Based Networked Control Systems: Stability, 2002.
- [MAC+08] Maiti, D.; Acharya, A.; Chakraborty, M.; Konar, A.; Janarthanan, R.: Tuning PID and PI λ D δ Controllers using the Integral Time Absolute Error Criterion: Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on, 2008; S. 457–462.
- [Mar07] Marsal, G.: Evaluation of time performances of ethernet-based automation systems by simulation of high-level petri nets. Shaker, Aachen, 2007.
- [MEA07] Münz, U.; Ebenbauer, C.; Allgöwer, F.: Stability of Networked Systems with Multiple Delays Using Linear Programming: ACC 2007. American Control Conference, New York City, USA, 11.-13. Juni 2007, 2007; S. 5515–5520.
- [MEH+09] Münz, U.; Ebenbauer, C.; Haag, T.; Allgöwer, F.: Stability Analysis of Time-Delay Systems With Incommensurate Delays Using Positive Polynomials. In Automatic Control, IEEE Transactions on, 2009, 54; S. 1019–1024.
- [MH04] Meffert, B.; Hochmuth, O.: Werkzeuge der Signalverarbeitung. Grundlagen, Anwendungsbeispiele, Übungsaufgaben. Pearson Studium, München, 2004.
- [Mis14] Miskowicz, M.: Event-based sampling strategies in networked control systems: WFCS 2014. 10th IEEE International Workshop on Factory Communication Systems, Toulouse, France, 05.-07. Mai 2014, IEEE, Piscataway, NJ, USA, 2014; S. 1–10.
- [NBW98] Nilsson, J.; Bernhardsson, B.; Wittenmark, B.: Stochastic analysis and control of real-time systems with random time delays. In Automatica, 1998, 34; S. 57–64.
- [Neu04] Neumann, P.: Communication in industrial automation—What is going on?: INCOM 2004. 11th IFAC Symposium on Information Control Problems in Manufacturing, Salvador-Bahia, Brazil, 5.-7.04.2004, 2004; S. 1332–1347.
- [Nil98] Nilsson, J.: Real-time control systems with delays, Lund, 1998 ;
- [NO08] Natori, K.; Ohnishi, K.: A Design Method of Communication Disturbance Observer for Time-Delay Compensation, Taking the Dynamic Property of Network Disturbance Into Account. In IEEE Transactions on Industrial Electronics, 2008, 55; S. 2152–2168.
- [Nus10] Nuzskowski, H.: Digitale Signalübertragung im Mobilfunk. Vogt, Dresden, 2010.
- [Obj15] Object Management Group, I.: UML. Unified Modeling Language. <http://www.omg.org/spec/UML/>, Geprüft am: 21.09.2015.
- [Ope15] OpenSim Ltd.: OMNeT++. Discrete Event Simulator. <http://omnetpp.org/>, Geprüft am: 10.05.2015.
- [Pad05] Padeffke, M.: Entwurfsverfahren für asynchrone Schaltungen unter Verwendung von Standardsoftware. Univ., Diss.--Erlangen-Nürnberg, 2005. Cuvillier, Göttingen, 2005.
- [Pat82] Patzak, G.: Systemtechnik -- Planung komplexer innovativer Systeme. Grundlagen, Methoden, Techniken. Springer Berlin Heidelberg, Berlin, Heidelberg, 1982.

- [Pes03] Pesch, B.: Bestimmung der Messunsicherheit nach GUM. Books on Demand, Norderstedt, 2003.
- [PH08] Pachner, D.; Havlena, V.: An Approach to Out-of-Sequence Measurements in Feedback Control Systems: WFCS 2008. 7th IEEE International Workshop on Factory Communication Systems, Dresden, Germany, 21.-23. Mai 2008, IEEE, Piscataway, NJ, USA, 2008.
- [PHB04] Priester, D.; Hetzel, P.; Bauch, A.: Zeit- und Normalfrequenzverbreitung mit DCF77. In PTB-Mitteilungen, 2004, 2004; S. 345–367.
- [Pos09] Postlethwaite, B.: Measures of controlled system performance. http://www.online-courses.vissim.us/Strathclyde/measures_of_controlled_system_pe.htm, Geprüft am: 19.05.2015.
- [PRO06] Profile Drive Technology PROFIdrive, PROFIBUS Nutzerorganisation, 2006.
- [Pry08] Prytz, G.: A performance analysis of EtherCAT and PROFINET IRT: ETFA 2008. IEEE 13th International Conference on Emerging Technologies and Factory Automation, Hamburg, Germany, 15.-18.09.2008, IEEE, Piscataway, NJ, USA, 2008; S. 408–415.
- [Rei08] Reissenweber, B.: Feldbussysteme zur industriellen Kommunikation. Oldenbourg Industrieverlag, München, 2008.
- [RGR+12] Robert, J.; Georges, J.-P.; Rondeau, E.; Divoux, T.: Minimum Cycle Time Analysis of Ethernet-Based Real-Time Protocols. In International Journal of Computers, Communications and Control, 2012.
- [RHG07a] Rauchhaupt, L.; Hintze, E.; Gnad, A.: Über die Bewertung der Zuverlässigkeit industrieller Funklösungen. Teil 1: Die Theoretischen Grundlagen. In atp, 2007a, 2007; S. 38–47.
- [RHG07b] Rauchhaupt, L.; Hintze, E.; Gnad, A.: Über die Bewertung der Zuverlässigkeit industrieller Funklösungen. Teil 2: Die praktische Umsetzung. In atp, 2007b, 2007; S. 50–57.
- [RHQ+05] Rupp, C.; Hahn, J.; Queins, S.; Jeckle, M.; Zengler, B.: UML 2 glasklar. Praxiswissen für die UML-Modellierung und -Zertifizierung. Hanser, München, 2005.
- [Riv12] Riverbed: SteelCentral for Performance Management and Control | Riverbed application and network performance management solutions | Riverbed. <http://www.riverbed.com/products/performance-management-control/opnet.html>, Geprüft am: 17.05.2015.
- [RLH+14] Riedl, M.; Lüder, A.; Heines, B.; Drath, R.: Kommunikation mit AutomationML beschreiben. In atp edition - Automatisierungstechnische Praxis, 2014, 56; S. 44.
- [RS15] Reichardt, J.; Schwarz, B.: VHDL-Synthese. Entwurf digitaler Schaltungen und Systeme. De Gruyter Oldenbourg, Berlin, 2015.
- [Run14] Runde, M.: Echtzeitfähige Protokollerweiterung zum Schutz Ethernet-basierter Automatisierungskomponenten. Dissertation, Magdeburg, 2014.
- [Sch05] Scholz, P.: Softwareentwicklung eingebetteter Systeme. Grundlagen, Modellierung, Qualitätssicherung. Springer, Berlin, 2005.

-
- [Sch06] Schnell, G.: Bussysteme in der Automatisierungs- und Prozesstechnik. Grundlagen, Systeme und Trends der industriellen Kommunikation. Vieweg + Teubner, Wiesbaden, 2006.
- [Sch11] Schlichthärle, D.: Digital filters. Basics and design. Springer, Berlin, 2011.
- [Sch12] Schroeder, W.: PROFINET-Integration mit Highspeed. In Etz, 2012.
- [Sch13] Schwager, J.: Real-Time-Ethernet in der Industrieautomation. Informationsportal für Echtzeit-Ethernet in der Industrieautomation. <http://www.real-time-ethernet.de/>, Geprüft am: 16.05.2015.
- [Sha98] Shannon, C. E.: Communication In The Presence Of Noise. In Proceedings of the IEEE, 1998, 86; S. 447–457.
- [SJ11] Schriegel, S.; Jasperneite, J.: Taktsynchrone Applikationen mit PROFINET IO und Ethernet AVB: Automation 2011. 12. Branchentreff der Mess- und Automatisierungstechnik, Baden-Baden, 28.-29.06.2011, VDI Verlag, Düsseldorf, 2011.
- [SMS+11] Schrom, H.; Michaels, T.; Stein, S.; Ernst, R.: SmallCAN - A Reliable, Low-Power and Low-Cost Distributed Embedded System for Energy Efficient Building Automation: Energy2011, 2011.
- [Spr15] Springer Gabler Verlag Hrsg.: Gabler Wirtschaftslexikon. Stichwort: Monte-Carlo-Methode. <http://wirtschaftslexikon.gabler.de/Archiv/137904/monte-carlo-methode-v5.html>, Geprüft am: 31.10.2015.
- [SS06] Scheler, F.; Schroeder-Preikschat, W.: Time-Triggered vs. Event-Triggered: A matter of configuration?. In (Dulz, W.; Schröder-Preikschat, W. Hrsg.): MMB Workshop Proceedings 2006. ITG FA 6.2 Workshop on Model-Based Testing, GI/ITG Workshop on Non-Functional Properties of Embedded Systems, 13th GI/ITG Conference Measuring, Modelling, and Evaluation of Computer and Communication Systems, VDE Verlag, Berlin, 2006; S. 1–6.
- [SS85] Shimokawa, Y.; Shiobara, Y.: Real-time Ethernet for industrial applications: IECON 1985. Annual Conference of the IEEE Industrial Electronics Society, San Francisco, USA, 18.-22.11.1985, 1985; S. 829–834.
- [SSF+07] Schreiber, S.; Schmidberger, T.; Fay, A.; May, J.; Drewes, J.; Schnieder, E.: UML-based safety analysis of distributed automation systems: ETFA 2007. IEEE 12th International Conference on Emerging Technologies and Factory Automation, Patras, Greece, 25 - 28 Sept. 2007, IEEE, Piscataway, NJ, USA, 2007; S. 1069–1075.
- [Ste83] Stegmüller, W.: Kausalitätsprobleme, Determinismus und Indeterminismus Ursachen und Inus-Bedingungen Probabilistische Theorie der Kausalität. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983.
- [Sto11] Stotz, D.: Computergestützte Audio- und Videotechnik. Multimediatechnik in der Anwendung. Springer, Berlin, Heidelberg, 2011.
- [Syr05] Syrakow, M.: Modellbildung Modellbildung zur Bewertung und Optimierung von eingebetteten Systemen. Vorlesung "Modellbildung", 2005.

- [TC03] Tipsuwan, Y.; Chow, M.-Y.: Control methodologies in networked control systems. In Special Section on Control Methods for Telecommunication, 2003, 11; S. 1099–1111.
- [Tec03] Technische Universität Bergakademie Freiberg - Institut für Werkstoffwissenschaft: Faltung - Entfaltung. [www.wv.tu-freiberg.de/mk/Dokumente/Sga_ii/Entfaltung\(3,5\).ppt](http://www.wv.tu-freiberg.de/mk/Dokumente/Sga_ii/Entfaltung(3,5).ppt), Geprüft am: 16.02.2015.
- [Tex14] Texas Instruments: Industrial Temp, Single Port 10/100Mbps Ethernet Physical Layer Transceiver. TLK105, TLK 106. <http://www.ti.com/lit/ds/symlink/tlk105.pdf>, Geprüft am: 17.09.2015.
- [The13] Theuerkauf, K.: Zeitverhalten in einer verteilten Steuerung. Bachelorarbeit, Magdeburg, 2013.
- [The15] The MathWorks, Inc.: MATLAB and Simulink for Technical Computing - MathWorks Deutschland. <http://de.mathworks.com/>, Geprüft am: 06.01.2015.
- [TM04] Tatikonda, S.; Mitter, S.: Control under communication constraints. In Automatic Control, IEEE Transactions on, 2004, 49; S. 1056–1068.
- [TM95] Tyler, M. L.; Moriari, M.: Performance assesment for unstable and nonminimum-phase systems: IFAC Workshop on on-line fault detection and supervision in the chemical process industries, Newcastle-upon-Tyne, UK, 1995.
- [Tre13] Tretter, A.: Industrial Requirements for a Converged Network. IEEE 802.3 SG DMLT. <http://www.ieee802.org/3/DMLT/public/jan13/Tretter-01-0113-Industrial%20Requirements-DMLT.pdf>, Geprüft am: 30.09.2015.
- [VDI07] VDI/VDE 2185 Blatt 1: Funkgestützte Kommunikation in der Automatisierungstechnik, VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 2007.
- [WA05] Wendemuth, A.; Andelić, E.: Grundlagen der digitalen Signalverarbeitung. Ein mathematischer Zugang. Springer, Berlin, 2005.
- [WB05] Wörn, H.; Brinkschulte, U.: Echtzeitsysteme. Grundlagen, Funktionsweisen, Anwendungen ; mit 32 Tabellen. Springer, Berlin, 2005.
- [WEK15] WEKA FACHMEDIEN GmbH Hrsg.: Lexikon – computer-automation.de. Begriff Deterministisch. <http://www.computer-automation.de/lexikon/?s=2&id=3057&page=0&search=deterministisch>, Geprüft am: 08.10.2015.
- [WG09] Werner, T.; Gerdelbracht, S.: Abschlussbericht zum Forschungsprojekt „Entwicklung einer Testmethodik zur Ermittlung der Anwendungsreaktionszeiten unter Verwendung unterschiedlicher Industrial-Ethernet-Lösungen“. Ermittlung der Anwendungsreaktionszeiten. AiF-Forschungs-Vorhaben Nr. 14592 BR, 2009.
- [Wik12] Wikipedia: Isochronität. <http://de.wikipedia.org/w/index.php?oldid=111846090>, Geprüft am: 22.04.2013.
- [WLF08] Wagner, F.; Liu, L.; Frey, G.: Simulation of Distributed Automation Systems in Modelica. In (Modelica Hrsg.): Proceedings of the 6th International Modelica Conference, Bielefeld, Germany, 2008; S. 113–122.

-
- [WSJ+12] Wisniewski, L.; Schumacher, M.; Jasperneite, J.; Schriegel, S.: Fast and simple scheduling algorithm for PROFINET IRT networks: WFCS 2012. 9th IEEE International Workshop on Factory Communication Systems, Lemgo/Detmold, Germany, 21.-24. Mai 2012, IEEE, Piscataway, NJ, USA, 2012; S. 141–144.
- [WYB02] Walsh, G. C.; Ye, H.; Bushnell, L. G.: Stability analysis of networked control systems. In IEEE Transactions on Control Systems Technology, 2002, 10; S. 438–446.
- [ZBP01] Zhang, W.; Branicky, M. S.; Phillips, S. M.: Stability of networked control systems. In IEEE Control Systems Magazine, 2001, 21; S. 84–99.

Anhang A Simulation des Zeitverhaltens am Zyklusübergang

In diesem Kapitel sollen die Herleitungen zum Zeitverhalten an Zyklusübergängen in Abschnitt 4.3 überprüft werden. Dafür werden numerische Simulationen mittels MATLAB Simulink durchgeführt.

Die Simulation ist dem Messprinzip für Klemme-Klemme-Reaktionszeiten, wie in Abschnitt 6.3.1 beschrieben, nachempfunden. Ein kontinuierliches Testsignal wird durch zyklische Elemente (Zero-Order-Hold-Blöcke) verzögert und die Zeitpunkte von Ein- und Ausgangssignal gespeichert. Daraus wird offline die Signallaufzeit bestimmt. Als Testsignal wird die Simulationszeit genutzt, diese ist kontinuierlich und streng monoton steigend.

Anhang A.1 Zusammenfassen synchroner Teilprozesse

Im ersten Schritt werden die Untersuchungen zum Zusammenfassen von synchronen Teilprozessen überprüft. Dazu wird das Simulink-Modell in Abbildung 154 verwendet.

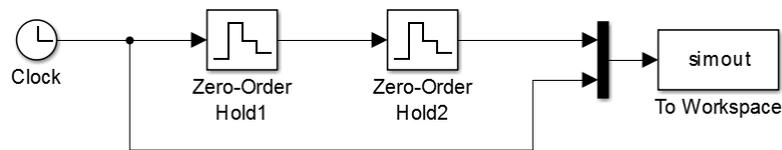


Abbildung 154: Simulink-Modell zur Untersuchung eines synchronen Zyklusübergangs

Die Blöcke werden mit den Zykluszeiten der beiden Teilprozesse parametrisiert, konkret:

- Zero-Order Hold1: Sample Time: $T_{\text{Cycle},1}$
- Zero-Order Hold2: Sample Time: $T_{\text{Cycle},2}$
- To Workspace: Sample Time: $T_{\text{Cycle},2}$

In der Simulation wird die Wartezeit zwischen den beiden Teilprozessen t_{wait} bestimmt. Zur Vereinfachung werden hier die Parameter t_{process} und φ als Null angenommen.

Anhang A.1.1 Szenario 1

Dieses Szenario entspricht der Betrachtung in Abschnitt 4.3.3.2.2.1 mit $T_{\text{Cycle},1} = 2 \text{ ms}$ und $T_{\text{Cycle},2} = 3 \text{ ms}$. Entsprechend der theoretischen Betrachtung wird eine diskrete Gleichverteilung $t_{\text{wait}} \in \{0; 1 \text{ ms}\}$ mit einer Häufigkeit von jeweils 50 % erwartet.

Das Ergebnis der Simulation (Abbildung 155) bestätigt die Erwartung.

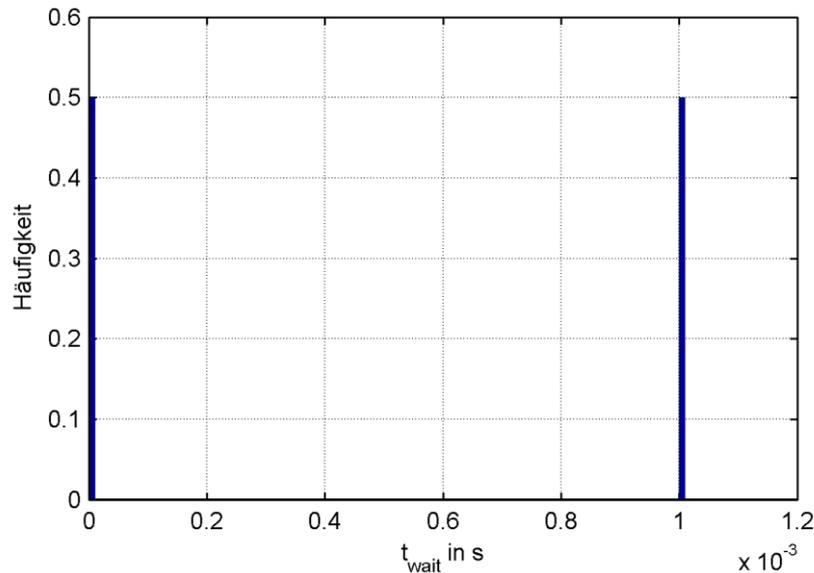


Abbildung 155: Simulierte Häufigkeitsverteilung der Wartezeit für Szenario 1

Anhang A.1.2 Szenario 2

Dieses Szenario entspricht der Betrachtung in Abschnitt 4.3.3.2.2.2 mit $T_{\text{Cycle},1} = 1,5 \text{ ms}$ und $T_{\text{Cycle},2} = 2,5 \text{ ms}$. Entsprechend der theoretischen Betrachtung wird eine diskrete Gleichverteilung $t_{\text{wait}} \in \{0; 0,5 \text{ ms}; 1 \text{ ms}\}$ mit einer Häufigkeit von jeweils 33 % erwartet.

Das Ergebnis der Simulation (Abbildung 156) bestätigt die Erwartung.

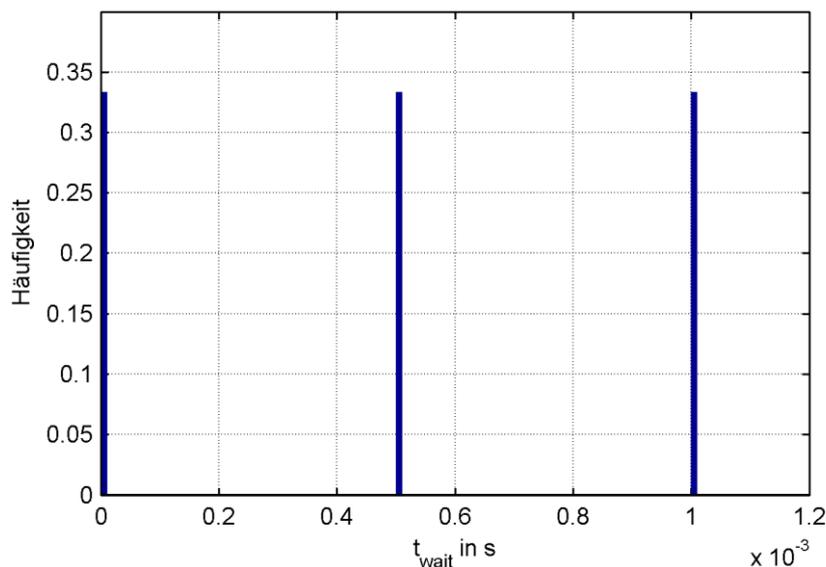


Abbildung 156: Simulierte Häufigkeitsverteilung der Wartezeit für Szenario 2

Anhang A.1.3 Szenario 3

Dieses Szenario entspricht Szenario 2 mit vertauschter Reihenfolgen der Teilprozesse 1 und 2. Da Over- und Undersampling bei synchronen Systemen keinen Einfluss auf das Zeitverhalten hat, wird ein Ergebnis identisch zu Szenario 2 erwartet.

Das Ergebnis der Simulation (Abbildung 157) bestätigt die Erwartung.

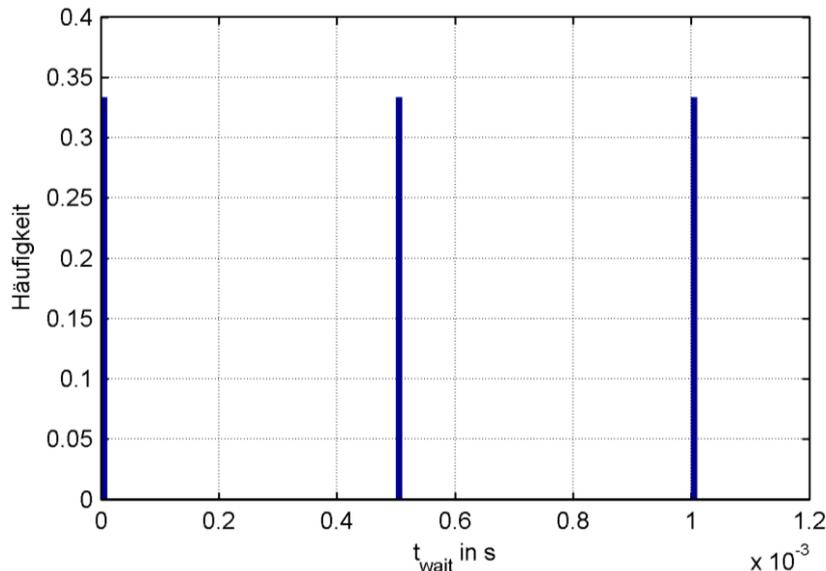


Abbildung 157: Simulierte Häufigkeitsverteilung der Wartezeit für Szenario 3

Anhang A.1.4 Szenario 4

In diesem Szenario wird für die Simulation $T_{\text{Cycle},1} = 1,1$ ms und $T_{\text{Cycle},2} = 7$ ms gewählt. Entsprechend der Betrachtung in Abschnitt 4.3.3.3 ergibt sich ein Basistakt $T_{\text{Base}} = 0,1$ ms und ein Teiler $c = 11$. Die theoretische Betrachtung ergibt eine diskrete Gleichverteilung mit $t_{\text{wait}} \in \{0; 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1\}$ ms und Häufigkeiten von jeweils $1/11 = 9,1\%$.

Die Simulation (Abbildung 158) bestätigt die Erwartung.

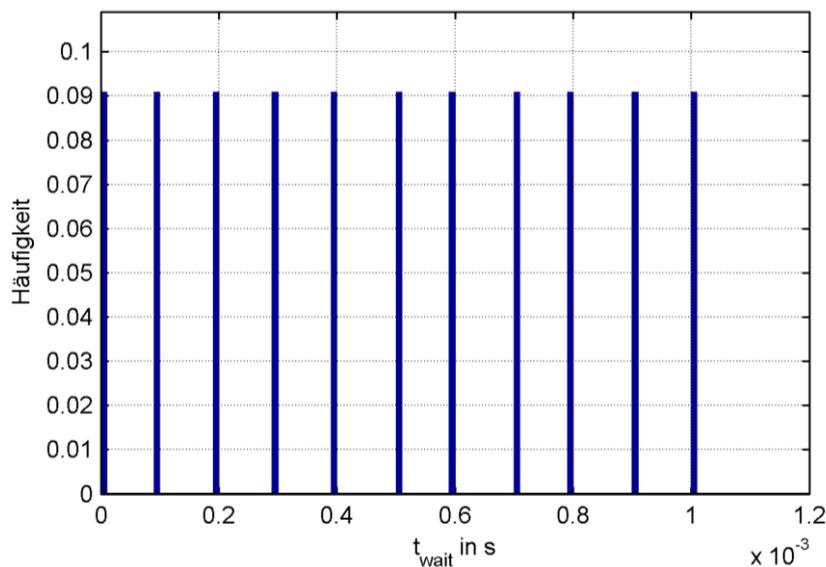


Abbildung 158: Simulierte Häufigkeitsverteilung der Wartezeit für Szenario 4

Anhang A.2 Simulation von einem asynchronen Zyklusübergang

Zur Überprüfung der Thesen zur Verzögerung durch Übergänge zwischen asynchronen Teilprozessen werden Simulationen durchgeführt. Das entsprechende Simulink-Modell ist in Abbildung 159 dargestellt. Der erste Zyklus in dieser Simulation stellt das Zustandsdiskrete Testsignal dar, d. h. hier wird die asynchrone Ankopplung an den technischen Prozess simuliert. Wie bei den Messungen gilt auch hier, dass die Zykluszeit des Testsignals größer sein muss als die größte zu erwartende Zykluszeit, da sonst keine eindeutige Zuordnung von Ein- und Ausgangssignalen möglich ist.

Um die Asynchronität zweier Teilprozesse in der Simulation abzubilden, wird eine Zykluszeit minimal verändert. Nachteil dieser Simulationen ist der hohe Rechenaufwand, da aufgrund der abweichenden Zykluszeiten mit sehr kleinen Schrittweiten simuliert wird.

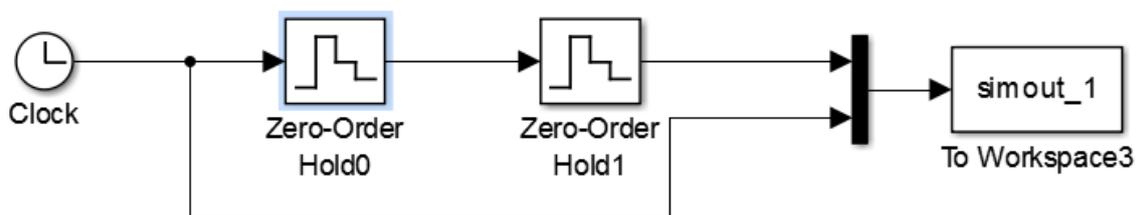


Abbildung 159: Simulink-Modell für einen asynchronen Zyklusübergang

Die Zykluszeit des Testsignals beträgt $T_{\text{Cycle},0} = 10 \text{ ms}$, das erste Teilsystem wurde mit $T_{\text{Cycle},1} \approx 1 \text{ ms}$ parametrisiert. Für die Simulation wurde die Samplettime von Zero-Order-Hold1 auf $1,00011 \text{ ms}$ gesetzt (um Asynchronität abzubilden). Die Simulationszeit beträgt 2000 s .

Entsprechend der theoretischen Betrachtungen wird als Ergebnis eine Gleichverteilung im Intervall $t_{\text{wait}} \in [0; 1,00011 \text{ ms})$ erwartet. Die Simulation liefert $t_{\text{wait}} \in [0; 1,0001 \text{ ms}]$, die Erwartung wird bestätigt.

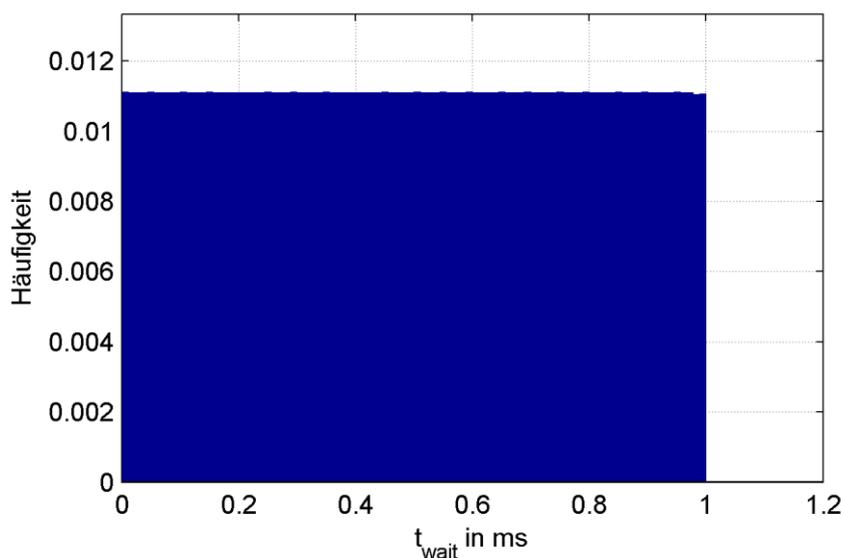


Abbildung 160: Simulierte Verzögerung durch einen asynchronen Zyklusübergang

Anhang A.3 Simulation von zwei asynchronen Zyklusübergängen

Zur Simulation von zwei asynchronen Zyklusübergängen wird das Modell um einen weiteren Teilprozess erweitert, siehe Abbildung 161.

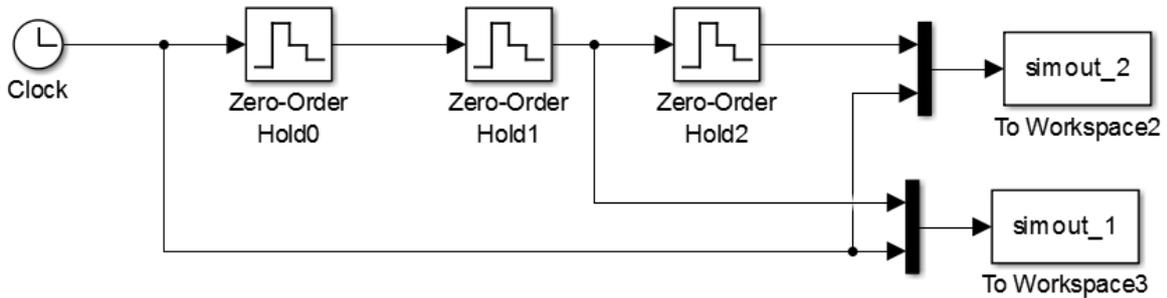


Abbildung 161: Simulink-Modell für zwei asynchrone Zyklusübergänge

Die Zykluszeit des Testsignals beträgt auch hier $T_{cycle,0} = 10$ ms, die Zykluszeiten der beiden anderen Teilprozesse werden in drei Szenarien variiert. Dabei werden die drei möglichen Varianten gleiche Zykluszeit, Oversampling und Undersampling abgebildet. Die Simulationslaufzeit beträgt 100000 s.

Anhang A.3.1 Szenario 1 – gleiche Zykluszeit

Parameter:

- $T_{\text{Cycle},1} = 1 \text{ ms}$
- $T_{\text{Cycle},2} = 1 \text{ ms}$

Erwartung:

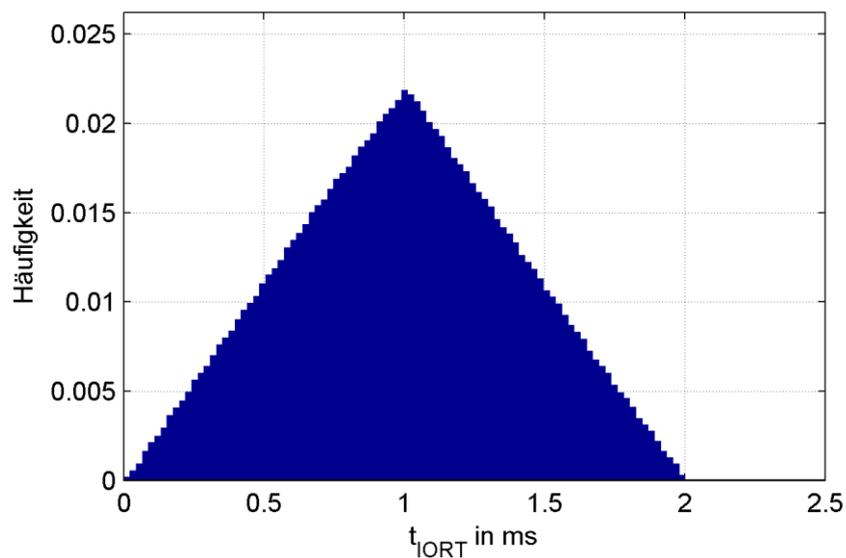
- Erster Zyklusübergang $t_{\text{wait},1} \in [0; 1 \text{ ms})$, gleichverteilt
- Zweiter Zyklusübergang $t_{\text{wait},2} \in [0; 1 \text{ ms})$, gleichverteilt
- Summe: $t_{\text{IORT}} \in [0; 2 \text{ ms})$, Dreiecksverteilung

Simulink-Modell:

- $T_{\text{cycle}_1} = 0,99999 \text{ ms}$
- $T_{\text{cycle}_2} = 1,00011 \text{ ms}$

Ergebnis:

- $t_{\text{IORT}} \in [0; 1,999 \text{ ms}]$
- Erwartung bestätigt

**Abbildung 162: Simulierte Verzögerung durch zwei asynchrone Zyklusübergänge – Szenario 1**

Anhang A.3.2 Szenario 2 – Undersampling

Parameter

- $T_{\text{Cycle},1} = 2 \text{ ms}$
- $T_{\text{Cycle},2} = 3 \text{ ms}$

Erwartung:

- Erster Zyklusübergang $t_{\text{wait},1} \in [0; 2 \text{ ms})$, gleichverteilt
- Zweiter Zyklusübergang $t_{\text{wait},2} \in [0; 3 \text{ ms})$, gleichverteilt
- Summe: $t_{\text{IORT}} \in [0; 5 \text{ ms})$, Trapezverteilung mit Knicken bei $\{2; 3\} \text{ ms}$

Simulink-Modell

- $T_{\text{cycle}_1} = 1,99999 \text{ ms}$
- $T_{\text{cycle}_2} = 3,00011 \text{ ms}$

Ergebnis:

- $t_{\text{IORT}} \in [0; 4,9997 \text{ ms}]$
- Erwartung bestätigt

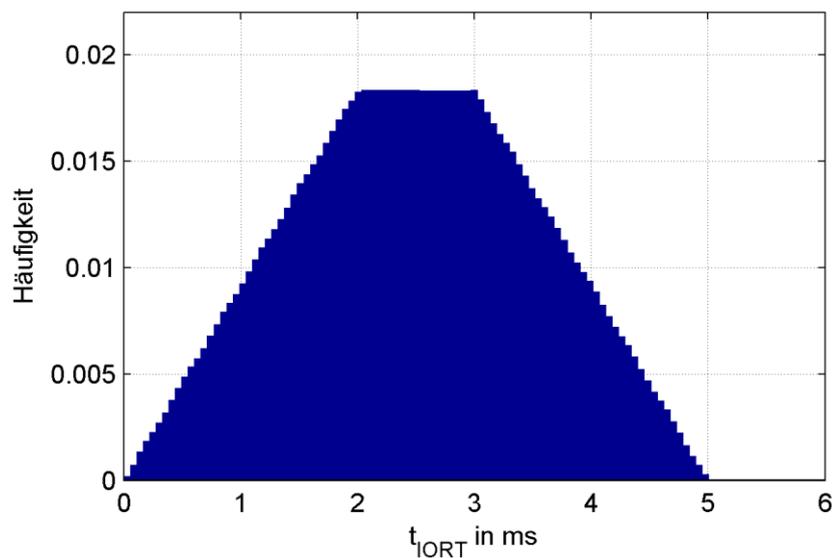


Abbildung 163: Simulierte Verzögerung durch zwei asynchrone Zyklusübergänge – Szenario 2

Anhang A.3.3 Szenario 3 - Oversampling

Parameter:

- $T_{\text{Cycle},1} = 7 \text{ ms}$
- $T_{\text{Cycle},2} = 1 \text{ ms}$

Erwartung:

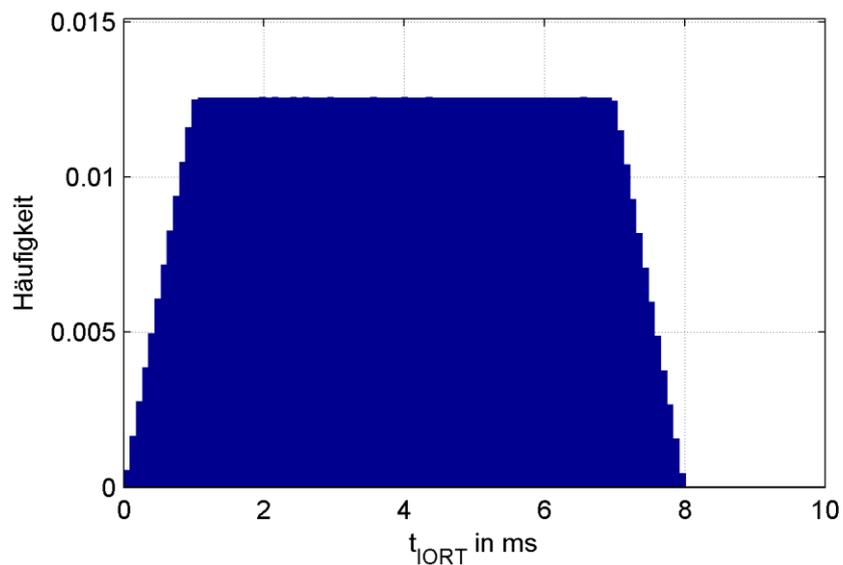
- Erster Zyklusübergang $t_{\text{wait},1} \in [0; 7 \text{ ms})$, gleichverteilt
- Zweiter Zyklusübergang $t_{\text{wait},2} \in [0; 1 \text{ ms})$, gleichverteilt
- Summe: $t_{\text{IORT}} \in [0; 8 \text{ ms})$, Trapezverteilung mit Knicken bei $\{1; 7\}$ ms

Simulink-Modell:

- $T_{\text{Cycle}_1} = 6,9998 \text{ ms}$
- $T_{\text{Cycle}_2} = 1,0011 \text{ ms}$

Ergebnis:

- $t_{\text{IORT}} \in [0; 7,9925 \text{ ms}]$
- Erwartung bestätigt

**Abbildung 164: Simulierte Verzögerung durch zwei asynchrone Zyklusübergänge**

Anhang A.4 Simulation von drei asynchronen Zyklusübergängen

Zur Simulation von drei asynchronen Zyklusübergängen wurde das Modell um einen weiteren Teilprozess erweitert, siehe Abbildung 165.

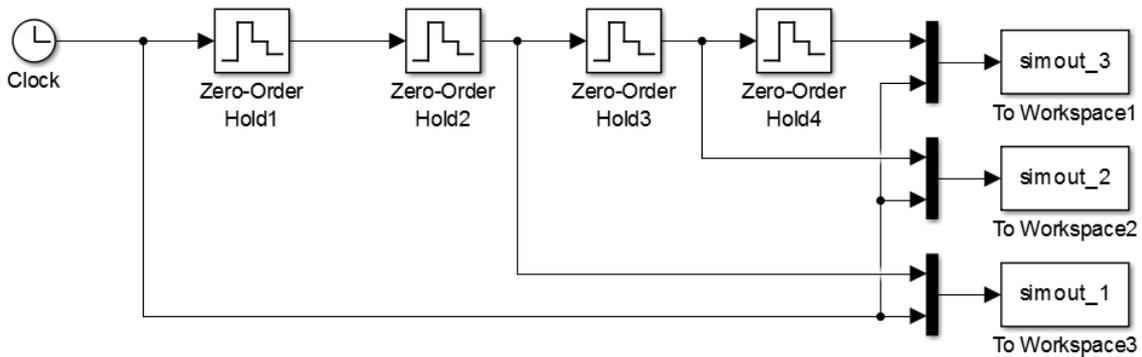


Abbildung 165: Simulink-Modell für drei asynchrone Zyklusübergänge

Die Zykluszeit des Testsignals beträgt auch hier $T_{\text{Cycle},0} = 10 \text{ ms}$, die Zykluszeiten der drei anderen Teilprozesse wurden in drei Szenarien variiert.

Die Szenarien 2 und 3 zeigen dabei nochmals, dass Over- und Undersampling keinen Einfluss auf das Zeitverhalten von asynchronen Zyklusübergängen haben.

Anhang A.4.1 Szenario 1

Parameter

- $T_{\text{Cycle},1} = 3 \text{ ms}$
- $T_{\text{Cycle},2} = 1 \text{ ms}$
- $T_{\text{Cycle},3} = 5 \text{ ms}$

Erwartung:

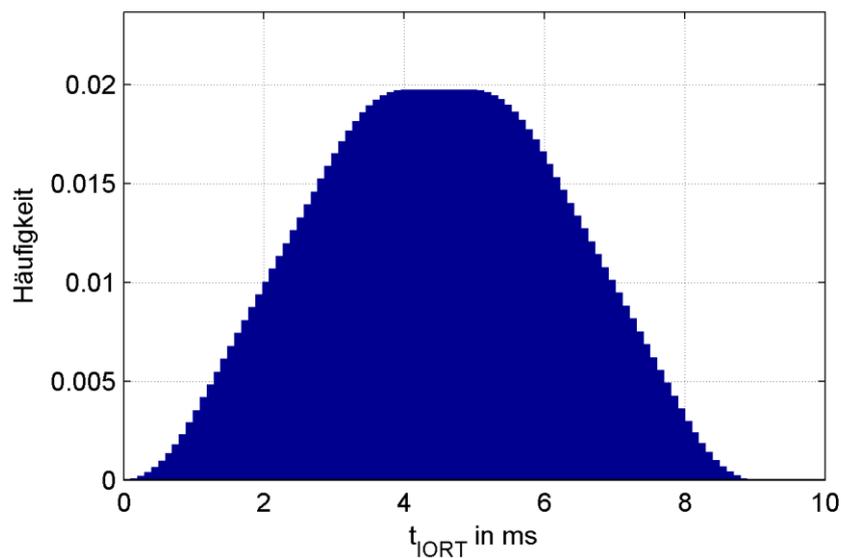
- Erster Zyklusübergang $t_{\text{wait},1} \in [0; 3 \text{ ms})$, gleichverteilt
- Zweiter Zyklusübergang $t_{\text{wait},2} \in [0; 1 \text{ ms})$, gleichverteilt
- Dritter Zyklusübergang $t_{\text{wait},3} \in [0; 5 \text{ ms})$, gleichverteilt
- Summe: $t_{\text{IORT}} \in [0; 9 \text{ ms})$

Simulink-Modell:

- $T_{\text{Cycle}_1} = 2,99912 \text{ ms}$
- $T_{\text{Cycle}_2} = 1,00001 \text{ ms}$
- $T_{\text{Cycle}_3} = 5,00123 \text{ ms}$

Ergebnis:

- $t_{\text{IORT}} \in [0; 8,9176 \text{ ms}]$
- Erwartung bestätigt

**Abbildung 166: Simulierte Verzögerung durch drei asynchrone Zyklusübergänge (Szenario 1)**

Anhang A.4.2 Szenario 2

Parameter:

- $T_{\text{Cycle},1} = 3 \text{ ms}$
- $T_{\text{Cycle},2} = 2 \text{ ms}$
- $T_{\text{Cycle},3} = 1 \text{ ms}$

Erwartung:

- Erster Zyklusübergang $t_{\text{wait},1} \in [0; 3 \text{ ms})$, gleichverteilt
- Zweiter Zyklusübergang $t_{\text{wait},2} \in [0; 2 \text{ ms})$, gleichverteilt
- Dritter Zyklusübergang $t_{\text{wait},3} \in [0; 1 \text{ ms})$, gleichverteilt
- Summe: $t_{\text{IORT}} \in [0; 6 \text{ ms})$

Simulink-Modell:

- $T_{\text{Cycle}_1} = 2,99912 \text{ ms}$
- $T_{\text{Cycle}_2} = 2,00001 \text{ ms}$
- $T_{\text{Cycle}_3} = 1,00123 \text{ ms}$

Ergebnis:

- $t_{\text{IORT}} \in [0; 5,9166 \text{ ms}]$
- Erwartung bestätigt

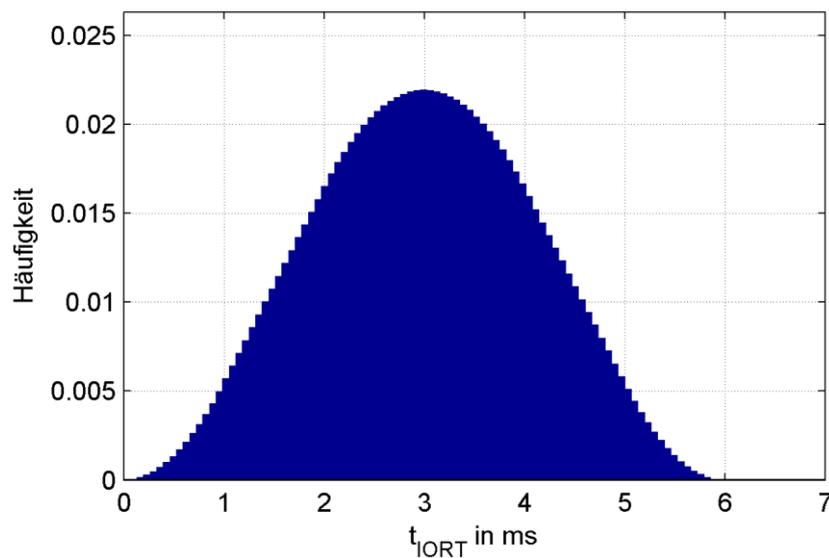


Abbildung 167: Simulierte Verzögerung durch drei asynchrone Zyklusübergänge (Szenario 2)

Anhang A.4.3 Szenario 3

Parameter:

- $T_{\text{Cycle},1} = 1 \text{ ms}$
- $T_{\text{Cycle},2} = 2 \text{ ms}$
- $T_{\text{Cycle},3} = 3 \text{ ms}$

Erwartung:

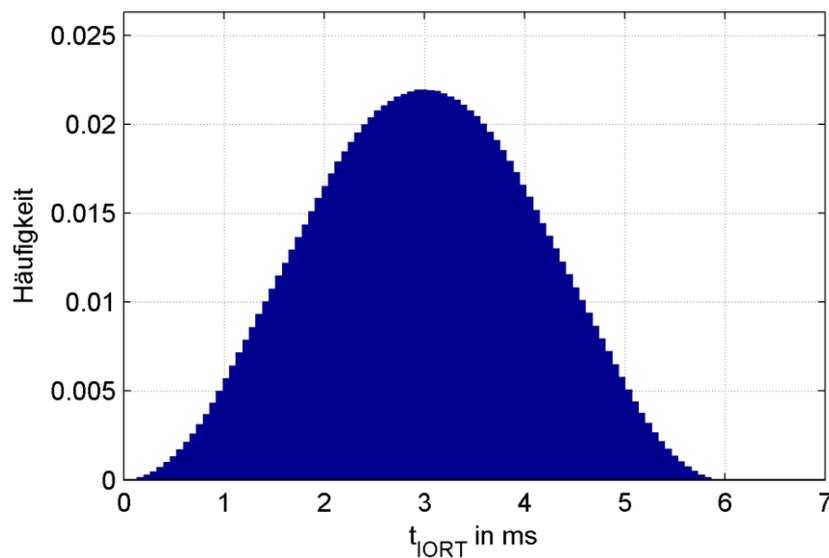
- Erster Zyklusübergang $t_{\text{wait},1} \in [0; 1 \text{ ms})$, gleichverteilt
- Zweiter Zyklusübergang $t_{\text{wait},2} \in [0; 2 \text{ ms})$, gleichverteilt
- Dritter Zyklusübergang $t_{\text{wait},3} \in [0; 3 \text{ ms})$, gleichverteilt
- Summe: $t_{\text{IORT}} \in [0; 6 \text{ ms})$

Simulink-Modell:

- $T_{\text{Cycle}_1} = 1,00123 \text{ ms}$
- $T_{\text{Cycle}_2} = 2,00001 \text{ ms}$
- $T_{\text{Cycle}_3} = 2,99912 \text{ ms}$

Ergebnis:

- $t_{\text{IORT}} \in [0; 5,9266 \text{ ms}]$
- Erwartung bestätigt

**Abbildung 168: Simulierte Verzögerung durch drei asynchrone Zyklusübergänge (Szenario 2)**

Anhang B Simulation des Einfluss von Messrauschen

In Abschnitt 6.3.1.1 wurde eine Erklärung für die Abweichung der mittels Rechteck-Testsignal gemessenen Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit eines isochronen Steuerungssystems geliefert (siehe Abbildung 169). Der Einfluss von Messrauschen und endlicher Anstiegszeit wird im Folgenden mittels Simulation verifiziert.

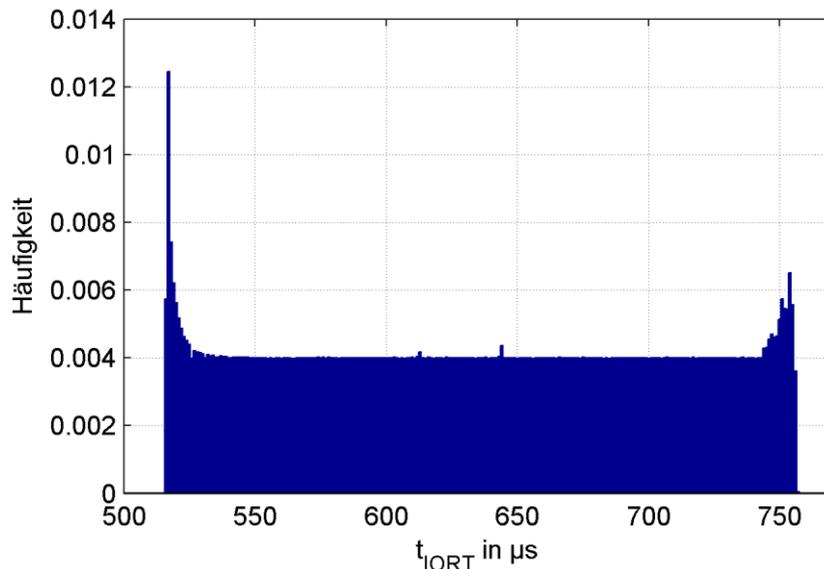


Abbildung 169: Häufigkeitsverteilung t_{IORT} für isochrones Steuerungssystem (diskretes Testsignal)

Zu diesem Zweck wurde ein Modell in Simulink erstellt, welches die beschriebenen Effekte enthält. Das isochrone Steuerungssystem wurde durch eine Abtastung ohne Verzögerung abstrahiert.

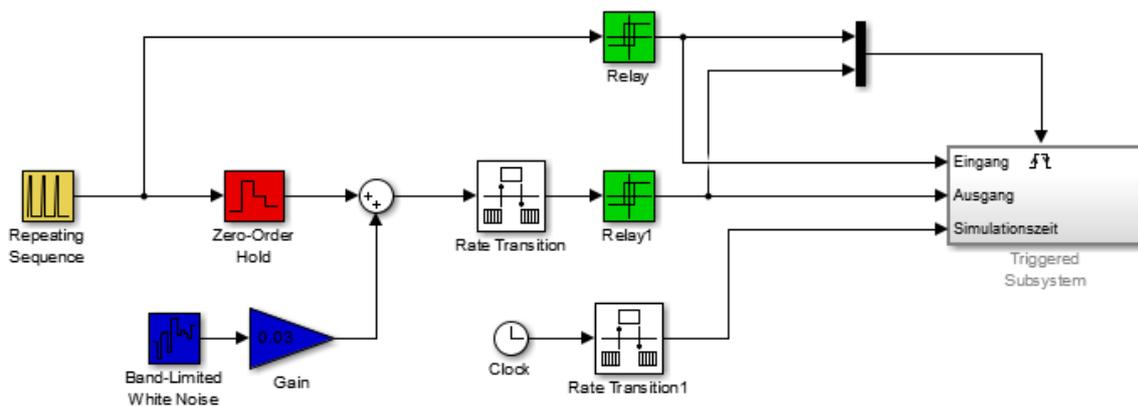


Abbildung 170: Simulink-Modell zur Simulation des Messaufbaus für t_{IORT}

Das System überträgt ein Testsignal (Trapezsignal als realer Verlauf eines Rechtecksignals) zyklisch (rot). Ein- und Ausgangssignal werden entsprechend binarisiert (grün). Die Auswertung erfolgt in einem Subsystem (grau), hier findet die Zeitstempelung von Ein- und Ausgangsflanken und die anschließende Ermittlung der Klemme-Klemme-Reaktionszeit sowie das Speichern der Daten statt. Zum Ausgangssignal wird ein Rauschen addiert (blau), es soll das Signal- und Messrauschen am Ausgang darstellen. Die Blöcke „Rate Transition“ sowie das Triggern des Subsystems dienen zur Reduktion des Simulationsaufwandes.

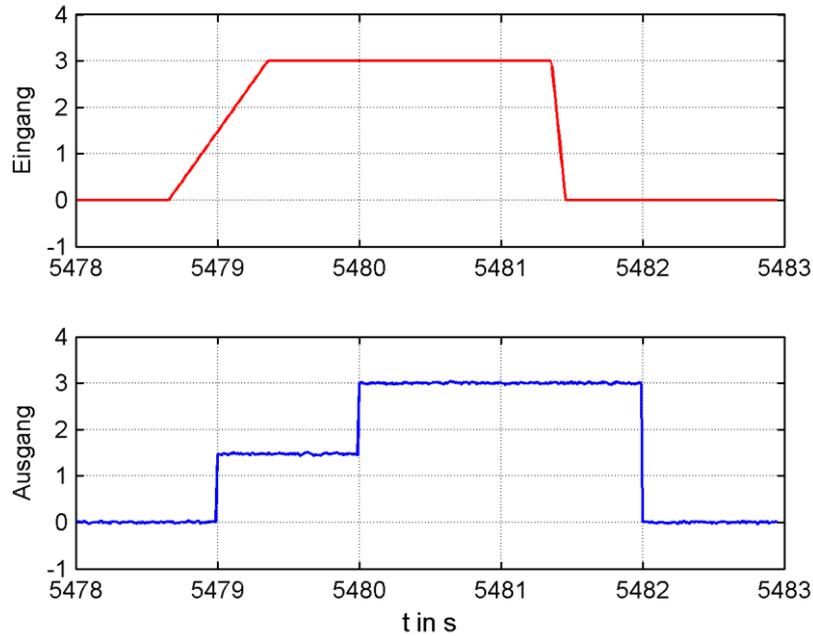


Abbildung 171: Simulierter zeitlicher Verlauf von Ein- und Ausgangssignal (Ausschnitt)

In Abbildung 171 ist der Signalverlauf am Ein- und Ausgang des simulierten Steuerungssystems dargestellt. Man erkennt die Bildung der Treppenstufe durch Abtastung während des Anstieges. Die Zykluszeit des Steuerungssystems wird auf 1 s und die Periodendauer des Testsignals auf 9,97 s gesetzt. Die Simulation wird für 10 Mio. Sekunden gerechnet, dies ergibt mehr als 1 Mio. Messpunkte für die Klemme-Klemme-Reaktionszeit. Die Simulation erfolgt mit einem Variable-Step-Solver, dessen maximale Schrittweite auf 0,01 s gesetzt wurde. Dies ist die maximale Genauigkeit der berechneten Zeiten t_{IORT} .

Der Rechenaufwand für diese zeitliche Simulation ist sehr hoch, trotz Optimierungen dauert dieser Durchlauf auf aktueller Hardware ca. 2 Stunden.

Die Häufigkeitsverteilung der Klemme-Klemme-Reaktionszeit ist in Abbildung 172 dargestellt. Man erkennt deutlich, dass das Signalrauschen am Ausgang die Überhöhung am linken Rand der Verteilung bedingt. Der Wert dieser Überhöhung ist abhängig von der Frequenz und Amplitude des Messrauschens sowie von der numerischen Schrittweite der Simulation.

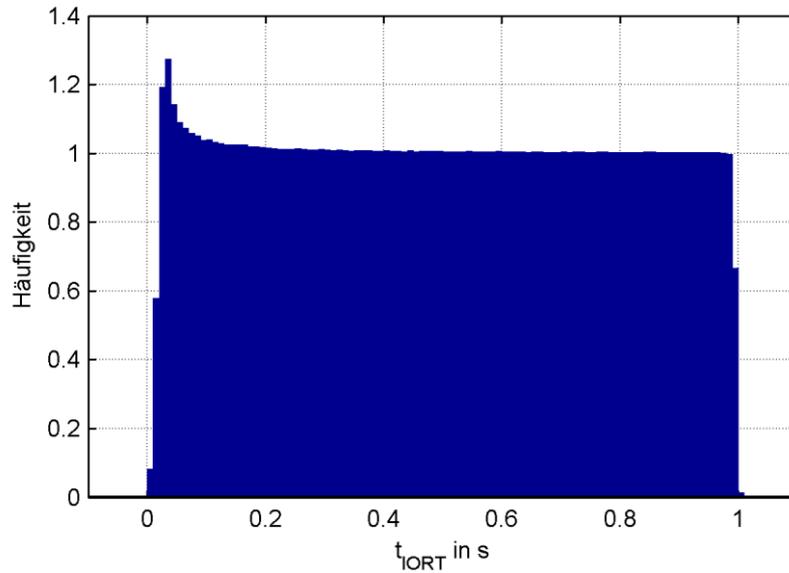


Abbildung 172: Simulierte Klemme-Klemme-Reaktionszeit für Rauschen am Ausgang

Die in Abbildung 172 nicht sichtbare Überhöhung am rechten Rand der Verteilung (vgl. Abbildung 169) wird durch das Messrauschen am Eingang bedingt. Eine simulative Untersuchung dieses Effektes ist nur mit deutlich geringeren Simulationsschrittweiten möglich. Aufgrund des steigenden Rechenaufwandes und damit der notwendigen Zeit für die Simulationen wurde darauf verzichtet.

Es konnte gezeigt werden, dass die Bildung der Überhöhung in den Häufigkeitsverteilungen von isochronen Steuerungssystemen nicht durch das System selbst, sondern durch den Messaufbau entsteht.

Anhang C Verwendete Hardware für Messung mit PROFINET

Für die Messungen in dieser Arbeit wurde folgende Hardware verwendet:

- Hilscher netANALYZER NANL-C500-RE
- Signalgenerator PeakTech 4050
- Oszilloskop HAMEG HMO 2024

Für die Beispielkonfigurationen in Kapitel 7 wurde folgende Hardware genutzt:

- Steuerung Siemens SIMATIC S7-300:
CPU 315-2 PN/DP (6ES7 315-2EH14-0AB0)
- Steuerung Siemens SIMATIC S7-300:
CPU 317-2 PN/DP (6ES7 317-3EK14-0AB0)
- Feldgeräte Siemens ET200S:
IM 151-3 PN HighSpeed (6ES7 151-3BA60-0AB0)
 - Analoges Eingangsmodul für ET200S:
2AI U HS (6ES7 134-4FB52-0AB0)
 - Analoges Ausgangsmodul für ET200S:
2AO U HS (6ES7 135-4FB52-0AB0)
 - verschiedene Module zur Vergrößerung der IO-Datenmenge
- Feldgeräte Siemens ET200S IM 151-3 PN HighFeature als Netzwerkteilnehmer zur Vergrößerung der Topologie

Anhang D Simulink-Modell des Portalkrans

Für die zeitliche Simulation des geschlossenen Regelkreises unter Berücksichtigung der Verzögerungen durch das Steuerungssystem wurde ein Simulink-Modell erstellt, siehe Abbildung 173.

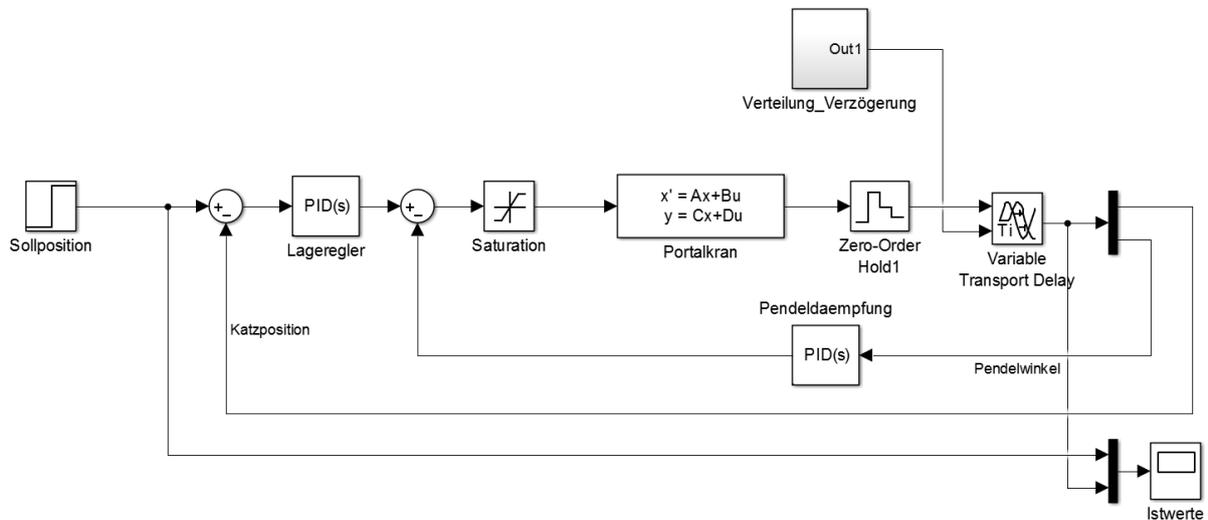


Abbildung 173: Simulink-Modell des geschlossenen Regelkreises des Portalkrans

Das Modell entspricht der Struktur in Abschnitt 8.3.1. Anstelle der Zustandsrückführung wurde der Regler aus zwei PD-Reglern aufgebaut. Funktional stellt dies keinen Unterschied dar. In Abbildung 173 werden PID-Blöcke von Simulink genutzt, der I-Anteil ist jedoch deaktiviert. Die Abtastung und Zeitverzögerung durch das Steuerungssystem wurde in Form eines Abtastgliedes (Zero-Order Hold1) und einer variablen Transportverzögerung (Variable Transport Delay) implementiert.

In Abbildung 174 ist die Struktur des Regelkreises im Simulink-Modell dargestellt. Es besteht aus der Strecke (grün), der Verzögerung durch das Steuerungssystem (orange) und dem Regler (blau). Es wird deutlich, dass es sich um einen Einschleifen-Regelkreis entsprechend Abbildung 32 handelt.

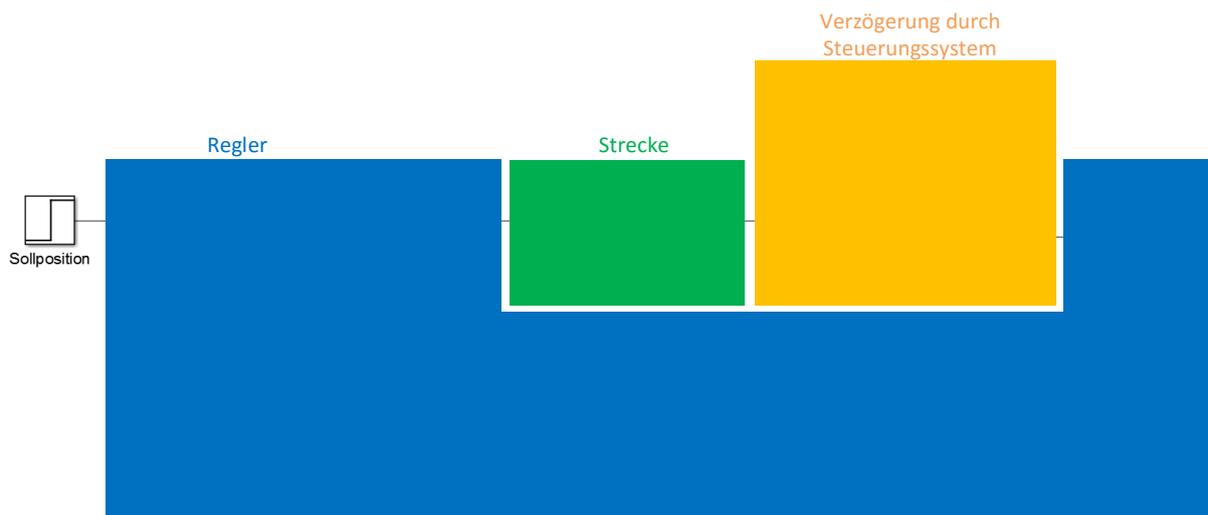


Abbildung 174: Struktur des Regelkreises im Simulink-Modell

Anhang E Häufigkeitsverteilungen der Regelgüte-Kriterien des Portalkrans

Die Regelgüte-Kriterien des Portalkrans (siehe Abschnitt 8.3.5) für die Szenarien B und C sind keine konstanten Werte, sondern folgen Häufigkeitsverteilungen.

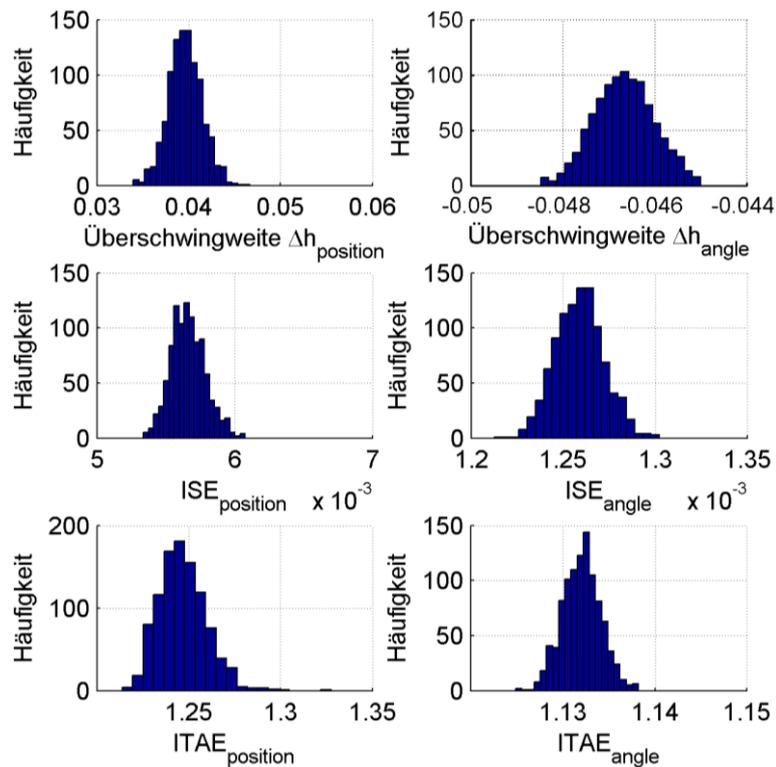


Abbildung 175: Verteilung der Regelgüte-Kriterien für Szenario B

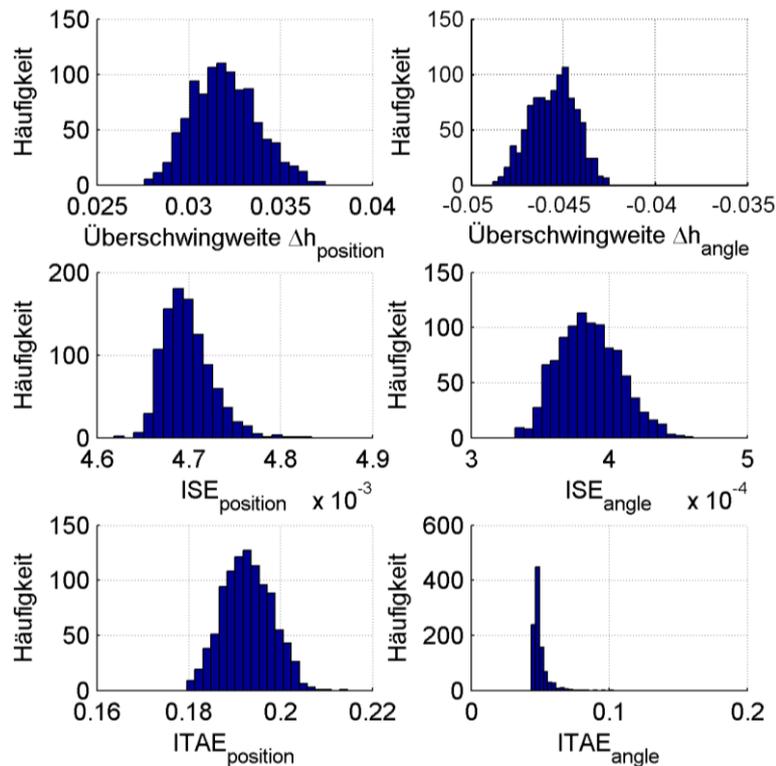


Abbildung 176: Verteilung der Regelgüte-Kriterien für Szenario C