



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Kai Müller

**Ein Peer-to-Peer-basierter Synchronisationsalgorithmus am
Beispiel von Ethernet**

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Kai Müller

**Ein Peer-to-Peer-basierter Synchronisationsalgorithmus am
Beispiel von Ethernet**

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Franz Korf
Zweitgutachter: Prof. Dr.-Ing. Andreas Meisel

Eingereicht am: 8. May 2018

Kai Müller

Thema der Arbeit

Ein Peer-to-Peer-basierter Synchronisationsalgorithmus am Beispiel von Ethernet

Stichworte

Synchronisation, Echtzeit Ethernet, TTEthernet, Automotive Anwendungen

Kurzzusammenfassung

Der Einsatz von echtzeitfähigen Netzwerken nimmt mehr und mehr zu. Unterschiedliche Industriezweige wie dem Automotive Bereich oder der Medizintechnik erwägen diese Technologie in vielen Bereiche einzusetzen, da die ursprünglichen Bussysteme bereits an ihre Grenzen stossen oder dies in naher Zukunft zu erwarten ist. Zudem können erweiterte Funktionen umgesetzt werden, für die die zur Zeit bereitgestellten Leistungen nicht ausreichen. Bisherige Synchronisationsalgorithmen verfolgen ein standardisiertes Master-Slave-Prinzip und weisen so in Hinblick auf Single Point of Failure und Unterstützung von Energiesparfunktionen erhöhten Planungsbedarf und Kostenaufwand auf. Ein neuer auf Peer-to-Peer basierender Ansatz könnte diese Probleme durch seine Architektur auffangen und dadurch Komplexität reduzieren und Wartungs- und Produktionskosten einsparen.

Kai Müller

Title of the paper

A peer-to-peer based synchronisation algorithm applied on Ethernet

Keywords

Synchronization, Real-time Ethernet, TTEthernet, Automotive Applications

Abstract

The amount of real-time capable networks increases fast. Because bus-systems used today come to their limit, several lines of industry, for example automotive or medical engineering, are planning to adopt this technology. Besides additional functions with heavier demands could be build in. Today's synchronisation algorithms uses standard master-slave-concepts, and therefore have high demands in planing with the use of cost excessive solutions for Single point of Failure strategies and energy saving. A new P2P-based architecture could solve these problems with the use of its concept and so, reduces the extra costs in maintenance and production.

Inhaltsverzeichnis

1	Einleitung	1
2	Motivation	4
2.1	Problemstellung und Zielsetzung	5
2.1.1	Synchronisation durch Peer-to-Peer	5
2.2	Aufbau der Arbeit	7
3	Hintergrund und verwandte Arbeiten	8
3.1	Using an IEEE 802.1AS Network as a Distributed IEEE 1588 Boundary, Ordinary, or Transparent Clock	10
3.1.1	Fehlererkennung und Kompensation	11
3.2	Skewless Network Clock Synchronization	12
4	Nachrichtenfluss und Informationsgewinnung	15
5	Ableitung des Synchronisationsmodells	19
5.1	Methoden der Kopplung und Abstraktion	20
5.1.1	Maschinelles Lernen	20
5.1.2	Schätzfunktionen	21
5.1.3	Regelungsvorgänge	21
5.2	Bildung des Modells als Regelungsvorgang	22
5.3	Gewichtungsfunktion	22
5.4	Klassifizierung des Systems	24
6	Beweisführung	26
6.1	Vollständige Modellbetrachtung eines Knotens	26
6.1.1	Systembetrachtung	27
6.1.2	Zustandsraummodell	28
6.2	Systembetrachtung des Netzwerkes	32
6.3	Stabilität des Gesamtsystems	36
7	Diskretisierung des Modells	41
7.1	Transformation: Eigenschaften und Grenzen	42
7.1.1	Strategie der Transformation	42
7.2	Systemdarstellung im Z-Bereich	43
7.3	Diskretisierung des Systems	44
7.3.1	Impulsinvariante Transformation	45

7.3.2	Diskretisierung im speziellen Fall	47
8	Anomalien der rollenlosen Synchronisation	51
8.1	Nachbarschaftsabweichung	51
8.2	Gruppenauflösung	53
8.3	Laufzeitabweichung	54
9	Realisierung eines Synchronisationssystems	55
9.1	Struktur eines Peers	55
9.1.1	Systembestandteile	56
9.2	Architektur und Umsetzung	58
9.3	Gleichungen und Übertragungsfunktionen	59
9.4	Testaufbau	60
9.5	Ergebnisse der Realisierung	62
10	Zusammenfassung	64
11	Ausblick	66

1 Einleitung

In zunehmend mehr Disziplinen erfolgt die Informationsverarbeitung in verteilten Netzwerkarchitekturen. Hierbei stehen nicht nur funktionale sondern auch mehr und mehr nicht-funktionale Anforderungen im Vordergrund. Während Mechanismen im Internet durch wiederholtes Versenden dafür sorgen, dass eine Antwort das Ziel erreicht so ist in anderen Bereichen nicht nur der Erhalt dieser Antwort, sondern auch die zum Erhalt benötigte Zeit immer wichtiger geworden. Verteilte Anwendungen müssen nach einer definierten Zeit antworten (z.B. industrieller Produktionsablauf) oder müssen ihre Aktionen unter einer Menge von Teilnehmern abstimmen (z.B. Fahrwerksregelung im Automobil). Während in Bereichen wie Flugzeug- und Automatisierungstechnik bereits Erfahrung mit entsprechenden Technologie gesammelt werden konnte (vgl. [Garner u. a. \(2009\)](#)), wird diese in Zukunft auf weitere Gebiete wie der Medizintechnik und den Bereich Automotive adaptiert (vgl. [Bruckmeier \(2010\)](#)). Da je nach Anwendungsfall das Einhalten der Anforderungen einen signifikanten Einfluss auf den Komfort, Sicherheit oder die Funktionsfähigkeit des Systems hat, muss das Einhalten dieser Kommunikationssystem-weiten Anforderungen garantiert werden können.

Hierzu werden zum Erstellungszeitpunkt dieser Arbeit noch größtenteils Bussysteme eingesetzt, welche sich durch ein gemeinsames Kommunikationsmedium auszeichnen. Der Zugriff auf das gemeinsame Medium erfolgt im gegenseitigen Ausschluss und kann zuverlässig und vorhersehbar über die Bussarbitrierung bestimmt und festgelegt werden. Somit ist das Verhalten und somit die Leistung eines Bussystems voll prognostizierbar und kann mit den Anforderungen einer verteilten Anwendung abgeglichen werden. Das Zeitverhalten einer Anwendung kann durch diese Kommunikationsinfrastruktur lückenlos vorhersagbar gemacht werden, was eine Aussage über das Einhalten gewünschter Garantien über das Zeitverhalten führt. Dieses Konzept verbietet jedoch parallele Kommunikation, wodurch die Übertragungsleistung der Netzwerke nicht mit zunehmender Größe skaliert. Zudem ist das Übertragungsmedium ein fester Bestandteil des Bussystems, womit die maximal mögliche Bandbreite begrenzt ist und nicht mit dem technologischen Fortschritt wachsen kann (vgl. Physical Layer von Ethernet). In Verbindung mit den stetig wachsenden Anforderungen von verteilten Anwendungen an die

zugrunde liegende Kommunikationsinfrastruktur führt dies dazu, dass in Zukunft mehr und mehr alternative Übertragungstechnologien eingesetzt werden.

Untersuchungen beschäftigen sich zur Zeit mit dem Einsatz von echtzeitfähigen Netzwerken in sicherheitskritischen Umgebungen (speziell etablierte Technologie wie Ethernet) als Alternative zu Bussystemen. Um die Vorhersagbarkeit zu gewährleisten werden spezielle echtzeitfähige Kommunikationsprotokolle verwendet, welche auf den jeweiligen Einsatzbereich hin optimiert sind. Sie zeichnen sich unter anderem durch ein vorhersagbares Zeitverhalten und einer uneingeschränkten Zuverlässigkeit in der Datenübertragung aus, welche unabhängig von der Belastung des Netzwerks garantiert werden kann. Hierzu werden je nach Anwendungszweck und den damit verbundenen Anforderungen an das Netzwerk unterschiedliche Konzepte, wie bandbreitenregulierende, tokenbasierte oder zeitscheibenbasierende (Time Division Multiple Access - TDMA) Verfahren eingesetzt. Bandbreiten- und tokenbasierte Verfahren zeichnen sich durch regulierende Konzepte des Nachrichtenflusses und Dynamik zur Laufzeit aus. Werden jedoch durch die Anwendung (z.B. Fahrwerksregelung) sehr hohe Zeitanforderungen an die Kommunikationsinfrastruktur gestellt, so ist dies nur mit Hilfe eines TDMA-fähigen Protokolls umzusetzen, welches ein hohes Maß an Präzision im Zeitverhalten des Nachrichtenflusses erreichen kann.

Ein TDMA-basiertes echtzeitfähiges Übertragungsprotokoll setzt durch seinen zeitlich festgelegten Ablauf eine globale Zeitbasis aller miteinander interagierenden Teilnehmer eines Netzwerks voraus. Zu diesem Zweck kommen spezielle Synchronisationsverfahren zum Einsatz, welche die globale Zeitbasis innerhalb eines Netzwerks bilden und aufrechterhalten. Die Zeitsynchronisation kann dabei entweder über eine Quelle unabhängig vom Netzwerk (z.B. GPS-Zeit oder einfaches TTL Signal) erfolgen oder als zusätzliches Protokoll, integriert in das zu synchronisierende Netzwerk, realisiert werden. Ein unabhängiger Synchronisationspfad bietet je nach Anwendungsfall mehr Ausfallsicherheit für das betroffene System, kann jedoch auch mehr Komplexität und zusätzliche Kosten mit sich bringen. Auch ist sicherzustellen, dass ein zusätzlicher Synchronisationspfad ebenso wie der eigentliche Kommunikationspfad robust genug ist, um durch Störung oder Ausfall die Echtzeitfähigkeit des Systems nicht zu gefährden. Es ist somit speziell für den jeweiligen Anwendungszweck abzuwägen, welche Eigenschaften am Ende ausschlaggebend sind. Aufgrund der hohen Stückzahl bei möglichst geringen Stückkosten kann z.B. für den Bereich automotive prognostiziert werden, dass Ansätze Anwendung finden werden, welche als Protokoll in das Kommunikationsnetz integriert sind. Diese Ansätze

sind zudem unabhängig vom zu synchronisierenden System und stellen somit den Fokus dieser Arbeit dar.

Teilen sich das Synchronisationsprotokoll und Kommunikationsprotokoll den Kommunikationspfad so muss folglich die Bandbreite zwischen den Protokollen aufgeteilt werden. Ein Synchronisationsprotokoll zeichnet sich daher durch eine möglichst geringe Zusatzbelastung im Kommunikationspfad, bei einer ausreichend hohen Präzision der Synchronisation, aus. Die Eigenschaften des Synchronisationsalgorithmus sind im besten Fall unabhängig von physikalischen oder topologischen Einflüssen wie z.B. der Länge der Übertragungsstrecke (Hop-Count) oder der Übertragungsdauer. Echtzeitprotokolle realisieren daher eigene Implementierungen (z.B. TTEthernet [AS6802 \(2011\)](#)), welche auf die Anforderungen des speziellen Protokolls hin optimiert sind oder nutzen einen allgemeinen Synchronisationsstandard wie das Precision Time Protocol ([IEEE 1588 \(2002\)](#)). Für das Design eines Synchronisationsprotokolls stellt sich somit der Herausforderung als Standard für eine Vielzahl von Echtzeitprotokollen anwendbar zu sein, und dabei auf Anforderungen wie Präzision und Bandbreitenbeschränkung eingehen zu können.

2 Motivation

Heutige Synchronisationsmechanismen, wie das Network Time Protocol (NTP) oder das Precision Time Protocol (PTP) werden als rollenbasierte Ansätze realisiert, welche in vordefinierten hierarchischen Strukturen konfiguriert sind. Das Netzwerk kann mit seinen Teilnehmern anschließend als zyklusfreier gerichteter Baum mit definierten Ebenen verstanden werden. Je nach verwendetem Synchronisationsprotokoll werden unterschiedlich viele Ebenen und somit Synchronisationsrollen für die Netzwerkteilnehmer unterstützt. Dies ermöglicht ein einfaches Erkennen von typischen Synchronisationsstörungen wie z.B. dem Ausfall einer Synchronisationsquelle („Dead Node“) oder nichtdeterministisches Senden von willkürlichen Zeitinformationen („Babbling Idiot“). Redundanz ist somit nicht nur über die Topologie, sondern auch über die Synchronisationsrolle vorhanden. Ist ein Fehler erkannt, so können Korrekturverfahren wie z.B. der „Best Master Clock Algorithm“ (BMCA) eingesetzt werden, welche anhand unterschiedlicher Faktoren aus einem vordefinierten Pool an Synchronisationsquellen, die für den nächsten Zyklus angemessenste Quelle selektieren (vgl. [IEEE 1588 \(2002\)](#)). Somit können innerhalb einer Ebene der hierarchischen Synchronisationsstruktur Ausfälle kompensiert werden. Jedoch stehen keine einfachen Mechanismen für ebenen- und somit rollenübergreifende Problemstellungen bereit, weshalb schon früh in der Planungsphase eines TDMA-basierten Echtzeitnetzwerks eine Abschätzung der zu kompensierenden Fehlerfälle und somit der Festlegung der Rollen aller Netzwerkteilnehmer erfolgen muss (vgl. [Weibel \(2005\)](#)).

Hierarchische Protokolle lassen sich zwischen statischen und dynamischen Modellen unterscheiden. Ist der Synchronisationsprozess durch einen Fehler gestört so können statische Modelle dies zwar erfassen, haben jedoch keine Möglichkeit eigene Synchronisationsstrukturen zur Kompensation des Fehlers zu bilden. Mögliche Problemfälle müssen zuvor erkannt und Gegenmassnahmen im Netzwerk eingeplant bzw. das Netzwerk darauf ausgelegt werden. Dynamische Modelle erkennen auftretende Fehler ebenfalls aber sind anschliessend auch in der Lage nicht nur zu reagieren, sondern auch vordefinierte Problemkategorien auszugleichen. Dies kann jedoch während eines Fehlerfalls zu einer vorübergehenden Unterbrechung der aktiven Synchronisation innerhalb des betroffenen Pfades des Netzwerks bis hin zum gesamten

Systems führen. Tritt innerhalb dieser Phase ein weiteres systemkritisches Problem auf, so kann dieses ggf. nicht mehr restlos vom System erkannt werden und führt im Worst-Case zu einer Gefährdung der angestrebten Servicequalität des verteilten Systems. Somit sind dynamische Ansätze nicht für den allgemeinen Fall anwendbar und ein Einsatz solcher Systeme ist für jeweiligen Einsatzzweck durch Abschätzung des Risikos abzuwägen.

2.1 Problemstellung und Zielsetzung

Für die Bildung von Systemen mit einer hohen Fehlerabsicherung werden zusätzliche Konzepte hinzugezogen. Redundante Knoten können Aufgaben ausgefallener Knoten übernehmen, Netzwerkpfade werden gespiegelt sowie je nach Sicherheitsgrad ganze Schattennetzwerke umgesetzt. Ist das System für die fehlerfreie Ausführung auf einem Synchronisationsalgorithmus angewiesen, so gelten die gleichen Anforderungen an die Redundanz. Dies resultiert in erhöhten Kosten und in einer komplexen Konfiguration. Diese Komplexität führt dazu, dass selbst bei vergleichsweise kleinen Netzen eine Konfiguration mit ökonomischen Aufwand nicht mehr ohne die Hilfe einer ausgereiften Toolchain möglich ist. Die Toolchain ermöglicht Abhängigkeiten zwischen Nachrichten und Ereignissen aufzulösen, sowie Konflikte im Zeitverhalten oder der Bandbreite aufzuzeigen. Es wäre wünschenswert über ein Synchronisationsprotokoll zu verfügen, welches ohne eine statische Rollenverteilung auskommt und somit keine Notwendigkeit einer hohen Vorlaufzeit in der Planung nach sich zieht. Zudem ließe sich das Einsatzgebiet weiter ausbauen, da mit den traditionellen Synchronisationsmechanismen ausschließlich vollständige und geschlossene Systeme in ihrer Vorhersagbarkeit abgeschätzt und betrachtet werden können.

Ein Synchronisationsprotokoll, welches nicht auf einer gerichteten Baumstruktur aufbaut könnte im Fehlerfall alleine durch seine Topologie Fehler kompensieren. Es hätte folglich keine feste Rollenverteilung, womit sich die Planung von Konzepten zur Ausfallsicherheit im Netzwerk allein auf Redundanz über die Topologie des Echtzeitnetzwerks beschränkt.

2.1.1 Synchronisation durch Peer-to-Peer

Ein Peer-to-peer-basierter Synchronisationsansatz würde ein Verständnis der globalen Zeitbasis eines verteilten Systems in erster Näherung allein über die Vermaschung der Kommunikationswege innerhalb des Netzwerks erreichen. In solch einem Netzwerk wäre jeder Teilnehmer des Systems gleichberechtigt und würde versuchen mit seinen Nachbarn zu einer Einigung

über das Verständnis der globalen Zeit zu kommen. Dieses Einverständnis würde sich somit durch das Netzwerk propagieren und für die globale Synchronisation sorgen. Je stärker die Kopplung zwischen den Teilnehmern, desto schneller kann eine Verbreitung der Aushandlung erfolgen. Da jedoch auch fehlerhafte Teilnehmer die gleiche Kopplungsstärke erfahren, würde eine Störung durch fehlerhafte Informationen allein durch das Verhältnis von ungestörten und gestörten Kopplungen kompensiert werden können. Für eine topologiespezifische Fehlerkompensation können an dieser Stelle weitere Mechanismen, wie heterogene Kopplungsstärken und Plausibilitätsprüfungen zum Einsatz kommen.

Das Konzept der verteilten Synchronisation als Peer-to-Peer-Ansatz kann in erster Näherung als das klassische physikalische Problem der lose-gekoppelten Federpendel beschrieben werden. Hierzu wird jeder Teilnehmer eines Netzwerkes als Pendel, sowie die Topologie des Netzes über die Kopplungen der Pendel untereinander dargestellt. Das Modell des gekoppelten Federpendels ist in seiner Grundform jedoch nicht ausreichend, da es sich bei diesen Betrachtungen um instabile, oder grenzstabile Systeme handelt, welche keine Konvergenz von Amplituden (Erregungen) aufweisen. Daher wird das Element der Dämpfung einer jeden Übertragungsstrecke hinzugefügt.

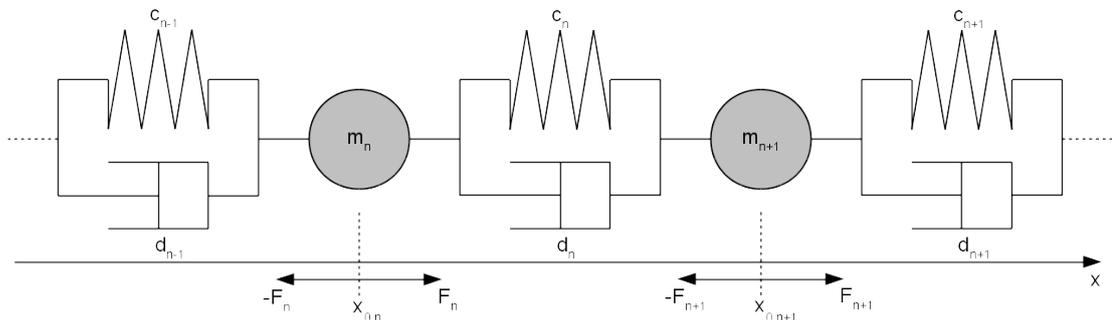


Abbildung 2.1: Feder-Masse-System

Aus Aufbau und Eigenschaften der klassischen Mechanik des Pendels lassen sich Analogien zur Netzwerkkommunikation herleiten. Jedes Pendel eines Systems entspricht hierbei dem lokalen Zeitverständnis des entsprechenden Knotens innerhalb des Netzwerkes. Die Abbildung 2.1 zeigt Federpendel am Beispiel einer Daisy-Chain-Topologie. Die Ruhelage x_0 beschreibt die Geschwindigkeit der lokalen unsynchronisierten Uhr. Synchronisationseinflüsse der Nachbarknoten werden durch die entsprechenden Kräfte F abgebildet und bewirken eine Auslenkung der Masse m bzw. eine Geschwindigkeitsänderung des lokalen Zeitgebers. Die Geschwindigkeit variiert somit über die Zeit. Der maximale Bereich der Auslenkung ist durch

die Korrekturfähigkeit des lokalen Zeitgebers definiert und wird nur durch die Plattform und die Umsetzung des Synchronisationskonzeptes beschränkt. Daher wird diese Beschränkung im weiteren Verlauf dieser Arbeit nicht betrachtet.

Die Rückstellkräfte F des Pendels summieren sich aus dem Grad der Abweichung bezogen auf den Synchronisationsruhepunkt und der Ungenauigkeit des simulierten Knotens und sind durch die Parameter c und d des Systems bestimmt. Die Synchronisationskräfte treiben das Pendel somit zurück in die dynamische Ruheposition, während die Ungenauigkeiten dem entgegen wirken. Die für diesen Vorgang nötige Zeit lässt sich durch den Masseparameter m bestimmen, welcher eine entsprechende Trägheit auf den jeweiligen Knoten simuliert und somit eine zusätzliche Priorisierung innerhalb des synchronisierenden Systems erlaubt.

2.2 Aufbau der Arbeit

Diese Arbeit gliedert sich wie folgt: Im Kapitel 3 wird ein Überblick über die verwandten Arbeiten gegeben. Des weiteren wird die Position des neuen Ansatzes im Vergleich zu bestehenden Protokollen wie dem Precision Time Protocol und dem Network Time Protocol aufgezeigt. Ein Überblick in die nötigen Kommunikationsstrukturen zur erfolgreichen Umsetzung einer Synchronisation wird in Kapitel 4 gegeben. Auf Basis dieser Kommunikationsstrukturen kann anschliessend in Kapitel 5 die Ableitung des Synchronisationsmodells erfolgen, welches in Kapitel 6 unter Zuhilfenahme der Konvergenzgesetze aus dem Bereich der Regelungstechnik auf Stabilität geprüft wird. Dies mündet in Kapitel 7 in der Abschliessenden Systembetrachtung als diskretes Modell. Es stellt sich heraus, dass zum fehlerfreien Betrieb eines rollenlosen Synchronisationsprotokolls weitere Parametrisierung erforderlich sein wird, was in Kapitel 8 dargelegt wird. Die Evaluierung als prototypische Umsetzung findet in Kapitel 9 statt. Es werden Testszenarien diskutiert sowie die Ergebnisse dargestellt. Abschließend wird im Kapitel 10 ein Überblick über die Kernpunkte dieser Arbeit gegeben, was in einer Abschlussdiskussion in Kapitel 11 seinen Abschluss findet.

3 Hintergrund und verwandte Arbeiten

Die Synchronisation von Verteilten Kommunikationssystemen erfolgt über ein Synchronisationsprotokoll, welches den Transport der Zeitinformationen und das Verhalten der Netzwerkteilnehmer beschreibt. Betrachtet man hier das Segment Ethernet, so sind heute die Protokolle für paketbasierte Kommunikationsnetze Network Time Protocol (NTP) und das Precision Time Protocol (PTP) weit verbreitet. (hierbei ist zu beachten, dass ihre Funktionsweise durch deren Standards keineswegs auf Ethernet beschränkt sind.) NTP ist ein Standard zur Synchronisierung von Uhren in Computersystemen mit Hilfe des Verbindungslosen Transportprotokolls UDP (User Datagram Protocol). Es wurde speziell entwickelt, um eine zuverlässige Zeitangabe über Netzwerke mit variabler Paketlaufzeit zu ermöglichen. Auch PTP ist ein Netzwerkprotokoll, das die Synchronität der Uhrzeiteinstellungen mehrerer Geräte in einem Computernetzwerk bewirkt. Anders als bei dem Network Time Protocol liegt der Fokus von PTP auf höherer Genauigkeit und lokal begrenzten Netzwerken. PTP kann in Hardware-Ausführung eine Genauigkeit im Bereich von Nanosekunden und in Software-Ausführung im Bereich weniger Mikrosekunden erzielen. PTP ist definiert in der IEEE 1588 und in IEC 61588 übernommen worden. Im Folgenden werden die Kernpunkte der Protokolle kurz erläutert und gegenübergestellt.

Das Network Time Protocol findet seit 1992 seinen Einsatz in vielen Anwendungsbereichen und ist das weit verbreitetste Protokoll zur Synchronisation über das Internet. Version 3 stellt den aktuellen Standard dar, welcher sich zurzeit weiterhin in der Entwicklung befindet und weitere Unterstützungen einpflegt. Das Protokoll definiert die Rolle eines *trusted Timeservers*, sowie beliebig vielen Clients. Mit diesem Ansatz können innerhalb von Netzwerken mit unterschiedlichen Übertragungslaufzeiten, Genauigkeiten im zweistelligen Millisekundenbereich erreicht werden. Hierbei ist zu beachten, dass die Architektur des Protokolls erfordert, Genauigkeiten zwischen einzelnen NTP-Knoten zueinander, sowie die Abweichung der NTP-Knoten zur gemeinsamen Zeitquelle, getrennt zu betrachten. Die Anzahl der Etappen in der Übertragung (Hops) stellt bei angemessener Topologie nur ein zu vernachlässigendes Faktor da, hier das Konzept von Stratis (vgl. [Abbildung 3.1 auf der nächsten Seite](#)) eingesetzt wird. Ein Stratum ist eine Synchronisationsebene und stellt in sich eine Gruppe von Netzwerkteilnehmern

dar, welche die gleiche oder ähnliche Beziehung im Synchronisationspfad aufweisen. Hierbei stellen Geräte eines Stratum die Synchronisationsgeber (Master) einer zu synchronisierenden Teilnehmergruppe (Client) da. Dieses Konzept lässt sich beliebig staffeln, jedoch sinkt die bestmöglich erreichbare Synchronisationsgenauigkeit mit jedem Stratum. Daher wird üblicher Weise die Genauigkeiten der lokalen Zeitgeber jeder Ebene gleichverhaltend über die Topologie abgebildet, womit die Güte der lokalen Zeitgeber mit absteigender Hierarchieebene sinkt.

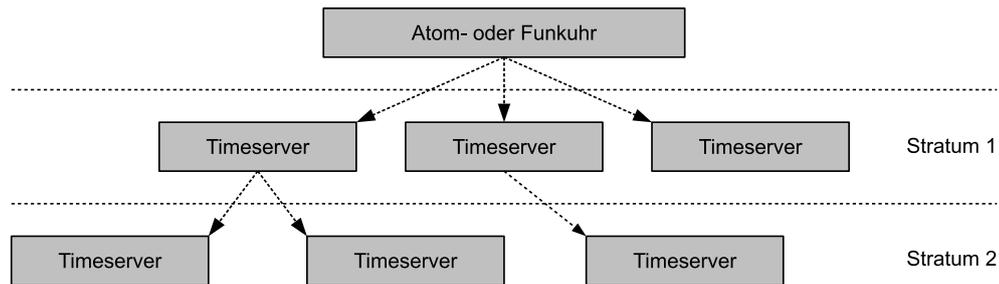


Abbildung 3.1: Hierarchische Synchronisationstopologie am Beispiel von NTP

Das Precision-Time-Protocol bildet den Standard für Synchronisationen, welche einen hohen Grad an Genauigkeit aufweisen müssen. Es wird zur Synchronisation in verteilten Messinstallationen, sowie der verteilten Regelungs- und Steuerungstechnik eingesetzt. Es ist ebenso wie NTP ein Protokoll, welches eine globale Zeitbasis innerhalb des Netzwerkes herstellt. Dieses muss ebenso paketbasiert sein und Mechanismen zur Kommunikation über Multicast verfügen. Der Standard wurde speziell für den Einsatz in Testumgebungen und der Produktions- und Automatisierungstechnik, welche mit Genauigkeitsanforderungen unterhalb einer Mikrosekunde spezifiziert sind, ausgelegt. Auch hier wird der hierarchische Ansatz verfolgt. Es existieren ein oder mehrere Synchronisationsmaster (Grand Master Clock), welcher einer Gruppe von Clients die Repräsentation einer Globalzeit vorgibt und somit die obere Schicht des Synchronisationsstruktur darstellt. Weitere Master (Boundary Clock) verbreiten die Synchronisationsinformationen unter Berücksichtigung vom Zeitverhalten der Übertragungsstrecke. Um Ausfallsicherheit zu gewährleisten können Master definiert werden, welche über einen *Best-Master-Clock-Algorithmus* den besten Zeitgeber bestimmen, so dass zu jedem Zeitpunkt eine Synchronisationsquelle als Referenz im Netzwerk vorhanden ist. Im Gegensatz zu NTP werden Anforderungen an die Übertragungsstrecke gestellt, was den Erhalt der hohen Genauigkeitsklasse fördert.

Der Begriff einer Peer-to-Peer-Synchronisation tritt erstmals in der Überarbeitung des Standards IEEE 1588 in der Version 2.0 auf. Hierbei wird die Synchronisation zweier Teilnehmer

verstanden, welche ohne eine Klassifikation von Grand Master Clocks, Boundary Clocks und Clients auskommt. Der Ansatz ist jedoch nicht rollenlos, da sehrwohl eine eindeutige Richtung der Synchronisation zwischen den Peers gegeben sein muss, womit es wiederum einen eindeutigen Zeitgeber und einem Synchronisationsempfänger gibt. Der Synchronisationsempfänger kann wiederum als Zeitgeber für einen oder mehrere Empfänger sein. Eine solche Topologie weist eine gewisse Dynamik auf, da zu beliebiger Zeit zusätzliche Synchronisationsempfänger oder ganze Zweige dieser Topologie zur Laufzeit hinzugefügt oder entfernt werden können. Dardurch kann jedoch kein Best-Master-Clock-Algorithmus mehr angewendet werden, da kein wohldefinierter Pool an potentiellen Zeitgebern existiert. Die Architektur der Synchronisation ist jedoch zu jeder Zeit so zu bilden, dass keine Rückkopplungen oder Schleifen im Synchronisationspfad entstehen, da dies die Stabilität gefährden kann. Es ist jedoch nicht näher spezifiziert, welche Randbedingungen zu einem endgültig instabilen System führen und welche z.B. nur mit einer erhöhten Ungenauigkeit je nach Anwendungsfall akzeptiert werden könnten.

3.1 Using an IEEE 802.1AS Network as a Distributed IEEE 1588 Boundary, Ordinary, or Transparent Clock

Diese Arbeit untersucht die Möglichkeit ganze Netzwerke als ein Teilnehmer innerhalb eines Synchronisationsmechanismus anzusetzen. Dies kann als erster Schritt in Richtung Peer-to-Peer Architektur angesehen werden. Es ermöglicht die Kopplung von nicht-timetriggered Netzen (wie z.B. den Audio Video Bridging Standart) mit Netzwerken wie TTEthernet oder dem IEEE 1588 Standart. Zudem werden in der Arbeit potentielle Fehlerquellen klassifiziert, welche zu Störungen oder Abweichungen in der Synchronität führen können, sowie Lösungen präsentiert, welche von zukünftigen Algorithmen aufgegriffen werden können, um die eigene Stabilität zu gewährleisten.

Die Basis bildet ein Netzwerk, welches durch ein Precision Time Protocoll synchronisiert wird. Einzelne Knoten aus diesem Netz werden durch ein eigenständiges Netzwerk substituiert und das Verhalten des PTP Prozesses untersucht, sowie seine Metriken aufgezeichnet. Bei einem Knoten kann es hierbei um den Grandmaster Clock, einer Boundary Clock oder auch einer Ordinary Clock handeln. Das substituierende Netz bedarf keiner eigenen Synchronisierung, muss jedoch topologisch über die Möglichkeit eines gemeinsamen Zeitgebers verfügen. Wird eine Ordinary Clock substituiert, so wird über den Synchronisationsmechanismus die gemeinsame Zeitbasis des nicht-timetriggered fähigen Netzes angepasst. Der Netzwerkverkehr

kann somit netzübergreifend propagiert werden. Es konnte jedoch beobachtet werden, dass die Ungenauigkeit des clients linear proportional von der Ungenauigkeit des substituierenden Netzes abhängt. Zudem konnte allgemein bewiesen werden, dass ein Netzwerk bestehend aus n Knoten nach außen hin wie ein Knoten mit m Ports angesehen werden kann (wobei gilt: $n \geq m$). Dieser Ansatz wird bei der substitution einer Boundary Clock herangezogen. Dies ist jedoch nur möglich, wenn alle Zeitinformationen vom eingehenden Port verzögerungslos an die ausgehenden weitergeleitet werden kann, da jede weitere Verzögerung eine Verfälschung der Messung des Übertragungsdelay darstellt. Um dies zu erreichen wurde das Konzept der Transparent Clock herangezogen, welche die übertragenen Zeitinformationen dementsprechend anpasst, dass die Übertragung transparent und ohne ersichtliche Abweichung in der Verzögerung erfolgt (vgl. [Ferrari u. a. \(2008\)](#)). Betrachtet man die Erkenntnisse dieser Arbeit iterativ über ein gesamtes Netzwerk mit gleichberechtigten Teilnehmern, so erhält man Ansätze eines Peer-to-Peer basierten Netzes, welches die Funktionalität des neuartigen Ansatzes unterstreicht und somit ein wichtiger Hinweis der Machbarkeit darstellt.

3.1.1 Fehlererkennung und Kompensation

Ein weiterer Teil der Arbeit bestand darin, potentielle Fehlerquellen in der Synchronisierung zu kategorisieren, zu analysieren und ggf. Lösungsansätze zu präsentieren, mit deren Hilfe die Auswirkungen vernachlässigbar gering gehalten werden kann. Ein Peer-to-Peer basierendes Modell könnte diese Ansätze aufgreifen und somit die Stabilität des Ansatzes unter der Architektur eines offenen Netzwerkes gewährleisten. Zu den Fehlerkategorien zählen, asymmetrische Übertragungszeiten, Schwankungen der Ausführungszeit zwischen der Anwendungsschicht und der physikalischen Übertragungskomponente, Ungenauigkeit beim Abnehmen der Zeiten, sowie die alterungsbedingte und thermische Instabilität des eigentlichen Taktgebers des Knotens. Asymmetrische Übertragungszeiten können durch Reduzierung der Anzahl der Hops minimiert werden. Auch ist es möglich, die Übertragungszeiten nicht dynamisch zur Laufzeit zu bestimmen, sondern zur Konfigurationsphase zu setzen. Dies ist jedoch für einen Peer-to-Peer basierenden Ansatz nicht möglich, da hier dynamisch zur Laufzeit Links zwischen Knoten entstehen und wieder wegfallen können, womit der Fokus auf der Beschränkung der Anzahl an Hops liegt. Schwankungen zwischen der Ausführungs- und Übertragungsschicht lassen sich nur mit entsprechender Hardwareunterstützung unterdrücken. Reduzierung der Komplexität der Algorithmen und Abarbeitungspfade mit gleichem Aufwand vermindern zwar die Wahrscheinlichkeit von Schwankungen, können diese jedoch nicht eingrenzen. Um Unge-

naugigkeiten beim Abnehmen der Zeiten entgegen zu wirken können in der Kommunikation sogenannte *follow-up* Pakete eingesetzt werden. Diese werden ebenfalls innerhalb des Precision Time Protocol eingesetzt und trennen das Ereignis des Ablesens des Zeitgebers von der Übertragung der Information, was zu einer erhöhten Präzision führt (vgl. [Loeser und Haertig \(2004\)](#)). Jedoch wird das Nachrichtenaufkommen um die *follow-up* Nachrichten erweitert, was in einer zusätzlichen Belastung der Bandbreite resultiert. Die letzte Kategorie, das Schwanken des Taktgebers, lässt sich weder durch das Protokoll noch mithilfe der Architektur eingrenzen und bleibt somit einer der Hauptfaktoren für die Ungenauigkeit eines Peer-to-Peer basierten Algorithmusses.

3.2 Skewless Network Clock Synchronization

Die Arbeit [Mallada u. a. \(2013\)](#) der School of Electrical and Computer Engineering Cornell in Zusammenarbeit mit Mitarbeitern des Thomas J. Watson Research Centers von IBM entstand ein Synchronisationsalgorithmus, welcher ein Verteiltes System in den synchronisierten Zustand überführt, ohne die genauen Abweichungen von Offset und Taktrate unter den Teilnehmern zu ermitteln und in einer Reihe von Versuchen bedingt auf den Anwendungsfall Stabilität zeigt. Das Verhalten wurde mathematisch nachgewiesen, der Algorithmus implementiert und unter unterschiedlichen Bedingungen getestet. Der Ansatz besteht aus einer Differenzgleichung deren Glieder aus exponentiell gewichteten Termen besteht. Der Algorithmus definiert sich durch das Anpassen der gewichteten Termen in Abhängigkeit eines gemessenen Offsets, was im aktuellen Korrekturwert für die lokale Zeit resultiert. Mit diesem Algorithmus wird erreicht, dass sich Netzwerke, welche als Graphen dargestellt wurden in einen Zustand der Konvergenz bewegen. Die Konvergenz konnte über Theoreme der Graphentheorie für verbundene Graphen mit realem Eigenwert nachgewiesen werden.

Die Versuchsreihe erfolgte über einen konfigurierbaren Server-Cluster, welcher über das entwickelte Protokoll synchronisiert wurde. Zum Vergleich erfolgte bei gleicher Topologie eine Synchronisation des Clusters über NTP (Network Time Protocol) und CCT (Coordinated Cluster Time), einem Produkt von IBM. Unter diesen Bedingungen wurden verschiedene Aspekte des Algorithmusses untersucht. Ausgehend von einem synchronisierten Netzwerk wurden diesem zunächst Störeinflüsse in Form von variierenden einzelnen Pulsen sowie ein gleichmäßig verteiltes Rauschen zugeführt. Es konnte erkannt werden, dass die Stabilität des vorgestellten Algorithmuses von der Frequenz abhängt, mit welcher ein Knoten die Informationen über die Zeitwerte seiner Nachbarn erhält. Obwohl für den Zusammenhang kein formeller Ausdruck

genannt wurde, so konnte dennoch experimentell festgestellt werden, dass das Zusammenspiel zwischen Messfrequenz und Topologie Einfluss darauf ausübt ob ein System einen stabilen Zustand erreicht bzw. diesen halten kann. Die Untersuchung Topologischer Veränderungen wurden speziell mit dem Focus auf Timingloops durchgeführt. Die Höhe der Rekursionsstufe einer Timingloop beeinflusste die Synchronisationseigenschaften nicht zwingend negativ.

Ein Weiterer Aspekt der Untersuchung besteht aus dem Leistungsvergleich des vorgestellten Ansatzes gegenüber Protokollen wie NTPv4 und CCT. Der Vergleich betrachtet die Konvergenzgeschwindigkeit der Algorithmen sowie die Präzision nach langer Laufzeit. Alle Messläufe erfolgten über einen Hop mit einem unidirektionalen Synchronisationspfad. Für die Betrachtung der Konvergenzgeschwindigkeit wurde eine Differenz der zu vergleichenden Uhren von 25 ms als Startbedingung gewählt. Die Messung ergab eine Konvergenz des Algorithmusses auf eine Restabweichung von $20\ \mu\text{s}$ nach einer Laufzeit von ca. einer Stunde. Hierbei wurden keine Informationen über Zyklusanzahl oder Messfrequenz genannt. Für die Betrachtung der Präzision wurde ein Langzeittest von 30 Stunden geführt. Hierbei wurde von einem synchronisierten System ausgegangen und ein gleichverteilter Jitter von $160\ \mu\text{s}$ generiert. Der eigene Algorithmus erreicht auch hier die geringsten Abweichungen der lokalen Uhr von maximal $10\ \mu\text{s}$. Diese Präzision konnte von den Protokollen CCT und NTPv4 im Vergleich nicht erreicht werden. Es ist zu beachten, dass die Parametrisierung des vorgestellten Algorithmusses für die jeweiligen Testszenarien optimiert wurden.

Es konnte somit gezeigt werden, dass der Algorithmus für spezielle Netze stabil ist. In weiteren Untersuchungen konnte für weiterführende Fälle exemplarisch festgestellt werden, wie sich die Parameter des Ansatzes auf die Stabilität des Systems auswirken. Dies wurde jedoch nicht formal beschreiben und bietet somit keine Gesetzmässigkeit um im Planungsstadium eines beliebigen oder speziellen Netzwerks mit Timingloops die Stabilität zu zeigen. Die experimentelle Leistungsfähigkeit des Algorithmus übertraf die der Vergleichsprotokolle, jedoch mit speziellen Parametren, deren Ermittlung nicht genauer erläutert wurde. Für einen Netzwerkdesigner wäre ein Regel oder Rechenwerk von nutzen, dessen Ergebnis ein Parametersatz ist, welcher für eine spezielle (oder allgemeine) Topologie einen stabilen Algorithmus mit den bestmöglichen Konvergenzeigenschaften erreicht.

Die hier präsentierte Arbeit soll einen neuartigen Ansatz zur Betrachtung und Umsetzung eines rollenlosen Peer-to-Peer-basierten Synchronisationsalgorithmus aufzeigen. Es gilt die Stabilität eines solchen Algorithmus zu untersuchen und Möglichkeiten der Parametrisierung herauszuarbeiten. Im Fokus stehen hierbei Topologien mit Timingloops, da jede bidirektionale

3 Hintergrund und verwandte Arbeiten

Verbindung zwischen zwei sich gegenseitig synchronisierenden Knoten als Loop dargestellt werden kann. Dies ist das Basiskonzept des in dieser Arbeit dargestellten Algorithmus.

4 Nachrichtenfluss und Informationsgewinnung

Zur Synchronisation zweier beliebiger Systeme miteinander ist ein Informationsaustausch zwischen diesen notwendig, was durch ein festgelegtes Nachrichtenprotokoll beschrieben wird. Das folgende Kapitel beschreibt eine Abschätzung, welche Informationen minimal für jeden Synchronisationspartner vonnöten sind um eine bidirektionale Synchronisation in einem Systeme zu erreichen. Hierzu wird die Art der nötigen Informationen, sowie die Anzahl der zum Informationsaustausch nötigen Nachrichten bzw. deren Frequenz betrachtet.

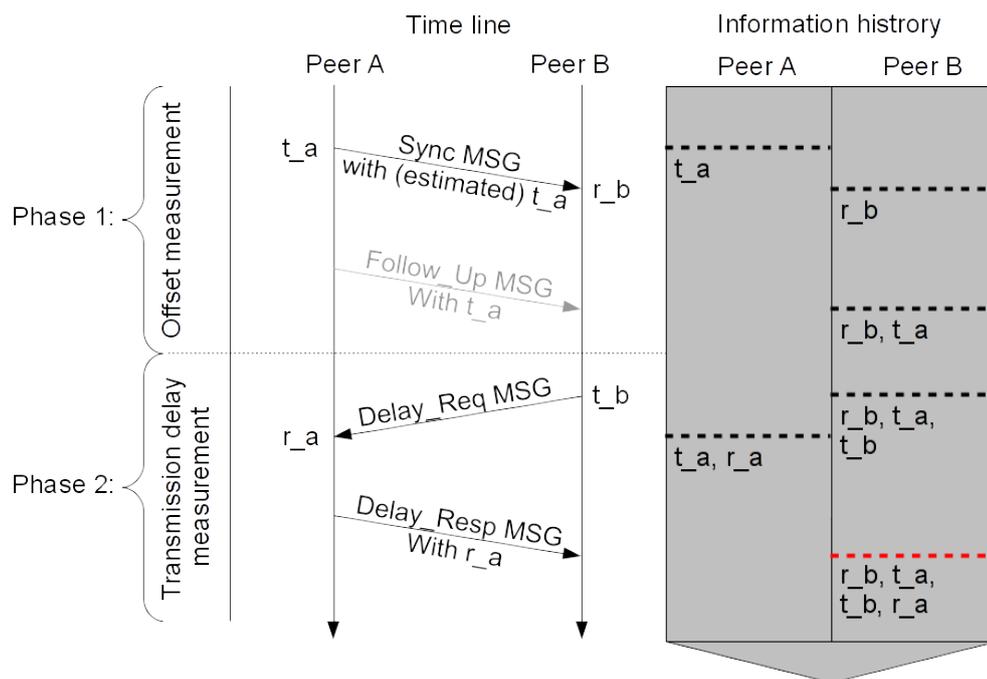


Abbildung 4.1: Time Precision Protocol - Informationsfluss

Im Folgendem soll am Beispiel vom Precision Time Protocol der Einsatz von standardisierten Protokollen zur Erzeugung eines einfachen P2P-Pfades gezeigt werden. Ist auf einem Pfad

eine bidirektionale Synchronisation abbildbar, so ist das P2P-Netzwerk als Ganzes analog darzustellen und kann mit standardisierten CTOS-Komponenten realisiert werden. Für eine bidirektionale Synchronisation ist es für einen rollenbasierten Synchronisationsalgorithmus erforderlich, dass Knoten zur gleichen Zeit mehrfache Rollen annehmen können. Das Precision Time Protocol erlaubt die Konfiguration eines Knotens sowohl als Master wie auch als Slave und definiert diesen Zustand als Boundary clock.

Der Synchronisationsprozess des Precision Time Protocols definiert sich durch zwei Phasen (vgl. Abbildung 4.1 auf der vorherigen Seite), der Offset- und der Delaymessung. Diese wiederholen sich zyklisch, sind jedoch über individuelle Zyklenlängen voneinander entkoppelt. Dies ermöglicht die vergleichsweise bandbreitenhungrige Delaymessung, unter der Annahme, dass alle Delays im Netzwerk keiner regelmässigen Änderung unterworfen sind, weniger oft einzusetzen. In der Phase der Offsetmessung wird vom Master (Peer A) der aktuelle Zeitstempel an alle Slaves (Peer A) versendet. Der Slave hat die Option zu einer beliebigen Zeit eine Delaymessung vom Master anzufordern. Über die Laufzeit der Anfrage und der resultierenden Antwort des Masters kann unter der Annahme, dass Sende- und Empfangsdelay nahezu gleich sind die Übertragungsverzögerung ermittelt werden.

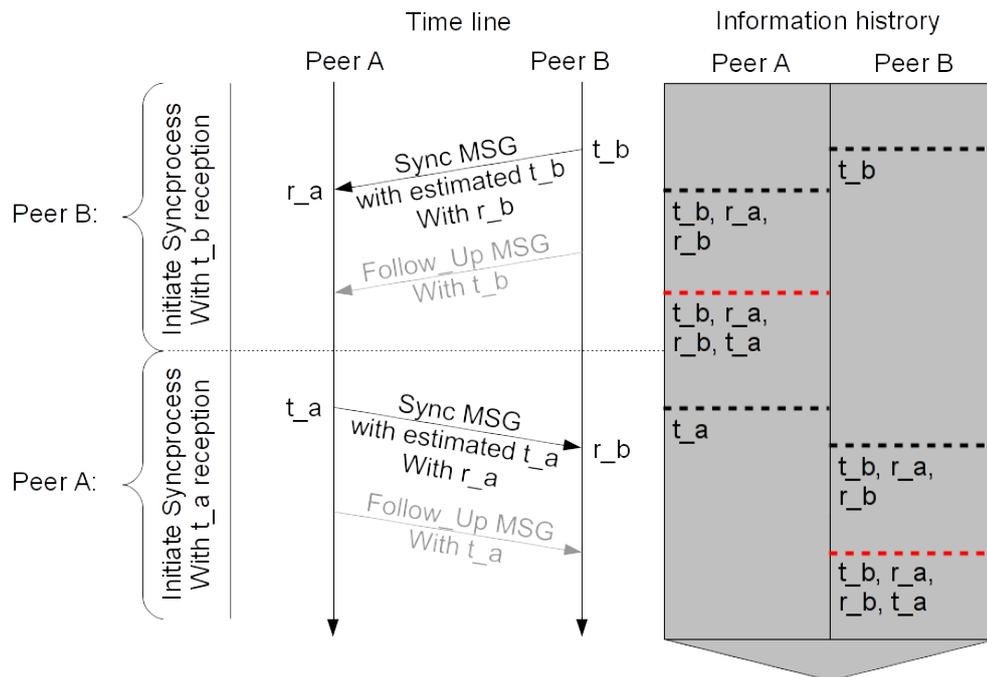


Abbildung 4.2: Peer-to-Peer Protocol - Informationsfluss

Am Aufbau des Precision Time Protocols ist ersichtlich, dass für eine erfolgreiche Synchronisation in einer Richtung ausschliesslich die in Abbildung 4.1 auf Seite 15 dargelegten vier Zeitpunkte vonnöten sind. Hierzu zählt der Sendezeitpunkt t_{t_a} der vom Master gesendeten Nachricht zur Offsetmessung und der zugehörige Empfangszeitpunkt t_{r_b} des Slaves. Weiterhin ist es notwendig eine Nachricht in umgekehrter Richtung zu verschicken, welche innerhalb des PTP als Delayanfrage operiert. Sind der Sendezeitpunkt t_{t_b} aufseiten des Slaves, sowie der Empfangszeitpunkt t_{r_a} des Masters bekannt, so hat der Slave das Wissen über alle vier Zeitpunkte und ist so in der Lage die Uhrendifferenz zum Master, sowie die Übertragungsverzögerung nach Vorschrift 4.2 zu errechnen. Für eine Implementierung als Peer-to-Peer-Ansatz setzt dies eine direkte Kopplung des Zyklus der Offsetmessung mit dem der Delaymessung voraus. Zudem gilt für solch einen Ansatz entsprechende Zeitpunkte bidirektional für beide Peers zu ermitteln. Somit eignet sich jedes Synchronisationsprotokoll, welches diese Informationen bereit stellt für die Realisierung eines P2P-basierten Ansatzes. Aus dieser Prämisse resultiert das folgende spezielle P2P-Protokoll, welches das Ziel verfolgt die nötigen Informationen mit einem möglichst geringen Kommunikationsaufwand zu verteilen.

$$\begin{aligned} t_{r_b} - t_{t_a} &= \text{Offset} + \text{TransmitDelay}(M \rightarrow S) \\ t_{r_a} - t_{t_b} &= -\text{Offset} + \text{TransmitDelay}(S \rightarrow M) \end{aligned} \quad (4.1)$$

$$\begin{aligned} \text{Offset} &= \frac{(t_{r_b} - t_{t_a}) - (t_{r_a} - t_{t_b})}{2} \\ \text{Delay} &= \frac{(t_{r_b} + t_{t_a}) - (t_{r_a} - t_{t_b})}{2} \end{aligned} \quad (4.2)$$

Abbildung 4.2 auf der vorherigen Seite zeigt den Aufbau des P2P-Synchronisationsprozesses. Alle erforderlichen Zeitinformationen können durch zwei Synchronisationsnachrichten erzeugt werden. Um nach Vorschrift 4.1 und folglich 4.2 die eigene Uhr mit der eines anderen Peers in Relation stellen zu können, muss die von einem Peer versendete Synchronisationsnachricht nicht nur den aktuellen Versendezeitpunkt, sondern auch den Empfangszeitpunkt der vorangestellten Nachricht des anderen Peers enthalten. Unterstützt der Peer das Einfügen des Absendezeitpunkts nicht on-the-fly (d. h., er kann diese Information nicht eigenständig in die zu sendende Nachricht einfügen), so muss eine Follow-Up-Nachricht erzeugt werden, welche

diese Information nachreicht. Ist diese Übertragung sichergestellt, so hat ein Peer, welcher eine Nachricht empfängt, mit genau diesem Empfangszeitpunkt alle Informationen, die zur Berechnung notwendig sind. Diese Kommunikationsstruktur führt zu einer Verschränkung der Synchronisationszyklen benachbarter Peers miteinander, da die Antwort des einen Peers die Anfrage des anderen darstellt und umgekehrt. Dieser Ansatz erlaubt so den Austausch aller zur Synchronisation notwendigen Informationen mit einem minimalen Kommunikationsaufwand von einer Nachricht pro Peer und stellt so das erreichbare Minimum an Bandbreitenbelastung durch ein bidirektionales Synchronisationsprotokoll dar.

5 Ableitung des Synchronisationsmodells

Diese Arbeit untersucht die Einsetzbarkeit des Modells oder einer modifizierten Variante für die in Kapitel 2.1 auf Seite 5 dargestellte Problemstellung und hat ggf. einen entsprechenden Nachweis über die Konvergenz des Systems zu erbringen. Hierzu sollen Theorien über Mehrgrößensysteme von Regelungsalgorithmen herangezogen werden, in welchen der Nachweis der Konvergenz zur Untersuchung von Systemeigenschaften etabliert ist. Vorerst ist es jedoch notwendig, die zu synchronisierenden Systemeigenschaften in die Regelungstechnik zu portieren. Im Folgenden wird hierzu die zu untersuchende Umgebung durch Mechaniken der Regelungstechnik abstrahiert und anschliessend in Kapitel 6 die Beweisführung der Konvergenz erbracht.

Das zu modellierende System besteht aus einer beliebigen Anzahl Knoten, welche jeder für sich ein lokales Verständnis der Zeit mit sich bringt. Aussagen über die Eigenschaften eines solchen Systems zu treffen erfordert eine Beobachtbarkeit und Vergleichbarkeit der Knoten zu schaffen. Hierzu wird das Verständnis einer globalen Zeit eingeführt, welche vergleichbar mit der Zeit eines externen Beobachters ist. Das Modell lässt sich somit aus einer Menge von Knoten $M = \{k \mid k \in nodes\}$ und deren Topologie T beschreiben. Da eine Betrachtung der Synchronisationseigenschaften und mit diesen eine Aussage über dessen Stabilität im Vordergrund steht, können alle Knoten in ihrer Funktion als identisch angenommen werden. Die Topologie bildet über eine Relation ab, welcher Knoten auf welche Nachbarn Einfluss in hinsicht auf die Uhrenkorrektur hat. Hierbei wird die logische Topologie betrachtet. Sollten logische und physikalische Topologie nicht übereinstimmen so kann dies über entsprechende verzögerungszeiten im Nachrichtentransfer abgebildet werden. Ein Knoten wird im Netzwerk durch seine lokale Zeit repräsentiert. Sei t_l die lokale, sowie t_g die globale Zeit so resultiert für jeden Knoten die euklidischen Geradengleichung:

$$g = \{(t_g|t_l) \mid t_l = s_f t_g + t_0\} \quad (5.1)$$

Hierbei werden mit Hilfe der Parameter s_f die Geschwindigkeit der internen Uhr sowie t_0 die Startabweichung der lokalen zur globalen Zeit definiert. Wird die aus der Fragestellung resultierende Annahme betrachtet, es sei ein System zu synchronisieren folgt, dass die Parameter s_f und t_0 unter den Teilnehmern variieren müssen. Für diesen Fall entsteht durch die Kopplung der Topologie ein eigendynamisches System, welches potentielle Energien durch die Differenz der lokalen Uhren unter den Teilnehmern aufbaut und zwischen ihnen austauscht. Es ist die Regel, dass das System hierbei nach beliebig langer Zeit keinen stabilen Zustand erreicht (vgl. lose gekoppelte Pendel). Daraus folgt, dass durch eine einfache direkte Kopplung der Knoten untereinander keine ausreichende Stabilität des Systems gewährleistet werden kann.

5.1 Methoden der Kopplung und Abstraktion

Da eine einfache direkte Kopplung der Knoten zum Erhalt eines stabilen Systems, wie in Kapitel 5 auf der vorherigen Seite diskutiert unzureichend ist, soll im weiteren Verlauf eine indirekte Variante der Kopplung gewählt werden. Hierzu ist eine ausreichende Abstraktion des Gesamtsystems Voraussetzung. Die hierzu eingesetzte Methode muss eindeutige Aussagen über die Eigenschaften des Systems nicht nur zu einem speziellen Zeitpunkt treffen können, sondern auch Rückschlüsse auf jeden Zustand des Systems zu zukünftigen oder vergangenen Zeitpunkten zulassen. Zusätzliche Beachtung findet die Realisierbarkeit und Art der Umsetzung eines solchen Ansatzes, da möglichst schonend mit gegebenen Ressourcen umgegangen werden muss. Der Algorithmus stellt die Synchronisation der Knoten untereinander sicher und teilt sich dabei Kommunikationsmedium und Rechenleistung mit weiteren Kommunikationsprotokollen und Anwendungen.

Der Austausch von Informationen oder Energien innerhalb eines verteilten Systems zum Erhalt eines globalen Potentials ist ein grundlegendes Konzept dessen Verhalten in einer Vielzahl von Disziplinen untersucht und bestimmt wird. Hierbei treten unterschiedliche Kopplungsvarianten auf, die im Folgenden untersucht und bewertet werden:

5.1.1 Maschinelles Lernen

Der Ansatz des Maschinellen Lernens dient dazu, anhand von Beispielen oder durch den kontinuierlichen Vergleich von Eingangs- und Ausgangsdaten durch Manipulation des Systems ein spezielles Verhalten zu provozieren. Da ein Knoten des Netzwerks ausschliesslich Infor-

mationen über die Differenz von lokalem Zeitgeber zu den Uhren der angrenzenden Knoten erhält, währe der geeignete Ansatz der des Reinforcement Learnings. Hierzu reicht der Vergleich der Zeitwerte für den Angleich des Algorithmus zum Erreichen eines synchronisierten Netzwerks. Die Schwierigkeit liegt nicht nur in der Vorherbestimmbarkeit der Konvergenz bei unterschiedlichen Topologien, sondern auch Aussagen über Synchronisationsgenauigkeit und Verbreitungsgeschwindigkeit durch das Netzwerk zu liefern, sei ein konvergentes Verhalten vorausgesetzt.

5.1.2 Schätzfunktionen

Da die lokale Zeit eines Knotens als lineare Abhängigkeit dargestellt werden kann (vgl. 5 auf Seite 19) lässt sich das Verhalten innerhalb eines Knotens als Schätzfunktion darstellen. Diese nutzen empirisch gesammelte Informationen um Eingangsinformationen zu vergleichen und ggf. den eigenen Algorithmus darauf hin abzustimmen. Durch aufstellen einer auf Stabilität optimierten Funktion kann vergleichsweise einfach eine Robustheit gegenüber Störungen innerhalb des Netzwerks erreicht oder auf dynamisches Verhalten innerhalb der Topologie reagiert werden. Die Realisierung auf jedem Knoten kann somit auf die selbe Weise erfolgen, wie die Beschreibung des Modells. Es lassen sich mit diesem Ansatz jedoch keine direkten Aussagen über das systemübergreifende Verhalten eines solchen Netzwerkes treffen, da über die Funktion das Verhalten eines jeden Knotens isoliert betrachtet wird.

5.1.3 Regelungsvorgänge

Vergleichbar mit dem Ansatz des maschinellen Lernens setzt der Regelungsvorgang die lokale Zeit eines Knotens ins Verhältnis mit der seiner Nachbarn. Mit Hilfe der Regelungseinheit und der Rückkopplung kann hierbei ein Potential möglichst konstant gehalten (Festwertregelung) oder einer vorgegebenen zeitlichen Änderung gefolgt werden (Folgeregelung). Der Regelungsvorgang lässt sich über gewöhnliche lineare Differentialgleichungen beliebiger Ordnung beschreiben. Betrachtet man das gesamte Netzwerk als zu regelndes System, so entsteht ein System von Differentialgleichungen. Aussagen über die Konvergenz eines verteilten Systems lassen sich jedoch nicht für Differentialgleichungssysteme direkt, sondern ausschliesslich im Zustandsraum treffen. Hierzu wird das Mittel der Zustandsstabilität herangezogen.

Mit Hilfe der Laplacetransformation werden Differentialgleichungen und Systeme in den Zustandsraum überführt. Somit kann anschliessend der Nachweis der Zustandsstabilität erbracht werden. Hierzu wird im Folgenden das Modell eines regelungsbasierten Ansatzes aufgestellt.

5.2 Bildung des Modells als Regelungsvorgang

Für die Erzeugung eines Regelungsmodells dient als Basis ein gedämpftes Feder-Masse-System, welches lokal immer nach endlich langer Zeit einen ausgleich der potentiellen Energie herstellt und somit in einen energielosen Zustand wechselt. Die Komponenten eines solchen Systems sind die auslenkende Kraft f_a , die dämpfende Kraft f_d , sowie die rücktreibende Federkraft f_f . Die auslenkende Kraft lässt sich mit Hilfe des newtonschen Aktionsprinzips durch $f_a = m \frac{d^2x}{dt^2}$ beschreiben. Analog zum physikalischen Modell wird durch die Masse m die Trägheit eines Knotens festgelegt. Dies definiert somit die Priorität, welche ein Knoten in einem Netzwerk einnimmt (interne Priorität). Die dämpfende Kraft ist durch $f_d = d \frac{dx}{dt}$ und die rücktreibende Kraft durch das Gesetz der allgemeinen Mechanik $f_f = cx$ bestimmt. Die Kombination aus Dämpfungskoeffizient d und Federkonstante c bestimmt hierbei die Dynamik des Systems. Diese beschreibt im Netzwerkmodell somit in welcher Zeit ein Knoten einen Ausgleich des Offsets anstrebt. Aus dem Wechselwirkungsprinzip nach Newton resultiert in diesem Feder-Masse-System die gewöhnliche Differentialgleichung 2. Ordnung:

$$m\ddot{x} + d\dot{x} + c \cdot (x - Offset_{ges}) = 0 \quad (5.2)$$

Ein Knoten muss jedoch in der Lage sein, sich auf multiple Nachbarn einstellen zu können. Hierzu ist es notwendig den Einfluss der einzelnen Knoten untereinander zu definieren. Dieser Einfluss kann mit Hilfe unterschiedlicher Verteilungsfunktionen erfolgen, um höherpriorisierte Anbindungen (z.B. über einen Backbone) zu unterstützen. Würde dieses Konzept über interne Prioritäten abgebildet, so wären statische Backbone-Gateways vonnöten, welche über ihre Masse einen vergleichbaren Stabilisierungsverlauf der Synchronisierung erreichen. Im weiteren Verlauf dieser Arbeit wird zur Veranschaulichung des Nachweises über die Stabilität des Verfahrens die Gleichverteilung $\frac{1}{p} \sum_{j=1}^p$ angenommen, welche über die Anzahl an Nachbarknoten p gebildet wird.

5.3 Gewichtungsfunktion

Werden keine weiteren Massnahmen wie z.B. Filterung der Synchronisationsnachrichten ergriffen, so bestimmt die Verteilungsfunktion zudem, wie ein Knoten auf Fehlerfälle reagiert. Der regelungstechnische Ansatz kompensiert im gegensatz zu Ansätzen des Reinforcement

Learnings optimal kurzfristige Störungen, kann jedoch kontinuierliche Fehlerquellen wie z.B. ein "Babbling Idiot" nicht selbstständig erkennen oder kompensieren. Alle eingehenden Synchronisationsnachrichten unterschiedlicher Knoten werden mit der selben Priorität betrachtet, was zur Folge hätte, dass die mögliche Fehlerkompensation direkt proportional zum Verhältnis von funktionsfähigen Knoten zu fehlerhaften Knoten steht. Dies stellt insbesondere in grob-vermaschten Netzen ein Problem dar, da durch einen einzelnen fehlerhaften Knoten die Qualität der Synchronisation benachbarter funktionsfähiger Knoten stark beeinträchtigt oder vollständig gestört werden kann.

Um auch eine Kompensation kontinuierlicher Fehlerquellen zu ermöglichen wird das Modell um dynamische zeitabhängige Gewichtungsfaktoren q erweitert. Diese sind verbindungsorientiert und bilden eine externe Priorisierung von Knoten ab. Ein Knoten kann somit anhand von unterschiedlichen Parametern wie z.B. Laufzeit oder Jitter seine Nachbarn einschätzen und klassifizieren. Diese Einschätzung ermöglicht es ihm den Einfluss, den er jeweils von seinen Nachbarn erhält, gesondert über die Funktion $\prod_{k=1}^m q_k$ zu Gewichten. Sollte ein Knoten über eine längere Zeit gestört sein oder dauerhaft ausfallen, so wird dieses über die externe Priorisierung von seinen Nachbarn erkannt und mögliche von ihm gesendete fehlerhafte Synchronisationsinformationen verlieren über die Zeit ihren Einfluss. Umgekehrt kann hierdurch ein Cluster, welcher über eine lange Laufzeit mit einer hohen Präzision zusammenarbeitet die Synchronisationsinformationen innerhalb dieses Clusters höher priorisieren und somit die Bindung verstärken.

Die zuvor diskutierte funktionale Betrachtung erfordert jedoch, dass bezogen auf eine einzelne Verbindung alle zur Synchronisation nötigen Informationen nicht mehrfach im Synchronisationsintervall eintreffen. Dies hätte den unerwünschten Effekt, dass eine Verbindung mit jeder zusätzlichen Information an Priorität gewinnt, was wiederum von fehlerhaften Knoten als Möglichkeit der indirekten Priorisierung misbraucht werden kann. Um eine Entkopplung des funktionalen Ansatzes zu schaffen bildet jeder Knoten verbindungsorientiert das arithmetische Mittel $\frac{1}{n} \sum_{i=1}^n$ über alle Informationen, die er bis zu seinem Synchronisationszeitpunkt erhalten hat. Hierdurch wird eine Unabhängigkeit des funktionalen Ansatzes gegenüber unterschiedlicher Synchronisationsintervalle erreicht.

Die betrachteten Ansätze des Verarbeitens mehrerer unterschiedlicher Synchronisationsquellen, die Behandlung verbindungsorientierter Prioritäten zur Kompensation kontinuierlicher Fehlerquellen, als auch die Unabhängigkeit des funktionalen Ansatzes gegenüber unter-

schiedlicher Synchronisationsintervalle lassen sich zusammenfassen. Es resultiert hieraus die Gewichtungsfunktion:

$$\frac{1}{p} \sum_{j=1}^p \left[\prod_{k=1}^m q_k \cdot \left(\frac{1}{n} \sum_{i=1}^n Offset_i \right) \right]_j = Offset_{ges} \quad (5.3)$$

Alle diskutierten Konzepte, die Systembetrachtung 5.1, Bildung der Differentialgleichung 5.2, sowie die funktionale Betrachtung der Prioritäten 5.3 bilden zusammen das Modell der Peer-to-Peer-Synchronisation.

5.4 Klassifizierung des Systems

Für die Wahl des richtigen Ansatzes ist es erforderlich, dass jeweilige zu regelnde System zu klassifizieren um die jeweiligen regelungstechnischen Konzepte richtig nutzen zu können. Statische Systeme zeichnen sich dadurch aus, dass der Wert des Ausgangssignals $y(t)$ zu jedem Zeitpunkt t nur vom aktuellen Wert des Eingangssignals $u(t)$ abhängt. Diese Systeme lassen sich mit Hilfe einer algebraischen Beschreibung darstellen. Hängt das Ausgangssignal von vergangenen Eingangssignalen ab, so spricht man von dynamischen Systemen, welche sich mit Hilfe der Differentialgleichungen beschreiben lassen. Das beschriebenen Systems (Gleichung 5.2 auf Seite 22) kann als Wirkungsanordnung mit ortsunabhängigen Signalen angesehen werden, da es ausschliesslich über konzentrierte Parameter (Masse m , Dämpfungskonstante d und Federkonstante c) verfügt. Somit ist keine Beschreibung als partielle Differentialgleichung vonnöten.

Abschätzung bekannter Regelungsverfahren

Ist ein System zeitinvariant, so lassen sich unterschiedliche Stabilitätsverfahren zum Nachweis der Konvergenz eines Systems für den Reglerentwurf heranziehen. Verfügt das System über mehrere Ein- und Ausgänge so spricht man vom Mehrgrößensystem oder MIMO-System. Hier sind die Ansätze der klassischen Regelungstechnik unzureichend. Für die Untersuchung netzwerkweiter Topologien ist somit eine Betrachtung als Zustandsregelung notwendig, mit deren Hilfe unter Anwendung verschiedener Normalformen Aussagen über das Regelverhalten eines Systems getroffen werden kann. Im folgenden Kapitel wird eine Abbildung in den

5 Ableitung des Synchronisationsmodells

zustandsraum und die Aufteilung der Teilkomponenten in die Reglerstruktur unternommen. Anschliessend kann die Strategie des Beweises festgelegt werden.

6 Beweisführung

Die Beweisführung unterteilt sich in zwei Stufen. Zunächst wird das Modell eines Knotens gebildet, welches dessen Verhalten vollständig beschreibt (vgl. 6.1). Anschliessend erfolgt die netzwerkweite Betrachtung durch Bildung eines Gesamtsystems aus einer Menge von Knoten und ihrer Topologie (vgl. 6.2 auf Seite 32). Der erste Schritt in der Beweisführung stellt die Vereinigung der drei Systemkomponenten zu einem in sich geschlossenem System dar. Diese vollständige Repräsentation des Systems kann im weiteren Verlauf als Basis für netzwerkweite Herleitungen herangezogen werden.

6.1 Vollständige Modellbetrachtung eines Knotens

Zunächst wird die Systembetrachtung in eine Differentialgleichung überführt. Hierzu wird die Geschwindigkeit isoliert und die Ableitung gebildet. Somit folgt aus der Gleichung 5.1 die gewöhnliche lineare DGL $x' - y = 0$ erster Ordnung. Um die Kopplung zwischen dieser DGL und die des Feder-Masse-Systems 5.2 zu ermöglichen wird eine Überführung der zwei Differentialgleichungen in den Spektralbereich des Laplace-Raums \mathcal{L} vorgenommen.

$$x' = y \tag{6.1}$$

$$mx'' + dx' + cx = c \cdot Offset_{ges} \tag{6.2}$$

Das Verhältnis vom $\mathcal{L}\{Ausgangssignal\}$ zum $\mathcal{L}\{Eingangssignal\}$ bildet die Übertragungsfunktion des Systems. Mit $X_a = \{x\}$ und $X_e = \{y, Offset_{ges}\}$ folgt:

$$X_a(s) \cdot s = X_e(s) \Rightarrow \frac{1}{s} \quad (6.3)$$

$$X_a(s) \cdot \left(s^2 + s \frac{d}{m} + \frac{c}{m} \right) = X_e(s) \cdot \frac{c}{m} \Rightarrow \frac{\frac{c}{m}}{s^2 + \frac{d}{m}s + \frac{c}{m}} \quad (6.4)$$

Die Gewichtsfunktion 5.3 stellt ausschliesslich einen funktional-linearen Zusammenhang dar und erfordert daher im Sinne der Beweisführung auf Stabilität keine Umwandlung. Es bietet sich jedoch für die weitere Betrachtung an, alle externen Prioritäten q zu vernachlässigen um somit eine Vereinfachung der Funktion zu erzielen. Zudem werden die konfigurierten Zykluslängen aller Teilnehmer als gleich angenommen. Mit $q_1, \dots, q_k = 1$ und $n = 1$ resultiert die Gewichtsfunktion:

$$\frac{1}{p} \sum_{j=1}^p (Offset_j) = Offset_{ges} \quad (6.5)$$

Somit ist die direkte Kopplung der Teilsysteme zu einem Gesamtsystem möglich und kann für eine Stabilitätsbetrachtung herangezogen werden.

6.1.1 Systembetrachtung

Das Gesamtsystem folgt aus der direkten Reihenschaltung der Teilsysteme. Abbildung 6.1 auf der nächsten Seite zeigt hierbei den Aufbau eines Knotens. Um jegliche Topologie zu unterstützen ist die Gewichtsfunktion mit j Nachbarn ($j \in \mathbb{N}_0$) koppelbar. Hierzu wird die eigene lokale Zeit als Referenz zurückgeführt um die Bestimmung des Deltas der jeweiligen Nachbaruhren (wie in Kapitel 4 auf Seite 15 beschrieben) abzubilden. Über das anschliessende Feder-Masse-System erfolgt die Parametrisierung des Systems. Mit Hilfe der Gewichtung m und der aus der Kombination aus Federkonstante c und Dämpfungsproportionalitätsfaktor d entstehenden Eigendynamik, lassen sich die Eigenschaften wie Stabilität und Schwingungsverhalten festlegen und bilden den für die Beweisführung zu untersuchenden Parametersatz. Das lokale Uhrenmodell ist nach Gleichung 6.3 als zusätzlicher Integralteil dargestellt. Mit Hilfe des eigenen Parameters werden Geschwindigkeitsschwankungen, sowie unterschiedliche Basisgeschwindigkeiten der lokalen Zeit auf verschiedenen Knoten provoziert, was zu einer

Geschwindigkeitskorrektur der lokalen Zeit der Knoten untereinander führen muss. Eine Offsetänderung kann mit Hilfe künstlich erzeugter Sprünge als Regelstörgröße modelliert werden.

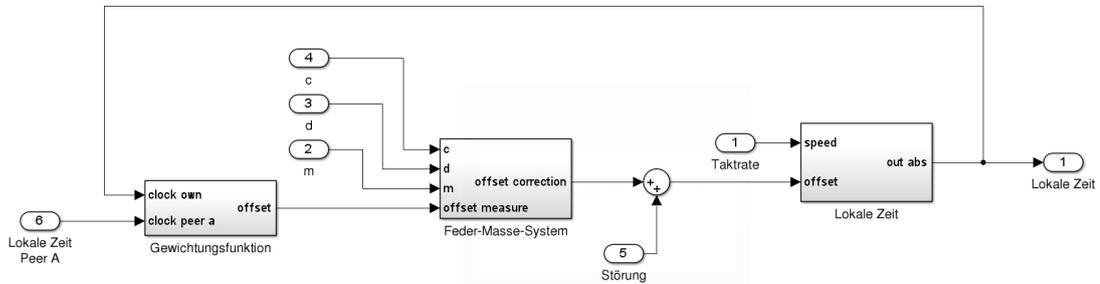


Abbildung 6.1: Ansatzselektion

6.1.2 Zustandsraummodell

Ein Netzwerk aus n Knoten besteht somit aus n Systemen, welche je nach Topologie untereinander Einfluss ausüben. Verfügt ein System über mehrere Ein- und Ausgänge, so spricht man von einem Mehrgrößensystem oder auch MIMO-System (vgl. [Golnaraghi und Kuo \(2009\)](#)). Die Stabilitätsbetrachtung eines MIMO-Systems erfolgt über die Zustandsraumbetrachtung, welche ein dynamisches Übertragungssystem durch seine Zustandsgrößen beschreibt. Diese Zustandsgrößen beschreiben durch ihre Eigenschaften das System in der Zeit und geben Aufschluss auf das energetische Gleichgewicht des systems. Dabei sind systembeschreibende Differentialgleichungen n -ter Ordnung mit n konzentrierten Energiespeichern in n Differenzialgleichungen 1. Ordnung zu zerlegend und in eine standartisierte Matrix- und Vektordarstellung zu bringen. Die Anzahl der Zustandsvariablen $x_1(t) \cdots x_n(t)$ des Zustandsvektors $X(t)$, ist die Dimension des Zustandsraumes. Im Zustandsvektor $X(t)$ sind alle Informationen des dynamischen Übertragungssystems zum beliebigen Zeitpunkt t_0 enthalten. Die Beschreibung eines Übertragungssystems im Zustandsraum erfolgt über das Zustandsraummodell und die angewandte Normalform. Das Zustandsraummodell kann für nicht sprungfähige Systeme direkt aus den Koeffizienten der systembeschreibenden Differentialgleichung oder der zugehörigen Übertragungsfunktion erstellt werden. Als normalform wird im Folgenden die kanonische Normalform gewählt, welche ein System als Sammlung von Teilsystemen, den Knoten des Netzwerks, beschreibt. Alle Knoten werden als von einander unabhängige Komponenten betrachtet. Die Kopplung der Knoten erfolgt ausschliesslich über den Regler, welcher sich über die Topologie

definiert und somit die Relation der Teilsysteme zueinander herstellt. Hierdurch kann eine Skalierung des Ansatzes auf beliebige Topologien erzielt werden.

Zur Überführung des Systems in die kanonische Normalform ist es erforderlich einen Knoten vollständig über eine Übertragungsfunktion zu definieren. Die Bestimmung der Übertragungsfunktion erfolgt bezüglich dieses Modells durch multiplikatives Zusammenfassen der Teilkomponenten: Gewichtung, Feder-Masse-System und der lokalen Zeit. Hierzu ist es notwendig alle Übertragungsfunktionen isoliert zu betrachten. Durch trennung mit Hilfe von Superposition des integrierenden Gliedes des Uhrenmodells von seiner Additionsstelle wird erreicht, dass Geschwindigkeitsparameter und Regelstörgröße als gemeinsame Störgröße Z betrachtet werden können. Um die Eigenstabilität des Systems zu untersuchen wird diese Regelstörgröße zunächst zu 0 angenommen. Anschliessend ist eine Untersuchung des Systems auf Störeinflüsse nötig. Hierzu wird der Eingangsvektor des Systems zu 0 angenommen und die Störgröße Z als Eingang interpretiert. Das Verhalten des Systems resultiert aus der Überlagerung beider Betrachtungen. Die Verteilungsfunktion 5.3 auf Seite 24 bildet hierbei den Knotenpunkt der systembeschreibenden Matrix. Dieser Ansatz berücksichtigt so jedoch nicht die Anzahl der gekoppelten Knoten j . Diese Anzahl kann jedoch als Liniarfaktor $\frac{1}{j}$ auf die Übertragungsstrecke eines jeden der gekoppelten Knoten übertragen werden.

Das Gesamtsystem bildet sich somit aus der Übertragungsfunktion des Feder-Masse-Systems $G_f(s)$, der des Systems $G_s(s)$ und der Rückkopplungsfunktion $G_r(s)$ der lokalen Zeit. Sei $G_f(s)$ Gleichung 6.4, und $G_s(s)$ Gleichung 6.3 so folgt für die Übertragungsfunktion des Knotens $G_k(s)$:

$$G_k(s) = \frac{G_f(s) \cdot G_s(s)}{1 + G_f(s) \cdot G_s(s) \cdot G_r(s)} \quad (6.6)$$

$$G_k(s) = \frac{\frac{c}{m}}{s^3 + \frac{d}{m}s^2 + \frac{c}{m}s + \frac{c}{m}}$$

Die Übertragungsfunktion $G_k(s)$ beschreibt nun das Eigenverhalten eines Knotens vollständig. Das Eigenverhalten lässt sich somit in ein Nenner- und Zählerpolynom aufteilen. Das Zählerpolynom definiert alle Nullstellen des Systems. es handelt sich hierbei um Invarianten, welche von der Rückführung des Systems nicht beeinflusst werden. Anhand der Nullstellen, welche das Ein- und Ausgangsverhalten des Systems beeinflussen, lassen sich bereits Aussagen

über die Stabilität treffen. Diese Betrachtung gibt jedoch unzureichender Weise ausschliesslich Aufschluss über die Stabilität eines einzelnen Knotens, jedoch nicht über die Eigenschaften in einem vernetzten System. Das Nennerpolynom beschreibt durch Definition der Polstellen die Eigenwerte des Systems. Durch sie wird die Eigendynamik des Systems bestimmt. Über die Lage der Polstellen lassen sich ebenfalls zwar Rückschlüsse über die Stabilität des isolierten Systems zu, jedoch nicht Aussagen über die Eigendynamik eines verteilten Netzes treffen.

Für eine netzwerkweite Betrachtung ist es somit notwendig die Übertragungsfunktion in das Zustandsmodell zu überführen, da anschließend mittels Superposition das Verhalten eines beliebigen Netzwerks als Zusammenschluss einzelner Stimuli (Knoten) bestimmt werden kann. Hierzu wird durch Anwendung der inversen Laplace-Transformation aus der Funktion die zugehörige lineare Differentialgleichung 3. Ordnung gewonnen. Mit $X_a = y$ und $X_e = u$ folgt somit:

$$\frac{\frac{c}{m}}{s^3 + \frac{d}{m}s^2 + \frac{c}{m}s + \frac{c}{m}} \bullet \text{---} \circ \quad \frac{c}{m}y = x''' + \frac{d}{m}x'' + \frac{c}{m}x' + \frac{c}{m} \quad (6.7)$$

Anschliessend kann eine Aufteilung der Differentialgleichung in ihre Zustandsvariablen $\underline{X} = \{x_1(t), x_2(t), x_3(t)\}$ erfolgen. Es folgt daraus das Gleichungssystem 6.8 bestehend aus drei Differentialgleichungen 1. Grades, sowie die Abhängigkeitsfunktion 6.9 des Ausgangs:

$$\begin{aligned} x_1' &= x_2 \\ x_2' &= x_3 \end{aligned} \quad (6.8)$$

$$x_3' = -\frac{d}{m}x_3 - \frac{c}{m}x_2 - \frac{c}{m}x_1 + \frac{c}{m}u$$

$$y = x_1 \quad (6.9)$$

Das Gleichungssystem lässt sich in eine Matrixdarstellung 6.10 und der Ausgangsgleichung 6.11 überführen:

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{c}{m} & -\frac{c}{m} & -\frac{d}{m} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{c}{m} \end{bmatrix} \cdot u(t) \quad (6.10)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (6.11)$$

Aus ihnen lassen sich nun die spezifischen Matrizen der Regelungsnormalform ableiten und anschliessend in die Form 6.12 bringen:

$$\begin{aligned} \underline{x}'(t) &= A \cdot \underline{x}(t) + B \cdot u(t) \\ y(t) &= C \cdot \underline{x}(t) + D \cdot u(t) \end{aligned} \quad (6.12)$$

Das System lässt sich nun in der Regelungsnormalform (vgl. Lyons (2010)) abbilden. Die Form definiert sich durch 4 spezifische Matrizen A , B , C und D . Die Matrix A stellt hierbei die Systemmatrix dar, welche die Zustandsvariablen enthält und somit das Systemverhalten beschreibt. Durch sie sind alle Rückführungen der Regelungsstrecke festgelegt und beschreibt so, das Eigenverhalten des Systems ohne jegliche Einwirkungen von außen. Die Eingangsmatrix B legt die Ansteuerung des Systems durch die Eingänge dar. Somit ist die Art der Einkopplung des Eingangssignal durch sie festgelegt. Die Auskopplung des Systems erfolgt über die Ausgangsmatrix C und beschreibt die beobachtbaren Auswirkungen des Systems. Die Durchgangsmatrix D bildet eventuelle direkte Einflüsse der Eingänge auf die Ausgänge des Systems ab. Durch sie wird ein System sprungfähig, d.h. eine Änderung am Eingang des Systems kann zu einer direkten Folge am Ausgang des Systems führen.

Durch Koeffizientenvergleich der Form 6.8 und 6.9 kann die Bestimmung der spezifischen Matrizen der Regelungsnormalform erfolgen. Für die Beschreibung des Verhaltens eines Netzwerkknotens leiten sich somit die folgenden Matrizen ab:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{c}{m} & -\frac{c}{m} & -\frac{d}{m} \end{bmatrix} \quad (6.13)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{c}{m} \end{bmatrix} \quad (6.14)$$

$$C = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.15)$$

$$D = [\emptyset] \quad (6.16)$$

Die Nennerkoeffizienten (Pole) der Übertragungsfunktion sind ausschließlich in der Systemmatrix A enthalten. Sie weist eine spezielle Struktur über ihre Elemente $a_{ij} \in \mathbb{R}^{n \times n}$ auf. Von diesen sind nur die n Elemente der letzten Zeile (a_{nj}) von den Koeffizienten innerhalb des Übertragungssystem abhängig. Da die Anzahl der Pole größer ist als die der Nullstellen, handelt es sich um ein nicht sprungfähiges System. Dies resultiert in einer Nullmatrix als Durchgangsmatrix D . Für die Stabilitätsuntersuchung ist der Einfluss auf den Ausgang des Systems interessant. Im folgenden wird zunächst aus der hier erarbeiteten Systembeschreibung eine einzelnen Knotens das Zustandsraummodell eines Netzwerkes abgeleitet um aus ihm eine Untersuchung des Antwortverhaltens von Ausgangssignal auf Eingangssignal zu ermöglichen.

6.2 Systembetrachtung des Netzwerkes

Ein sich selbst regelndes Netzwerk mit beliebig vielen Teilnehmern stellt ein System mit mehreren Ein- und Ausgängen dar. Jedes der Systeme kann zum Synchronisationsstatus des Netzwerkes beitragen (Ausgang) oder von diesem in seinem eigenen Synchronisationsvorhaben beeinflusst werden (Eingang). Zur Erstellung eines solchen Systems bedient sich die Rege-

lungstechnik des Konzepts der Mehrgrößensysteme oder MIMO-Systeme. Bildet man dieses in der kanonischen Normalform ab, so erhält man ein System aus beliebig vielen entkoppelten Teilnehmern. Ein Netzwerk in der Normalform ist somit eine geordnete Menge an Knoten, und benötigt eine zusätzliche Beschreibung der Topologie. Dies stellt somit eine unabhängige Beschreibungsform von Eigenschaften der Knoten und der der Netzwerkarchitektur dar. Die Gleichungen 6.17 sowie 6.18 beschreiben eine beliebig große Menge an Knoten eines Netzwerks in kanonischer Normalform.

$$\begin{bmatrix} x'_{11} \\ \vdots \\ x'_{nk} \end{bmatrix} = \begin{bmatrix} A_1 & & \emptyset \\ & \ddots & \\ \emptyset & & A_k \end{bmatrix} \cdot \begin{bmatrix} x_{11} \\ \vdots \\ x_{nk} \end{bmatrix} + \begin{bmatrix} B_1 & & \emptyset \\ & \ddots & \\ \emptyset & & B_k \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ \vdots \\ u_k \end{bmatrix} \quad (6.17)$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} C_1 & & \emptyset \\ & \ddots & \\ \emptyset & & C_k \end{bmatrix} \cdot \begin{bmatrix} x_{11} \\ \vdots \\ x_{nk} \end{bmatrix} + \underline{\emptyset} \cdot \begin{bmatrix} u_1 \\ \vdots \\ u_k \end{bmatrix} \quad (6.18)$$

Die Matrizen (A , B , C und D) eines jeden der einzelnen Systeme werden als Blockmatrizen in die systembeschreibenden Matrizen des Gesamtsystems und somit des Netzwerks eingesetzt. Sie bilden die Hauptdiagonalen der Matrizen der Knotenmenge. So entsteht das Gesamtsystem als entkoppeltes System aus beliebig vielen Eingrößensystemen. Der Index läuft hierbei über alle Knoten und bestimmt die Dimension der systembeschreibenden Matrix und des Aus- sowie Eingangsvectors. Die Dimension ($k \cdot k$) aller Matrizen des Gesamtsystems richtet sich bei dieser Betrachtung immer nach der Anzahl an Teilnehmern innerhalb des Netzwerks. Sie ist weiterhin stets quadratisch, so dass eine Abhängigkeit für jeden Knoten von jedem anderen zu beschreiben ist.

Auf der Basis dieses Mehrgrößensystems können, wie im folgenden Kapitel beschrieben, die spezifischen Matrizen für beliebige aber jeweils spezielle Topologien des Netzwerkes aufgestellt werden.

Betrachtung: Topologie als Regler

Das System bestehend aus den Gleichungen 6.17 und 6.18 beschreibt in seiner Modellierung eine Menge von Knoten, welche auf Eingangsveränderungen und lokale Störgrößen in Abhängigkeit ihrer lokalen Parameter, welche die Gewichtung und Eigendynamik eines jeden

Knotens festlegen, reagieren. Diese sind von einander zwar entkoppelt, jedoch als ein zusammengehöriges System beschrieben. Die Topologie kann nun unabhängig von der Darstellung des Systems als Reglerpfad modelliert werden. Dieser Pfad definiert die Rückführungen und somit den Eingang für jeden einzelnen Netzwerkteilnehmer. Das Gesamtsystem kann somit wie in Abbildung 6.2 beschrieben dargestellt und modelliert werden. Es wurde auf die Darstellung der Durchgangsmatrix verzichtet, da für das System gilt: $D = [\emptyset]$

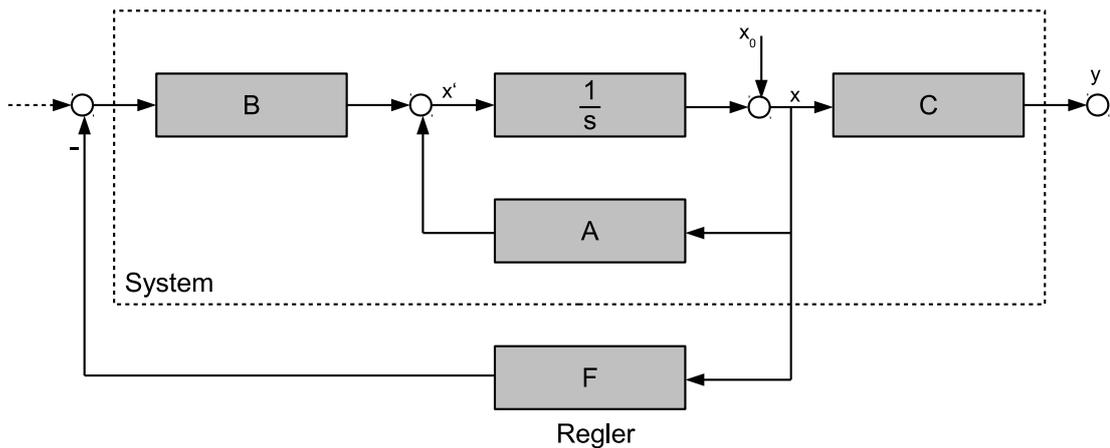


Abbildung 6.2: Der Regler als Netzwerktopologie

Aus der Abbildung 6.2 lässt sich der Stellvektor des Reglers als $\underline{u}(t) = -F\underline{x}(t)$ ableiten. Soll nun für ein gegebenes Netzwerk die Matrix des Reglers bestimmt werden, so müssen alle Größen in Abhängigkeit der abstrakten Zustandsgrößen x_{0k} bis x_{2k} innerhalb jedes einzelnen Knotens k ermittelt werden. Eine direkte und effiziente Beschreibung einer Netzwerktopologie ist an dieser Stelle wünschenswert, welche von den Verhaltensweisen innerhalb eines einzelnen Netzwerkteilnehmers abstrahiert. Eine abstrakte Netzwerkbeschreibung kann durch die Definition des Einflusses eines Netzwerkknotens auf seine Nachbarn erreicht werden. Der Einfluss ist unidirektional und definiert sich wie folgt:

$$\tau = \begin{cases} 1, & \text{falls Einfluss dieses Ausgangs auf Knoten} \\ 0, & \text{sonst} \end{cases} \quad (6.19)$$

Der Wert ließe sich für erweiterte Protokollfunktionalität durch einen Gewichtungsfaktor beeinflussen, was innerhalb der Basisfunktionalität des P2P-Protokolls jedoch nicht vorgese-

hen ist und daher im weiteren Verlauf dieser Arbeit nicht betrachtet wird. Mit der direkten Beschreibung lässt sich der Einfluss der Ausgänge und somit der Knoten eines Netzwerks auf die Eingänge durch die Übertragungsmatrix T in der Form $\underline{u}(t) = T\underline{y}(t)$ (6.20) darstellen.

$$\begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} = \begin{bmatrix} \tau_{11} & & \tau_{1r} \\ & \ddots & \\ \tau_{m1} & & \tau_{mr} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ \vdots \\ y_r \end{bmatrix} \quad (6.20)$$

Die Matrix T kann zeilenweise interpretiert bzw. generiert werden. Jede Zeile beschreibt das Einflussverhalten u für einen bestimmten Knoten aus $[1, m]$. Das Einflussverhalten, welches dieser am Eingang erfährt bildet sich aus der Summe aus Ausgangskomponenten y möglicher Nachbarknoten innerhalb des Intervalls $[1, r]$. Da die Matrix den Einfluss von jedem Knoten auf jeden Knoten beschreibt ist diese immer quadratisch. Alle unidirektionalen Einflüsse lassen sich paarweise, durch Spiegelung auf der Hauptdiagonalen, als bidirektionale Einflüsse darstellen.

Diese Form ermöglicht die Darstellung jeder möglichen Topologie während die Funktionsweise jedes einzelnen Knotens gekapselt wird. Sie benötigt jedoch die folgende Umformungsvorschrift 6.21, welche die Matrix F des Reglers erzeugt. Durch einsetzen der Ausgangsgleichung 6.12 der Regulationsnormalform in die Beschreibungsgleichung 6.20 der Topologie resultiert die Form für den Stellvektor des Reglers:

$$\begin{aligned} \underline{u}(t) &= TC\underline{x}(t) \\ \Rightarrow F &= -TC \end{aligned} \quad (6.21)$$

Somit kann für ein beliebiges Netzwerk die Matrix des Reglers bestimmt werden. Das Ergebnis ist ein geregeltes Mehrgrößensystem, welches das Verhalten des Netzwerks abbildet. Da nun eine vollständige Beschreibung bereit steht, kann diese Form die Basis einer Stabilitätsbetrachtung eines Netzwerks bilden.

6.3 Stabilität des Gesamtsystems

Die Stabilität eines geregelten Systems lässt sich durch unterschiedliche Ansätze zeigen. Hierbei ist zunächst die Kategorisierung der zu betrachtenden Regelstrecke notwendig, da für unterschiedliche Systeme jeweils spezielle Ansätze der Stabilitätsbedingungen herangezogen werden.

Reagiert ein System auf jede beschränkte Eingangsfunktion $u(t)$ mit einer beschränkten Ausgangsfunktion $y(t)$ so ist das System extern stabil oder BIBO stabil. Es existiert somit eine Schranke M für die für alle Zeitpunkte gilt: $x(t) < M < \infty$. Dieser Ansatz findet Anwendung bei Systemen, deren interne Struktur nicht bekannt und auch nicht ermittelbar ist. Sind alle Informationen über das interne System vorhanden, so kann die Definition der internen Stabilität herangezogen werden. Hierbei bestimmen die Pole der Übertragungsfunktion die Stabilität und die Reaktionsfreudigkeit mit der sich das System in einer bestimmten Zeit einem Zustand annähert. Die Nullstellen der Übertragungsfunktion geben hierbei ausschliesslich Aufschluss auf die Amplitude des Systems und sind somit für die Stabilitätsbetrachtung nicht von Relevanz. Ein Übertragungssystem ist intern stabil, wenn alle Übertragungsfunktionen ausschließlich Pole mit negativen Realteil aufweisen, also sich auf der linken s-Halbebene befinden.

Die Position der Pole kann über das charakteristische Polynom des Systems gezeigt werden, welches im Folgenden hergeleitet wird. Mit Hilfe der Zustandsgleichungsgleichungen [6.17 auf Seite 33](#), sowie der Ausgangsgleichung [6.18](#) nach Anwendung der Reglergleichung des Stellvektors aus [6.2](#) beschreibt sich das System wie folgt:

$$\underline{x}'(t) = A\underline{x}(t) + B \cdot (V\underline{w} - T\underline{y}) \quad (6.22)$$

$$\underline{y}(t) = C\underline{x}(t) + D \cdot (V\underline{w} - T\underline{y}) \quad (6.23)$$

Über die Gleichung 6.23 kann die Isolierung des Ausgangsvektors \underline{y} erreicht werden:

$$\begin{aligned} \underline{y}(t) &= C\underline{x}(t) + DV\underline{w} - DT\underline{y}(t) \\ \Rightarrow \underline{y}(t) + DT\underline{y}(t) &= C\underline{x}(t) + DV\underline{w} \\ \Rightarrow (E + DT) \cdot \underline{y}(t) &= C\underline{x}(t) + DV\underline{w} \\ \Rightarrow \underline{y}(t) &= (E + DT)^{-1} \cdot (C\underline{x}(t) + DV\underline{w}) \end{aligned} \tag{6.24}$$

Durch die Zusammenführung der Ausgangsgleichung 6.22 und der Zustandsgleichung 6.24 resultiert die allgemeine Systemgleichung:

$$\begin{aligned} \underline{x}'(t) &= A\underline{x}(t) + B \cdot (V\underline{w} - T \cdot (E + DT)^{-1} \cdot (C\underline{x}(t) + DV\underline{w})) \\ &= (A - BT \cdot (E + DT)^{-1} \cdot C) \cdot \underline{x}(t) + B \cdot (E - T \cdot (E + DT)^{-1} D) \cdot V\underline{w} \end{aligned} \tag{6.25}$$

Für das Produkt einer quadratischen Matrix A mit deren inverser Matrix ist die Kommutativität der Multiplikation gegeben. Ebenso für die Multiplikation einer beliebigen (in diesem Fall quadratischen) Matrix mit der Einheitsmatrix. Hieraus resultiert die Identität:

$$E - T \cdot (E + DT)^{-1} \cdot D = (E + TD)^{-1} \tag{6.26}$$

Angewendet auf die Gleichungen 6.25 und 6.24 resultiert jeweils:

$$\underline{x}'(t) = (A - BT \cdot (E + DT)^{-1} \cdot C) \cdot \underline{x} + B \cdot (E + TD)^{-1} \cdot V\underline{w} \tag{6.27}$$

$$\underline{y} = (E + DT)^{-1} \cdot C\underline{x} + (E + DT)^{-1} \cdot DV\underline{w} \tag{6.28}$$

Über einen Koeffizientenvergleich mit der Zustandsgleichung 6.22 des Zustandsmodells und der zugehörigen Ausgangsgleichung 6.23 resultieren die Gesetzmässigkeiten 6.29 bis 6.32 für das System mit Regelungsrückführung.

$$A_{RA} = A - BT \cdot (E + DT)^{-1} \cdot C \quad (6.29)$$

$$B_{RA} = B(E + TD)^{-1} \cdot V \quad (6.30)$$

$$C_{RA} = (E + DT)^{-1} \cdot C \quad (6.31)$$

$$D_{RA} = (E + DT)^{-1} \cdot DV \quad (6.32)$$

Für die Beschreibung der Stabilität des Systems entsteht die Charakteristische Gleichung:

$$P(s) = |sE - (A - BT \cdot (E + DT)^{-1} \cdot C)| = 0 \quad (6.33)$$

Aus dieser entsteht nach dem Laplace'schen Entwicklungssatz [$\det(A) = \sum_{i=1}^n (-1)^{i+j} \cdot a_{ij} \cdot \det(A_{ij})$] das charakteristische Polynom 6.34, aus welchem sich die Stabilitätsbedingungen für das System ablesen lassen.

$$\sum_{i=0}^n s^i \cdot mdc = 0 \quad (6.34)$$

Für einen gegebenen Parametersatz der Priorität und Eigendynamik (m_k, d_k, c_k) aller Knoten und einer bestimmten Topologie T sind die Nullstellen und somit die Stabilität für ein festgelegtes Netzwerk gegeben. Die Bestimmung der Nullstellen erfolgt durch das Polynom 6.34 und lässt sich im allgemeinen Fall nur numerisch lösen. Die Dimension lässt sich mit nk angeben, wobei n die Anzahl unterschiedlicher Zustandsvariablen der Regelung (in jedem Netzwerk unabhängig der Topologie genau 3) und k die Anzahl der Knoten innerhalb des Netzwerks darstellt. Es ist zu erkennen, dass selbst bei kleinen Netzwerken die Dimension des Polynoms Ausmaße annimmt, bei denen eine Lösung nicht allgemein zu bestimmen ist.

Die resultierenden Nullstellen liegen stets als konjugiert komplexe Paare vor. In einem stabilen System weisen diese einen negativen Realteil auf und ordnen sich somit in der linken Hälfte der komplexen Ebene ein (vgl. Abbildung 6.3 grauer Bereich). Die Position des Realanteils auf der entsprechenden Achse bestimmt die Geschwindigkeit mit der eine Anpassung auf den Sollwert erfolgt. Diese wird einerseits durch die Umsetzbarkeit (Registerbreite, Begrenzungen durch Hardwaretimer, etc.) und das maximal tolerierte Überschwingverhalten beschränkt. Die Schwingungsbereitschaft des Systems wird durch den Betrag des Imaginäranteils bestimmt. Der Kompromiss aus beiden bestimmt die Leistungsfähigkeit des Regelungsalgorithmus und somit die Güte der Synchronisation. Ein Algorithmus mit hoher Güte erreicht nach geringer Zeit einen stabilen Ausgleich unter den Knoten, ist jedoch träge genug Störungseinflüsse ohne Beeinträchtigung der Nachbarn zu absorbieren.

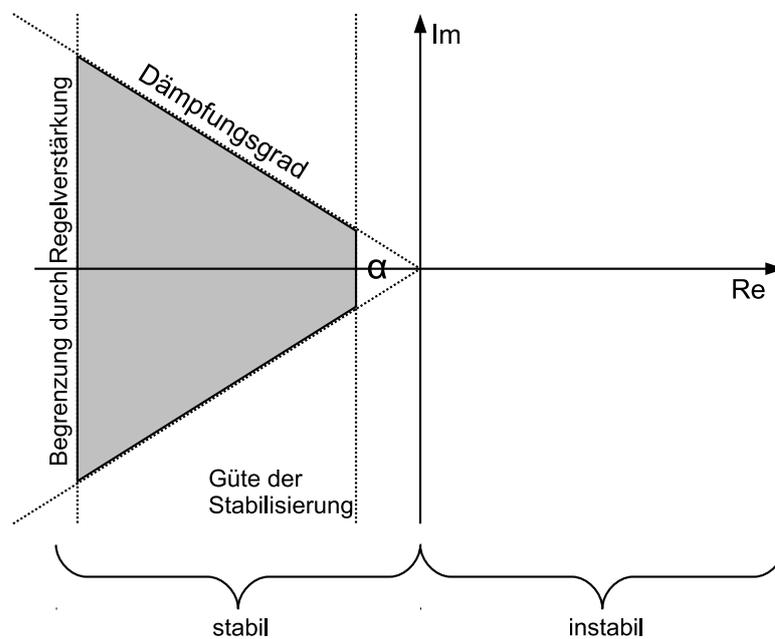


Abbildung 6.3: Nullstellendiagramm

Es ist somit erstrebenswert, Parametersätze für ein sich synchronisierendes System entsprechend zu wählen, dass eine möglichst keine Fläche durch die Verteilung der konjugiert komplexen Nullstellen innerhalb des Nullstellendiagramms aufgespannt wird. Dieser Bereich wird von Tangenten, welche sich im Ursprung treffen eingeschlossen und bestimmen den Winkel α . Je kleiner α für ein gegebenes Netzwerk durch die Wahl günstiger Parametersätze

erreicht werden kann, desto schneller konvergiert das System (d.h. desto schneller erreicht das System einen synchronisierten Zustand und zeigt Stabilität gegenüber dem Einfluss von Störgrößen).

In Vorarbeiten konnte der Stabilisierungsnachweis an unterschiedlichen Topologien erfolgreich angewendet werden (vgl. Müller (2017b)). Es wurden exemplarisch Topologien vom Typ Ring, Stern und Daisy-Chain untersucht. Hierbei konnte gezeigt werden, dass für einen vordefinierten Parametersatz eine Stabilität des Netzes in allen Fällen erlangt werden konnte. Es wurde weiterhin gezeigt, dass ein für eine Topologie speziell hin optimierter Parametersatz an einer anderen Topologie zwar auch ein konvergentes Verhalten verursachen kann, jedoch nicht die optimalen verhaltensweisen im bezug auf synchronisationsgeschwindigkeit und störkompensation zeigt. somit ist in einem optimiertes system für jede topologie speziell ein parametersatz zu ermitteln, welcher die gewünschten eigenschaften in der Zieltopologie zur auslösung bringt. Weiterhin steigt der Berechnungsaufwand für den Konvergenzbeweis eines Netzwerks exponentiell mit der Anzahl der Knoten, was aus einem stark anwachsenden charakteristischen Polynom selbst für kleine Netze resultiert. Die Lösung dieser Polynome ist mit wirtschaftlichem Aufwand nur maschinell möglich.

7 Diskretisierung des Modells

Im Kapitel 6 auf Seite 26 konnte dargelegt werden, dass eine rollenlose Synchronisation eines Netzwerks möglich ist und unterstützt somit das Ziel diesen Synchronisationsmechanismus für Netzwerktopologien einzusetzen. Dieser Nachweis wurde jedoch unter theoretischen Bedingungen erbracht. Diese würden ein Netzwerk mit einer unendlich schnellen Informationsverbreitung und einer ebenfalls gegen unendlich strebenden Aktualisierungsrate fordern. Da sich diese Annahmen in der Praxis nicht realisieren lassen, ist eine Überführung des in der Arbeit dargelegten Konzeptes in ein verzögerungszeitbehaftetes System mit dedizierten Ereignissen (wie dem Empfangen und Versenden von Nachrichten) notwendig. Dieser Abschnitt soll mögliche Umformungsvarianten untersuchen, effektive Methoden klassifizieren und umsetzen, sowie diese Umsetzung anhand eines MATLAB-Modells darstellen und validieren. Innerhalb des Modells können anschließend beliebige Topologien betrachtet werden, unter anderem Ringe, welche in anderen Synchronisationsprozessen destruktive Timing- bzw. Synchronisation-loops hervorrufen können (vgl. Tan (2007) und Union (2015)).

Synchronisationsprotokolle wie das IEEE 1588 oder das Synchronous Ethernet erfordern für den fehlerfreien Betrieb eine ringfreie Topologie, da andernfalls die Synchronisation innerhalb des betroffenen Zweiges zu kollabieren droht (vgl. Tan (2007) und Union (2015)). Tritt im Netzwerk ein Fehler auf, welcher die Topologie beeinflusst kann es im ungünstigsten Fall zur Bildung von Synchronisationsschleifen kommen, durch welche selbst Teile des Netzwerks von Synchronisationsstörungen beeinflusst werden können, die andernfalls z.B. physikalisch nicht unmittelbar durch den auftretenden Fehler betroffen wären. Es ist daher eine genaue Planung der Synchronisationsarchitektur vonnöten, um eine Absicherung des Netzwerks gegenüber Fehlerfällen zu erreichen. Der in dieser Arbeit diskutierte Ansatz erlaubt richtungslose enge Vermaschung und ist somit ein Kandidat für den Einsatz in Echtzeitanwendungen, welche auf Fehlerfälle reagieren und Knoten- sowie Linkausfälle kompensieren. Es ist zu zeigen, dass Planung und Konfiguration dieser Netze im Peer-to-Peer-Verfahren durch die stabilitätsfördernden Eigenschaften durch Vermaschung begünstigt und vereinfacht wird.

7.1 Transformation: Eigenschaften und Grenzen

Die Systemdefinition stellt das Verhalten eines Knotens dar und beischreibt sich durch eine Menge von Funktionen. Im Folgenden werden Umformungsstrategien analysiert und unter der Berücksichtigung der Eigenschaften des zugrunde liegenden Modells diskutiert.

Als Vorbereitung für die Transformation der in der Vorarbeit (Müller (2017b)) erarbeiteten Beschreibungsform eines Knotens sind zunächst die einzelnen Stufen des Ansatzes gegenüber der Transformation zu untersuchen. Dieser setzt sich aus den drei Stufen Informationsbeschaffung, Gewichtungsfunktion und Übertragungsfunktion zusammen (vgl. Kapitel 5 auf Seite 19). In der ersten Stufe erfolgt die Informationsaggregation über den Transfer von Nachrichten. Es werden nach Möglichkeit von jedem Peer vier Zeitinformationen gesammelt aus welchen die momentane Uhrenabweichung des lokalen Zeitgebers zum jeweiligen Peer ermittelt wird. Die gesammelten Zeitinformationen über alle Nachbarn wird anschliessend über eine lineare Funktion gewichtet und stellt die zweite Stufe des Ansatzes dar. Die Gewichtungsfunktion ermöglicht dem Knoten den äusseren Einfluss durch seine Nachbarn zu kontrollieren. Mit einem entsprechenden Regelwerk kann somit anhand unterschiedlicher Faktoren ein Knoten auf fehlerbehaftete Einflüsse von aussen reagieren und z.B. dem Problem des ‚Babbling Idiot‘ entgegenwirken. Diese Kontrolle ermöglicht je nach Regelwerk und erkanntem Verhalten eine beliebige Verstärkung oder Abgeschwächung des Informationsflusses eines jeden Nachbarn. Weiterhin stellt die Gewichtungsfunktion fest, wie viele Informationen von einem Knoten empfangen wurden und stellt die Menge der Information in Relation zur Information selbst. Dieses Verfahren stellt, ähnlich einer Normierung, die Vergleichbarkeit zwischen allen Informationslieferanten sicher und verhindert somit eine Bevorzugung von Knoten, welche öfter ihre Informationen propagieren. Die letzte Stufe besteht im Regelungsverhalten des Knotens und bildet die Reaktion des Knotens dar, nachdem eine valide, normierte und anschliessend gewichtete Abweichung zum Netzwerk festgestellt wurde. Ein aus einem physikalischen Modell abgeleiteter systembeschreibender Parametersatz (bestehend aus Masse m , Federkonstante c und der Dämpfungskonstante d) definiert hierbei das Verhalten und lässt den Knoten unter anderem Reaktionsfreudig oder Träge auf äussere Einflüsse reagieren.

7.1.1 Strategie der Transformation

Da die ersten beiden Stufen der Informationsbeschaffung und Gewichtungsfunktion über eine Menge algebraischer Funktionen gebildet werden, können Berechnungen unmittelbar durchgeführt werden, sollten neue Informationen vorliegen. Ein Kommunikationsnetzwerk stellt

über sein Nachrichtenfluss eine ereignisgetriebene Umgebung dar. Durch die Einteilung des Ablaufs in dedizierte Empfangs- und Sendezeitpunkte werden neue Zeitinformationen der einzelnen Peers zu einem beliebigen aber dedizierten Zeitpunkt empfangen. Aufgrund des linearen Verhaltens der ersten beiden Stufen bewirkt der Ablauf innerhalb eines Netzwerks somit eine Zeitdiskretisierung womit eine spezielle Umformung an dieser Stelle nicht vonnöten ist. Die letzte Stufe stellt die systembeschreibende Übertragungsfunktion des lokalen Knotens dar, welche aufgrund der vorangegangenen Systembetrachtung im Kontinuierlichen als Differentialgleichung vorliegt. Sollte keine Umformung erfolgen muss die Berechnung dieser Gleichung analytisch auf dem lokalen Knoten mithilfe eines iterativen rechenintensiven Annäherungsverfahrens erfolgen. Um zu jedem Zyklus neue Zeitinformationen für die Synchronisationsroutine zur Verfügung stellen zu können ist es notwendig während jedes Zyklusses die Annäherung erneut durchzuführen. Für jede Annäherung muss eine Schrittweite und eine Abbruchbedingung festgelegt werden, mit denen die Iterationen durchgeführt werden, welche das Verhalten des Knotens beeinflussen und damit beschrieben und untersucht werden müssen. Daher ist eine Umformung der Systembeschreibenden Übertragungsfunktion in eine diskrete Beschreibungsform notwendig, welche vom Knoten für jede Periode direkt gelöst werden kann

Es konnte in der Vorarbeit gezeigt werden, dass ein System aus Knoten beliebiger Topologie, welche nach dem Konzept der Peer-to-Peer-Synchronisation handeln als Mehrgrößensystem beschrieben werden kann. Somit können Konzepte der Regelungstechnik für die Überführung der Übertragungsfunktion in eine diskrete Beschreibungsform herangezogen werden. Bei der Systembeschreibung der Differentialgleichung handelt es sich um ein theoretisches Modell, womit selbst Ansätze herangezogen werden können, welche in der Regelungstechnik zwar als theoretisches Konzept betrachtet werden, jedoch nicht für den realen Einsatz bestimmt sind. Sie basieren auf Forderungen wie z.B. einer idealen Abtastung eines Signals oder der idealen Rekonstruktion des Ausgangssignals, welche für ein reales physikalisches System unerreichbar sind. Da der Synchronisationsansatz nur nach außen hin ein regelndes System darstellt ist nur wichtig, dass alle Zustandsgrößen in ihrem Wertebereich beschränkt sind, um die Umsetzbarkeit des Algorithmus zu gewährleisten.

7.2 Systemdarstellung im Z-Bereich

Differenzgleichungen beschreiben das Verhalten zeitdiskreter Systeme ähnlich wie Differentialgleichungen das Verhalten zeitkontinuierlicher Systeme beschreiben. Wie für Differentialgleichungen der Laplace-Bildbereich für den algebraischen Umgang mit Differentialgleichungen

genutzt werden kann, so kann der Z-Bereich für den Umgang mit Differenzgleichungen eingesetzt werden. Differenzgleichungen können zwar im Zeitbereich gelöst werden, jedoch ist das Verfahren dazu sehr aufwendig. Es besteht aus mehreren Iterationsstufen und steigt an Komplexität mit der Zunahme von Anfangsbedingungen (vgl. Nelles (2015)). Eine effizientere Möglichkeit zur Lösung von Differenzgleichungen kann mit der Z-Transformation erreicht werden. Das Vorgehen wird durch die Abbildung 7.1 dargestellt.

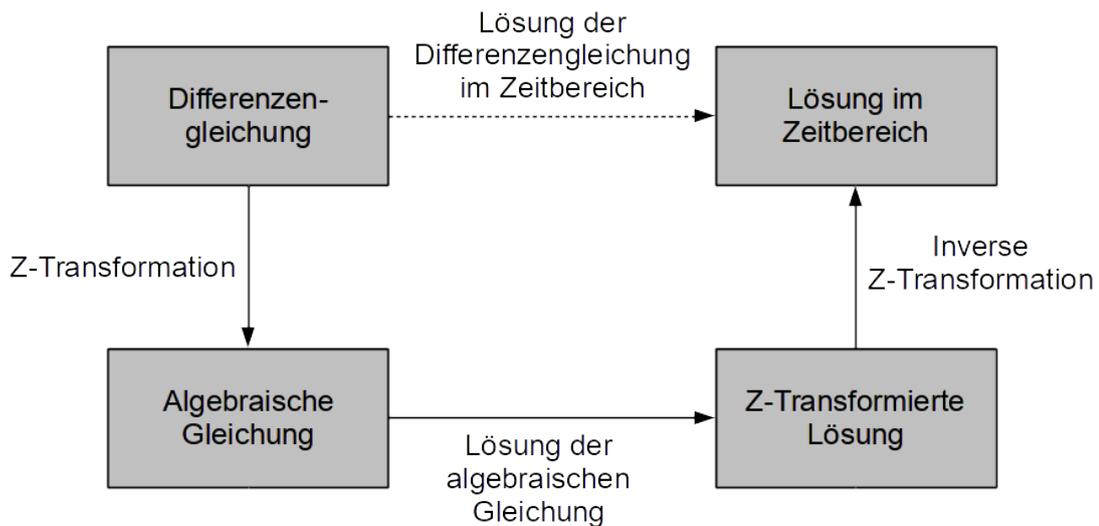


Abbildung 7.1: Lösungsverfahren für Differenzgleichungen

Die Differenzgleichung wird mithilfe der Z-Transformation in den Z-Bereich transformiert. Aufgrund des Aufbaus der Z-Transformation ergibt sich eine algebraische Gleichung, die im Z-Bereich gelöst wird. Mithilfe der Rücktransformation kann anschliessend die Lösung im Zeitbereich bestimmt werden.

7.3 Diskretisierung des Systems

Für die Darstellung der diskreten Übertragungsfunktion als Differenzgleichung ist die Transformation der Übertragungsfunktion eines Knotens in den Z-Bereich notwendig. Für die Transformation werden unterschiedliche Strategien, je nach den Eigenschaften der umzuwandelnden Systemfunktion, herangezogen.

Eine direkte Variante der Umwandlung des kontinuierlichen Systems in ein diskretes kann über den Einsatz von Integrationsverfahren erreicht werden. Ausgangspunkt ist die Darstellung

des Systems im Zustandsraum. Von zentraler Bedeutung ist die Systemmatrix, aus der die Eigenwerte und damit die Systemdynamik und dessen Stabilität abgeleitet wurde. Aufgrund des besonderen Aufbaus der Durchgriffsmatrix ($D = \emptyset$) des zu betrachtenden Systems führt dieses Umwandlungsverfahren zu einer algebraischen Schleife und kann somit nicht als Realisierung umgesetzt werden.

Der Entwurf eines zeitdiskreten Systems kann zudem aus der Umwandlung der Übertragungsfunktion eines zeitkontinuierlichen Systems $G(s)$ in eine Übertragungsfunktion des zeitdiskreten Systems erfolgen. Das Verfahren ist eine bilineare Transformation. Es wird insbesondere zur Entwicklung rekursiver Filter eingesetzt, welche sich durch eine unendliche Impulsantwort auszeichnen. Obwohl es sich bei dem umzuwandelnden System um ein logisches Konzept handelt, welches über eine unendliche und physikalisch nicht realisierbare Sprungantwort verfügen könnte, so ist es dennoch aus einem physikalischen Konzept abgeleitet und ist somit nicht ohne Anpassungen über diese Methode zu transformieren. Für eine angepasste Übertragungsfunktion kann die durchgeführte Stabilitätsbetrachtung nicht mehr als Grundlage herangezogen werden und müsste somit erneut untersucht werden.

Eine weitere Möglichkeit der Transformation ergibt sich aus der diskretisierten Nachbildung einer kontinuierlichen Systemantwort. Dieser Ansatz hat zum Vorteil, dass auf bestehenden Stabilitätsbetrachtungen eines kontinuierlichen Systems aufgebaut werden kann (vgl. [Lunze \(2013\)](#)). Da sich eine Systemantwort aus der Impuls- oder Sprungantwort des Systems heraus definieren lässt, ist die zeitdiskrete Nachbildung einer der beiden Antworten notwendig. Wird die Impulsantwort des zeitkontinuierlichen Systems zeitdiskret nachgebildet, so ist das zeitdiskrete System so zu entwerfen, dass dessen Impulsantwort $g[k]$ an den Stellen $t = k \cdot T_A$ dieselben Werte aufweist, wie die zeitkontinuierliche Impulsantwort $g(t)$. Analog kann das System so nachgebildet werden, dass die Sprungantworten von zeitdiskretem und zeitkontinuierlichem System an den Stellen $t = k \cdot T_A$ dieselben Werte aufweisen. Diese Verfahren werden in der Regelungstechnik als impuls- bzw. sprunginvarianter Entwurf zeitdiskreter Systeme bezeichnet. Im Folgenden wird die Diskretisierung des Systemverhaltens durch Nachbildung der Systemantwort betrachtet.

7.3.1 Impulsinvariante Transformation

Die zeitdiskrete Übertragungsfunktion lässt sich über die Abtastung der Impulsantwort $g(t)$ des entsprechenden zeitkontinuierlichen Systems transformieren. Hierfür ist es notwendig, dass das Ursprungssystem kausal und stabil ist. Die Kausalität des Systems (vgl. Gleichung [6.4](#)

auf Seite 27) lässt sich direkt an der Zustandsdarstellung des Systems ablesen. Ebenso wurde in der Vorarbeit das stabile Verhalten der Übertragungsfunktion gezeigt. Das Ursprungssystem besitzt über dies hinaus ausschliesslich einfache reelle Polstellen und konjugiert komplexe Polpaare. Da mit diesen Voraussetzungen der Zählergrad der Übertragungsfunktion kleiner dem Nennergrad ist, lässt sich die Übertragungsfunktion als Summe einzelner Partialbrüche darstellen und entspricht einem realisierbaren System:

$$G(s) = A_0 + \sum_{n=1}^N \frac{A_n}{s - \alpha_n} \quad (7.1)$$

Diese Form beschreibt die Übertragungsfunktion über einen konstanten Anteil A_0 , sowie Gewichtete Anteile A_n über die jeweiligen Polstellen α_n der Funktion. Die Übertragungsfunktion ist eine Transformation einer Zeitfunktion in den laplaceschen Raum. Wird die Gleichung (7.1) aus dem laplaceschen Raum zurück in den Zeitbereich transformiert, folgt:

$$g(t) = A_0 \cdot \delta(t) + \sum_{n=1}^N A_n \cdot e^{\alpha_n \cdot t} \cdot \sigma(t) \quad (7.2)$$

Aus der Beschreibung zur Rücktransformation resultieren hierbei die zeitabhängigen Parameter des konstanten Anteils $\delta(t)$ und die der gewichteten Anteile $\sigma(t)$, sowie die laplacesche Erweiterung $e^{\alpha_n \cdot t}$. Die Impulsantwort eines zeitdiskreten Systems definiert sich dadurch, dass sie an den Stellen $t = k \cdot T_A$ dieselben Werte erzeugt, wie die entsprechende zeitkontinuierliche Funktion. Damit ergibt sich die zeitdiskrete Impulsantwort:

$$\begin{aligned} g[k] &= g(k \cdot T_A) \\ &= A_0 \cdot \delta(k \cdot T_A) + \sum_{n=1}^N A_n \cdot e^{\alpha_n \cdot k \cdot T_A} \cdot \sigma(k \cdot T_A) \end{aligned} \quad (7.3)$$

Über die Schrittweite T_A ist für alle Schritte k mithilfe der laplaceschen Erweiterung die Übertragungsfunktion im Zeitbereich definiert. Zur Darstellung als Differenzengleichung ist eine

Projektion der impulsinvarianten Funktion in den Z-Bereich notwendig. Die z-Transformierte $G(z)$ der Impulsantwort $g[k]$ berechnet sich über die Definitionsgleichung der z-Transformation (vgl. Gleichung 7.4) zu:

$$G(z) = \sum_{k=0}^{\infty} g[k] \cdot z^{-k} \quad (7.4)$$

$$\begin{aligned} G(z) &= \sum_{k=0}^{\infty} (A_0 \cdot \delta(k \cdot T_A) + \sum_{n=1}^N A_n \cdot e^{\alpha_n \cdot k \cdot T_A} \cdot \sigma(k \cdot T_A)) \cdot z^{-k} \\ &= A_0 + \sum_{k=0}^{\infty} \sum_{n=1}^N A_n \cdot e^{\alpha_n \cdot k \cdot T_A} \cdot z^{-k} \end{aligned} \quad (7.5)$$

Es entsteht eine geometrische Reihe, welche sich mithilfe der Vereinfachungsgesetze und Anwendung der Summenformel wie folgt zusammenfassen lässt:

$$\begin{aligned} G(z) &= A_0 + \sum_{n=1}^N A_n \cdot \sum_{k=1}^{\infty} (e^{\alpha_n \cdot T_A} \cdot z^{-1})^k \\ &= A_0 + \sum_{n=1}^N A_n \cdot \frac{z}{z - e^{\alpha_n \cdot T_A}} \end{aligned} \quad (7.6)$$

Die so gewonnene spezielle Form (vgl. Gleichung 7.6) ist zur Anwendung auf die für das System entwickelte kontinuierliche Übertragungsfunktion (6.4) ausgelegt. Sie kann zur Erzeugung entsprechender Differenzgleichungen herangezogen werden und dient somit zur Verhaltensbeschreibung eines jeden Peers in einem Netzwerk. Im Folgenden wird die Umwandlung für das kontinuierliche System vorgenommen.

7.3.2 Diskretisierung im speziellen Fall

Eine Differenzgleichung, welche das Verhalten eines bestimmten Systems abbildet, kann aus einer entsprechenden diskreten Übertragungsfunktion gewonnen werden. Zur Wandlung der in der Vorarbeit ermittelten kontinuierlichen Systembeschreibung (6.4) in die diskrete Form

(7.6) des Kapitels 7.3.1 ist es notwendig einen Parametersatz $s = \{m, c, d\}$ festzulegen, welcher die Basis für das Verhalten des Knotens bildet. Diese Parametersätze sind auf Stabilität hin zu untersuchen und ggf. für die gewünschte Topologie zu optimieren.

Im Folgenden wird als Verhaltensbeschreibung der heuristisch bestimmte Parametersatz $s = \{1, 20, 20\}$ gewählt, welcher für die spezielle Übertragungsfunktion ein regelungsstabiles System erzeugt. Durch die heuristische Bestimmung ist der Parametersatz nicht für eine spezielle Topologie hin optimiert, lässt sich jedoch für eine Vielzahl unterschiedlicher Topologien als stabiles Mehrgrößensystem verwenden wobei die lokalen Zeitinformationen jedes Peers des Netzwerks zu einem globalen Verständnis der Zeit konvergieren. Dies schafft eine Vergleichbarkeit unterschiedlicher Topologien bei gleichem Verhaltensmuster ihrer einzelnen Peers. Über den festgelegten Parametersatz folgt somit die Übertragungsfunktion eines Peers:

$$G(s) = \frac{1}{\frac{1}{20}s^2 + s + 1} \quad (7.7)$$

Das System kann anschliessend als eine Summe ihrer Partialbrüche dargestellt werden. Es resultiert die Übertragungsfunktion eines zunächst noch kontinuierlichen Systems:

$$G(s) = \frac{1,118}{s + 1,056} - \frac{1,118}{s + 18,944} \quad (7.8)$$

Durch Koeffizientenvergleich mit der Gleichung 7.1 auf Seite 46 ergeben sich die Koeffizienten A_n und α_n wie folgt:

$$\begin{aligned} A_0 &= 0 \\ A_1 &= 1,11803 \\ A_2 &= -1,11803 \\ \alpha_1 &= -1,05573 \\ \alpha_2 &= -18,9443 \end{aligned} \quad (7.9)$$

Nach Anwendung der speziellen Form 7.6 auf Seite 47 kann nun die systembeschreibende Funktion des zeitdiskreten Systems gebildet werden:

$$G(z) = 1,118 \cdot \frac{z}{z - e^{-1,056 \cdot T_A}} - 1,118 \cdot \frac{z}{z - e^{-18,944 \cdot T_A}} \quad (7.10)$$

Für die Schrittweite $T_a = 1$ folgt somit die Übertragungsfunktion des Systems:

$$G(z) = \frac{0,389z}{z^2 - 0,348z + 2,06 \cdot 10^{-9}} \quad (7.11)$$

Diese Übertragungsfunktion lässt sich jedoch nicht direkt auf einem System realisieren, da sie eine Abbildung des Z-Bereichs darstellt. Es ist somit notwendig die Funktion (9.3) durch Rücktransformation nach ihrer Definition $G(z) = \frac{Y(z)}{U(z)}$ in den Zeitbereich zu übertragen. Die somit entstandene Differenzgleichung lässt sich nach ihren Ausgangs-, sowie Eingangsgrößen sortieren (vgl. Gleichung 7.13) und anschliessend nach dem Ausgang $y[k]$ umstellen.

$$G(z) = \frac{0,389z^{-1}}{1 - 0,348z^{-1} + 2,06 \cdot 10^{-9}z^{-2}} \quad (7.12)$$

$$\Rightarrow Y(z) \cdot (1 - 0,348z^{-1} + 2,061z^{-2} \cdot 10^{-9}z^{-2}) = U(z) \cdot 0,389z^{-1} \quad (7.13)$$

$$\Rightarrow y[k] = 0,348 \cdot y[k-1] - 2,061 \cdot 10^{-9} \cdot y[k-2] + 0,389 \cdot u[k-1] \quad (7.14)$$

Die Gleichung (9.5) beschreibt nun das Verhalten am Ausgang eines Systems vollständig und bildet eine direkte Abhängigkeit zwischen dem Ausgang und allen notwendigen Eingangsgrößen ab. Hierbei sind, begründet durch den induktiven Aufbau der Differenzenfunktion, Ausgangsgrößen vergangener Zyklen in die Berechnung der aktuellen Ausgangsgröße $y[k]$ mit einzubeziehen.

Die Differenzgleichung ist eine implementier- und simulierbare Verhaltensweise eines Knotens in einem Netzwerk und besteht ausschliesslich aus einfachen algebraischen Verfahren. Ein Knoten kann somit die Zeitinformationen seiner Nachbarn in die Berechnung der eigenen

lokalen Zeit einfließen lassen und anschliessend die eigene Zeit im Netzwerk propagieren, ohne Rechenaufwand in die Lösung (bzw. Annäherung) von Differenzialgleichungen investieren zu müssen.

In aufbauenden Arbeiten (vgl. Müller (2017a)) konnte eine spezielle Umsetzung eines Knotens über den Verlauf seiner Sprungantwort ausgewertet werden. Hierzu wurde die kontinuierliche Systembeschreibung der diskreten Umsetzung gegenübergestellt. Die kontinuierliche Übertragungsfunktion wurde simuliert, während die diskrete Verhaltensweise des Knotens durch die Differenzgleichung einer expliziten Realisierung entsprach. Würde das kontinuierliche System ebenfalls explizit realisiert so müsste dies durch numerische Verfahren umgesetzt werden, welche den Signalverlauf aufgrund des Einsatzes zusätzliche Parameter wie Schrittweite und Abbruchbedingen (bedingt durch die Anwendung entsprechender Annäherungsverfahren) beeinflussen und somit den Vergleich beeinträchtigen könnten.

Die Implementation wurde durch einen Sprung definierter Höhe zu einem bestimmten Zeitpunkt angeregt und das Verhalten der Implementation während dieser Zeit aufgezeichnet. Die Untersuchung ist vergleichbar mit einem Knoten, welcher sprunghaft neue Zeitinformationen (z.B. durch Einkoppeln in ein bestehendes Netzwerk) erhält. Dabei wurde jedoch nur eine unidirektionale Kommunikation gegeben womit keine Einflüsse durch Rückkopplung in die Betrachtung einfließen. Es hatte sich gezeigt, dass das diskrete System in seinem Verhalten seinem Vorbild entspricht. Der für die Umwandlung in dieser Arbeit eingesetzte Parametersatz $s = \{1, 20, 20\}$ erzeugt ein Überschwingen, erreicht jedoch nach 12 Zyklen einen Ausgleich der Abweichung von über 80% und nach 22 Zyklen eine Abweichung von $< 96\%$. Ab ca. Zyklus 40 war die Restabweichung klein genug um von den systemeigenen Ungenauigkeiten (z.B. Rundungsungenauigkeiten der Implementierung und Schwankungen in der Frequenz des Quarzes) nicht mehr differenzierbar zu sein, womit der Vorgang der Korrektur als abgeschlossen betrachtet werden kann. Durch Optimierung der Parameter an eine spezielle Topologie kann die Anzahl der Zyklen, sowie das Überschwingen deutlich reduziert werden.

8 Anomalien der rollenlosen Synchronisation

Im Rahmen dieser Arbeit wurde eine mathematische Simulation der Peer-to-Peer-Simulation unter MATLAB-Simulink erstellt um Schwingungsverhalten, Parametrisierung und funktionale Abhängigkeiten studieren zu können. Hierzu wurde ein skalierbares Modell erstellt, sodass es das Verhalten auch für große Netze und beliebig komplexen Topologien zur Laufzeit nachbilden konnte. An ihm konnten alle, für einen Peer-to-Peer-basierenden Ansatz, relevanten Parameter abgeleitet und untersucht werden, indem kontinuierliche und diskrete Konzepte in den Vergleich gingen. Zudem konnte ein Regelwerk über die Abhängigkeiten der Parameter zueinander aufgestellt werden, was die Konfiguration eines solchen Netzes für unterschiedliche Topologien in Hinsicht auf Synchronisationsgeschwindigkeit und Stabilität hin optimiert.

Untersuchungen zeigten eine Reihe von Anomalien auf. Eine Anomalie bezeichnet den unerwarteten Zustand eines Netzwerkes durch reguläre Einflüsse, begründet durch seine Synchronisation über eine Peer-to-Peer-Architektur. Diese Anomalien stellen insbesondere ein Problem dar, insofern sie das Netzwerk im gesamten (oder zumindest partiell) in einen nichtoperativen Zustand überführen können. Hierbei bleibt das System über die gesamte Laufzeit hinweg regelungstechnisch stabil. Jedoch ist es Architekturgebunden möglich durch eine Vernetzung über viele Hops, dem Hinzufügen einzelner Knoten mit bestimmter Priorität in Sterntopologien oder auch nach einer langen Laufzeit eines Netzes, dass das verteilte System in einen vorübergehend unsynchronisierten Zustand übergeht, in dem nicht mehr ausreichend das Verständnis der globalen Zeit propagiert werden konnte.

8.1 Nachbarschaftsabweichung

Abbildung 8.1 auf der nächsten Seite zeigt eine Anomalie, entstanden durch die Überwindung vieler Hops. Die Achsen x und y spannen hierbei eine Ebene auf, die lediglich der topologischen Darstellung des Netzwerks dient. Knoten des Netzwerks sind hierbei grüne oder rote Kugeln.

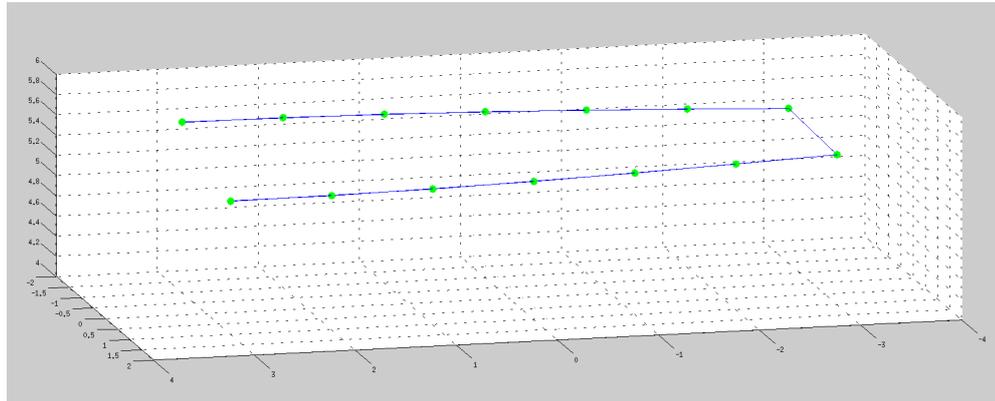


Abbildung 8.1: Hop-Count-Anomaly

Der Link zwischen den Knoten ist als blaue Verbindung dargestellt. Alle im weiteren Verlauf beschriebenen Verbindungen sind bidirektional und ermöglichen den Austausch des Zeitverständnis der Knoten untereinander. Der z-Vektor (im Bild nach oben) stellt die Geschwindigkeit dar, mit der ein Knoten seine lokale Uhr betreibt. Je größer z für einen Knoten, desto mehr Zyklen laufen lokal im Verhältnis zur Zeit des externen Beobachters (der Simulationszeit).

Erzeugt wurde die Anomalie indem zwei lineare Netze simuliert wurden, welche von einander getrennt, sich über die Zeit hinweg in den synchronisierten Zustand überführten. Anschließend wurde ein Link zwischen diesen gleichstarken Netzen erzeugt, welcher eine netzwerkweite Synchronisation erzwingt. Beide Endknoten (links im Bild) des neu entstandenen Netzes betrachten sich zu ihren jeweiligen Nachbarn als synchron (grüne Färbung) obwohl sie zueinander betrachtet nicht als synchron eingestuft werden können (ersichtlich durch eine hohe Differenz in z -Richtung). Sind beide Knoten Teil einer verteilten Anwendung, so existiert zwischen ihnen kein genügend bestimmtes Zeitverhalten, womit es zum Fehler kommt, welcher von den jeweiligen Knoten zudem nicht selbstständig erkannt werden kann.

Diese Art der Anomalie konnte in der Arbeit nicht kompensiert werden. Eine Möglichkeit den Effekten entgegenzuwirken besteht darin die Anzahl der Hops zu minimieren um eine maximale Abweichung einhalten zu können. Da die mögliche Abweichung zwischen den Knoten linear in Abhängigkeit zur Priorität anwächst ist die maximale Abweichung, welche das Netz mit einer gegebenen Topologie erfahren kann, zu ermitteln. Ebenso können zusätzliche direkte Verbindungen die Abweichung eines Netzwerkteilnehmers weiter reduzieren. Hier gilt, je stärker die Vermaschung der Knoten untereinander, desto geringer der Einfluss dieser Anomalie.

8.2 Gruppenauflösung

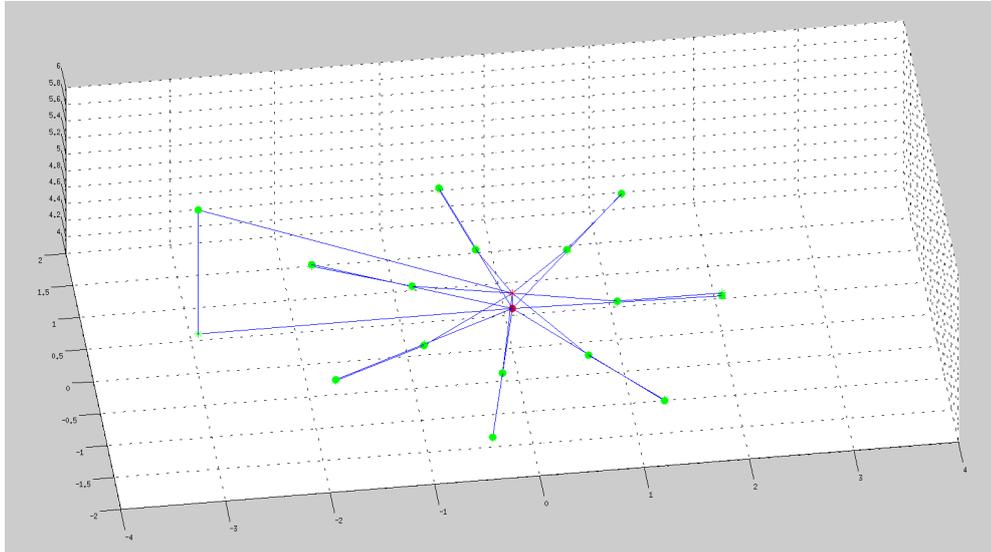


Abbildung 8.2: Priority-Anomaly

Eine weitere Anomalie konnte durch Simulation von dynamischen Charakteristika der Peer-to-Peer-Struktur gezeigt werden und wird in [Abbildung 9.3 auf Seite 62](#) dargestellt. Hier wurde einer synchronisierten Sterntopologie ein Knoten (links im Bild) mit einer bestimmten Gewichtung zur Laufzeit hinzugefügt. Zum Zwecke der Demonstration wurde der Link zwischen dem neuen Knoten und dem Zentrum des Sterns unidirektional gewählt, sodass der Stern eine Anpassung erfährt, der externe Knoten jedoch nicht. Durch den massiven Einfluss geht der zentrale Knoten der Sterntopologie in den unsynchronisierten Zustand über (rote Färbung). Die dynamische Kopplung des Knotens stört somit die Funktion des Sterns, was zwar lokal auf dem zentralen Knoten des Sterns erkannt wird, jedoch eine Gefahr für Peer-to-Peer-Lösungen in offenen Systemen mit echtzeitfähigen Eigenschaften darstellt. Ein ähnliches Szenario zeigt sich, bei der Kopplung zur Laufzeit von zwei Sternern miteinander.

Hier kann durch die Priorisierung des zentralen Knoten des Sterns und Gruppierung des Sterns insich erreicht werden, dass ein Knoten des Sterns sich nicht so stark durch externen Einfluss von seinem Cluster entfernt. Der Einfluss von externen Verbindungen ist somit stark reduziert und ein globales Zeitverständnis von gekoppelten Systemen würde sich nach der Kopplung nur langsam ausbreiten. Jedes der Teilsysteme kann in der Zeit für sich weiter

fehlerfrei operieren, jedoch ist es mit vergleichsweise vielen Synchronisationszyklen verbunden, bis sich das gesamte Netzwerk als eine zeitgebende Einheit versteht.

8.3 Laufzeitabweichung

Eine letzte Anomalie konnte über längere Laufzeit eines Netzwerks erkannt werden. Hier war zu sehen, dass ein Netzwerk, welches keinen Schranken in hinsicht maximalen Korrekturfaktoren bezüglich der Zykluszeit und des Offsets unterliegt, ein mehr und mehr verfälschtes Bild vom Verständnis der Zeit erhält. Durch die Summe an Ungenauigkeiten in der Taktgebung, Zeitauslesung und Beeinflussung der lokalen Zeit ist ein Knoten stets etwas langsamer oder schneller als seine Nachbarn. Überwiegen diese Abweichungen über einen topologisch zusammenhängenden Teil des Netzwerks, so entsteht eine kleine kollektive Verlangsamung oder Beschleunigung über das Verständnis der Globalen Zeit innerhalb dieser Netzwerkpartition. Dieser Effekt ist sehr klein und würde erst bei einer sehr hohen Anzahl an Hops zu einer Ende-zu-Ende-Asynchronität führen, beeinflusst jedoch das Netzwerk im Ganzen über die Zeit hinweg. Je nach Verteilung wird diese minimale Abweichung im gesamten Netzwerk propagiert und bewirkt eine Verlangsamung oder Beschleunigung der globalen Zeit. Ist das Netzwerk über eine längere Zeit aktiv und isoliert, so kann dieser Effekt das gesamte Netzwerk verlangsamen oder beschleunigen.

Dem kann entgegengesteuert werden, indem Schranken für die maximal mögliche Korrektur innerhalb eines Knotens festgelegt wird. Die Einschränkung eines Knotens im gesamten Netzwerks würde ausreichen, den Effekt des Abdriftens zu unterbinden. Dieser Knoten dient als eine Art zeitlicher Anker und verhindert eine stetige Neuorientierung des globalen Zeitverständis. Auch externe Knoten oder netzwerke können die Funktion des Ankers annehmen, womit der isolierte Betrieb des Netzes nicht mehr gegeben ist. Ein solches Netzwerk ist nurnoch zu Zeitkompensationen innerhalb der festgelegten Schranken fähig. Ebenso gilt es zu beachten, dass es sich streng genommen nicht mehr um einen rollenlosen Ansatz handelt und zusätzliche Konfiguration erfordert.

Es ist zu erkennen, dass allein die Bestimmung von Konvergenz zum erfolgreichen Aufrechterhalt einer Synchronisation im Peer-to-Peer-Betrieb unzureichend ist. Es gild für den jeweiligen Anwendungsfall die richtige Kombination aus Netzwerktopologie, Parametersatz und Konfiguration der Knoten zu erreichen.

9 Realisierung eines Synchronisationssystems

In Kapitel 4 auf Seite 15 wurde innerhalb der Kommunikationsebene die Umsetzbarkeit des Ansatzes auf Basis unterschiedlicher Fähigkeiten der Kommunikationshardware dargestellt. Um Folgenden wird die Umsetzung beispielhaft auf eine spezielle Plattform dargestellt. Dieses Kapitel zeigt somit die Umsetzbarkeit des zuvor als Simulation betrachteten Ansatzes aus der MATLAB-Simulink-Umgebung.

9.1 Struktur eines Peers

Zum Zwecke der Vergleichbarkeit wurden für dieses Szenario zur Umsetzung der Netzwerkeigenen Peers gleiche Endsysteme verwendet, was zudem die Rollenlosigkeit des vorgestellten Ansatzes unterstreicht. Hierbei wurde eine x64-Architektur auf Basis von COTS (commercial-off-the-shelf) Komponenten gewählt. Die folgende Abbildung (9.1) beschreibt die Architektur der Umsetzung eines Synchronisationsknotens als Layer-Modell.

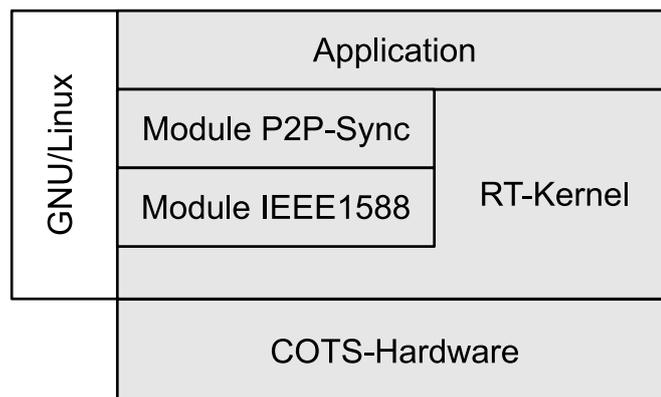


Abbildung 9.1: Layer-Struktur eines Synchronisations-Peers

Die Hardwareplattform zeichnet sich dadurch aus, dass sie über eine PTP-fähige Netzwerkkarte verfügt. Diese ermöglicht einen nach IEEE-1588 standardisierten Nachrichtenaustausch und bietet eine Zeitgeberstruktur, welche zur Kontrolle und Überwachung der lokalen Zeit genutzt werden kann. Der Ansatz bildet somit die Brücke zwischen einer möglichst kostengünstigen und einfach zu wartenden Plattform bei einer hohen erreichbaren Genauigkeit. Als Betriebssystem wird ein Open-Source-Derivat einer GNU/Linuxdistribution eingesetzt, welche verbunden mit einem Open-Source-Treiber der Netzwerkkarte vollen Einblick in das Zeitverhalten des systems ermöglicht. Das Betriebssystem setzt auf einem RT-Kernel auf, welcher eine vorhersagbare Reaktionszeit garantiert. Der Open-Source-Treiber ist von der Firma Intel als Kernelmodul realisiert und ermöglicht beliebigen Anwendungen auf die Synchronisierte Zeitbasis zuzugreifen, sowie Empfangs und Sendezeitpunkte von Ethernet-Nachrichten in Relation zu dieser Zeitbasis zu stellen. Diese Schnittstelle wird vom entwickelten P2P-Synchronisationsalgorithmus aufgegriffen um die Synchronisation nach dem in dieser Arbeit vorgestellten Protokoll durchzuführen. Der P2P-Synchronisationsalgorithmus ist ebenfalls als Kernelmodul realisiert und orientiert sich hierbei an der Umsetzung des PTP-Moduls. Hierbei ist jedoch zu beachten, dass das Kernelmodul die Funktionalitäten des PTP-Modul nicht neu implementiert, sondern Basisfunktionalität nutzt um mit eigenen Algorithmen darauf aufbauend nach eigenem Protokoll eine Synchronisation durchzuführen. Für eine Anwendung wird ein dem PTP-Modul entsprechendes Interface angeboten, sodass eine beliebige Anwendung, welche das PTP-Modul eingesetzt hat, sich mit nur wenigen Modifikationen auf das P2P-Modul hin abstimmen lässt und somit die Peer-to-Peer-Synchronisation nutzt. Es ist jedoch darauf zu achten, dass zum Zeitpunkt dieser Arbeit das Interface des P2P-Moduls nicht in seinem vollen Funktionsumfang implementiert wurde, sodass ggf. um die jeweilige Anwendung voll zu unterstützen, Erweiterungen des Funktionssatzes notwendig sind.

9.1.1 Systembestandteile

Die hier vorgestellte Umsetzung eines Peers basiert auf folgenden Anforderungen. Eine Realisierung ist ebenfalls mit vergleichbaren Mitteln möglich. Genauigkeiten des Systems können jedoch je nach Plattform variieren.

- **Host-System:** Die Hardwareplattform stellt eine Hostbridge und zugehöriger PCIe-Controller der Intelcorporation Skylake Serie dar. Es ist zu beachten, dass Technologien wie die Intel Management Engine (ME) nach Möglichkeit zu deaktivieren oder in den Modus „High Assurance Platform“ einzustellen sind, da diese für ein Betriebssystem

unvorhersagbare Verarbeitungszeit aus dem System fordern. Wird diese Vorkehrung nicht getroffen, so ist es für ein Betriebssystem nicht möglich Programmabläufe unterbrechungssicher oder mit vorhersagbarem Ablaufverhalten zu gestalten, was sich negativ auf die Präzision und somit die Qualität der Synchronisation auswirkt. Der eingesetzte Network Controller ist vom Typ *I350_{rev01}* und ebenfalls ein Produkt der Intel Corporation. Er zeichnet sich durch zwei Ethernetports aus, welche unabhängig voneinander von unterschiedlichen Ethernetstacks bedient werden können. Jedoch steht als Referenz für die lokale Zeit nur ein hochauflösender Zeitgeber zur Verfügung, womit ein Endsystem ein eindeutiges Verständnis von der lokalen Zeit besitzt. Multiple Ethernetstacks erfordern somit eine Abstimmung untereinander, welcher Ethernetstack den Zeitgeber bedient. Die Ausführung protokolleigener Programmteile kann innerhalb von hochpriorisierten low latency Interrupts erfolgen, was die Performance von Synchronisationsalgorithmen verbessert. Weiterhin verfügt der Controller über eine Technik des sogenannten Interrupt-Throttling, was eine Überlastung des Hostsystems durch Ethernetbursts abfängt. Sollte z.B. eine Flut von kleinstmöglichen (64 Bytes) Ethernetpaketen empfangen werden, so wird verhindert, dass für jedes Empfangsereignis ein ausführendes Event angestoßen wird. Somit sind Algorithmen zur Garbagecollection einfach umzusetzen.

- **Kernel:** Als Kernel wurde ein GNU/Linux-Kernel der Version 4.9.20 eingesetzt. Dieser wurde mit dem RT-Patch version *rt16* versehen und mithilfe der Konfiguration "full preemtable" in einen echtzeitfähigen Zustand versetzt. Der RT-Preemption-Patch trennt Interrupts vom Hardwarekontext so daß er nicht mehr unmittelbar die korrespondierende Interrupt-Service-Routine (ISR) aufruft. In einer zusätzlichen Funktion wird zwischen direkt aufzurufenden ISR und virtualisierten Interrupts unterschieden. Beim virtualisierten Interrupt wird lediglich dessen Auftreten vermerkt. Der eigentliche Aufruf der ISR erfolgt durch einen dedizierten Kernel-Thread mit mittlerer RT-Priorität. Ist kein RT-Task mit einer höheren Priorität rechenbereit wird die ISR durch den Kernel-Thread aufgerufen. Ansonsten muß dieser warten, bis kein anderer RT-Task höherer Priorität mehr rechnen möchte. Wird umgekehrt während der Abarbeitung der ISR ein RT-Task höherer Priorität rechenbereit, dann unterbricht dieser den Kernel-Thread. Wie ersichtlich, werden damit ISR's unterbrechbar, was sie normalerweise nicht sind.
- **PTP-Kernelmodul:** Als Basis wird der open-source PTP-Treiber der Firma Intel in version 1.2.1 genutzt. Der modifizierte Netzwerktreiber des Peers registriert sich auf eine nicht virtualisierte ISR. Diese kann direkt im HW-Kontext ausgeführt werden. Der

Prozess ist fest und exklusiv an einen CPU-Kern gebunden und nutzt den Systemeigenen high resolution timer um ein entsprechendes Zeitverhalten abzubilden. Er ist ebenfalls auf low latency Interrupts registriert.

9.2 Architektur und Umsetzung

Der eingesetzte Opensourcetreiber der PTP-fähigen Netzwerkkarte (Intel I350-T2) ermöglicht eine eigenständige Synchronisation, welche vollständig Protokollkonform zu IEEE 1588v2 ist. Hierzu werden vom IEEE1588-Modul mithilfe der synchronisationsnachrichten Empfangs und Versendezeitpunkte in Relation zur eigenen Uhr, sowie der des Synchronisationspartners ermittelt (vgl. Kapitel 4 auf Seite 15 bezüglich der Zeitpunkte t_1, t_2, t_3 und t_4). Das Modul nutzt die Eigenschaften der Netzwerkkarte ein- und ausgehende Nachrichten mit einem präzisen Hardwarezeitstempel zu versehen, um eine vom Betriebssystem latenzunabhängige Zeitbasis zu schaffen. aus diesen Zeitwerten kann die Übertragungsverzögerung (unter der Annahme, die verzögerung sei unabhängig von der Kommunikationsrichtung) errechnet werden. Sind diese Informationen vorhanden, so kann vom IEEE1588-Modul die aktuelle Abweichung der eigenen Uhr zur Uhr des Synchronisationspartners ermittelt und die Korrektur an der lokalen Uhr entsprechend vorgenommen werden. Das entwickelte P2P-Modul nutzt die ermittelten Zeitpunkte t_1 - t_4 und die Informationen über die Verzögerung durch die Übertragungsstrecke, deaktiviert jedoch die automatische Korrektur der lokalen Uhr durch das IEEE1588-Modul. Das P2P-Modul lässt über das IEEE1588-Modul die Zeitpunkte t_1 - t_4 , sowie die Verzögerungsinformationen für jeden Link ermitteln. Es wird für jeden Link nach Gleichung 4.2 auf Seite 17 der Sollwert bestimmt. Anschliessend erfolgt eine Gewichtung und Zusammenführung der Sollwerte nach der Gewichtungsfunktion (Gleichung 5.3 auf Seite 24). Das Ermitteln der Zeitinformationen für einen Link ist hierbei ein zyklischer Prozess. Konnten in einem Zyklus (noch) nicht alle nötigen Informationen für einen Link zusammengetragen werden, so trägt dieser Link (noch) nicht zur Gewichtung des ebenfalls zyklusgebundenen Ergebnis bei. Der resultierte gewichtete Sollwert ist nun der Eingang der Differenzgleichung (vgl. 7.6 auf Seite 47), welche das Regelungsverhalten des Endsystems bestimmt. Das Ergebnis ist der Wert für die Offsetkorrektur des lokalen Systems und wird zur Justierung der Hardware-Uhr der Netzwerkkarte über das IEEE1588-Modul eingesetzt. Sollten in einem Zyklus alle Links keine Zeitinformationen liefern so wird in diesem Zyklus nicht korrigiert und auch kein vorangegangener Schritt für die Errechnung der Differenzgleichung vorgehalten. Durch diesen Ansatz erfolgt eine kontinuierliche Synchronisation des Systems mit potentiellen Nachbarn. Eine

Anwendung kann sich nun zu beliebiger Zeit die Zeitinformationen der Hardware-Uhr über die Schnittstelle des P2P Moduls anfordern und als synchronisierte Referenzzeit nutzen.

9.3 Gleichungen und Übertragungsfunktionen

Aus Vorarbeiten (vgl. Müller (2017a)) konnten entsprechende Parametersätze für Topologien mit weniger als 5 Knoten ermittelt werden. Der dort vorgeschlagene Parametersatz $s = \{1, 20, 20\}$ wird in dieser Anwendung herangezogen und als Basis für das Verhaltensmuster der Knoten innerhalb der betrachteten Topologie realisiert.

$$G(s) = \frac{1}{\frac{1}{20}s^2 + s + 1} \quad (9.1)$$

Durch Koeffizientenvergleich mit der Gleichung 7.1 auf Seite 46 ergeben sich die Koeffizienten A_n und α_n wie folgt:

$$\begin{aligned} A_0 &= 0 \\ A_1 &= 1,11803 \\ A_2 &= -1,11803 \\ \alpha_1 &= -1,05573 \\ \alpha_2 &= -18,9443 \end{aligned} \quad (9.2)$$

Es folgt somit die Übertragungsfunktion des Systems:

$$G(z) = \frac{0,389z}{z^2 - 0,348z + 2,06 \cdot 10^{-9}} \quad (9.3)$$

$$G(z) = \frac{0,389z^{-1}}{1 - 0,348z^{-1} + 2,06 \cdot 10^{-9}z^{-2}} \quad (9.4)$$

$$\Rightarrow y[k] = 0,348 \cdot y[k-1] - 2,061 \cdot 10^{-9} \cdot y[k-2] + 0,389 \cdot u[k-1] \quad (9.5)$$

Im weiteren Verlauf wird jeder Knoten mit der Übertragungsfunktion instanziiert. Somit ist die Verhaltensweise der Knoten vollständig abgebildet.

9.4 Testaufbau

Da in traditionellen Ansätzen Rückkopplungen im Synchronisationspfad Probleme in der Synchronisation verursachen können wird als Topologie für den Messaufbau der Ring gewählt. Somit kann durch den Versuch gezeigt werden, dass Rückkopplungen für ein Peer-to-Peer-Algorithmus kein Problem darstellen und diese sich sogar Positiv auf die Dauer der Konvergenz innerhalb des Netzwerks auswirkt. Der Ring wird durch drei baugleiche Knoten gebildet (vgl. Abbildung 9.2). Dies unterstreicht den rollenlosen Ansatz, ist jedoch keine Voraussetzung.

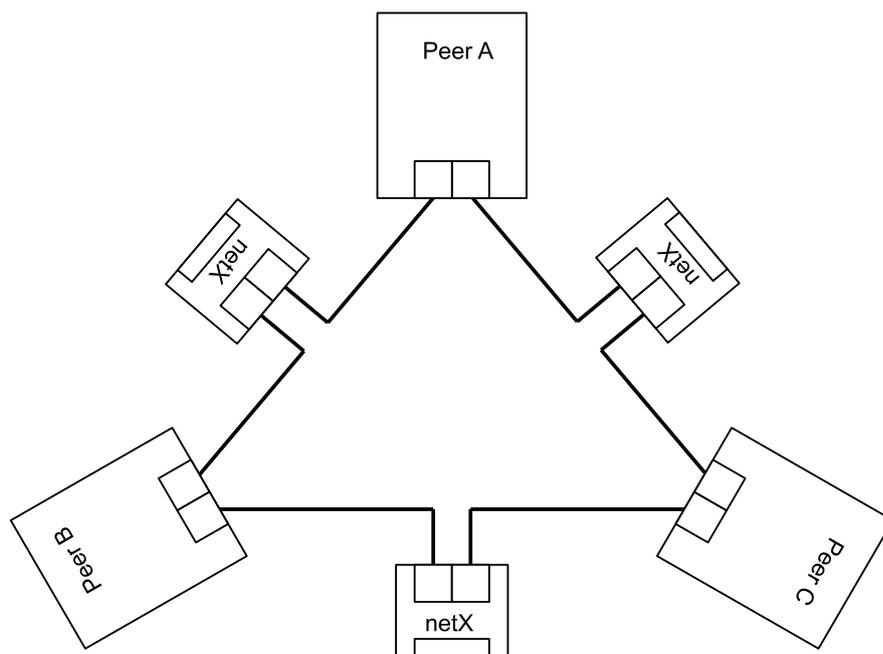


Abbildung 9.2: Topologie des Messaufbaus

Hierbei wird von jedem Knoten eine Netzwerkverbindung zum jeweils anderen Knoten aufgebaut. Um das zeitliche Verhalten eines Knotens vollständig beobachten zu können wird ein dedizierter Messcontroller verwendet (netX), welcher aus den Nachrichten des Knotens die Stellung der internen Uhr zurückführt. Somit kann zwar die Lokale Uhr eines Teilnehmers nicht

direkt gelesen werden. Sie kann jedoch Indirekt über die Informationen bestimmt werden, welche im Kommunikationsfluss des synchronisierenden Knotens vorzufinden sind. Der Controller leitet den Nachrichtenverkehr transparent von einem Port auf den anderen weiter. Um die Beeinflussung durch Verzögerungszeiten auf den zu untersuchenden Synchronisationsalgorithmus zu minimieren, stellt der netX die gleiche Verzögerungszeit zwischen dem Weiterreichen in die eine oder andere Richtung sicher. Da der Synchronisationsalgorithmus die Zeiten der Übertragungsstrecke unter der Prämisse, dass beide Richtungen mit gleicher Verzögerung behaftet sind, herausrechnet ist somit keine große Beeinflussung durch die netX-Controller zu erwarten. Das Zeitverhalten der Nachrichten entspricht somit denen beim Transfer über ein sehr langes Kabel.

Damit der Controller dies erreichen kann wurden speziell für den Nachrichtenverkehr ausgelegte Controller verwendet. Kernstück der Architektur ist ein System on Chip Design, dem netX. Es ermöglicht die unabhängige Kommunikation über mehrere Kanäle zur gleichen Zeit. Die Kompatibilität auf Physical Layer Basis wird mit Hilfe von Modulen erreicht. Alle Kommunikationskanäle des Controllers werden jeweils durch unabhängige Transmission- und Reception-Units betrieben. Angebunden über eine Dataswitch-Architektur können Alle Kommunikationskanäle gleichzeitig auf den Arbeitsspeicher des Controllers zugreifen und somit verzögerungs und jitterarm den Transport der Daten sicherstellen. Weiterhin sorgt eine Timestamping-Einheit dafür, dass jeder empfangende Start-of-Frame eine Zeit erhält, die der Controllereigenen Zeit entspricht.

Zeitwerte können somit innerhalb des Linux Systems vom IEEE1588-Modul ausgelesen und in Nachrichten verpackt werden. Da die Netzwerkkarte nur über eine Hardware-Uhr verfügt werden von ihr beide Ethernetports gleichzeitig bedient. Diese Uhr wird vom PTP-Modul korrigiert. Alle Referenzzeitpunkte als gemessene Zeiten oder automatisch generierten Zeitstempel an Nachrichten gebunden referenzieren somit die gleiche Zeitbasis. Mit dem Ansatz wird es zudem möglich, eine Anwendung gegen die Schnittstelle des Intel IEEE-1588 Netzwerktreibers zu linken und dennoch den neuen P2P-Ansatz zu nutzen. (Es ist jedoch darauf zu achten, dass zum Zeitpunkt dieser Arbeit die Schnittstelle nicht in ihrem vollen Funktionsumfang implementiert ist.) Die vom Treiber versendeten Nachrichten können vom netX Ausgewertet werden und aus ihnen die Differenz der benachbarten Knoten zueinander ermittelt werden. Die Differenzen der Zeiten können anschliessend über eine Schnittstelle am Controller (UART) abgerufen werden.

9.5 Ergebnisse der Realisierung

Der Versuch gliedert sich in zwei Phasen. Zunächst sind Peer B und C bidirektional miteinander gekoppelt. Peer A ist zu diesem Zeitpunkt noch nicht Teil des Netzwerks. Hier lässt sich schon die Auswirkung der Wechselwirkung zwischen zwei gleichberechtigten Knoten ablesen. Zu Beginn der zweiten Phase wird Peer A jeweils bidirektional mit Peer B und C verbunden. Zu diesem Zeitpunkt sind zwei vollständige Synchronisationsringe entstanden, deren Auswirkungen in der zweiten Phase analysiert werden können.

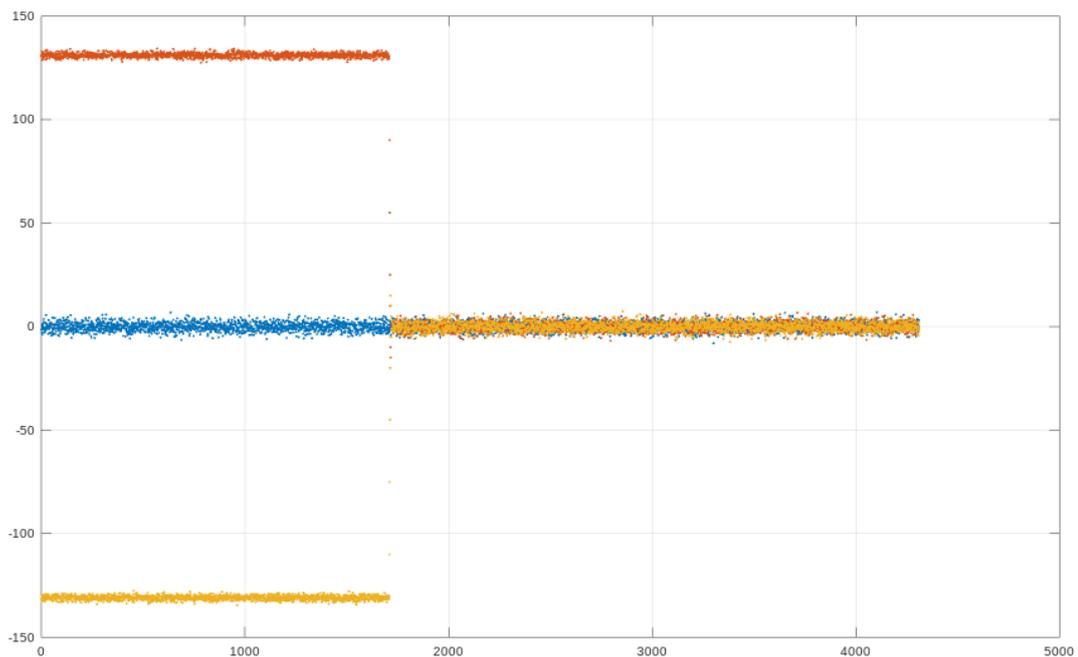


Abbildung 9.3: Abweichungsverteilung der lokalen Uhren

Die Abbildung 9.3 beschreibt die gemessene Differenz zwischen den aus den Synchronisationsnachrichten ermittelten lokalen Uhren der Peers zueinander. Die Differenzen sind über die Zyklusnummer aufgetragen. Im ersten Teil des Graphen ist zu erkennen, dass der netX zwischen Peer B und C (blau) keine größeren Unterschiede feststellen kann und ein synchronisiertes System trotz Wechselwirkung sieht. Die netX zwischen Peer A und B (rot) und zwischen A und C (gelb) können hingegen wie zu erwarten eine Differenz feststellen, da auf dieser Strecke keine Nachrichten innerhalb der netX auf den jeweils anderen Port weitergeleitet wird. Zum Übergang in die zweite Phase wird innerhalb der netX die Weiterleitung aktiviert. Da

sich im Netzwerk nur Knoten gleicher Gewichtung befinden und es sich um eine symmetrische Topologie handelt ist diese Symmetrie ebenfalls in der Aushandlung einer gemeinsamen Zeitbasis wiederzufinden. Zum Zeitpunkt des Zusammenschaltens ist zu sehen, wie innerhalb weniger Zyklen die Abweichung zum Peer A in den Bereich des Jitters sinkt, womit man von einem synchronisierten System sprechen kann.

Es ist jedoch zu beachten, dass die Messung keine Aussage über die Geschwindigkeitsänderung des Netzes trifft. Die Zyklen innerhalb der Kopplung in der ersten Phase können durch die Kopplung des neuen Knotens in ihrer Dauer beeinträchtigt worden sein.

10 Zusammenfassung

Neue Synchronisationsmechanismen erschließen auf dem Gebiet der echtzeitfähigen Netzwerkstrukturen neue potentielle Einsatzmöglichkeiten. Diese Arbeit konnte die Machbarkeit von Peer-to-Peer-Technologie als Synchronisationsmechanismus in Paket basierten Kommunikationsstrukturen darstellen.

Eine Realisierung der Diskretisierung des ursprünglichen Ansatzes konnte die gleichen Konvergenzeigenschaften des Netzwerks zeigen, wie von der Simulation über MATLAB-Simulink dargestellt. Für die Erzeugung der verhaltensbeschreibenden Differenzgleichung eines Knotens ist die Transformation der Übertragungsfunktion nach Festlegung des Parametersatzes notwendig. Dieser Parametersatz lässt sich anschliessend nicht mehr ohne eine neue Transformation anpassen, was eine nachträgliche Optimierung der Regelungseigenschaften erschwert. Ist eine Optimierung erwünscht, so ist zur Planungszeit das Wissen über zukünftige Topologien innerhalb derer Knoten mit der speziellen Übertragungsfunktion zum Einsatz kommen notwendig. Je genauer diese Topologien bekannt sind desto effizienter fällt die mögliche Optimierung des Parametersatzes aus. Da nur die Übertragungsfunktion transformiert wird, lässt sich das Netzwerk, wie im Kontinuierlichen, als Mehrgrößensystem betrachten. Dies erlaubt eine getrennte Beschreibung vom Verhalten der Netzwerkteilnehmer und der Informationen über die Topologie. Ist einmal die Transformation durchgeführt kann die Topologie zwecks einer allgemeinen Systembetrachtung manipuliert und die Auswirkungen des eingesetzten Parametersatzes betrachtet werden.

Es gilt jedoch zu beachten, dass der regelungstechnische nachweis der Konvergenz nur ein Teil eines fehlerfreien Synchronisationsmechanismus ist. Durch den rollenfreien Ansatz der Peer-to-Peer-Architektur können Nebenefekte auftreten, welche Asynchronität unter bestimmten Bedingungen in Teilen des Netzwerks auslösen oder ganze Netzwerke in ihrem zeitverständnis abdriften lassen. Hier gilt es Vorkehrungen zu treffen, welche auf den jeweiligen Anwendungsfall zugeschnitten sind. Dies stellt jedoch einen erhöhten Aufwand an die Konfiguration solcher Netzwerke.

Ist der Algorithmus korrekt konfiguriert und parametrisiert, so synchronisiert dieser trotz Rückkopplungen und Ringe im Kommunikationspfad. Der Ablauf der Synchronisation des Netzwerks teilt sich dabei in Zyklen ein. Für jeden Zyklus ist eine direkte algebraische Bestimmung der lokalen Differenzgleichung nötig. Zudem übermittelt jeder Knoten des Netzwerks die Information über den Stand seines lokalen Zeitgebers an seine direkten Nachbarn (Peers) mindestens einmal pro Zyklus. Sind diese Anforderungen erfüllt, so erreicht ein Netzwerk, mit der Implementierung einer stabilen Diskretisierung, bei allen Teilnehmern ein globales Verständnis von der Zeit.

11 Ausblick

Es wurde gezeigt, dass der Einsatz einer Peer-to-Peer-Synchronisation ebenso mit einem hohen Masse an Vorarbeit und Konfigurationsaufwand verbunden ist wie alternative Ansätze. Es gilt zu untersuchen, ob eine Vereinfachung der Konfiguration in Abhängigkeit zur Topologie möglich ist. Möglichkeiten könnten hier in der Klassifizierung von Typischen Netzwerktopologien liegen um aus diesen Rückschlüsse auf die Parameter einer speziellen Topologie ziehen zu können. Auch wäre die automatische Berechnung der Parameter durch geeignete Algorithmen denkbar, was den Peer-to-Peer-Ansatz wieder näher zur Idee eines Zero-Conf-Ansatzes Bringen würde.

Zudem ist der Nachweis der Stabilität für bestimmte Parametersätze durch die Matritzen-darstellung sehr aufwendig. Die Komplexität steigt mit der Größe des Netzes Exponentiell, was den Ansatz in seiner jetzigen Form für Anwendungen auf Komplexen Netzen unpraktikabel erscheinen lässt. Hier könnte eine Vereinfachung der Matritzendarstellung den Grad der Komplexität reduzieren um mit numerischen Verfahren mit ökonomischen Aufwand eine Berechnung durchführen zu können.

Sind die Fragestellungen der Komplexitätsbetrachtung und der Parametrisierung mit weniger Aufwand in den Griff zu bekommen, so ist der Einsatz eines Peer-to-Peer-Algorithmus in Bereichen denkbar, in denen ein Kompromis zwischen der Genauigkeit der Echtzeitkritischen Kommunikation und einer Anpassungsfähigkeit des Netzes gewünscht ist.

Literaturverzeichnis

- [AS6802 2011] AS6802: *Time-Triggered Ethernet AS6802*. SAE Aerospace. November 2011. – URL <http://standards.sae.org/as6802/>
- [Bruckmeier 2010] BRUCKMEIER, Robert: *Ethernet for Automotive Applications*. Vortrag. Juni 2010. – URL http://www.freescale.com/files/ftf_2010/Americas/WBnr_FTF10_AUT_F0558.pdf. – Zugriffsdatum: 2010-12-10
- [Ferrari u. a. 2008] FERRARI, P. ; FLAMMINI, A. ; RINALDI, S. ; GADERER, G.: Evaluation of clock synchronization accuracy of coexistent Real-Time Ethernet protocols. In: *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. Piscataway, NJ, USA : IEEE Press, September 2008, S. 87–91
- [Garner u. a. 2009] GARNER, Geoffrey M. ; GELTER, Aaron ; TEENER, Michael D. J.: New simulation and test results for IEEE 802.1AS timing performance. In: *International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS 2009)*. Piscataway, NJ, USA : IEEE Press, 2009, S. 1–7. – ISBN 978-1-4244-4391-8
- [Golnaraghi und Kuo 2009] GOLNARAGHI, Farid ; KUO, Benjamin C.: *Automatic Control Systems*. 9th. Wiley Publishing, 2009. – ISBN 0470048964, 9780470048962
- [IEEE 1588 2002] IEEE 1588: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems / IEEE. 2002 (IEEE Std. 1588). – Standard. – ISBN 0-7381-3369-8
- [Loeser und Haertig 2004] LOESER, Jork ; HAERTIG, Hermann: Low-latency hard real-time communication over switched Ethernet. In: *Real-Time Systems, 2004. ECRTS 2004. Proceedings. 16th Euromicro Conference on*, Juni 2004, S. 13–22. – ISSN 1068-3070
- [Lunze 2013] LUNZE, Jan: *Regelungstechnik 2, Mehrgrößen- und Mehrgliedersysteme, Digitale Regelung*. 2013

- [Lyons 2010] LYONS, Richard G.: *Understanding Digital Signal Processing (Edition 3)*. Upper Saddle River, NJ, USA : Prentice Hall PTR, 2010. – ISBN 0131089897
- [Mallada u. a. 2013] MALLADA, E. ; MENG, Xiaoqiao ; HACK, M. ; ZHANG, L. ; TANG, A.: Skewless network clock synchronization. In: *2013 21st IEEE International Conference on Network Protocols (ICNP)*, Oct 2013, S. 1–10. – ISSN 1092-1648
- [Müller 2017a] MÜLLER, Kai: P2P-basierte Zeitsynchronisierung in verteilten Systemen: Ein diskreter regelungstechnischer Ansatz / CoRE Research Group, Hochschule für Angewandte Wissenschaften Hamburg. Juli 2017. – Forschungsbericht
- [Müller 2017b] MÜLLER, Kai: Zeitsynchronisierung in verteilten Systemen: Ein Peer-to-Peer-basierter Ansatz im kontinuierlichen Fall / CoRE Research Group, Hochschule für Angewandte Wissenschaften Hamburg. Juli 2017. – Forschungsbericht
- [Nelles 2015] NELLES, Oliver: Zustandsraum und Digitale Regelung / Universität Siegen. März 2015. – Forschungsbericht
- [Tan 2007] TAN, Alexander E.: IEEE 1588 Precision Time Protocol Time Synchronization Performance / National Semiconductor Corporation . URL <http://www.national.com/an/AN/AN-1728.pdf>. – Zugriffsdatum: 2010-12-10, 2007 (AN-1728). – Forschungsbericht
- [Union 2015] UNION, International T.: Timing characteristics of a synchronous Ethernet equipment slave clock / ITU-T, Telecommunication Standardization Sector of ITU. Januar 2015. – Forschungsbericht
- [Weibel 2005] WEIBEL, Hans: High Precision Clock Synchronization according to IEEE 1588 Implementation and Performance Issues. In: *Embedded World Conference 2005*. Design & Elektronik, WEKA-Fachzeitschriften-Verlag, Februar 2005, S. 981–989

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 8. May 2018

 Kai Müller