

Konzeption und Inbetriebnahme einer Anlage und Modellbildung zur Raumheizungsregelung für den Betrieb mit Modellprädiktiver Regelung

Conception and startup operations of an installation and modelling for a space
heating control to run with model predictive control

Master-Thesis
im Studiengang Wirtschaftsingenieurwesen

zur Erlangung des akademischen Grades
Master of Science (M.Sc.)

vorgelegt von
Daniel Johannes Mayer
aus Sulzfeld

Erstkorrektor: Prof. Dr. Angelika Altmann-Dieses

Zweitkorrektor: Prof. Dr.-Ing. Marco-Braun

Matr.-Nr.: 51968

E-Mail: daniel-j-mayer@gmx.de

Bearbeitungszeitraum: 17.03.2015 – 29.02.2016

Tag der Einreichung: 17.03.2016

Fakultät für Wirtschaftswissenschaften
Hochschule Karlsruhe – Technik und Wirtschaft
2016

Kurzfassung

Abstract

This text should be *italic*

Schlüsselwortliste: Test1, Test2, Test3

Keywords: *Test1, Test2, Test3*

Danksagung

Inhaltsverzeichnis

Tabellenverzeichnis	III
Abbildungsverzeichnis	IV
Quelltextverzeichnis	V
Symbolverzeichnis	VI
1 Einleitung	1
1.1 Motivation und Problemstellung	1
1.2 Zielsetzung und Aufbau der Arbeit	2
2 Theoretische Grundlagen	3
2.1 Model Predictive Control	3
2.2 Technische Grundlagen zur Kommunikation mit Bussystemen	3
2.2.1 Bussysteme	3
2.2.1.1 Informationsaustausch	4
2.2.1.2 Netzwerk und Topologie	4
2.2.1.3 Buszugriffsverfahren	7
2.2.1.4 Datensicherung	8
2.2.1.5 Schnittstellen	9
2.2.2 OSI-Kommunikationsmodell	13
2.2.3 Modbus Kommunikationstechnologie	18
2.3 Technische Grundlagen zur Modellbildung	24
2.3.1 Thermodynamische Systeme	24
2.3.2 Erster Hauptsatz der Thermodynamik	26
2.3.3 Wärmeübertragung	27
2.4 Technische Grundlagen zur Solar- und Gebäudetechnik	28
2.4.1 Klima	28
2.4.2 Sonnenbahn und Strahlung	28
2.4.3 Gebäudetechnik Glas und Wärmedurchgangskoeffizienten	28
3 Anlagendesign	29
3.1 Idee und Konzept der Anlage	29
3.1.1 Analyse der Anforderungen und Rahmenbedingungen	29
3.1.2 Idee	34
3.1.3 Idee weiter konkretisiert	34
3.1.4 Konkretisierte Planung der Anlage/Konkretisierung des Plans	35
3.2 Bauteile	36
3.2.1 Gateway	36

3.2.2	Temperatursensoren	36
3.2.3	Wärmemengenzähler	36
3.2.4	Signalwandler und Stellantrieb	36
3.2.5	Die Gesamtanlage mit Schaltplan und allem	36
4	Modellbildung des Raumes	37
4.1	Anforderungen an das Raummodell	37
4.2	Das Grundmodell des Raumes	37
4.3	Modellerweiterung durch Berücksichtigung der realen Umgebung	39
4.4	Modellerweiterung durch Berücksichtigung der räumlichen Gegebenheiten	41
4.5	Modellerweiterung durch Berücksichtigung von Störgrößen und der Sonneneinstrahlung	43
4.6	Validierung des Modells	43
4.7	Anpassung des Modells mit Parameterschätzung	43
5	Schlussbetrachtung	44
5.1	Fazit	44
5.2	Ausblick und Ansatzpunkte für weitere Arbeiten	44
A	Fertiges Raummodell in Modelica	45
	Literaturverzeichnis	50

Tabellenverzeichnis

Tab. 3.1: Umsetzung der Ziele in Anforderungen der Anlage	31
Tab. 4.1: Eigenschaften des Raummodells	39
Tab. 4.2: Weitere Eigenschaften des Raummodells	41

Abbildungsverzeichnis

Abb. 2.1:	Bus-Struktur	5
Abb. 2.2:	Impulsverzerrung auf einer Leitung	6
Abb. 2.3:	Baumstruktur	7
Abb. 2.4:	<i>Cyclic Redundancy Check</i>	10
Abb. 2.5:	Parallele und serielle Datenübertragung	11
Abb. 2.6:	Spannungspegel und Stecker der EIA 232-Schnittstelle	12
Abb. 2.7:	Spannungspegel und Stecker der EIA 485-Schnittstelle	12
Abb. 2.8:	Die sieben Schichten des Open System Interconnection Modells . .	15
Abb. 2.9:	Die vier Dienstvorgänge	16
Abb. 2.10:	Die Modbus Kommunikation im OSI-Referenzmodell	19
Abb. 2.11:	Allgemeiner Rahmen für Telegramme nach dem Modbus Anwendungsprotokoll	19
Abb. 2.12:	Transaktion mit dem Modbus Protokoll	20
Abb. 2.13:	Datenmodell und Adressierung nach dem Modbus Protokoll	21
Abb. 2.14:	Serielle Kommunikation über Modbus RTU	23
Abb. 3.1:	Raumskizze K004A vom K Gebäude der Hochschule Karlsruhe – Technik und Wirtschaft	32
Abb. 4.1:	Grundmodell eines Raumes	38
Abb. 4.2:	Erweitertes Raummodell	42
Abb. 4.3:	Erweitertes Raummodell	43

Quelltextverzeichnis

4.1 Einfaches Gleichungssystem für das Grundmodell des Raumes in Modelica	39
4.2 Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der realen Umgebung in Modelica	40
4.3 Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der räumlichen Gegebenheiten in Modelica	41
4.4 Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der Sonneneinstrahlung und Störgrößen	43
listings/room_model_listing.mo	45

Symbolverzeichnis

Hinweis: Bei der Angabe der Symbole soll sich auf die Wesentlichen beschränkt werden. Die jeweils zutreffende Bedeutung ergibt sich entweder aus dem Kontext oder ist explizit im Text angegeben.

Formelzeichen

$A_{exchange}$	Wärmeaustauschoberfläche [m^2]
c_p	Spezifische Wärmekapazität eines Stoffes [$\frac{J}{kg \cdot K}$]
E	Gesamtenergie eines Systems [J]
E_{kin}	Kinetische Energie eines Systems [J]
E_{pot}	Potenzielle Energie eines Systems [J]
f_{max}	Maximale Übertragungsfrequenz [Hz]
T_0	Celsius Nullpunkt bei [$273,15K$]
T	Kelvin Temperatur [K]
m	Masse [kg]
\dot{m}	Massenstrom [$\frac{kg}{s}$]
m_{sys}	Masse eines Systems [kg]
P	Leistung [W]
p_{high}	High-Pegel für ein Signal, entspricht logischer „1,“
p_{low}	Low-Pegel für ein Signal, entspricht logischer „0,“
Q	Wärme [J]
\dot{Q}	Wärmestrom [W]
t	Celsius Temperatur [$^{\circ}C$]
Δt_{Imp}	Impulsverzerrung [s]
$U - Wert$	Materialabhängiger Wärmedurchgangskoeffizient [$\frac{W}{K \cdot m^2}$]
U	Innere Energie eines Systems [J]
W	Arbeit [J]

Griechische Buchstaben

κ	Erwartete Umsetzungsschwierigkeit
----------	-----------------------------------

Ω	Systemelement (Raum, Organisation, Technik)
ρ_n	Gewichtung des Wandlungspotentialmerkmals n
$\lambda_{abs,n}$	Absolute Teilweichtigkeit des Merkmals n
$\lambda_{rel,n}$	Relative Teilweichtigkeit des Merkmals n

Lateinische Buchstaben

kt	Kundentakt
$rw_{\emptyset}(t)$	Mittlere Lagerreichweite in Periode
t_{AZ}	Verfügbare Arbeitszeit in Periode
$T_{\emptyset BZ}$	Mittlere Gesamtbearbeitungszeit
$T_{\emptyset DLZ}$	Mittlere Gesamtdurchlaufzeit eines Produkts
$T_{\emptyset PZ}$	Mittlere Gesamtprozesszeit
$x_{\emptyset B(t)}$	Mittlerer Bedarf in Periode
$x_{\emptyset LB}$	Mittlerer Lagerbestand

Mathematische Operatoren

$(a; b]$	Halboffenes Intervall
\Leftrightarrow	Genau dann, wenn
\cong	Näherungsweise
\forall	Für alle
\in	Ist Element von
\mathbb{N}^+	Menge der natürlichen Zahlen ohne 0
\mathbb{R}	Menge der reellen Zahlen

*„Erfolgreich zu sein setzt zwei Dinge voraus: Klare Ziele
und den brennenden Wunsch, sie zu erreichen.“*

— JOHANN WOLFGANG VON GOETHE

1 Einleitung

1.1 Motivation und Problemstellung

Von allgemein solare Anwendung hin zu Forschung HSKA wird das Ziel sein.

Hinführung und Forschung der HSKA aufzeigen und daher einordnen Forschen derzeit bereits auf dem Gebiet (MPC mit) der solaren Anwendungen. Derzeit wird eine große Solaranlage zur Kühlung des Atriums installiert. Die Anlage nutzt die in Solarkollektoren gewonnene Wärmeenergie um eine Adsorptionskälteanlage anzutreiben. Die Anlage ist aufgrund ihrer Größe träge in der Reaktion und bisher noch im Aufbau. Komplementär dazu soll eine kleine Anlage mit solarem Bezug aufgebaut werden um die Forschung auch für ergänzende Dinge betreiben zu können.

Die Modellprädiktive Regelung beschäftigt sich damit, ein technisches System oder allgemeiner einen Prozess – im mathematisch exakten Sinne – optimal zu regeln.

Im Dialog mit den Projektverantwortlichen der Hochschule Karlsruhe für die Erforschung von solaren Anwendungen, in Person von Herrn ADRIAN BÜRGER und MARKUS BOHLAYER, die Einsatzziele gemeinsam erarbeitet und sind im Detail in Kapitel 3.1.1 ausgeführt.

Wichtigste Ergebnisse aus Dialog ist die Motivation dieser Forschungsarbeit:

Die Motivation dieser Arbeit ist also, einen komplementären Forschungsbeitrag für solare Anwendungen zu leisten im Hinblick auf die große Anlage.

Dass heißt jene Chancen zu nutzen, die die große Anlage nicht bieten kann einen möglichst großen Nutzen zu realisieren, also was die große Anlage nicht leisten kann und die große Anlage zu entlasten im Sinne von Empfindlichkeit/Versuche.

Außerdem sollen möglichst Erfahrungen gesammelt werden, die beim Anlauf der großen Anlage von Nutzen sein können.

!!!!!!!!!!!!!! Forschungsfrage, vorher Begriffe klären: MPC, Anlage, Modell, Steuerung/Regelung Als Forschungsfrage und Problemstellung dieser Arbeit ergibt sich die Frage, wie eine Anlage und ein Modell derselben aufgebaut sein müssen, um die Regelung einer Raumtemperatur mit Modellprädiktiver Regelung zu ermöglichen.
!!!!!!!!!!!!!!

1.2 Zielsetzung und Aufbau der Arbeit

Das übergeordnete Ziel dieser Arbeit ist es, eine Test- und Anwendungsumgebung zu schaffen, um die Forschung der Hochschule Karlsruhe auf dem Gebiet der Modellprädiktiven Regelung von solaren Anwendungen weitere Forschung weiter voran zu treiben und komplementär zu ergänzen.

Aus der Forschungsfrage lässt sich als konkretes Ziel der Arbeit ableiten, die Konzeption, Planung und Umsetzung einer Anlage und Bildung eines Modells zur Regelung einer Raumtemperatur mit Modellprädiktiver Regelung. Also ein Labor Weiterhin soll mit dieser Arbeit Know-How und Erfahrung generiert werden das bei der weiteren Forschung von Nutzen sein kann/ist.

Im Rahmen dieser Arbeit werden – anschließend an die Einleitung in 1 – die theoretischen Grundlagen in 2 ausgeführt. Zunächst wird die grundlegenden Theorie zu Model Predictive Control vorgestellt bevor anschließend weitere technische Grundlagen erklärt, welche für das weitere Verständnis dieser Arbeit benötigt werden.

Danach wird das technische System in Kapitel 3 Schritt für Schritt entwickelt, ausgehend von der Idee und den räumlichen Gegebenheiten/Nebenbedingungen, und weiter konkretisiert bis zur realisierten Umsetzung in eine funktionierende Anlage.

Dementsprechend werden zunächst das Konzept, die Planung und die technische Umsetzung der konkreten Anlage dargestellt, bevor anschließend die theoretischen Grundlagen von Model Predictive Control und zur die Modellbildung erläutert werden. Anschließend wird das Modell für Model Predictive Control gebildet und ein erstes grobes Konzept zur Steuerung der Raumtemperatur vorgestellt. Abschließend wird eine Validierung des Modells versucht und findet eine Anpassung des Modells statt damit es künftig mit Model Predictive Control genutzt werden kann.

„Theorie ist die Mutter der Praxis.“

— LOUIS PASTEUR

2 Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen erläutert die benötigt werden, um die Forschungsfrage zu beantworten. Zunächst MPC dann eher technisch

2.1 Model Predictive Control

Model Predictive Control an sich ist eine Methodik zur Steuerung von Systemen. Diese versucht zunächst, zu sich periodisch wiederholenden, diskreten Zeitpunkten das Verhalten eines Systems in der Zukunft – also einer immer gleich weit in die Zukunft hineinreichenden Periode – zu beschreiben. Hierzu bedient Model Predictive Control sich der Kenntnis des aktuellen Zustandes und eines physikalischen-mathematischen Modells des Systems, um dessen zukünftiges Verhalten „vorherzusagen“ bzw. abzubilden. Des Weiteren wird versucht das Verhalten des Systems mit minimalem Aufwand zu beeinflussen, um einem eigens- oder vordefinierten Zielkriterium zu folgen beziehungsweise diesem zu entsprechen.

2.2 Technische Grundlagen zur Kommunikation mit Bussystemen

In diesem Kapitel werden die Grundlagen von Hard- und Software beleuchtet die für die Kommunikation der Steuerung mit den einzelnen Anlagenteilen benötigt werden. Diese umfassen zunächst Bussysteme im Allgemeinen, dann das OSI Modell für techn. Kommunikation und werden anhand des spezifischen/konkreten Anwendungsfalls Modbus erläutert.

Die Einführung wird an der Struktur von Schnell u. Wiedemann [2006] anlehnen.

2.2.1 Bussysteme

Um allgemein Prozesse zu überwachen, zu steuern oder regeln zu können, müssen zwischen den verschiedenen Prozessbeteiligten/einheiten Informationen ausgetauscht werden. Im Kontext dieser Arbeit gilt es einen Prozess zur Temperatursteuerung/Halten zu regeln. Die Prozessbeteiligten sind hierbei technische Bauteile, Aktoren und Sensoren sowie ein Steuerungsrechner/Controller, die zusammen ein technisches System bilden, dass im Folgenden als Anlage bezeichnet wird. Die Anlage zeichnet sich dadurch aus, dass sie eine eigenständige funktionale Einheit bildet einen eigenen Zweck

verfolgt, die Raumtemperaturregelung und einen Mehrwert gegenüber ihrer Einzelteile hat, was dem Zusammenspiel der einzelnen Bauteile und Geräte entspricht. Um den Mehrwert zu realisieren und den Zweck zu erfüllen, werden Kommunikationssysteme benötigt. Mit Hilfe derer kann Kommunikation erfolgen und die Anlage ihren Zweck erreichen. Diese Kommunikationssysteme werden von technischer Seite oftmals als Bussysteme realisiert/umgesetzt. Bussysteme lassen sich anhand ihrer verschiedenen Ausprägungen von Merkmalen klassifizieren. Im folgenden Abschnitt werden zunächst die Merkmale von Bussystemen zum allgemeinen Verständnis dargestellt bevor auf die konkreten Ausprägungen des später eingesetzten Modbus-Bussystems eingegangen wird.

2.2.1.1 Informationsaustausch

Die Informationen über einen Prozess und dessen Zustand, werden auf höherer Ebene durch Daten und auf unterster Ebene durch einzelne Bits repräsentiert. Der Austausch dieser Daten zwischen den einzelnen Geräten findet in Form von Telegrammen statt. Ein Telegramm besteht grundsätzlich aus den zu übertragenden Daten und zusätzlich aus Informationen zur Übertragung. Die Daten werden vor der Übertragung in Rahmen, sogenannte data frames, eingeteilt, deren genauer Aufbau abhängig vom verwendeten Kommunikationsprotokoll innerhalb des Netzwerks ist [Schnell u. Wiedemann, 2006, S. 11f.]. Der Aufbau eines Telegramms am Beispiel des Modbus-Kommunikationsprotokolls wird in Abschnitt 2.2.3 erläutert und ist in Abb. 2.11 graphisch dargestellt.

2.2.1.2 Netzwerk und Topologie

Werden einzelne Prozesseinheiten miteinander über sogenannte Verbindungsleitungen, die zur Übertragung Informationen genutzt werden können, verknüpft, entstehen dabei Netzwerke und die Prozesseinheiten werden als Teilnehmer des Netzwerks bezeichnet. Ein Netzwerk lässt sich in einzelne Segmente einteilen und kann je nach Ausführung der Verbindungsleitungen und Anzahl der Teilnehmer unterschiedlich ausgeprägt sein. Anhand der geometrischen Anordnung lassen sich die folgenden, verschiedenen Netzwerktopologien unterscheiden.

Die einfachste Art, um eine Verbindung zwischen zwei Teilnehmern eines Netzwerks herzustellen, ist die sogenannte Zweipunktverbindung. Dazu sind die Netzwerkteilnehmer durch eine direkte Leitung miteinander verbunden. Jedoch steigt mit der Anzahl von Teilnehmern auch der Verbindungsaufwand überproportional an, um bei solchen vermaschten Netzwerk alle Teilnehmer miteinander zu verbinden. Dies hat für große, vermaschte Netze zur Folge, dass eine unübersichtliche große Anzahl von Schnittstellen, ein extrem hoher Verkabelungsaufwand und damit verbundene hohe Kosten entstehen. Um diese Kosten zu vermeiden, ergeben sich noch verschiedene andere Möglichkeiten zur Anordnung von Teilnehmern in Netzwerken [Schnell u. Wiedemann, 2006, S. 1f.].

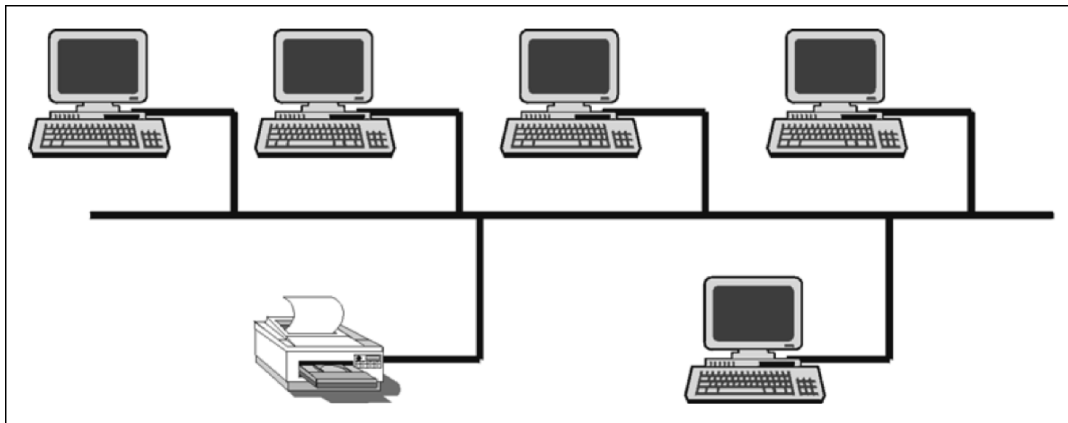


Abb. 2.1: Bus-Struktur aus [Schnell u. Wiedemann, 2006, S. 3]

Um dem hohen Verkabelungsaufwand zu vermeiden, wird bei großen Netzwerken zu einer Linienstruktur übergegangen, die auch als Bus-Struktur bezeichnet wird und in Abb. 2.1 visualisiert ist. Charakteristisch für die Bus-Struktur ist, dass alle Teilnehmer entlang einer langen Verbindungsleitung, dem sogenannten Buskabel, angeordnet sind. Sie sind mit Hilfe von kurzen Stichleitungen an das gemeinsame Buskabel angebunden, über das die gesamte Kommunikation im Netzwerk erfolgt. Durch diese Anordnung wird der Verkabelungsaufwand sowie die Anzahl an Schnittstellen, insbesondere für sehr große Netzwerke, stark reduziert. Jedoch wird durch Nutzung einer gemeinsamen Kommunikationsleitung die gleichzeitige Kommunikation von Teilnehmern erschwert und es müssen sogenannte Buszugriffsverfahren definiert werden, welche lediglich Regeln für Zugriff auf den Bus festlegen. Weiterhin müssen durch die Parallelschaltung alle Teilnehmer ständig alle Sendungen mitverfolgen, wodurch der Sender stark belastet wird. Die Busleitungslängen sind meist sehr lange¹ und da die Länge auf die zu übertragende Wellenlänge bezogen nicht mehr vernachlässigbar klein ist, müssen Reflexionen durch Leitungsabschlusswiderstände an den beiden Enden der Busleitung unterbunden werden. Außerdem werden die Leitungslängen und die Teilnehmer je Netzwerksegment begrenzt??S. 3f.]schn06.

Ein weiterer begrenzender Faktor für die Leitungslänge ist der Fakt, dass die maximale Übertragungslänge und die maximale Übertragungsrate miteinander verknüpft sind und sich gegenseitig beschränken. Der Leitungs- und Kapazitätswiderstand einer Leitung hängen von der Länge der Leitung ab und lassen sich durch das Ersatzschaltbild eines RC-Gliedes repräsentieren, wie in Abbildung Abb. 2.2 a) zu sehen ist. Durch die beiden Widerstände entsteht auf der Leitung eine Impulsverzerrung Δt_{Imp} , die somit mittelbar von der Leitungslänge abhängt. Je länger die Leitung wird, desto größer werden auch beiden Widerstände. Durch die erhöhte Leitungskapazität $C_{Leitung}$ erhöht sich die Ladezeit und gleichzeitig sinkt durch den erhöhten Leitungswiderstand $R_{Leitung}$ die Lastspannung U_G . Damit vergrößert sich die Impulsverzerrung Δt_{Imp} , wie in Abb. 2.2 b) und Abb. 2.2 c) dargestellt. Dadurch wird die maximale Frequenz

¹ Diese reichen von mehreren hundert Metern bis teilweise in den Kilometerbereich, je nach Art und Einsatzort der Anwendung.

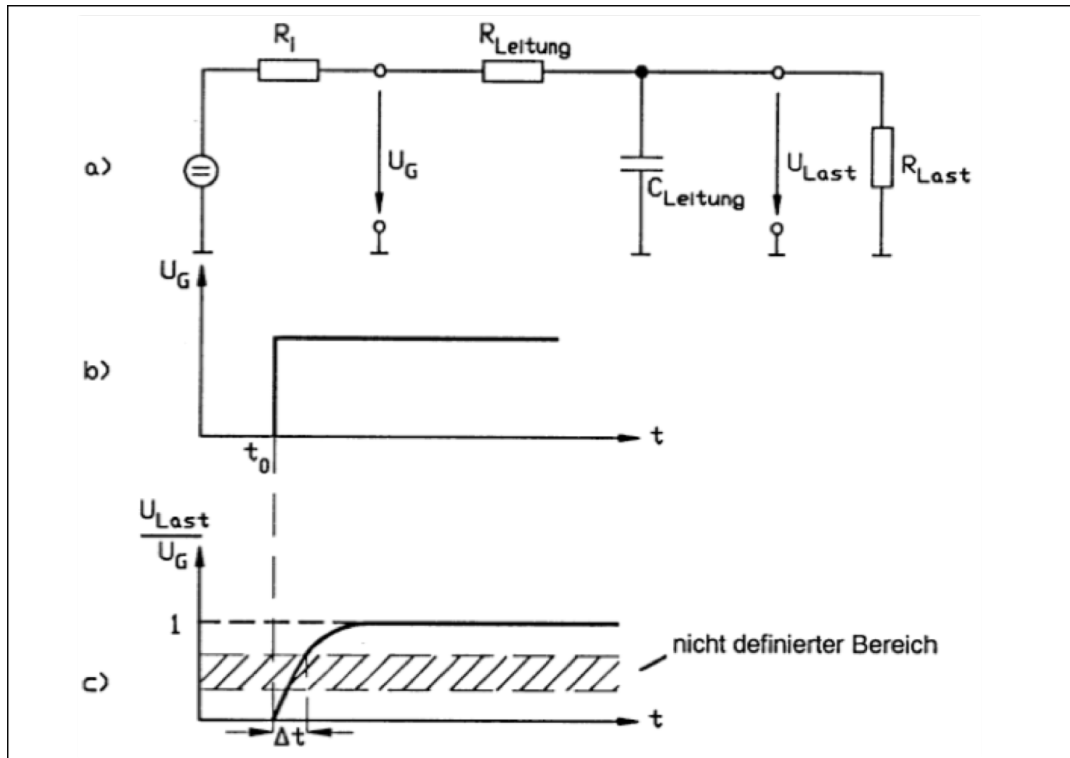


Abb. 2.2: Impulsverzerrung auf einer Leitung: a) Ersatzschaltbild der Anordnung b) Ausgangsspannung des Generators c) Empfängerspannung aus [Schnell u. Wiedemann, 2006, S. 4]

f_{max} der Datenübertragung auf den Kehrwert der Impulsverzerrung $f_{max} = \frac{1}{\Delta t_{imp}}$ beschränkt, da ansonsten der Empfänger den Wechsel des logischen Zustandes nicht mehr registrieren kann. In der Praxis bedeutet dies, dass die maximale Übertragungslänge und die maximale Übertragungsrate miteinander verknüpft sind und sich gegenseitig beschränken [Schnell u. Wiedemann, 2006, S. 4f.].

Um die Begrenzung der Leitungslängen zu korrigieren, wurde die Bus-Struktur zu einer Baumstruktur weiterentwickelt, welche in Abb. 2.3 dargestellt ist. Darin werden einzelne Netzwerk-Segmente, also einzelne Bus-Strukturen, durch Verstärkerelemente, sogenannte Repeater, zu einem großen Netzwerk verknüpft. Um damit jedoch größere Flächen als mit der Bus-Struktur zu vernetzen und gleichzeitig die maximale Leitungslänge und die maximale Anzahl der Busteilnehmer zu vergrößern, wird jedoch eine galvanische Trennung der Teilnehmer voneinander benötigt [Schnell u. Wiedemann, 2006, S.5 f.]. Die Besonderheit in dieser Struktur liegt also darin, dass sich durch Ihren Aufbau bestehende Bus-Strukturen auch nachträglich einfach erweitern oder miteinander verknüpfen lassen. Die Bauteile zur Erweiterung von Netzwerken werden im Abschnitt 2.2.1.5 Schnittstellen vorgestellt.

Weitere wichtige Netzwerk Topologien, die für das Verständnis dieser Arbeit keine weitere Relevanz haben, sind die Ring- und die Stern-Struktur. Die Ring-Struktur ist durch einen physikalischer Ring von Zweipunktverbindungen aufgebaut und gekennzeichnet durch die Kommunikation der Teilnehmer übereinander hinweg. Die Stern-Topologie hingegen ist um eine Zentralstation herum ausgebaut, die mit al-

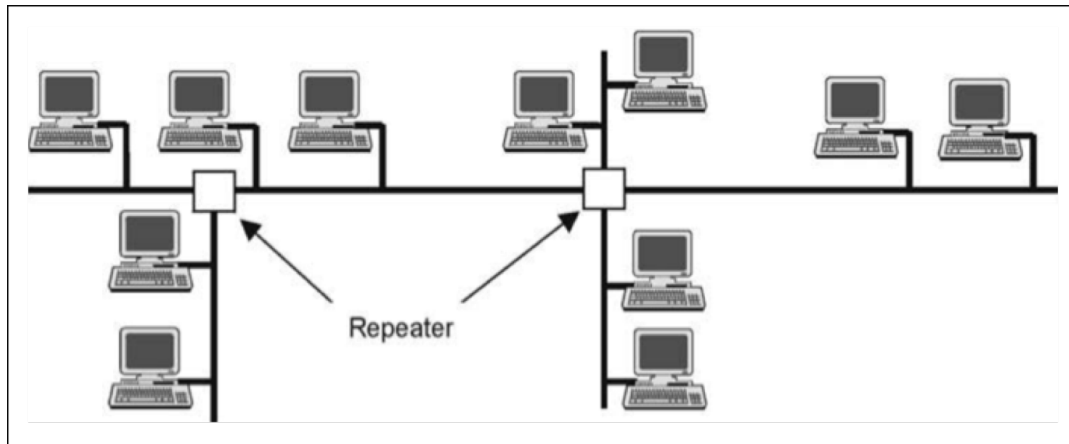


Abb. 2.3: Baumstruktur aus [Schnell u. Wiedemann, 2006, S. 5]

len Teilnehmer verbunden ist und über die die gesamte Kommunikation abläuft. Der interessierte Leser findet in [Schnell u. Wiedemann, 2006, S. 6f.] detailliertere Ausführungen.

2.2.1.3 Buszugriffsverfahren

Die meisten Netzwerktopologien kommunizieren gemeinsam über eine Verbindungsleitung. Daher werden Regeln für den Zugriff definiert, um eine reibungslose Kommunikation zu ermöglichen. Die Buszugriffsverfahren lassen sich in zwei Gruppen aufteilen, der kontrollierten und zufälligen Verfahren [Schnell u. Wiedemann, 2006, S. 19].

Bei den kontrollierten Verfahren ist der Sender bereits vor Sendebeginn eindeutig bestimmt und eine Zuteilung des Busses ist nicht notwendig. Der Buszugriff findet entweder zentral innerhalb einer Zentralstation statt, bei sogenannten Master/Slave-Verfahren, oder wird dezentral durch Steuereinheiten vorgenommen, wie z.B. beim Tokenring und Tokenbus. Ein solches Verfahren heißt echtzeitfähig, wenn die Zykluszeit zur Datenübertragung berechenbar ist, aufgrund einer Beschränkung der Länge des Übertragungsintervalls und der maximalen Datenlänge. Bei zufälligen Buszugriffsverfahren greifen die Teilnehmer bei Bedarf auf die Verbindungsleitung zu und müssen sicherstellen, dass diese nicht gerade von einem anderen Teilnehmer belegt ist. Da nicht vorhergesehen werden kann, an welchem Zeitpunkt Informationen übertragen werden kann keine Echtzeitfähigkeit erreicht werden [Schnell u. Wiedemann, 2006, S. 19].

Das Master/Slave-Verfahren besteht in der Regel aus einer Bussteuerungseinheit, dem sogenannten Master, und mehreren passiven Teilnehmern, den Slaves. Die Kommunikation wird ausschließlich vom Master initiiert, der die Verbindung zu den Slaves aktiv durch ein Request herstellt, in welchem die angeforderten Daten spezifiziert sind. Die Slaves treten nur nach Anfragen in Aktion und antworten darauf unmittelbar mit einer Response, die die angeforderten Daten des Masters enthält. In der Regel erfolgt die Kommunikation zyklisch zu allen Slaves gleichzeitig (Polling), damit

der Master ein umfassendes und aktuelles Bild über den Systemzustand bekommt. Dadurch ergeben sich einfache Slaves, die günstig in den Bus eingebunden werden können, weil die gesamte „Intelligenz“ im Master implementiert ist. Jedoch gilt es bei diesem Verfahren zu beachten, dass der Informationsaustausch zwischen verschiedenen Slaves längere Zeit in Anspruch nehmen kann und bei einem Ausfall des Masters das gesamte Bussystem stillliegt [Schnell u. Wiedemann, 2006, S. 19ff.]. Auf eine weitere Beschreibung der übrigen Verfahren wird aufgrund der fehlenden Relevanz für diese Arbeit verzichtet, der interessierte Leser findet jedoch bei Schnell u. Wiedemann [2006] im Kapitel Buszugriffsverfahren ausführliche Informationen.

2.2.1.4 Datensicherung

Bei der Übertragung von Informationen besteht die Gefahr von Störungen, welche sich als Fehler in einer Nachricht durch Invertierung von Bits äußern. Störungen sind in der Regel technischer Art, wie zum Beispiel elektromagnetische Störsignale, Rauschen oder Potentialdifferenzen, und gegen einen Großteil lassen verschiedene Vorkehrungen treffen. Dadurch ergibt sich die Möglichkeit Störungen vorzubeugen oder diese nach Einsatz zu beseitigen. Der erste Ansatz ist also eine Verminderung des Auftretens durch technische Vorkehrungen, wie zum Beispiel Schirmung der Kabel, galvanische Trennung von Netzwerken oder differenzielle Signalübertragung. Der zweite Ansatz beschäftigt sich mit der Überwachung des Nachrichtenverkehrs und dem Ausbessern/Gegenmaßnahmen bei Fehlern [Schnell u. Wiedemann, 2006, S. 30].

Auf dem Gebiet der Buskommunikation entspricht wird eine Information oder Nachricht oder Telegramm codiert. Es werden stets Transparente Codes betrachtet, Welche bitorientiert sind. Dabei sind jegliche Kombination von Bits erlaubt und man kann allein Aus der Folge der Blitz nicht auf einen Fehler schließen. Definition Telegramm und Bit(0 und 1)

Die Vorkehrungen technischer Art Können Sind in aller Regel in den Spezifikationen der einzelnen Bussysteme enthalten, daher Werden diese nicht weiter beschrieben und lediglich die Überwachung und Ausbesserung das Nachrichtenverkehrs erläutert. Bei der Übermittlung von Nachrichten können drei Arten von Fehlern auftreten: Der Fehler kam erkennbar und korrigierbar sein, Erkennbar und nicht korrigierbar Oder nicht erkennbar und damit auch nicht korrigierbar.

Kann der Fehler erkannt werden ist bereits ein großer Teil der Arbeit getan.

Fehlermaße sind die Bitfehlerrate $p = \text{Anzahl fehlerhafter bits} / \text{gesamtzahl gesendete bits}$, schlechter Wert ist $p=0,5$, da durch invertierung immer wieder umstellbar. üblich in Technik $p = 10^{-4}$. ARQ(Error detection and automatic request repeat) ist normale Reaktion auf erkannte Fehler, dass eine einfache Wiederholung der Übertragung.

Restfehlerrate R ist eine wichtige Kennzahl weil, sie die unerkannten, fehlerhaften Bitfolgen die nach der Anwendung von Fehlererkennungstrategien noch verbleiben

misst. R = Anzahl unerkannt fehlerhafter Bitmkombinationen/Gesamtlänge in Bits der Information. \rightarrow Maß für Unversehrtheit der Daten

Telegrammeffizienz auch wichtig, weil sie eine Aussage erlaubt wie viel Informationen transportiert werden können und sie steht im Gegensatz zur Restfehlerquote, je sicherer Übertragung desto weniger Effizienz. E = fehlerfreie Infobits/Gesamtzahl übertragene bits (incl. Adresse Error check usw.) [Schnell u. Wiedemann, 2006, S. 31ff.]

Die genauen Berechnungen der Wahrscheinlichkeiten und der mittleren Zeit zwischen zwei Fehlern sind und noch viel mehr Details finden sich im Kapitel Datensicherung in [Schnell u. Wiedemann, 2006, S. 31f.] zu finden.

Um Fehler systematisch zu erkennen, existieren verschiedene Fehlererkennungsstrategien. Die einfachste ist der Paritätsbit, der lediglich die Quersumme des Telegramms angibt, $P=0$ für eine ungerade und $p=1$ für eine gerade Quersumme. Damit können diejenigen Fehler entdeckt werden, die eine ungerade Anzahl an Bitflips haben, die mit geraden leider nicht. Eine Erweiterung der Paritätssicherung ist die Blocksicherung, bei der die Paritäten über ein Array aus mehreren Telegrammen überprüft werden kann [Schnell u. Wiedemann, 2006, S. 34f.].

Beim sogenannten Cyclic Redundancy Check, in der Literatur häufig als CRC-Check bezeichnet, wird ein Telegramm als Zahl aufgefasst. Im Sender wird diese Zahl durch das Generatorpolynom G geteilt. Das Ergebnis wird verworfen, lediglich der Rest bei der Division wird an das Telegramm angehängt. Der Empfänger dividiert das empfangene Telegramm durch dasselbe Polynom G und falls sich ein Rest von 0 ergibt, war die Übertragung fehlerfrei. Dadurch lassen sich abhängig vom Generatorpolynom G unterschiedliche Güten der Fehlererkennung realisieren. In Abb. 2.4 sind die Vorgänge beim Cyclic Redundancy Check graphisch zusammengefasst.

2.2.1.5 Schnittstellen

Die physikalische Übertragung der Daten/Telegramme kann über verschiedenste Schnittstellen geschehen und erfolgt binär. Je nach Topologie des Bussystems und Protokoll, kann die Datenübertragung seriell oder parallel geschehen wie in Abb. 2.5 zu sehen. Bei einer parallelen Datenübertragung werden immer mehrere Bits gleichzeitig übertragen, weshalb eine hohe Übertragungsgeschwindigkeit erreicht werden kann, jedoch eine aufwändige im Sinne eines vermaschten Netzes notwendig ist. Daher erfolgt in der Praxis eher eine serielle Datenübertragung, also die einzelne Übertragung von Bits nacheinander über eine Leitung, wie sie zum Beispiel mit einer Bus-Struktur möglich ist [Schleicher, 2008, S. 13].

Bei der binären Datenübertragung werden lediglich zwei Zustände unterschieden, deren Signalwerte einen High-Pegel p_{high} und einen Low-Pegel p_{low} besitzen, die durch einen Bereich, in dem das Signal nicht definiert ist, voneinander getrennt sind [Schleicher, 2008, S. 9]. Die Geschwindigkeit der Datenübertragung wird in der Einheit Baud gemessen und definiert als übertragene Bits pro Sekunde (Bps) [Schleicher, 2008,

CRC (Cyclic Redundancy Check)	
1. Die Nachricht sei I	
Beispiel dezimal: I = 14	binär: I = 110101
2. Das Prüfpolyinom sei G	
$G = \dots a_4 \cdot x^4 + a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_0 \cdot x^0$	
d.: $\dots 0 \cdot x + 3 \cdot x^0 = 3$	b.: $\dots 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x + 1 \cdot x^0 = 1011$
3. Der Hilfsvektor ist dann H = 100 ... 0	
d.: H = 10	b.: H = 10000
4. Die Information I wird mit H multipliziert:	
$B = I \cdot H$	
d.: B = 140	b.: B = 1101010000
5. Das Produkt B wird durch G dividiert:	
$\frac{B}{G} = Q + \frac{R}{G}$	
d.: $\frac{B}{G} = \frac{140}{3} = 46 + \frac{2}{3}$	b.: $\frac{B}{G} = \frac{1101010000}{1011}$
	$= 1111011 + \frac{101}{1011}$
6. Der Rest R wird B hinzuaddiert und das Ganze gesendet	
d.: B + R = 142	b.: B + R = 1101010101
7. Der Empfänger bildet	
$(B - R) : G$	
d.: $(B - R) : G = 138 : 3$	b.: $(B - R) : G$
	$= (B + R) : G = 1101010101 : 1011$
$= 46, R = 0$	$= 1111011, R = 0$
8. Es bedeutet R = 0 fehlerfreie Übertragung	

Abb. 2.4: Cyclic Redundancy Check aus [Schnell u. Wiedemann, 2006, S. 38]



Abb. 2.5: Parallele und serielle Datenübertragung aus [Schleicher, 2008, S. 13]

S. 22].

Die gängigsten Übertragungsverfahren sind elektrische und optische Schnittstellen. Die elektrischen Schnittstellen gliedern sich wiederum in Strom- und Spannungsschnittstellen auf. Im weiteren Verlauf der Arbeit sind lediglich die Spannungsschnittstellen EIA² 232 und EIA 485, welche auch als EIA 232 und EIA 485 bezeichnet werden, relevant, weshalb der interessierte Leser weitere Informationen zu weiteren Schnittstellen in [Schleicher, 2008, S. 13ff.] und [Schnell u. Wiedemann, 2006, S. 57ff.] findet.

Die EIA 232 Schnittstelle ist wie bereits erwähnt eine Spannungsschnittstelle, die für Punkt-zu-Punkt Verbindungen geeignet ist und deren Pegel in Abb. 2.6 zu sehen sind. Die Pegel sind für Spannungen zwischen $-3V < p_{high} < -15V$ als logische „1“, der Low-Pegel für Spannungen zwischen $-3V < p_{low} < -15V$ als die logische „0“ definiert. Im Intervall $[-3V, 3V]$ ist das Signal nicht definiert, weshalb dieser Bereich möglichst schnell durchlaufen werden sollte. Da der Signalpegel von der Datenleitung hin zur Masse gemessen wird kann er nicht symmetrisch sein und ist damit erdunsymmetrisch. [Schnell u. Wiedemann, 2006, S. 57f.]

Typischerweise steht eine solche Schnittstelle jedem Rechner als COM-Port zur Verfügung und in der Automatisierungstechnik werden nur die RxD (Receive Data), TxD (Transmit Data) und GND Leitungen, die ein gemeinsames Bezugspotenzial definiert, verwendet. Wichtig bei der Verkabelung ist, dass die übertragende Leitung TxD immer mit der empfangenden Leitung RxD verbunden ist [Schleicher, 2008, S. 14f.].

Die EIA 485 Schnittstelle ist ebenfalls eine Spannungsschnittstelle, die jedoch für

² Abkürzung der Normen und Standards die von der Electronic Industries Alliance entwickelt wurden.

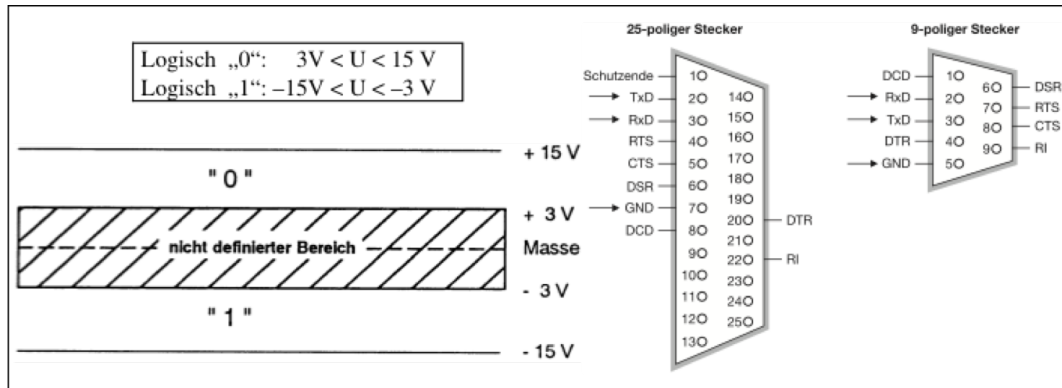


Abb. 2.6: Links: Spannungspegel EIA 232-Schnittstelle aus [Schnell u. Wiedemann, 2006, S. 57]
 Rechts: Stecker EIA 232-Schnittstelle aus [Schleicher, 2008, S. 14]

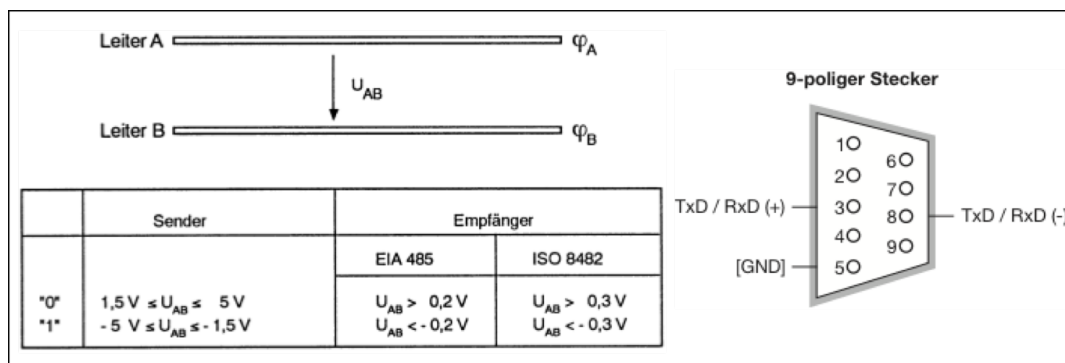


Abb. 2.7: Links: Spannungspegel EIA 485-Schnittstelle aus [Schnell u. Wiedemann, 2006, S. 60]
 Rechts: Stecker EIA 485-Schnittstelle aus [Schleicher, 2008, S. 19]

Mehrpunktverbindungen geeignet ist und die in der Norm ISO 8482 beschrieben ist. Die Signalübertragung erfolgt über zwei Übertragungsleitungen, die in der Regel als verdrehtes und abgeschirmtes Zweidrahtleitung ausgeführt sind. Die Signalpegel entsprechen der Differenzspannung U_{AB} zwischen den beiden Leitungen, die innerhalb des Intervalls von $[-7V, 12V]$ bezogen auf die Masse liegen muss. Durch das verdrehte Leitungspaar, wirken sich mögliche Störgrößen auf die Spannung beider Leitungen gleichermaßen aus, wodurch die Spannungsdifferenz unverändert bleibt und eine erhöhte Störfestigkeit gegenüber der EIA 232-Schnittstelle erreicht wird. Bei der Pegelfestlegung werden für Empfänger und Sender gibt es verschiedene Vorgaben, wie in Abb. 2.7 graphisch dargestellt. So müssen die Sender eine Differenzspannung zwischen $-1,5V < U_{AB} < -5V$ für logische „1“ und $1,5V < U_{AB} < -5V$ für logische „0“ leisten können. Empfänger hingegen müssen in der Lage sein Spannungsdifferenzen von $U_{AB} < -0,3V$ als logische „1“ und $0,3 < U_{AB}$ als logische „0“ zu detektieren und zu interpretieren [Schnell u. Wiedemann, 2006, S. 59ff.].

Es besteht ein geringer Installationsaufwand und die Datenübertragung erfolgt über die TxD/RxD + und TxD/RxD - Leitungen, ein gemeinsames Bezugspotential über die GND Leitung wird nicht zwingend benötigt, jedoch wird ein Leitungsabschluss an beiden Enden des Buskabels benötigt. [Schleicher, 2008, S. 19f.].

Um Netzwerk zu erweitern, können verschiedene Bauteile eingesetzt werden. Sogenann-

te Repeater sind aktive Bauteile, die lediglich eine kurze Zeitverzögerung verursachen und nur Netzwerke der selben Schnittstelle miteinander verbinden kann. Er ist ein aktives Bauteil, das das Datensignal lediglich verstärkt, indem er die empfangenen Bits blind kopiert und verstärkt auf das angeschlossene Netzwerk überträgt, weshalb er für die Kommunikationsteilnehmer unsichtbar ist [Schnell u. Wiedemann, 2006, S. 79f.].

Eine Möglichkeit um Netzwerke verschiedener Art zu verbinden bieten Bridges und Gateways. Erstere werden auch als Schnittstellenumsetzer bezeichnet und kommen zum Einsatz, wenn trotz gleichem Übertragungsprotokoll genutzt unterschiedliche physikalische Schnittstellen oder Übertragungsmedien genutzt werden sollen [Schnell u. Wiedemann, 2006, S. 80f.]. Er erkennt die Datenflussrichtung automatisch und wandelt die Pegel der Schnittstellen in den jeweils anderen um [Schleicher, 2008, S. 21]. Die sogenannten Gateways dienen der Kopplung von Netzwerken, die verschiedene Architekturen aufweisen, also neben unterschiedlichen physikalischen Schnittstellen auch andere Übertragungsprotokolle verwenden. Das Gateway ist demnach umfassender als die Bridge und erweitert deren Funktionen um die Übersetzung der Signale von einem Übertragungsprotokoll in das jeweils andere [Schnell u. Wiedemann, 2006, S. 84f.].

2.2.2 OSI-Kommunikationsmodell

Aufgrund der großen Anzahl verschiedener technischer Systeme existieren auch viele verschiedene Arten der Kommunikation untereinander. Bei der genaueren Betrachtung der Kommunikation wird ersichtlich, dass diese oftmals ähnlich abläuft und sich durch ein Meta-Schema beschreiben lässt [Schnell u. Wiedemann, 2006, S. 8]. Um die Kommunikation auch über verschiedenen Systeme hinweg zu ermöglichen und sie zu formalisieren, wurde von der *International Organization for Standardization* 1984 ein abstraktes Referenz-Modell entwickelt, das in der *ISO-Norm 7498-1* beschrieben ist. Es dient der Entwicklung und Verbesserung von Standards für den Informationsaustausch sowie als Referenz für bestehende Standard um eine gewisse Konsistenz zu wahren [osi, 1996, S. 1]. Das Ziel bei dem Entwurf des Modells war es, eine Menge von Standards zu schaffen um autonomen Systemen die Kommunikation untereinander zu ermöglichen [osi, 1996, S. 4].

Das sogenannte Open System Interconnection Modell wird zunächst allgemein erläutert, da die Kommunikation von technischen Systemen im Rahmen der Arbeit eine zentrale Rolle spielt, und wird anschließend im Anwendungskontext mit den eingesetzten Protokollen und Schnittstellen referenziert.

Zunächst wird im Standard definiert, womit sich das Modell beschäftigt und abgegrenzt welche Aspekte im Modell keine Berücksichtigung finden [osi, 1996, S. 3]:

*„OSI is concerned with the exchange of information between open systems
(and not the internal functioning of each individual real open system).“*

Das OSI-Modell beschäftigt sich also zentral mit dem Austausch von Informationen zwischen verschiedenen offenen Systemen und allen dabei anfallenden Aktivitäten. Diese sind sehr umfangreich und lassen sich in folgende Bereiche gliedern [osi, 1996, S. 3f.]:

- Der Austausch von Informationen zwischen offenen Systemen,
- die physischen Medien zur Verbindung von offenen Systemen und deren Transportmöglichkeit von Informationen,
- die Vernetzung von offenen Systemen,
- die Interaktion zwischen offenen Systemen und deren Fähigkeit zur Kooperation bei der Datenübertragung.

Bezogen auf den Austausch von Informationen überschneiden sich die physische Verbindung und die Vernetzung und entsprechen zusammen der Infrastruktur und deren Architektur, die zur Übertragung zur Verfügung steht. Die Interaktion umfasst weitaus mehr Aufgaben: Neben der Synchronisation der Prozesse, die Daten austauschen wollen, muss auch die Darstellung der auszutauschenden Daten und eventuell notwendige Transformationen beachtet werden, um eine Kompatibilität unterschiedlicher Systeme zu erreichen. Weitere wichtige Aufgaben sind die Datenspeicherung, deren Integrität und die Sicherheit beim Austausch hinsichtlich Fehler und Einsicht von Außen [osi, 1996, S. 4]. Es ist leicht zu erkennen, dass die technische Kommunikation einen sehr umfangreichen und komplizierten Prozess darstellt. Daher wird der Kommunikationsprozess im OSI-Modell stark abstrahiert und in sieben abstrakte Ebenen gegliedert. Die einzelnen Ebenen sind in Abb. 2.8 dargestellt und dienen dazu verschiedene Aufgaben des Kommunikationsprozesses in Teilaufgaben zusammenzufassen.

Die Ebenen werden Schichten genannt und haben klar definierte Aufgaben und Schnittstellen zu ihren Nachbarschichten. An diesen Schnittstellen werden Dienste bereitgestellt, die von den anderen Ebenen genutzt werden können. Durch diesen Aufbau können einzelne Schichten einfach bearbeitet oder ausgetauscht werden, ohne die Gesamtfunktionalität zu gefährden. Außerdem kann ein System auch aus Komponenten verschiedener Hersteller zusammengesetzt werden, womit diese Architektur nachweislich als Basis für offene Systeme dient. In Abb. 2.8 ist ebenfalls dargestellt, dass die Schichten eins bis vier auch als Übertragungsschichten beziehungsweise Transportsystem zusammengefasst werden, weil sie für die Datenübertragung zwischen Systemen als gemeinsame Aufgabe haben. Die Schichten fünf bis sieben werden als Anwendungsschichten bezeichnet weil sie bei der Datenübertragung die Zusammenarbeit zwischen der Anwendersoftware und dem Betriebssystem sicherstellen [Schnell u. Wiedemann, 2006, S. 8f.].

Die Schnittstellen/Dienste zwischen den Schichten werden als Service Access Points bezeichnet und besitzen jeweils eine eindeutige Adresse, die oberhalb liegende Schicht ist der Service user, da er den Service daer unterhalb liegenden Schicht nutzt, dem service provider. Die Dienste können in verbindungsorientierte und verbindungsunabhängige

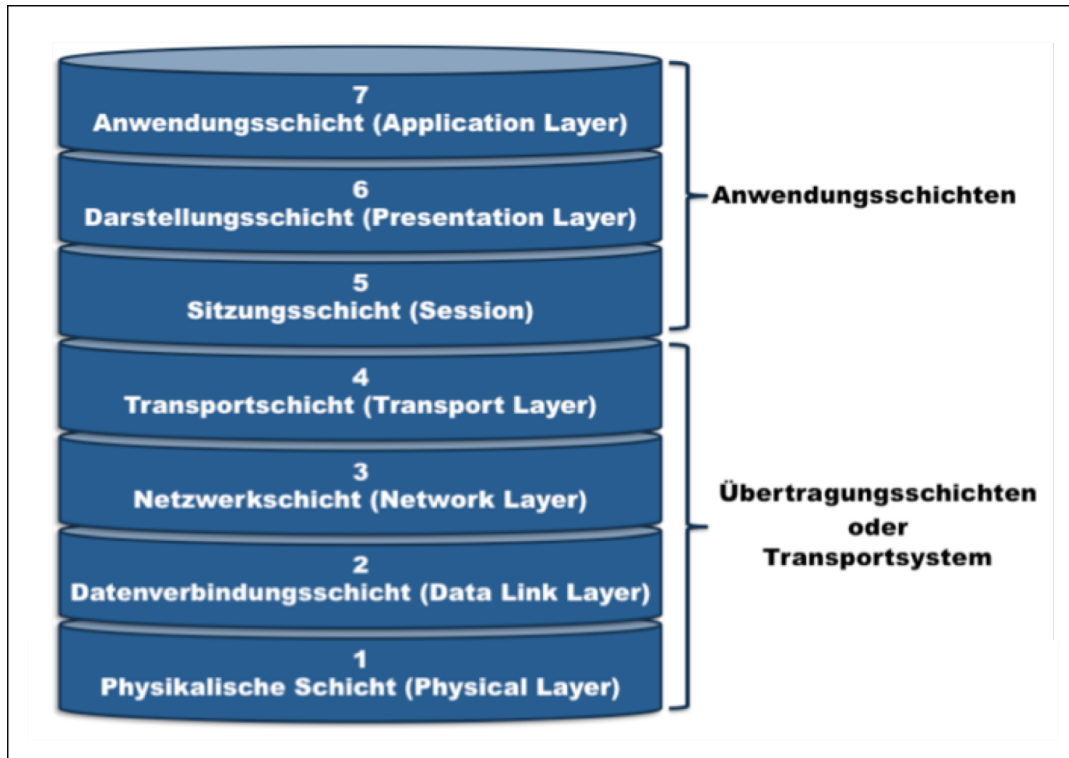


Abb. 2.8: Die sieben Schichten des *Open System Interconnection* Modells verändert nach [Schnell u. Wiedemann, 2006, S. 10] und [osi, 1996, S. 28]

unterschieden werden. Für den Datenaustausch stehen folgende Dienste zur Verfügung. Bei der Abhandlung der Dienstaufgaben stehen vier Dienstvorgänge zur Verfügung, die zusammengefasst in Abb. 2.9 abgebildet sind:

request - Anforderung indication - Meldung response - Antwort confirmation - Bestätigung. Bestätigten Diensten stehen alle vier Vorgänge zur Verfügung, unbestätigten lediglich die Anforderung und Meldung. Typische Dienste sind Connect, disconnect, data.

[Schnell u. Wiedemann, 2006, S. 14f.].

Im Folgenden wird kurz auf die einzelnen Schichten von Unten nach oben eingegangen, bevor das Zusammenwirken der einzelnen Schichten anhand eines Beispiels verdeutlicht wird.

Die erste, physikalische Schicht stellt die mechanischen und elektrischen Möglichkeiten zur physischen Verbindung von Systemen zur Verfügung, um die Datenübertragung der einzelnen Bits zu ermöglichen [osi, 1996, S. 49f.]. Sie legt also die mechanischen und elektrischen Eigenschaften der Übertragung fest, also die Endsystemkopplung (Stecker), die Kabelspezifikationen und die Zuordnung der Anschlüsse sowie die Art der Codierung und die Spannungspegel zur Übertragung. In der Regel werden dazu bestehende Normen genutzt, wie zum Beispiel die elektrischen Übertragungsstrecke nach EIA 485-Norm, welche im Folgenden noch erläutert wird.

Ein wichtiger Aspekt der Schicht ist es, dass die Spezifikation der Strecke und nicht

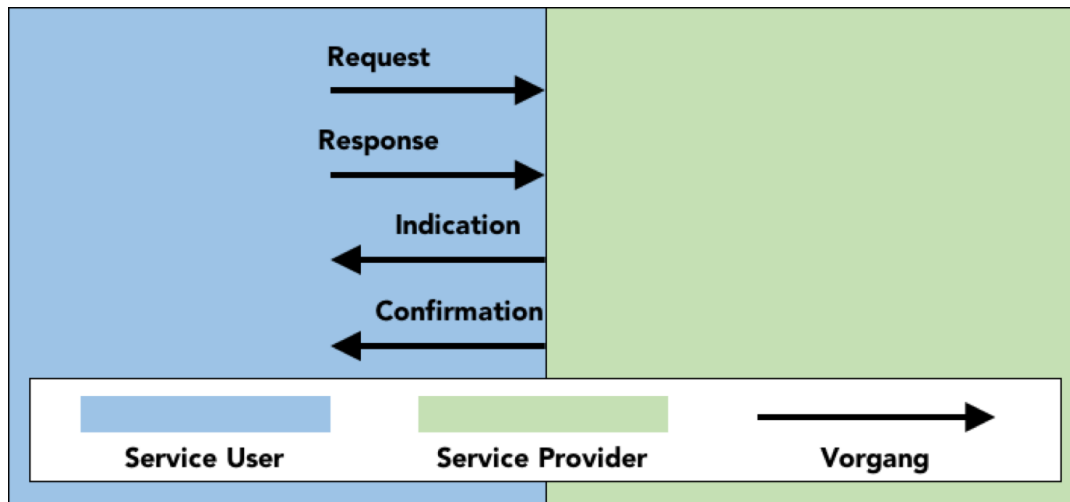


Abb. 2.9: Die vier Dienstvorgänge

das physikalische Medium selbst Teil der Schicht eins ist, denn die Kommunikation ist unabhängig von der konkreten Ausprägung der Schicht [Schnell u. Wiedemann, 2006, S. 9].

Die zweite Schicht betrachtet Kommunikation zwischen zwei Systemen. Deshalb stellt die Datenverbindungsschicht, stellt funktionale und prozedurale Möglichkeiten für den Verbindungsaufbau/-trennung/-erhaltung und den Transfer von Dateneinheiten Verfügung. Ermöglicht dem Netzwerkschicht die Kontrolle über die Verbindung von Data circuits physikalisch, sowie Fehlerabfangen der physikalischen Schicht [OSI, 1996, S. 46f.] Aufgabe ist sicherer Transport von Station zu Station. Datensicherung → Verpacken um Übertragungsfehler erkenntlich zu machen in Data Frames. In Frames sind die maximale Anzahl Datenbits für Rohdaten spezifiziert, weiterhin wird Information zur Übertragung hinzugefügt. Die Zusatzinfo kann Prüfsumme und Anfang und Ende des Rahmens enthalten oder Quittierung eines Telegramms und dient dazu fehlerhafte Übertragung oder etwas verloren gegangen zu überprüfen. MAC mit Schicht eins, LLC mit Schicht drei. Wuelle Wiki bisher MAC regelt den Zugriff auf das physische Medium zur Kommunikation, kontrolliert oder konkurriert. LLC verteilt die Daten passend in Schicht drei und gibt die Daten von Schicht drei an passende MAC für Schicht eins weiter und fügt diese Infos von oben hinzu (Adressen Empfänger und Sender und Zusatzinfo wie Control für Steuerung von Datenfluss oder so).

Wichtig Die Schicht hat jedoch keine Kenntnis über Inhalte der Daten! [Schnell u. Wiedemann, 2006, S. 9ff.]

Verbindungsloser Dienst heisst keine Verbindung zwischen Kommunikationspartnern, Datenpakete werden wie Brief ganz in Netzwerk gespeist mit Zieladresse versehen und weitertransportiert, ohne Beeinflussung des Transportweges durch Benutzer des Netzwerkdienstes. Später ist Modbus ein Beispiel dazu erklärt. Verbindungsorientierte Dienste heisst ein virtueller Kanal zwischen Kommunikationspartnern wird zur Verfügung gestellt, eingerichtet: Verbindungsaufbau, Datenaustausch, Verbindungsabbau, wie Telefongespräch. [Schnell u. Wiedemann, 2006, S. 11f.]

Die dritte Schicht beschäftigt sich mit dem Netzwerk als Ganzes. Die Aufgaben der Netzwerkschicht hängen ein wenig ab von verbindungsorientierung. Daher beschäftigt sie sich mit dem Aufbau, der Erhaltung und Datenaustausch und dem Trennen von Netzwerk-Verbindungen zwischen offenen Systemen im Netzwerk, also Schnittstellen. Weiterhin ist sie für den Transport von Daten im Netzwerk zuständig, also insbesondere auch für die Festlegung der Route (Wegsteuerung) der Daten im Netzwerk [osi, 1996, S. 41f.]. Also Kontrolle von Verkehr im Netzwerk, d.h. Anzahl Pakete im Netzwerk, Status [Schnell u. Wiedemann, 2006, S. 11f.]

Die vierte Schicht ist die Transportschicht und zuständig für die transparente Übertragung von Daten zwischen Prozessen und ist völlig unabhängig bzw. losgelöst von den Gedanken an Kosten und Verlässlichkeit der Datenübertragung, da dies Aufgaben der unteren Schichten sind. Sie kümmert sich um die optimale Nutzung von Netzwerk-Services/Nutzung [osi, 1996, S. 37f.]. Adressierung der Teilnehmer, Aufbau und Abbau für Transportverbindung zwischen Kom.-Partner-Prozessen (Sammel Einzel Mehrere), Fehlerbehandlung Verbindung und Flusskontrolle, Synchronisierung der Datenaustauschenden Prozesse. Zerlegung der Daten aus Sitzungsschicht in Transportierbare Einheiten. Internetworking, Umsetzung verschiedener Protokolle Gateway Aufgaben. Aufbau Verbindung legt Art fest, Punkt zu Punkt oder Broadcast/Multicast (Alle bzw. einige Teilnehmer gleichzeitig) [Schnell u. Wiedemann, 2006, S. 12f.].

Die fünfte Schicht, die Sitzungsschicht, startet eine Sitzungsverbindung mit bestimmter Adresse wenn dieser Prozess von einer höheren Schicht angefordert wird. Diese Verbindung dient dazu, den Dialog von kooperierenden Prozessen auf der eines höheren Darstellungsebene durch eine Sitzungsverbindung zu synchronisieren und deren Datenaustausch zu organisieren. Verknüpft die Sitzungsadressen mit den Transportadressen, also die Anwendungsschichten mit dem Transportsystem [osi, 1996, S. 35]. Benutzung des Transportsystems über die Schnittstelle zur Transportschicht. Je nach Funktionen der höheren Schichten entsprechender Funktionsumfang BCS Basic Combined Subset - Verbindungssteuerung und Datenübertragung BAS Basic Activity Subset - Aktivitätsverwaltung BSS Basic synchronized Subset - Synchronisierung [Schnell u. Wiedemann, 2006, S. 13].

Die sechste Schicht, die Darstellungsschicht ist nach [osi, 1996, S. 33f.] für die Darstellung der Daten die von Anwendung-Entitäten entweder kommuniziert oder bei deren Kommunikation referenziert werden. Sie stellt außerdem eine gemeinsame Repräsentation der übertragenen Daten dar zwischen Anwendungs-Entitäten und befreit diese dadurch von Syntaxabhängigkeiten. [Schnell u. Wiedemann, 2006, S. 13f.] stellt fest, dass die Dienste die der Darstellung der transferierten Daten dienen wie die Codierung der zu übertragenden Daten, der verwendete Zeichensatz und die Darstellung der Daten auf dem Bildschirm oder Drucker. Semantik/Syntax beim Nachrichtenaustausch und der beiden Kommunizierenden Prozesse. Evtl. Komprimierung um Zeit und Kosten zu sparen.

Die siebte und letzte Schicht stellt lediglich eine Möglichkeit für Anwendungsprozesse

se zur Verfügung um auf die OSI Umgebung zuzugreifen. Jeder Anwendung stellt im OSI genau einen Anwendungsprozess dar, verschiedene Anwendungsprozesse für verschiedene Anwendungen und vice versa [osi, 1996, S. 32] stellt Funktionen bereit, mit denen der Benutzer auf das Kommunikationssystem zugreifen kann, wobei der Benutzer idR ein Computerprogramm und kein Mensch ist. [Schnell u. Wiedemann, 2006, S. 14]

Im folgenden werden die verwendeten Modbus Protokolle und Spezifikation in Anwendung Bezug zum OSI Modell gebracht um den praktischen Nutzen davon klar zu machen.

2.2.3 Modbus Kommunikationstechnologie

Zunächst erfolgt die Einordnung ins OSI Modell und anschließend eine Klassifizierung der Spezifikationen gemäß der zuvor von Bussystemen.

Das Modbus Protokoll teilt sich auf verschiedene Protokolle auf, zum einen auf das Application Layer Messaging Protocol auf oberster Ebene, welches auf das Modbus Over Serial Line Protocol sowie das Ethernet TCP/IP Protokoll aufbaut. Das Application Layer Messaging Protocol lässt sich im OSI Referenzmodell in die siebte und oberste Schicht (Anwendungsschicht) einordnen wie in Abb. 2.10 dargestellt. Bei der Nutzung des Modbus over Serial line Protocol implementiert diese die zweite Schicht und die Ebenen drei bis sechs sind leer implementiert. Als physikalische Schicht werden die Übertragungsstandards nach EIA 485 oder nach EIA 232 [mod, 2006b, S. 2]. Anstatt des Over serial line kann auch das Modbus Messaging On TCP/IP Protocol verwendet, dann implementiert dieses zusammen mit den dem TCP/IP Standard zusammen die Netzwerkschicht im OSI-Referenzmodell. Das Ethernet implementiert dabei die Datenschichtung Nummer zwei und dessen physikalische Spezifikationen die unterste physikalische Schicht und deren Schnittstellen. Die Ebenen vier bis sechs sind auch bei Nutzung dieser Kommunikationsweise leer implementiert [mod, 2012, S. 2f.]. Eine Einführung zum Ethernet und TCP/IP Standard findet sich in Schnell u. Wiedemann [2006], detaillierte Ausführungen dazu in Furrer [2003].

Nun wird das Modbus Application Layer Messaging Protocol erläutert. Anschließend werden die Modbus Protokolle der unteren Ebenen erläutert bis hin zu den physikalischen Schnittstellen.

Das Modbus Application Layer Messaging protocol ist ein Protokoll, dass verschiedene Netzwerke und Bussysteme zur Master/Slave Kommunikationen von verbundenen Geräten nutzen kann [mod, 2012, S. 2f.], jedoch übernimmt im Rahmen des Protokolls der Master die Rolle des Clients und die Slaves die Rollen als Server, da der Client alleinig die Anfragen stellt und die Server lediglich auf Anfragen antworten. Das Modbus Protokoll definiert eine gemeinsame Telegrammstruktur, bezogen auf Inhalte und Rahmen der Nachricht. Damit ermöglicht es die Kommunikation zwischen verschiedenen Geräten innerhalb eines Netzwerks, welche unabhängig von Art und Typ des

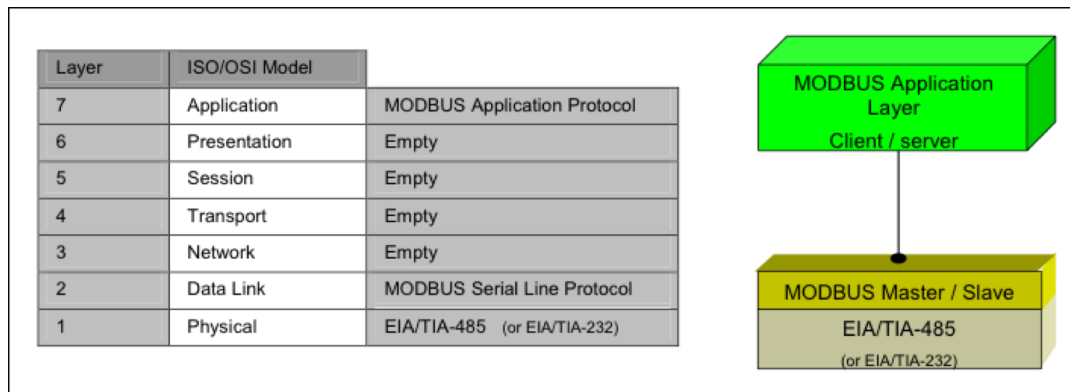


Abb. 2.10: Die Modbus Kommunikation im OSI-Referenzmodell aus [mod, 2006b, S. 5]

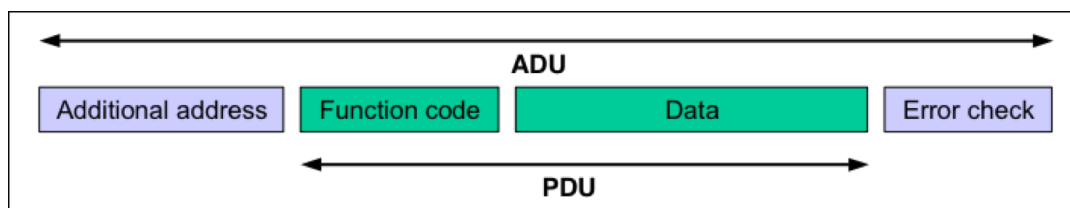


Abb. 2.11: Allgemeiner Rahmen für Telegramme nach dem Modbus Anwendungsprotokoll aus [mod, 2012, S. 3]

darunter liegenden Netzwerks sind. Es beschreibt außerdem wie die Kommunikation abläuft, wie der Client eine Anfrage an ein Server stellt und wie diese auf die Anfragen reagieren und antworten und beschreibt wie Fehler bei der Übertragung entdeckt und darauf hingewiesen werden [MODICON, 96, S. 2f.]. Die Kommunikation kann dabei über eine serielle Leitung nach EIA 485 sowie über Ethernet Netzwerk erfolgen. Über Gateways, wie in Abschnitt 2.2 bereits erläutert, kann die Kommunikation auch über verschiedene Typen von Bussystemen oder Netzwerken geschehen [mod, 2012, S. 3f.].

Das Modbus Protokoll stellt ein allgemeines Gerüst für Telegramme zur Verfügung, welches in Abb. 2.11 dargestellt ist. Die einfache Protocol Data Unit (PDU) enthält die eigentlichen Informationen die ausgetauscht werden sollen und sind daher unabhängig vom Netzwerk oder dem Bussystems. Zu den Informationen gehören die eigentlichen Daten, welche leer sein können oder Daten enthalten die der Slave, in Rahmen von Modbus auch als Server bezeichnet, benötigt, und ein Funktionscode, der ein Byte groß ist und beschreibt welche Art der Aktion Reaktion gefordert ist/wird. Die sogenannte Application Data Unit (ADU) enthält zusätzlich zur PDU weitere Informationen zur Adressierung im Netzwerk und zur Fehlererkennung und ist daher abhängig vom Netzwerk. Der Buszugriff ist damit als Master-Slave geregelt.

Für die Kommunikation wird innerhalb des Masters, der im Rahmen von Modbus auch als Client bezeichnet wird, die Kommunikation durch eine ADU initialisiert. Das genaue Format der ADU wird ist nach Modbus Protokollspezifikation festgelegt. Das Schema einer Transaktion läuft nach dem Prinzip in Abb. 2.12 ab. Wenn eine ADU fehlerfrei empfangen wurde nutzt der Server das Funktionscode Feld um anzuzeigen,

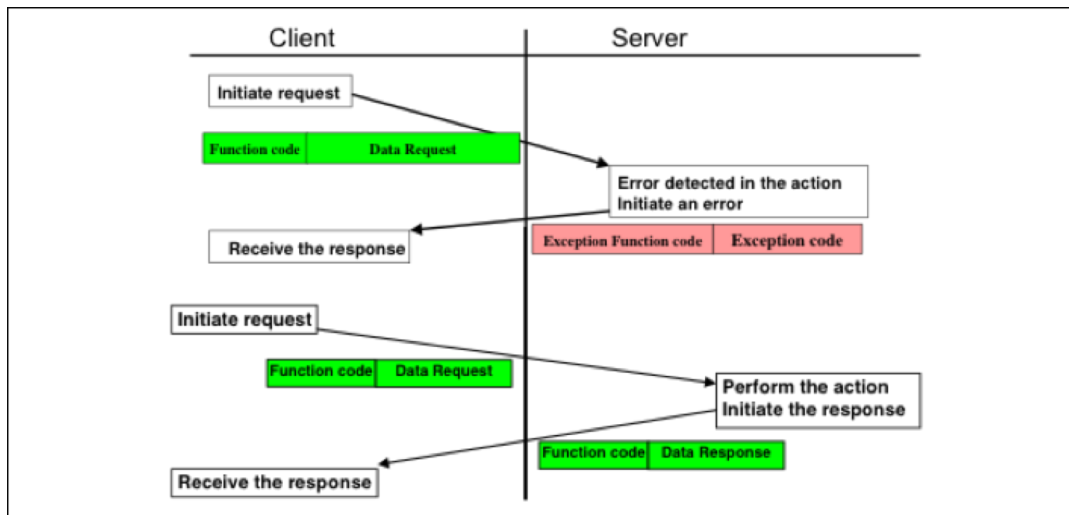


Abb. 2.12: Transaktion mit dem Modbus Protokoll nach [mod, 2012, S. 4]

ob seine Antwort ebenfalls eine normale, fehlerfrei Antwort, durch eine einfaches Echo des empfangenen Funktionscodes, ist oder ob ein irgendein Fehler aufgetreten ist durch einen Exception code. Die Größe der PDU ist grundsätzlich durch die serielle Kommunikation begrenzt auf 256 Bytes, da jedoch noch zwei Bytes für einen Cyclic Redundancy Check und ein Byte für die Server Adresse reserviert werden müssen ist die PDU auf 253 Bytes begrenzt. Ein weiterer, wichtiger Aspekt ist, dass Modbus die „big-Endian“ Codierung/Repräsentation für Daten verwendet verwendet, falls der numerische Wert größer als ein einzelner Byte ist [mod, 2012, S. 3ff.]. Anhand des Beispielder Uhrzeit kann die Bedeutung der Big-Endian Repräsentation einfach erläutert werden: Die Daten werden so aufgeteilt, dass zunächst die Daten mit der höchsten Wertigkeit, also den Stunden zuerst gesendet werden, anschließend den Minuten und zum Schluss die Sekunden, unabhängig von deren numerischem Wert. Also werden nacheinander die Code 03, 50, 12 empfangen bedeutet dues dass die Uhrzeit 03:50:12 ist. Der interessierte Leser wird für eine weitere Ausführungen auch zu little-Endian Darstellung in Bertrand Blanc [2005].

Das Modbus Datenmodell basiert darauf, dass auf die Daten in vier verschiedenen Tabellen mit unterschiedlichen Funktionen und Datenobjekten zugegriffen werden kann:

- In die Discretes Input, welche Single Bit Objekte enthält und lediglich gelesen werden kann,
- die Coils, welche ebenfalls Single Bit Objekte enthält jedoch gelesen und beschrieben werden darf,
- die Input Registers, die Datenobjekte als 16-Bit Wort enthält und wiederum nur gelesen werden kann
- und die Holding Registers, dessen 16-Bit Wort Objekte wiederum gelesen und beschrieben werden dürfen.

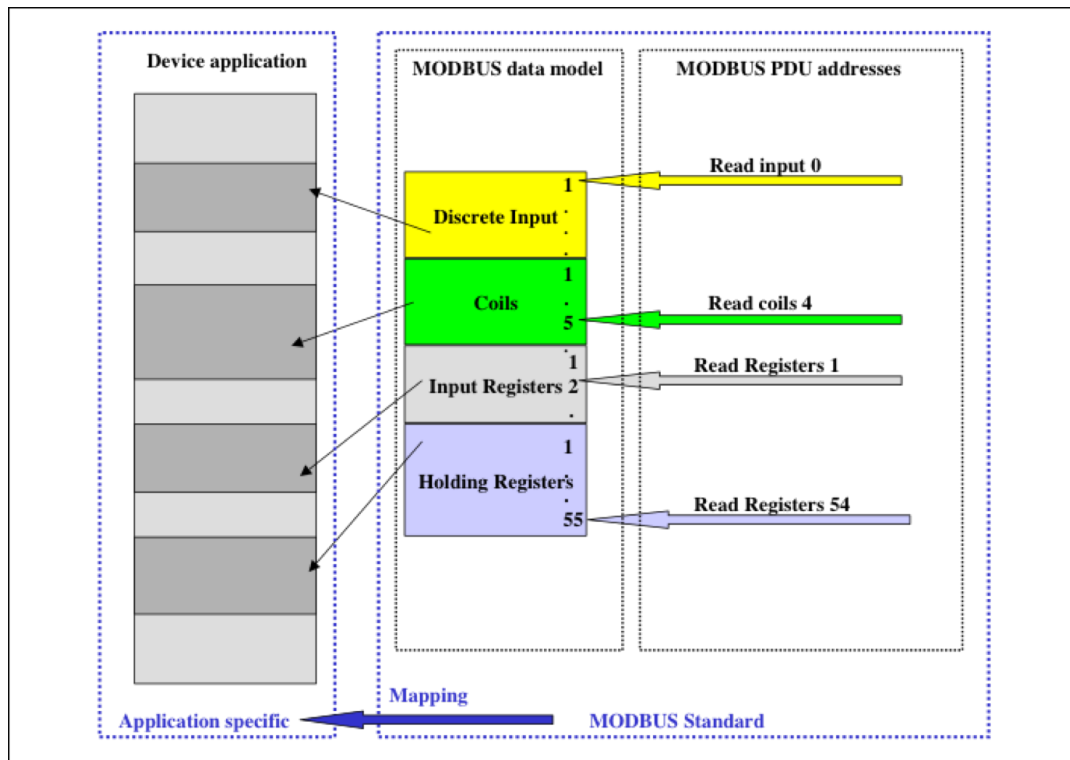


Abb. 2.13: Datenmodell und Adressierung nach dem Modbus Protokoll aus [mod, 2012, S. 8]

Die Daten selbst können auch innerhalb einer Tabelle abgelegt sein, müssen lediglich über die vier angegebenen Tabellen zugreifbar sein, also die Referenz auf die Daten muss gegeben sein, der Rest ist egal. Jede Tabelle besitzt Daten adressiert zwischen 0 und 65535 und ist damit auch nach oben begrenzt was Daten angeht. Welche Daten wo genau stehen, also unter welcher Adresse kann von Gerät zu Gerät unterschiedlich sein und wird vom Gerätehersteller festgelegt. [mod, 2012, S. 6ff.]

Die Funktionscodes starten bei 1 und können bis 255 genutzt werden und sind wie bereits angesprochen in der PDU enthalten und definieren welche Aktion ein Server ausführen soll. In erster Linie dienen sie dem Datenzugriff in den Tabellen, können aber auch für Diagnosen oder nutzerdefinierte Aktionen benutzt werden. Daher lassen sich die Funktionscodes in drei große Gruppen aufteilen, die öffentlichen, nutzerdefinierbaren und die reservierten Funktionscodes. Die öffentlichen sind wohldefiniert, unique, dokumentiert und sind auf Konformität getestet und sind daher einfach, schnell und sicher nutzbar. Die vom Nutzer definierbaren Funktionscodes können genutzt werden um von den öffentlich bereitgestellten nicht bereitgestellte Funktionen eigens zu definieren/nutzbar zu machen. Die reservierten Codes sind von wenigen Unternehmen, welche an der Entwicklung des Modbus Protokolls beteiligt waren, für deren hinterlassene Produkte reserviert und daher nicht öffentlich nutzbar [mod, 2012, S. 10ff.].

Als nächstes folgen die beiden verschiedenen Modbus Over Serial Line Protocol und Modbus Messaging On TCP/IP Protocol

Beim Modbus Over Serial Line Protocol können die Daten über zwei unterschiedliche

Modi übertragen werden, der RTU und ASCII Modus. Näheres Der ASCII Modus ist optional und wird in mod [2006b] detailliert beschrieben. Der RTU Modus wird von allen modbusfähigen Komponenten unterstützt und spezifiziert das folgende Format zur Übertragung der einzelnen Bytes: Jede Byteübertragung beginnt mit einem Startbit, auf das zu übertragende Byte, bestehend aus acht einzelnen Bits, folgt, bevor die Übertragung optional von einem Paritätsbit und einem Stoppbit beziehungsweise lediglich von zwei Stoppbits abgeschlossen wird. Dabei wird jedes zu übertragende Byte als zwei 4-bit hexadezimales Zeichen übertragen[mod, 2006b, S. 12f.]. Die Paritätsprüfung ist optional und dient der Fehlerüberprüfung des Telegramms, wie bereits in Abschnitt 2.2 erläutert. Der Rahmen eines gesamten Modbus RTU Telegramms besteht aus der Slave Adresse, die für jeden Slave eindeutig ist und zwischen 1 und 247 liegt, und dem Function Code, die jeweils aus einem Byte bestehen. Darauf folgen die eigentlichen Informationen für die 0 bis 252 Bytes vorgesehen sind. Abgeschlossen wird der Rahmen durch ein CRC Feld, dass aus einem CRC Low und einem CRC High byte aufgebaut ist und dazu dient das Telegramm auf Fehler zu überprüfen. Der Ablauf und Vorgang des CRC Checks ist detailliert in mod [2006b] beschrieben. Die Übertragung eines Telegramms erfolgt bytewise, wie zuvor beschrieben. Die Datensicherung findet also durch Parität und CRC auf verschiedenen Ebenen statt. Die genaue Übertragungszeit eines Bytes und einer Nachricht hängt von der Baudrate ab. Um den Beginn und den Abschluss eines RTU Rahmens eindeutig zu definieren, geschieht dies in Abhängigkeit von der Übertragungsgeschwindigkeit. Zwischen einzelnen Bytes innerhalb eines Rahmens folgt ein stilles Intervall, dass je nach Länge angibt ob das Telegramm beendet ist. Auf eine Intervall kleiner gleich der anderthalbfachen Übertragung eines Bytes folgt ein weiteres Byte. Ist das stille Intervall länger als die dreieinhalbfache Byteübertragungszeit markiert dies das Ende eines Telegramms und den Beginn eines nächsten Telegramms [mod, 2006b, S. 13]. Diese Zusammenhänge sind zur Veranschaulichung in Abb. 2.14 zusammengefasst.

Der Implementierungsleitfaden legt auch die Spezifikationen der physikalischen Schicht fest, die nun folgen. Er schlägt vor die EIA 438 Schnittstelle als elektrisches/physikalisches Interface zu verwenden, erlaubt aber auch weiterhin die Implementierung durch die EIA 232 Schnittstelle, beides über ein verdrehtes Leiterpaar. Weiterhin werden die Datenraten von 9.600 und 192.000 bps und eine Even Parität bei der Byteübertragung als Standard festgelegt. Die Standardverdrahtung der Komponenten erfolgt bei beiden elektrischen Standards über ein verdrehtes Leiterpaar und einer gemeinsamen Verbindungsleitung common. Die beiden Leitungen des verdrehten Paares werden mit D1, welche auch als D+ oder A Leitung bezeichnet wird, und D0, welche auch als D- oder B Leitung bezeichnet wird, bezeichnet. Ein Standard Netzwerk besteht aus maximal 32 Teilnehmern, dass durch den Einsatz von Repeatern auch vergrößert werden kann. Außerdem wird die Bus-Struktur als Topologie beschrieben, nach der die einzelnen Komponenten im Netzwerk angeordnet werden, unter der Voraussetzung, dass die Busleitung an beiden Enden durch einen Widerstand von 150 Ohm zwischen der D0 und D1 Leitung abgeschlossen werden [mod, 2006b, S. 20ff.]. Die Verbindung

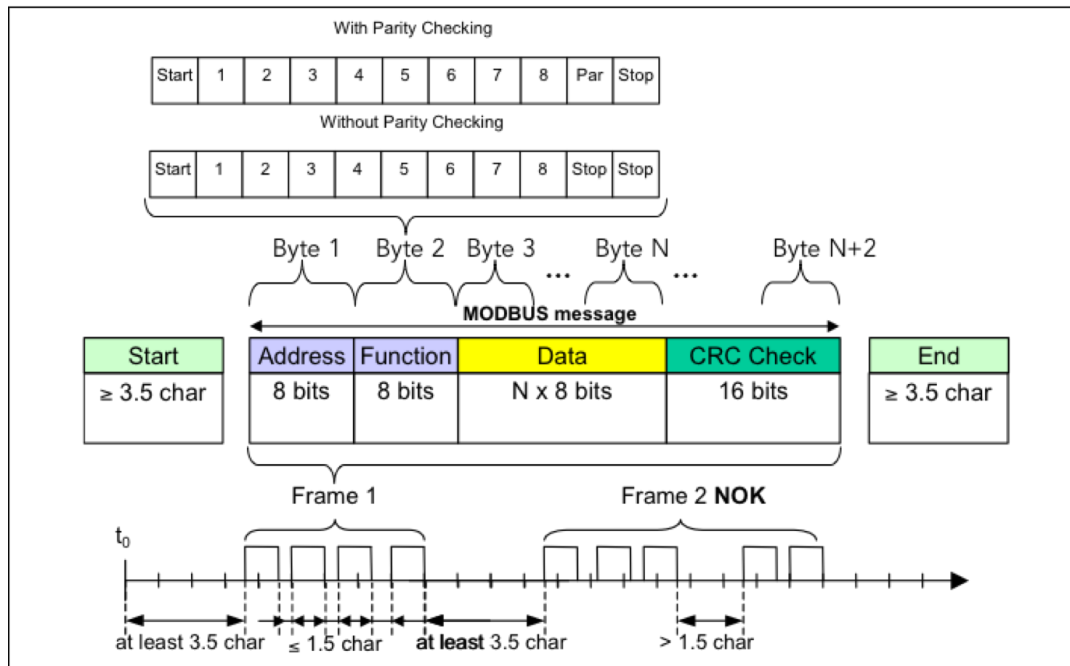


Abb. 2.14: Serielle Kommunikation über Modbus RTU nach [mod, 2006b, S. 12f.]

der Kabel kann im einfachsten Fall durch Schraubklemmen erfolgen, jedoch können auch genormte mechanische Interfaces, also Standard Steckverbindungen genutzt werden, deren Einsatz und Verkabelung/Anschlüsse in [mod, 2006b, S. 29ff.] detailliert beschrieben sind.

XXXXXXXXXX

Beim Modbus Messaging on TCP/IP Protocol stellt die Möglichkeit zur Verfügung, Geräte, die über ein Ethernet (IP Netzwerk) miteinander verbunden sind, über ein Client/Server Modell kommunizieren zu lassen. Des Weiteren erlaubt es dieses Protokoll explizit über Bridges, Gateways oder Router Netzwerke miteinander zu verbinden. Dabei dürfen auch serielle Subnetzwerke zu verbinden und erlaubt auch zwischen diesen Endgeräten die Kommunikation [mod, 2006a, S. 2f.]. Die Kommunikationsarchitektur ist auf in REF ARCHITEKTUR dargestellt. Außerdem ist eine leicht verschiedene ADU vorhanden, wie in Abbildung REF ADU TCP zu sehen. Modbus Application Protocol Header MBAP Header, enthält unit identifier ähnlich slave adress/id, adresse für modbus routing, ein bytecount, der die länge des folgenden Telegramms angibt(inklusive Unit identifier und Daten) auch wenn gesplittet, CRC-32 error check, protocol identifier mit modbus 0, transaction identifier vom client, der nur kopiert wird vom server um transaktionen einander zuzuordnen. Alle Kommunikation erfolgt über TCP Port 502 [mod, 2006a, S. 4f.]

Modbus tcp Komponenten können sowohl client als auch server interface haben. [mod, 2006a, S. 7f.]

2.3 Technische Grundlagen zur Modellbildung

In diesem Kapitel werden die technischen Grundlagen zur Bildung eines mathematischen Modells des Raumes erläutert. Thermodynamische Systeme 1. HS thermo Wärmeübertragung

2.3.1 Thermodynamische Systeme

Im Raummodell müssen Energieströme, genauer betrachtet Wärmeströme, untersucht werden. Um diese thermodynamischen Vorgänge mit Hilfe von Bilanzierungsgleichungen zu beschreiben, folgt zunächst eine kurze Einführung in die Thermodynamische Systembildung nach [Baehr u. Kabelac, 2012, S. 11ff.].

Thermodynamische Systeme werden durch den zu untersuchenden Raum abgegrenzt. Sie dienen dem Zweck der Bilanzierung von Massen- und Energieströmen und alles, was diesen abgegrenzten Raum an den Systemgrenzen umgibt, wird als Umgebung bezeichnet. Die begrenzenden Flächen können gedanklicher, physischer oder beider Natur zugleich sein, wichtig ist jedoch, dass die Systemgrenzen eindeutig festgelegt sind [Baehr u. Kabelac, 2012, S. 11].

Anhand der Eigenschaften von den Systemgrenzen lassen sich die thermodynamischen Systeme weiter differenzieren. Solche Systeme, deren Grenzen undurchlässig für Materie sind, werden als *geschlossene Systeme* bezeichnet und werden durch eine konstante Stoffmenge innerhalb des Systems gekennzeichnet. Die Grenzen eines geschlossenen Systems sind meistens räumlich anhand eines fixen Volumens definiert, können aber auch beweglich sein, wie z.B. das Volumen einer vorgegebenen Stoffmenge unabhängig von dessen räumlicher Ausdehnung [Baehr u. Kabelac, 2012, S. 12].

Sind die Grenzen von thermodynamischen Systemen für Materie durchlässig, werden diese als *offene Systeme* bezeichnet. In der Regel werden diese von Stoffströmen durchflossen und durch räumlich festgelegte Grenzen beschrieben. Diese werden in der Literatur auch als *Kontrollraum* oder *Kontrollvolumen* bezeichnet [Baehr u. Kabelac, 2012, S. 12].

Ein *abgeschlossenes System* umfasst in der Regel mehrere Systeme oder ein einzelnes System und dessen Umgebung, so dass es zwischen den Grenzen des abgeschlossenen Systems und seiner Umgebung keine Wechselwirkungen gibt. Die Systemgrenzen werden also so gelegt, dass über sie hinweg keine beziehungsweise keine relevanten³ Flüsse von Materie und Energie [Baehr u. Kabelac, 2012, S. 13].

Nach der Abgrenzung folgt die *Beschreibung* von thermodynamischen Systemen und dessen *Eigenschaften*. Diese erfolgt durch *Variablen* und *physikalische Größen*, die ein System kennzeichnen. Falls die Variablen feste Werte annehmen, werden diese als *Zustandsgrößen* bezeichnet, da sie den *Zustand* eines Systems bestimmen [Baehr u. Kabelac, 2012, S. 13]. Im Rahmen der Modellbildung in Kapitel 4 ist es ausreichend

³ Relevant im Sinne von kaum messbarer Fluss und nicht messbare Auswirkung auf das System.

die Vorgänge und Effekte auf systemischer Ebene zu betrachten, wodurch sich Modelle mit wenigen Variablen und physikalischen Größen beschreiben lassen.

Die Variablen lassen sich in *äußere Größen*, welche den mechanischen Zustand eines Systems beschreiben⁴, und *innere Größen* gliedern, welche den thermodynamischen Zustand, also die Eigenschaften der Materie innerhalb der Systemgrenzen, beschreiben [Baehr u. Kabelac, 2012, S.13 f.].

Innerhalb der Grenzen eines thermodynamischen Systems, und damit implizit auch für das Raummodell⁵ wird *Homogenität* angenommen. Dies bedeutet, dass die physikalischen Eigenschaften, wie zum Beispiel Temperatur und Druck, sowie die chemische Zusammensetzung an jeder Stelle innerhalb des Systems homogen ist, also die gleiche Ausprägung besitzt [Baehr u. Kabelac, 2012, S.15].

Da wir im Rahmen der Modellbildung Zustände betrachten müssen auch deren Änderungen genauer untersucht werden. Die *Zustandsänderungen* eines Systems werden durch Änderungen von Energie oder Materie über dessen Grenzen hinweg bedingt und finden meist im Austausch der Umgebung statt. Während einer solchen Änderung des Systemzustands wird ein Prozess durchlaufen, der eine zeitliche Abfolge von Ereignissen ist. Eine Änderung des Zustands eines Systems mit der gleichen Wirkung kann also durch verschiedene Prozesse bewirkt werden. Daher beschreibt ein *Prozess* nicht nur die Veränderung des Zustands sondern viel mehr die Beziehungen zwischen einem System und seiner Umgebung [Baehr u. Kabelac, 2012, S.21 f.].

Ein Prozess kann aber auch innerhalb eines Systems stattfinden, dass heißt ohne äußere Einwirkungen. Dies geschieht zum durch das Aufheben innerer Hemmungen oder dem Wegfall Zwängen von Außen. Diese Prozesse laufen in abgeschlossenen Systemen meist von selbst ab und streben als Ziel einen ausgeglichenen, also homogenen, Endzustand an. *Ausgleichsprozesse* dienen somit dazu, einen *Gleichgewichtszustand* zu erreichen und repräsentieren Wechselwirkungen zwischen verschiedenen Teilen eines abgeschlossenen Systems. Dabei gleichen sich die Zustandsgrößen von einzelnen Subsystemen wie zum Beispiel der Druck oder die Temperatur einander an. Der Gleichgewichtszustand wird also durch die Zustände in den einzelnen Subsystemen bestimmt und ist dadurch charakterisiert, dass ein System diesen Zustand nicht von sich aus sondern nur durch äußere Eingriffe verlässt, zum Beispiel durch eine Veränderungen in der Umgebung. Die Erfahrung lehrt, dass ein System einem Gleichgewichtszustand entgegen strebt, wenn es sich selbst überlassen wird [Baehr u. Kabelac, 2012, S.22 f.]. Im Rahmen der Modellbildung in Kapitel 4 nehmen diese *Ausgleichsprozesse* eine zentrale Rolle ein, weil der Großteil an Änderungen von einzelnen Zustandsgrößen innerhalb des Raumes darauf zurückgeführt werden können.

⁴ Zum Beispiel die Koordinaten im Raum oder die relative Geschwindigkeit zum Beobachter)

⁵ Diese Annahme wird im Kapitel 5 noch überprüft und kritisch hinterfragt werden müssen

2.3.2 Erster Hauptsatz der Thermodynamik

Der erste Hauptsatz der Thermodynamik wird im Folgenden als allgemeiner Energieerhaltungssatz formuliert und anschließend angewendet um eine Energiebilanzgleichung für geschlossene thermodynamische Systeme zu erhalten.

Der erste Hauptsatz der Thermodynamik erweitert den mechanischen Energieerhaltungssatz um die Energieformen Wärme und innere Energie. Er handelt ganz allgemein vom Prinzip der Energieerhaltung und dient er der Bilanzierung von Systemen [Baehr u. Kabelac, 2012, S. 43].

Die Gesamtenergie eines Systems E setzt sich zusammen aus der potenziellen E_{pot} und kinetischen Energie E_{kin} wie in der Mechanik und wird durch die innere Energie U ergänzt [Baehr u. Kabelac, 2012, S. 49]:

$$E := E_{pot} + E_{kin} + U \quad (\text{Gl. 1})$$

Im weiteren Verlauf der Arbeit werden nur ortsfeste Systeme betrachtet die sich dadurch auszeichnen, dass deren potenzielle Energie E_{pot} in etwa konstant ist. Weiterhin erfahren sie im betrachteten Inertialsystem Erde auch nur sehr kleine Änderungen in ihrer Geschwindigkeit, weshalb auch die kinetische Energie E_{kin} in etwa konstant ist. Da die Änderungen der mechanischen Energien in Bezug auf die Änderung der inneren Energie sehr klein sind werden im Folgenden nicht weiter betrachtet und die Gesamtenergie eines Systems E vereinfacht und lediglich aus der inneren Energie bestehend angenommen.

Die innere Energie hängt von der spezifischen Wärmekapazität c_p , der Masse eines Systems m_{sys} und der Temperatur t beziehungsweise T ab [Baehr u. Kabelac, 2012, S. 54]:

$$U := m * c_p * T = m * c_p * t + u_0, \text{ mit } t = T - T_0 \quad (\text{Gl. 2})$$

Nach dem Prinzip der Energieerhaltung, kann die Energie eines Systems also weder erzeugt noch vernichtet werden sondern lediglich durch den Energietransport über dessen Grenzen hinweg verändert werden. Daraus ergeben sich folgende qualitative Formen des Energietransports [Baehr u. Kabelac, 2012, S. 48f.]:

- Die Arbeit W , die entweder von oder an einem System verrichtet wird, in differentieller Form die Leistung P .
- Die Wärme Q , die entweder in das System hinein- oder herausfließt, in differentieller Form der Wärmestrom \dot{Q} .
- Der Transport von Materie, also das Einbringen oder Wegnehmen von Masse m eines System, in differentieller Form die Materialflüsse \dot{m} .

Mit der zuvor getroffenen Annahme, dass die innere Energie der des Systems entspricht,

und unter Beachtung der Vorzeichenkonvention, welche besagt dass zugeführte Energie positiv und abgeführte Energie negativ zu bewerten ist, lassen sich die Änderungen der Energie eines Systems mit der folgenden Gleichung quantitativ beschreiben [Baehr u. Kabelac, 2012, S. 54]:

$$\Delta U = Q + W + m_{in} * c_p * T_{in} - m_{out} * c_p * T_{out}$$

beziehungsweise in differentieller Form

$$\frac{dU}{dt} = \dot{U} = \dot{Q} + \dot{P} + \sum \dot{m}_{in} * c_p * T_{in} - \sum \dot{m}_{out} * c_p * T_{out} \quad (\text{Gl. 3})$$

2.3.3 Wärmeübertragung

Wärmeströme spielen bei der Modellbildung in Kapitel 4 eine wichtige Rolle, daher ist eine genauere Betrachtung dieser unumgänglich und im Folgenden werden die Grundlagen dazu erläutert.

Die Definition von Wärmeübertragung ist nach [Böckh u. Wetzel, 2014, S. 1] „[...] der Transfer der Energieform Wärme aufgrund einer Temperaturdifferenz. „Die Definition umfasst also einen zuvor beschriebenen Ausgleichsprozess und eine Änderung der inneren Energie eines thermodynamischen Systems. Die Wärmeübertragung kann nach *Nußelt*⁶ grundsätzlich durch zwei verschiedene Arten stattfinden [Böckh u. Wetzel, 2014, S. 3f.]:

- Durch Strahlung, bei der die Übertragung von Wärme ohne stofflichen Träger durch elektromagnetische Wellen zwischen Oberflächen erfolgt. Weil diese Art der Wärmeübertragung keine Relevanz für die weiteren Betrachtungen hat wird er interessierte Leser an dieser auf Böckh u. Wetzel [2014] verwiesen der diese Thematik detailliert ausführt.
- Durch Wärmeleitung, die sich wiederum in die Wärmeübertragung zwischen ruhenden Stoffen, und die Konvektion, die eine Wärmeübertragung zwischen einem ruhenden und einem strömenden Fluid beschreibt, aufteilen lässt.

Die übertragene Wärmemenge ist bei der reinen Wärmeleitung lediglich von den Stoffeigenschaften und der Temperaturdifferenz abhängig, bei der Konvektion hingegen, unabhängig davon ob erzwungen oder frei, hängt sie von der Strömung der Fluide ab. Die Konvektion ist ein Effekt zusätzlich zur reinen Wärmeleitung auftritt und ist im weiteren Verlauf der Arbeit nicht relevant und wird deshalb nicht detaillierter ausgeführt [Böckh u. Wetzel, 2014, S. 3f.]. Erfolgt der Wärmetransport stationär, dass heißt er ist von äußeren Anregungen bedingt und unabhängig von der Zeit, lässt er sich qualitativ einfach als konstanter Wärmestrom \dot{Q} beschreiben und gibt an wie viel Wärme pro Sekunde übertragen wird [Böckh u. Wetzel, 2014, S. 5ff.]. Der Wärmestrom ist wie zuvor bereits erwähnt von den Stoffeigenschaften abhängig, welche

⁶ Beschrieben in seinem Aufsatz „Das Grundgesetz des Wärmeüberganges“, 1915.

von der Wärmedurchgangszahl U – Wert⁷ und der Austauschoberfläche $A_{exchange}$, an der der Wärmeaustausch stattfindet. Typische U-Werte für verschiedene Materialien und Komponenten finden in der einschlägigen Literatur und beziehen sich bei der Übertragung durch eine Wand im europäischen Raum auf die Außenfläche [Böckh u. Wetzl, 2014, S. 28]. Damit lässt sich der Wärmestrom unter Berücksichtigung der Abhängigkeiten durch die kinetische Kopplungsgleichung quantifizieren [Böckh u. Wetzl, 2014, S. 6f.]:

$$\dot{Q} := u * A * (t_1 - t_2) \quad (\text{Gl. 4})$$

Unterschiedliche geometrische Ausprägungen, wie zum Beispiel ein Wärmeaustausch durch eine Wand oder ein Rohr hindurch, finden damit implizit bei der Austauschoberfläche Berücksichtigung.

2.4 Technische Grundlagen zur Solar- und Gebäudetechnik

2.4.1 Klima

[Peter Häupl, 2013, S. 295ff.]

2.4.2 Sonnenbahn und Strahlung

Nicolai [2013] [Quaschning, 2011, S. 63ff.] [Kaltschmitt, 2013, S. 61ff.] [Peter Häupl, 2013, S. 315ff.] Bild

Algorithmus nach Reda [2008]

Nutzung Berechnung/Implementierung von pys

2.4.3 Gebäudetechnik Glas und Wärmedurchgangskoeffizienten

Auf gehts

Glas nach [Peter Häupl, 2013, S. 61ff.] Durchlassgrad nach [Peter Häupl, 2013, S. 604ff.] Transmissionsgrad

⁷ Der U-Wert wurde bis zu der Umstellung auf die europäischen Prüfnormen 2003 als k-Wert bezeichnet und ist unter dieser Bezeichnung noch häufig in der Literatur zu finden [Sack, 2004, S.1 f.]

„Design is the appropriate combination of materials in order to solve a problem.“

— CHARLES EAMES

3 Anlagendesign

Ziel dieses Kapitel ist es, eine Anlage zur Raumtemperaturregelung für den Betrieb mit Modellprädiktiver Regelung zu konzipieren, zu konkretisieren und im letzten Schritt umzusetzen. Dazu werden zunächst die Anforderungen an die Anlage analysiert und weiterhin die Vorgaben und Rahmenbedingungen zur Anlage von Seiten der Hochschule Karlsruhe spezifiziert und ausgeführt. Daraus wird eine Idee abgeleitet, die anschließend zu einem Konzept weiterentwickelt und in ein konkretes Anlagendesign umgesetzt wird. Dabei werden die einzelnen Anlagenteile und deren Funktionsweisen näher beschrieben und auf die realen Einsatzbedingungen ausgelegt. Abschließend wird die Installation und dabei aufgetretenen Besonderheiten der Anlage beschrieben.

3.1 Idee und Konzept der Anlage

3.1.1 Analyse der Anforderungen und Rahmenbedingungen

Um die Anforderungen an eine MPC-fähige Anlage zu bestimmen, wird zunächst der Zweck und die Einsatzziele der Anlage untersucht. In Kapitel 1.1 wurde bereits darauf hingewiesen, dass es die Vorgabe von Seiten der Hochschule war, die Einsatzziele in Einklang zu der bisherigen Forschung zu bringen und komplementär zu wählen. Daher wurden im Dialog mit den Projektverantwortlichen⁸ für die Forschung im Bereich solarer Anwendungen an der Hochschule Karlsruhe gemeinsam konkrete Einsatzziele der Anlage erarbeitet. Als Ergebnis wurden die folgenden, konkreten Ziele vereinbart:

- Die Einarbeitung in die Thematiken Modellbildung, Kommunikation von technischen Systemen und Modellprädiktive Regelung soll durch eine praktisches Anwendung unterstützt werden.
- Es soll Know-how im Bereich der Kommunikation von technischen Systemen aufgebaut werden, insbesondere im Umgang mit der Software, der Hardware und zahlreichen Schnittstellen.
- Die Anlage soll eine hohe Funktionalität, also möglichst wartungsarm, und eine hohe Robustheit gegenüber Fehlern und Beschädigungen besitzen, da bei der Einarbeitung eine erhöhte Wahrscheinlichkeit der Fehlbedienung besteht und Schäden dadurch vermieden werden sollen.
- Es soll ein Vergleich verschiedener Regelungsmethodiken beim Einsatz von Modellprädiktiver Regelung ermöglicht werden.

⁸ In Person von Herrn ADRIAN BÜRGER und MARKUS BOHLAYER

- Außerdem soll ein Vergleich von Ergebnissen bei der Variation von Steuerungsparametern sowie beim Einsatz verschiedener Steuerungs- und Regelungsalgorithmen ermöglicht werden.
- Des Weiteren soll die Anlage möglichst flexibel ansteuerbar und erweiterbar sein, damit der Grad der Komplexität anpassbar ist und die Anlage um weitere Funktionen oder Features ergänzt werden kann.
- Der temperaturerhöhende Effekt der Sonneneinstrahlung auf die Raumtemperatur soll untersucht werden können.
- Im Rahmen der Anwendungsforschung soll der Raum zur Temperaturregelung möglichst nahe an der Realität sein, also Störgrößen beinhalten und nicht ungenutzt beziehungsweise leerstehend sein.

Zusammenfassend wurde festgehalten, dass die Anlage als Forschungsumgebung für Entwicklungs-, Test- und Anwendungszwecke von verschiedenen Steuerungen und Regelungen dienen soll.

Weiterhin wurden von Seiten der Hochschule Karlsruhe⁹ Rahmenbedingungen definiert, die im Folgenden zusammengefasst sind:

- Der Raum K004a im K Gebäude der Hochschule Karlsruhe wird zur Installation der Anlage und Einrichtung der Forschungsumgebung zur Verfügung gestellt.
- Die Installation der Anlage muss mit minimalem baulichem und finanziellem Aufwand zu realisieren sein.
- Für die Kommunikation innerhalb der Anlage soll die Modbus Kommunikationstechnologie mit mindestens zwei verschiedenen Übertragungsprotokollen genutzt werden.
- Die Modellprädiktive Regelung soll mit Hilfe der Plattform JModelica.org erfolgen.

Diese Einsatzziele und Rahmenbedingungen definieren implizit Anforderungen an eine Anlage, welche im Nachfolgenden explizit ausgeführt werden und aus Gründen der Übersichtlichkeit die wichtigsten in Tabelle Tab. 3.1 zusammengefasst sind.

⁹ In Person von Frau Professor ANGELIKA ALTMANN-DIESES, Herrn Professor MARCO BRAUN und Herrn ADRIAN BÜRGER

Einsatzziele & Rahmenbedingungen	Anforderungen
Raum K004a als Umgebung	– Die Anpassung der Anlage an K004a.
Minimaler baulicher Aufwand	– Die Nutzung bestehender Heizkörper anstatt einer Klimatisierung des Raumes.
Minimaler finanzieller Aufwand	– Die Beschränkung auf eine minimale Funktionalität und Anzahl der einzelnen Komponenten.
Modellprädiktive Regelung mit JModelica.org und CasADi	– Die Modellbildung erfolgt in Modelica.
Einsatz der Modbus Kommunikationstechnologie	– Die Kompatibilität aller Softwareschnittstellen ist in Python durch verschiedene, bereits bestehende Pakete gegeben.
Flexible Ansteuerung der Anlage	– Die Kommunikation der Anlage erfolgt gemäß den Modbus RTU und TCP Protokollspezifikationen.
	– Über Modbus TCP erfolgt die Ansteuerung der Anlage innerhalb eines gesamten lokalen Netzwerks.
Einarbeitung in die Thematiken:	
– Modellbildung,	– Komplexität ist notwendig, darf jedoch nicht zu hoch sein.
– Kommunikation technischer Systeme,	
– und Modellprädiktive Regelung.	– Es sind möglichst wenige thematische Überschneidungen erwünscht, daher wird eine klare Struktur mit möglichst scharfer Trennung benötigt.
Know-how für Kommunikation technischer Systeme	
Vergleich von Ergebnissen durch:	
– die Variation von Steuerungsparametern,	– Die Reaktion des Systems muss schnell messbar sowie günstig und einfach zu erfassen sein.
– den Einsatz verschiedener Regelungsmethodiken,	– Der Einsatz von robusten und einfachen Bauteile.
– und den Einsatz verschiedener Algorithmen.	– Nutzung eines wartungsarmen Systems.
Hohe Funktionalität und Robustheit	– Eine einfache, modulare Erweiterbarkeit des Systems für weitere Schritte muss gegeben sein.
Erweiterbarkeit der Anlage	

Tab. 3.1: Umsetzung der Ziele in Anforderungen der Anlage

Die grundlegendste Anforderung an die Planung ist die Anpassung an die räumliche Lage und den Gegebenheiten des Raumes K004a, welche in Abb. 3.1 skizziert sind. Der Raum befindet auf dem Campus der Hochschule Karlsruhe an der südwestlichen Ecke des K Gebäudes im Erdgeschoss. Die südliche und westliche Außenwand teilt sich der Raum mit der Außenumgebung. Die östliche und nördliche Innenwand sowie die Decke und der Boden des Raumes teilt der Raum hingegen mit angrenzenden Räumen im K Gebäude. Des Weiteren ist in der südlichen Außenwand eine Fensterfront mit Jalousien

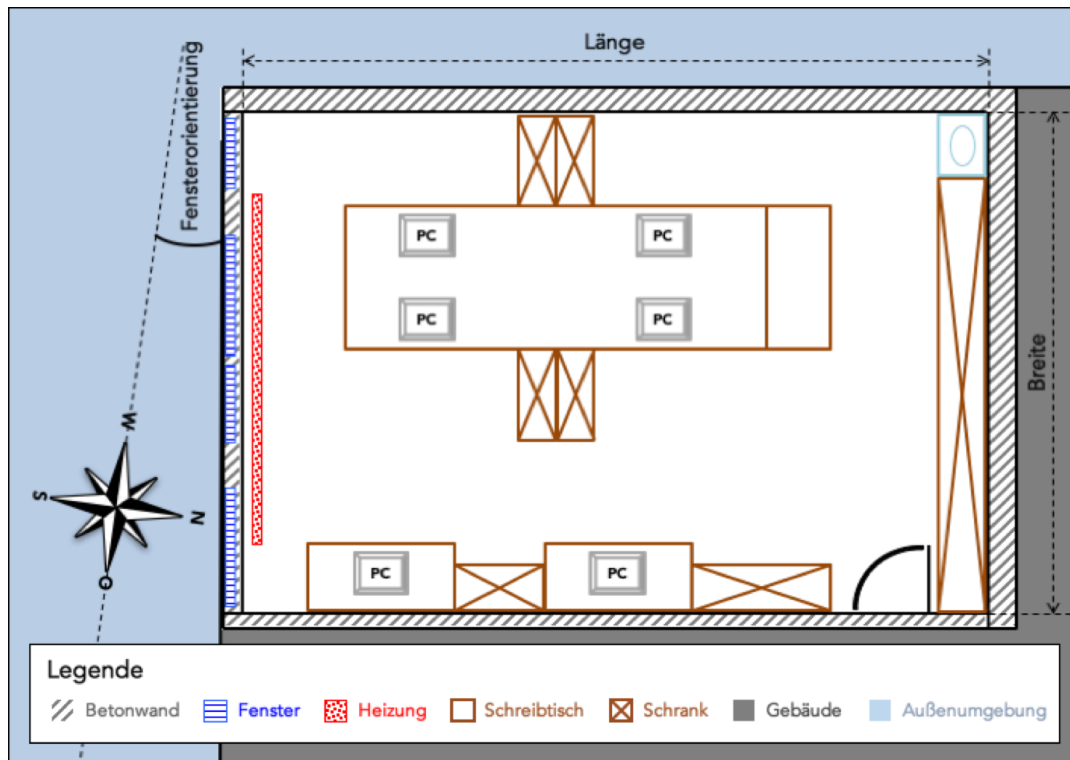


Abb. 3.1: Raumskizze K004A vom K Gebäude der Hochschule Karlsruhe – Technik und Wirtschaft

zur Verschattung eingebaut und direkt darunter ist ein Heizkörper installiert.

Dadurch werden bereits zwei wichtige Anforderungen erfüllt, denn durch die Fensterfront an der Südseite des Raumes kann der Einfluss der Sonneneinstrahlung auf die Raumtemperatur untersucht werden und der vorhandene Heizkörper kann in die Anlage integriert werden, um einen minimalen baulichen und finanziellen Aufwand sicherzustellen. Damit wird auch eine Klimatisierung des Raumes zur Temperaturregelung zunächst ausgeschlossen, weil diese mit einem erheblichen finanziellen und baulichen Aufwand verbunden ist. Da es jedoch eine weitere Anforderung ist, Erweiterungen der Anlage explizit zu berücksichtigen und ermöglichen, wird bei der Planung darauf geachtet, dass eine Klimatisierung und Ansteuerung der Jalousie für weitere Schritte möglich sein soll

Weiterhin dient der Raum als Büro für wissenschaftliche Mitarbeiter, weshalb sich neben sechs Computerarbeitsplätzen auch Büromobiliar, bestehend aus Schränken und Schreibtischen, darin befindet.

Damit wird auch der Forderung nach einer anwendungsnahen Umgebung Sorge getragen, denn durch die Rechner und Mitarbeiter sind nicht vorhersehbar/ bedingt steuerbare parameter größen innerhalb der Anlage.

Eine weitere restriktive Vorgabe ist, dass die Modellprädiktive Regelung mit Hilfe der Plattform JModelica.org und erfolgen soll. JModelica.org ist eine Umgebung zur Simulation und Optimierung von dynamischen Systemen [AB, 2015, S. 1f.] und besitzt eine eigene Klasse für Modellprädiktive Regelung. Der Compiler übersetzt

Modelica Modelle direkt in OptimizationProblem Objekte, welche eine symbolische Repräsentation des Optimierungsproblems dass mit CasADi Andersson [2013] zur Optimierung benutzt werden kann [AB, 2015, S. 12]. Besitzt eine Python Schnittstelle Daher die Anforderung an die Modellbildung, da das Modell ebenfalls ein Teil der Anlage für MPC ist, in Kapitel 4, dass die Modellbildung in Modelica stattfindet. CasADi wiederum ist eine Open-Source Software für numerische Optimierung allgemein und für die Optimalsteuerung von nichtlinearen Gleichungssystemen. CasADi eignet sich aufgrund seiner effizienten Ableitungserzeugung durch Algorithmische Differentiation besonders für Optimierungszwecke [Joel Andersson, 2015, S. 5f.] und besitzt eine Python Schnittstelle

Die Optimalsteuerung stellt in diesem Fall den begrenzenden Faktor dar, da die Optimierungsumgebung CasADi für dynamische Systeme nur unter JModelica.org läuft. Daher wird darauf aufbauend das benötigte Modell für die MPC in Modelica gebildet unter Berücksichtigung der Restriktionen bezüglich JModelica. Die gemeinsame Schnittstelle beider ist Python, über die damit auch die Kommunikation mit den Hardwarekomponenten der Heizungsteuerung erfolgen muss/soll.

Des Weiteren soll die Kommunikation mit Hilfe der Modbus Kommunikationstechnologie stattfinden, da die große Anlage der HsKa einfacher in Betrieb genommen werden kann. Um weitere Erfahrungen und das Know-how ein wenig breiter werden die beiden Modbus Protokolle RTU und TCP Anwendung finden und über verschiedene elektrische Schnittstellen genutzt werden. Das Modbus Protokoll wurde in verschiedenen Python Paketen bereits umgesetzt.

Damit die Einarbeitung in die Themengebiete Modellbildung, Modellprädiktive Regelung und Kommunikation technischer Systeme vereinfacht wird, sollte die Anlage möglichst wenig Komplexität aufweisen. Damit die Zusammenhänge in den einzelnen und Wechselwirkungen zwischen den Fachgebieten und Komponenten möglichst einfach begreifbar sind, wird außerdem eine klare Abgrenzung und Strukturierung der Anlage gefordert. Im Gegensatz dazu steht die Forderung nach dem Aufbau von Know-how für die Kommunikation technischer Systeme durch den Einsatz verschiedener Hard- und Software sowie Schnittstellen, denn mit einer wachsenden Anzahl von Schnittstellen, verschiedener Soft- und Hardware geht auch eine Erhöhung der Komplexität einher. Deshalb muss nach einem Kompromiss zwischen Verständlichkeit und Komplexität gesucht werden, weshalb ein bestimmtes Maß an Komplexität erwünscht ist.

Um das Vergleichen von Ergebnissen zu ermöglichen und weitestgehend zu vereinfachen, sollen die Reaktionen auf Veränderungen schnell stattfinden¹⁰ und einfach zu messen sein. Das bedeutet konkret, dass die Anlage und Raumtemperatur zum einen „schnell“ eine Reaktion auf Steuerungsimpulse zeigen soll, zum anderen soll die Reaktion einfach, das heißt ohne großen technischen und monetären Aufwand und möglichst direkt, messbar sein. Außerdem sollen eine hohe Funktionalität gegeben sein, dass heißt eine möglichst wartungsarm und Fehlerquellen auszuschließen und damit

¹⁰ Im Kontext von thermischen Systemen heißt schnell im Minutenbereich

die wissenschaftliche Arbeit zu erleichtern. Entsprechend wird auch eine Robustheit gegenüber Fehlern gefordert, da bei Testeinsätzen von Steuerungs- und Regelungsalgorithmen sowie bei der Einarbeitung in die Anlage und Themengebete eine erhöhte Gefahr/Wahrscheinlichkeit in Bezug auf das Fehler passieren besteht und diese keinen Schaden an der Anlage verursachen sollen. Deshalb sollen die einzelnen Komponenten der Anlage möglichst einfach aufgebaut sein und ohne technischen Schnickschnack sein, was auch wiederum zur Forderung der minimalen finanziellen Belastung passt.

Da alle bereits genannten Hard- und Softwarekomponenten haben eine Python Schnittstelle besitzen beziehungsweise frei nutzbare Pakete in Python als Schnittstelle zur Verfügung stehen, soll auch die Ansteuerung der gesamten Anlage in Python stattfinden. Damit kann Python als gemeinsamer Nenner für Ansteuerung Anlage und Optimalsteuerung genutzt werden und es kann ein minimaler Rechner dazu genutzt werden, wie z.B. ein Raspberry Pie. Daher wird die gesamte folgende Planung darauf aufbauen, dass sie sich in/aus Python steuern lässt.

3.1.2 Idee

Eine erste, grobe Idee, welche in den folgenden Abschnitten sukzessive an die an die obigen Anforderungen angepasst wird, ist also eine Anlage zur Regelung der Temperatur im Raum, die so minimal wie möglich aufgebaut ist. Konkret gilt es zunächst die Raumtemperatur über Sensoren zu erfassen und diese weiterhin mit Hilfe des Heizkörpers zu steuern. Ein minimaler Aufbau lässt sich in drei Gruppen/Klassen von Anlagenteilen gliedern: Die Sensorik zur Ermittlung des Zustandes im Raum und der Steuergrößen, der Aktorik zur Beeinflussung des Raumzustandes und einen zentralen logischen Controller, der das Zusammenspiel der einzelnen Anlagenteile und Komponenten koordiniert und MPC macht.

3.1.3 Idee weiter konkretisiert

Nachdem die Anforderungen konkretisiert und erläutert wurden und die Idee der Anlage klar ist, wird diese nun weiter konkretisiert, entsprechend der drei zuvor gebildeten Gruppen.

Der logische Controller ist die zentrale Komponente der Anlage, da dieser die gesamten Steuerungs- und Kommunikationsaufgaben übernimmt. Dazu nutzt er die Sensorik und Aktorik um sein Ziel der Temperaturregelung zu erreichen und da dazu Optimalsteuerungspläne berechnet werden müssen wird dazu eine ausreichende Rechenkapazität benötigt. Um diesen Aufgabenumfang abarbeiten zu können und um dem minimalen finanziellen Aufwand genüge zu tragen, wird hierzu zunächst ein freier Rechner der Hochschule Karlsruhe genutzt.

Bei der Temperaturregelung interessiert lediglich die Raumtemperatur, welche als Zustand aufgefasst wird, und es wird zunächst ein Sensor zur Bestimmung der Raumtemperatur benötigt. Da jedoch die Temperatur innerhalb des Raumes mit hoher

Wahrscheinlichkeit nicht homogen ist, werden mehrere Sensoren zur Temperaturmessung benötigt, weshalb mindestens zwei Sensoren benötigt werden.

Für die Erwärmung des Raumes muss gemäß den Anforderungen der Heizkörper innerhalb des Raumes genutzt werden. Um den Heizkörper steuern zu können, wird ein Aktor benötigt, der das Ventil am Heizkörper öffnen und schließen kann. Jedoch ist die eingebrachte Wärmemenge des Heizkörpers in den Raum abhängig von verschiedenen Faktoren, welche der Controller zur Modellprädiktiven Regelung benötigt. Dazu gehören zum einen der Massendurchfluss sowie die Temperatur des Heizwassers am Ein- und Ausgang des Heizkörpers. Um diese zu erfassen werden also zwei weitere Temperatursensoren und ein Durchflusssensor benötigt.

Dieses Set-Up/Einrichtung ist in ?? graphisch dargestellt.

Dieses Konzept sieht auch eine Erweiterungsmöglichkeit vor, so dass das Öffnen und Schließen der Jalousien ebenfalls von der Anlage aus gesteuert werden kann. Außerdem erlaubt das Konzept auch eine Erweiterung zum Beispiel durch eine Klimatisierung, um die Raumtemperatur ganzjährig beziehungsweise in beide Richtungen zu regeln.

Damit steht das Konzept der Anlage und im nächsten Abschnitt kann eine detaillierte Planung der Bauteile, Verdrahtung und Komponenten und Protokolle erfolgen.

3.1.4 Konkretisierte Planung der Anlage/Konkretisierung des Plans

Wie bereits erwähnt stellt der Rechner das Zentrum der Anlage dar. Deshalb wird darauf aufbauend zunächst die Kommunikationsleitungen und das Netzwerk geplant. Da diese über die Modbus Protokolle RTU und TCP geschehen soll und Modbus TCP über ein normales Ethernet auf OSI Ebene 0/1 kommuniziert und der Rechner einen Netzwerkanschluss/Ethernetanschluss besitzt wird dieser an ein solches Netzwerk angebunden, damit keine weitere Karte bzw. kein Umsetzer für den Anschluss eines RS 232-Netzwerkes bzw. RS 485 nicht möglich benötigt wird an dessen COM port. Da ebenfalls Schnittstellen gefordert sind, wird über ein Gateway das Modbus TCP Ethernet Netzwerk mit einem RS 485 verbunden.

Die Temperatursensoren sollen als Raumtemperaturfühler mit Modbusfähigkeit ausgeführt werden und an zwei Stellen des Raumes die Temperatur messen. Jeweils auf einer mittig gedachten Achse in nord-südlicher Richtung, einer in der am Ende der Schreibtische in Richtung der Fenster und der andere gegenüber am anderen Ende der Schreibtische. Um die Heizung mit den benötigten Sensoren auszustatten wird ein Wärmemengenzähler vorgesehen, der zusätzlich zu den Sensoren noch ein Rechenwerk mit Modbusanschluss zum Auslesen aller drei Werte enthält. Um den Massendurchfluss im Heizkörper steuern zu können wird ein Stellantrieb benötigt. Dieser kann durch einen Elektromotor oder durch ein thermisches Element, welches wiederum durch elektrische Energie erhitzt wird und durch die Ausdehnung einstellt, angetrieben werden. Um der Forderung nach weiteren Schnittstellen genüge zu tragen und da es üblich ist solche Aktoren mit einem digitalen oder analogen Signal, also

durch eine Spannung oder einen Strom zu steuern, soll Signalwandler von Modbus zu analog/digital eingesetzt werden.

Damit sind die Komponenten spezifiziert und es werden in den folgenden Abschnitten die einzelnen zum Einsatz kommenden Bauteile erläutert bevor die gesamte Anlage und deren Funktion als Ganzes beschrieben werden.

3.2 Bauteile

3.2.1 Gateway

3.2.2 Temperatursensoren

3.2.3 Wärmemengenzähler

3.2.4 Signalwandler und Stellantrieb

3.2.5 Die Gesamtanlage mit Schaltplan und allem

„Das beste Modell für eine Katze ist eine Katze; möglichst dieselbe Katze.“

— NORBERT WIENER

4 Modellbildung des Raumes

Ziel dieses Kapitels ist es, ein hinreichend exaktes Modell zur Berechnung der Raumtemperatur, basierend auf den thermodynamischen Prozessen mit dessen Umgebung und der Anlage aus Kapitel 3, zu bilden, um damit und unter Zuhilfenahme der Anlage Modellprädiktive Regelung zu ermöglichen. Dazu wird zunächst ein einfaches Grundmodell für einen hypothetischen Raum gebildet, dass anschließend schrittweise an den bestehenden Raum erweitert angepasst wird, bis eine die Qualität des Modells ausreichend ist.

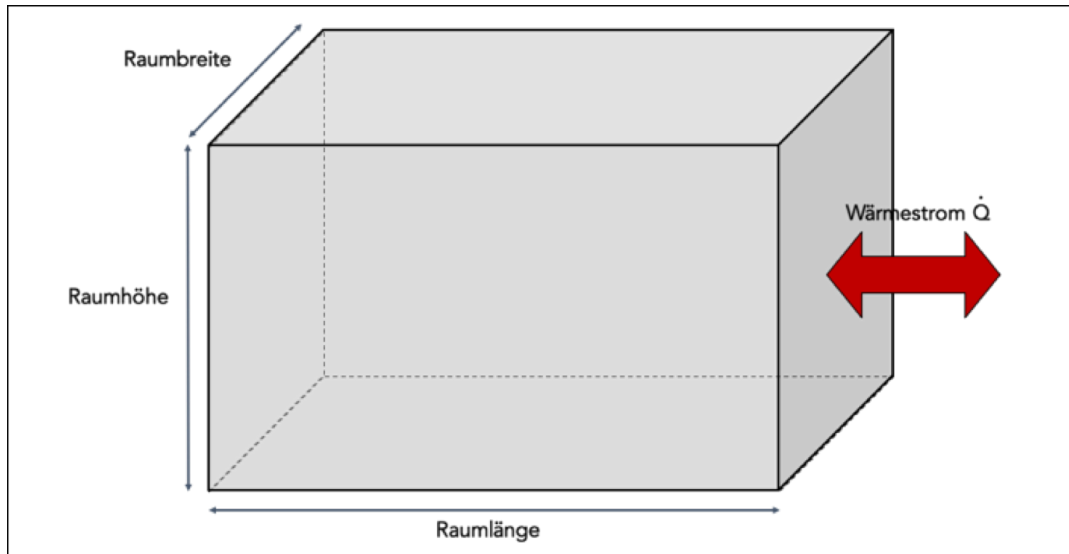
Physikalisch motiviert. Anhand von Zustand, Steuergröße und Parameter/Eigenschaft

4.1 Anforderungen an das Raummodell

Da MPC die Lösung von Nichtlinearen Gleichungssystemen erfordert wird ein erhöhter Rechenbedarf benötigt. Daher sollte das Modell so einfach wie möglich gehalten werden. Die Krux liegt also darin, einen geeigneten Kompromiss zwischen Komplexität und Genauigkeit des Modells zu finden, der eine sinnvolle MPC Regelung ermöglicht. Des Weiteren werden gradientenbasierte Ableitungen bei der Optimierung/Lösung des LGS generiert weshalb auch keine Unstetigkeiten im Modell vorkommen dürfen. Die triviale Aufgabe ist die hinreichend genaue Beschreibung der Realität bzw realen Vorgänge. hinreichend definieren. Damit eine Steuerung mit Hilfe von Modellprädiktiver Regelung möglich ist, darf das Modell keine hohe Kompliziertheit aufweisen und sollte durch möglichst wenig Gleichunge trotzdem ein möglichst genaues Abbild der Realität abbilden. Das Modell soll zunächst so simpel wie möglich gestaltet werden um eine Optimierung mit Hilfe von MPC zu ermöglichen. Dessen Verfahren zur Optimierung sind gradientenbasiert und erfordern damit die Erzeugung von stetigen Ableitungen bis zum zweiten Grad. Daher soll die Komplexität des Modells zunächst sehr gering gehalten werden und dann Stück für Stück erhöht werden und die damit die Genauigkeit des modells erhöht werden

4.2 Das Grundmodell des Raumes

Um ein möglichst einfaches Grundmodell zu erhalten, wird zunächst ein hypothetischer Raum betrachtet. Dieser Raum bildet zusammen mit der ihn umgebenden Luft ein abgeschlossenes thermodynamisches System, wie in Kapitel 2.3 beschrieben. Der Raum

**Abb. 4.1:** Grundmodell eines Raumes

ist selbst mit Luft gefüllt und wird zu allen sechs Seiten hin durch Wände begrenzt. Damit bildet der Raum ein geschlossenes System, da keine Massenströme über die Grenzen hinweg fließen können. An den Grenzflächen kann also lediglich Wärme zwischen der Umgebung und dem Raum ausgetauscht werden. Des Weiteren wird eine homogene Temperatur innerhalb des Raumes und der Umgebung angenommen, welche in der Realität eingeschwungenen Gleichgewichtszuständen innerhalb der beiden Teilsysteme entspricht. Um die Annahme für den Raum zu überprüfen, muss noch festgestellt werden auf welcher zeitlichen Skala der Einschwingvorgang für eine homogene Temperatur innerhalb des Raumes stattfindet und ob dieser damit eine Relevanz für die Modellbildung besitzt.

Zur Bestimmung der Temperatur innerhalb des Raumes, ausgehend von einer initialen Raumtemperatur und dem externen Steuerungsparameter der Umgebungstemperatur, muss der Ausgleichsprozess zwischen Raum und Umgebung untersucht werden, konkret der ausgetauschte Wärmestrom. Um diesen nach Gl. 4 zu berechnen, müssen zunächst die verschiedenen modellrelevanten Eigenschaften des Raumes durch physikalische Größen und Variablen beschrieben werden. Zur Berechnung der Austauschoberfläche wird die Raumbreite, -länge und -höhe benötigt und weiterhin sind der U-Wert einer Betonwand, die spezifische Wärmekapazität und Dichte von Luft für die Bestimmung des Wärmestroms relevant. Diese modellrelevanten Eigenschaften sind allesamt mit ihren Zahlenwerten in Tabelle Tab. 4.1 zusammengefasst.

Modellrelevante Eigenschaften	Wert	Einheit
Raumbreite	7,81 ¹⁾	[m]
Raumlänge	5,78 ¹⁾	[m]
Raumhöhe	2,99 ¹⁾	[m]
Wärmedurchgangskoeffizient Betonwand	2,0 ²⁾	$[\frac{W}{m^2 \cdot K}]$
Spezifische Wärmekapazität von Luft	1.000,0 ³⁾	$[\frac{J}{kg \cdot K}]$
Dichte von Luft	1,25 ³⁾	$[\frac{kg}{m^3}]$

¹⁾Werte durch eigene Vermessung des Raumes K004b vom 07.12.2015.

²⁾Schätzwert, geschätzt nach [Recknagel, 2013, S. 409] mit Richtwerten aus [Recknagel, 2013, S. 194ff.].

³⁾Tabellenwert aus [Peter Häupl, 2013, S. 139].

Tab. 4.1: Eigenschaften des Raummodells

Erfolgt nun die Bilanzierung des Raumes mit Hilfe des ersten Hauptsatzes der Thermodynamik nach Gl. 3 und die Berechnung der inneren Energie des Raumes nach Gl. 2 ergibt sich folgendes, einfaches Gleichungssystem zur Bestimmung der Raumtemperatur in Abhängigkeit vom Steuergrößen Außentemperatur im Grundmodell in Modelica:

```

1 equation
  /* calculate room volume */
3  room_volume = room_length * room_height * room_breadth;
  /* calculate room mass */
5  room_mass = room_volume * rho_air;
  /* calculate surface of heat exchange */
7  exchange_surface = 2 * (room_length * room_breadth) + 2 * (room_length * room_height) + 2
    * (room_breadth * room_height);
  /* calculate inner energy*/
9  room_u = room_mass * cp_air * room_temperature;
  /* calculate derivative of the inner energy */
11 der(room_u) = environment_qdot;
  /* calculate heatflow between room and environment */
13 environment_qdot = u_wall * exchange_surface * (environment_temperature - room_temperature
    );

```

Listing 4.1: Einfaches Gleichungssystem für das Grundmodell des Raumes in Modelica

Damit ist ein Grundmodell für einen Raum gebildet, wie in Abb. 4.1 graphisch dargestellt, um die Temperatur innerhalb eines Raumes zu berechnen. Dieses wird im Folgenden nun schrittweise erweitert und zum Abschluss überprüft, ob es der Realität genüge zu trägt.

4.3 Modellerweiterung durch Berücksichtigung der realen Umgebung

Im nächsten Schritt wird das einfache Raummodell zunächst an die reale Umgebung des Raums K004b angepasst. Die Lage von K00b ist in Abb. 3.1 ersichtlich und es ist zu

erkennen, dass der Raum lediglich zwei Außenwände besitzt, die an die Umgebungsluft grenzen: Die Wände in Richtung Süden und Westen. Die anderen beiden Wände, sowie die Decke und der Boden, grenzen an weitere Gebäudeteile des K-Gebäudes. Somit entspricht der Raum im Modell nach wie vor einem geschlossenen System und bildet weiterhin, zusammen mit dem umgebenden K-Gebäude und der Umgebungsluft, ein abgeschlossenes System. Jedoch müssen nun potenziell verschiedene Wärmeströme zwischen dem Raum und der Außenumgebung sowie dem Raum und dem K-Gebäude betrachtet werden. Da die fließenden Wärmeströme im Vergleich zur sehr großen Energie innerhalb des gesamten K-Gebäudes und der Umgebung nur verschwindend gering sind, wird der erwärmende beziehungsweise kühlende Effekt der Wärmeströme auf die beiden Teilsysteme vernachlässigt und es wird von konstanten, homogenen Temperaturen beider ausgegangen.

Durch diese Erweiterung des Modells hängt die Raumtemperatur nun von zwei Wärmeströmen und damit indirekt von zwei externen Steuergrößen, den Temperaturen in der Umgebung und im K-Gebäude, ab. Um die Wärmeströme separat berechnen zu können, wird die gesamte Oberfläche zum Wärmeaustausch aufgeteilt in die Austauschoberfläche mit der Umgebung und die Austauschoberfläche mit dem K-Gebäude. Des Weiteren werden im Modell die Temperatur der Außenumgebung und die Temperatur innerhalb des K-Gebäudes als externe Steuergrößen berücksichtigt. Das Gleichungssystem des Grundmodells in 4.1 erweitert sich also um folgende Änderungen:

```

1 equation
  [...]
3  /* calculate surface of heat exchange with the environment */
   environment_surface = room_length * room_height + room_breadth * room_height;
5  /* calculate surface of heat exchange with the remaining building */
   building_surface = 2 * (room_length * room_breadth) + room_length * room_height +
     room_breadth * room_height;
7  /* calculate derivative of the inner energy */
   der(room_u) = environment_qdot + building_qdot;
9  /* calculate heatflow between room and environment */
   environment_qdot = u_wall * environment_surface * (environment_temperature -
     room_temperature);
11 /* calculate heatflow between room and building */
   building_qdot = u_wall * building_surface * (building_temperature - room_temperature);

```

Listing 4.2: Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der realen Umgebung in Modelica

Damit wurde das Raummodell an die reale Umgebung angepasst und um die Temperatur innerhalb des Raumes zu bestimmen, wird nun neben der Ausgangstemperatur im Raum und der Umgebungstemperatur noch die Temperatur innerhalb des restlichen K-Gebäudes berücksichtigt. Im nächsten Schritt werden die realen, räumlichen Gegebenheiten im Modell abgebildet.

4.4 Modellerweiterung durch Berücksichtigung der räumlichen Gegebenheiten

Um das Modell an die realen Gegebenheiten des Raumes K004b anzupassen, müssen zwei bauliche Gegebenheiten beachtet werden. Wie in Abb. 3.1 dargestellt ist, ist in der südlichen Außenwand eine Fensterfront vorhanden. Da der U-Wert eines Fensters erheblich von dem U-Wert einer Wand abweicht, entsteht ein zusätzlicher Wärmestrom zwischen dem Raum und der Umgebung durch das Fenster hindurch. Das Öffnen und Schließen der Fenster mit daraus resultierenden Massenströmen wird jedoch zunächst nicht explizit berücksichtigt, weshalb das Raummodell weiterhin als geschlossenes System betrachtet wird. Des Weiteren ist es möglich, den Raum über einen Heizkörper zu beheizen. Mit dem Heizkörper, der zunächst als einfache Wärmequelle im Modell ergänzt wird, erhöht sich die Anzahl der externen Steuergrößen erneut, da die Temperatur innerhalb des Raumes auch von dieser abhängig ist.

Durch diese Erweiterungen werden auch weitere physikalische Größen zur Beschreibung der Eigenschaften des Raummodells benötigt. Wie bereits erwähnt werden die Eigenschaften um den U-Wert eines Fensters, sowie die Breite und Höhe der Fensterfront ergänzt wie in Tabelle Tab. 4.2 zusammengefasst.

Modellrelevante Eigenschaften	Wert	Einheit
Fensterbreite	7,0 ¹⁾	[m]
Fensterhöhe	2,08 ¹⁾	[m]
Wärmedurchgangskoeffizient Glas	4,0 ²⁾	$[\frac{W}{m^2 \cdot K}]$

¹⁾Werte durch eigene Vermessung des Raumes K004b vom 07.12.2015.

²⁾Tabellenwert, geschätzt nach [Hauser, 2000, S. 270ff.].

Tab. 4.2: Weitere Eigenschaften des Raummodells

Durch diese Anpassung verändert sich die Austauschoberfläche mit der Umgebung, die sich nun auf zwei Flächen mit verschiedenen Wärmedurchgangskoeffizienten verteilt. Des Weiteren wird eine Wärmequelle für die Heizung ergänzt, so dass sich folgende Änderungen des Gleichungssystems im Vergleich zum bisherigen Modell in 4.1 ergeben:

```

equation
2  [...]
   /* calculate surface of heat exchange with the environment */
4  environment_surface = room_length * room_height + room_breadth * room_height -
    window_surface;
   /* calculate surface of heat exchange with the remaining building */
6  building_surface = 2 * (room_length * room_breadth) + room_length * room_height +
    room_breadth * room_height;
   /* calculate surface of window with the environment */
8  window_surface=(window_length*window_height);
   /* calculate derivative of the inner energy */
10 der(room_u) = environment_qdot + building_qdot + window_qdot + radiator_qdot;

```

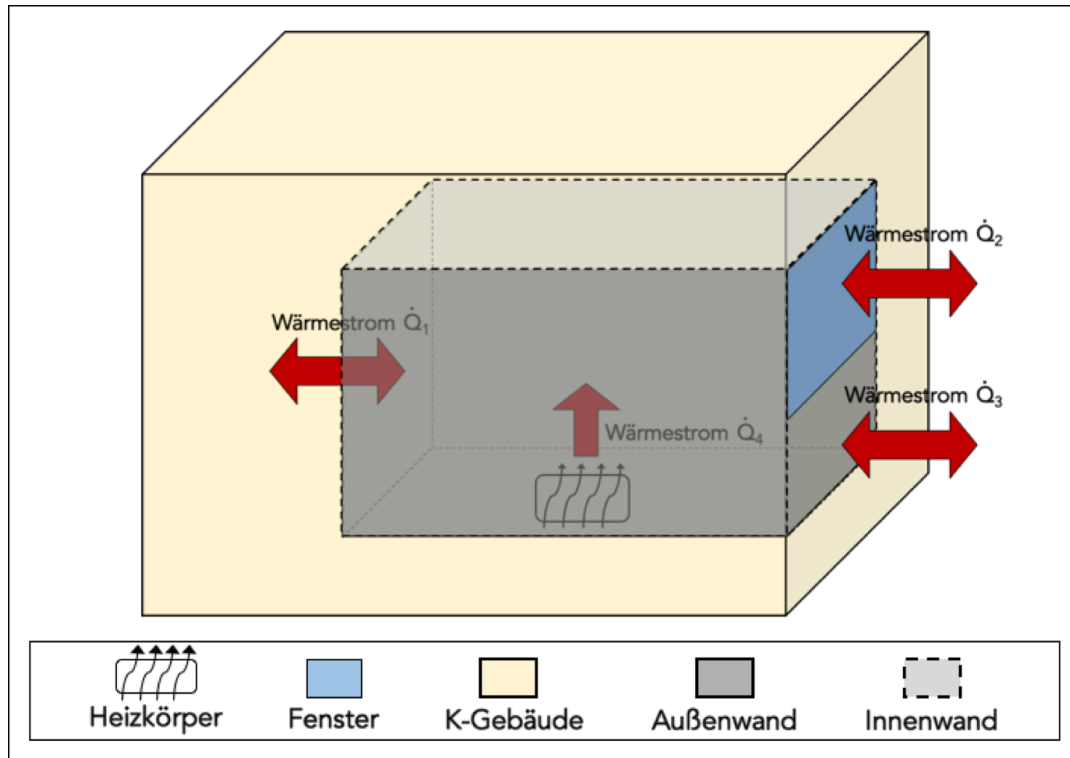


Abb. 4.2: Erweitertes Raummodell

```

/* calculate heatflow between room and environment through the walls */
12 environment_qdot = u_wall * environment_surface * (environment_temperature -
    room_temperature);
/* calculate heatflow between room and environment through the window */
14 building_qdot = u_glass * window_surface * (environment_temperature - room_temperature);
/* calculate heatflow between room and building */
16 building_qdot = u_wall * building_surface * (building_temperature - room_temperature);

```

Listing 4.3: Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der räumlichen Gegebenheiten in Modelica

Damit ist das Modell auch an die räumlichen Gegebenheiten angepasst und beschreibt dadurch in groben Zügen die realen Zusammenhänge. Allerdings kann eine Steuerung/Regelung eine Heizung nicht einfach beliebig als einfache Wärmequelle einsetzen, weshalb eine detailliertere, physikalisch motivierte Modellierung folgen wird. Die bisherigen Zusammenhänge des Modells sind in Abb. 4.2 graphisch dargestellt.

Bereits bei den Einsatzzielen der Anlage in 3.1.1 war gefordert, den Zusammenhang zwischen der Sonneneinstrahlung und der Raumtemperatur zu untersuchen sowie Störgrößen explizit in Kauf zu nehmen. Daher ist es passend, dass die Fensterfront in Richtung Süden ausgerichtet ist und der Raum K004b als Büro genutzt wird. Daher wird im nächsten Abschnitt das Modell an diese beiden Faktoren angepasst.

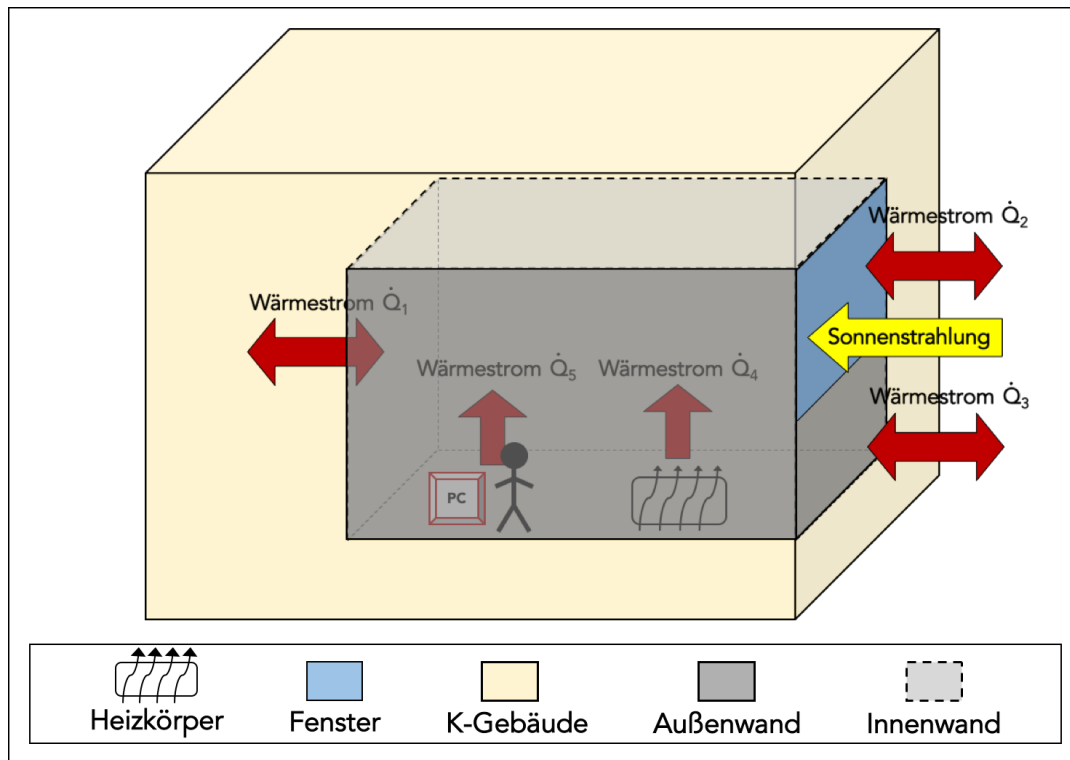


Abb. 4.3: Erweitertes Raummodell

4.5 Modellerweiterung durch Berücksichtigung von Störgrößen und der Sonneneinstrahlung

Der Raum K004b wird regulär als Büro genutzt, weshalb verschiedene Faktoren als Störgrößen in Bezug auf die Raumtemperatur betrachtet werden können. Zum einen wird durch die Menschen und deren Rechner weitere Wärme in den Raum eingebracht und zum Anderen werden die Fenster und die Türen manchmal geöffnet. Des Weiteren trifft insbesondere auf die südseitigen Fenster Sonnenstrahlung, welche ebenfalls Wärme in den Raum einbringen und damit einen Einfluss auf die Raumtemperatur haben.

```
equation
2  [...]
   /* calculate derivative of the inner energy */
4  der(room_u) = environment_qdot + building_qdot + window_qdot + radiator_qdot + sun_qdot +
   otherfactors_qdot;
```

Listing 4.4: Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der Sonneneinstrahlung und Störgrößen

4.6 Validierung des Modells

4.7 Anpassung des Modells mit Parameterschätzung

*„Knowing where things are, and why, is essential to
rational decision making.“*

— JACK DANGERMUND, Esri

5 Schlussbetrachtung

5.1 Fazit

5.2 Ausblick und Ansatzpunkte für weitere Arbeiten

Welche art der Verwendung? MPC mit JModelica.org also deren mpc klasse eigene in casadi etc?

A Fertiges Raummodell in Modelica

```

1 package room_model_backup

3 connector Temperature
  Modelica.SIunits.Conversions.NonSIunits.Temperature_degC t;
5 end Temperature;

7 connector HeatFlow
  Modelica.SIunits.HeatFlowRate qdot;
9 end HeatFlow;

11 connector MassTemperature
  Modelica.SIunits.Conversions.NonSIunits.Temperature_degC t;
13 Modelica.SIunits.MassFlowRate mdot;
  end MassTemperature;

15 connector RadiantEnergyFluenceRate
17 Modelica.SIunits.DensityOfHeatFlowRate radiation_sun;
  end RadiantEnergyFluenceRate;

19
21 model CV_Radiator "control volume for a discretized radiator"

23 /** parameter **/
  outer parameter Modelica.SIunits.CoefficientOfHeatTransfer u_radiator
    "heat transfer coefficient of the radiator";
25 outer parameter Modelica.SIunits.SpecificHeatCapacity cp_water
    "specific heat capacity of water";
27 outer Modelica.SIunits.Mass cv_m
    "mass within one control volume";
29 outer Modelica.SIunits.Area exchange_surface
    "surface of one control volume at which heat transfer takes place";
31 Modelica.SIunits.Energy cv_u
    "inner energy of the control volume";
33 Modelica.SIunits.HeatFlowRate cv_qdot
    "heatflowrate over the borders of the control volume";
35 Modelica.SIunits.Conversions.NonSIunits.Temperature_degC cv_temperature_out(start=21.2,
    fixed=true)
    "temperature of the fluid leaving the control volume";

37 /** states **/
39 outer Modelica.SIunits.Conversions.NonSIunits.Temperature_degC room_temperature_cv
    "temperature within the room";

41 /** controls **/
43 Modelica.SIunits.Conversions.NonSIunits.Temperature_degC cv_temperature_in=inlet.t
    "temperature of the fluid streaming in the control volume";
45 outer Modelica.SIunits.MassFlowRate mdot
    "massflowrate within the radiator";

47
49 equation
  /* calculate inner energy */
  cv_u = cv_m * cp_water * cv_temperature_out;
51 /* calculate derival of the inner energy */
  der(cv_u) = mdot * cp_water * (cv_temperature_in - cv_temperature_out) - cv_qdot;
53 /* calculate heatflowrate */
  cv_qdot = u_radiator * exchange_surface * (cv_temperature_out - room_temperature_cv);
55 /* commit calculated temperature */
  outlet.t = cv_temperature_out;

```

```

57 end CV_Radiator;

59 model Radiator "model for a discretized radiator within a room"

61 /** parameter **/
parameter Real radiator_element_number=106
63   "number of normed elements of which the radiator consists";
parameter Real radiator_tubes_element=2
65   "number of parallel tubes in one element";
parameter Integer cv_number = 20
67   "number of control volumes in which the radiator is discretized";
parameter Modelica.SIunits.Length radiator_element_length=0.045
69   "length of one element depending on type (Recknagel 2013/2014: Heizung und Klimatechnik
    S.815ff.)";
parameter Modelica.SIunits.Height tube_length_vertical=0.4
71   "height of one element depending on type (Recknagel 2013/2014: Heizung und Klimatechnik
    S.815ff.)";
parameter Modelica.SIunits.Diameter tube_diameter_horizontal=0.05
73   "diameter of the horizontal tubes depending on type (Recknagel 2013/2014: Heizung und
    Klimatechnik S.815ff.)";
parameter Modelica.SIunits.Diameter tube_diameter_vertical=0.0255
75   "diameter of the vertical tubes depending on type (Recknagel 2013/2014: Heizung und
    Klimatechnik S.815ff.)";
parameter Modelica.SIunits.Density rho_water = 1000
77   "density of water";
parameter Modelica.SIunits.Mass radiator_element_mass=0.35
79   "mass within one element of the radiator depending on type (Recknagel 2013/2014:
    Heizung und Klimatechnik S.815ff.)";
inner parameter Modelica.SIunits.CoefficientOfHeatTransfer u_radiator = 12.9872
81   "heat transfer coefficient of the radiator";
inner parameter Modelica.SIunits.SpecificHeatCapacity cp_water = 4200
83   "specific heat capacity of water";
CV_Radiator[cv_number] cv_radiator
85   "array of control volumes to discretize the radiator";
Modelica.SIunits.HeatFlowRate radiator_qdot_out
87   "heatflow which is leaving the radiator";
Modelica.SIunits.Conversions.NonSIunits.Temperature_degC radiator_temperature_out
89   "calculated(predicted) temperature of the water leaving the radiator";
inner Modelica.SIunits.Mass cv_m
91   "mass within one control volume";
inner Modelica.SIunits.Area exchange_surface
93   "surface of one control volume at which h";

95 /** states **/
inner Modelica.SIunits.Conversions.NonSIunits.Temperature_degC room_temperature_cv
97   "temperature within the room for the control volume";
outer Modelica.SIunits.Conversions.NonSIunits.Temperature_degC room_temperature
99   "temperature within the room from the room";

101 /** controls **/
inner Modelica.SIunits.MassFlowRate mdot=inlet.mdot
103   "massflowrate within the radiator";
Modelica.SIunits.Conversions.NonSIunits.Temperature_degC radiator_inlet = inlet.t
105   "temperature of the inflowing fluid";

107 equation
  /* calculate surface of one control volume */
109   exchange_surface = (radiator_element_number * radiator_element_length * 2 *
    Modelica.Constants.pi * tube_diameter_horizontal + radiator_element_number *
    radiator_tubes_element * tube_length_vertical * Modelica.Constants.pi *
    tube_diameter_vertical)/cv_number;

```



```

111  /* calculate mass within one control volume */
112  cv_m = (radiator_element_mass * radiator_element_number)/cv_number;
113  /* commit temperature of radiator fluid inlet to the first control volume */
114  cv_radiator[1].inlet.t = radiator_inlet;
115  /* commit fluid temperatures within the radiator control volumes */
116  for i in 1 : (cv_number-1) loop
117    connect( cv_radiator[i].outlet, cv_radiator[i+1].inlet);
118  end for;
119  /* save fluid temperature of the last radiator control volumes */
120  radiator_temperature_out=cv_radiator[cv_number].outlet.t;
121  /* calculate and save the heatflowrate which is leaving the radiator*/
122  radiator_qdot_out = sum(cv_radiator.cv_qdot);
123  /* commit roomtemperature*/
124  room_temperature=room_temperature_cv;
125  outlet.mdot = mdot;
126  outlet.t = radiator_temperature_out;
127  end Radiator;
128
129  model Room_radiator_window "model of a room for mpc purpose with JModelica.org"
130
131    /** parameter p **/
132    parameter Modelica.SIunits.Length room_length=7.81
133      "length of the room";
134    parameter Modelica.SIunits.Breadth room_breadth=5.78
135      "breadth of the room";
136    parameter Modelica.SIunits.Height room_height=2.99
137      "height of the room";
138    parameter Modelica.SIunits.Length window_length=7
139      "length of the window";
140    parameter Modelica.SIunits.Height window_height=2.08
141      "height of the window";
142    parameter Modelica.SIunits.Density rho_air = 1.2
143      "density of air";
144    parameter Modelica.SIunits.CoefficientOfHeatTransfer u_glass=2.0
145      "heat transfer coefficient for glass ";
146    parameter Modelica.SIunits.CoefficientOfHeatTransfer u_wall=0.612986
147      "heat transfer coefficient for the walls of the room";
148    parameter Modelica.SIunits.SpecificHeatCapacity cp_air=1000
149      "specific heat capacity of air";
150    Radiator heating
151      "instance of a radiator";
152    Window window
153      "instance of a window";
154    Modelica.SIunits.Volume room_volume
155      "volume of the room";
156    Modelica.SIunits.Mass room_mass
157      "mass of air within the room";
158    Modelica.SIunits.Area building_surface
159      "sum of contacting surfaces (walls) with other rooms of the building";
160    Modelica.SIunits.Area environment_surface
161      "sum of contacting surfaces (walls) with the environment";
162    inner Modelica.SIunits.Area window_surface
163      "sum of contacting surfaces (windows) with the environment";
164    Modelica.SIunits.Energy room_u
165      "inner energy of the system room";
166    Modelica.SIunits.HeatFlowRate building_qdot
167      "rate of heat flow with other rooms within the building";
168    Modelica.SIunits.HeatFlowRate environment_qdot
169      "rate of heat flow with the environment";
170    Modelica.SIunits.HeatFlowRate environment_qdot_wall
171      "rate of heat flow with the environment through the wall";

```

```

171   Modelica.SIunits.HeatFlowRate environment_qdot_window
      "rate of heat flow with the environment through the window";
173   Modelica.SIunits.HeatFlowRate qdot_loss
      "summed up rate of heatflow leaving the system";
175   Modelica.SIunits.HeatFlowRate radiator_qdot
      "heat flow rate at the radiator surfaces streaming into the room";
177
179   /** states x */
      inner Modelica.SIunits.Conversions.NonSIunits.Temperature_degC room_temperature(start
          =24, fixed=true)
          "temperature within the room (Initially about 24 degree celsius)";
181
183   /** controls u */
      Modelica.SIunits.MassFlowRate mdot=inlet_radiator.mdot
          "commitment of the massflowrate to the radiator";
185   Modelica.SIunits.Conversions.NonSIunits.Temperature_degC environment_temperature=
          inlet_environment.t
          "temperature of the environment";
187   Modelica.SIunits.Conversions.NonSIunits.Temperature_degC building_temperature=
          inlet_building.t
          "temperature of the rest of the building";
189   Modelica.SIunits.HeatFlowRate qdot_sun
          "rate of heat flow brought in by the sun";
191   Modelica.SIunits.HeatFlowRate qdot_otherfactors=inlet_other.qdot
          "rate of heat flow brought in by other factors (e.g. people, computer)";
193
195   equation
      /* calculate room volume */
      room_volume=room_length*room_height*room_breadth;
197   /* calculate room mass */
      room_mass=room_volume*rho_air;
199   /* calculate surface of the room with other rooms of the building */
      building_surface=(room_length*room_breadth*2)+(room_length*room_height*2);
201   /* calculate wall surface of the room with the environment */
      environment_surface=(room_length*room_height)+(room_breadth*room_height)-window_surface;
203   /* calculate window surface of the room with the environment */
      window_surface=(window_length*window_height);
205   /* calculate inner energy*/
      room_u = room_mass * cp_air * room_temperature;
207   /* calculate derival of the inner energy */
      der(room_u) = radiator_qdot + qdot_loss + qdot_sun + qdot_otherfactors;
209   /* sum up the lost heat flow */
      qdot_loss = building_qdot+environment_qdot;
211   /* calculate lost heatflow with other rooms of the building */
      building_qdot = u_wall * building_surface * (building_temperature-room_temperature);
213   /* sum up the lost heatflow with the environment */
      environment_qdot = environment_qdot_window + environment_qdot_wall;
215   /* calculate lost heatflow with the environment through the window */
      environment_qdot_window = u_glass * window_surface * (environment_temperature -
          room_temperature);
217   /* calculate lost heatflow with the environment through the wall */
      environment_qdot_wall = u_wall * environment_surface * (environment_temperature -
          room_temperature);
219   /* commit the inflowing heat flow of the radiator */
      radiator_qdot=heating.radiator_qdot_out;
221   /* connect radiator and sun with room */
      connect(heating.inlet, inlet_radiator);
223   connect(inlet_sun, window.inlet_sun);
      qdot_sun = window.outlet_room.qdot;
225   end Room_radiator_window;

```

```

227 model SourceTemp
    Modelica.SIunits.Conversions.NonSIunits.Temperature_degC t=inlet;
229    Modelica.Blocks.Interfaces.RealInput inlet;
    equation
231    outlet.t=t;
    end SourceTemp;
233
    model SourceHeat
235    Modelica.SIunits.HeatFlowRate qdot=inlet;
    Modelica.Blocks.Interfaces.RealInput inlet;
237    equation
    outlet.qdot=qdot;
239    end SourceHeat;

241 model SourceTempMass
    Modelica.SIunits.Conversions.NonSIunits.Temperature_degC t=inlet_t;
243    Modelica.SIunits.MassFlowRate mdot=inlet_mdot;
    Modelica.Blocks.Interfaces.RealInput inlet_t;
245    Modelica.Blocks.Interfaces.RealInput inlet_mdot;
    equation
247    outlet.t=t;
    outlet.mdot=mdot;
249    end SourceTempMass;

251 model Window
    Modelica.SIunits.DensityOfHeatFlowRate radiation_arriving=inlet_sun.radiation_sun;
253    Modelica.SIunits.HeatFlowRate qdot_effective;
    parameter Real window_transmission = 0.00687213;
255    outer Modelica.SIunits.Area window_surface;
    equation
257    qdot_effective = radiation_arriving * window_transmission * window_surface;
    qdot_effective = outlet_room.qdot;
259    end Window;

261
    model SourceSun
263    Modelica.SIunits.DensityOfHeatFlowRate radiation_sun = inlet;
    Modelica.Blocks.Interfaces.RealInput inlet;
265    RadiantEnergyFluenceRate outlet;
    equation
267    outlet.radiation_sun = radiation_sun;
    end SourceSun;
269
    end room_model_backup;

```

Literaturverzeichnis

[pys] *Pysolar by pingswept*

[osi 1996] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: ISO 7498-1: Information technology – Open Systems Interconnection – Basic Reference Model: The basic model. Genf, Switzerland, März 1996. – ISO 7498-1 Standard

[mod 2006a] MODBUS ORGANIZATION: Modbus messaging on TCP/IP implementation guide v1.0b. Hopkinton, USA, Oktober 2006. – Forschungsbericht

[mod 2006b] MODBUS ORGANIZATION: Modbus over serial line specification and implementation guide V1.02. Hopkinton, USA, Dezember 2006. – Forschungsbericht

[mod 2012] MODBUS ORGANIZATION: Modbus application protocol specification v1.1b3. Hopkinton, USA, April 2012. – Forschungsbericht

[AB 2015] AB, Modelon (Hrsg.): *JModelica.org User Guide - Version 1.17*. Ideon Science Park, SE-223 70 Lund : Modelon AB, 2015

[Andersson 2013] ANDERSSON, Joel: *A General-Purpose Software Framework for Dynamic Optimization*. Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, Arenberg Doctoral School, KU Leuven, Diss., October 2013

[Baehr u. Kabelac 2012] BAEHR, Hans ; KABELAC, Stephan: *Thermodynamik: Grundlagen und technische Anwendungen*. 15. Auflage 2012. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012

[Bertrand Blanc 2005] BERTRAND BLANC, Bob M.: *Endianness or Where is Byte 0?* <http://3bc.bertrand-blanc.com/endianness05.pdf>, December 2005

[Böckh u. Wetzel 2014] BÖCKH, Peter v. ; WETZEL, Thomas: *Wärmeübertragung: Grundlagen und Praxis*. 5. überarbeitete und erweiterte Auflage 2014. Berlin, Heidelberg : Springer Vieweg, 2014

[Furrer 2003] FURRER, Frank J.: *Industriautomation mit Ethernet-TCP/IP und Web-Technologie*. 3. neu bearbeitete und erweiterte Auflage. Heidelberg: Hüthig, 2003

[Hauser 2000] HAUSER, Höttges K G.: Bauphysik in Kürze: U-Werte von Fenstern. In: *Bauphysik 22* (2000), S. S.270–273

[Joel Andersson 2015] JOEL ANDERSSON, Moritz D. Joris Gillis G. Joris Gillis: *User Documentation for CasADi v2.2.0+1.cf4be18*. Joel Andersson, Joris Gillis, Moritz Diehl, 2015

[Kaltschmitt 2013] KALTSCHMITT, Wolfgang ; Wiese A. Martin ; Streicher ;. Martin ; Streicher: *Erneuerbare Energien : Systemtechnik, Wirtschaftlichkeit, Umweltaspekte*. 5. erweiterte Auflage. Springer Berlin Heidelberg, 2013

[MODICON 96] MODICON, Inc.: *Modicon Modbus Protocol Reference Guide*. One High Street, North Andover, Massachusetts 01845: MODICON, Inc., Industrial Automation Systems, June 96

- [Nicolai 2013] NICOLAI, Dr. A.: Physikalische Grundlagen des thermischen Raummodells THERAKLES / Technische Universität Dresden, Fakultät Architektur, Institut für Bauklimatik. 2013. – Forschungsbericht
- [Peter Häupl 2013] PETER HÄUPL, Christian Kölzow Olaf Riese Anton Maas Gerrit Höfker Christian N. Martin Homann ; WILLEMS, Wolfgang (Hrsg.): *Lehrbuch der Bauphysik : Schall - Wärme - Feuchte - Licht - Brand - Klima*. Springer Vieweg, 2013
- [Quaschnig 2011] QUASCHNIG, Volker: *Regenerative Energiesysteme : Technologie, Berechnung, Simulation*. Bd. 7. aktualisierte Auflage. München : Hanser, 2011
- [Recknagel 2013] RECKNAGEL, Schramek Sprenger: *(Taschenbuch für Heizung + Klimatechnik) : 76.2013/14*. Oldenbourg Industrieverlag, 2013
- [Reda 2008] REDA, A. I. ; A. I. ; Andreas: Solar Position Algorithm for Solar Radiation Applications (Revised) / National Renewable Energy Laboratory (NREL), Golden, CO. 2008. – Forschungsbericht
- [Sack 2004] SACK, Dipl.-Phys. N.: Von k zu U - Was ändert sich bei Fensterrahmen und -profilen? / ift Rosenheim Bauphysik. 2004. – Forschungsbericht
- [Schleicher 2008] SCHLEICHER, Manfred: *Digitale Schnittstellen und Bussysteme*. JUMO, 2008
- [Schnell u. Wiedemann 2006] SCHNELL, Gerhard ; WIEDEMANN, Bernhard: *Bussysteme in der Automatisierungs- und Prozesstechnik : Grundlagen, Systeme und Trends der industriellen Kommunikation*. 6. überarbeitete und aktualisierte Auflage. Wiesbaden : Vieweg+Teubner, 2006

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Quellen und Hilfsmittel angefertigt habe. Alle Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Quellen entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegen.

Kandel, den 21. März 2016

Daniel Johannes Mayer