

Diplomarbeit

Entwurf mechatronischer Systeme auf der methodischen Grundlage des Systems Engineering Konzepts

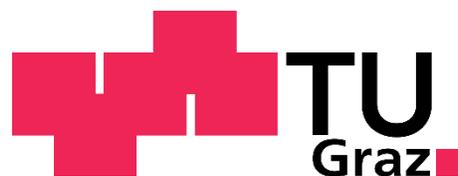
Michael Hörzer

Matr.-Nr.: 0530263

Institut für Maschinenbau- und Betriebsinformatik

Technische Universität Graz

Vorstand: Univ.-Prof. Dipl.-Ing. Dr.techn. Siegfried Vössner



Begutachter: Ass.-Prof. Dipl.-Ing. Dr.techn. Gerald Lichtenegger

Betreuer: Dipl.-Ing. Dietmar Neubacher

Graz, Februar 2014

Kurzfassung

Im Zuge der vorliegenden Arbeit sollte ein technisches System entwickelt und gebaut werden, welches dazu dient, Tätigkeiten an Prüf- und Versuchseinrichtungen zu vereinfachen. Für die Entwicklung dieses Systems sollten Grundlagen des Systems Engineering (SE) Anwendung finden. Ziel war es, die Grundsätze des Systems Engineering anhand des Entwicklungsprozesses eines mechatronischen Produktes greifbar zu machen. Dabei sollten Gestaltungsprinzipien wie das *Systemdenken*, das *Vorgehen vom Groben ins Detail* sowie das *Denken in Varianten* helfen, die Qualität des Zielsystems zu verbessern. Für die Systementwicklung wurde das Vorgehensmodell nach *Hall-BWI* zur Anwendung gebracht, welches sich dadurch auszeichnet, dass es Gestaltungsprinzipien, Phasenmodell und Problemlösungszyklus vereint. Neben allgemeinen Methoden und Werkzeugen des SE, wie die Systemgestaltung durch Modellbildung, wurden einige im Produktentwicklungsprozess entstehende problemspezifische Vorgehensweisen aufgezeigt und beschrieben. Anhand des Beispiels ließ sich zeigen, wie ausgehend von einem allgemeinen Modell eines mechatronischen Systems der Untersuchungsbereich abgesteckt und in weiterer Folge der Gestaltungsbereich eingrenzt werden kann. Dies gilt unter der Voraussetzung, dass das Zielsystem einen mechatronischen Charakter aufweist. Durch stufenweises Detaillieren konnte ein System gebaut werden, welches seinen Anforderungen gerecht wird. Trotz erfolgreicher Produktentwicklung zeigten sich auch einige Kritikpunkte am Vorgehensmodell, welche sich auf dessen *plangetriebenen* Charakter zurückführen ließen. Insbesondere Produkte oder Systeme, die sich durch Interdisziplinarität auszeichnen, bedürfen, so scheint es, einem hohen Grad an Flexibilität sowohl im Projektablauf als auch im Prozess der Problemlösung.

Abstract

In the course of this work, a technical System should be designed and built, which is purposed to simplify the work on proofing and testing facilities. For the development of these Systems, fundamentals of systems engineering (SE) should find application. The main goal was to demonstrate the principles of systems engineering on the basis of a mechatronic product development process. Design principles such as systems thinking, proceeding from rough to detail and also thinking in variants should be helpful to improve the quality of the target system. For the system development the process model by *Hall-BWI* was used, which is distinguished by the fact that it combines design principles, phase model and problem-solving cycle. Beside common methods and tools of SE, such as model based system design (MBSD), some during the development process discovered methods specific to the problem have been shown and described. The example demonstrates, how to mark out the area of investigation and subsequently to fence off the area of design, on the basis of the common mechatronic systems model. This can only be done on the condition that the target system shows up mechatronic characteristics. By detailing the system step by step a product could be built that meets its requirements. Despite to a successful product development, some criticisms to the process model showed up, which could be attributed to its plan driven character. Especially products or systems, that distinguish by interdisciplinary, seem to need a high value of flexibility in the process of the project as well as in the process of solving problems.

Danksagung

Zunächst möchte ich mich bei all jenen bedanken, welche sowohl durch fachliche Hilfestellung wie auch durch persönliche Unterstützung zum Gelingen dieser Arbeit beigetragen haben.

Besonderer Dank gilt natürlich auch Herrn Prof. Dr. Vössner, der bereitwillig sein Einverständnis für die Betreuung dieser Diplomarbeit am Institut für Maschinenbau- und Betriebsinformatik gegeben hat. Ebenso möchte ich mich auch bei Herrn Prof. Dr. Lichtenegger wie auch Herrn Dipl.-Ing. Neubacher für die herzliche und engagierte Betreuung der Arbeit bedanken.

Des Weiteren gilt mein Dank Herrn Dipl.-Ing. Rupert Rohrmoser wie auch Herrn Dr. Daniel Kollreider sowohl für ihre Unterstützung als auch für ihre Bemühungen, ohne welche diese Arbeit gar nicht erst zustande gekommen wäre. Des Weiteren möchte ich mich bei meinen Arbeitskollegen und auch all jenen Mitarbeitern der Firma LOGICDATA bedanken, welche mich bei der Ideenfindung unterstützt haben und jederzeit für Fragen zur Verfügung standen.

Auch meinen Freunden und Kollegen, die mich in meiner Studienzeit begleitet haben, möchte ich meinen Dank ausdrücken.

Zu guter Letzt möchte ich mich bei meiner Familie bedanken, die mir meine bisherige Ausbildung ermöglicht hat und mich stets in allen Lebenslagen unterstützte. Insbesondere bin ich meinem Vater Josef Hörzer dankbar dafür, dass er mir während dieser Arbeit immer wieder den Rücken gestärkt hat und mit Rat und Tat zur Seite stand.

Graz, Februar 2014

Michael Hörzer

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am

.....

(Unterschrift)

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....

date

.....

(signature)

Inhaltsverzeichnis

1	Einleitung	12
1.1	Motivation.....	12
1.2	Zielsetzung	13
1.3	Aufbau und Inhalt.....	14
2	Systeme.....	16
2.1	Systemtheorie	16
2.1.1	Systembegriff	17
2.1.2	Allgemeine Systemtheorie	17
2.1.3	Kybernetik	18
2.1.4	Spieltheorie	19
2.1.5	Chaostheorie	19
2.2	Klassifikation, Bestandteile und Aufbau	20
2.2.1	Systemaspekte	21
2.2.2	Systemfunktion	24
2.2.3	Systemstruktur	25
2.2.4	Systemhierarchie.....	26
2.3	Modellbildung und Simulation	28
2.3.1	Modell	28
2.3.2	Simulation	30
2.3.3	Modellierungssprachen.....	31
3	Systems Engineering	34
3.1	Grundlagen.....	35
3.1.1	Begriff und Definition	36
3.1.2	Systemdenken	37
3.1.3	Problem, Methode und Lösung.....	38

3.1.4	Vorgehensmodell	39
3.2	Abgrenzung	43
3.2.1	Aufgaben des Systems Engineering	43
3.2.2	Rollenverteilung im Systems Engineering	45
3.3	Ausgewählte Modelle.....	49
3.3.1	Münchner Vorgehensmodell (Lindemann, 2009).....	49
3.3.2	SIMILAR-Prozess nach <i>Bahill</i> und <i>Gissing</i> (Bahill & Gissing, 1998)	52
3.3.3	SE-Konzept nach <i>Hall-BWI</i> (Haberfellner, Weck, Fricke, & Vössner, 2012)	54
4	Mechatronische Systeme.....	58
4.1	Mechatronik.....	58
4.2	Aufbau mechatronischer Systeme	60
4.2.1	Systemelemente.....	60
4.2.2	Systemstruktur	63
4.3	Modellierung mechatronischer Systeme	65
4.3.1	Strukturuntersuchung	65
4.3.2	Technische Modellbildung	66
4.3.3	Technische Simulation.....	68
5	Fallbeispiel	69
5.1	Anstoß.....	70
5.1.1	Ausgangssituation	70
5.1.2	Problembewusstsein	70
5.1.3	Entscheidung.....	72
5.2	Vorstudie.....	73
5.2.1	Problemidentifikation	73
5.2.2	Untersuchungsbereich	74
5.2.3	Gestaltungsbereich	78

5.2.4	Projektziele.....	83
5.2.5	Lösungsprinzipien.....	86
5.2.6	Machbarkeit	93
5.2.7	Projektentscheidung	94
5.3	Hauptstudie.....	94
5.3.1	Projektteam und Budget	95
5.3.2	Systemarchitektur	95
5.3.3	Gesamtkonzept	100
5.3.4	Teilprojekte	106
5.4	Detailstudie	108
5.4.1	Detailkonzept	108
5.4.2	Teilprobleme	110
5.4.3	Teillösungen	111
5.4.4	Integrierbarkeit	114
5.5	Systembau.....	117
5.5.1	Rahmenbedingungen	117
5.5.2	Hardware.....	117
5.5.3	Software	121
5.5.4	Tests	124
5.6	Systemeinführung und Projektabschluss	126
6	Fazit.....	127
6.1	Zusammenfassung	127
6.2	Stellungnahme	129
6.3	Ausblick	132
	Literaturverzeichnis	133
	Anhang.....	137

Abbildungsverzeichnis

Abbildung 1: Modellierung von Systemen (vgl. Stein, 2003/04)	30
Abbildung 2: Beispiel für die Darstellung eines ER-Modells.....	32
Abbildung 3: SysML Diagrammtypen (vgl. OMG, 2012, S. 167)	32
Abbildung 4: NASA - SE in context of overall project management (NASA, 2007, S. 4)	43
Abbildung 5: Das Münchener Vorgehensmodell (MVM) (Lindemann, 2009, S. 47).....	50
Abbildung 6: SIMILAR Prozess nach BAHILL und GISSING (vgl. Bahill & Gissing, 1998).....	52
Abbildung 7: Systems Engineering Konzept nach Hall-BWI.....	54
Abbildung 8: Mechatronik-Diagramm des Rensselaer Polytechnic Institute (Rohde, 2008).....	59
Abbildung 10: Beispiel für ein mechanisches System	61
Abbildung 11: Beispiel für ein elektrisches System.....	61
Abbildung 12: Getaktetes RS-Flip-Flop (vgl. Tietze & Schenk, 2002).....	62
Abbildung 13: Allgemeine Darstellung eines mechatronischen Systems (vgl. TransMechatronic, 2014) .	64
Abbildung 14: Bond-Graph eines mechanischen und elektrischen Systems.....	68
Abbildung 15: Dauerlaufprüfstände der Fa. LOGICDATA, v. l. n. r.: Säulenprüfstand, Hebelprüfstand, Massagemotorenprüfstand	74
Abbildung 16: Beispiele für Aktuatoren bzw. Prüflinge, v. l. n. r. Inlineantrieb, 90-Grad-Antrieb, Massagemotor	75
Abbildung 17: Steuerelektronik, v. l. n. r. Motorsteuerungen, digitaler Handschalter, analoger Handschalter	76
Abbildung 18: Zusammenwirken der einzelnen Prüfstandkomponenten.....	77
Abbildung 19: Geschlossene Wirkungskette innerhalb des Hebelprüfstandes.....	79
Abbildung 20: Abgrenzung des Gestaltungsbereich am Modell eines mechatronischen Systems	81
Abbildung 21: Mind-Map zu denkbaren Zuständen des Prüfstandes	87
Abbildung 22: Mind-Map möglicher Fehlerfälle bzw. der Lage möglicher Fehler.....	87
Abbildung 23: Mind-Map erfassbarer Messdaten	88
Abbildung 24: Drei Modelle als Varianten von Lösungsprinzipien.....	90
Abbildung 25: Variantenbildung und Detaillierung des gewählten Lösungsprinzips	92
Abbildung 26: Modell des Zielsystems (Black-Box) und seiner Umgebung.....	96
Abbildung 27: Auswahl an Umwelteinflüssen in industrieller Umgebung	97
Abbildung 28: Vertikale Integration in der Automatisierungstechnik.....	98

Abbildung 29: Top-Down Vorgehen bei der Auswahl des Mikrocontrollers mit Variantenbildung in jeder Ebene.	99
Abbildung 30: Erste Detaillierungsebene der Systemstruktur des Zielsystems (Grey-Box)	100
Abbildung 31: Drei-Phasen-Modell zur Sensorauswahl	101
Abbildung 32: Prinzipvergleich der Messprinzipien bei der Temperaturmessung.....	102
Abbildung 33: Produktvergleich am Beispiel von Widerstandsthermometern.....	103
Abbildung 34: Variantenvergleich am Beispiel des gewählten DS1820 Sensors	104
Abbildung 35: Phasenmodell für die Detailstudie als Alternative zu Teilprojekten	107
Abbildung 36: Vereinfachtes Modell des Detailsystems	109
Abbildung 37: Simulationsmodell zur technischen Modellbildung am Beispiel der Motorspannungsmessung	114
Abbildung 38: Qualitative Darstellung der Ergebnisse aus der technischen Simulation	115
Abbildung 39: Ergebnis der Lösungsfindung zum Teilproblem der Spannungsmessung am Motor	115
Abbildung 40: Bottom-Up Vorgehen beim Feststellen der Integrierbarkeit	116
Abbildung 41: Schrittweises Vorgehen in der Phase der Hardwarerealisierung.....	117
Abbildung 42: Schaltplan des Zielsystems.....	118
Abbildung 43: Schrittweises Vorgehen im Leiterplattenentwurf.....	119
Abbildung 44 Leiterplattendesign des Zielsystems	119
Abbildung 45: Fertig mit Bauteilen bestückte Leiterplatte des Zielsystems	120
Abbildung 46: Softwareuntersuchung durch iterative Analyse und Synthese	122
Abbildung 47: Systemidentifikation zur Untersuchung der Fremdsoftware (Black-Box)	123
Abbildung 48: Modell der durch Analyse vereinfachten Fremdsoftware	123
Abbildung 49: Variantenbildung in der Phase der Softwareentwicklung.....	124
Abbildung 50: Untersuchung der Übertragungsqualität der Telnet-Kommunikation als Beispiel für Einzeltests	125
Abbildung 51: Schaltplan	137
Abbildung 52: Leiterplattendesign.....	138
Abbildung 53: User interface design index.html	139
Abbildung 54: User interface design setup.html.....	140

Tabellenverzeichnis

Tabelle 1: Grundprinzipien des Handelns (vgl. Lindemann, 2009, S. 55 ff) (vgl. Winzer, 2013, S. 15 ff) ...	42
Tabelle 2: Die Elemente des Münchner Vorgehensmodells (vgl. Lindemann, 2009, S. 47 ff)	51
Tabelle 3: Elemente im SIMILAR-Prozess (vgl. Bahill & Gissing, 1998) (vgl. Haberfellner, Weck, Fricke, & Vössner, 2012)	53
Tabelle 4: Projektphasen im Vorgehensmodell nach Hall-BWI (Haberfellner, Weck, Fricke, & Vössner, 2012)	56
Tabelle 5: Teilschritte des Problemlösungszyklus nach Hall-BWI (Haberfellner, Weck, Fricke, & Vössner, 2012)	57
Tabelle 6: Potential- und Flussgrößen in verschiedenen Fachgebieten	67
Tabelle 7: Projektziele	85
Tabelle 8: Beispiele für Kausalzusammenhänge zwischen Fehler und Messdaten	88
Tabelle 9: Vor- und Nachteile des Server-Modells	91
Tabelle 10: Vor- und Nachteile des Standalone-Modells	92
Tabelle 11: Ungefähre Preise von am Markt erhältlichen Produkten	94
Tabelle 12: Kurzbeschreibung verschiedener Messprinzipien zur Temperaturmessung	102
Tabelle 13: Tabellarische Übersicht der gewählten Sensoren	104
Tabelle 14: Verwendeten Web-Services und ihr Einsatzzweck	106
Tabelle 15: Verwendete Software für die Durchführung von Einzeltests	125

1 Einleitung

Seit mehr als 50 Jahren spielt Systems Engineering eine wesentliche Rolle bei der Entwicklung moderner Produkte. Dabei hat diese Disziplin, ausgehend von der Rüstungsindustrie sowie der Luft- und Raumfahrt, Einzug gehalten in verschiedenste Branchen und gewann zunehmend an Bedeutung. Moderne Produkte und Systeme weisen einen hohen Grad an Komplexität auf, weshalb diese ohne geeignete Maßnahmen nicht mehr zu realisieren sind. Auch steigende Qualitätsanforderungen an industrielle Produkte sind treibender Faktor für die Anwendung von SE-Methoden. Die Komplexität ist mitunter bedingt durch das Zusammenwachsen vieler Fachrichtungen zu einem interdisziplinären Feld. Die Mechatronik stellt ein Paradebeispiel für die Symbiose verschiedener eigenständiger Disziplinen dar. Dank der Elektronik und der Informatik verfügen moderne technische Produkte über ein gewisses Maß an „Intelligenz“, wodurch diese sich wesentlich von ihren Vorgängern unterscheiden. Dieser Umstand zwingt viele, bislang rein durch den Maschinenbau geprägte Unternehmen, zu einem Umdenken und zu einer Neu- oder Umgestaltung ihrer bisherigen Erzeugnisse. Neuentwicklungen verlangen somit eine interdisziplinäre und damit ganzheitliche Herangehensweise an ihren Entwurf und ihre Gestaltung. Aufgrund dieser Gegebenheiten entsteht gleichzeitig ein Bedarf an Fachkräften, welche mit den veränderten Umgebungsbedingungen umgehen können.

1.1 Motivation

Im Laufe meines Studiums an der Technischen Universität Graz wurde mir zunehmend die Bedeutung der Mechatronik in der modernen Produktentwicklung bewusst, was dazu führte, dass ich mich für diesen Studiengang entschieden habe. Dabei konnte ich jedoch feststellen, dass in keiner der von mir besuchten Lehrveranstaltungen der Gegenstand des Fachbereichs Mechatronik eindeutig geklärt wurde. Auch der Schwerpunkt dieses Studiengangs liegt meines Erachtens im Bereich des allgemeinen Maschinenbaus. Einzig die studienzweigspezifischen Fächer lassen eine Tendenz in Richtung Elektrotechnik, Regelungstechnik und Automatisierungstechnik erkennen. Durch mein persönliches Interesse an Elektrotechnik und vor allem der Elektronik habe ich mich zwischenzeitlich in dieser Abteilung im TU Graz Racing Teams engagiert und viel neue Erfahrungen sammeln dürfen. Dieses Engagement führte unter anderem auch dazu, dass ich seit geraumer Zeit in der Firma LOGICDATA tätig bin.

Die Firma LOGICDATA Electronic Software Entwicklungs- GmbH ist Innovationsführer und internationale Größe im Bereich microcontrollerbasierter Motorsteuerungen und Bedienelemente für ergonomische Lösungen der Möbelbranche (LOGICDATA, 2014). Mittlerweile haben sich die firmeneigenen

Kompetenzen, neben der Elektronik und Embedded Systems, auch auf die Entwicklung mechatronischer Produkte ausgeweitet. Kerngebiet der Mechatronikabteilung ist die Entwicklung von Linearaktuatoren welche zum größten Teil in verstellbaren Tisch- und Bettsystemen zum Einsatz kommen.

Im Laufe meiner Tätigkeit entstand die Möglichkeit ein Projekt aus dem Bereich der Mechatronik zu bearbeiten, welches voraussichtlich einen tieferen Einblick in den Bereich der Automatisierungstechnik, der Elektronik und der Informatik bietet. Die Tatsache, dass gerade eben diese Fachbereiche im Studium leider nur eine Nebenrolle spielten, motivierte mich besonders dieses Projekt im Zuge meiner Diplomarbeit durchzuführen.

So war es notwendig, aufgrund anfangs noch begrenzter Fachkenntnisse in den benötigten Themengebieten, systematisch vorzugehen, um die Aufgabenstellung bestmöglich zu erfüllen. Damit ergab sich der Bezug zum Thema Systems Engineering, was auch den Schwerpunkt dieser Arbeit darstellt.

1.2 Zielsetzung

Teilgebiet der Entwicklung neuer Produkte ist ein systematisches Überprüfen der Funktionalität und Qualität während des gesamten Entstehungsprozesses, von der Konzeptphase bis hin zur Serienfertigung. Ziel dieser Qualitätsplanungsmaßnahme ist die Gewährleistung, dass die Konstruktion bzw. das Endprodukt den Anforderungen im Einsatzfall gerecht wird. Auch bei der Konstruktion mechatronischer Systeme ist die Versuchsplanung wesentlicher Bestandteil zur Feststellung der Eignung des Produktes in seinem zukünftigen Anwendungsgebiet. Um Versuche oder Tests zu bewerkstelligen, kommen verschiedene Formen von Prüfständen zum Einsatz. Aus vielen verschiedenen Prüfverfahren mit unterschiedlichster Parametrierung ergibt sich eine Vielzahl durchzuführender Versuche, woraus ein enormer Aufwand zur Überwachung und Wartung entsteht. Diesen Aufwand gilt es durch Automatisierung, aber auch durch Vereinfachung des Prüfverfahrens zu reduzieren. Aus dieser Situation entstand auch der Wunsch nach der Entwicklung eines Systems, welches die Mitarbeiter in ihren Tätigkeiten unterstützen kann. Primäres Ziel hierbei ist eine Verbesserung der Prüfstandsituation. Um dies zu erreichen, soll nach den Prinzipien des Systems Engineering vorgegangen werden.

Die vorliegende Arbeit befasst sich folglich nicht mit der technischen Umsetzung dieses Systems, sondern legt den Fokus auf die praktische Anwendung von Grundlagen aus der Fachdisziplin Systems Engineering. Aus diesem Grund wird versucht, nur auf relevantes technisches Fachwissen einzugehen, welches für diverse Darstellungen im praktischen Teil von Nöten ist. Hauptaugenmerk wird auf die Vorgehensweise gelegt, die letztlich zur Lösung der Aufgabenstellung führt. Damit sollen die Grundsätze des Systems

Engineering am Prozess der Produktentwicklung mechatronischer Systeme veranschaulicht werden, wobei Handlungsprinzipien und die Anwendung eines geeigneten Vorgehensmodells im Vordergrund stehen. So werden schließlich auch jene Methoden aufgezeigt, die im Zuge der Entwicklungstätigkeit identifiziert werden können.

1.3 Aufbau und Inhalt

Thema dieser Arbeit ist der Entwurf mechatronischer Systeme auf der methodischen Grundlage des SE-Konzepts. Neben der praktischen Anwendung des Systems Engineering Konzepts gilt es zunächst grundlegende Fragen zu beantworten, die mit der Thematik in Zusammenhang stehen.

Die Diplomarbeit gliedert sich in einen theoretischen und einen praktischen Teil. Der theoretische Teil beschäftigt sich mit Systemen im Allgemeinen, der Bedeutung des Systems Engineering und der Mechatronik. Der praktische Teil behandelt das Fallbeispiel der Entwicklung eines mechatronischen Teilsystems zur Verbesserung der Prüfstandsituation.

Im zweiten Kapitel wird versucht, die Ursprünge der Systemtheorie und des Systemdenkens zu ergründen. Danach erfolgt eine nähere Betrachtung des Begriffs *System* und eine kritische Beleuchtung einiger dazugehöriger Definitionen. Anschließend werden die Systembestandteile erklärt und Möglichkeiten zur Klassifizierung von Systemen gesucht. Hierbei werden einige Systemmerkmale dargelegt und der Frage auf den Grund gegangen, wie diese sich äußern. Schließlich folgt ein kurzer Einblick in allgemeine Möglichkeiten zur Systemdarstellung.

Das dritte Kapitel widmet sich den Grundlagen des Systems Engineering. Nach einem Klärungsversuch, was unter Systems Engineering verstanden wird, sollen auch dessen Grenzen abgesteckt werden. Zentrale Frage hierbei ist, wodurch sich Systems Engineering von ähnlichen Disziplinen wie Systems Architecture und Systems Design unterscheidet. Schließlich werden für die praktische Umsetzung des Fallbeispiels drei verschiedene SE-Konzepte oder Vorgehensmodelle betrachtet, wovon eines zur praktischen Systementwicklung angewendet werden soll.

Im vierten Kapitel erfolgt ein genauerer Blick auf den Fachbereich der Mechatronik. Die gestellten Fragen sind einerseits, was Mechatronik bedeutet und andererseits wie ein mechatronisches System definiert werden kann. Danach wird ein Einblick in die Modellbildung derartiger Systeme gegeben.

Im fünften Kapitel folgt der praktische Teil dieser Arbeit, in dem das erwähnte Prüfstandsystem zu entwickeln und umzusetzen ist. Aufgabe des Systems ist es, die mit den Prüfständen verbundenen

Tätigkeiten zu vereinfachen und eine Möglichkeit zu schaffen, diese auch von den Arbeitsplätzen der Mitarbeiter aus durchzuführen. Steuerungs-, Überwachungs- und Wartungsaufgaben sollen einfacher und besser automatisiert werden. Das entworfene Teilsystem soll in den Prüfstand integriert werden. Dabei wird angestrebt, grundlegende Prinzipien und Methoden des Systems Engineering auf den Entwurf des Zielsystems zu übertragen. Ziel ist es, eine Aussage darüber zu treffen, in welchen Phasen der Entwicklung welche Prinzipien zur Anwendung gebracht werden können und inwiefern sich diese im vorliegenden Beispiel als effektiv erweisen.

Abschließend soll eine kritische Beurteilung der Anwendbarkeit des gewählten SE-Konzepts auf das Fallbeispiel erfolgen. Dabei ist festzuhalten, wo sich Probleme bei der Anwendung ergeben haben und worauf diese zurückzuführen sind. Neben den identifizierten Hindernissen sollen auch die Vorteile des angewandten SE-Konzepts in Hinblick auf das Fallbeispiel erfasst werden.

2 Systeme

In vielen verschiedenen Wissenschaften, wie auch den Ingenieurwissenschaften, ist der Begriff des Systems von zunehmender Bedeutung. In diesem Kapitel soll der Systembegriff näher untersucht und dabei die folgenden vier Fragen beantwortet werden:

1. Was ist ein System?
2. Welche Arten von Systemen gibt es?
3. Woraus bestehen Systeme?
4. Wie lassen sich Systeme darstellen?

Im ersten Abschnitt wird kurz der Systembegriff erklärt. Durch die vielen verschiedenen Wissenschaftsdisziplinen und unterschiedlichen Auffassungen des Begriffes *System* hat sich eine Vielzahl an Definitionen ergeben. Diese weisen jedoch viele Gemeinsamkeiten auf. Nach einem Klärungsversuch des Systembegriffes wird ein kurzer Einblick in die Systemtheorie gewährt.

Im zweiten Abschnitt wird versucht eine Antwort auf die Fragen nach den Arten und Bestandteilen von Systemen zu geben. Dabei wird auf die Möglichkeit zur Klassifizierung von Systemen eingegangen. Auch diese ist stark von unterschiedlichen Wissenschaftsdisziplinen geprägt. Besonderes Augenmerk wird auf die Klasse der komplexen Systeme gelegt. Anschließend werden die Elemente und der Aufbau eines Systems gezeigt und die Hierarchie sowie die Struktur von Systemen erläutert.

Im dritten und letzten Abschnitt sollen Methoden und Werkzeuge für die graphische und formelle Darstellung von Systemen erläutert werden. Zu Beginn erfolgt eine nähere Begutachtung der Begriffe *Modell* und *Simulation*, wonach einige aktuelle Modellierungssprachen offengelegt werden. Für die Darstellung von Systemen haben sich verschiedene Modellierungssprachen etabliert, die dazu dienen, sich einen Überblick über ein System zu verschaffen. Systemdarstellung ist ein bedeutendes Fachgebiet des Systems Engineering und soll hier überblicksartig gezeigt werden.

2.1 Systemtheorie

Mit der Beschreibung von Systemen, insbesondere mit den oft komplexen inneren und äußeren Zusammenhängen, setzt sich die Systemforschung oder auch Systemtheorie auseinander.

Biologe, Humanwissenschaftler und Systemtheoretiker Ludwig von Bertalanffy gilt als Begründer der Allgemeinen Systemtheorie. In seinen Werken wie „An Outline of General Systems Theory“ von 1950 und

„General System Theory: Foundations, development, applications“ von 1978 publizierte er diese neue Art des Denkens, deren Motto „Einheit durch Vielfalt“ (engl. *unity through diversity*) lautet (vgl. Hofkirchner, 2005).

In Kombination mit der Kybernetik bildet die Allgemeine Systemtheorie die Grundlage für die moderne Systemtheorie. Kybernetik bezeichnet dabei die Wissenschaft der Steuerung und Kommunikation von Systemen (siehe Kapitel 2.1.3).

Über diese und zwei weitere eng verknüpfte Theorien – die Spieltheorie und die Chaostheorie – soll hier ein Überblick geschaffen werden. Zunächst ist jedoch der Systembegriff zu klären.

2.1.1 Systembegriff

Der Begriff *System* stammt aus dem spätlateinischen *systema* bzw. griechischen *sýstēma* und bedeutet so viel wie „ein aus mehreren Teilen zusammengesetztes, gegliedertes Ganzes“. (vgl. DUDEN, 2014).

In der Literatur findet sich eine Vielzahl an Definitionen für ein System. PATZAK beschäftigt sich in seinem Werk „Systemtechnik – Planung komplexer innovativer Systeme“ von 1982 mit verschiedensten Auffassungen. Seine sowie andere sehr bekannte Definitionen lauten wie folgt:

„Unter einem System soll die Gesamtheit von Elementen verstanden werden, die miteinander durch Beziehungen verbunden sind.“ (Daenzer, 1976/77, S. 11)

„Ein System besteht aus einer Menge von Elementen, welche Eigenschaften besitzen und durch Relationen miteinander verknüpft sind.“ (Patzak, 1982, S. 19)

„Ein System ist eine in einem betrachteten Zusammenhang gegebene Anordnung von Gebilden, die miteinander in Beziehung stehen. Diese Anordnung wird aufgrund bestimmter Vorgaben von ihrer Umgebung abgegrenzt.“ (DIN19226)

Zusammenfassend kann also gesagt werden, dass ein System, bestehend aus Elementen und Beziehungen, als ein Ganzes zu verstehen ist. Diese zwei Bestandteile alleine erscheinen jedoch nicht ausreichend zur Definition eines Systems. Es fehlt beispielsweise eine Aussage darüber, wodurch sich ein System von seiner Umgebung abgrenzt, d.h. der Lage von Systemgrenzen. Die Eigenschaften der Elemente und der Beziehungen sind ebenfalls wesentlich für ein System. Ein weiterer, offensichtlich vernachlässigter Aspekt ist der Sinn und Zweck eines Systems, der nicht minder von Bedeutung ist.

2.1.2 Allgemeine Systemtheorie

Die von Bertalanffy begründete Allgemeine Systemtheorie kann als Ausgangspunkt für sämtliche systemtheoretische Ansätze gesehen werden. Wesentlicher Inhalt dieser Theorie ist die generelle

Systembetrachtung, also ein Denken in Systemen. Der Begriff der Komplexität, die Offenheit sowie die Dynamik von Systemen und der Versuch, solche Systeme (mathematisch) zu beschreiben, sind ebenfalls Teil dieser Theorie.

„Die Systemtheorie ist in ihren Grundzügen eine disziplinübergreifende Wissenschaft, welche versucht, die abstrakte Anordnung von Phänomenen unabhängig ihrer Substanz, Typ, räumlicher oder zeitlicher Ausdehnung der Existenz des jeweiligen Phänomens zu beschreiben und versucht, die gemeinsamen Prinzipien aller komplexen Systeme zu entdecken und diese mathematisch zu formulieren.“ (Masak, Der Architekturreview, 2010, S. 268)

2.1.3 Kybernetik

Als Begründer der Kybernetik gilt der US-amerikanische Mathematiker Norbert Wiener. In seinem Werk „Regelung und Nachrichtenübertragung im Lebewesen und in der Maschine“ von 1948 prägte er diesen Begriff als die Wissenschaft der Steuerung und Regelung von komplexen Systemen wie Maschinen, lebenden Organismen oder auch sozialen Strukturen.

Der Begriff *Kybernetik* entstammt dem griechischen *kybernētiké* und bedeutet so viel wie Steuermannskunst. (vgl. DUDEN, 2014a)

Die Kybernetik als wissenschaftliches Forschungsgebiet untersucht verschiedene Systeme wie technische Maschinen oder auch den menschlichen Organismus. Anhand der erkannten Gemeinsamkeiten zwischen derartigen Systemen – in Hinblick auf selbsttätige Regelungs- und Steuerungsmechanismen – konnte eine Theorie entwickelt werden, welche diese Mechanismen in ihrer Gesamtheit erklärt. Ziel der Systemuntersuchungen ist es, Aussagen über die innere Struktur der Systeme zu treffen, um Modelle zu entwerfen, die das Systemverhalten nach den eigenen Bedürfnissen beeinflussen können. (vgl. Bandte, 2007)

Kybernetik erster Ordnung – Die von Wiener begründete Kybernetik wird in der Fachliteratur oft als *Kybernetik erster Ordnung* bezeichnet. Diese legt ihren Schwerpunkt auf Fließgleichgewichte, also der Mengengleichheit von der in das System ein- und ausfließenden Materie und/oder Energie. Durch einen Vergleich von Ist- und Sollwerten sowie einer Abweichungsanalyse sollen mittels Rückkopplungsschleifen (*feedback-loop*) Gleichgewichtszustände hergestellt werden (vgl. Bardmann, 2011).

Diese Betrachtungsweise entspricht dem *Black-Box* Prinzip, bei dem die inneren Zusammenhänge des Systems unbekannt sind (vgl. Kapitel 2.2.2).

Kybernetik zweiter Ordnung – Die modernen Betrachtungen in der Kybernetik gehen zurück auf den österreichischen Physiker und Systemtheoretiker Heinz von Foerster und seinen Begriff der *Kybernetik*

zweiter Ordnung. Dieser, von Foerster auch als *Kybernetik der Kybernetik* bezeichneten Theorie, liegt die Tatsache zugrunde, dass die Systembeschreibung wesentlich vom Beobachter abhängt. Die Kybernetik zweiter Ordnung legt den Schwerpunkt auf die Auswirkungen des Beobachters auf das beobachtete System. (vgl. Bandte, 2007)

Diese Beeinflussung durch den Beobachter ist auch aus der modernen Quantentheorie bekannt.

2.1.4 Spieltheorie

Die Bezeichnung *Spieltheorie* lässt auf den ursprünglichen Zusammenhang mit Gesellschaftsspielen schließen, anhand derer diese Theorie anschaulich erklärt werden kann. Die Entscheidung eines Spielers für einen strategischen Spielzug ist nicht nur auf das System des Spiels und den vorangegangenen Entschlüssen des Spielers, sondern auch auf Entscheidungen anderer Beteiligter (beispielsweise dem Gegenspieler) zurückzuführen. Aufgabe der Spieltheorie ist es, Rückschlüsse auf den Verlauf eines solchen Spiels, das von mehreren Entscheidungsträgern abhängig ist, zu ziehen. (vgl. Rieck, 1993)

„Gegenstand der Spieltheorie sind Entscheidungssituationen, in denen das Ergebnis für einen Entscheider nicht von seinen eigenen Entscheidungen abhängt, sondern auch von dem Verhalten anderer Entscheider. Spieltheorie ist also eine Theorie sozialer Interaktion.“ (Rieck, 1993, S. 16)

Entscheidungstheorie – Im Gegensatz dazu beschäftigt sich die *Entscheidungstheorie* mit einfachen Entscheidungssituationen und deren Beeinflussung durch die Umwelt sowie mit dem Einfluss der Entscheidungen auf die Umwelt.

„Die klassische Entscheidungstheorie (Entscheidungstheorie im engeren Sinn) betrachtet ausschließlich Situationen, in denen gegen die Natur gespielt wird. Das Wort Natur steht hier für eine Wahrscheinlichkeitsverteilung über Umweltzustände, die vom eigenen Verhalten abhängig sind.“ (Rieck, 1993, S. 16)

2.1.5 Chaostheorie

Betrachtungsgegenstand der Chaostheorie sind nichtlineare dynamische Systeme, bestehend aus deterministischen Entitäten, welche jedoch im Zusammenspiel als System ein nicht determinierbares Systemverhalten zeigen. Hierbei wird vor allem dem Begriff der Emergenz besondere Bedeutung beigemessen (siehe Kapitel 2.2.1).

Andreas Liening gibt folgende vier notwendige Bedingungen für das Entstehen von Chaos in dynamischen Systemen an (vgl. Liening, 1999):

- Nicht-Linearität: Die Ausgangsgröße eines Systems verhält sich nicht proportional zur Eingangsgröße.

- Determinismus: Das System weist eine deterministische Dynamik auf.
- Rekursion: Das System entsteht durch Iteration oder Rückkopplung.
- Endogenität: Es existieren keine externen Störgrößen.

Dynamische Systeme lassen sich im Regelfall durch entsprechende mathematische Modelle beschreiben, auch wenn deren Verhalten regellos zu sein scheint (z.B. aperiodische Schwingungen). Charakteristisch für solche chaotischen Vorgänge ist, dass geringste Veränderungen der Anfangsbedingungen (z.B. der Anfangsauslenkung eines nichtlinearen Oszillators) ein völlig anderes Ergebnis zur Folge haben können. (vgl. Czichos & Hennecke, HÜTTE - Das Ingenieurwissen, 2012)

Da ein exaktes Wiederholen eines Experimentes unmöglich ist, lassen sich solche Systeme mit deterministisch chaotischer Dynamik in der Praxis nicht analysieren. Mit Hilfe von numerischen Berechnungsmethoden und genauen Angaben zu den Anfangsbedingungen lässt sich das zeitliche Verhalten solcher Systeme eindeutig und wiederholbar beschreiben, weshalb man in einem solchen Fall von einem *deterministischen Chaos* spricht.

Erkenntnisse aus dem Bereich der Quantenphysik zeigen aber auch das Vorhandensein von nicht deterministischen chaotischen Systemen. Hierauf soll jedoch nicht näher eingegangen werden.

2.2 Klassifikation, Bestandteile und Aufbau

Systeme lassen sich nach verschiedenen Merkmalen oder Eigenschaften, so genannten *Systemaspekten*, klassifizieren. Die Summe der einzelnen Merkmale des Systems gibt Aufschluss darüber, um welche Art von System es sich handelt. Beispielsweise kann ein Unternehmen als ein reales, künstliches, offenes, dynamisches, zielorientiertes, komplexes, soziotechnisches, selbstorganisierendes und lernfähiges System betrachtet werden.

Unabhängig von der Art des Systems, besteht dieses wie in Kapitel 2.1.1 beschrieben aus Elementen und Beziehungen. Was das System genau ist, obliegt dem Beobachter des Systems. Der Beobachter definiert das System und seine Grenzen. (vgl. Masak, Der Architekturreview, 2010)

In Bezug auf diesen Gesichtspunkt lassen sich die gezeigten Definitionen erweitern. Ein System ist demnach eine willkürlich abgegrenzte Menge von Elementen. In der Fachliteratur werden neben Elementen und Beziehungen noch weitere Systembestandteile identifiziert. Üblicherweise bedient man sich hierbei der hierarchischen Systemdarstellung. Im Gegensatz zu dieser Systemhierarchie, durch welche Systembestandteile aufzeigt werden können, zeigt die Systemstruktur den Aufbau eines Systems.

2.2.1 Systemaspekte

Je nach Wissenschaftsgebiet sind unterschiedliche Aspekte mehr oder weniger von Bedeutung. Im Fachbereich der Regelungstechnik sind beispielsweise Zeitkontinuität, Zeitvarianz oder Linearität von besonderer Bedeutung.

In der Fachliteratur werden häufig folgende Systemaspekte unterschieden:

- Real oder ideell
- Einfach, kompliziert oder komplex
- Offen, geschlossen oder isoliert
- Deterministisch oder stochastisch
- Natürlich oder künstlich
- Zweck- oder zielorientiert
- Sozial, technisch oder soziotechnisch
- Zeitvariant oder zeitinvariant
- Statisch oder dynamisch
- Zeitdiskret oder zeitkontinuierlich

Im Folgenden sollen drei für das SE wichtige Systemaspekte näher besprochen werden.

Offenheit von Systemen – Die Offenheit von Systemen bezeichnet deren Wechselwirkung mit der Umwelt.

„A system is closed if no material enters or leaves it; it is open if there is import and export and, therefore, change of the components.“ (Bertalanffy, 1950, S. 23)

Kategorisiert man Systeme nach deren Beziehung zu ihrer Umwelt, können drei verschiedene Systemtypen unterschieden werden. Die unten angeführten Definitionen sind sinngemäß in verschiedenster Fachliteratur zu finden.

- Offene Systeme: Wechselwirkt ein System mit seiner Umwelt, wobei ein Austausch von Energie und Materie erfolgt, so wird dieses System als offen bezeichnet. Grundsätzlich handelt es sich bei den meisten realen Systemen um offene Systeme.
- Geschlossene Systeme: Besitzt ein System lediglich eine energetische Beziehung zu seiner Umwelt, wobei keine Materie ausgetauscht wird, so bezeichnet man es als geschlossen.
- Abgeschlossene (isolierte) Systeme: Hat ein System keinerlei Beziehung zu seiner Umwelt, so wird es als abgeschlossen bezeichnet.

Komplexität – Hinsichtlich der Definition des Begriffes *Komplexität* gibt es in der Literatur verschiedenste Auffassungen, dennoch spielt sie stets eine vordergründige Rolle. Komplexität kann als Ursache für die Entstehung der wissenschaftlichen Disziplin der Systemtechnik gesehen werden. Komplexität liegt im

Auge des Betrachters. So werden Systeme, die ein Einzelner nicht mehr überblicken kann, oft als komplex bezeichnet. Aus systemtheoretischer Sicht sind komplexe Systeme jene, die nicht auf einer einfachen Parametrisierung oder einer simplen Kausalität beruhen. (vgl. Masak, Moderne Enterprise Architekturen, 2005)

Die Ansätze zur Beschreibung von mechanischen Systemen aus der Newton'schen Mechanik folgen dem Kausalitätsprinzip. Kausalität bezeichnet den direkten Zusammenhang zwischen Ursache (Aktion) und deren Wirkung (Reaktion). Solche Systeme bestehen meist aus wenigen einfachen Teilen mit wenigen wechselwirkenden Beziehungen. (vgl. Masak, Moderne Enterprise Architekturen, 2005)

Oft lassen sich Wirkungen, aufgrund von zeitlicher Latenz, ihrer Ursache nicht mehr eindeutig zuordnen. Dieses Verhalten kann die Komplexität eines Systems steigern. Systeme, die dem Kausalitätsprinzip unterliegen, bezeichnet man üblicherweise als einfache Systeme. (vgl. Masak, Der Architekturreview, 2010)

Komplexe Systeme weisen hingegen meist folgende fünf Charakteristika auf (Masak, Der Architekturreview, 2010, S. 269 ff):

1. Offenheit: Offene Systeme stehen in Wechselwirkung zu ihrer Umgebung.
2. Flexibilität: Flexibilität bezeichnet das Vorhandensein von Freiheitsgraden, wodurch verschiedene Möglichkeiten gegeben sind, auf äußere Einflüsse (*inputs*) zu reagieren. Je mehr Alternativen dafür zur Verfügung stehen, desto höher ist die Zahl der Freiheitsgrade und umso größer ist die Flexibilität des Systems. Sie ist sowohl von der inneren Struktur des Systems als auch von der Umwelt abhängig. Einschränkungen in der Flexibilität bezeichnet man auch als *Zwangsbedingungen*.
3. Dimensionalität: Die Dimension von Systemen steht im Zusammenhang mit ihren Parametern. Komplexe Systeme sind mehrdimensional, d.h. sie hängen von mehreren Parametern ab. Eindimensionalität bedeutet eine einfache Kausalität.
4. Emergenz: Diese bezeichnet die Abhängigkeit des Systemverhaltens vom Zusammenwirken und Wechselwirken seiner einzelnen Teile. Aus der Selbstorganisation der Teile entstehen neue Ordnungen.

„Das Ganze ist mehr als die Summe seiner Teile“. [Aristoteles]

Die Voraussetzungen für Emergenz in einem System sind:

- a. die Wechselwirkung der einzelnen Teile

- b. ein Mindestmaß an Komplexität
 - c. Reproduktion der Systemteile (*feedback*)
5. Nichtintuitivität: Aus der Definition von Komplexität lässt sich ableiten, dass intuitive Systeme nicht komplex sind. Aufgrund der Wechselwirkungen lassen sich die Auswirkungen des Systems nicht eindeutig vorherbestimmen. Die Kausalität geht verloren, was bedeutet, dass Wirkungen nicht mehr auf ihre Ursachen zurück geführt werden können oder nicht mehr unterscheidbar sind (Was war Ursache und was ist Wirkung?). Entsprechend der Kybernetik zweiter Ordnung kann die reine Beobachtung eines Systems Einfluss auf dessen Verhalten haben.

Diese Eigenschaften bestimmen wesentlich ob es sich um ein komplexes System handelt.

Dynamik – Als dynamische Systeme werden jene bezeichnet, deren Verhalten sich über einen bestimmten Zeitraum verändert und somit ein dynamisches aufweisen.

Zur Beschreibung von dynamischen Systemen ist oft die Kenntnis der Zustandsgrößen essentiell. In der Praxis sind Aussagen über das zeitliche Verhalten von Systemen und den Veränderungen der Systemgrößen sowie die damit verbundenen Auswirkungen auf die Umwelt von Interesse. Oft ist die Beobachtung der äußeren Zusammenhänge nicht ausreichend, um Aussagen über das Systemverhalten treffen zu können. In vielen Fällen (z.B. zur Realisierung eines Zustandsreglers) müssen die Zustandsgrößen erfasst (gemessen) werden. Die Zustandsgrößen sind jene Größen, die eindeutig den Zustand des Systems beschreiben (z.B. Auslenkwinkel eines Pendels). (vgl. Bossel, 2004)

Bossel unterscheidet zwei Ursachen für die Entstehung von Systemdynamik (Bossel, 2004):

1. Intern erzeugte Systemdynamik: Bei der intern erzeugten Systemdynamik spielen Rückkopplungen eine wesentliche Rolle. Rückkopplungen bedeuten, dass Zustandsgrößen Einfluss auf Zustandsveränderungen haben. Dadurch kann es zu einem analytisch nicht mehr vorhersehbaren Verhalten kommen, welches bereits als *deterministisches Chaos* bekannt ist. Solche Systemverhalten können nur noch durch numerische Simulationen modelliert und analysiert werden. Dynamisches Verhalten, das sich nicht auf äußere Einflüsse zurückführen lässt, wird auch als Eigendynamik bezeichnet.

Bei dynamischen Systemen müssen anstelle linearer Modelle der Art „aus A folgt B“ so genannte zirkuläre Modelle der Form „aus A folgt B und aus B folgt A“ angewandt werden. (Haberfellner, Weck, Fricke, & Vössner, 2012)

2. Extern erzeugte Systemdynamik: Extern erzeugte Dynamik ist auf Einwirkungen der Systemumwelt zurückzuführen. Umwelteinwirkungen sind vom System unabhängige, konstante oder variable Parameter.

Hier besteht ein Zusammenhang zwischen den Systemeigenschaften Dynamik und Komplexität. *Ulrich* und *Probst* definieren die Komplexität als:

„[...] Fähigkeit eines Systems, in einer gegebenen Zeitspanne eine große Zahl von verschiedenen Zuständen annehmen zu können.“ (Ulrich & Probst, 1995, S. 58)

In Bezug auf diese beiden Eigenschaften unterscheiden sich vier verschiedene Systemtypen (Ulrich & Probst, 1995, S. 58 ff):

Einfache Systeme – Einfache Systeme zeichnen sich durch ihre geringe Anzahl an Entitäten und Beziehungen aus. Sie genügen einer einfachen Kausalität und lassen sich analytisch beschreiben, d.h., dass sie im Regelfall eindeutig vorhersagbar sind.

Komplizierte Systeme – Komplizierte Systeme weisen eine Vielzahl an Systemelementen (hohe Varietät) ohne dynamische Beziehungen auf.

Dynamisch komplizierte Systeme – Dynamisch komplizierte Systeme sind Systeme von geringer Varietät und mit schwachen dynamischen Interaktionen.

Komplexe Systeme – Komplexe Systeme sind durch eine große Anzahl von Elementen und dynamischen Beziehungen gekennzeichnet. Im Gegensatz zu komplizierten Systemen weisen sie chaotischen Charakter auf.

2.2.2 Systemfunktion

Hierbei handelt es sich um eine Betrachtung der äußeren Wirkzusammenhänge, d.h. das System wird auf Basis seines Verhaltens und seiner Schnittstellen zur Umwelt betrachtet. Der innere Aufbau des Systems ist nicht von Relevanz. Das System wird als so genannte *Black-Box* aufgefasst. Lediglich die Eingangsgrößen, die Ausgangsgrößen und der Zustand des Systems sind von Interesse.

An dieser Stelle sei angemerkt, dass *Systemfunktion* nicht im Sinne von einer mathematischen Funktion aufzufassen ist. Der Begriff beschreibt vielmehr den Zusammenhang zwischen Eingangs- und Ausgangsgröße. Die Bezeichnung als *Wirkungsorientierte Betrachtung* erscheint somit vernünftiger.

2.2.3 Systemstruktur

Bei der strukturellen Betrachtung des Systems wird dessen innerer Aufbau untersucht. Sofern die innere Struktur vollständig bekannt ist, spricht man von *White-Box*. Sind lediglich Teile der inneren Struktur bekannt, so bezeichnet man die Struktur als *Grey-Box*. Im Gegensatz zur funktionalen Betrachtung kann hier tatsächlich die mathematische Funktion des Systems von Interesse sein, die als *Übertragungsfunktion* bezeichnet wird. Durch die strukturelle Betrachtung können auch Kausalzusammenhänge zwischen Eingängen und Ausgängen analysiert werden.

Systeme weisen aufgrund der Beziehungen zwischen ihren Elementen eine innere Ordnung, eine so genannte Systemstruktur, auf. Diese innere Ordnung der Systeme verschafft dem Betrachter eine bessere Übersicht über das oft komplexe System und sorgt somit für ein besseres Verständnis des Gesamten. (vgl. Haberfellner, Weck, Fricke, & Vössner, 2012)

„Jede Form der Organisation eines Systems gibt diesem auch eine Struktur. Aus Sicht des Systems lässt sich der Begriff Struktur wie folgt definieren: Die Struktur ist eine Eigenschaft des Systems, durch das dieses die Freiheitsgrade der Elemente und Subsysteme einschränkt.“ (Masak, Der Architekturreview, 2010, S. 275)

Beispiel – Diese Definition der Struktur kann anhand des Beispiels eines mechanischen Systems leicht erklärt werden. Betrachtet man einen einfachen massebehafteten Körper im dreidimensionalen Raum, so hat dieser sechs Freiheitsgrade (drei translatorische und drei rotatorische). Nun kann dieser Körper über einen masselosen Stab und ein Kugelgelenk mit einer starren Wand verbunden werden. Das bedeutet, dass der Körper mit der Wand eine Beziehung eingeht. Körper und Wand sind somit Elemente des gebildeten Systems. Durch die Beziehung verliert der massebehaftete Körper jedoch an Bewegungsfreiheit bzw. an Freiheitsgraden. Einzelne Attribute dieser Beziehung, wie z.B. die Länge des Stabes, bestimmen nun wesentlich die Lage des Körpers.

Dadurch ist aber auch ein Zusammenhang zwischen Systemstruktur und Komplexität gegeben. Die Struktur und vor allem die Beziehungen zwischen den einzelnen Systemelementen können sich auf zwei entgegengesetzte Arten auf die Komplexität auswirken. Einerseits steigt die Komplexität des Systems durch zusätzliche Beziehungen und weitere Systemelemente, andererseits reduzieren Beziehungen die Freiheitsgrade, woraus eine Reduktion der Flexibilität und schließlich auch der Komplexität folgt.

Dieser Umstand lässt sich unter anderem an dem gewählten Beispiel veranschaulichen. Wird der massebehaftete Körper über den Stab starr mit der Wand verbunden, so verliert der Körper jegliche Freiheitsgrade. Seine relative Lage zur Wand ist konstant und nur durch die Länge des Stabes bestimmt.

2.2.4 Systemhierarchie

Bei der hierarchischen Betrachtung arbeitet man sich von der obersten zur untersten Systemebene vor. Das bedeutet eine Arbeitsweise von der *Black-Box* hin zur *White-Box*. Diese Methodik bezeichnet man auch als *Top-Down-Vorgehen*. Man spricht hierbei auch von *Systems-Design-Methoden*. Die drei wesentlichsten hierbei sind:

- *Top-Down*
- *Bottom-Up*
- *Meet-in-the-Middle*

Im Gegensatz zu Top-Down kann auch in umgekehrter Weise vorgegangen werden. Ein solches schrittweises Vereinfachen eines komplexen Systems bezeichnet man als *Bottom-Up-Vorgehen*. In der Praxis wird jedoch meist das so genannte *Meet-in-the-Middle* Prinzip angewandt. Bei *Meet-in-the-Middle* wird der größte Teil nach der Top-Down-Methode aber auch einige Systembestandteile nach einem *Bottom-Up-Vorgehen* entwickelt. (vgl. Gajski, Abdi, Gerstlauer, & Schirner, 2009, S. 35ff)

Die Betrachtungsweise der Systemhierarchie dient der Reduktion der Komplexität. Hierbei wird das betrachtete System in Untersysteme (Subsysteme) gegliedert. Jedes Untersystem wird in weitere Subsysteme aufgeteilt bis zum kleinsten Element. So entsteht eine hierarchische Darstellung des gesamten Systems. Ebenso ist jedes System einem so genannten Übersystem (Supersystem) untergeordnet. Das Aufteilen eines Systems in seine Komponenten bezeichnet man auch als *Systemdekomposition*. (vgl. Schneider & Vecellio, 2011)

Dabei hängt der Detaillierungsgrad davon ab, ob eine detailliertere Auflösung notwendig ist oder ob dadurch die Übersicht verloren geht. (vgl. Haberfellner, Weck, Fricke, & Vössner, 2012)

Im Unterschied zur Hierarchie weisen Systeme auch Rekursionen auf. Nicht der hierarchische Aspekt der Unterordnung, sondern der Aspekt der sich wiederholenden Strukturen steht hierbei im Zentrum. (vgl. Masak, Der Architekturreview, 2010)

In verschiedenster Literatur zur Thematik finden folgende Systembestandteile Erwähnung (vgl. Haberfellner, Weck, Fricke, & Vössner, 2012):

- Elemente (Entitäten): Elemente sind die kleinste betrachtete Systemebene und werden nicht in weitere Subsysteme aufgeteilt (auch wenn es theoretisch möglich wäre). Jedes Element hat

gewisse Eigenschaften und Aufgaben (Funktionen), die durch qualitative und quantitative Parameter ausgedrückt werden können.

- Teilsystem: In vielen Fällen können Elemente eines Systems sinnvoll zu Teilsystemen zusammengefasst werden. Dies dient ebenfalls einer besseren Überschaubarkeit eines Systems.
- Untersystem (Subsystem): Können Elemente von Systemen in weitere Elemente unterteilt werden, so spricht man von einem Untersystem.
- Übersystem (Supersystem): Jedes System kann als Teil eines übergeordneten Systems betrachtet werden. Das Berücksichtigen des Übersystems hilft dabei, sich ein besseres Bild von der Situation zu machen.
- System von Systemen (System of Systems): Das System von Systemen (SvS) grenzt sich durch die Autonomie der Subsysteme vom Übersystem ab. Ein SvS ist demnach ein System, das aus mehreren, voneinander unabhängigen Systemen besteht. Die Subsysteme können unabhängig vom übergeordneten System und den umgebenden Systemen bestehen.
- Systemgrenzen: Die Systemgrenzen sind mehr oder weniger willkürlich gezogene Grenzen, um das betrachtete System von seiner Umwelt (Umfeld, Umgebung) abzuheben. Systemgrenzen müssen keine physikalischen Grenzen sein und können je nach Betrachtungsweise unterschiedlich gelegt werden. Typisch für die Lage der Systemgrenzen sind jedoch wesentlich stärkere Beziehungen innerhalb des Systems als zwischen System und Umwelt.
- Systemumwelt: Jener Bereich außerhalb der Systemgrenzen wird als Systemumwelt bezeichnet.
- Umsystem: Einzelne Elemente der Umwelt können zu Teilsystemen, so genannten Umsystemen, zusammengefasst werden.
- Beziehungen (Relations): Elemente interagieren mit anderen Elementen innerhalb und außerhalb des Systems. Diese Wechselwirkungen werden auch als Beziehungen bezeichnet. Dabei handelt es sich um einen Austausch von Informationen, Energie oder Materie.
- Schnittstellen (Interface): Die Übergänge über Systemgrenzen werden Schnittstellen genannt. Demnach erfolgen Wechselwirkungen zwischen System und Umwelt über Schnittstellen, welche häufig auch als Ein- und Ausgänge bezeichnet werden. (vgl. Schneider & Vecellio, 2011)
- Rückkopplungen (Feedbacks): Unter Rückkopplung versteht man eine Ausgangsgröße des Systems, die in veränderter Weise wiederum als Eingangsgröße des Systems dient. Rückkopplungen können als treibendes Element für die Komplexität von Systemen gesehen werden. Eine besondere Bedeutung kommt der Rückkopplung in Regelsystemen zu.

2.3 Modellbildung und Simulation

Die Grenzen eines Systems werden vom jeweiligen Beobachter stets mit der Absicht gesetzt, die Komplexität zu reduzieren. Demnach impliziert diese Systembildung eine Modellbildung. (vgl. Masak, Der Architekturreview, 2010)

Mithilfe von Modellen lassen sich Systeme darstellen. Bevor der Frage nachgegangen wird, was Modellbildung im Speziellen bedeutet, soll zunächst der Modellbegriff näher untersucht werden.

2.3.1 Modell

Der Begriff *Modell* entstammt dem italienischen Wort *modello*, was so viel bedeutet wie Muster oder Entwurf. Seinen Ursprung hat es im lateinischen Begriff *modulus*, zu Deutsch Maß(stab). (vgl. DUDEN, 2014b)

Einige der am häufigsten anzutreffenden aber auch kritisierten Definitionen des Modellbegriffes ist, dass ein Modell ein vereinfachtes (häufig auch beschränktes, isomorphes oder abstraktes) Abbild der Wirklichkeit (oder Realität) ist. (vgl. Thomas, 2005)

Kritikpunkt ist der Bezug zur Realität, da nicht notwendigerweise die Wirklichkeit mit einem Modell abgebildet werden muss. Außerdem bedarf der Realitätsbegriff hierbei noch einer Klärung. Grund dafür mag die unterschiedliche Wahrnehmung der Realität aus verschiedenen Perspektiven sein. Unterschiedliche Betrachter nehmen die Realität auf divergente Weise wahr. Es stellt sich somit die Frage, was Realität eigentlich bedeutet. Auf diese Problematik soll hier jedoch nicht näher eingegangen werden.

Eine ähnliche Definition des Modellbegriffes, die für die Betrachtung von Systemen geeigneter erscheint, lautet:

„Unter einem Modell versteht man ein abstraktes Abbild eines Systems.“ (Eley, 2012, S. 3)

Demzufolge bilden Modelle Systeme ab, wobei sie diese abstrahieren. Der Abstraktion wird hier ein hohes Maß an Bedeutung beigemessen. Wie oben bereits angeführt, sind Systeme oft komplexe Konstrukte aus einer Vielzahl von Elementen und dynamischen Beziehungen. Anhand von Modellen, die Elemente und Beziehungen durch Abstraktion auf das für die Analyse Wesentlichste reduzieren, lässt sich die Komplexität von Systemen mindern. Damit wird Modellen eine wichtige Aufgabe, ein Modellzweck, zuteil.

„Ein Modell ist ein vereinfachtes, zielgerichtetes Abbild eines Ausschnitts des von einem Modellierer wahrgenommenen, realen oder geplanten Systems, das für eine definierte Zielsetzung bzw. einen Beobachtungsbereich die wesentlichen Systemeigenschaften enthält bzw. ein analoges Systemverhalten aufweist.“ (Bandte, 2007, S. 199)

Zusammengefasst soll ein Modell...

- ein (reales) System abbilden,
- dieses System vereinfachen (Komplexität reduzieren) und
- eine Analyse des Systems ermöglichen (Modellzweck).

Modellzweck – Allgemein betrachtet ist der Zweck eines Modells, Aussagen über das System treffen zu können, wobei kein Experiment am System, sondern am Modell durchgeführt wird (Simulation). Mögliche Gründe dafür sind (vgl. Stein, 2003/04):

- Das System selbst existiert (noch) nicht.
- Ein Experiment am System ist zu kostspielig.
- Das Experiment beeinflusst das System (z.B. durch Messungen) und damit die Ergebnisse.
- Das Systemverhalten ist nicht beobachtbar (z.B. in der Astrophysik).
- Zustandsänderungen des Systems sind zu langsam oder zu schnell.

Anhand des Modellzwecks lassen sich verschiedene Arten von Modellen klassifizieren. Im Wesentlichen können die folgenden fünf Modellzwecke unterschieden werden (vgl. Kloth, 2012, S. 69):

- Beschreibungsmodelle skizzieren lediglich Verhalten und Struktur der Realität (z.B. in der Architektur).
- Erklärungsmodelle versuchen Verhalten und Struktur der Realität zu begründen, wobei Zusammenhänge und Gesetzmäßigkeiten ergründet und Theorien aufgestellt werden.
- Simulationsmodelle (Vorhersagemodelle) versuchen mittels Erklärungsmodellen Prognosen über zukünftiges Verhalten von Systemen zu treffen.
- Gestaltungsmodelle (Problemlösungsmodelle) sind erweiterte Erklärungsmodelle, die Aussagen über Sinn und Ziele des Modells ermöglichen.
- Entscheidungsmodelle basieren auf Zielvorstellungen und dienen als Entscheidungshilfe.

Modellbildung – Von Modellbildung ist dann die Sprache, wenn versucht wird, das Verhalten eines Systems zu beschreiben. Dies kann auf funktionale oder strukturelle Weise geschehen (siehe Kapitel 2.2).

„Hinter dem Begriff Modellbildung verbirgt sich ein Prozess, in welchem durch Auswahl und Entscheidung für bestimmte Aspekte eine Abstraktion und somit eine Vereinfachung der Realität vorgenommen wird.“ (Kloth, 2012, S. 70)

In technischen Systemen stellt die Modellbildung den Schritt von der Systembeschreibung hin zu einem mathematischen Modell dar. In einem solchen Fall wird meist von technischer Modellbildung gesprochen.

„Um die Kinematik und die Dynamik eines komplexen Systems anschließend behandeln zu können und darauf aufbauend ein Steuerungs- und Regelungskonzept des Systems zu entwickeln, ist immer zuerst eine solche Modellbildung erforderlich, d.h. letztendlich die Bildung eines Satzes mathematischer Beschreibungen des Systemverhaltens.“ (Roddeck, 2006, S. 5)

Modellierung – Der Begriff *Modellierung* umfasst den gesamten Prozess der Modellbildung bis hin zur Validierung und Verifizierung. In Abbildung 1 ist dieser Prozess in seiner Gesamtheit dargestellt.

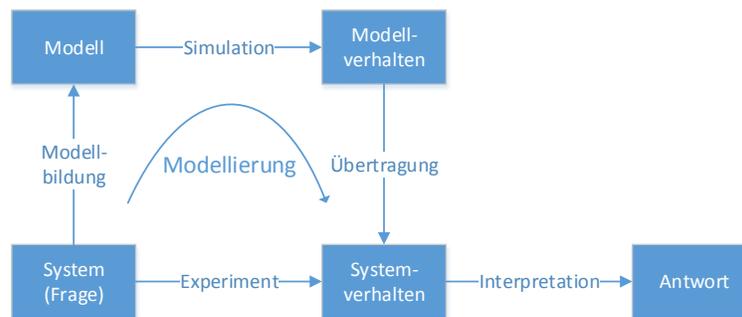


Abbildung 1: Modellierung von Systemen (vgl. Stein, 2003/04)

Metamodell – Ein weiterer häufig in der Systemgestaltung anzutreffender Begriff ist jener des Metamodells. Als *Metamodell* wird ein Modell einer übergeordneten Systemebene verstanden. Das heißt, dass das Modell des Übersystems das so genannte Metamodell ist.

2.3.2 Simulation

Eine Vorgehensweise zur Analyse von Systemen stellt die Simulation dar. Voraussetzung für die Simulation ist ein Simulationsmodell, welches formal beschrieben werden kann. Mithilfe von Simulationen ist es möglich, Prognosen über das Verhalten von Systemen zu treffen, sofern die Anfangsbedingungen für das System bekannt sind.

„Unter einer Simulation versteht man den Prozeß der Bildung einer Prognose mit Hilfe des Experimentierens innerhalb der Modellebene [...] in einem abstrakten Modell eines Systems.“ (Kohorst, 1996)

Computersimulation – Lassen sich mathematische Modelle analytisch nicht mehr lösen, so ist man auf numerische Lösungsverfahren angewiesen. Die meisten mathematischen Modelle werden formal als Differentialgleichungssysteme dargestellt und mithilfe numerischer Verfahren (Algorithmen) rechnergestützt gelöst. Sobald Computer zur Lösung von Modellen zum Einsatz kommen, spricht man von

Computersimulationen. Beispiele für den Einsatz von Computerunterstützung sind die *Mehrkörpersimulationen (MKS)* oder auch die *Finite Elemente Methode (FEM)*.

2.3.3 Modellierungssprachen

Mittels Modellierungssprachen versucht man Systeme und deren innere Strukturen auf einer höheren Abstraktionsebene graphisch darzustellen. Modellierungssprachen kommen besonders häufig in der Wirtschaftsinformatik und Softwareentwicklung zum Einsatz. Sie dienen vordergründlich dazu, die funktionalen Anforderungen zu spezifizieren und ein verständliches, anwendungsorientiertes Modell einer geplanten Lösung zu erstellen. Jede Modellierungssprache verfügt über ihre eigene Syntax, eine Art Grammatik. Je nach Darstellungsform kann man zwischen textuellen oder visuellen (graphischen) Sprachen unterscheiden. Zunehmend besteht hier ein Trend von den textuellen Systembeschreibungen zu visuellen Darstellungen des Systems. Im SE spricht man hier von *Modellbasiertem Systems Engineering (MBSE)*. (vgl. Engels, 2012)

Für die visuelle Darstellung von Systemen eignen sich Graphen oder Matrizen. Diese beiden Darstellungsformen lassen sich auch ineinander überführen. (vgl. Haberfellner, Weck, Fricke, & Vössner, 2012)

Gerade in der Wirtschaftsinformatik hat sich hier eine Vielzahl an speziellen Modellierungssprachen entwickelt. Einige der bekanntesten sind (vgl. Becker, 2012):

- Entity Relationship Model (ERM)
- Unified Modeling Language (UML)
- Ereignisgesteuerte Prozessketten (EPK)
- Business Process Model and Notation (BPMN)
- System Modeling Language (SysML)

Diese Modellierungssprachen unterscheiden sich vor allem durch ihre Syntax und durch ihr spezielles Anwendungsgebiet. Dabei bedienen sich UML oder SysML beispielsweise einer Vielzahl an unterschiedlichen Diagrammtypen (siehe Abbildung 3). Nachstehend soll ein kleiner Einblick in zwei dieser Modellierungssprachen gegeben werden.

Entity Relationship Model (ERM) – Bei dem ER-Modell handelt es sich um eine sehr einfache Modellierungssprache bestehend aus zwei Grundkonstrukten. Zum einen sind das die *Entitäten (entities)* als Objekte der realen Welt, zum anderen die *Beziehungen (relationships)* zwischen den Objekten. Die

Beziehung zwischen Entitäten wird mithilfe von so genannten *Kardinalitäten* näher charakterisiert. Objekte der realen Welt besitzen Eigenschaften, die im ER-Modell als *Attribute* spezifiziert werden können. Attribute, welche die Objekte eindeutig identifizieren lassen, bezeichnet man auch als *Schlüsselattribute* (z.B. Identifikationsnummern). Auch Beziehungen können Attribute besitzen, welche diese näher beschreiben. Der Vorteil dieser graphischen Darstellung ist, dass komplexe, konzeptionelle Zusammenhänge anhand einer leicht verständlichen Notation beschrieben werden. (vgl. Becker, 2012)

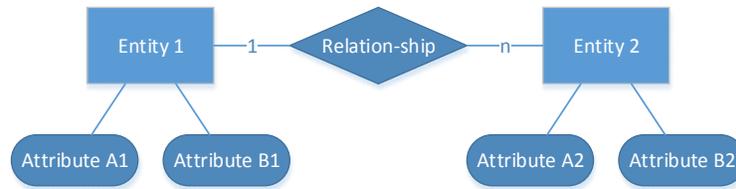


Abbildung 2: Beispiel für die Darstellung eines ER-Modells

Im Anbetracht der Definition eines Systems (siehe Kapitel 2.1.1) zeigt die Syntax eines ER-Modells eindeutig, dass es dazu dient, Systeme darzustellen.

System Modeling Language (SysML) – SysML kann als eine Abänderung der Modellierungssprache UML betrachtet werden. Dabei verwendet SysML einen Teil der Gestaltungsprinzipien von UML und erweitert diese um einige Aspekte. Mithilfe von SysML soll ermöglicht werden, Modelle unabhängig von ihrer Art (z.B. soziologisch, technisch etc.) zu entwerfen. SysML ist demnach eine *general purpose modeling language*, welche explizit auf die Disziplin des Systems Engineering zugeschnitten ist. (vgl. OMG, 2012)

„SysML supports the specification, analysis, design, verification and validation of a broad range of complex systems. These systems may include hardware, software, information, processes, personnel, and facilities.“ (OMG, 2012, S. 1)

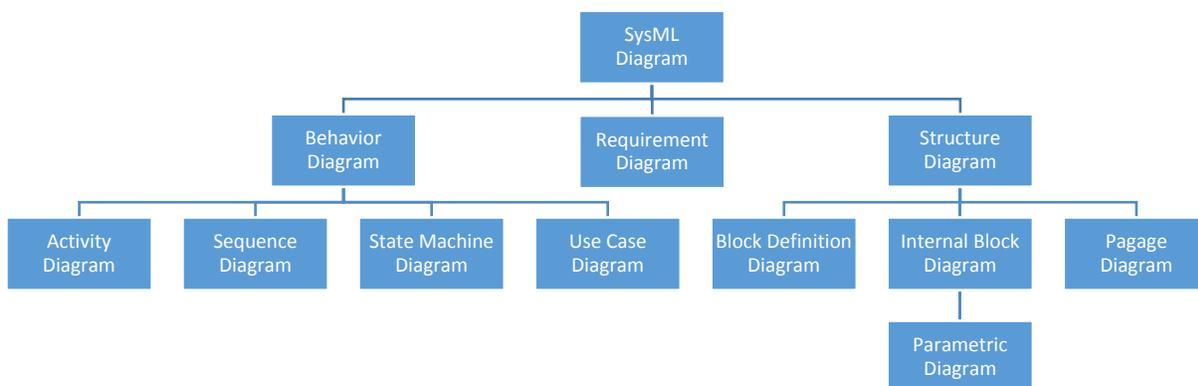


Abbildung 3: SysML Diagrammtypen (vgl. OMG, 2012, S. 167)

SysML soll helfen, den Weg von dokumentorientierter Systementwicklung hin zum Modellbasierten Systems Engineering zu ebnen. Bislang wurden für die verschiedenen unabhängigen Teilmodelle eines Systems unterschiedliche Modellierungssprachen verwendet. Woran es fehlte, war eine disziplinübergreifende Modellzusammenführung zur Integration der unterschiedlichen Modelle in einem Modell des Gesamtsystems. Diese Modellzusammenführung scheint gerade in Hinblick auf mechatronische Systeme und deren unterschiedliche Modelle diverser Fachbereiche besonders bedeutungsvoll.

3 Systems Engineering

Die Systeme der modernen Wirtschaft und Wissenschaft nehmen in ihrer Komplexität stetig zu. Der Zuwachs an Komplexität führt unweigerlich zu Problemen, welche bewältigt werden müssen. Dabei ist es dem Menschen ein Bedürfnis, diese schnell und fehlerfrei zu lösen. Die Entwicklung komplexer Produkte oder auch die Erforschung komplexer Systeme verlangt einen geordneten, systematischen Umgang mit Komplexität. Komplexe Aufgaben können meist nur von einer Gruppe von Menschen gelöst werden. Dies erfordert eine Verteilung der einzelnen Aufgaben auf verschiedenste Fachkräfte aus unterschiedlichen Fachgebieten, ohne dabei den Überblick über das Ganze zu verlieren. Ein solches Gliedern von komplexen Sachverhalten in kleinere Teilaspekte und die Vernetzung dieser Aspekte ist Gegenstand des *Systemdenkens*. (vgl. Winzer, 2013)

„Increased complexity of systems recently developed in the fields of communications, instruments, computation, and control has led to an emphasis on the field of systems engineering.“ (Schlager, 1956, S. 64)

In diesem Kapitel soll die interdisziplinäre Wissenschaft des Systems Engineering (SE) behandelt werden. Dabei stehen die folgenden Fragen im Zentrum der Interesses:

1. Was ist Systems Engineering?
2. Welche Aufgaben hat das Systems Engineering?
3. Wodurch unterscheiden sich Systems Engineering Konzepte?

Im ersten Abschnitt dieses Kapitels soll das Fachgebiet Systems Engineering vorgestellt werden. Hierbei wird den Fragen nachgegangen, was unter SE zu verstehen ist, wie es entstand und welche Definitionen sich finden lassen.

Im zweiten Abschnitt wird der Versuch unternommen, die Aufgabengebiete des Systems Engineering von anderen Fachbereichen abzugrenzen. Im Zuge dessen sollen Ansätze gefunden werden, um die Unterschiede zwischen Systems Engineering und den verwandten Disziplinen des Systems Architecting bzw. dem Systems Design hervorzuheben.

Im letzten Abschnitt werden ausgewählte SE-Konzepte oder Vorgehensmodelle präsentiert. Ziel aller Systems Engineering Ansätze ist es, Probleme zu lösen oder Produkte zu entwickeln. Im Allgemeinen unterscheiden sie sich vorwiegend in ihrer Herangehensweise. Eine Aussage darüber, welche Vorgehensweise zur Problemlösung die „bessere“ ist, kann vermutlich nicht getroffen werden und ist auch nicht Thema dieser Arbeit.

3.1 Grundlagen

Der Begriff *System* wurde bereits geklärt (siehe Kapitel 2.1.1), doch es bedarf noch der Konkretisierung des Begriffes *Engineering*. Diese getrennte Begriffsklärung ermöglicht es, einen ersten Eindruck vom Fachbereich Systems Engineering zu gewinnen.

System – Ein System ist nach den gezeigten Definitionen ein (von der Umwelt) abgegrenztes Ganzes, bestehend aus mit Eigenschaften behafteten Elementen und Beziehungen, das einen gewissen Zweck erfüllt.

Engineering – Das englische Wort *Engineering* bedeutet im Deutschen *Ingenieurwissenschaften* aber es kann auch als *Konstruktion* oder *Entwicklung* verstanden werden. (vgl. PONS, 2014)

Was Engineering jedoch genau bedeutet, sollen die folgenden Definitionen zeigen.

„The field has been defined by the Engineers Council for Professional Development, in the States, as the creative application of ‘scientific principles to design or develop structures, machines, apparatus, or manufacturing processes, or works utilizing them singly or in combination; or to construct or operate the same with full cognizance of their design; or to forecast their behaviour under specific operating conditions; all as respects an intended function, economics of operation and safety to life and property.’“ (Smith, 2013)

„Engineering is the application of science to develop, design, and produce logical and/or physical objects such as buildings, machines, or a computer program to fulfill a desired need or to achieve an objective.“ (SMC, 2005, S. 13)

Nachstehend soll ein kurzer und äußerst gewagter Ansatz für eine freie Interpretation des Systems Engineering vorgestellt werden.

Engineering kann als die Anwendung von wissenschaftlichen Praktiken zum Entwurf, zur Konstruktion und/oder zum Bau von Maschinen, Geräten, Prozessen oder, verallgemeinert, von Systemen gesehen werden. Sehr vereinfacht gesagt, bedeutet dies den Entwurf von Systemen (engl. *design of systems*).

Ersetzt man nun das Wort *Engineering* durch die eben erwähnte Begriffsauslegung, erhält *Systems Engineering* weitestgehend die Bedeutung von *systems design of systems*. Durch die so entstandene Redundanz des Wortes *System* kann darauf geschlossen werden, dass sich das wissenschaftliche Gebiet des Systems Engineering damit auseinandersetzt, Systeme auch als solche zu betrachten und diese dementsprechend zu entwickeln. Frei interpretiert, ist Systems Engineering also ein *systematischer Entwurf von Systemen*.

Im Folgenden wird erläutert, was Systems Engineering genau umfasst.

3.1.1 Begriff und Definition

Systems Engineering als eigenständige Disziplin hat seine Wurzeln in der Kommunikationstechnik. So fand der vermutlich erste Einsatz von Systems Engineering in den *Bell Telephone Laboratories* in den 1940er Jahren statt. (vgl. Schlager, 1956)

Danach hat Systems Engineering auch Einzug in die US-amerikanische militärische Geschichte sowie in die Entwicklungen der Luft- und Raumfahrt gehalten. (vgl. INCOSE, 2006).

Die Ansätze des SE wurden von verschiedenen Institutionen, wie z.B. der *National Aeronautics and Space Administration (NASA)* oder dem US-amerikanischen *Department of Defense (DoD)*, weiterentwickelt.

In den 1990ern wurde die *International Council on Systems Engineering (INCOSE)* als eine Non-Profit-Organisation mit dem Ziel gegründet, die Idee und die Vorzüge des Systems Engineering weltweit zu verbreiten und es damit zu verbessern. (vgl. INCOSE, 2013)

Unter anderem bedingt durch die unterschiedlichen Institutionen, die die Entwicklung des Systems Engineering verantrieben haben, ergaben sich verschiedene Auffassungen in Hinblick auf die Bedeutung dieser Disziplin. Folgende Beschreibungen des SE sind in diversen Handbüchern zu finden:

„In summary, systems engineering is an interdisciplinary engineering management process that evolves and verifies an integrated, life-cycle balanced set of system solutions that satisfy customer needs.“ (DOD, 2001, S. 3)

„Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.“ (INCOSE, 2010, S. 7)

„Systems engineering is a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system. [...] It is a way of looking at the ‘big picture’ when making technical decisions. It is a way of achieving stakeholder functional, physical, and operational performance requirements in the intended use environment over the planned life of the systems. In other words, systems engineering is a logical way of thinking.“ (NASA, 2007, S. 3)

Fasst man die wesentlichen Aspekte obiger Definitionen zusammen, lässt sich Folgendes schließen:

Bei Systems Engineering handelt es sich um eine fachübergreifende Management-Disziplin, deren Aufgabe es ist, Probleme durch methodisches Vorgehen möglichst effizient zu lösen oder bestimmten Bedürfnissen nachzukommen. Dabei sollen Vorgehensprinzipien und Denkmodelle helfen, die Komplexität von Systemen zu reduzieren und den Überblick über das Ganze zu bewahren. Ziel ist eine erfolgreiche Entwicklung von Produkten oder die Lösung von Problemen, die sich durch Qualität auszeichnet und die Bedürfnisse oder gestellten Anforderungen bestmöglich erfüllt.

Es ist Aufgabe des Systems Engineering, Risiken und Kosten zu reduzieren, indem Konzeptentscheidungen nicht zu früh bzw. ohne ausreichende Informationen und Analysen getroffen werden. Die zunehmende Komplexität der Produkte hat direkte Auswirkung auf den Lebenszyklus, indem es zu längeren Entwicklungszeiten führt. SE soll diesem Dilemma entgegenwirken und damit die Wettbewerbsfähigkeit sicherstellen. (vgl. INCOSE, 2010)

3.1.2 Systemdenken

Wie für alle auf der Systemtheorie (siehe Kapitel 2.1) basierenden Disziplinen spielt *Systemdenken* auch im SE eine zentrale Rolle. Laut *Winzer* ist von Systems Engineering dann die Sprache, wenn das Systemdenken auf technische Systeme übertragen und mit einem Gestaltungsprozess verbunden werden kann. Grundlegende Philosophie aller SE-Ansätze ist das Systemdenken, nach dessen Auffassung alles (der Mensch sowie seine Umgebung) Systemcharakter aufweist. Dies ist die Voraussetzung, um einen Problemlösungsprozess zu strukturieren. (vgl. Winzer, 2013)

„Im Systemdenken oder systembezogenen Denken geht es darum, den zu lösenden Problemkreis als ein System mit seinen Elementen und Interaktionen sowohl innerhalb des Systems als auch mit den Umsystemen, d.h. der Außenwelt des Systems zu verstehen.“ (Schönsleben, 2007, S. 942)

Systemdenken wird als fixer Bestandteil des Systems Engineering verstanden und soll helfen, komplexe Systeme besser zu verstehen und diese zu gestalten.

„Systems thinking occurs through discovery, learning, diagnosis, and dialog that lead to sensing, modeling, and talking about the real-world to better understand, define, and work with systems.“ (INCOSE, 2010, S. 7)

Dieser Denkansatz ermöglicht es, komplexe Sachverhalte derart zu gliedern und auf einzelne Fachdisziplinen zu verteilen, dass ein transdisziplinäres Arbeiten gewährleistet wird. Wonach es jedoch bedarf, ist eine interdisziplinäre Systembeschreibungssprache zur Abbildung des Systems als Basis für eine gemeinsame Systemgestaltung. Erst wenn es gelingt, das Gesamtsystem in einer gemeinsamen Sprache

abzubilden, können die jeweiligen Fachkräfte gemeinsam ein Metamodell (siehe Kapitel 2.3.1) gestalten. Damit wird Nachvollziehbarkeit und Transparenz für die beteiligten Teams sichergestellt. (vgl. Winzer, 2013)

„Wenn es gelingt, dass sich interdisziplinäre Teams gemeinsam des Systemdenkens bedienen und von dem betrachteten System, zum Beispiel der logistischen Anlage, ein gemeinsames Denkmodell (Metamodell) entwickeln, das von jedem Fachspezialisten dieses Teams verstanden wird, dann ist somit die Basis für einen transdisziplinären, transparenten Problemlösungsprozess gegeben.“ (Winzer, 2013)

Dadurch wird die Bedeutung der Modellbildung im Systems Engineering deutlich. Mithilfe einheitlicher Modellierungssprachen (siehe Kapitel 2.3.3) gelingt es, insbesondere in fachübergreifenden Disziplinen wie der Mechatronik, an komplexen Modellen zu arbeiten.

3.1.3 Problem, Methode und Lösung

Wird Systems Engineering außerhalb des unternehmerischen Kontexts betrachtet, handelt es sich um eine Disziplin zur Problemlösung (vgl. Kapitel 3.1.1). SE soll Wegbereiter zur Bewältigung komplexer Probleme sein und dabei Methoden und Werkzeuge liefern, die eine Lösungsfindung unterstützen, um möglichst zufriedenstellende Resultate zu erzielen.

Problem – Ein Problem gilt im allgemeinen Sprachgebrauch als „schwierige, ungelöste Aufgabe“ oder „schwer zu beantwortende Frage“. (vgl. DUDEN, 2014c)

Diese Definitionen sind jedoch mit Vorsicht zu genießen. Ein Problem muss nicht notwendigerweise ungelöst oder schwierig sein. Für Probleme können bereits mehrere verschiedene Lösungen existieren, wovon lediglich eine angewandt werden muss. Ebenso ist die Schwierigkeit zur Lösung eines Problems vordergründig von der Person, die das Problem zu lösen gedenkt, deren Fähigkeiten und Erfahrungen abhängig.

Im Systems Engineering versteht man unter dem Begriff *Problem* – weit umfassender – die Differenz zwischen dem Ist-Zustand und der Vorstellung vom Soll-Zustand. (vgl. Haberfellner, Weck, Fricke, & Vössner, 2012)

Methode – Der Begriff *Methode* stammt vom altgriechischen Wort *méthodos* ab, was so viel bedeutet wie „Weg einer Untersuchung“, worunter eine „Art und Weise eines Vorgehens“ verstanden wird. (vgl. DUDEN, 2014d)

Im SE ist Methodik das zentrale Schlüsselement zwischen Problem und Lösung. Methodik selbst löst keine Probleme; methodisches Vorgehen kann weder Erfahrung noch Fachwissen ersetzen, sondern soll vielmehr als Werkzeug dienen, um komplexe Problemstellungen effizient zu lösen. Methoden sind (systematische, strukturierte, übersichtliche...) Vorgaben für auszuführende Tätigkeiten zur Erreichung eines vorgegebenen Ziels. Sie sind Hilfsmittel oder Werkzeug zur Lösungsfindung, garantieren jedoch noch keine Lösung. Die Methode kann sich jedoch positiv auf die Qualität einer Lösung und auf die Effizienz der Lösungssuche auswirken. (vgl. Haberfellner, Weck, Fricke, & Vössner, 2012)

„Methoden bieten Vorschläge für die Abfolge bestimmter Tätigkeiten an und die Art und Weise, in der diese Tätigkeiten durchzuführen sind.“ (Lindemann, 2009, S. 56)

Lösung – Lösen einer Aufgabe bedeutet das Bewältigen dieser. (vgl. DUDEN, 2014)

Die Lösung ist Ergebnis der Handlungen, der Überlegungen sowie der Anwendung von Kenntnissen, Fähigkeiten und Fertigkeiten.

Systems Engineering ist aber mehr als lediglich ein Problemlösungsprozess. SE liefert Ansätze, die Ordnung und Struktur in Projekte bringen und stellt für Unternehmen eine Grundlage zur Produktentwicklung dar. Umfangreiche Entwicklungen, an denen eine Vielzahl von Personen aus unterschiedlichsten Fachgebieten (z.B. Mechanik, Elektronik, Informatik etc.) und den verschiedensten Bereichen (z.B. Entwicklung, Produktion etc.) beteiligt ist, bedürfen einen hohen Maßes an Organisation und Koordination. Um dies zu ermöglichen, muss der gesamte Entwicklungsprozess in überschaubare Abschnitte unterteilt werden. Hilfsmittel, die hierbei zur Anwendung kommen, sind zum Beispiel Vorgehensmodelle.

3.1.4 Vorgehensmodell

Ein Vorgehensmodell ist eine strukturierte Arbeitsanleitung zur Durchführung eines Projektes, in der festgelegt wird, in welcher Reihenfolge Tätigkeiten auszuführen sind. Aufgabe des Vorgehensmodells ist es, einen Gestaltungs- oder Problemlösungsprozess und die damit verbundenen Aktivitäten in eine logische Abfolge zu bringen.

„Vorgehensmodelle stellen ein Hilfsmittel dar, das den Entwickler bei der Planung zukünftiger Prozesse, der Orientierung innerhalb aktueller Prozesse und der Reflexion abgeschlossener Prozesse unterstützt.“ (Lindemann, 2009, S. 33)

„Während das Vorgehensmodell angibt, ‚WAS‘ zu tun ist (welche Schritte durchzuführen sind), beschreibt die Methode, ‚WIE‘ etwas zu tun ist (auf welche Art und Weise und mit welchem Ergebnis die Schritte durchzuführen sind).“ (Lindemann, 2009, S. 58)

Im Vergleich verschiedener SE-Vorgehensmodelle stellt sich heraus, dass diese aus drei zentralen Elementen bestehen. Zum Ersten sind das Handlungsprinzipien, welche während des gesamten Projekts eingehalten werden sollten, zum Zweiten handelt es sich um die Gestaltung des Projekts in Projektphasen, die den Projektfortschritt gliedern und zum Dritten spricht man von Problemlösungszyklen, die dabei helfen, die Problemstellung der jeweiligen Phasen strukturiert zu lösen. Häufig wird das Projektphasenmodell oder der Problemlösungszyklus bereits als eigenes Vorgehensmodell bezeichnet.

Projektphasen – Das Phasenmodell wird oftmals als *Makrologik* oder *Makroebene* bezeichnet. Dahinter steckt die Idee, das gesamte Projekt in überschaubare Etappen (Projektphasen) zu gliedern. Jede dieser Phasen ist logisch aber auch zeitlich von der anderen abgrenzbar und durch einen definierten Input und Output gekennzeichnet. Prinzipiell handelt es sich dabei um eine Anwendung der Prinzipien *vom Groben ins Detail* und *vom Abstrakten zum Konkreten* auf das Projekt (siehe Tabelle 1). Eine Aufteilung des Projekts in Etappen soll einen stufenweisen Planungs-, Entscheidungs- und Konkretisierungsprozess ermöglichen. Durch dieses Vorgehen wird Struktur in den zeitlichen und inhaltlichen Ablauf eines Projektes gebracht. Die am Ende jeder Phase stehenden Ergebnisse dienen als Bemessungsgrundlage für ein weiteres Vorgehen und verlangen somit das Hinzuziehen der Entscheidungsinstanz. Dies gewährleistet einen laufenden Austausch zwischen Auftragnehmer (Projektteam) und Auftraggeber. Der Übergang zwischen den einzelnen Projektphasen bietet Raum für Änderungen und Verbesserungen oder auch für die Möglichkeit eines Widerrufs, der seinerseits mit einem Projektabbruch enden kann. Mit zunehmendem Projektfortschritt reduziert sich dieser Spielraum für Korrekturen. Jedes Projekt kann sich, je nach dessen Art und Größe, im Umfang und Inhalt seiner einzelnen Projektphasen unterscheiden.

Problemlösungszyklus – Der Problemlösungszyklus wird auch oft als *Mikrologik* oder *Mikroebene* bezeichnet. Er soll den Entwickler bei der Problembewältigung unterstützen und lässt sich in jeder Projektphase anwenden. Dabei muss es sich nicht notwendigerweise um einen Zyklus handeln (vgl. Münchner-Modell in Kapitel 3.3.1). Ähnlich dem Phasenmodell kann auch der Problemlösungszyklus als eine Abfolge logisch abgrenzbarer Schritte verstanden werden. Im Problemlösungszyklus begibt man sich durch schrittweise Analyse und Synthese des Problems auf eine Lösungssuche. Die Verwendung des Begriffs *Zyklus* unterstreicht, dass es sich in der Realität um einen iterativen Vorgang handelt.

Handlungsprinzipien – Die Grundprinzipien des Handelns stehen wohl an oberster Stelle im Vorgehensmodell. Sie sollen während des gesamten Entwicklungsprozesses eingehalten werden oder zumindest so oft wie möglich Anwendung finden. Die Handlungsprinzipien für die Produktentwicklung oder für die Lösung von Problemen sollten im Sinne des Systems Engineering eingehalten werden,

unabhängig davon, welche Art von Problem zu Grunde liegt. Es handelt sich dabei um allgemeine Strategien, welche Einfluss auf die Vorgehensweise nehmen und verschiedene Methoden prägen. Diese auf Erfahrungen basierenden Prinzipien sollen bei der technischen Problemlösung helfen. Ebenso wie Methoden können sie selbst keine Probleme lösen, erhöhen jedoch die Wahrscheinlichkeit für eine erfolgreiche Produktentwicklung. Systemdenken kann als eines dieser Grundprinzipien des Handelns gesehen werden. (vgl. Lindemann, 2009)

Lindemann und *Winzer* liefern eine Auswahl einiger Prinzipien aus unterschiedlichen SE-Konzepten (siehe Tabelle 1).

Tabelle 1: Grundprinzipien des Handelns (vgl. Lindemann, 2009, S. 55 ff) (vgl. Winzer, 2013, S. 15 ff)

Grundprinzip ...	Erklärung
Systemdenken	Die wohl zentralste Denkweise im SE, die helfen soll komplexe Erscheinungen zu verstehen und zu gestalten, nennt sich Systemdenken (siehe Kapitel 3.1.2).
Vom Groben ins Detail	Hierbei wird das Betrachtungsfeld zunächst erweitert und erst nach und nach eingeschränkt.
Vom Abstrakten zum Konkreten	Die Entwicklung soll sich in Konkretisierungsstufen (Abstraktionsebenen) gliedern. Hierbei arbeitet man sich stufenweise von einem abstrakten Konstrukt zu einer konkreten Lösung vor. Wechsel zwischen den Stufen (vor- und rückwärts) sind möglich. Dieses Vorgehen soll verhindern, sich schon zu Beginn in den Details zu verlieren.
Wiederkehrende Reflexion	Zwischen einzelnen Aktivitäten soll über das bislang Erreichte oder nicht Erreichte kritisch reflektiert werden. Ergebnisse einer Aktivität sollen mit der Zielsetzung verglichen werden, bevor die nächste Aktivität gesetzt wird.
Problemzerlegung	Problemzerlegung ist eine Möglichkeit mit Komplexität umzugehen, indem Probleme in Teilprobleme gegliedert werden. Teilprobleme sind weniger komplex, lassen sich besser überschauen und sind somit leichter zu lösen.
Strukturierung	Mithilfe von Hierarchien bzw. der Bildung von Gruppen, Clustern oder Modulen lässt sich Komplexität reduzieren.
Minimale Modelle	Nach diesem Grundprinzip sollen, unter Zuhilfenahme von einfachen, allgemein verständlichen Modellen, Lösungen gesucht werden. Dabei konzentriert man sich auf wenige, aber wichtige Modellelemente. D.h., das Modell wird so minimalistisch wie möglich gehalten.
Verständlichkeit	Konzepte und Modelle sollen konsistent und jederzeit überprüfbar sein. Dies erfordert Vollständigkeit und die Freiheit von Redundanz. Eindeutigkeit, Verständlichkeit und Nachvollziehbarkeit stehen im Mittelpunkt der Konzept- und Modellgestaltung.
Anwendung mehrerer Sichten	Systeme sollen anhand verschiedener Gesichtspunkte (Aspekte) betrachtet werden. Dies führt zu einem besseren Problemverständnis.
Neutralität	Dieses Prinzip besagt, dass zuerst festzulegen ist, „WAS“ gemacht werden soll, bevor Überlegungen über das „WIE“ getroffen werden.
Mehrfachverwendbarkeit	Mehrfachverwendbarkeit bedeutet, dass komplexe Systeme so in Module zerlegt werden, dass einzelne Module im selben System wiederholt Verwendung finden.
Standardisierung	Bei der Standardisierung werden, im Gegensatz zur Mehrfachverwendbarkeit, komplexe Systeme in wiederverwendbare Teile (für neue Projekte) zerlegt.
Informationskapselung	Das Prinzip hat die Austauschbarkeit einzelner Komponenten zum Ziel.
Diskursives Vorgehen	Bei diesem Vorgehen ist jeder Handlung eine Zielsetzung vorangestellt. Jedes Ergebnis einer Handlung kann zu einer neuen Zielsetzung führen.
Denken in Alternativen	Bei der Entwicklung von Lösungen wird in Alternativen (Varianten) gedacht. Für jedes Problem gibt es verschiedene Lösungsmöglichkeiten, weshalb man sich nicht mit der erstbesten zufrieden geben sollte.
Modalitätenwechsel	Dieses Prinzip hilft, wenn die Routine ein Problem darstellt. In solchen Fällen gilt es, den Betrachtungsgegenstand oder die Perspektive innerhalb bestimmter Kategorien zu wechseln (z.B. Gesamtsystem zu Detailsystem, Synthese zu Analyse oder <i>Top-Down</i> zu <i>Bottom-Up</i>).
Minimierung von Schnittstellen	Systeme sollen so abgegrenzt werden, dass die entstehenden Schnittstellen innerhalb und außerhalb des Systems minimal sind.

3.2 Abgrenzung

Eine der vermutlich am häufigsten gestellten Fragen in Bezug auf SE lautet: „Wo liegt eigentlich der Unterschied zwischen einem Systems Engineer und einem Systems Architect?“ oder „Wodurch unterscheidet sich Systems Engineering von Systems Architecting bzw. von Systems Design?“. Zudem ist die Aussage: „Das ist nicht Aufgabe des Systems Engineering!“ in einschlägigen Foren häufig zu lesen. Aufgrund unterschiedlicher Ansichten zu SE lassen sich die gestellten Fragen nicht eindeutig beantworten. Dennoch soll im Folgenden kurz auf diese Problematik eingegangen werden.

3.2.1 Aufgaben des Systems Engineering

SE stellt sicher, dass ein System den technischen Bedürfnissen nachkommt und den richtigen Entwicklungsansatz verfolgt. Mithilfe von Systems Engineering soll die Übersicht über die einzelnen Aktivitäten des technischen Entwicklungsteams bewahrt werden, um Aufgaben kommunizieren, koordinieren und überwachen zu können. Der Systems Engineer prüft und evaluiert die technischen Aspekte des Projekts und gewährleistet, dass die Entwicklungsprozesse des Systems und dessen Subsysteme entsprechend funktionieren. Durch Anwendung von SE wird ein System von der Idee bis zum fertigen Produkt entwickelt, wobei das gesamte technische Team in den SE-Prozess integriert ist. (vgl. NASA, 2007)

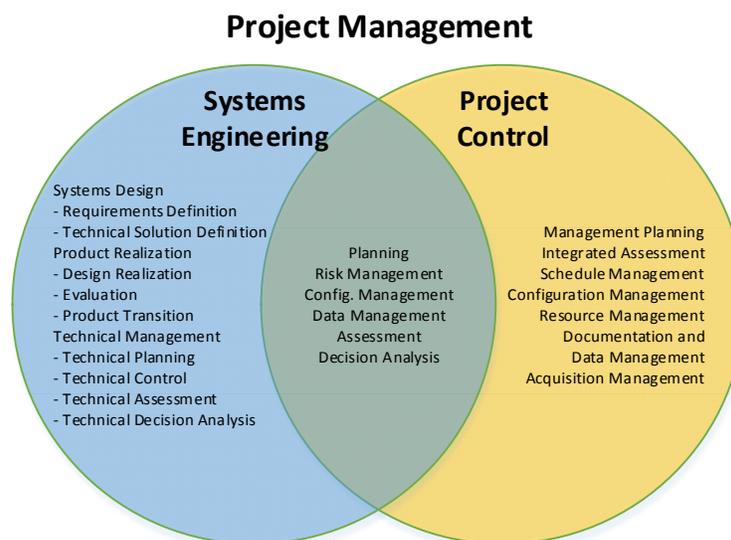


Abbildung 4: NASA - SE in context of overall project management (NASA, 2007, S. 4)

In der Fachliteratur wird in diesem Zusammenhang meist eine Unterscheidung zwischen Systems Engineering und Projektmanagement bzw. Projektcontrolling getroffen (vgl. Abbildung 4). Weder dieser

Ansatz noch die angeführten Aufgaben des SE können die eingangs gestellte Frage beantworten. Aus diesem Grund wird hier eine Herangehensweise verfolgt, bei der zunächst die allgemeinen Begriffe *engineering*, *architecting* und *designing* differenziert werden.

Was unter *engineering* verstanden wird, war bereits in Kapitel 3 Thema. Daher werden in diesem Abschnitt vor allem die Begriffe *architecture* und *design* näher behandelt.

Engineering – Die Ingenieurwissenschaft ist nach Definition die Anwendung der Wissenschaft, um Systeme (z.B. Gebäude, Maschinen, Computer-Programme) zu entwickeln, zu gestalten und zu produzieren, welche die gewünschten Anforderungen erfüllen oder die gesetzten Ziele erreichen. (vgl. SMC, 2005)

Architecture – Üblicherweise versteht man unter Architektur die Baukunst. (vgl. DUDEN, 2014f)

Architecting beschäftigt sich mit dem Entwurf der Architektur. Im Falle von Systemen wäre das die Systemarchitektur. Das *Institute of Electrical and Electronics Engineers (IEEE)* definiert *architecture* als:

„[...] the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.“
(ISO/IEC/IEEE 42010:2011)

Architektur bezeichnet demnach das, was mithilfe der Systemmodelle (siehe Kapitel 2.3.1) dargestellt wird und beschreibt, wie ein System im Inneren organisiert oder aufgebaut ist (vgl. Kapitel 2.2.3, Systemstruktur).

„The System Architecture describes the entire system. It includes the physical architecture produced through design synthesis and adds the enabling products and services required for life cycle employment, support, and management.“ (DOD, 2001, S. 73)

Spricht man von Systemarchitektur, ist anzumerken, dass es stets vom Kontext abhängt, was darunter verstanden wird. Verschiedene Auffassungen von Systemarchitektur finden sich z.B. in den Bereichen der Entwicklung von Hard- oder Software. Dennoch handelt es sich dabei meist um Darstellungen von Systemelementen und deren Strukturen oder Beziehungen. (vgl. SMC, 2005)

Design – Design wird im allgemeinen Sprachgebrauch als das optische Erscheinungsbild verstanden. Das englische Wort *design* steht jedoch für einen (konzeptionellen) Entwurf oder die Gestaltung eines Objekts. Aus technischer Sicht kann sich *design* auch auf die Konstruktion oder die Bauart eines Gegenstands beziehen. (vgl. PONS, 2014a)

Analog zu *architecting* beschäftigt sich *designing* mit dem Entwurf eines Designs. Handelt es sich hingegen um Systeme, spricht man von Systemdesign. Das IEEE bezeichnet *designing* als:

„[...] the activities of defining, documenting, maintaining, improving, and certifying proper implementation of an architecture.“ (ISO/IEC/IEEE 42010:2011)

Vollständig beantwortet sind damit die eingangs gestellten Fragen jedoch noch nicht. Aus diesem Grund wird im folgenden Ansatz versucht, die Disziplinen anhand der Rollenverteilungen im SE fest zu machen. Die dabei gestellte Frage lautet: „Wo liegt der Unterschied zwischen einem System Engineer einem Systems Architect und einem Systems Designer?“.

3.2.2 Rollenverteilung im Systems Engineering

Bislang stand die Bezeichnung *Systems Engineer* als ausführende Rolle sämtlicher Aufgabenbereiche des SE. Im Lebenszyklus (engl. *lifecycle*) eines Systems lassen sich verschiedene Tätigkeiten und damit verbundene Rollenbezeichnungen identifizieren. *Sheard* unterscheidet in diesem Zusammenhang die folgenden zwölf Rollen (Sheard, 1996):

- | | |
|--|------------------------------------|
| 1. <i>Requirements Owner (RO)</i> | 7. <i>Customer Interface (CI)</i> |
| 2. <i>System Designer (SD)</i> | 8. <i>Technical Manager (TM)</i> |
| 3. <i>System Analyst (SA)</i> | 9. <i>Information Manager (IM)</i> |
| 4. <i>Validation/Verification Engr. (VV)</i> | 10. <i>Process Engineer (PE)</i> |
| 5. <i>Logistics/Ops Engineer (LO)</i> | 11. <i>Coordinator (CO)</i> |
| 6. <i>Glue Among Subsystems (G)</i> | 12. <i>Classified Ads SE (CA)</i> |

Die oben genannten Rollen stehen in Beziehung zu jenen aus dem Bereich des Projektmanagements. Diese Aufgabenverteilungen sind sehr spezifisch, weshalb hier eine noch einfachere und vor allem relativ allgemeine Rollenzuschreibung gewählt wird.

Muller unterscheidet fünf Rollen im SE, welche universeller Natur sind. Dabei erscheint es sinnvoll, diese Einteilung um die Funktion des System Analytikers (engl. *systems analyst*), welche aus der Rollenverteilung von *Sheard* entnommen wurde, zu erweitern. Damit wird eine zusätzliche Unterscheidung vom System Tester ermöglicht.

Wie aus Kapitel 2.3 ersichtlich, besteht ein wesentlicher Unterschied zwischen dem Experiment und der Simulation (siehe Abbildung 1). Dies lässt sich durch die Einführung des System Analytikers somit auf die Rollenverteilung übertragen.

Damit kann man die folgenden sechs allgemeinen Rollen im Systems Engineering unterscheiden (vgl. Muller, 2013, S. 48 ff):

- | | |
|-----------------------------|----------------------------|
| 1. <i>Systems Manager</i> | 4. <i>Systems Engineer</i> |
| 2. <i>Systems Architect</i> | 5. <i>Systems Analyst</i> |
| 3. <i>Systems Designer</i> | 6. <i>Systems Tester</i> |

Systems Manager – Management bedeutet stets eine leitende oder führende Aufgabe bzw. Rolle zu übernehmen. Diese umfasst Tätigkeiten der Planung und Entscheidung sowie die Erteilung von Anweisungen. (vgl. DUDEN, 2014e) Manager übernehmen Verantwortung, was wiederum entsprechende Kompetenzen erfordert. Die Rolle des System Managers ist damit im gesamten Verlauf des Projektes bedeutsam.

„Systems manager is the overall responsible for all systems aspects, ranging from strategically positioning in the portfolio and the time to final operational performance in the field. [...] An alternate term for this role can be program manager.“ (Muller, 2013, S. 48-49)

Systems Architect – Architektur ist charakterisiert durch den Umgang mit unstrukturierten Situationen, in denen weder Mittel noch Zweck bekannt zu sein scheinen. Oft wissen Kunden selbst nicht, was sie benötigen, sondern sie kennen bloß das Problem und dieses oft nur in unzureichendem Maße. Darum wenden sie sich in solchen Situationen an Architekten, deren Aufgabe es ist, den Kunden zu unterstützen und zu beraten, um dessen unklare Zielvorstellungen zu konkretisieren. Der Architekt erforscht gemeinsam mit dem Kunden die Anforderungen an das Design. Im Gegensatz dazu steht der klassische Ingenieursansatz, der die Suche nach einer optimalen Lösung für klar definierte Problemstellungen vorsieht. Schwerpunkt der Architektur ist der künstlerische, heuristische Aspekt. (vgl. Maier & Rehtin, 2000)

Für Architekten scheint somit die technische Umsetzbarkeit wenig im Vordergrund zu stehen. Die Umsetzung eines Gedankenkonstrukts ist die anschließende Aufgabe der Ingenieure. Diese versuchen anhand der Vorgaben technische Lösungen zu finden, um Ideen auch in die Tat umzusetzen. Der Tätigkeitsbereich des Systemarchitekten ist demnach vor allem am Beginn des Projekts angesiedelt.

„Systems architect who combines understanding of the context with in depth understanding of the solution to create an appropriate system. Note that the architect role combines some perceptive and creative modes of operation with more analytical modes. This mixture limits how far the architect can go in the real engineering.“ (Muller, 2013, S. 49)

Systems Designer – Systemdesigner gestalten *high level*-Systemarchitekturen, wobei sie die wichtigsten Komponenten auswählen. Die Gestaltungsvarianten für die Systeme und die dafür benötigten Bestandteile werden verglichen und deren Eignung zur Erfüllung der Systemanforderungen festgestellt. Dazu kommt die Auswahl gewisser Teilsysteme und deren Beschreibung in nächst niedrigerer Detaillierungsebene. (vgl. Sheard, 1996)

Im Vergleich zu Systemarchitekten beschäftigen sich Systemdesigner mit den inneren Zusammenhängen und Strukturen eines Systems, also mit einem höheren Detaillierungsgrad des Systems. Die Arbeit der Systemdesigner beginnt, wenn die Systemarchitektur festliegt.

„Systems designers take the product specification as starting point and work on (potential) solutions. System designers are ‘inward’ focused, where system architects connect the outward and inward perspectives.“ (Muller, 2013, S. 49)

Systems Engineer – Die Arbeit der Systemingenieure bezieht sich vor allem auf die allgemeinen Ingenieurstätigkeiten. Der Schwerpunkt liegt weniger auf planenden, sondern auf ausführenden Aktivitäten. Im Gegensatz zum Systemarchitekten beschäftigt sich der Ingenieur mit der technischen Umsetzbarkeit. Dies bedeutet, dass er, verglichen mit dem Systemarchitekten, über höhere Entscheidungskompetenzen verfügt. Der Systemingenieur entscheidet, ob die gewählte Systemarchitektur tatsächlich realisierbar ist. Das Kerngebiet des Systemingenieurs umfasst die Entwicklung des Detailkonzepts und die Systemrealisierung.

„The systems engineer will usually play the key role in leading the development of the system architecture, defining and allocating requirements, evaluating design tradeoffs, balancing technical risk between systems, defining and assessing interfaces, providing oversight of verification and validation activities, as well as many other tasks.“ (NASA, 2007, S. 3)

„Systems engineer is very close to the systems architect, but the emphasis shifts from perceptive and creative more to engineering. With engineering we mean the capability to finalize and document all details required for the later processes such as logistics, manufacturing, sales, and customer support.“ (Muller, 2013, S. 49)

In Bezug auf den Ablauf folgen die Aufgaben des Systemingenieurs jenen des Systemarchitekten und des Systemdesigners.

Systems Analyst – Aufgabe der Systemanalytiker ist es, herauszufinden, ob das zu entwickelnde System den gestellten Anforderungen entspricht. Je komplexer das zu analysierende System, desto mehr

Teilmodelle müssen geschaffen werden, um feststellen zu können, inwiefern das entwickelte System in seiner zukünftigen Umgebung wunschgemäß funktioniert. Dies hat zur Folge, dass mit der Systemkomplexität auch der Bedarf an Systemanalytikern steigt. (vgl. Sheard, 1996)

Systemanalytiker gestalten und entwerfen Modelle. Die Modellbildung und Simulation ist demnach ihr Tätigkeitsfeld. Systemmodelle werden sowohl vor als auch während des gesamten Produktentwicklungs- oder Problemlösungsprozesses erstellt und sollen den Entwicklungsablauf unterstützen und vorantreiben. Im Vergleich zu Systemtestern überprüfen Analytiker das System, noch bevor es als fertiges Produkt zur Verfügung steht.

Systems Testers – Aufgabe der Systemtester ist das Planen und Realisieren der Systemverifikation, um sicherzustellen, dass die Systeme anforderungsgerecht konstruiert und gebaut wurden. Dazu zählen Tätigkeiten des Entwurfs von detaillierten Testplänen und -prozeduren. Auch die Testingenieure sollen bereits im Laufe des gesamten Entwicklungsprozesses Fragen der System Entwickler beantworten, um diese in ihrer Arbeit zu unterstützen. Testingenieure werden zur Feststellung von Anomalien benötigt und tragen damit zu einem besseren Verständnis des Systemdesigns bei. (vgl. Sheard, 1996)

Die Arbeit der Systemtester beginnt, wenn Systeme oder zumindest Teilsysteme fertig gebaut sind.

„System testers verify that the solution performs as specified. In practice system testers need also trouble shooting capabilities to diagnose the cause of lacking performance.“ (Muller, 2013, S. 49)

3.3 Ausgewählte Modelle

Wie bereits erwähnt, handelt es sich beim Systems Engineering um eine fachübergreifende Management-Disziplin. Dabei stellt SE keine Vorschrift dar, sondern ist Hilfsmittel, um Probleme effizient lösen zu können. Um technische Produkte zu entwickeln, hat sich eine Vielzahl unterschiedlicher Vorgehensmodelle etabliert.

Im Bereich der mechatronischen Produkte aber auch bei anderen fachübergreifenden Entwicklungen stellt die VDI Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme“ wohl eines der bekanntesten Konzepte dar. In anderen Fachbereichen, wie zum Beispiel der Softwareentwicklung, lassen sich oft das V-Modell oder das V-Modell XT finden.

In diesem Abschnitt werden drei im Bereich der Mechatronik seltener anzutreffende Vorgehensmodelle betrachtet und deren Aufbau und Besonderheiten beschrieben.

3.3.1 Münchner Vorgehensmodell (Lindemann, 2009)

Beim *Münchner Vorgehensmodell (MVM)* handelt es sich um ein allgemeines Modell zur Lösung von Problemen. Es ist laut *Lindemann* kompatibel mit bestehenden Vorgehensmodellen und seine Grundlage bilden die drei allgemeinen Problemlösungsschritte

- Problem klären,
- Lösungsalternativen suchen,
- Entscheidung herbeiführen.

Das MVM stellt folglich einen Problemlösungszyklus dar, bei dem die operativen Arbeitsschritte in eine logische Abfolge gebracht werden. Aus diesem Modell geht nicht eindeutig hervor, dass Schwerpunkte auf gewisse Handlungsprinzipien gelegt werden. Eine Makrologik in Form eines Phasenmodells ist in dem MVM ebenfalls nicht ersichtlich. *Lindemann* stellt in diesem Zusammenhang jedoch das *Münchner Produktkonkretisierungsmodell (MKM)* vor, welches auf bestehenden Phasenmodellen aufbaut. Dabei handelt es sich um vier Produktmodellebenen (Anforderungsmodell, Funktionsmodell, Wirkmodell und Baummodell). Der Entwicklungsprozess durchläuft diese vier Modellebenen in mehreren Iterationsschritten.

Problemlösungszyklus – Beim Entwurf des Münchner Modells wird versucht, die Schwächen bestehender Vorgehensmodelle zu eliminieren. Im Wesentlichen dient das MVM laut *Lindemann* als:

- Hilfsmittel zur Planung von Entwicklungsprozessen

- Orientierungshilfe in Problemlösungsprozessen
- Instrument zur Reflexion und Analyse des Vorgehens

Forschungsergebnisse, die auf praktischen Problemlösungen basieren, zeigen, dass die drei allgemeinen Problemlösungsschritte sequenziell als auch iterativ durchlaufen werden. Das MVM detailliert die drei Hauptschritte in sieben Elemente.

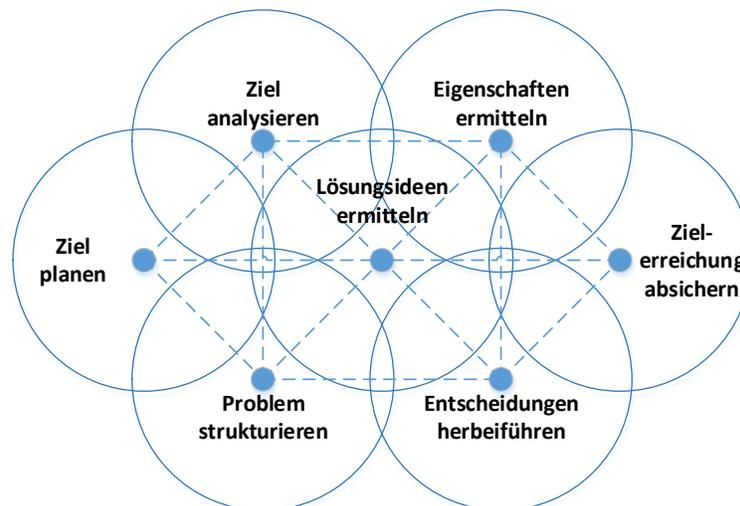


Abbildung 5: Das Münchener Vorgehensmodell (MVM) (Lindemann, 2009, S. 47)

Im Gegensatz zu anderen Vorgehensmodellen weist es, anstelle eines linearen Grundmusters, eine vernetzte Struktur auf und ähnelt somit einer Entwicklungslandkarte. Die Punkte auf der Karte stehen für bestimmte Arbeitsschritte, die prozessabhängig miteinander verbunden werden. Die in der graphischen Darstellung des MVM (siehe Abbildung 5) abgebildeten und sich überschneidenden Kreise symbolisieren die verlaufenden Grenzen der einzelnen Elemente. In Tabelle 2 werden diese näher beschrieben.

Tabelle 2: Die Elemente des Münchner Vorgehensmodells (vgl. Lindemann, 2009, S. 47 ff)

MVM Elemente	Beschreibung
Ziel planen	Zweck der Situationsanalyse ist es, konkrete Maßnahmen für das weitere Vorgehen abzuleiten. So kann die Situationsanalyse bei Bedarf auch eine Einschätzung der zukünftigen Situation gewährleisten.
Ziel analysieren	Die Zielanalyse beschäftigt sich mit der Zielsetzung, d.h., der Klärung und Beschreibung der Ziele. Mittels der Zielanalyse soll ein anforderungsgerechtes Produkt entworfen werden, weshalb es klar definierten und detaillierten Anforderungen bedarf. Sowohl Produkthanforderungen (Endergebnis) als auch Prozessanforderungen (Zwischenziele) müssen entsprechend formuliert und dokumentiert werden. Die Zielanalyse findet im Entwicklungsprozess an vielen Stellen statt.
Problem strukturieren	Durch die Problemstrukturierung kann die anschließende Lösungssuche vereinfacht werden. Eine geeignete Darstellung des Systems hilft dabei, die Problembearbeitung zu unterstützen. Die Abstraktion bildet in diesem Fall das Schlüsselement. Vom Systemmodell sollen Schwerpunkte für die Zielerreichung abgeleitet werden.
Lösungsideen ermitteln	Variantendenken oder das Denken in Alternativen ist grundlegendes Handlungsprinzip der Lösungsermittlung. Zahlreiche Teilprobleme und jeweils eine große Menge an Lösungsideen führen zu vielen Lösungsalternativen. Einzelne Teillösungen sollen zu einer möglichst optimalen Gesamtlösung zusammengesetzt werden, was eines hohen Maßes an Ordnung bedarf. Dabei ist anzumerken, dass nicht alle Teillösungen miteinander kombiniert werden können.
Eigenschaften ermitteln	Das Ermitteln von Eigenschaften bedeutet, dass relevante Merkmale der Systeme und deren Ausprägung erfasst werden. Die Eigenschaften von Lösungsideen oder Lösungsalternativen geben Aufschluss über die Zielerreichung bzw. die Erreichung der Anforderungen eines Systems. Anfänglich handelt es sich dabei um Eigenschaften wie die Machbarkeit. In späteren Phasen sind die Kosten sowie die Qualität von Interesse.
Entscheidungen herbeiführen	Entscheidungen sind in allen Phasen der Produktentwicklung zu treffen. In den anfänglichen Projektphasen üben sie deutlich mehr Einfluss auf das Endergebnis aus. Die Vielzahl späterer Detailentscheidungen ist maßgebend für die Qualität des Endproduktes.
Zielerreichung absichern	Die Absicherung der Zielerreichung soll dabei helfen, Risiken während der Umsetzung zu vermeiden. Es handelt sich um präventive Maßnahmen, die bereits in frühen Phasen des Entwicklungsprozesses zu treffen sind. Zunächst werden Risiken identifiziert und bewertet, woraufhin Vorkehrungen getroffen werden müssen, um diese auszuschließen.

Die einzelnen Elemente des MVM unterliegen keiner strikten Reihenfolge. Sie werden situationsbedingt ausgewählt und zu einer logischen Abfolge kombiniert. Wurde ein Arbeitsschritt abgeschlossen, so kann über das Vorgehen reflektiert werden, um weitere Handlungen zu planen. Im Falle auftretender Probleme kann das Modell flexibel an die Situation angepasst werden. Somit ist der Entwickler im Stande, aktiv in die Prozessgestaltung eingreifen. Ähnlich anderer Vorgehensmodelle können auch hier gewisse Standardabfolgen identifiziert werden.

Durch die vernetzte Struktur lässt sich das Vorgehensmodell individuell an die Situation anpassen. So muss das Vorgehen nicht notwendigerweise mit der Zielplanung beginnen. Einzelne Elemente können in Schleifen öfter durchlaufen werden, wodurch es zur Bildung iterativer Prozesse kommt. Der parallele Ablauf von Aktivitäten ist möglich, wobei die Kommunikation zwischen den ausführenden bzw. verantwortlichen Personen besonders von Bedeutung ist. Im MVM können darüber hinaus Rekursionen, d.h. die Verschachtelung von Vorgehensmodellen unterschiedlicher Ebenen, dargestellt werden.

3.3.2 SIMILAR-Prozess nach *Bahill* und *Gissing* (*Bahill & Gissing, 1998*)

Bahill und *Gissing* liefern mit ihrem so genannten SIMILAR-Prozess ein allgemeines Vorgehensmodell. Die Bezeichnung *SIMILAR* ist eine Abkürzung und steht für die einzelnen Prozessschritte. Im Gesamtprozess werden die Anforderungen an das Systemmodell über mehrere Stufen detailliert, wodurch sich schließlich ein komplexes System entwickelt. Dies lässt sich wie folgt darstellen:

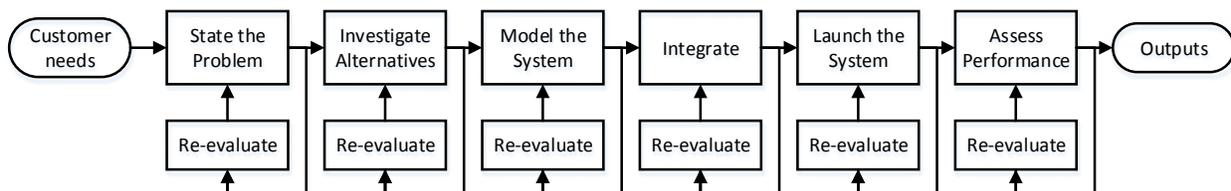


Abbildung 6: SIMILAR Prozess nach BAHILL und GISSING (vgl. Bahill & Gissing, 1998)

Den wohl wichtigsten Schritt im SIMILAR-Prozess stellt das Element *Re-evaluate* dar. Am Ende eines jeden Prozessschrittes erfolgt eine Rückkopplung auf andere Prozessteile, bei welcher eine Evaluierung stattfindet. In Tabelle 3 werden die einzelnen Schritte detailliert beschrieben.

Am Anfang des SIMILAR-Prozesses stehen die Kundenwünsche. Alleine daraus lässt sich schließen, dass es sich bei diesem Vorgehensmodell um eine Makrologik, also um ein Phasenmodell handelt. Im Gegensatz zum Münchner Modell liegt hier der Schwerpunkt auf dem Entwickeln von *Systemen*. Wie zu erkennen ist, setzt SIMILAR auch auf modellbasiertes Systems Engineering.

Tabelle 3: Elemente im SIMILAR-Prozess (vgl. Bahill & Gissing, 1998) (vgl. Haberfellner, Weck, Fricke, & Vössner, 2012)

Schritte	Beschreibung
State the Problem	Dieser Schritt beinhaltet die funktionale Beschreibung dessen, was das System können soll, bzw. eine Auflistung der Mängel, die behoben werden müssen (Beschreibung aller Anforderungen an das System, Systemziele). Informationen über die Systemziele liefern dabei unter anderem die Auftraggeber, Kunden, Anwender, Betreiber, Betroffenen, Eigentümer, Sponsoren und viele mehr. Eine Beschreibung der Anforderungen bedeutet, dass eine Aussage darüber getroffen wird, WAS getan werden muss und nicht WIE es getan werden muss.
Investigate Alternatives	Hierbei werden alternative Lösungsvarianten erarbeitet und hinsichtlich einiger Kriterien (Leistung, Kosten, Risiko) beurteilt. Es ist anzunehmen, dass keine Variante alle Kriterien zur Gänze erfüllt, weshalb Bewertungsmethoden als Entscheidungshilfe herangezogen werden können. Stehen zu einem späteren Zeitpunkt genauere Information zur Verfügung, müssen diese Analysen und Bewertungen voraussichtlich wiederholt werden. Mehrere Lösungsalternativen reduzieren die Risiken.
Model the System	Für die meisten Lösungsalternativen werden Modelle zur Datenerhebung und Analyse erstellt. Das Modell der bevorzugten Lösungsvariante wird anschließend erweitert, um den Prozess der Produktentwicklung über den gesamten Systemlebenszyklus zu unterstützen. Dabei kommen verschiedenste Modellvarianten zum Einsatz, wie analytische Gleichungen, State-Machines, Block- oder Flussdiagramme, objektorientierte Modelle, geistige Modelle als auch Simulationen. Modelle dienen u. a. dazu, Anforderungen zu ermitteln, Probleme zu erkennen und Kosten zu sparen.
Integrate	System, Geschäftsprozess und Menschen müssen, um miteinander interagieren zu können, integriert betrachtet werden. Integration bezeichnet das Zusammenführen der Teile, sodass diese entsprechend zusammenarbeiten können. In dieser Phase werden Schnittstellen der Subsysteme derart geplant, dass ein möglichst geringer Informationsaustausch (wenig Interaktion) zwischen ihnen notwendig ist. Bestenfalls übergeben Subsysteme vollständige Ergebnisse an andere.
Launch the System	Launch the System bedeutet, das System in Gang zu bringen und damit Output zu produzieren. Die Ingenieure entwickeln das Produkt und die Prozesse, um das System in die Realität umzusetzen. In einer Produktionsumwelt kann dies heißen, dass benötigte Hard- und Software, Produktionsmittel und Ressourcen etc. geplant, beschafft oder auch hergestellt werden.
Assess Performance	In diesem Schritt werden quantifizierbare Kennzahlen zur Feststellung der Leistung, Qualität, Kosten, Sicherheit, Effektivität etc. eines Systems erhoben, um das System in Hinblick auf die Erfüllung der gestellten Anforderungen zu beurteilen. Technische Daten dienen dazu, Risiken während der Entwicklung oder Produktion zu reduzieren. Außerdem sollen sie dabei helfen, die Prozesse zu bewerten und zu steuern. Was nicht zu messen ist, kann nicht kontrolliert und damit auch nicht verbessert werden!
Re-evaluate	Re-evaluate ist die <i>feedback control</i> , um die Systeme zu steuern und Leistungen zu verbessern. Diese Funktion löst Iterationen aus und ist damit die wohl wichtigste im gesamten Prozess. Die Evaluierung ist ein kontinuierlicher Prozess mit mehreren Rückkopplungen. Jede Evaluierung bietet mehrere Optionen von <i>belassen wie es ist</i> , <i>geringfügige Änderungen</i> , <i>vollständige Überarbeitung des Konzepts</i> bis hin zu <i>Projektabbruch</i> . Re-evaluate bedeutet das Beobachten des Systemoutputs und der daraus resultierenden Veränderungen des Inputs.

3.3.3 SE-Konzept nach *Hall-BWI* (Haberfellner, Weck, Fricke, & Vössner, 2012)

In Abbildung 7 ist das SE-Konzept nach *Hall-BWI* dargestellt.

An oberster Stelle steht in diesem Konzept die SE-Philosophie mit den Komponenten Systemdenken und Vorgehensmodell. Im Zentrum befindet sich der Problemlösungsprozess als die Methode zur Lösung von Problemen. Diesem sind wiederum Systemgestaltung und Projektmanagement untergeordnet. Bei der Systemgestaltung setzt man sich mit den inhaltlichen Fragen zu Problem und Lösung auseinander. Hierbei werden die Systemarchitektur erarbeitet und Konzepte einer ausgewählten Architektur konkret geschaffen. Im Gegensatz zu vielen anderen SE-Konzepten wird hier das Projektmanagement als Teil des Systems-Engineering betrachtet und nicht umgekehrt. Diese beiden Auffassungen sind jedoch nicht konträr zueinander. So geht Projektmanagement über das Systems-Engineering hinaus. Dennoch stellt es einen Teil im Systems-Engineering dar. Projektmanagement im SE sind die organisatorischen und dispositiven Maßnahmen zur Projektplanung und -steuerung. Das beinhaltet alle inhaltlichen, zeitlichen und wirtschaftlichen Managementtätigkeiten, Ressourcenplanung und -einteilung, das Treffen von Entscheidungen sowie die Betreuung von Kunden und vieles mehr.

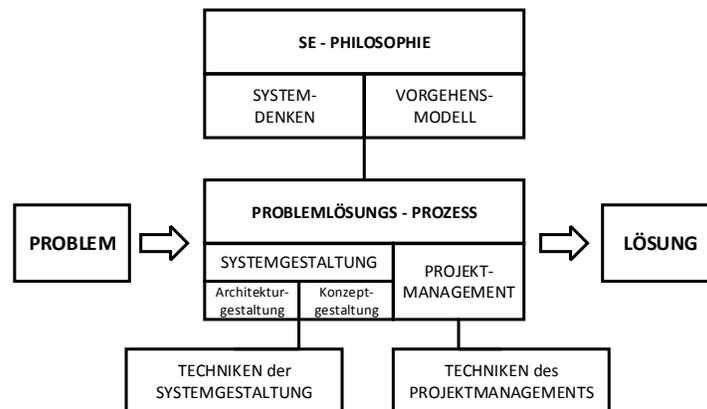


Abbildung 7: Systems Engineering Konzept nach Hall-BWI (vgl. Haberfellner, Weck, Fricke, & Vössner, 2012)

Die Methoden, Techniken und Werkzeuge bilden, sowohl für die Systemgestaltung als auch für das Projektmanagement, das Fundament des SE. Bewährte Methoden sollen insbesondere bei Entscheidungsfindungen helfen, die richtigen Entschlüsse zu treffen und somit das Risiko zu minimieren. Dazu zählen Risikomanagementmethoden, Bewertungs- und Entscheidungstechniken, Werkzeuge zur Verifikation und zur Validierung. Diese stehen jedoch nicht im Mittelpunkt des SE, sondern sind unterstützende Hilfsmittel und sollen projektabhängig zum Einsatz kommen. Die Methoden können und dürfen die Entscheidungen nicht übernehmen, jedoch erleichtern sie die Entscheidungsfindung.

Prinzipien des Handelns – Das Konzept nach *Hall-BWI* basiert, wie auch alle anderen Ansätze, auf dem Systemgedanken. Das Problem soll als System mit Elementen und Zusammenhängen betrachtet werden. Systemdenken dient dazu, das Problem als Ganzes zu sehen und zu verstehen und ganzheitliches Denken zu unterstützen. Dazu werden verschiedene Modellansätze wie die funktionale (*Black-Box*), die strukturierte (*Grey-/White-Box*) oder die systemhierarchische Betrachtungsweise herangezogen. Zur Modellierung eignen sich graphische Darstellungen wie *Bubble-Charts* oder *Matrizendarstellungen*. Dabei soll das System stets unter der Berücksichtigung verschiedener Aspekte betrachtet werden (mit verschiedenen „Brillen“). Die Betrachtung des Systems von verschiedenen Standpunkten aus führt zu einem besseren System-/Problemverständnis. Im Vorgehensmodell wird, wenn möglich, nach der *Top-Down*-Methode bzw. dem Prinzip *vom Groben ins Detail* vorgegangen. Dabei soll man sich nicht vorschnell in eine detaillierte Lösung des Problems stürzen. Die ausgiebige Ideenfindung ist als wesentlicher Bestandteil und Grundlage im gesamten Lösungsfindungsprozess zu sehen. Hiervon ausgenommen sind lediglich Probleme, die nur noch im Detail stecken. Die Anwendung des Prinzips *vom Groben ins Detail* liegt in der Qualitätsverbesserung der Problemlösung begründet. Oftmals werden Probleme zu spät erkannt, was auf eine zu voreilige Konzept-Entscheidung oder ein nicht vorhandenes Gesamtkonzept zurück zu führen ist. Um gerade in frühen Phasen des Entwicklungsprozesses gezielt richtige Management-Entscheidungen treffen zu können, ist es von Nöten, mehrere umsetzbare Lösungsvarianten einer Problemstellung zu erarbeiten. Fehlannahmen in den Anfangsphasen eines Projektes können schwerwiegende Konsequenzen für dessen Erfolg nach sich ziehen. Zur wirksamen Risikoreduktion ist die *Variantenbildung* deshalb unumgänglich. Den Schwerpunkt scheinen die folgenden vier Grundprinzipien zu bilden:

- | | |
|--|---------------------------------|
| 1. <i>Systemdenken</i> | 3. <i>Vom Groben ins Detail</i> |
| 2. <i>Betrachtung unter verschiedenen Aspekten</i> | 4. <i>Denken in Varianten</i> |

Diese Prinzipien sollten nach Möglichkeit in jeder Phase des Lebenszyklus sowie in jedem Schritt des Problemlösungszyklus eingehalten werden.

Phasenmodell – Die im *Hall-BWI*-Modell beschriebenen Projektphasen zeigen eine ähnliche Grundstruktur wie andere Phasenmodelle aus dem Systems Engineering. Dabei wird im *Hall-BWI*-Modell zwischen *Lebensphasen* und *Projektphasen* unterschieden. In Tabelle 4 werden die einzelnen Projektphasen näher beschrieben.

Tabelle 4: Projektphasen im Vorgehensmodell nach Hall-BWI (Haberfellner, Weck, Fricke, & Vössner, 2012)

Projektphase	Inhalt
Anstoß	Ein Problem, eine Idee oder eine Kundenanfrage kann Anstoß für ein Projekt sein. Der Anstoß ist die Zeitspanne von der Wahrnehmung einer unliebsamen Situation über das Bewusstwerden, diese Situation verändern zu können, bis hin zum konkreten Entschluss, diese Situation verbessern zu wollen. In dieser Phase wird unter anderem versucht, ein Problembewusstsein zu schaffen, den Nutzen einer angestrebten Lösung aufzuzeigen sowie eine Entscheidung über einen Projektstart zu treffen.
Vorstudie	Mit möglichst überschaubarem Aufwand soll die Machbarkeit (technisch, wirtschaftlich, politisch, sozial) eines Projektes geklärt werden. Zunächst ist die Notwendigkeit einer Problemlösung zu diskutieren. Die Frage, welches Problem angegangen wird und wo die Grenzen des Untersuchungsbereichs liegen, gilt es zu beantworten. Für die Abgrenzung des Untersuchungsbereichs bedarf es der Kenntnis des Ist-Zustandes. Ist der Untersuchungsbereich festgelegt, kann der Umfang der Lösung, der Gestaltungsbereich, abgesteckt und damit die Projektziele definiert werden. Am Ende der Vorstudie soll eine Entscheidung über die Fortsetzung des Projektes getroffen werden. Um eine Entscheidungsgrundlage bieten zu können, benötigt man erste Lösungsprinzipien.
Hauptstudie	In der Hauptstudie wird auf Basis des gewählten Lösungsprinzips ein Gesamtkonzept erstellt. Die Aufgaben der Hauptstudie umfassen die Zusammenstellung eines geeigneten Projektteams, die Gliederung des Projekts in Teilprojekte, die Aufteilung auf die einzelnen Projektmitglieder bis hin zur Festlegung des Projektbudgets.
Detailstudie	Die Detailstudie dient dazu, detaillierte Lösungskonzepte zu erarbeiten, um anschließend darüber entscheiden zu können, welche Gestaltungsvariante weiterverfolgt wird. Die damit verbundenen Teilprobleme oder Detailprobleme sind zu klären und die Lösung soweit zu konkretisieren, dass diese auch realisierbar wird. Außerdem ist festzustellen, ob die gestellten Anforderungen erfüllt werden und ob die Lösungen integrierbar sind.
Systembau	Hier soll das Projekt/Produkt realisiert werden. Im Gegensatz zu den vorherigen Konzeptphasen, die einen gestalterischen Prozess darstellen, handelt es sich beim Systembau um einen produktiven Prozess. Zunächst sind alle Rahmenbedingungen für eine Realisierung zu schaffen. Daraufhin erfolgt die Erstellung und Dokumentation von Hard- und Software, das Verfassen von Bedienungsanleitungen und vieles mehr. Einzelne Systemkomponenten werden, ebenso wie das integrierte Gesamtsystem, getestet. Das System ist so weit fertigzustellen, dass es eingesetzt werden kann.
Systemeinführung	Bei besonders komplexen Systemen ist eine stufenweise Systemeinführung zu empfehlen. Bevor das System übernommen wird, müssen die Spezifikationen des Auftraggebers überprüft werden. Anschließend sind die Rahmenbedingungen für die Systemeinführung zu schaffen, insbesondere in Hinblick auf die benötigte Infrastruktur. Darüber hinaus wird der Anwender geschult und das System übergeben. Die Systemeinführung kann im Rahmen einer Abschlussfeier beendet werden.
Projektabschluss	Mit einer ordentlichen Übernahme gilt das Projekt als abgeschlossen. Abschlussarbeiten wie Abrechnungen, <i>Lessons Learned</i> , Auflösung der Projektgruppe etc. sind durchzuführen. Im Anschluss an diese Phase folgt die Nutzung, Instandhaltung und Wartung oder Überarbeitung des Systems, bis das System ausgeschieden oder ersetzt wird.

Problemlösungszyklus – Grundlage des Problemlösungszyklus ist die Dewey'sche Problemlösungslogik. Dabei stellen *Zielsuche*, *Lösungssuche* und *Auswahl* die drei wichtigsten Schritte dar. Parallel zu den einzelnen Problemlösungsschritten werden Informationen beschafft und die Ergebnisse der einzelnen Schritte nachvollziehbar dokumentiert. Der Problemlösungszyklus kann in jeder Phase des Projekts zur Anwendung kommen. In

Tabelle 5 werden die einzelnen Teilschritte des Problemlösungszyklus näher erklärt.

Tabelle 5: Teilschritte des Problemlösungszyklus nach Hall-BWI (Haberfellner, Weck, Fricke, & Vössner, 2012)

Teilschritt	Inhalt
Anstoß	Dieser Schritt initiiert die Arbeitslogik. Anstoß für den Problemlösungszyklus innerhalb einer Phase können die Ergebnisse der vorangegangenen Phase im Phasenmodell sein.
Situationsanalyse	In diesem Schritt soll man sich mit der Ausgangssituation vertraut machen. Je nach Projektphase kann sich der Schritt anders gestalten. Ziel ist es, mit der Aufgabenstellung vertraut zu werden und Ziele zu definieren.
Zielformulierung	Am Ende der Zielsuche sollen mithilfe der in der Situationsanalyse gewonnenen Informationen Entscheidungen über die zu verfolgenden Ziele getroffen werden. Dabei sind die Ziele möglichst lösungsneutral, vollständig und realistisch zu formulieren. Es kann zwischen Soll-, Muss- und Wunschzielen unterschieden werden. Abschluss bildet die Zielentscheidung.
Synthese	Die Synthese von Lösungen ist ein kreativer konstruktiver Schritt, bei dem aus den Informationen vorheriger Schritte Varianten von Lösungen erarbeitet werden. Diese sollen soweit konkretisiert werden, sodass sie möglichst objektiv miteinander vergleichbar sind. Hierbei kommen Kreativitätstechniken zum Einsatz.
Analyse	Die Analyse ist ein kritischer destruktiver Schritt, bei dem die gefundenen Lösungsvarianten analytisch auseinander genommen und auf ihre Tauglichkeit überprüft werden. Es gilt herauszufinden, ob die Lösungen den Anforderungen gerecht werden oder ob sie Schwachstellen aufweisen. Außerdem soll die Analyse zeigen, inwieweit sich die Lösungen ins Gesamtsystem integrieren lassen, ob das System sicher und zuverlässig ist und welche Konsequenzen die Lösungen mit sich bringen.
Bewertung	In der Bewertung werden jene Varianten, die den Anforderungen entsprechen, verglichen und beurteilt. Häufig ist eine rein objektive Beurteilung nicht möglich, weshalb in diesem Schritt Beurteilungstechniken zum Einsatz kommen. Diese Methoden können und dürfen jedoch die Entscheidung nicht übernehmen.
Entscheidung	Anhand der Ergebnisse aus dem Bewertungsschritt soll eine Entscheidung über die zu verfolgende Lösungsvariante getroffen werden. Dabei muss man nicht auf die laut Bewertung „beste“ Lösung zurückgreifen. Oftmals werden Ziele neu überarbeitet, Änderungen am Konzept durchgeführt oder das Problem neu definiert.
Ergebnis	Ein Ergebnis sollte eine zufriedenstellende Lösung sein. Ergebnisse einer Phase können Anstoß für die darauffolgende Phase sein.

4 Mechatronische Systeme

In vielen Industriezweigen wird zunehmend auf mechatronische Systeme gesetzt, um die Produkte an die wachsenden Anforderungen anpassen zu können. Der Fachbereich der Mechatronik stellt ein Musterbeispiel dar, in dem Systems Engineering eingesetzt werden kann. In diesem Kapitel werden grundlegende Betrachtungen zur Mechatronik gemacht. Gesucht sind Antworten auf folgende drei Fragen:

1. Was bedeutet Mechatronik?
2. Was ist ein mechatronisches System?
3. Wie gestaltet sich der Entwurf von mechatronischen Systemen?

Der erste Abschnitt soll einen Einblick in die Disziplin der Mechatronik geben. Zunächst wird kurz auf die Geschichte der Mechatronik eingegangen und anschließend aufgezeigt, aus welchen Teildisziplinen diese besteht. Danach wird untersucht, was unter einem mechatronischen System verstanden wird. Ausgangspunkt der Betrachtungen sind zuvor beschriebene Definitionen. Darauf folgend wird durch funktionale und strukturelle Systembetrachtung (siehe Kapitel 2.2.3) ein allgemeines Modell für ein mechatronisches System erstellt.

Im zweiten Teil wird ein Überblick über die Modellierung von mechatronischen Systemen gegeben und gezeigt, welche Methoden zur Modellbildung und zur Simulation zur Verfügung stehen.

4.1 Mechatronik

Moderne technische Produkte sind, mitunter durch ihren interdisziplinären Charakter, geprägt durch einen hohen Grad an Komplexität. Alleine diese Eigenschaft ermöglicht es, dass heutige Produkte überhaupt erst ihren Anforderungen gerecht werden können. Unterschiedliche Fachdisziplinen kennzeichnen sich vor allem durch ihre unterschiedlichen Kompetenzen aus. Jedes Fachgebiet weist gewisse Spezialitäten auf, die von anderen Disziplinen gar nicht oder nur schwer zu realisieren sind. So könnte sich heutzutage niemand mehr einen rein mechanischen Rechner mit der gleichen Rechenleistung wie ein moderner Taschenrechner vorstellen. Durch die Kombination von verschiedenen Fachdisziplinen, sodass diese einander sinnvoll ergänzen, können Synergien erzeugt und damit Produkte geschaffen werden, die zuvor nicht denkbar waren. Hier kommt die Mechatronik ins Spiel.

Mechatronik ist ein Begriff, der 1969 von den Japanern Tetsura Mori und Ko Kikuchi als Kunstwort aus MECHANik und ElekTRONIK erschaffen und vom japanischen Unternehmen Yaskawa Electric Corporation geprägt wurde. In ihrem Sinne wurde die Erweiterung mechanischer Systeme um elektrische Komponenten verstanden. (vgl. Mori, 1969) (vgl. Rohde, 2008)

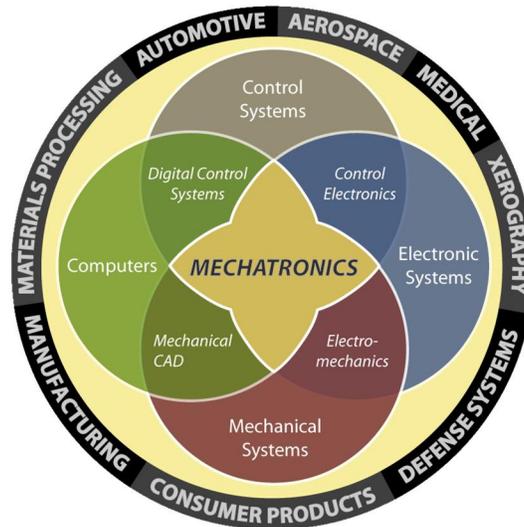


Abbildung 8: Mechatronik-Diagramm des Rensselaer Polytechnic Institute (Rohde, 2008)

Im Laufe der Zeit hat sich der Begriff jedoch einem steten Wandel unterzogen und sich zu einem geläufigen Terminus in der Technikwelt etabliert. Was jedoch nun unter Mechatronik verstanden wird, ist nicht immer einheitlich. Am geläufigsten ist wohl die allgemeine Auffassung, dass Mechatronik eine interdisziplinäre Wissenschaft aus den Fachbereichen Maschinenbau, Elektrotechnik und Informatik ist.

So definiert auch der *Verein Deutscher Ingenieure (VDI)* in seiner Richtlinie „VDI 2206 – Entwicklungsmethodik für mechatronische Systeme“ den Begriff *Mechatronik* wie folgt:

„Mechatronik bezeichnet das synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik beim Entwurf und der Herstellung industrieller Erzeugnisse sowie bei der Prozessgestaltung.“ (VDI 2206)

Dies entspricht sinngemäß der Definition des *IEEE* bzw. der *American Society of Mechanical Engineers (ASME)*, deren Definition folgendermaßen lautet:

„Mechatronics is the synergetic integration of mechanical engineering with electronic and intelligent computer control in the design and manufacturing of industrial products and processes.“ (IEEE/ASME)

Moderne Auffassungen von Mechatronik gehen jedoch über diese drei Fachrichtungen hinaus. So kann Mechatronik als eine Disziplin gesehen werden, welche eine Vielzahl unterschiedlicher, vorwiegend

technischer Gebiete, miteinander vereint. In Abbildung 8 wird eine graphische Darstellung der interdisziplinären Fachgebiete für Mechatronik gezeigt. Die Mechatronik kann Schnittstelle zwischen diesen Fachgebieten sein, eine Art *missing link*. Sie vermag dabei zu helfen, die Kommunikation zwischen unterschiedlichen Disziplinen zu verbessern und damit Barrieren zu überwinden. Mechatronik wäre demnach mehr als bloß eine Fachrichtung. YASKAWA, das Unternehmen, welches den Begriff eingeführt hat, sieht Mechatronik als eine Denkweise und schreibt:

„Mechatronics is the concept of working smarter – not harder [...] it is a blend of mechanics and the synergistic use of precision engineering, control theory, computer science, and finally sensor and actuator technology – all designed to improve products and processes.“ (Rohde, 2008)

Mechatronik hat sich zu einer Ideologie entwickelt, die der des Systems Engineering nicht unähnlich ist. Nach *Janschek* ist die Mechatronik ein:

„[...] ingenieurwissenschaftlicher Ansatz [...] der den Systemgedanken (thinking in systems) in den Mittelpunkt der Produktgestaltung stellt.“ (Janschek, 2010)

Mit diesem Denkansatz landet man auch schon direkt beim Begriff des *mechatronischen Systems*.

4.2 Aufbau mechatronischer Systeme

Systeme sind laut Definition ein Ganzes bestehend aus Elementen, die miteinander in Beziehung stehen. Nach obigen Ausführungen zum Fachbereich Mechatronik lässt sich folgende Behauptung aufstellen: Ein mechatronisches System ist ein System, das mechanische, elektrische und informationsverarbeitende Elemente in sich vereint.

4.2.1 Systemelemente

Mechanik – Mechanik ist in der Technik die Lehre von bewegten Körpern, den auf sie wirkenden und von ihnen verursachten Kräften. Aufgabe der Mechanik ist es, mathematische Modelle zur Beschreibung mechanischer Systeme zu entwerfen. (vgl. Isermann, 2008, S. 119)

Die klassische Mechanik ist die Lehre von bewegten Körpern, den auf sie wirkenden und von ihnen verursachten Kräften. Sie gliedert sich in die Teilbereiche Kinematik und Dynamik. Die Dynamik ist die Lehre von Kräften, wohingegen die Kinematik Bewegungen unabhängig von Kräften behandelt. In der Dynamik wird wiederum zwischen Statik und Kinetik unterschieden. Die Statik handelt von Kräften in ruhenden Körpern und die Kinetik beschäftigt sich mit Kräften in bewegten Körpern.

In der Mechatronik bedeutet Mechanik die Grundlage für die Realisierung von Bewegungen, Kräften als auch mechanischen Energieflüssen. (vgl. Czichos, Mechatronik, 2008, S. 23)

Basis für nahezu alle Betrachtungen bilden die klassischen *Newton'schen Gesetze*. Die wichtigsten Elemente mechanischer Modelle sind Massen, Federn sowie Dämpfer (siehe Abbildung 9).

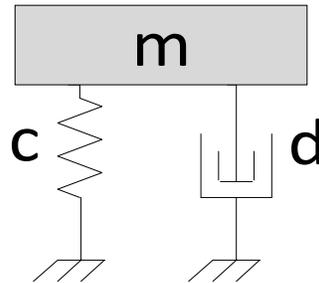


Abbildung 9: Beispiel für ein mechanisches System

Elektrotechnik – Elektrotechnik ist eine Ingenieurwissenschaft, welche sich mit den physikalischen Gesetzen von elektrischer Energie befasst. Ebenso beschäftigt sie sich mit den Erkenntnissen aus der Elektrizitätslehre (z.B. Bewegung von Ladungsträgern, Magnetismus, elektrische und magnetische Felder uvm.) sowie der analogen Schaltungstechnik (z.B. dem Aufbau elektronischer Schaltungen aus aktiven und passiven Bauelementen). Ausgangspunkt der Betrachtungen bieten die *Maxwell'schen Gleichungen*. Die wichtigsten Elemente elektrotechnischer Systeme sind Induktivitäten (Spulen), Kapazitäten (Kondensatoren) und Widerstände. Abbildung 10 veranschaulicht einen LCR-Schwingkreis als Beispiel für ein elektrisches System.

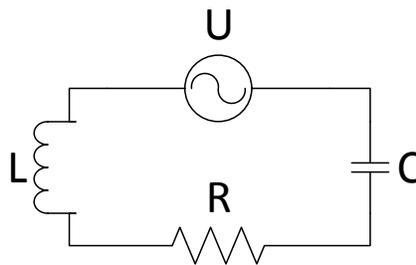


Abbildung 10: Beispiel für ein elektrisches System

Informatik – Als Informatik bezeichnet man die Wissenschaft von der elektronischen Datenverarbeitung (EDV), also die digitale Verarbeitung von Informationen mithilfe von Computern. Erst mit dem Aufkommen der Mikroprozessortechnik galt die Informationstechnik als wesentlicher Teil der Mechatronik. Grundlage der elektronischen, digitalen Datenverarbeitung und damit der Informatik stellt die *binäre Zahlendarstellung* und die *Boole'sche Algebra* dar. Die wichtigsten Elemente informationsverarbeitender Systeme sind die booleschen Operatoren *AND*, *OR*, *XOR* und *NOT*. Abbildung 11 zeigt eine Digitallogikschaltung symbolisch für ein informationsverarbeitendes System.

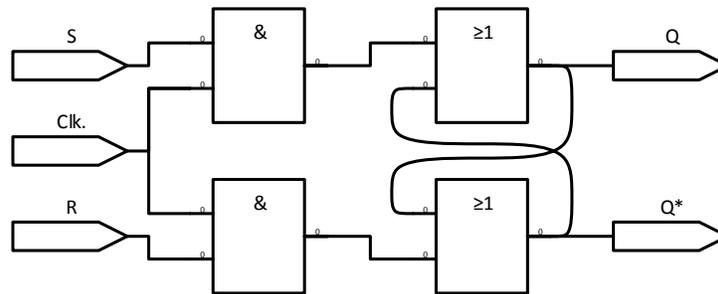


Abbildung 11: Getaktetes RS-Flip-Flop (vgl. Tietze & Schenk, 2002)

Die Frage, die sich nun stellt, ist: „Reicht für ein System das alleinige Vorhandensein von elektrischen, mechanischen und informationsverarbeitenden Komponenten aus, um als mechatronisches System zu gelten?“.

Im Gegensatz zu konventionellen mechanisch-elektrischen Systemen, bei denen die mechanischen von den elektronischen Komponenten räumlich getrennt sind, wird beim Entwurf von mechatronischen Systemen die Mechanik, Elektronik und digitale Informationsverarbeitung als integriertes Gesamtsystem betrachtet, mit dem Ziel Synergie-Effekte zu erwirken. (vgl. Isermann, 2008)

„Letztlich existiert in mechatronischen Bauteilen keine exakte Grenze mehr zwischen Mechanik, Elektronik und Informatik.“ (Winzer, 2013, S. 13)

Isermann definiert angelehnt an das *International Federation of Automatic Control - Technical Committees (IFAC-TC)* mechatronische Systeme wie folgt:

„Mechatronische Systeme entstehen durch simultanes Entwerfen und die Integration von folgenden Komponenten oder Prozessen:

Mechanische und mit ihr gekoppelte Komponenten/Prozesse

Elektronische Komponenten/Prozesse

Informationstechnik (einschließlich Automatisierungstechnik)

Die Integration erfolgt durch die Komponenten (Hardware) und durch die informationsverarbeitenden Funktionen (Software). Ziel ist dabei, eine optimale Lösung zu finden zwischen der mechanischen Struktur, Sensor- und Aktor-Implementierung, automatischer digitaler Informationsverarbeitung und Regelung. Zusätzlich werden synergetische Effekte geschaffen, die erweiterte Funktionen und innovative Lösungen ergeben.“ (Isermann, 2008, S. 18)

Über weitere Unterschiede zu konventionellen Systemen sollen die folgenden Betrachtungen Aufschluss geben.

4.2.2 Systemstruktur

Bei mechatronischen Systemen handelt es sich in aller Regel um technische Systeme. Diese kennzeichnen sich durch ihren funktionalen Charakter und werden dazu erschaffen, einen gewissen Zweck zu erfüllen. Dabei interagieren mechatronische Systeme im Wesentlichen mit zwei Akteuren, der Umwelt und dem Menschen (bzw. dem Nutzer) als Teil dieser Umwelt.

„Ein mechatronisches System existiert nie als Selbstzweck, sondern besitzt immer eine direkte Interaktion mit einem Nutzer und damit eine aufgabenbezogene Mensch-Maschine-Schnittstelle.“ (Janschek, 2010, S. 4)

Im Sinne des Systemdenkens ist ein System durch die Ganzheit bestimmt und nicht nur durch die Summe seiner Einzelteile und deren Eigenschaften. Die inneren Zusammenhänge bestimmen demnach auch wesentlich die Funktion eines mechatronischen Systems.

„Bei mechatronischen Systemen erfolgt die Lösung einer Aufgabe sowohl auf mechanischem als auch digital-elektronischem Wege. Hierbei spielen die Wechselbeziehungen bei der Konstruktion eine Rolle.“ (Isermann, 2008, S. 4)

Janschek schreibt das Verhalten eines mechatronischen Systems seiner funktionellen Struktur zu, die er wie folgt bezeichnet:

„[...] eine technologieunabhängige Wirkstruktur [...] auf abstrakter Modellebene das Zusammenwirken (Interaktion) physikalischer Systemgrößen.“ (Janschek, 2010, S. 5)

Blickt man auf die innere Struktur eines mechatronischen Systems, so lässt sich in den meisten Fällen eine geschlossene Wirkungskette erkennen, welche deutlich macht, dass die verschiedenen Technologieelemente funktionell rigoros zusammen arbeiten müssen.

„Ein mechatronisches System kann eben nur dann seine Produktaufgaben ordnungsgemäß erfüllen, wenn alle Elemente der geschlossenen Wirkungskette aufeinander abgestimmt sind.“ (Janschek, 2010, S. 6)

Solch eine geschlossene Wirkungskette, oder auch Feedbackschleife, ist im Allgemeinen als *Regelkreis* bekannt.

„Mechatronische Systeme [...] können in Funktionsgruppen unterteilt werden, die meist Regelkreise bilden und aus Modulen mit mechanisch-elektrisch-magnetisch-thermisch-optischen Bauelementen, Sensorik zur Erfassung von Messgrößen des Systemzustandes, Aktorik zur Regelung und Steuerung sowie Prozessorik und Informatik zur Informationsverarbeitung bestehen.“ (Czichos, Mechatronik, 2008, S. 1)

Zusammenfassend kann also gesagt werden, dass ein mechatronisches System ein zweckdienliches technisches System ist, welches in sich mechanische, elektrische und informationsverarbeitende

Elemente vereint. In seinem Inneren weist ein mechatronisches System eine geschlossene Wirkungskette auf. Dieser Regelkreis wird aus einem (meist mechanischen) Grundsystem, einer Sensorik zur Erfassung der Zustandsgrößen, elektronischen und informationsverarbeitenden Bestandteilen (Signalkonditionierung, Prozessoren etc.) zur Verarbeitung der erfassten Daten sowie Aktorik zur aktiven Beeinflussung der mechanischen Größen gebildet.

Abbildung 12 zeigt die wesentlichen Elemente eines mechatronischen Systems in Anlehnung an einen Regelkreis. Dabei sind die nach der funktionalen Betrachtung bedeutsamen Beziehungen zwischen System und Umwelt als allgemeine Ein- und Ausgänge und auch die von *Janschek* (Janschek, 2010) beschriebene Mensch-Maschine-Schnittstelle berücksichtigt.

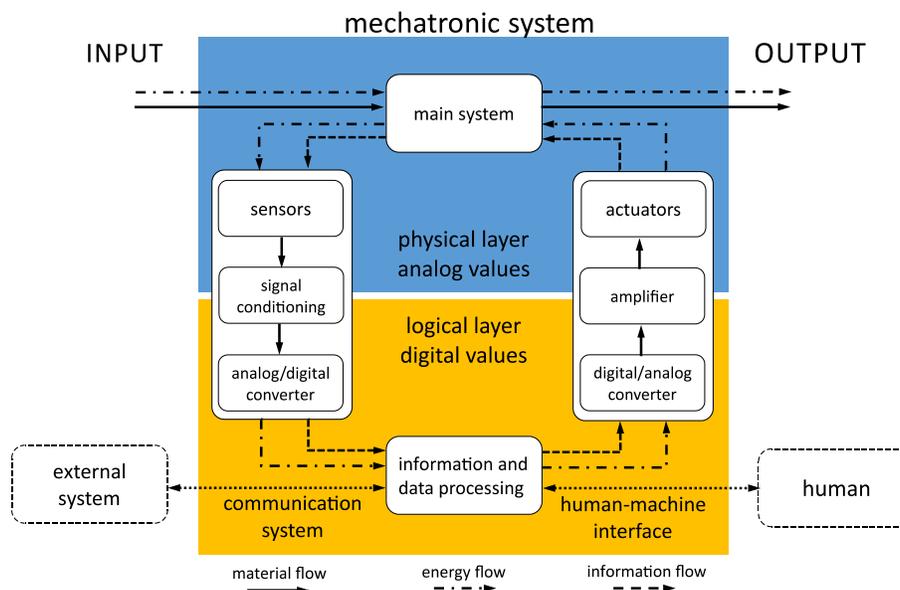


Abbildung 12: Allgemeine Darstellung eines mechatronischen Systems (vgl. *TransMechatronic*, 2014)

Ebenso lässt sich der laut strukturorientierter Betrachtung identifizierte Regelkreis im Inneren des Systems erkennen. Wie von *Czichos* (Czichos, Mechatronik, 2008) beschrieben, zeigt das Modell auch die vier zentralen Funktionsgruppen, wobei Sensorik und Aktorik in der Darstellung durch Signalkonditionierung und Wandler erweitert wurden.

4.3 Modellierung mechatronischer Systeme

Dieses Unterkapitel setzt sich mit dem Entwurf mechatronischer Systeme und der damit notwendigerweise verbundenen Nutzung ingenieurwissenschaftlicher Mittel auseinander. Selbst wenn die Struktur eines mechatronischen Systems eine geschlossene Wirkungskette aufweist, ist der Entwurf eines solchen Systems nicht mit dem eines Regelkreises vergleichbar.

„Der Entwurf eines mechatronischen Systems geht weit über den Entwurf eines Regelkreises hinaus“ (Janschek, 2010, S. 7)

Wie in Kapitel 3 bereits geklärt wurde, bedarf es zur Problemlösung solcher Systeme einem geordneten Umgang mit Komplexität. Dazu kommen Methoden und Werkzeuge des Systementwurfs zum Einsatz. Hierbei werden im Allgemeinen Vorgehensmodelle und Verhaltensmodelle unterschieden.

Vorgehensmodelle dienen dem Entwurf des (technischen) Systems im Entwicklungsprozess. Das bedeutet, sie strukturieren das Projekt in Hinblick auf die erforderlichen Tätigkeiten oder Arbeitsschritte. Beispiele sind Projektphasenmodelle, Problemlösungszyklen oder auch gestalterische Grundsätze (siehe Kapitel 3.1.4).

Im Gegensatz dazu sollen Verhaltensmodelle technischer Systeme (vgl. Simulationsmodell in Kapitel 2.3.1) das reale System so modellieren, dass mittels Simulationen Aussagen über das (dynamische) Verhalten der Systeme getroffen werden. Beispiele sind mathematische Modelle, Rechnermodelle oder allgemein qualitative Systemmodelle, welche die Funktion und Struktur der mechatronischen Systeme beschreiben.

Wenn von technischer Modellbildung gesprochen wird, so handelt es sich in der Regel um die Bildung von Verhaltensmodellen und deren Simulation. Aufgabe der Modellbildung technischer Systeme ist es, aus einer verbalen Beschreibung des technischen Systems eine mathematische Beschreibung des Systems zu erstellen. Erst mithilfe mathematischer Modelle können Computersimulationen durchgeführt oder geeignete Regler entworfen werden. Dabei handelt es sich üblicherweise um Differentialgleichungen (DGL) oder Differentialgleichungssysteme. Mithilfe der Anfangsbedingungen lassen sich genaue Aussagen über das Modell, jedoch nicht über das reale System treffen. Das Modell muss nicht unbedingt das Verhalten des realen Systems exakt beschreiben. (vgl. Roddeck, 2006)

4.3.1 Strukturuntersuchung

Reale technische Systeme müssen wegen ihrer Komplexität in vereinfachte Modelle übergeführt werden. Aufgrund solcher vereinfachter Annahmen in Hinblick auf die innere Struktur als auch die

Anfangsbedingungen besitzen Modelle stets begrenzte Gültigkeit. Um ein technisches Modell des realen Systems zu entwickeln, gibt es prinzipiell zwei Möglichkeiten (vgl. Roddeck, 2006, S. 132 & 301):

- Systemanalyse: Dabei handelt es sich um eine *strukturorientierte Methodik* (vgl. Kapitel 2.2.3). Wenn relativ genaue Aussagen über die innere Struktur des Systems gemacht werden können (*System = White-Box*), so besteht die Möglichkeit, den direkten Weg über die Systemanalyse zu nehmen. Das heißt, dass ein Modell ohne Umwege als Abbildung der Systemstruktur entworfen wird.
- Systemidentifikation: Bei dieser Möglichkeit handelt es sich um eine funktionsorientierte Methodik (vgl. Kapitel 2.2.2). Wenn keine Aussagen über die innere Struktur des Systems gemacht werden kann (*System = Black-Box*), so hilft die Anwendung experimenteller Methoden. Anhand von Experimenten und Analysen der Eingangs- und Ausgangsgrößen können Modelle gestaltet werden, die ein vergleichbares Systemverhalten aufweisen.

4.3.2 Technische Modellbildung

Entsprechend dem Gedankengut des Systems Engineering hat Richard P. Feynman in den 1960er Jahren einen ganzheitlichen Denkansatz zur mathematischen Beschreibung physikalischer Systeme entwickelt. In seiner Arbeit „The Feynman Lectures on Physics“ wurden Analogien zwischen verschiedensten Bereichen in der Physik gemacht. Vorrangig aus dem Bereich Mechanik und Elektrotechnik aber auch der Wärme- und Strömungslehre. So zeigen beispielsweise mechanische und elektromagnetische Systeme ähnliche mathematische Zusammenhänge. (vgl. Czichos, Mechatronik, 2008)

Erst diese Zusammenhänge bieten eine mathematische Grundlage für eine technische Modellbildung von mechatronischen Systemen. Dank dieser können mechanische wie elektrische Komponenten in ein gemeinsames Modell übergeführt werden.

Dabei sind fünf verschiedene Strukturelemente zu unterscheiden:

- Quelle (z.B. Kräfte, Momente, elektische oder magnetische Felder)
- Speicher (z.B. Federn oder Kondensatoren)
- Übertrager (z.B. Hebel, Getriebe oder Transformatoren)
- Wandler (z.B. Kolben, Elektromotoren oder Generatoren)
- Senke (z.B. Dämpfer, Reibung oder Widerstände)

Ziel der Modellbildung ist es, ein mathematisches Modell des Systems zu bilden. Dazu haben sich in der Mechanik verschiedene Verfahren etabliert, die durch methodisches Vorgehen zu einer Lösung führen.

Für das Aufstellen des mathematischen Modells haben sich einige Methoden entwickelt. Diese entstammen zu großem Teil der technischen Mechanik, können aber, mit obigen Analogien, partiell auch auf die elektrischen Modelle angewandt werden. In der technischen Mechanik dienen sie dazu, Bewegungsgleichungen aufzustellen.

Die folgenden Prinzipien oder Methoden sind hierbei fundamental:

- Prinzip der virtuellen Verschiebung
- Prinzip von d'Alembert
- Prinzip von Joardin
- Prinzip nach Hamilton
- Prinzip nach Lagrange
- Bond-Graphen

Auf die genannten Prinzipien soll hier jedoch nicht näher eingegangen werden, mit Ausnahme der Modellierung von Systemen durch Bond-Graphen.

Bond-Graphen – Im Vergleich zu obigen mathematischen Methoden sind Bond-Graphen eine Möglichkeit einer graphischen Modellbildung. Die Methode der Bond-Graphen zeichnet sich vor allem durch ihren interdisziplinären Charakter aus, weshalb sie sich besonders für mechatronische Systeme eignet.

Henry Paynter hat 1959 diese Methode am *Massachusetts Institute of Technology (MIT)* entwickelt, deren Grundsatz die Energieerhaltung eines Systems ist. In allen Fachgebieten (z.B. Mechanik, Elektrotechnik, Thermodynamik etc.) kann die Leistung oder Energie als Produkt zweier Größen – einer Potential- und einer Flussgröße – gebildet werden. Diese Größen werden im betreffenden Modell als *effort* (Anstrengung) und *flow* (Fluss) bezeichnet. (vgl. Roddeck, 2006)

Tabelle 6: Potential- und Flussgrößen in verschiedenen Fachgebieten

Fachgebiet	Effort (= Ursache)	Flow (= Wirkung)
Mechanik	Kraft	Geschwindigkeit
Elektrotechnik	Spannung	Strom
Strömungslehre	Druck	Volumenstrom
Thermodynamik	Temperatur	Entropiefluss
Chemie	Chemisches Potential	Molarer Fluss

Mithilfe der identifizierten allgemeinen Strukturelemente (Quellen, Senken, Speicher etc.), die es analog in jedem physikalischen System gibt, können Graphen erstellt werden. Durch ein analytisches bzw. methodisches Vorgehen kann anhand dieser Graphen ein mathematisches Modell erzeugt werden.

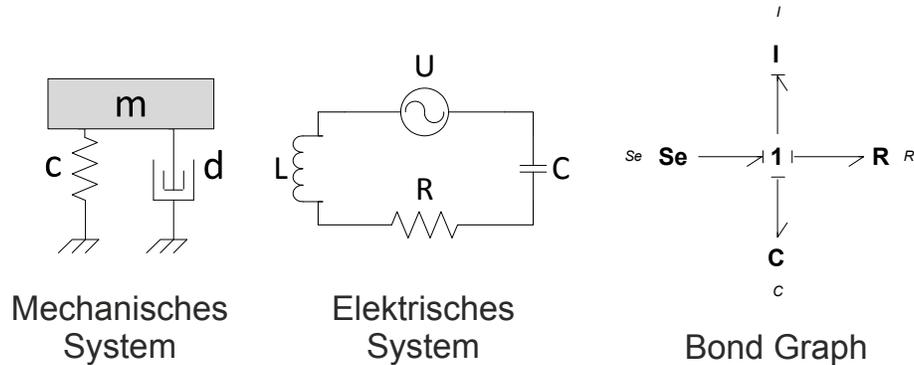


Abbildung 13: Bond-Graph eines mechanischen und elektrischen Systems

In Abbildung 13 ist der Bond-Graph für das abgebildete mechanische System gezeigt, welcher in analoger Weise auch für das elektrische System gilt. Ähnlich der Modellierungssprachen (siehe Kapitel 2.3.3) besitzen Bond-Graphen damit eine transdisziplinäre Syntax, wodurch eine gemeinsame technische Modellbildung mechatronischer Systeme möglich wird.

4.3.3 Technische Simulation

Um die aus der technischen Modellbildung erhaltenen mathematischen Modelle auch simulieren zu können, benötigt man Lösungsverfahren oder -algorithmen. Da im Normalfall die bei der Modellierung entstehenden Differentialgleichungen aufgrund ihrer Komplexität auf algebraischem Wege nicht lösbar sind, bedient man sich der numerischen Mathematik. Numerische Verfahren bedürfen eines enormen Rechenaufwandes, weshalb man zur Lösung auf Rechnerunterstützung setzt. Simulationen die auf diesem Prinzip beruhen, bezeichnet man als Computersimulationen (vgl. Kapitel 2.3.2). Zur numerischen Lösung von Differentialgleichungen haben sich verschiedenste Verfahren etabliert. Beispiele dafür sind:

- Runge-Kutta-Verfahren (Einschrittverfahren zur Lösung gewöhnlicher DGL)
- Prädiktor-Korrektor-Verfahren (Mehrschrittverfahren zur Lösung gewöhnlicher DGL)
- Finite-Elemente-Verfahren (Numerisches Verfahren zur Lösung partieller DGL)

Auf diese soll hier jedoch nicht näher eingegangen werden.

5 Fallbeispiel

Im angeführten Fallbeispiel wird versucht ein Projekt aus dem Fachgebiet der Mechatronik durchzuführen. Dabei soll das SE-Konzept nach *Hall-BWI* zu Hilfe genommen werden. Vorab sei erneut anzumerken, dass es sich bei mechatronischen Systemen um oft komplexe Systeme handelt die von mehreren Fachkräften entwickelt werden. Im Folgenden wird ein Teilsystem entwickelt, das in ein bestehendes Gesamtsystem integriert werden soll. Das vorliegende Kapitel gliedert sich wie folgt:

1. Anstoß
2. Vorstudie
3. Hauptstudie
4. Detailstudie
5. Systembau
6. Systemeinführung und Projektabschluss

Im ersten Abschnitt wird ein Einblick darüber gegeben, welche Situation zu dieser Arbeit geführt hat. So machten Beschwerden seitens der Mitarbeiter im Partnerunternehmen ein bestehendes Problem bewusst, welches in Angriff zu nehmen war.

In einer Vorstudie wird eine Situationsanalyse der aktuellen Prüfstandsituation durchgeführt. Mithilfe einfacher Modellierungstechniken werden die Prüfstände als System erfasst und geklärt ob es sich bei den vorliegenden Prüfständen um mechatronische Systeme handelt. Anschließend wird gezeigt, wie aus dem allgemeinen Modell für mechatronische Systeme der Gestaltungsbereich und Untersuchungsbereich identifiziert werden kann. Drei Varianten von Lösungsprinzipien werden gebildet, wovon eine weiterverfolgt wird.

In der Hauptstudie wird das gewählte Lösungsprinzip detailliert. In Zusammenarbeit mit dem Auftraggeber wird eine Systemarchitektur festgelegt. Darauf hin wird ein Blick ins Innere des Zielsystems geworfen, woraus das Gesamtkonzept abzuleiten ist. Die Systemkomponenten werden identifiziert und im Zuge einer Produktrecherche ausgewählt. Es folgen Schilderungen bezüglich Möglichkeiten zur Gliederung in Teilprojekte und die damit verbundenen Problematiken.

In der Detailstudie wird das gewählte Lösungskonzept detailliert und an einem Beispiel gezeigt, wie Teilprobleme durch iteratives Anwenden des Problemlösungszyklus gelöst werden. Im Anschluss daran wird gezeigt, wie durch technische Modellbildung und Simulation die Interoperabilität der einzelnen Elemente festgestellt wird.

Im sechsten Abschnitt wird die technische Realisierung von Hard- und Software gezeigt. Hier werden problemspezifische Vorgehensweisen aufgezeigt. Dabei wird gezeigt, wie auch in der Realisierungsphase die Grundprinzipien des Systems Engineering Anwendung finden. Der zweite Teil des Systembaus veranschaulicht die Umsetzung der Software. Dabei wurden Methoden zur zyklischen Analyse und Synthese der Software entwickelt. Abstraktion und Systemdenken spielen dabei eine wichtige Rolle.

Im letzten Abschnitt werden folgen einige abschließende Bemerkungen zur Arbeit.

5.1 Anstoß

5.1.1 Ausgangssituation

Im Zuge der Entwicklungstätigkeit der Mechatronik-Abteilung bei LOGICDATA werden die Aktuatoren in verschiedensten Prüfständen unter anderem auf ihre Belastungsfähigkeit im Dauereinsatz getestet, auf Produzierbarkeit in großer Serie überprüft, sowie Leistungsdaten ermittelt. Im Zentrum der Betrachtung stehen vor allem die Dauerlaufprüfstände. Verschiedene Aktuatoren werden hier auf ihren zukünftigen Einsatz im Feld getestet. Derzeit sind im Schnitt zwei Personen für diese Testaufbauten zuständig. Nachdem die Prüfstände von den zuständigen Mitarbeitern in Betrieb genommen werden, werden diese nur noch sporadisch überprüft. Nachteil der derzeit verwendeten Prüfstände liegt vor allem in deren ständigem Überwachungs- und Wartungsbedarf. Nachdem die Testaufbauten nicht voll automatisiert sind und die Überprüfung in unregelmäßigen Zeitabständen erfolgt, kann meist keine genaue Aussage über den exakten Ausfallszeitpunkt gemacht werden. Erschwerend kommt hinzu, dass der Arbeitsplatz des Prüfstandpersonals örtlich relativ weit von den Testräumen getrennt ist. Dies steht entgegen dem Gedankenansatz des *lean development (lean engineering)*, welches unter anderem darauf abzielt unnötige Wege zu vermeiden.

5.1.2 Problembewusstsein

Durch diese Situation entstand die Idee eine Möglichkeit zu schaffen, gewisse Tätigkeiten vom Büroarbeitsplatz aus durchführen zu können. Gespräche mit dem Prüfstandpersonal machten das Vorhandensein gewisser Missstände klar, ohne dass man sich über ein vorherrschendes Problem bewusst war. Das Personal tätigte Aussagen und äußerte Wünsche oder Ideen die im Folgenden sinngemäß wiedergegeben werden:

- Grundsätzlich funktionieren die Prüfstände ja ganz gut. Die meiste Arbeit steckt im Aufbau der Prüfstände und in der Inbetriebnahme. Wenn der Prüfstand einmal läuft, gibt es nur selten

schwerwiegende Probleme. Körperlich anstrengend ist es vor allem, den Prüfstand mit den Gewichten zu beladen, aber hier wollen wir in Zukunft sowieso auf eine Pneumatik oder Ähnliches umstellen.

- Es wäre toll, wenn man darüber informiert würde, wenn ein Aktuator die in den Spezifikationen angegebenen Durchläufe erledigt hat. Man müsse dann weniger oft den Test-Raum aufsuchen, um die Anzahl der durchlaufenen Zyklen zu überprüfen.
- Außerdem wissen wir nicht, wann ein Prüfstand defekt ist, manchmal sind die Getriebe der Aktuatoren beschädigt. Dann läuft der Prüfstand weiter, ohne dass sich der Aktuator in irgendeine Richtung bewegt! Das ist problematisch, denn der eingebaute Zähler, der die Anzahl der Versuche aufnimmt, läuft unermüdlich weiter. Schließlich können wir nicht exakt sagen, wie viele Durchläufe der Aktuator tatsächlich geschafft hat.
- Es muss ja nicht einmal ein mechanischer Defekt sein, es ist ja durchaus möglich, dass einfach nur ein Motor bei den Umbauarbeiten unbemerkt abgesteckt wurde. Dann tut sich gar nichts, bis der Fehler entdeckt wird.
- Es wäre schon eine große Hilfe, wenn wir den Prüfstand vom Arbeitsplatz aus überwachen können. Wenn wir früher wissen, dass ein Aktuator Schaden genommen hat, oder ein Fehler aufgetreten ist, dann könnten wir auch das Problem früher beheben und die Prüfstände stehen nicht still.
- Außerdem ist es relativ aufwändig einen neuen Prüfzyklus zu programmieren, es ist umständlich Änderungen durchzuführen. Man kann die neuen Einstellungen auch nicht einfach auf mehrere Prüfstände übertragen!
- Zwischendurch müssen Messungen an den Aktuatoren durchgeführt werden, um festzustellen ob z.B. die elektrischen Spezifikationen auch nach längerem Betrieb noch eingehalten werden. Dies ist mit Aufwand verbunden und wir müssen die Prüfstände zeitweilen abschalten.
- Diese Messungen haben noch einen gravierenden Nachteil, wir können nämlich keine Aussage darüber machen, wann die Grenzen der elektrischen Spezifikationen überschritten wurden. Jedoch kommt dieser Fehler nahezu nie vor!
- Toll wäre es auch, wenn wir vom Arbeitsplatz aus auf unsere Motorsteuerung zugreifen könnten. Wir haben hier verschiedene Softwaretools, die auch für die Auswertung der Messdaten verwendet werden können. Auf den Prüfständen wird dies derzeit selten gemacht, aber oft weiß man ja nicht, dass man etwas braucht, bevor man es tatsächlich hat. Außerdem können wir auf

diesem Wege neue Parameter in die Steuerung übertragen ohne diese aus dem Prüfstand auszubauen.

- Man könnte ja auch die Prüfstände aus der Ferne bedienen, z.B. um einen Prüfstand bzw. die Motorsteuerung zu resetieren. Also die ganz normalen Funktionen die auch unser Handschalter bietet, nur von unseren Bürocomputern aus.
- Zu den üblichen elektrischen Daten wäre auch eine Temperaturmessung der Aktuatoren toll. Vielleicht könnte man dann auch schon frühzeitig sagen, wenn ein Motorschaden oder ähnliches droht. Mit einer Temperaturmessung an den Prüfständen könnte man auch den Zusammenhang zwischen Temperatur und den Leistungsdaten herausfinden.

5.1.3 Entscheidung

Mit den zuvor genannten Misständen wurde an die Abteilungsleitung herangetreten und (formlos) der Vorschlag unterbreitet, die derzeitige Testsituation zu verbessern. Vorgeschlagen wurde die Durchführung vorliegender Diplomarbeit mit dem Ziel die ständigen Überwachungstätigkeiten zum Teil zu vereinfachen und die Möglichkeit zu schaffen, gewisse Tätigkeiten von jedem Arbeitsplatz aus durchführen zu können. Steuerungs- und Wartungsaufgaben sollen einfacher und besser automatisiert werden. Damit sollen Einflüsse durch von Menschen verursachte Fehler minimiert, Prüfabläufe beschleunigt und somit Arbeitszeit und Kosten gespart werden. Im Sinne der ISO 9001 ist das Management dazu angehalten, auf ständige Leistungsverbesserung zu achten. Aus diesem Grund werden konkrete Verbesserungsvorschläge mit mehr Selbstverständnis angenommen. Somit wurde von der Abteilungsleitung die Genehmigung erteilt eine Vorstudie durchzuführen und einen Projektantrag zu erstellen.

Der Projektantrag sollte folgende Punkte enthalten:

1. Projektbeschreibung: Dies umfasst das Thema, die Problembeschreibung, allgemeine Ziele, die Ausgangssituation, Vor- und Nachteile, mögliche Risiken sowie Lösungsvorschläge.
2. Projektressourcen: Das bedeutet sowohl personelle und materielle Ressourcen als auch, die benötigten internen und externen Betriebsmittel sowie natürlich auch finanzielle Ressourcen.
3. Arbeitsplan: Damit ist ein voraussichtlicher Zeitplan oder Meilensteinplan gemeint.

5.2 Vorstudie

In der Vorstudie soll weitestgehend die Umsetzbarkeit des Systems untersucht werden. Die Frage nach einer politischen Machbarkeit erübrigt sich in diesem Beispiel. Lediglich die technische sowie die wirtschaftliche Komponente sollen betrachtet werden. Bevor dies geschieht, wird kurz in einer Problemidentifikation das Problem spezifiziert. Danach soll in einer Ist-Analyse der Untersuchungsbereich festgestellt werden. Ausgehend vom Untersuchungsbereich erfolgt die Absteckung des Gestaltungsbereichs. Nach dem Aufstellen von einigen Lösungsprinzipien soll über die Fortsetzung des Projektes entschieden werden.

Damit sind die Schritte der Vorstudie:

1. Problemidentifikation
2. Untersuchungsbereich abstecken
3. Gestaltungsbereich abstecken
4. Projektziele definieren
5. Lösungsprinzipien erarbeiten
6. Machbarkeit feststellen
7. Projektentscheidung treffen

5.2.1 Problemidentifikation

Zentrale Frage bei der Identifikation lautet: „Was wollen wir überhaupt?“. Dazu sollen die vom Personal getätigten Aussagen, Wünsche etc. über die Prüfstandsituation näher betrachtet werden, um möglichst lösungsneutral das Problem zu erfassen.

Die folgenden lösungsneutralen Probleme wurden festgestellt:

- Großer Aufwand für Überwachungs- und Wartungstätigkeiten.
- Schlechte Erfassung der Zustandsdaten des Prüfstandes.
- Schlechter Zugang zu Zustandsinformationen des Prüfstandes.
- Unnötiger Weg der zurückgelegt wird.
- Veränderungen am Prüfprogramm sind mit entsprechendem Aufwand verbunden.
- Unnötige Stillstandzeiten der Prüfstände durch Defekte oder Fehler.

5.2.2 Untersuchungsbereich

Drei verschiedene Arten von Prüfständen sollen näher untersucht werden. Diese Prüfstandssysteme stellen somit den Untersuchungsbereich dar. Der Ist-Zustand der Systeme soll Aufschluss darüber geben, wo die Probleme in der aktuellen Situation vorherrschen. Dazu sollen die Prüfstände näher betrachtet werden, welche Versuche durchgeführt werden und wie sich diese zurzeit gestalten.

Prüfstände – Drei Arten von Dauerlaufprüfständen wurden betrachtet (siehe Abbildung 14):

- Säulenprüfstand: Bei den einfachen Säulenprüfständen werden so genannte *Inlineantriebe* auf den Dauereinsatz getestet. Dazu werden die Aktuatoren in Metallsäulen verbaut und auf einer stabilen Unterkonstruktion befestigt. An der Oberseite der Säule wird eine Deckplatte fest verschraubt und mit Stahlgewichten beschwert. Schließlich wird der Dauerlauf gestartet, bei welchem die Last in einem zuvor eingestellten Zyklus auf und ab bewegt wird.
- Hebelprüfstand: Die Hebelprüfstände wurden zum Testen der Linearaktuatoren mit 90-Grad-Antrieb entwickelt. Die Aktuatoren können je nach Einbaulage im Prüfstand auf Druck- oder auf Zugbelastung getestet werden. Mithilfe von Stahlgewichten kann die Belastung variiert werden. Nach Starten des Dauerlaufes wird der Hebelarm in einem eingestellten Zyklus auf und ab bewegt.
- Massagemotorenprüfstand: Zum Dauerlaufversuch der Massagemotoren wurden Prüfstände entworfen, bei welchen die Motoren zwischen zwei Schaumstoffwürfeln eingespannt werden. Über die Einspannkraft kann die Belastung auf die Motoren verändert werden. In dieser Konstruktion werden die Motoren auf ihren Dauereinsatz getestet.

Neben diesen drei Arten von Dauerlaufprüfständen gibt es noch eine Reihe von anderen Prüfständen auf welche jedoch nicht näher behandelt wurden.



Abbildung 14: Dauerlaufprüfstände der Fa. LOGICDATA, v. l. n. r.: Säulenprüfstand, Hebelprüfstand, Massagemotorenprüfstand

Aktuatoren – Drei Klassen von Aktuatoren werden getestet (siehe Abbildung 15), diese sind:

- Inlineantriebe: Die so genannten Inlineantriebe werden in erster Linie in höhenverstellbaren Bürotischen verbaut. Dabei sind Motor, Getriebe und Spindeltrieb in einer Linie aufgebaut. Um einen möglichst großen Verfahrweg bei kleiner Baulänge zu gewährleisten, wurde der Spindeltrieb zweistufig (teleskopartig) aufgebaut.
- 90-Grad-Antriebe: Im Gegensatz dazu verfügen die Linearaktuatoren mit 90-Grad-Antrieb nur über einen einfachen Spindeltrieb, bei dem der Motor über ein Schneckengetriebe um 90-Grad versetzt zur Spindelachse montiert ist. Dieser Aufbau ermöglicht eine hohe Stellkraft der Aktuatoren. Die gängigste Anwendung der 90-Grad-Antriebe sind verstellbare Betten und Couch-Sessel. Beide Linearaktuatoren, sowohl Inline- als auch 90-Grad-Antrieb, verfügen über Hallimpulsgeber zur Feststellung des Verfahrweges bzw. der Position sowie zur Drehrichtungserkennung. Die 90-Grad-Antriebe besitzen darüber hinaus auch Endschalter, welche den Verfahrweg begrenzen. Bei den Inlineantrieben hingegen wird sich ausschließlich auf die Referenzfahrt und die eingestellten Softwareschranken verlassen.
- Massagemotoren: Die Massagemotoren kommen ebenfalls bei Betten und eventuell Sesseln zum Einsatz. Wie bei allen anderen Antrieben handelt es sich hier um bürstenbehaftete Gleichstrommotoren. Diese sind mit einer Unwucht versehen, um Vibrationen zu erzeugen. Außerdem verfügen die Massagemotoren über einen integrierten Thermoschalter zum Schutz vor Überhitzung.



Abbildung 15: Beispiele für Aktuatoren bzw. Prüflinge, v. l. n. r. Inlineantrieb, 90-Grad-Antrieb, Massagemotor

Zur Ansteuerung der Prüfstände kommen einerseits ein Leistungsteil (Motorsteuerung) und andererseits ein Logikteil (Handscharter) zum Einsatz. Wobei anzumerken ist, dass die für die korrekte Steuerung notwendige Logik bereits in der Motorsteuerung integriert ist. Die Handscharter stellen lediglich das Bedienelement dar, das für die Automatisierung verwendet wird.

Elektronik – Folgende Steuerelektronik kommt bei den Prüfständen zum Einsatz (siehe Abbildung 16):

- Motorsteuerung: Die Prüfstände werden mit den von LOGICDATA entwickelten Motorsteuerungen betrieben. Für die Säulenprüfstände und Massagemotorenprüfstände sind das vorwiegend Steuerungen der Produktfamilie COMPACT und SMART. Für die Hebelprüfstände werden Steuerungen der Produktfamilie FLEX eingesetzt.
- Bedienelement: Für die Bedienung der Prüfstände werden für Dauerlaufversuche zwei Arten von Handschalter verwendet. Zum einen ist das ein digitaler, frei programmierbarer Handschalter, zum anderen ein analoger Automatik-Handschalter mit sehr eingeschränkter Funktionalität. Der digitale Handschalter bietet die wesentlichsten drei Modi. Diese sind manueller, zeitgesteuerter und positionsgesteuerter Betrieb.
- Betriebsmodi: Im manuellen Betrieb kann die Säule per Tastendruck einfach auf und abwärts bewegt werden. Der zeit- sowie der positionsgesteuerte Modus bieten jeweils 12 Speicherstellen. Jede Speicherstelle kann belegt werden mit einer Aufwärtsfahrt, Abwärtsfahrt oder einer Ruhepause. Im zeitgesteuerten Modus wird dazu noch die Dauer des einzelnen Vorgangs gespeichert (z.B. fahre fünf Sekunden aufwärts), im positionsgesteuerten Modus wird die angestrebte Position hinzugefügt (z.B. fahre abwärts auf Position 1). Die Programmierung eines Automatikzyklus anhand dieser 12 Speicherstellen gestaltet sich sehr zeitaufwändig. Der analoge Dauerlauf-Handschalter bietet ähnliche Funktionen. Verfahrszeiten sowie die Dauer von Pausen können jedoch nur mithilfe von analogen Drehreglern eingestellt werden.



Abbildung 16: Steuerelektronik, v. l. n. r. Motorsteuerungen, digitaler Handschalter, analoger Handschalter

Modell des Ist-Systems – Durch Modellbildung soll nun das Zusammenwirken oben beschriebener Elemente gezeigt werden. Entsprechend Kapitel 2.3.1 handelt es sich demnach um die Bildung eines Erklärungsmodells. Abbildung 17 zeigt, wie die einzelnen Komponenten miteinander in Beziehung stehen. Wie zu erkennen ist, wurden den einzelnen Beziehungen gewisse Attribute hinzugefügt, diese sind einerseits die Art der Flussgröße (z.B. Daten, Informationen, Energie) und andererseits die Wirkrichtung, welche durch Pfeile dargestellt wird.

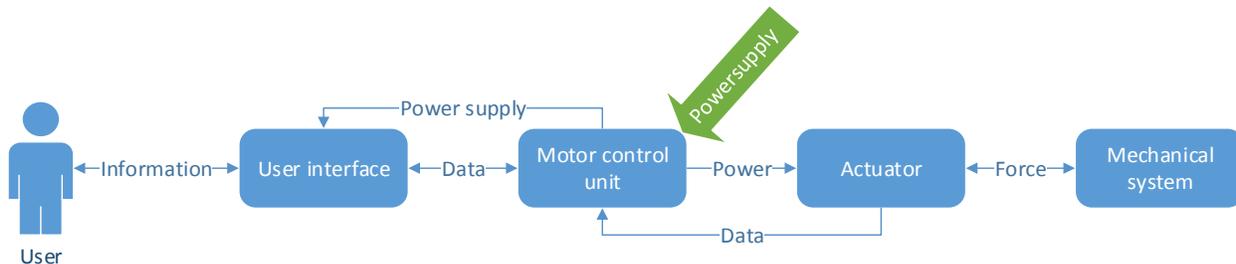


Abbildung 17: Zusammenwirken der einzelnen Prüfstandkomponenten

Nun zur Erklärung des in Abbildung 17 dargestellten Systemmodells:

- User interface: Das hier als Userinterface bezeichnete Element stellt den Handschalter dar. Über den Handschalter kann der Benutzer (*user*) den Aktuator (*actuator*) betätigen. Der Handschalter zeigt über das integrierte Display die Höheninformationen an, damit werden zwischen Handschalter und Benutzer Informationen ausgetauscht.
- Motor control unit: Der Handschalter betätigt natürlich nicht direkt den Aktuator. Diese Aufgabe wird von der Motorsteuerung (*motorcontrolunit*) übernommen. Die Versorgung des Handschalters (*powersupply*) wird ebenfalls von der Motorsteuerung übernommen. Der Handschalter gibt die vom Benutzer getätigten Eingaben an die Motorsteuerung weiter. Diese verarbeitet die empfangenen Daten des Handschalters und betätigt die Aktuatoren über die Leistungsausgänge (*power*). Der Aktuator liefert laufend Daten zu seiner aktuellen Position in Form von Drehimpulsen an die Motorsteuerung zurück, sodass diese ihre Regelaufgaben bewältigen kann.
- Actuator: Durch die ausgeübte Kraft (*force*) wird das mechanische System (*mechanical system*) vom Aktuator bewegt. Gemäß dem dritten newton'schen Axiom (*actio = reactio*) übt auch das mechanische System Kraft auf den Aktuator aus.

5.2.3 Gestaltungsbereich

Bevor der Gestaltungsbereich absteckt werden kann, soll sich hier zunächst näher mit einem der obigen Prüfstände auseinandergesetzt werden.

Mechatronisches System – Ziel der folgenden Betrachtungen ist es, herauszufinden, in wie weit es sich bei dem Gesamtsystem um ein mechatronisches System handelt. Dazu wird exemplarisch der Hebelprüfstand unter die Lupe genommen und seine Komponenten betrachtet. In Kapitel 0 wurde gezeigt, dass sich ein mechatronisches System durch drei Faktoren auszeichnet:

1. Ein mechatronisches System ist ein System bestehend aus mechanischen, elektrischen und informationsverarbeitenden Komponenten.
2. Die innere Struktur eines mechatronischen Systems weist eine geschlossene Wirkungskette auf in Form eines Regelkreises.
3. Das mechatronische System ist von integrativer Natur, d.h. die Funktionen der einzelnen Systemelemente sind voneinander abhängig.

In dieser Reihenfolge soll nun das bestehende System untersucht werden. Zunächst werden hierzu die drei grundlegenden Komponenten identifiziert:

- Mechanische Elemente: Beim Hebelprüfstand handelt es sich um eine Schweißkonstruktion. Die Hebelkonstruktion dient dazu, die durch eine Masse hervorgerufene Gewichtskraft zu übersetzen, um damit den 90-Grad-Antrieb möglichst stark zu belasten.
- Elektrische Komponente: Beim Aktuator handelt es sich in aller Regel um einen bürstenbehafteten Gleichstrommotor, welcher über einen Schneckentrieb eine Zahnstange (oder Ähnliches) antreibt. Versorgt wird der Elektromotor von der Motorsteuerung. Diese besteht einerseits aus einem Leistungsteil und andererseits aus einem Logikteil. Der Leistungsteil der Motorsteuerung besteht wiederum aus zwei weiteren Komponenten. Zum einen ist das ein Netzteil, welches dazu dient die Netzstromversorgung (230VAC bzw. 110VAC) in die benötigten Versorgungsspannungen (z.B. 30VDC Motorspannung) umzusetzen, zum anderen ist das eine Brückenschaltung (z.B. Vierquadrantensteller), die es u.a. ermöglicht, den Gleichstrommotor in beide Drehrichtungen zu betreiben.
- Informationsverarbeitende Komponente: Der Logikteil der Motorsteuerung übernimmt sämtliche steuerungs- und regelungstechnischen Aufgaben. Der Logikteil bietet eine Schnittstelle zum Userinterface (z.B. dem Handschalter), übernimmt die Positionserkennung, die Erkennung der

Endpositionen, eine Fehlererkennung und vieles mehr. Der Handschalter übernimmt die Benutzereingaben und gibt diese an die Motorsteuerung weiter. Außerdem zeigt er dem Benutzer Informationen wie Aktuatorposition oder Fehlercodes an.

Demnach handelt es sich also um ein Gesamtsystem bestehend aus mechanischen, elektrischen und informationsverarbeitenden Elementen, womit die erste wesentliche Bedingung erfüllt ist.

Die geschlossene Wirkungskette in einem mechatronischen System besteht in aller Regel aus dem mechanischen Grundsystem, Sensoren zur Erfassung der Messdaten, Prozessoren zur Datenverarbeitung und zur Bildung von Führungsgrößen sowie Aktoren, welche über Stellgrößen auf das mechanische Grundsystem rückwirken. Nach Abbildung 18 sind alle diese Komponenten im Gesamtsystem zu identifizieren.

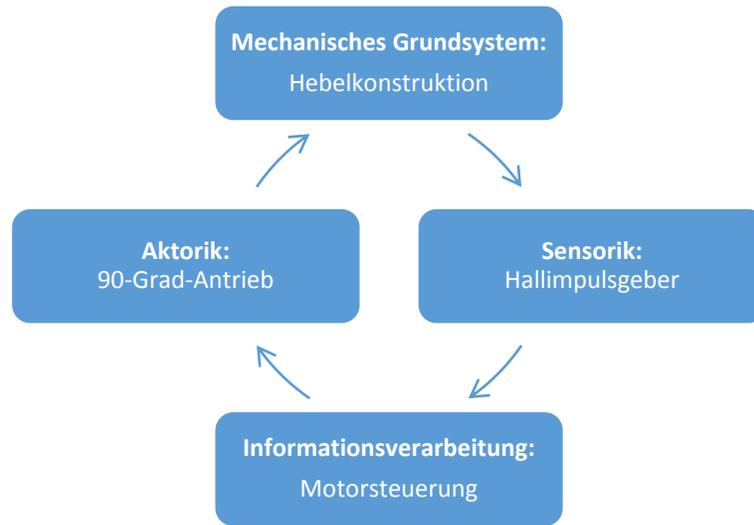


Abbildung 18: Geschlossene Wirkungskette innerhalb des Hebelprüfstandes

Die Hebelkonstruktion als mechanisches Grundsystem, die Linearaktuatoren als Aktorik, die Logikeinheit der Motorsteuerung als datenverarbeitendes Element sowie im Aktuator verbaute Hallimpulsgeber zur Positionserkennung als Sensorik. Die regelungstechnischen Aufgaben gestalten sich im betrachteten System als einfach wie z.B. das Anfahren vorprogrammierter Positionen über Fahrtrichtungserkennung und Positionsermittlung. Die geschlossene Wirkungskette lässt noch keinen Zweifel daran aufkommen, dass es sich bei dem Gesamtsystem um ein mechatronischen System handeln könnte.

Der letzte Aspekt, welcher betrachtet werden muss, ist die integrative Bauweise der mechanischen, elektrischen und informationsverarbeitenden Elemente. Dazu werden diese Elemente auf ihre Funktion untersucht und wie diese von den anderen Komponenten abhängt.

- Mechanische Funktion: Zweck der Hebelkonstruktion ist es, die von der Masse erzeugte Gewichtskraft zu übersetzen. Diese Funktion erfüllt die Konstruktion auch ohne das Vorhandensein der elektrischen und informationsverarbeitenden Komponenten.
- Elektrische Funktion: Der Aktuator hat die Aufgabe elektrischen Strom in mechanische Bewegung umzusetzen. Diese Aufgabe kann der Aktuator prinzipiell ohne die Hebelkonstruktion oder die Motorsteuerung erfüllen.
- Informationsverarbeitende Funktionen: Die Motorsteuerung ist ebenso nicht direkt gebunden an die anderen beiden Komponenten. So können verschiedenste Aktuatoren an die Steuerung angeschlossen oder unterschiedliche mechanische Systeme verwendet werden.

Dies ist natürlich eine sehr vereinfachte Betrachtung. Die Frage die sich dabei stellt, lautet: „Wann kann man wirklich von einer integrativen Bauweise sprechen?“. Was jedoch zum Ausdruck gebracht werden sollte, ist, dass es sich um einen modularen Aufbau handelt. Jede Komponente erfüllt ihre Aufgabe im Wesentlichen unabhängig von den anderen Systemkomponenten, weshalb es sich um ein System von Systemen handelt (siehe Kapitel 2.2.4).

Nachdem eben gestellte Frage geklärt werden muss, befindet sich das Gesamtsystem in einer Grauzone. Die Prüfstände fallen durchaus in den Fachbereich der Mechatronik, jedoch handelt es sich nicht zwingend um ein mechatronisches System. Dazu erfordert auch der Regelkreis kein besonderes regelungstechnisches Fachwissen. Das System kann als stabil betrachtet werden, womit auch kein Reglerentwurf nötig ist. Ein Entwurf eines Regelkreises ist auch nicht Ziel dieser Arbeit (vgl. Kapitel 4.3). Für die Lösung der Aufgabenstellung ist es auch nicht zielführend, ein technisches Modell des Gesamtsystems zu bilden.

Unabhängig davon können die Konzepte des Systems Engineering zur Anwendung kommen, um die Aufgabenstellung zielgerichtet zu lösen.

Gestaltungsbereich – Um den Gestaltungsbereich abzugrenzen soll das Gesamtsystem, unabhängig vorheriger Feststellungen, als mechatronisches System betrachtet werden. Anhand des allgemeinen Modells eines mechatronischen Systems (siehe Kapitel 4.2.2) werden dazu die einzelnen Komponenten identifiziert und separat abgegrenzt.

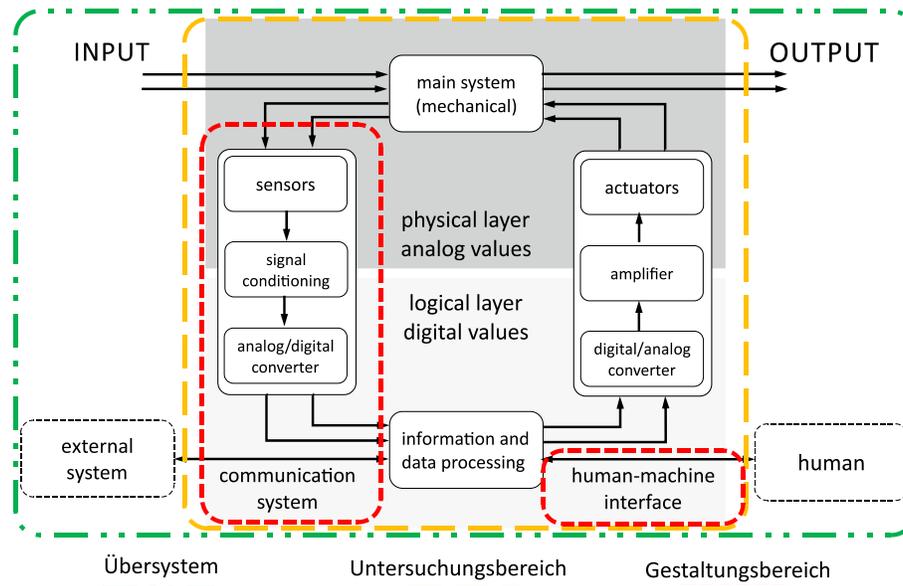


Abbildung 19: Abgrenzung des Gestaltungsbereichs am Modell eines mechatronischen Systems

- **Main system:** Beim mechanischen Hauptsystem (*main system*) handelt es sich um die Hebelkonstruktion, ein bislang etabliertes System, welches verändert werden kann, sofern dies erforderlich scheint. Um beispielsweise Zustandsinformationen des Gesamtsystems erfassen zu können, bedarf es zusätzlicher Sensorik. Müssen Sensoren an das mechanische System angebunden werden, sind Änderungen unweigerlich notwendig. Somit ist dieses System mitunter Teil des Gestaltungsbereichs, wenn auch nur begrenzt.
- **Sensors:** Sensoren (*sensors*) sind unter anderem der Hallimpulsgeber zur Positionsermittlung als Teil des Aktuators. Diese Sensoren können genutzt werden, geben jedoch nur wenig Informationen über das Gesamtsystem preis. Aus diesem Grund bedarf es zur Zustandsermittlung weiterer Sensorik. Hierbei handelt es sich um einen wesentlichen Teil des Gestaltungsbereichs. Einige Sensorik lässt sich auch in der Motorsteuerung finden.
- **Signal conditioning:** Die Signalkonditionierung (*signalconditioning*) wird oft direkt in Sensoren verbaut, teils ist jedoch eine externe Konditionierung nötig. Diese befindet sich dann entweder in

einem externen System oder ist in der Motorsteuerung integriert. Neue Sensoren bedürfen einer eigenen Konditionierung, deshalb ist dieser Teil ebenso wie die Sensoren wichtiger Bestandteil des Gestaltungsbereichs.

- A/D-converter: Zur Datenverarbeitung werden die von den Sensoren gelieferten Informationen in digitaler Form benötigt. In den seltensten Fällen liefern Sensoren digitale Informationen, darum bedarf es einem Analog-Digital-Wandler (*A/D-converter*), weshalb sie zum Gestaltungsbereich zählen. Die Impulssensoren am Aktuator liefern beispielsweise Digitale Informationen, in diesem Fall benötigt es beispielsweise Timing-Bausteine, um die Daten in verwertbare Information umwandeln zu können.
- Data processing: Zentrales System zur Datenverarbeitung (*data processing*) ist die Motorsteuerung. Sie nimmt die Eingangsdaten der Mensch-Maschine-Schnittstelle und anderer Eingänge auf und wandelt diese in Ausgangsdaten für den Aktuator. Hier findet auch die Verarbeitung der Sensorinformationen statt, diese können über Schnittstellen an andere Systeme der Umwelt (z.B. Mensch oder Umsystem) ausgegeben werden. Die Motorsteuerung kann, sofern notwendig, ersetzt werden, Veränderungen an der Motorsteuerung sind nur begrenzt möglich. Die Motorsteuerung zu ersetzen ist nicht empfehlenswert. Ansonsten bedarf es eines Systems, das die Eigenschaften und Funktionen der Motorsteuerung möglichst ähnlich nachbildet, um den Aktuator so realitätsnahe wie möglich zu betreiben. Die Motorsteuerung zählt, ähnlich wie das mechanische System, nur begrenzt zum Gestaltungsbereich.
- D/A-converter und amplifier: Der Digital-Analog-Wandler (*D/A-converter*) sowie der Verstärker (*amplifier*) befinden sich in der Motorsteuerung. Folglich sind diese Komponenten kein wesentlicher Teil des Gestaltungsbereichs, solange die Motorsteuerung nicht ersetzt wird.
- Actuator: Der Aktuator (*actuator*) ist das zu testende System und kann somit vollständig aus dem Gestaltungsbereich ausgeschlossen werden. Hier dürfen keine Veränderungen gemacht werden. Nichts desto trotz können weitere Aktoren in das System eingebunden werden, um auf das Gesamtsystem Einfluss nehmen zu können.
- Human-machine-Interface: Die Mensch-Maschine-Schnittstelle (*human-machine-interface*) ist wesentlicher Teil des Gestaltungsgebiets. Um an die Zustandsinformationen zu kommen oder einfache Veränderungen an dem Gesamtsystem machen zu können, bedarf es Verbesserungen an genau dieser Schnittstelle.

- Communication system: Das Kommunikationssystem (*communication system*) fällt ebenso in den Gestaltungsbereich. Über die Kommunikationsschnittstellen kann das mechatronische System mit anderen Systemen (z.B. Computersysteme) Daten austauschen.

Schlussfolgernd aus obigen Betrachtungen werden die Grenzen des Gestaltungsbereichs zunächst um die Sensorik inkl. Signalkonditionierung und Wandlung sowie Kommunikationssystem und Mensch-Maschine-Schnittstelle gelegt (siehe Abbildung 19).

5.2.4 Projektziele

Die aus den in Kapitel 5.2.1 identifizierten Problemen abgeleiteten Antworten auf die Frage: „Was wollen wir?“ lauten:

- Den Aufwand für Überwachungs- und Wartungstätigkeiten reduzieren.
- Eindeutige Zustandsdaten des Prüfstandes ermitteln und diese einfach zugänglich machen.
- Fehler und Defekte einfach und frühzeitig erkennbar machen.
- Stillstandzeiten reduzieren.
- Unnötige Wege reduzieren.
- Einfache Änderungen am Prüfprogramm ermöglichen.

Diese Antworten geben nur Auskunft darüber, was zu tun ist und nicht wie diese Ziele erreicht werden. Als Voraussetzung, um in weiterer Folge verschiedene Lösungsprinzipien zu erarbeiten, gilt es, die Fragen nach dem *Wie* zu beantworten. Nachstehend werden allgemeine Antworten gesucht, wie die Probleme gelöst werden können, um daraus konkrete Ziele abzuleiten.

Überwachungs- und Wartungstätigkeiten – Um den Aufwand für Überwachungs- und Wartungstätigkeiten zu reduzieren bedarf es einem System, das diese Aufgaben übernimmt oder das Gesamtsystem muss so verändert werden, dass dieses weder überwacht noch gewartet werden muss. Im betrachteten Beispiel eines Prüfstandes ist eine Wartungsfreiheit des Systems vollkommen unrealistisch, da gerade der Fehlerfall von Interesse für die Entwicklung ist. Somit gilt es ein System zu erstellen, das die für Überwachung- und Wartung nötigen Schritte übernimmt. Daraus ergibt sich die Fragestellung, welche Tätigkeiten übernommen werden müssen. Erst wenn diese Arbeiten festgestellt wurden, lässt sich ein Zielsystem entwerfen. Ziel der Lösungssuche ist demnach, zuallererst herauszufinden, welche Überwachungs- und Wartungstätigkeiten notwendig sind und wie diese von einem System übernommen werden können.

Zustandsdaten – Um eindeutige Zustandsdaten des Prüfstandes ermitteln und diese einfach zugänglich zu machen, muss geklärt werden, welche Informationen eindeutig den Zustand des Gesamtsystems repräsentieren. Damit klärt sich ebenfalls die Frage, was unter dem Begriff *Zustand* in diesem Beispiel verstanden wird. Ausgangspunkt dafür ist, dass die Zustände des Prüfstands im Wesentlichen durch messbare Größen erfasst werden können. Um einen einfachen Zugang zu den Zustandsinformationen zu ermöglichen, bedarf es im Wesentlichen einer Form von Mensch-Maschine-Schnittstelle. Die Messwerte müssen entsprechend aufbereitet werden, sodass erfasste Daten in verwertbare Informationen übergeführt werden können, um in weiterer Folge über den Prüfstands-Zustand Bescheid zu wissen. Folglich muss in der Lösungssuche eine Erhebung der messbaren Zustandsdaten, bzw. jener Daten die von Interesse für den Benutzer sind, erfolgen.

Stillstandzeiten – Stillstandzeiten entstehen an den Prüfständen vor allem durch Fehler oder Defekte. Indem Fehler frühzeitig festgestellt und behoben werden, lassen sich Stillstandzeiten vermeiden. Um Fehler und Defekte einfach und frühzeitig erkennbar zu machen, müssen diese vom System identifiziert werden. Eine Fehlererkennung setzt voraus, dass man über potentielle Fehler Bescheid weiß, welche Eigenschaften diese aufweisen und wodurch diese sich vom Normalzustand unterscheiden. Erst wenn diese Unterschiede zum Normalzustand über z.B. geeignete Sensorik erfasst werden können, ist eine Fehlererkennung oder gar eine Früherkennung möglich.

Wege – Unnötige Wege können im Wesentlichen auf zwei Arten reduziert werden. So können entweder Wege verkürzt werden oder die Häufigkeit, wie oft derselbe Weg gegangen wird, reduziert werden.

Prüfprogramm – Änderungen am Prüfprogramm erfolgen über die Mensch-Maschine-Schnittstelle. Einfache Änderungen am Prüfprogramm können somit durch eine Verbesserung der Mensch-Maschine-Schnittstelle ermöglicht werden.

Die folgende Tabelle 7 beinhaltet mit dem Auftraggeber vereinbarte Ziele. Wie eindeutig zu erkennen ist handelt es sich dabei vorrangig um nicht quantifizierbare Ziele, was die zukünftigen Projektentscheidungen erschwert.

Tabelle 7: Projektziele

	Ziele
Muss	<ul style="list-style-type: none"> • Maximalkosten für ein Testgerät darf 250 Euro pro System nicht überschreiten, dies beinhaltet nicht Kosten für Prototypenentwicklung (Projektbudget). Grund dafür ist die relativ große Anzahl an Prüfständen, die (im Erfolgsfall) mit dem System ausgestattet werden sollen. • Einfache Bedienbarkeit! Dies fällt unter die Kategorie Mussziel, da eine einfache Bedienung Voraussetzung dafür ist, dass die Mitarbeiter das Gerät einsetzen und nicht auf bewährte Methoden zurückgreifen. • Es muss eine Verbesserung gegenüber der herkömmlichen Testmethode bieten. • Genauigkeiten der gemessenen Daten müssen Aussagen über die Zustände ermöglichen. • Einfachere Überwachung der Prüfstände, verbesserte Automatisierung. • Fehlererkennung, wobei erkannte Fehler für den/die Mitarbeiter angezeigt werden müssen. • Einhaltung der Projektdauer, so soll am Ende des Projekts ein funktionstüchtiger Prototyp zur Verfügung stehen.
Soll	<ul style="list-style-type: none"> • Zugriff auf die gemessenen Daten direkt vom Büro-Arbeitsplatz des Mitarbeiters. • Bedienung der Prüfstände vom Büro-Arbeitsplatz des Prüfstandmitarbeiters. • Mitarbeiter über Fehlerfall automatisch informieren. • Gute Messgenauigkeit. • Fehler die durch den Menschen verursacht werden, sollen verringert werden. • Prüfabläufe sollen beschleunigt werden. • Erinnerung an Wartungsaufgaben (Wartungsintervall).
Wunsch	<ul style="list-style-type: none"> • Automatische Datenarchivierung in einer Datenbank (<i>data logging</i>). • Optische Fernüberwachung (z.B. Kameraüberwachung). • Geräuschemessungen. • Automatische Testprotokollierung. • Sicherheitsfunktionen um Fremdzugriffe auf die Prüfstände zu verhindern. • Graphische Darstellung des Prüfstandes mit Bewegungsdarstellung. • Lifebeobachtung der Prüfstände. • Fehlerfrüherkennung vor Eintritt eines Schadens.

5.2.5 Lösungsprinzipien

Bevor mit der Suche nach Lösungsprinzipien begonnen wird, müssen zunächst die Überwachungs- und Wartungstätigkeiten näher betrachtet werden.

Überwachungs- und Wartungstätigkeiten – Überwacht wird in erster Linie die Funktionstüchtigkeit der Aktuatoren, ob es zwischenzeitlich zu Ausfällen gekommen ist oder ob die geforderten Zyklen erreicht wurden. Desweiteren werden sporadisch Messungen an den elektrischen Komponenten durchgeführt. Dabei handelt es sich um Leistungsinformationen, ob die gesetzten Grenzwerte eingehalten werden. Die folgenden Messwerte werden dafür aufgezeichnet:

1. Motorstrom
2. Motorspannung
3. Motordrehzahl
4. Aktuatorposition

Diese Messwerte können unter anderem als die Zustandsinformationen gesehen werden. Wartungstätigkeiten befassen sich vorwiegend mit der Wiederherstellung von defekten Prüfständen. In den meisten Fällen handelt es sich um mechanische Defekte, welche behoben werden müssen. Mechanische Defekte bedürfen meist einer Reparatur und damit entsprechendem Fachpersonal. Wartungstätigkeiten dieser Art können kaum oder gar nicht automatisiert werden und befinden sich dazu außerhalb des gesetzten Gestaltungsbereichs.

Defekte dieser Art könnten jedoch festgestellt und dem Mitarbeiter gemeldet werden. Ebenso verhält es sich mit elektrischen Defekten am Prüfstand. Andere Wartungstätigkeiten sind Aktualisierungen der Firmware von Motorsteuerungen, oder das Ändern des Prüfablaufs. Fernwartungen am informationsverarbeitenden Teil des Systems wären denkbar und befinden auch sich im Rahmen des Gestaltungsbereichs.

Zustandsinformationen – Wie bereits angemerkt, sollen die Zustände des Prüfstands ermittelt und den Prüfstandmitarbeitern zugänglich gemacht werden. Dazu müssen jedoch zunächst die Zustände ermittelt werden, in welchen sich der Prüfstand befinden kann. Erst dann kann eine Aussage darüber gemacht werden, wie diese Zustände festgestellt werden können. Aus diesem Grund wurden denkbar mögliche Zustände in einem Brainstorming erfasst und in Abbildung 20 dargestellt.

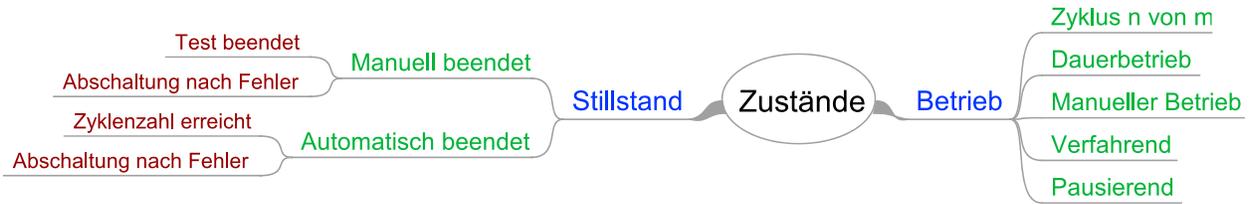


Abbildung 20: Mind-Map zu denkbaren Zuständen des Prüfstandes

Dabei wurde festgestellt, dass diese Zustände in zwei Kategorien eingeteilt werden können. Die erste Kategorie sind alle Zustände des ordnungsgemäßen Betriebs, die restlichen sind jene Zustände nach Eintreten eines Fehlers, womit man bei der Problematik der Fehlererkennung angelangt wäre.

Fehlererkennung – Um eine funktionierende Fehlererkennung zu ermöglichen, müssen vorerst die möglichen Fehler bekannt sein. Die häufigsten Fehlerfälle sind mechanischer Natur wie ein Getriebeschaden oder gebrochene Anbindungselemente. Ähnlich der *Fehlermöglichkeits- und Einflussanalyse (FMEA)* können potentielle Fehlerfälle im Voraus erkannt werden. In Abbildung 21 wurde versucht, ein Überblick über mögliche Fehler oder Defekte zu erstellen.

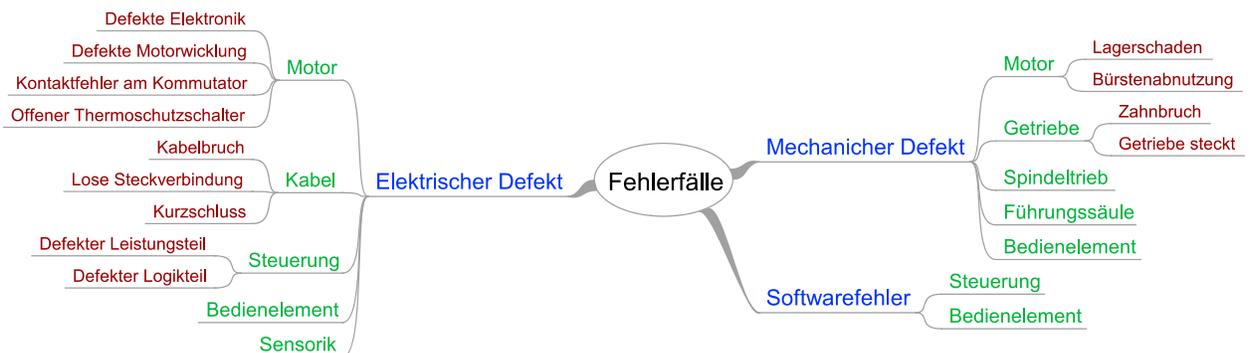


Abbildung 21: Mind-Map möglicher Fehlerfälle bzw. der Lage möglicher Fehler

Nun gilt es herauszufinden, wie einige dieser Fehler erkannt bzw. messtechnisch erfasst werden können. Um Fehler messtechnisch zu erkennen, wird entsprechende Sensorik und eine Auswertung der Sensordaten benötigt. Damit stellt sich bereits die Frage nach der Art der zu verwendenden Sensorik, um bekannte Fehler diagnostizieren zu können. Die Basis für das weitere Vorgehen bilden die erfassbaren Messdaten. Im Zuge eines Brainstorming-Prozesses wurden nun alle möglichen Ideen in Bezug auf messbare Daten gesammelt und entsprechend in Abbildung 22 dargestellt.

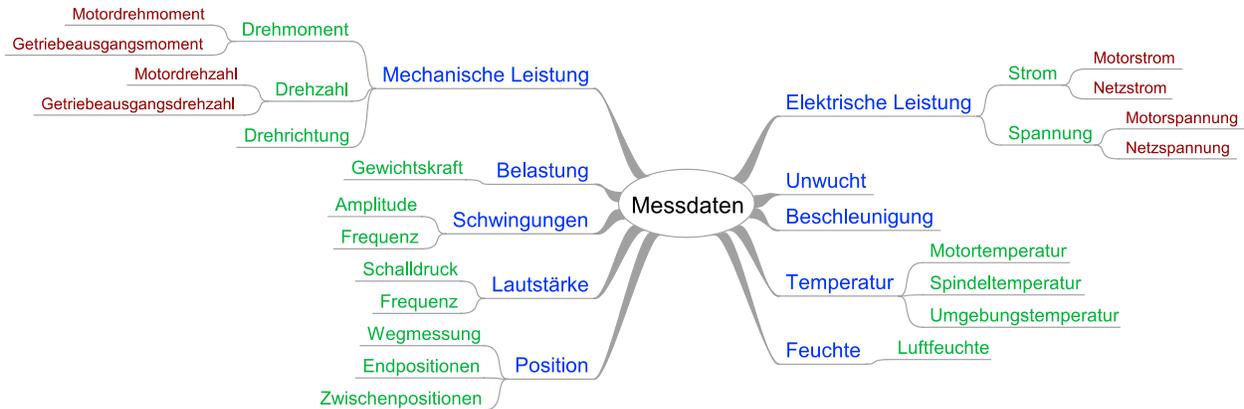


Abbildung 22: Mind-Map erfassbarer Messdaten

Im nächsten Schritt müssen Kausalzusammenhänge zwischen einem Fehler, als Ursache, und den Auswirkungen in den Messdaten, als Wirkung, gebildet werden. In Tabelle 8 werden einige Beispiele solcher Kausalzusammenhänge gezeigt.

Tabelle 8: Beispiele für Kausalzusammenhänge zwischen Fehler und Messdaten

Ursache	Wirkung
Getriebeschaden (Zahnbruch)	<ul style="list-style-type: none"> • Konstante Position trotz Motordrehzahl
Motor blockiert	<ul style="list-style-type: none"> • Keine Motordrehzahl • Erhöhte Stromaufnahme
Kabelbruch	<ul style="list-style-type: none"> • Kein Strom trotz angelegter Spannung
Kurzschluss	<ul style="list-style-type: none"> • Überstrom (bzw. Überstromabschaltung)

Trotz ausgiebiger Sensorik und der Erfassung verschiedenster Sensordaten ist jedoch eine eindeutige Fehlererkennung nicht immer einfach. So kann ein und dieselbe Wirkung verschiedene Ursachen haben. Der Kausalzusammenhang zwischen Ursache und Wirkung ist in vielen Fällen nicht anhand der erfassten Daten festzustellen. Nachstehendes Beispiel soll diese Problematik kurz anschaulich machen:

Die häufigsten Fehler, wie z.B. Getriebedefekte, äußern sich dadurch, dass trotz bestromtem Motor der Aktuator nicht verfährt. Solche Defekte könnten z.B. diagnostiziert werden, wenn externe Positionssensoren die tatsächliche Aktuatorposition ermitteln. Ändert sich die Aktuatorposition trotz einer erkennbarer Motordrehzahl nicht, so kann auf einen mechanischen Defekt nach der Motorwelle geschlossen werden. Dass es sich in diesem Fall tatsächlich um einen Getriebeschaden handelt, kann

jedoch nur vermutet werden. Ebenso könnten sich Verschraubungen gelöst haben oder die Sensoren selbst defekt sein.

Eine exakte Erkennung der Ursache eines Fehlers ist auch nicht erforderlich. Selbst wenn nicht direkt auf die Fehlerquelle geschlossen werden kann, können die Messdaten helfen die Ursache einzugrenzen. Letztlich soll im Falle eines mechanischen Defekts auch lediglich ein Mitarbeiter darüber informiert werden.

Schließlich wurden die sechs verschiedenen Messdaten mit dem Auftraggeber vereinbart, welche vom System aufgezeichnet werden sollen. Diese sind:

- | | |
|------------------|--------------------------|
| 1. Motorstrom | 4. Aktuatorposition |
| 2. Motorspannung | 5. Temperatur |
| 3. Motordrehzahl | 6. Externer Zyklenzähler |

Dabei gilt es anzumerken, dass die Auswahl sehr stark dadurch eingeschränkt wurde, welche Sensoren einfach an die Prüfstände angebunden werden können. Benutzerfreundlichkeit ist ein Teilaspekt dieser Entscheidung.

Lösungsprinzipien erarbeiten – Obige Betrachtungen zeigen, dass es eines Systems bedarf, welches Messdaten erfassen kann und über Ausgänge verfügt, über welche die Steuerungsaufgaben bzw. Regelungsaufgaben übernommen werden können. Diese Anforderungen sowie der Rahmen des Gestaltungsbereichs begrenzen die Lösungsprinzipien auf ein elektronisches, informationsverarbeitendes System. Dieser Zusammenhang legt auch gleich nahe, das System anhand der Aspekte aus Kapitel 2.2.1 zu klassifizieren. Entsprechend der wesentlichsten Aspekte handelt es sich bei dem Zielsystem um ein reales, offenes, künstliches, dynamisches, komplexes, zweckorientiertes technisches System.

Für die technische Umsetzung ist geeignete Hardware zu finden oder zu entwickeln, sowie eine passende Anwendersoftware zu entwerfen. Primäre Idee ist es, die Handschalter der aktuellen Prüfstände zu ersetzen aber die firmeneigene Motorsteuerung als Leistungsteil beizubehalten.

Die von LOGICDATA entwickelten Motorsteuerungen bieten eine Reihe von Vorteilen, diese sind unter anderem:

- Das Testsystem spiegelt den realen Einsatz wider, in dem ebenfalls die Motorsteuerungen verwendet werden. Einflussfaktoren der Motorsteuerung werden dadurch berücksichtigt.
- Einfache Parametrierung der Steuerung ermöglicht eine schnelle Anpassung an die Prüflinge.
- Steuerungen bieten viele Sicherheitsfunktionen wie beispielsweise Softwareschranken für Endpositionen, Referenzfahrten, Drehzahlregelung, Strombegrenzungen und vieles mehr.
- Dem Personal ist der Umgang mit den Steuerungen bekannt.
- Sie sind kostengünstig da sie bereits im Unternehmen vorhanden sind.
- Die Motorsteuerungen werden ohne einen Mehraufwand auf ihren Dauereinsatz getestet.

Ein Ersatz der Handschalter scheint aus diesem Grund die einzige, in vorgegebener Zeit, vernünftigt realisierbare Lösung. Nachdem die verschiedenen Typen von Handschaltern auch zwischen den einzelnen Motorsteuerungen (COMPACT, SMART, FLEX) nahezu vollständig kompatibel sind und somit auch mit den verschiedenen Aktuatoren, führt dieser Lösungsansatz auch zu einer flexiblen Lösung für nahezu alle Prüfstände.

Als Hardware für das Userinterface wählen wurde sich vorab für den Bürorechner entschieden, der jedem Mitarbeiter zur Verfügung steht. Auch andere Benutzerschnittstellen wären denkbar wie Mobiltelefone, ein Terminalrechner oder Ähnliches. Aber auch mit diesen Vorgaben sind verschiedene Lösungsprinzipien möglich.

Bei der Bildung der folgenden Lösungsprinzipien sollen die Handlungsprinzipien (siehe Kapitel 3.1.4) *vom Groben ins Detail* und *Variantenbildung* Anwendung finden. Zuerst wird versucht, den Betrachtungshorizont zu erweitern, woraus sich drei verschiedene Lösungsprinzipien ergaben (siehe Abbildung 23).

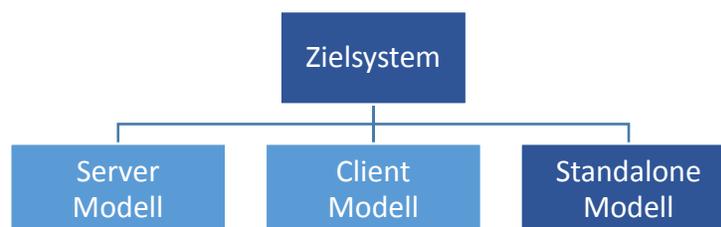


Abbildung 23: Drei Modelle als Varianten von Lösungsprinzipien

Folgende Ideen stecken hinter den einzelnen Lösungsprinzipien:

- Server-Modell: Idee des Server-Modells ist eine zentrale Steuereinheit für alle Prüfstände. Messgeräte erfassen die Sensordaten aller Prüfstände und übertragen diese direkt an die zentrale Steuereinheit (*server*). Der Server übernimmt die Datenverarbeitung und Datenaufbereitung und auch die Datensicherung ebenso wie alle Steuerungs- und Regelungsaufgaben. Steuerungsbefehle werden über ein geeignetes Bussystem an die Motorsteuerung übertragen. Zur Visualisierung der Daten kann vom Bürorechner auf den Server zugegriffen werden und die erfassten Daten aller Prüfstände ausgelesen werden.
- Client-Modell: Das Client-Modell ist eine sehr theoretische Möglichkeit die Prüfstände zu steuern und zu überwachen. Es erscheint in der Realität jedoch nicht sinnvoll anwendbar. Bei diesem Modell sollen die Aufgaben des Servers aus dem Server-Modell auf den Anwender-PC (*client*) übertragen werden. Der Client übernimmt alle Datenerfassungs- und Steuerungsaufgaben.
- Standalone-Modell: Die Idee hinter dem Standalone-System geht weg von einem zentralen Server hin zu einzelnen, unabhängigen Kleinsystemen für jeden Prüfstand. Das heißt, jeder Prüfstand wird mit einer eigenen Steuereinheit ausgestattet. Der Anwender hat somit separaten Zugriff auf jeden Prüfstand. Die dezentrale Steuereinheit übernimmt alle messtechnischen Tätigkeiten wie auch Steuerungs- und Regelungsaufgaben. Dieses Kleinsystem kann von einem Client über ein geeignetes Bussystem erreicht und parametrierbar werden. Das Standalone-System ist anschließend für die Steuerung und Überwachung des Prüfstands zuständig.

Der Vergleich von Vor- und Nachteilen von Server- und Standalone-Modell sollte die Entscheidung für ein Lösungsprinzip vereinfachen.

Tabelle 9: Vor- und Nachteile des Server-Modells

Vorteile	Nachteile
<ul style="list-style-type: none"> • Kostengünstig durch lediglich eine zentrale Einheit • Einfache Datenverarbeitung, -aufbereitung und -sicherung. • Übersicht über alle Prüfstände einfach realisierbar. 	<ul style="list-style-type: none"> • Externes Messgerät erforderlich zur Erfassung der Sensordaten. • Starke Buslast durch Übertragung der Messdaten und Steuerdaten zwischen Server und Prüfstandgeräten. • Örtliche Gebundenheit der Prüfstände. • Ausfälle oder Fehler des Servers betreffen alle Prüfstände. • Geringe Flexibilität. • Probleme bei Zugriffen mehrerer Benutzer

Tabelle 10: Vor- und Nachteile des Standalone-Modells

Vorteile	Nachteile
<ul style="list-style-type: none"> • Dezentralisierung. • Keine örtliche Gebundenheit. • Höhere Flexibilität. • Buslast wird reduziert. • Ein Ausfall wirkt sich lediglich auf einen Prüfstand aus. • Einfache Ein- und Ausgliederung einzelner Prüfstände (Flexibilität) 	<ul style="list-style-type: none"> • Kostspielig da ein System für jeden Prüfstand angeschafft werden muss. • Schwierigere Umsetzung einer zentralen Datensicherung

Prinzipiell können alle Modelle die gestellten Anforderungen erfüllen. Jedes Modell verfügt über gewisse Vor- aber auch Nachteile, welche jedoch keine quantifizierbaren Vergleichswerte bieten. Aufgrund der Dezentralisierung und der damit verbundenen geringeren Risikos eines Ausfalls wurde sich für das Standalone-Modell entschieden. Um jedoch die Machbarkeit zu überprüfen, bedarf es konkreterer Lösungsprinzipien. Dazu wurden entsprechend Abbildung 24 vier Detailvarianten des gewählten Lösungsprinzips erarbeitet.

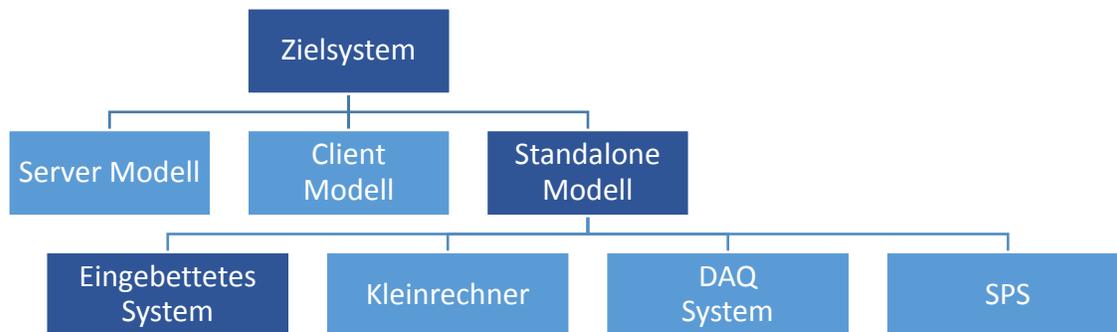


Abbildung 24: Variantenbildung und Detaillierung des gewählten Lösungsprinzips

Die folgenden Ideen verbergen sich hinter den einzelnen detaillierteren Lösungsprinzipien:

- Eingebettetes System: Bei der Variante *Eingebettetes System* (engl. *embedded system*) handelt es sich um eine Rechneinheit (z.B. ein Mikrocontroller), die in ein peripheres System eingebettet wird. Dieser Rechner übernimmt alle überwachenden, steuernden und regelnden Aufgaben sowie die Messdatenerfassung und -aufbereitung. Mikrocontroller verfügen bereits über eine breite Anzahl an digitalen und analogen Ein- und Ausgängen, welche für die Sensorauswertung genutzt werden können. Diese Variante bietet hohe Flexibilität in der Gestaltung.

- Kleinrechner: Idee hinter dieser Variante wäre die Verwendung eines einfachen Computers und eines Messgeräts. Über eine digitale Schnittstelle (z.B. USB, RS232) werden die Messdaten an den Rechner übertragen. Dieser übernimmt die Datenauswertung und die Steuerung des Prüfstands. Die Steuerinformationen werden an die Motorsteuerung übertragen, diese führt die gewünschten Operationen durch.
- DAQ-System: Viele Hersteller von Messgeräten bieten so genannte standalone DAQ-Systeme (*Data Acquisition Systeme*) an. Dabei handelt es sich meist auch um Eingebettete Systeme, welche über eine Vielzahl hochwertiger Schnittstellen sowie über verschiedene bereits integrierte Messverstärker und Bussysteme verfügen. Eine breite Palette an Sensoren kann an die Systeme angeschlossen und ausgewertet werden.
- SPS: Speicherprogrammierbare Steuerungen (SPS) sind Standard für Automatisierungsaufgaben. Dabei handelt es sich in aller Regel auch um Mikrocontrollersysteme, die zur Steuerung und Regelung von Maschinen und Anlagen eingesetzt werden. Im Gegensatz zu DAQ Systemen besitzen diese wenige, einfache digitale aber auch analoge Schnittstellen. Speicherprogrammierbare Steuerungen werden anwendungsspezifisch programmiert, wodurch sie eine Vielzahl an Aufgaben erfüllen können.

Für die gebildeten Varianten soll nun die Machbarkeit überprüft werden, welche in weiterer Folge auch Entscheidungsgrundlage für eine dieser Varianten sein soll.

5.2.6 Machbarkeit

Die Machbarkeit wird im Wesentlichen bestimmt durch die Einschränkungen in Kosten und der zur Verfügung stehenden Zeit. Zweifel an einer technologischen Machbarkeit bestehen aus Erfahrung keine, insbesondere da die Ziele sehr allgemein formuliert wurden. Aus diesem Grund wurde der Schwerpunkt hier auf die Kosten gelegt. Um eine Übersicht über die Kosten zu ermöglichen, wurden diese Varianten in Hinblick auf die am Markt angebotenen Produktlösungen betrachtet, ohne jedoch eine weitere Detaillierung durchzuführen.

Tabelle 11: Ungefähre Preise von am Markt erhältlichen Produkten

Variante	Beispiel Hardware	Preis (Feb. 2014)
Eingebettetes System	Entwicklungsboard ATEVK1100 AVR32 UC3A ARDUINO ETHERNET Board WizNet W5100 Mikrocontrollerboard AVR-NET-IO von Pollin	ca. 200 Euro ca. 50 Euro ca. 30 Euro
Kleinrechner	Linux Entwicklungsboard + 32MB ARM9 Modul RS Raspberry Pi Modell B Industrie-PCs (Siemens, Beckhoff, B&R)	ca. 300 Euro ca. 40 Euro ab ca. 500 Euro
DAQ-System	NICAI HTTP BOX+ National Instruments NI CompactDAQ-Systeme Agilent U2300A Series (z.B. U2331A)	ca. 550 Euro ab 342 Euro ca. 1800 Euro
SPS	Siemens Kleinststeuerung (z.B. LOGO 12/24 RCE) Siemens SIMATIC (z.B. S7-1200 inkl. Zubehör)	ca. 220 Euro ca. 400 Euro

Die in Tabelle 11 angegebenen Preise sind Richtwerte der Hersteller bzw. unterschiedlichen Händlern entnommen und sollen lediglich als Orientierungshilfe dienen. Benötigte periphere Geräte, wie zusätzliche Module oder Sensoren, sind dabei noch nicht enthalten. Außerdem wurde die angeführte Hardware nicht in Anbetracht aller Anforderungen ausgewählt. Dennoch kann anhand dieser Übersicht eine Abschätzung voraussichtlicher Kosten erfolgen.

5.2.7 Projektentscheidung

In Einvernahme mit dem Auftraggeber wurde die Fortsetzung des Projekts vereinbart. Dabei soll sich auf die Entwicklung eines eingebetteten, mikrocontrollerbasierenden Systems konzentriert werden.

5.3 Hauptstudie

Ausgehend vom gewählten Lösungsprinzip eines eingebetteten Systems auf Basis des Standalone-Modells soll nun in der Hauptstudie ein Gesamtkonzept erarbeitet werden. Dabei sind die Aufgaben der Hauptstudie neben dem Zusammenstellen des Projektteams und der Festlegung des Budgets auch die Gliederung des Projekts in Teilprojekte, welche auf die Projektmitglieder verteilt werden können.

Die folgenden Aufgaben sollen nun in der Hauptstudie durchgeführt werden:

1. Projektteam zusammenstellen
2. Projektbudget festlegen
3. Systemarchitektur auswählen
4. Gesamtkonzept erarbeiten
5. Projekt in Teilprojekte gliedern

5.3.1 Projektteam und Budget

Projektteam – Für dieses Fallbeispiel im Zuge der Diplomarbeit wurde kein eigenes Projektteam zusammengestellt. Da es sich um ein fachübergreifendes Projekt handelt, musste im Zuge der Arbeit das notwendige Fachwissen separat erarbeitet werden. Um gewisse Vorkenntnisse nutzen zu können, wurden Projektentscheidungen in Hinblick auf das vorhandene Fachwissen getroffen, entgegen eventueller Systemvorteile. Neben den betreuenden Personen standen sämtliche Mitarbeiter der Firma LOGICDATA bei Bedarf zur Verfügung.

Projektbudget – Im Wesentlichen setzt sich das Projektbudget aus kalkulierten Personalkosten für interne und externe Projektmitarbeiter, die benötigten Ressourcen wie Entwicklungsmaterial oder Fertigungskosten zusammen. Das tatsächlich kalkulierte Projektbudget kann hier jedoch nicht angeführt werden.

5.3.2 Systemarchitektur

In der Vorstudie wurde die Entscheidung zugunsten eines eingebetteten Systems getroffen. Um die Vorteile eines eingebetteten Systems nutzen zu können, zielt diese Variante auf eine selbstentwickelte Hardware ab. Das heißt, es müssen die für den Anwendungsfall benötigten peripheren Komponenten gesucht und daraus ein integriertes System entwickelt werden. Die Peripherie ist entscheidend für die Wahl der Systemarchitektur.

Das Festlegen der Systemarchitektur ist Hauptaufgabe des Systemarchitekten. Wie bereits in Kapitel 3.2.2 festgestellt wurde, ist die Zusammenarbeit mit dem Auftraggeber wesentlich beim Entwurf der Systemarchitektur. Dabei sollen die bereits in der Vorstudie ermittelten Projektziele bzw. Problemstellungen noch näher konkretisiert werden. Die Bildung von Modellen kann hier helfen abstrakte Vorstellungen, welche schwer zu formulieren sind, für die Beteiligten klar darzustellen.

Für das Fallbeispiel wurden dazu einfache Modellierungstechniken im Gegensatz zu den in Kapitel 2.3.3 gezeigten Modellierungssprachen gewählt.

Architekturmodell – Das in Abbildung 25 dargestellte Modell wurde gemeinsam mit dem Auftraggeber erarbeitet. Dabei handelt es sich um eine Vorstellung dessen, wie das System in seine Umgebung integriert werden kann. Es zeigt, welche allgemeinen Schnittstellen zwischen System und Umgebung zu identifizieren sind und liefert eine Idee davon, welche Aufgaben das System zu übernehmen hat.

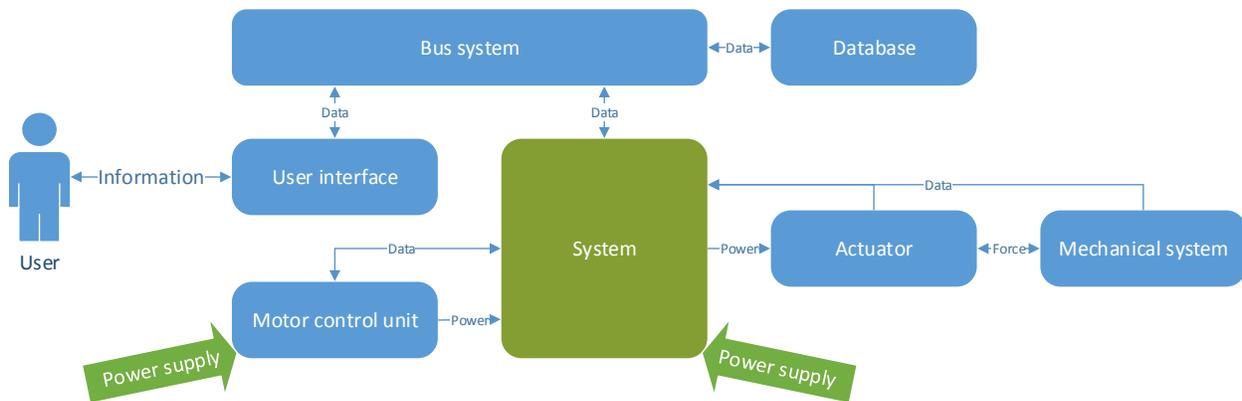


Abbildung 25: Modell des Zielsystems (Black-Box) und seiner Umgebung

Abbildung 25 zeigt die folgenden Umsysteme, welche mit dem Zielsystem interagieren:

- **User interface:** Der Benutzer (*user*) des Systems ist im Normalfall jene Person, die den Prüfstand betreut. Über eine Benutzerschnittstelle (*user interface*) hat diese auf den Prüfstand Zugriff. Die Datenübertragung von System zur Benutzerschnittstelle verläuft über ein geeignetes Bussystem (*bus system*).
- **System:** Das zu entwickelnde Zielsystem (*system*) selbst ist verbindendes Element zwischen Motorsteuerung (*motor control*) und Aktuator (*actuator*). Dabei werden alle Steuerbefehle vom Zielsystem an die Motorsteuerung gesendet. Neben Steuerbefehlen können auch Fehlerinformationen zwischen Motorsteuerung und System ausgetauscht werden.
- **Motor control unit:** Je nach Steuerbefehl versucht die Motorsteuerung den Aktuator zu betätigen. Der von der Motorsteuerung ausgegebene Motorstrom (*power*) wird zu Messzwecken über das Zielsystem geleitet.
- **Actuator:** Folglich übt der Aktuator entsprechend Kraft (*force*) auf das mechanische System aus. Aktuator und mechanisches System liefern Messwerte (*data*) an das System zurück. Die dazu nötige Sensorik ist entweder integrierter Teil des Aktuators oder Teil des Zielsystems.
- **Power supply:** Die Motorsteuerung wird mit Netzstrom (*power supply*) versorgt und könnte, abhängig von der notwendigen Leistung, auch das System versorgen. Sollte diese Versorgung nicht ausreichend sein, so wurde in diesem Modell auch schon eine externe Stromversorgung (*external power supply*) für das System angedacht.

Bevor ein Blick in die Systemstruktur gemacht werden kann, sind die Umsysteme näher zu betrachten. Diese bestimmen wesentlich die Schnittstellen. An der Motorsteuerung, dem Aktuator sowie dem mechanischen System sollen zunächst keine Veränderungen gemacht werden. Das bedeutet, die

Schnittstellen zu diesen Umsystemen sind definiert. Jedoch das verwendete Bussystem zur Kommunikation mit der Benutzerschnittstelle muss noch festgelegt werden.

Bussystem – Um ein geeignetes Bussystem für das Fallbeispiel zu finden wurde der Betrachtungsbereich zunächst auf Feldbusse aus der Automatisierungstechnik gelegt. In diesem Fachbereich werden so genannte Feldbusse eingesetzt, welche zur Anbindung von Sensorik und Aktorik an Steuergeräten dienen. Industrielle Feldbuslösungen zeichnen sich durch hohe Ausfallsicherheit und Robustheit oder geringe Anfälligkeit auf Störgrößen aus. Sie sind für den so genannten industriellen Feldeinsatz konzipiert und somit gegen einige Umwelteinflüsse (siehe Abbildung 26) gewappnet.

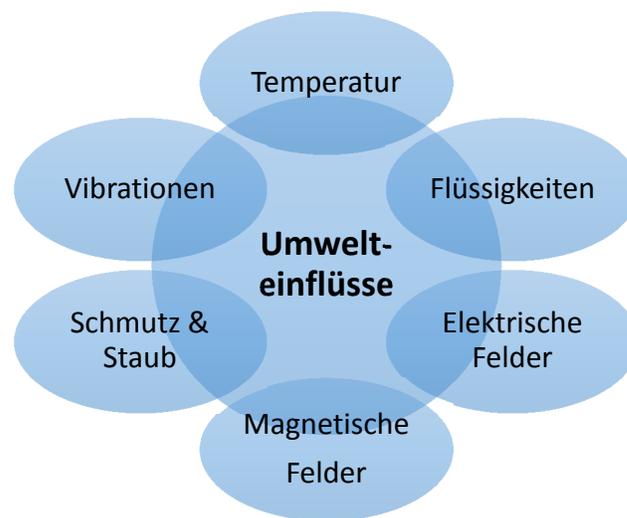


Abbildung 26: Auswahl an Umwelteinflüssen in industrieller Umgebung

Im Gegensatz zu Büro- oder Laborbedingungen herrschen in industriellem Umfeld besondere Umgebungsbedingungen. Dabei kann es sich um elektrische, mechanische, thermische oder sonstige Störgrößen handeln, welche eine einwandfreie Kommunikation verhindern.

Vertikale Integration: Im Bereich der Automatisierungstechnik tendieren moderne Anlagen zu integrativen Netzwerklösungen, welche vom Bürobereich (Managementebene) bis zu Teilsystemen industrieller Anlagen (Feldebene) reichen. Hierbei wird von *vertikaler Integration der Automatisierungstechnik* gesprochen. Abbildung 27 zeigt die unterschiedenen Ebenen, die in der Ebene anzutreffenden Systeme und die dort eingesetzte Bustechnologie.

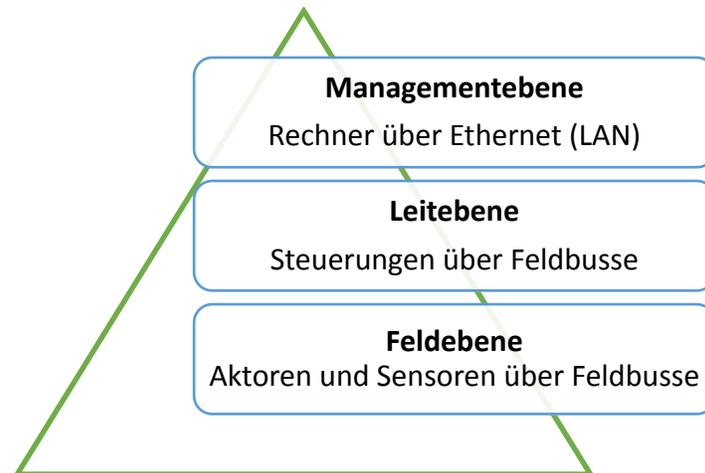


Abbildung 27: Vertikale Integration in der Automatisierungstechnik

Echtzeitethernet: Die für die vertikale Integration eingesetzten Feldbusstandards entwickeln sich zunehmend hin zu so genanntem Echtzeitethernet oder auch als Industrielles-Ethernet bezeichnet. Diese Strömung ist vorrangig damit begründet, dass Ethernet eine der in der EDV wohl am weitesten verbreitete Kommunikationstechnologie darstellt. Dabei ist der Standard vom Übertragungsmedium (Kuper, Glasfaser, Funk) unabhängig, bietet hohe Übertragungsraten und gute Ausfallsicherheit. Im Laufe der Zeit haben sich eine Reihe verschiedener industrieller Ethernet-Standards mit unterschiedlichen Schwerpunkten auf verschiedene industrielle Spezifikationen entwickelt. Bislang bestehen wenige Ansätze für einheitlich genormte Echtzeitethernet-Standards.

Entscheidung: Für den Anwendungsfall ist die Echtzeitfähigkeit des Bussystems keine notwendige Bedingung. Entgegen möglicher besserer Lösungen wurde sich zugunsten von Standard-Ethernet entschieden. Die Infrastruktur für Ethernet steht im gesamten Unternehmen zur Verfügung. Jeder Arbeitsplatz ist mit einem netzwerkfähigen Rechner ausgestattet und damit auch mit einem geeigneten Client-Gerät. Ethernet bietet einen stabilen Datenaustausch, ausreichende Datenübertragungsraten, Standards und Protokolle offen zugänglich und es lassen sich frei verfügbare Quellcodes finden, wodurch der Entwicklungsaufwand stark vereinfacht wird.

Entwicklungsplattform – Neben dem verwendeten Bussystem wird die Systemarchitektur auch noch von der Entwicklungsplattform bestimmt. Vereinfacht gesagt stellt sich die Frage welcher Mikrocontroller verwendet werden soll. Auch bei der Wahl des Mikrocontrollers sollte das Prinzip der *Variantenbildung* Anwendung finden. Auch hier bietet sich ein *Top-Down* Vorgehen in zwei Schritten an. Das in Abbildung 28 dargestellte Vorgehen soll zu einer möglichst guten Lösung führen.

1. Prozessorwahl: Zunächst wird eine geeignete Prozessorarchitektur gesucht. In der ersten Detaillierungsebene kann für die Design-Philosophie zwischen Prozessoren mit reduziertem Befehlssatz (RISC-Prozessoren) und Prozessoren mit komplexem Befehlssatz (CISC-Prozessoren) entschieden werden. In einer folgenden Detaillierungsebene kann die geeignetste Prozessorarchitektur gewählt werden.
2. Controllerwahl: Mikrocontroller unterscheiden sich zu einfachen Mikroprozessoren durch die integrierte Peripherie. So finden sich in einem Mikrocontroller neben dem Prozessor meist auch Arbeits- und Programmspeicher sowie eine Reihe anderer Komponenten wie A/D-Wandler, parallele und serielle Schnittstellen und Treiber integriert in einem einzigen Chip. Die integrierte Peripherie bestimmt somit im Wesentlichen den nächsten Auswahlschritt. Ein Produktvergleich soll schließlich zum passenden Controller führen für die entsprechende Entwicklungsplattform angeschafft werden kann.

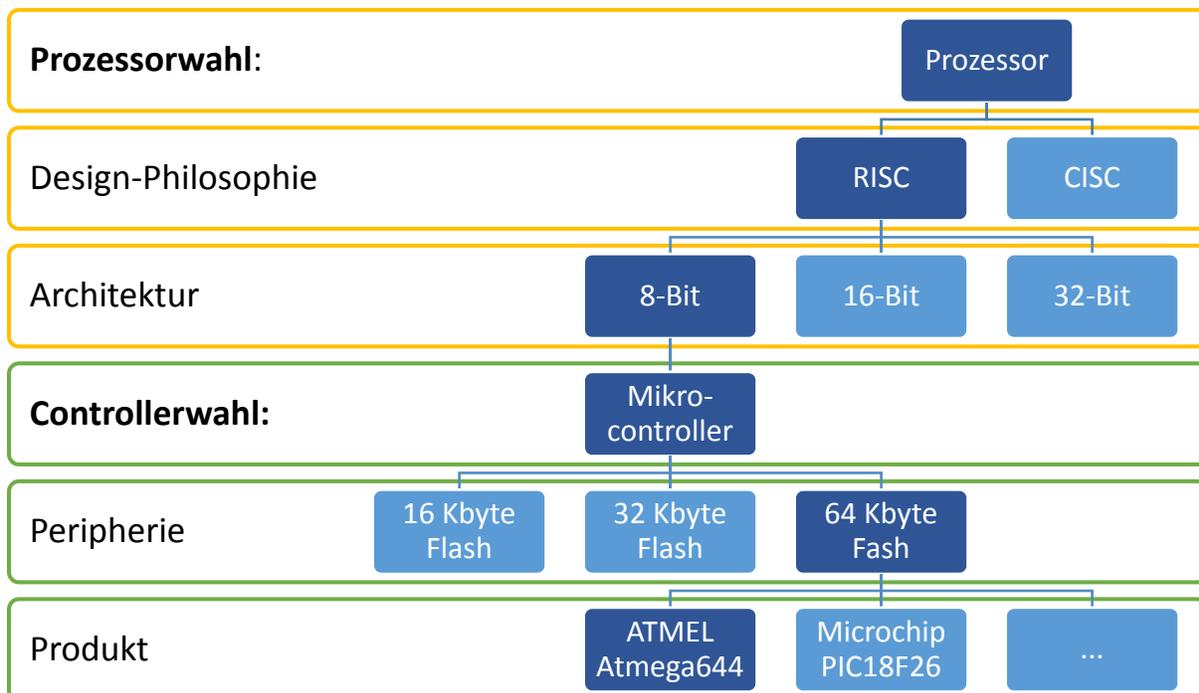


Abbildung 28: Top-Down Vorgehen bei der Auswahl des Mikrocontrollers mit Variantenbildung in jeder Ebene.

5.3.3 Gesamtkonzept

Nach dem Prinzip *vom Groben ins Detail* wird nun versucht, sich ein Bild vom Inneren des Systems zu schaffen. Abbildung 29 soll eine vereinfachte Darstellung der Systemstruktur des zu entwickelnden Zielsystems zeigen.

Kern des Systems sei der Mikrocontroller. Dieser benötigt gewisse periphere Komponenten wie z.B. einen Oszillator aber auch eine geeignete Stromversorgung, um operieren zu können. Sensoren und Umsysteme werden über die analogen und digitalen Ein- und Ausgänge mit dem Mikrocontroller verbunden.

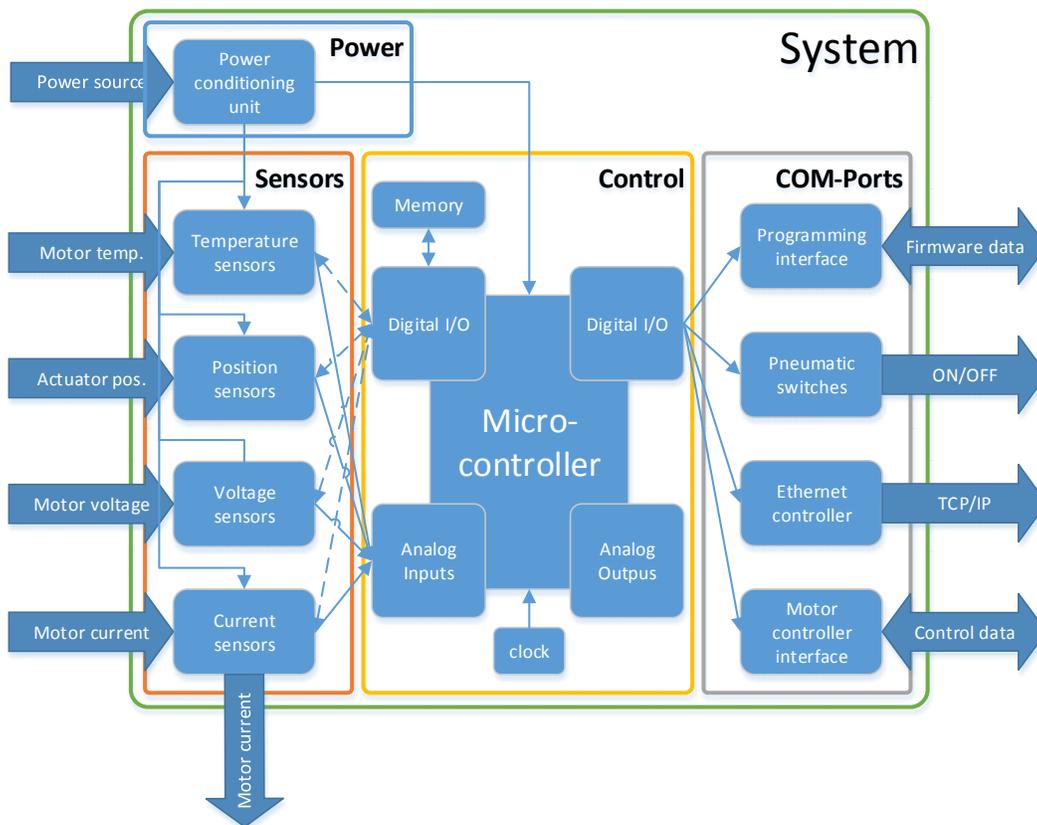


Abbildung 29: Erste Detaillierungsebene der Systemstruktur des Zielsystems (Grey-Box)

Eine eindeutige Trennung von Ein- und Ausgängen ist in obiger Darstellung nicht möglich. Dennoch wurde versucht, die Systemeingänge auf der linken, Systemausgänge auf der rechten Seite darzustellen. Die einzelnen Systemelemente in dieser Darstellung wurden anschließend zu Teilsystemen gruppiert. Dabei wurde zwischen Stromversorgung (*power*), Sensorik (*sensors*), Steuerung (*control*) und Kommunikationsschnittstellen (*COM-Ports*) unterschieden. Zur weiteren Detaillierung müssen diese Komponenten nun festgelegt werden.

Sensorik – Die Wahl der Sensoren, und deren Schnittstellen, bestimmt wesentlich die Wahl des Mikrocontrollers und umgekehrt. Analoge Sensoren übergeben ihre Messdaten meist in Form analoger Spannungen an den Mikrocontroller. Dies wird in Abbildung 29 dargestellt durch einen unidirektionalen Pfeil. Bei analogen Signalen ist wichtig, dass Mikrocontroller und Sensor sich auf dasselbe Potential beziehen, da sonst die gemessenen Daten jeglichen Informationsgehalt verlieren. Dies wird in der Darstellung über die gemeinsame Versorgungsquelle (*power conditioning*) gezeigt. Handelt es sich jedoch um digitale Sensoren, so werden die Messdaten in digitaler Form von Sensor zu Mikrocontroller übergeben. Um eine gemeinsame Referenz sicherzustellen, existiert hier ein Retourpfad von Mikrocontroller zu Sensor. Dieser Pfad ist in Abbildung 29 als bidirektionaler Pfeil dargestellt.

Für die Auswahl der Sensoren kann, unter Rücksichtnahme auf die vorhandenen Mikrocontrollerschnittstellen, das Prinzip der *Variantenbildung* zur Anwendung kommen. In diesem Fall kann das besagte Prinzip auch sehr anschaulich in mehreren Detaillierungsebenen stattfinden. In der ersten Ebene können verschiedene Messprinzipien und deren Eignung für die gegebene Problemstellung gefunden und verglichen werden. In der nächsten Detaillierungsebene kann, ausgehend vom gewählten Messprinzip, eine Reihe von Produkten verschiedener Hersteller recherchiert und verglichen werden. Schlussendlich können in einer dritten Detaillierungsebene verschiedene Bauvarianten des ausgewählten Produkts betrachtet werden, woraus die beste Variante auszusuchen ist.



Abbildung 30: Drei-Phasen-Modell zur Sensorauswahl

Beispielhaft soll dieses Vorgehen am Prinzip der Temperatursensoren durchgeführt werden.

Prinzipvergleich: Zur Temperaturmessung besteht eine Vielzahl an verschiedenen Messprinzipien, welche hier jedoch nicht alle detailliert erklärt und analysiert werden. Es soll hier lediglich das prinzipielle Vorgehen am Beispiel gezeigt werden. Dazu wurden die in Abbildung 31 dargestellten Messprinzipien verglichen, welche in Tabelle 12 kurz beschrieben werden.

Tabelle 12: Kurzbeschreibung verschiedener Messprinzipien zur Temperaturmessung

Messprinzip	Beschreibung
Volumenänderung	Dieses Prinzip basiert auf der Tatsache, dass eine Vielzahl von Stoffen mit einer Änderung ihres Volumens auf steigende oder fallende Temperaturen reagieren (z.B. bei Quecksilberthermometern). Nachteil dieses Messprinzips ist, dass diese Eigenschaft für die Messung schwer in eine elektrische Größe umzuwandeln ist. Eine Art von Sensor bei dem dieses Messprinzip zur Anwendung kommt, ist der Bimetall-Schalter.
Thermoelektrizität	Dabei handelt es sich um eine Eigenschaft, die auftritt, wenn zwei verschiedene Metalle (elektrische Leiter) miteinander in Kontakt stehen. Nach dem so genannten Seebeck-Effekt entsteht in einem geschlossenen Stromkreis im Falle einer Temperaturdifferenz an der Kontaktstelle dieser Metalle eine elektrische Spannung, welche zum Temperaturgefälle proportional ist. Diese, wenn auch sehr kleine Spannung, kann gemessen werden, benötigt jedoch eine hochwertige Auswerteelektronik.
Widerstandsänderung	Halbleiter oder Leiter ändern ihre elektrische Leitfähigkeit mit der Temperatur. Man unterscheidet zwischen Materialien mit positivem Temperaturkoeffizient, d.h. der Widerstand steigt mit der Temperatur an (Kaltleiter), und jenen mit negativem Temperaturkoeffizient, wobei der Widerstand mit steigender Temperatur sinkt (Heißleiter). Wesentlicher Nachteil dieses Prinzips sind die nichtlinearen Zusammenhänge zwischen Temperatur und Widerstand, insbesondere bei Halbleitermaterialien.
Pyrometrisch	Dabei handelt es sich um eine Messung der Wärmestrahlung. Jeder Gegenstand emittiert in Abhängigkeit seiner Temperatur eine elektromagnetische Strahlung, deren Wellenlänge vorrangig im Infraroten Bereich liegt. Diese Strahlung wird als Wärmestrahlung bezeichnet. Anhand der Strahlungsleistung (bzw. aus Frequenz und Amplitude) kann die Temperatur des Objekts festgestellt werden. Damit ist eine berührungslose Temperaturmessung über einen großen Temperaturbereich möglich. Messergebnisse können jedoch sehr leicht durch äußere Einflüsse verfälscht werden.

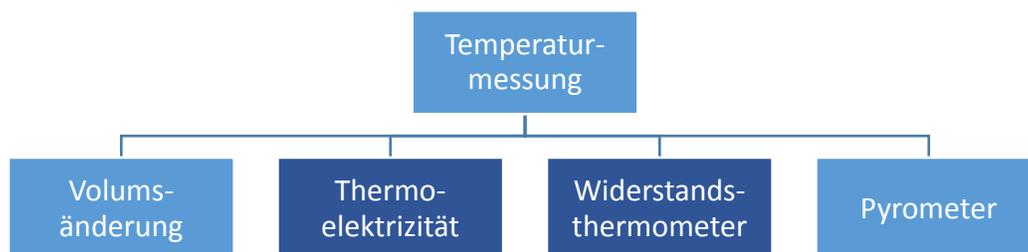


Abbildung 31: Prinzipvergleich der Messprinzipien bei der Temperaturmessung

Produktvergleich: Für den Produktvergleich empfiehlt es sich, gewisse Kriterien entsprechend der Anforderungsanalyse festzulegen. Diese können als Suchkriterien fungieren, wobei jene Produkte aus der Suche ausgeschlossen werden, welche die Anforderungen nicht erfüllen

Beispiele für Kriterien der Temperaturmessung sind:

- Temperaturbereich: -40°C bis 100°C
- Genauigkeit: $\pm 0,5^\circ\text{C}$
- Auflösung: 1°C
- Schutzklasse: IP54
- Betriebsspannung: 5V

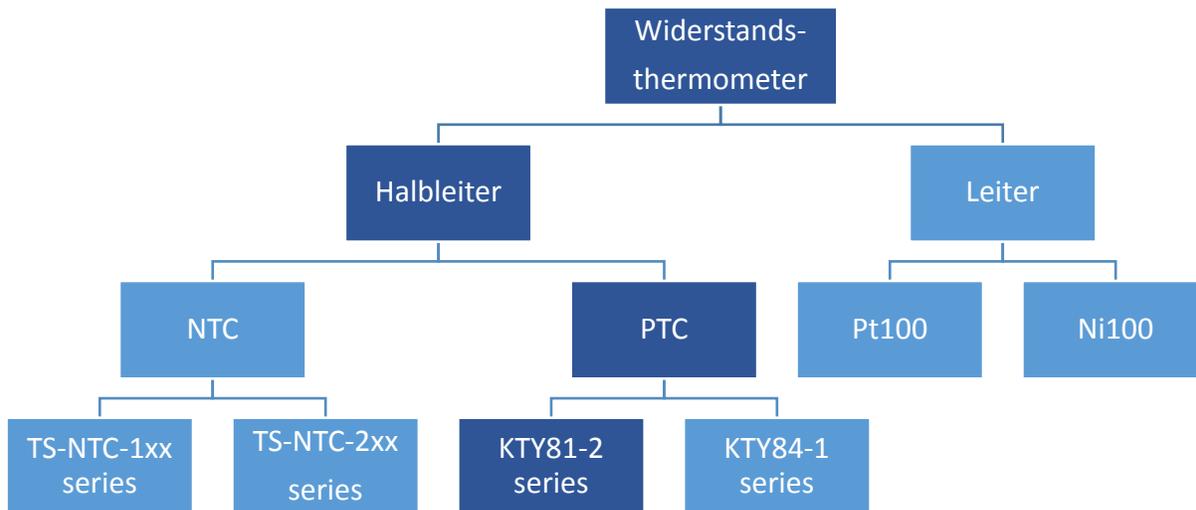


Abbildung 32: Produktvergleich am Beispiel von Widerstandsthermometern

In Tabelle 12 wurde ein Überblick über mögliche Produktgruppen von Widerstandsthermometern geschaffen. Dieses Messprinzip wurde als Beispiel dafür herangezogen, da hier eine stufenweise *Variantenbildung* in weiteren Detaillierungsebenen möglich ist.

Der Produktvergleich hat schlussendlich zu einem Temperatursensor der Serie DS1820 geführt. Dabei handelt es sich um digitale Sensoren, welche über einen so genannten 1-Wire Bus am Mikrocontroller angeschlossen werden. Wesentlich für diese Entscheidung war, dass keine besondere zusätzliche Elektronik notwendig ist, kein A/D-Wandler benötigt wird und der Sensor bereits kalibriert ist. Über den Sensorbus können dazu mehrere Temperatursensoren angeschlossen werden, ohne die am Mikrocontroller begrenzte Anzahl an Eingängen zu schmälern.

Variantenvergleich: Viele Produkte gibt es in unterschiedlichen Ausführungen. Diese können sich in den elektrischen Eigenschaften unterscheiden oder auch in der baulichen Ausführung. Der Gewählte Temperatursensor kann in unterschiedlichen Varianten erworben werden. Aufgrund des Einsatzgebiets wurde die robuste Metallausführung gewählt.

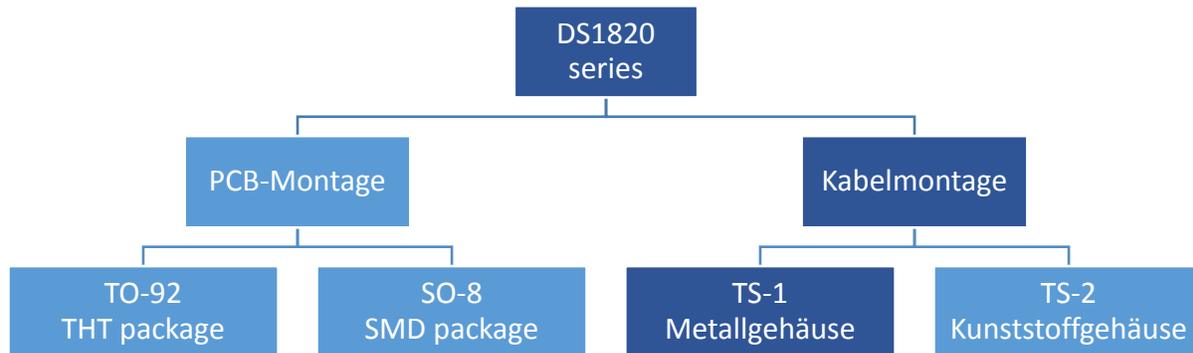


Abbildung 33: Variantenvergleich am Beispiel des gewählten DS1820 Sensors

Die Anwendung der *Variantenbildung* bei der Produktauswahl kann prinzipiell auf alle Sensoren erfolgen. Nicht immer sind dabei alle drei Detaillierungsebenen notwendig. Erfahrungswerte spielen bei der Produktauswahl ebenso eine große Rolle. Die in Tabelle 13 angeführten Sensoren wurden für das System ausgewählt.

Tabelle 13: Tabellarische Übersicht der gewählten Sensoren

Messgröße	Temperatur	Position	Motorstrom	Motorspannung
Sensor	Maxim DS1820	Megatron S501P 2000 R10k	LEM LTSR 6-NP	Eigenentwicklung OPV LM324N
Messprinzip	unbekannt	Potentiometrisch	Hallsonde	Differenzverstärker
Schnittstelle	Digital 1-Wire	Analog	Analog	Analog
Messbereich	-55 bis 125 °C	0 bis 2000 mm	0 bis 19 Ampere	± 50 Volt
Auflösung	9 Bit	Stufenlos	Stufenlos	Stufenlos
Versorgung	3 bis 5,5 Volt	max. 42 Volt	5 Volt	ca. 6 Volt

COM-Ports – Neben Sensoren verfügen Mikrocontroller auch über digitale Schnittstellen, um mit anderen Systemen kommunizieren zu können. In vielen Fällen werden jedoch zusätzliche so genannte Treiberbausteine oder eine andere Peripherie benötigt. Auch in diesem Fall bietet sich eine *Variantenbildung* an, was hier jedoch nicht näher erläutert wird. Dennoch soll kurz festgehalten werden, welche Entscheidungen gefällt wurden.

- Programmierschnittstelle: Nachdem die Programmierschnittstelle als dezidierte Systemschnittstelle nicht notwendig erscheint, wurde diese als solche auch nicht ausgeführt. Parametrierungen können über das Webinterface erfolgen. Deshalb wurde sich für eine ISP Variante entschieden. Änderungen an der Firmware können nicht von außen erfolgen.
- Pneumatik-Schnittstelle: Die Schnittstelle zur Pneumatik stellt den Blick in die Zukunft dar. Für die Zukunft sind Änderungen an der Prüfstandmechanik angedacht. Dabei sollen die schweren Stahlgewichte von Pneumatik-Zylinder ersetzt werden. Deshalb wurde eine Schnittstelle angedacht, welche auch Ventile schalten kann. Die Entscheidung fiel zugunsten von einfachen Relais aus, welche neben hohen Schaltspannungen und Strömen auch eine Potentialtrennung zwischen Steuergerät und Pneumatik ermöglicht.
- Netzwerkschnittstelle: Um auf den Ethernet-Bus zugreifen zu können, bedarf es eines passenden Treibers. Nach dem *Bottom-Up* Prinzip die Wahl aufgrund der verwendeten Entwicklungshardware getroffen.
- Motorsteuerungsschnittstelle: Für den steuernden Zugriff auf die Motorsteuerungen der Firma LOGICDATA stehen zwei Alternativen zur Auswahl. Zum einen handelt es sich dabei um einen vierpoligen parallelen Port, zum anderen um einen seriellen Port. Hier wurde sich für die Ausführung beider Schnittstellen entschieden. Mit dieser Entscheidung ergeben sich in Zukunft mehrere Möglichkeiten für den Systemeinsatz, wobei die Funktion ausschließlich von der Software abhängig ist.

Web-Services – Grundlegende Idee ist, dass jedes System über eine eigene IP-Adresse verfügt. Über einen Webbrowser (z.B. Internetexplorer) kann auf diese IP-Adresse zugegriffen werden. Das System fungiert als Webserver und retourniert dem Benutzer eine dynamische Webseite, auf welcher Zustandsdaten angezeigt werden und Steuerungsoptionen zur Verfügung stehen. Für diese Aufgabe werden zwei Komponenten benötigt. Zum Ersten das Übertragungsprotokoll *Hypertext Transfer Protokoll (HTTP)*, welches dazu dient, Webseiten von einem Webserver zu laden. Zum Zweiten die Auszeichnungssprache *Hypertext Markup Language (HTML)*, in welcher mehr oder minder die Webseite dargestellt wird.

In Tabelle 14 sind die verwendeten Web-Services aufgelistet und kurz erklärt. Außerdem wird gezeigt, für welchen Zweck sie genutzt werden.

Tabelle 14: Verwendeten Web-Services und ihr Einsatzzweck

Web-Service	Erklärung	Zweck
File Transfer Protocol (FTP)	Anwendungsprotokoll zum Dateitransfer	Zugriff auf gespeicherte Messdaten
Hypertext Transfer Protocol (HTTP)	Anwendungsprotokoll zur Übertragung von Webseiten	Zugriff auf die eigentliche Benutzer-Schnittstelle, in Form einer HTML Seite.
Network Time Protocol (NTP)	Anwendungsprotokoll zur Uhrensynchronisation	Synchronisation der Systemzeit für den Zeitstempel der Messdaten
Simple Mail Transfer Protocol (SMTP)	Anwendungsprotokoll zum Versandt von E-Mails	Alarmierung im Fehlerfahl (Fehlermeldung)
Telnet	Anwendungsprotokoll für Fernzugriffe	Fernzugriff auf die Motorsteuerung zu Wartungszwecken (Tunneling)

5.3.4 Teilprojekte

Eine Möglichkeit das Projekt zu gliedern ist es, Teilsysteme aus den einzelnen Elementen zu bilden. Jedes dieser Teilsysteme kann als Teilprojekt, jedes Element als einzelne Aufgabe gesehen werden.

Kritik – Eine solche Gliederung in Teilprojekte kann für große Projekte als sinnvoll erachtet werden, wenn jedes Teilprojekt von einem kleinen Projektteam ausgeführt wird. Ansonsten würden mögliche Aufgaben die Kompetenzen einzelner Teammitglieder übersteigen. Jede ermittelte Aufgabe beinhaltet Tätigkeiten aus verschiedenen Fachrichtungen. Anders ausgedrückt, handelt es sich bei einem Teilprojekt wiederum um ein eigenständiges Projekt, welches in Projektphasen gegliedert werden kann. Jedes Projektmitglied hat unterschiedliche Fachkompetenzen passend zu einzelnen Phasen. Damit können einzelnen Mitgliedern auch nur gewisse Verantwortlichkeiten übertragen werden.

Alternative – Um das Gesamtprojekt sinnvoller in mehrere Einzelprojekte zu gliedern, bedarf es einem Überblick über die verschiedenen Tätigkeiten die mit dem Projekt in Verbindung stehen. Wenn verschiedene Tätigkeiten einzelner Spezialgebiete identifiziert worden sind, kann das Gesamtprojekt möglichst effektiv und effizient umgesetzt werden.

Um für das Fallbeispiel einzelne Tätigkeiten festzustellen, wurde versucht, einen Überblick über einige aus dem Bereich des Entwurfs Eingebetteter Systeme zu schaffen. Die Erfassung von Einzelaufgaben kann auf Basis der Kompetenzen einzelner Teammitglieder erfolgen. So kann eine klare Zuordnung zwischen Mitarbeiter und Aufgabe geschaffen werden.

Wenn alle Tätigkeiten in ihrer Gesamtheit erfasst sind, können Abhängigkeiten zwischen diesen gebildet werden. Damit können Tätigkeiten in eine logische Reihenfolge gebracht werden, in welcher sie

abgearbeitet werden müssen. Bestehen zwischen Tätigkeiten keine Abhängigkeiten, können diese parallel durchgeführt werden.

Da im vorliegenden Fallbeispiel die Aufgaben von einer einzelnen Person erfüllt werden, können diese nicht simultan erfolgen. Deshalb wurde versucht, eine logische Arbeitsfolge zu schaffen. Prinzipiell handelt es sich dabei um ein eigenes Phasenmodell.

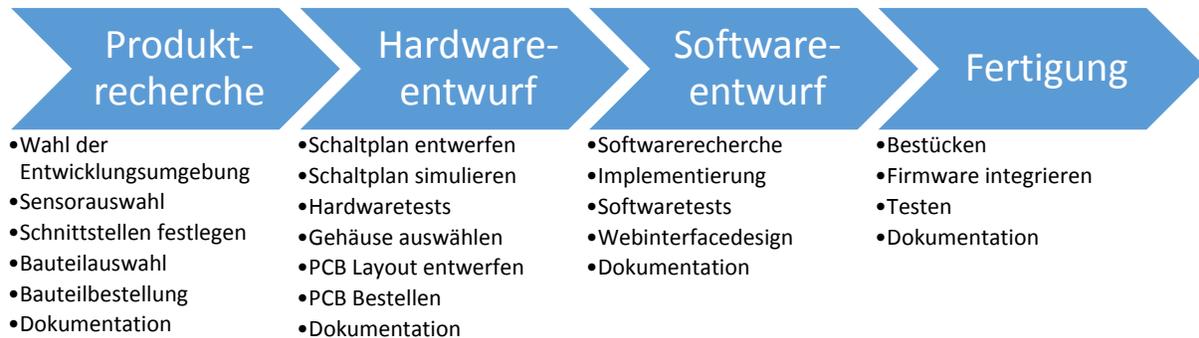


Abbildung 34: Phasenmodell für die Detailstudie als Alternative zu Teilprojekten

Für das Fallbeispiel wurde jede dieser Phasen als eigenes Teilprojekt betrachtet. Für jede dieser Projektphasen ist ein anderes spezifisches Fachwissen von Nöten.

- Produktrecherche: Bevor mit der Hardwareentwicklung begonnen werden kann, müssen die Bauteile und Komponenten feststehen, welche verwendet werden sollen. Natürlich handelt es sich hierbei um einen iterativen Prozess. So werden auch im Zuge des Schaltungsentwurfs Änderungen an den einzelnen Komponenten durchgeführt. Die Bestellung aller Zukaufteile soll auch erst erfolgen, wenn der Hardwareentwurf fertiggestellt ist. Dennoch kann für die Entscheidungsfindung ein Zukauf einiger der verwendeten Bauteile und Baugruppen nötig sein, wie zum Beispiel für Versuchsaufbauten. Hier kommt Fachwissen und Erfahrung aus dem Bereich der Produkt- und Marktanalyse zum Einsatz.
- Hardwareentwurf: Wenn die Komponenten festgelegt sind, können die Schaltpläne entworfen werden. Hier ist die technische Modellbildung und Simulation ein besonderes Entwicklungswerkzeug. So hilft dieses Werkzeug auch, geeignete Bauteile auszuwählen, ohne dass diese dem Entwickler für Versuche zur Verfügung stehen. Dies zeigt wiederum, dass die beiden Phasen der Produktrecherche und des Hardwareentwurfs nicht rein sequentiell ablaufen können. Einfach ausgedrückt, legt der Konstrukteur fest, welche Zukaufteile für die Konstruktion benötigt werden. Hingegen kann der Konstrukteur keine Bauteile in seiner Konstruktion

einplanen, die vom Einkauf nicht beschafft werden können. Für den Hardwareentwurf sind im Fallbeispiel Kenntnisse über Elektronik und Schaltungsentwurf notwendig.

- Softwareentwurf: Ähnlich verhält es sich mit der Softwareentwicklung. Eigentlich kann erst nachdem die Hardware festgelegt ist, mit dem Softwareentwurf gestartet werden. Änderungen an der Hardware wirken sich, wenn auch nicht immer wesentlich, auf die Software aus. Häufig ist jedoch zum Test der Hardware auch die zugehörige Software notwendig. Das bedeutet, nicht nur die Produktrecherche und der Hardwareentwurf sind parallele Phasen, auch der Softwareentwurf läuft zum Teil parallel dazu ab. Im Fallbeispiel werden für den Softwareentwurf Fachkenntnisse zur gewählten Programmiersprache sowie zu Kommunikationsprotokollen und auch verschiedenen Web-Services benötigt.
- Fertigung: Ist das Produkt fertig entwickelt, kann mit der Fertigung begonnen werden. Ein Teil der Fertigung erfolgt bereits am Ende der Hardwareentwurfsphase. So wird bereits im Anschluss an den Hardwareentwurf mit der Fremdfertigung einiger Teile begonnen. Somit konnten bei der Prototypenentwicklung durch das gut gewählte Phasenmodell Fertigungs- und Lieferzeiten zur Softwareentwicklung genutzt werden. Für die Fertigungsphase werden spezifische Kenntnisse wie auch Fertigkeiten aus dem Bereich der Elektronikfertigung benötigt.

5.4 Detailstudie

Nun sollen die Lösungskonzepte soweit detailliert werden und Teilprobleme oder Detailprobleme geklärt, sodass die Lösung im Systembau auch Realisierbar ist. Anschließend müssen die Integrierbarkeit der Teillösungen in das Gesamtsystem sowie die Integrierbarkeit des Gesamtsystems in seine Einsatzumgebung überprüft werden.

Damit werden in der Detailstudie die folgenden Aufgaben durchgeführt:

- | | |
|--------------------------------|---------------------------------|
| 1. Lösungskonzept detaillieren | 4. Integrierbarkeit feststellen |
| 2. Teilprobleme klären | 5. Realisierung ermöglichen |
| 3. Teillösungen konkretisieren | |

5.4.1 Detailkonzept

Nachdem die einzelnen Komponenten ausgewählt wurden und eine genaue Aussage über die Ein- und Ausgänge erfolgen kann, ist es möglich, das Systemmodell weiter zu detaillieren. Dazu wurde die Abbildung 29 aus Kapitel 5.3.3 erweitert. Die einzelnen Systemelemente lassen sich dabei durch die

ausgewählten Produkte und deren Schnittstellen ersetzen. Ferner wurde abgesteckt, welche der Komponenten zusammen auf einer Leiterplatte Platz finden und welche außerhalb liegen. Damit konnten neue physikalische Systemgrenzen gesetzt werden, wodurch neue Schnittstellen entstehen. Abbildung 35 zeigt dieses Modell des Detailsystems in einer vereinfachten Form.

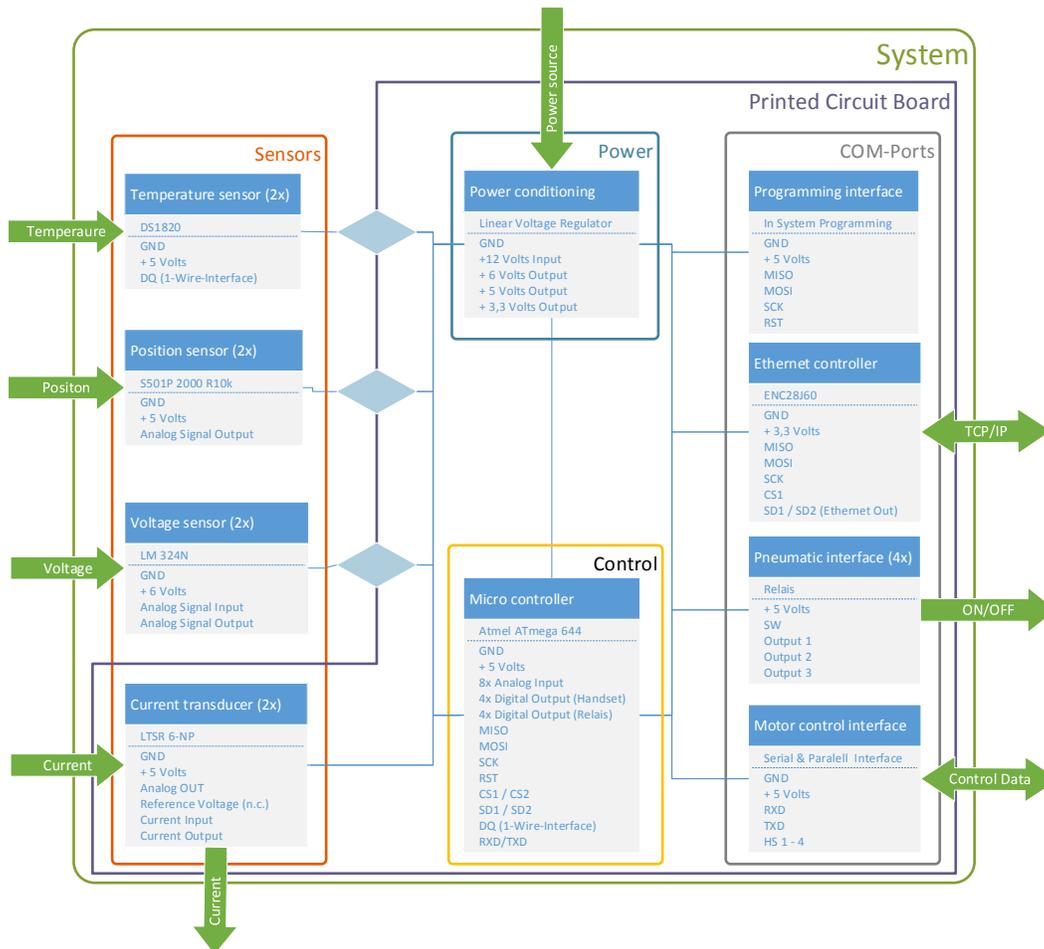


Abbildung 35: Vereinfachtes Modell des Detailsystems

Für die genaue innere Struktur müssen weitere Teilprobleme gelöst werden. Dabei ist im vorliegenden Beispiel vor allem die Peripherie der Systemelemente zu identifizieren, auszulegen und auszuwählen.

Hilfsmittel dabei sind technische Modelle zur Simulation sowie Versuchsaufbauten oder Prototypen zur Durchführung von Tests einzelner Komponenten. Das kann zu einem verbesserten Verständnis der Funktionsweise der Elemente beitragen und mögliche Probleme frühzeitig erkennbar machen. Des Weiteren lassen sich die physikalischen Schnittstellen zum System (Steckverbinder) auswählen. Damit verbunden, müssen Pin-Belegungen vereinbart und vor allem dokumentiert werden.

5.4.2 Teilprobleme

Im Folgenden soll sich exemplarisch auf die Detaillierung eines Teilproblems konzentriert werden, um den Umfang dieser Arbeit im Rahmen zu halten. Dabei soll ein iteratives Vorgehen durch wiederholtes Anwenden des Problemlösungszyklus dabei, helfen das Teilproblem soweit zu konkretisieren und zu lösen, dass dieses im Systembau umgesetzt werden kann.

Messung der Motorspannung – Das gewählte Detailproblem betrifft die Messung der Motorspannung an einem der Mikroprozessoreingänge. Die Messung der Motorspannung stellte bereits vor diesem Projekt ein gewisses Problem dar. So kam es bei Messungen bereits zu Schäden an teurem Messequipment oder zu Fehlern durch falsche Messaufbauten. Das wesentlichste Problem wurde durch Trennen des Bezugspotentials von Messgerät und Messaufbau behoben.

- Situationsanalyse: Die Motorsteuerung wird mit Netzstrom betrieben. Die Steuerung sowie sämtliche Ausgangsspannungen sind auf ein gemeinsames Potential (Erde) bezogen. Messgeräte wie ein Oszilloskop sind ebenfalls auf dieses Potential bezogen.
- Problemidentifikation: Durch einen falschen Messaufbau kann es bei potentialbezogenen Messgeräten in diesem Beispiel durch die Drehrichtungsänderung zu einem Kurzschluss am Messgerät kommen. Solche Probleme sind den Messtechnikern durchaus bekannt.
- Lösungsprinzipien: Um nun die Motorspannungen mittels eines Oszilloskops abgreifen zu können, wurde dieses über einen Trenntransformator von dem Erdpotential getrennt. Damit konnten beide Motordrehrichtungen gemessen werden, ohne dass es zu Kurzschlüssen kommt. Eine Alternative zum Trenntransformator stellte auch ein so genannter differentieller Tastkopf (engl. *differential probe*) dar. Im Fallbeispiel jedoch kann aufgrund von weiteren Aufgaben keine Potentialtrennung zwischen Motorsteuerung und System gemacht werden. Das Messen von Motorspannungen bringt dazu noch weitere Probleme auf wie Rippelströme, Spannungsspitzen und einiges mehr.

5.4.3 Teillösungen

Mehrfaches Anwenden des Problemlösungszyklus (siehe Kapitel 3.1.4) soll ein für den Anwendungszweck hinreichend gutes Ergebnis liefern. Dabei wurde der Problemlösungszyklus in die drei Grundschritte, Situationsanalyse, Zielsetzung und Lösungsfindung gegliedert.

Erster Iterationsschritt:

- Situationsanalyse: Grundsätzlich stellen Spannungsmessungen kein erhebliches Problem dar. Für die Messung von Spannungen werden Analog-Digital-Wandler eingesetzt. A/D-Wandler setzen eine analoge Größe wie eine Spannung in einen digitalen Wert um. Mikrocontroller verfügen üblicherweise über solche Eingänge. Wesentlich dabei ist, dass die gemessene Spannung nicht die maximale Eingangsspannung (meist die Versorgungsspannung des μC) überschreitet bzw. einen Minimalwert nicht unterschreitet. Die Grenzwerte für die Eingangsgrößen können dem Datenblatt des jeweiligen Mikrocontrollers entnommen werden. Im Beispiel eines Atmega 664 liegen die Grenzen bei -0,5 Volt unter der Versorgungsmasse (GND) beziehungsweise +0,5 Volt über der Versorgungsspannung (VCC). Die zu messenden Motorspannungen liegen jedoch weit über bzw. unter diesen Grenzen. Damit bedarf es einer Signalkonditionierung um innerhalb der Grenzwerte zu bleiben.
- Zielsetzung: Ziel ist es, die Motorspannungen soweit anzupassen, dass die Grenzen nicht überschritten werden. Die Motorspannungen liegen im Anwendungsfall bei etwa 30 Volt, dabei handelt es sich um die maximale Ausgangsspannung der Motorsteuerungen.
- Lösungssuche: Um große Spannungen zu messen bedient man sich einem Spannungsteiler. Dabei wird die anliegende Spannung über zwei Widerstände im Verhältnis der Widerstände geteilt. Über zwei geeignete Widerstände kann so die Motorspannung von 30 Volt auf eine geeignete Eingangsspannung von z.B. 5 Volt herunter gesetzt werden.

Zweiter Iterationsschritt:

- Situationsanalyse: Die Motorspannungen werden über einen Spannungsteiler an die möglichen Eingangsspannungen angepasst. Um die Drehrichtung eines Gleichstrommotors zu ändern, müssen die Anschlüsse verpolt werden. Dies geschieht innerhalb der Motorsteuerung über eine so genannte H-Brücke. Die Umschaltung geschieht damit bereits vor dem Messpunkt, wodurch auch die Spannung an den Eingängen des Mikrocontrollers umgekehrt wird. Nachdem beide

Geräte, d.h. Motorsteuerung und Mikrocontroller, dasselbe Bezugspotential besitzen, käme es dadurch zu einem Kurzschluss.

- Zielsetzung: Ziel ist also, eine Lösung zu finden, welche die Umkehrung der Spannungen an den Mikrocontrolleranschlüssen und damit einen Kurzschluss verhindert.
- Lösungssuche: Grundsätzlich suchen wir nach einem Verpolschutz. In einem solchen Fall kommen meist Dioden, welche eine Spannungsumkehr verhindern, zum Einsatz. Hier kommen zwei Varianten in Frage. Zum einen wäre das der Einweggleichrichter bestehend aus einer einzelnen Gleichrichterdiode, zum anderen ein Brückengleichrichter bestehend aus vier Dioden. Nachteil des Einweggleichrichters ist, dass damit die Motorspannung nur noch in einer Drehrichtung gemessen werden könnte. Der Brückengleichrichter bietet den Vorteil, dass beide Drehrichtungen erfasst werden können, wobei anzumerken ist, dass durch die Gleichrichtung die Information über die Drehrichtung verloren geht.

Dritter Iterationsschritt:

- Situationsanalyse: Mithilfe von Spannungsteiler und Brückengleichrichter können die Motorspannungen unabhängig von ihrer Drehrichtung gemessen werden. Bei näherer Untersuchung des Anwendungsfalles konnte festgestellt werden, dass die Motorsteuerungen an den Motorausgängen nicht nur positive Spannungen bezogen auf das Grundpotential anliegen haben. Das bedeutet, es müssen trotz Gleichrichtung auch negative Spannungen gemessen werden.
- Zielsetzung: Ziel ist, eine Möglichkeit zu finden, negative Spannungen an den Eingängen eines Mikrocontrollers messen zu können.
- Lösungssuche: Um negative Spannungen messen zu können, hilft weder ein Spannungsteiler gegen Masse, noch ein Brückengleichrichter. Eine einfache Lösung des Problems stellt jedoch ein Spannungsteiler gegen eine positive Spannung dar. Durch geeignete Widerstandsverhältnisse kann so die Spannung am Eingang des Mikrocontrollers auf einen geeigneten Wert angehoben werden.

Vierter Iterationsschritt:

- Situationsanalyse: Durch die getroffenen Maßnahmen scheinen nun die wesentlichen Probleme der Motorspannungsmessung behoben. Der Brückengleichrichter verhindert die Umpolung an den Eingängen bei einer Drehrichtungsumkehr. Am positiven Ausgang des Gleichrichters kann der

positive Teil der Motorspannung gemessen werden, am negativen Gleichrichterausgang die negative Spannung. Diese Situation zeigt jedoch, dass für die Messung zwei Mikrocontrollereingänge benötigt werden, um die Spannungsdifferenz zu messen.

- Zielsetzung: Aufgrund der begrenzten Anzahl an Eingängen am Mikrocontroller sollte vermieden werden, dass die Eingänge unnötigerweise verbraucht werden. Ziel ist es, eine Lösung zu finden, um die Spannungsdifferenz der Motorkanäle an lediglich einem Mikrocontrollereingang zu messen.
- Lösungssuche: Gesucht ist also eine Lösung, um differentielle Spannungen zu messen. Für diese Anwendung bestehen einige Lösungen, die übliche Variante ist eine Operationsverstärkerschaltung. Bei der gesuchten OPV-Schaltung handelt es sich um einen Differenzverstärker. Beim Differenzverstärker ist die Ausgangsspannung proportional zur Spannungsdifferenz an den OPV-Eingängen.

Fünfter Iterationsschritt:

- Situationsanalyse: Für die entstandenen Teilprobleme wurden Teillösungen gefunden. Jede dieser Teillösungen für sich erfüllt seine Aufgabe. Jedoch nach dem Prinzip, dass das Ganze ist mehr als die Summe seiner Teile ist, muss zuerst sichergestellt werden, dass die Kombination der Teillösungen das Gesamtproblem löst.
- Zielsetzung: Gesucht ist eine Kombination der Teillösungen, die sicherstellt, dass nicht nur die Teilprobleme sondern das Gesamtproblem gelöst ist.
- Lösungssuche: Für die Lösung des Gesamtproblems können die Einzellösungen (Gleichrichter, Spannungsteiler, Differenzverstärker) in unterschiedlicher Reihenfolge zusammengesetzt werden. Jedoch nicht jede Kombination führt auch zu einer sinnvollen Lösung. So ist situationsbedingt der Differenzverstärker auf eine vorherige Gleichrichtung der Spannung angewiesen. Aufgrund gewisser Rahmenbedingungen würde andernfalls der OPV außerhalb seines Funktionsbereichs betrieben. Die gewählte Reihenfolge ist: Gleichrichter, Spannungsteiler, Differenzverstärker.

5.4.4 Integrierbarkeit

Um die Integrierbarkeit einer Detaillösung festzustellen, bedarf es eines so genannten Interoperabilitätstests. Dabei kann zunächst auch die technische Modellbildung mit einer anschließenden Simulation zeigen, ob die einzelnen Komponenten miteinander kombiniert auch das gewünschte Ergebnis liefern. Interoperabilitätstests können in unterschiedlichen Detaillierungsebenen erfolgen.

Im Folgenden soll das Teilproblem *Motorspannungsmessung* weitergeführt werden. Es wird versucht, die Integrierbarkeit der gefunden Teillösungen innerhalb ihres Teilsystems festzustellen.

Technische Modellbildung – Ob die Kombination der Einzellösungen wirklich zu einer sinnvollen Gesamtlösung führt, soll zunächst eine technische Modellbildung (vgl. Kapitel 4.3.2) zeigen. Für die Schaltungssimulation bietet sich eine Reihe verschiedener Simulationsprogramme an. Aufgrund vorheriger Erfahrungen wurde auf das Programm PSPICE der Firma Cadence zurückgegriffen, welches in der Version 9.1 kostenlos als Studentenversion zur Verfügung steht. Abbildung 36 zeigt das gebildete technische Modell, welches anschließend simuliert werden soll.

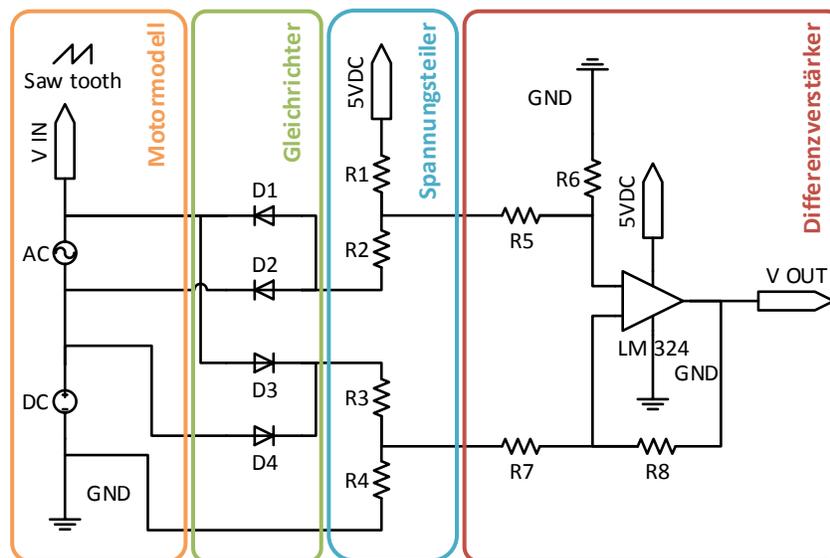


Abbildung 36: Simulationsmodell zur technischen Modellbildung am Beispiel der Motorspannungsmessung

Anstelle der Motorspannung wird im Modell am Eingang eine rampenförmige Spannungsquelle verwendet. Um das von der gemeinsamen Masseverbindung abgesenkte Potential zu veranschaulichen, wurde eine Gleichspannungsquelle angefügt, welche einen so genannten Offset generiert. Die angelegte Spannung wird wie beschrieben zuerst über vier Dioden gleichgerichtet. Der negative Brückenausgang wird über einen Spannungsteiler und einer definierten positiven Versorgungsspannung von 5 Volt

angehoben. Der positive Brückenausgang wird analog über einen Spannungsteiler gegen Masse auf eine geeignete Spannung abgesenkt. Über einen Differenzverstärker mit vier gleichen Widerständen wird eine einfache Spannungsdifferenz gebildet.

Die Simulationsergebnisse nach Abbildung 37 zeigen, dass die Gesamtlösung ein brauchbares Ergebnis liefert. Jedoch zeigen die Ergebnisse der Simulation ein weiteres Problem auf. Das über die Schaltung generierte Signal ist invertiert.

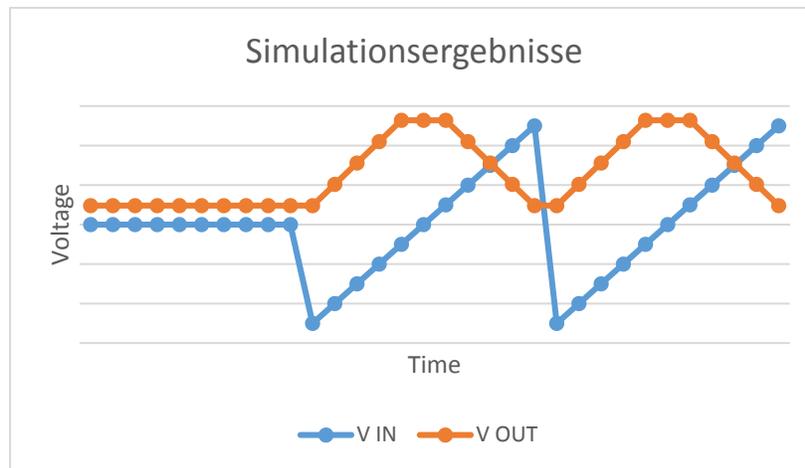


Abbildung 37: Qualitative Darstellung der Ergebnisse aus der technischen Simulation

Das bedeutet, wenn die Motorspannung ansteigt, sinkt die Ausgangsspannung am Operationsverstärker ab und umgekehrt. Weitere Durchläufe des Problemlösungszyklus führten schließlich zu einer guten Lösung des Teilproblems dargestellt in Abbildung 38.

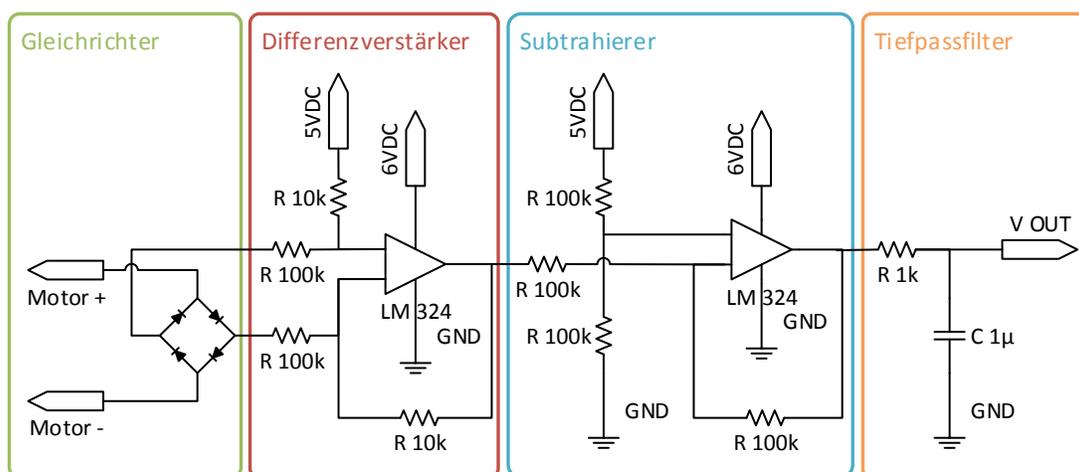


Abbildung 38: Ergebnis der Lösungsfindung zum Teilproblem der Spannungsmessung am Motor

Weitere Probleme die es dabei zu lösen gab, waren unter anderem:

- Auslegung der Spannungsteiler unter Berücksichtigung der Diodenvorwärtsspannung.
- Analoge Invertierung des Ausgangssignals.
- Anheben der Aussteuergrenzen des Operationsverstärkers.
- Filtern von Störungen durch die Motorkommutierung.

Test – Bevor die gefundene Lösung auch wirklich zum Einsatz kommen kann, muss diese getestet werden. Dazu wurde ein Versuchsaufbau mit den in der Simulation verwendeten Komponenten erstellt. Die ausgelegte Schaltung zeigte ein zur mittleren Motorspannung direkt proportionales Ausgangssignal. Die Schaltung wurde so gestaltet, dass Motorspannungen von bis zu 50 Volt gemessen werden können. Dabei wurde ein reales Teilungsverhältnis von 1:10 festgestellt. Bei einem Über- oder Unterschreiten der maximalen Eingangsspannungen erreicht die Schaltung ihre Aussteuergrenzen von ca. 0 Volt bzw. ca. 5 Volt. Durch die hochohmigen Eingänge ist die Schaltung auch gegen Überspannung gut geschützt.

Bottom-up – Bislang wurde nach Möglichkeit das *Top-Down* Prinzip angewendet, um von groben Problemstellungen zu Detailproblemen und daraus zu Detaillösungen zu kommen. Im Gegensatz zur Lösungsfindung bei welcher häufig das *Top-Down* Prinzip empfohlen wird, kann bei der Feststellung der Integrierbarkeit in umgekehrter Reihenfolge vorgegangen werden. Zunächst wird die Integrierbarkeit von Elementen innerhalb der Subsysteme festgestellt, danach die Integrierbarkeit der Subsysteme innerhalb des Gesamtsystems und schlussendlich das Gesamtsystem in seiner Umgebung. Dieses Vorgehen entspricht mehr oder minder dem *Bottom-Up* Prinzip (siehe Abbildung 39).

Im Bezug auf das betrachtete Beispiel würde man nach diesem Prinzip nun die Integrierbarkeit des Teilsystems der Motorspannungsmessung in das Gesamtsystem feststellen und nach Fertigstellung des Gesamtsystems die Integrierbarkeit in die reale Systemumgebung.

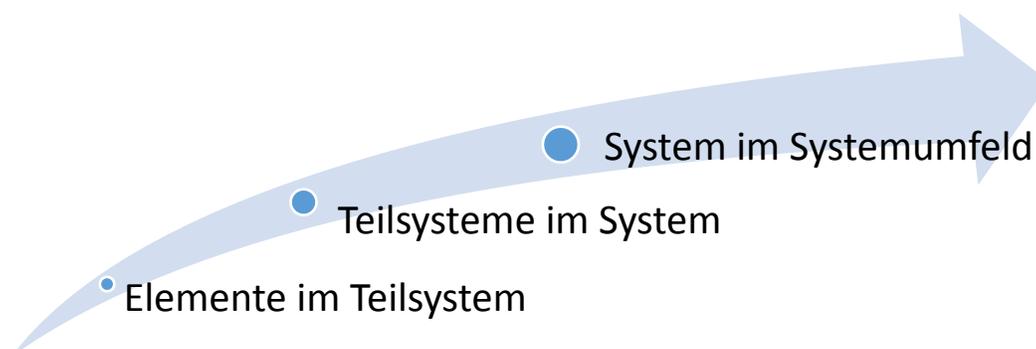


Abbildung 39: Bottom-Up Vorgehen beim Feststellen der Integrierbarkeit

5.5 Systembau

Nun soll das System auch realisiert werden. Dabei soll nach den in Kapitel 5.3.4 vorgestellten Phasen vorgegangen werden. Die Produktrecherche wurde bereits in Vor- und Hauptstudie erledigt, somit folgt die Hardware und Software zu realisieren. Zunächst sollen die notwendigen Rahmenbedingungen geschaffen werden, worauf hin Hardware und Software zu erstellen sind. Schließlich werden Tests des Systems durchgeführt, um Funktionalität und Einsatzfähigkeit sicherzustellen. Demnach sind nachstehende Schritte durchzuführen:

1. Rahmenbedingungen schaffen
2. Hardware realisieren
3. Software realisieren
4. Einzel- und Systemtests

5.5.1 Rahmenbedingungen

Die Rahmenbedingungen für die Realisierung zu schaffen, bedeutet im konkreten Fallbeispiel die notwendige Entwicklungshardware anzuschaffen sowie die benötigten Werkzeuge wie Entwicklungssoftware und Produktionsmittel zur Verfügung zu stellen. Außerdem wurden Fremdfertiger ermittelt, die jene Komponenten herstellen, welche mit den eigenen Fertigungsmitteln nicht hergestellt werden können. Kostenvoranschläge für die Fremdfertigung wurden eingeholt, Fertigungsaufträge erteilt und die Zukaufteile bestellt. Hierbei ist jedoch anzumerken, dass ein Großteil der Rahmenbedingungen schon zu Beginn des Projekts geschaffen wurde.

5.5.2 Hardware

Die im Fallbeispiel stufenweise gezeigte schrittweise Produktdetaillierung bezog sich im Wesentlichen auf die Systemhardware. Demnach ist das im Detailkonzept erstellte Lösungskonzept in der Realisierungsphase nur noch umzusetzen. Dies wird in drei Schritten durchgeführt.



Abbildung 40: Schrittweises Vorgehen in der Phase der Hardwarerealisierung

Schaltplanentwurf – Der Schaltplanentwurf wird mithilfe des Softwareprogramms Eagle durchgeführt. Die Software dient vorrangig der Leiterplattenentwicklung, wovon eine kostenlose Version zum Download zur Verfügung steht. Der Schaltplanentwurf stellt im Grunde eine Detaillierung des Systems auf elementarer Ebene dar. Dabei ist anzumerken, dass es sich beim Entwurf von Elektronikhardware meist um ein *Meet-in-the-Middle* Prinzip handelt. Die verwendeten elementaren Komponenten, insbesondere

integrierte Schaltkreise, können ebenso als Systeme betrachtet werden. In diese Detaillierungsstufe wird jedoch beim Schaltplangentwurf nicht eingetaucht. Ein wesentlicher Unterschied zwischen Schaltplangentwurf und dem Modell des Detailkonzepts ist, dass externe Komponenten wie externe Sensoren (z.B. Temperatursensoren) nicht integriert werden. Das bedeutet, hier werden nur die benötigten Schnittstellen ausgeführt.

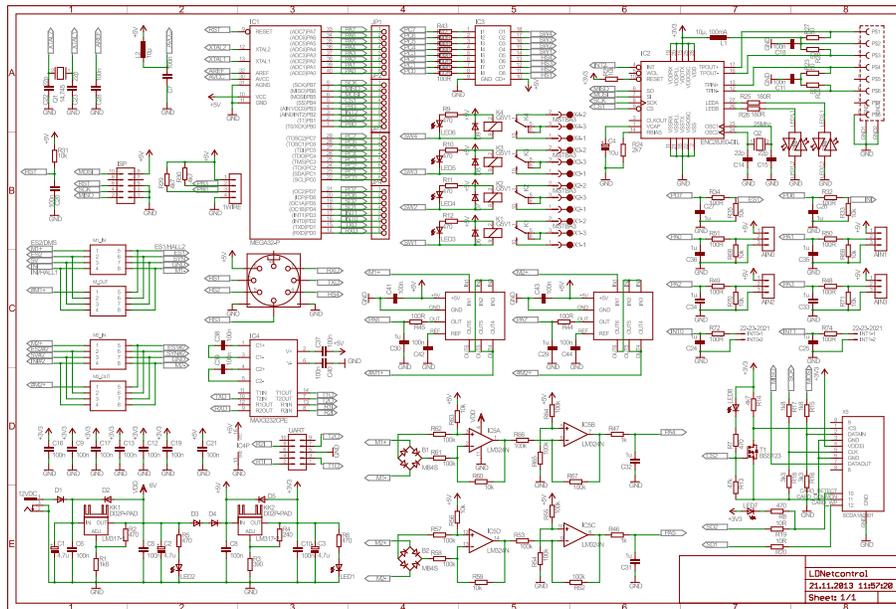


Abbildung 41: Schaltplan des Zielsystems

Leiterplattenentwurf – Der Leiterplattenentwurf stellt ebenso wie der Schaltplangentwurf eine Modellbildung des Systems auf Elementarer Ebene dar. Leiterplattenentwurf und Schaltplangentwurf verfügen über denselben Detaillierungsgrad, ihr Unterschied liegt jedoch im Betrachtungsaspekt. Demnach wird selbst noch im Systembau das Prinzip einer Betrachtung unter verschiedenen Gesichtspunkten gemacht. Der Schaltplangentwurf ist eine Modellbildung, wobei die einzelnen Komponenten in Form von genormten Schaltsymbolen dargestellt werden. Diese Darstellungsform dient dem Systemverständnis, das heißt, anhand des Schaltplans kann auf die Funktion der einzelnen Elemente geschlossen werden. Im Gegensatz dazu dient der Leiterplattenentwurf dazu, das System technisch realisieren zu können. Der Leiterplattenentwurf zeigt außerdem, wie das Endprodukt schließlich aussehen wird. Die Leiterplatten wurden ebenso mithilfe des Konstruktionsprogramms Eagle erstellt.

Für den Leiterplattenentwurf hat sich ein schrittweises Vorgehen nach Abbildung 42 als praktikabel erwiesen.



Abbildung 42: Schrittweises Vorgehen im Leiterplattenentwurf

Nachdem sämtliche Bauteile eingefügt wurden, kann eine Abschätzung der Platinengröße gemacht werden. Danach können Steckverbinder auf der Leiterplatte positioniert werden, sodass ein Gehäusemodell erstellen lässt. Nachdem Gehäusedesign und Steckerposition festgelegt sind können die Komponenten nach den einzelnen Modulen sortiert werden.

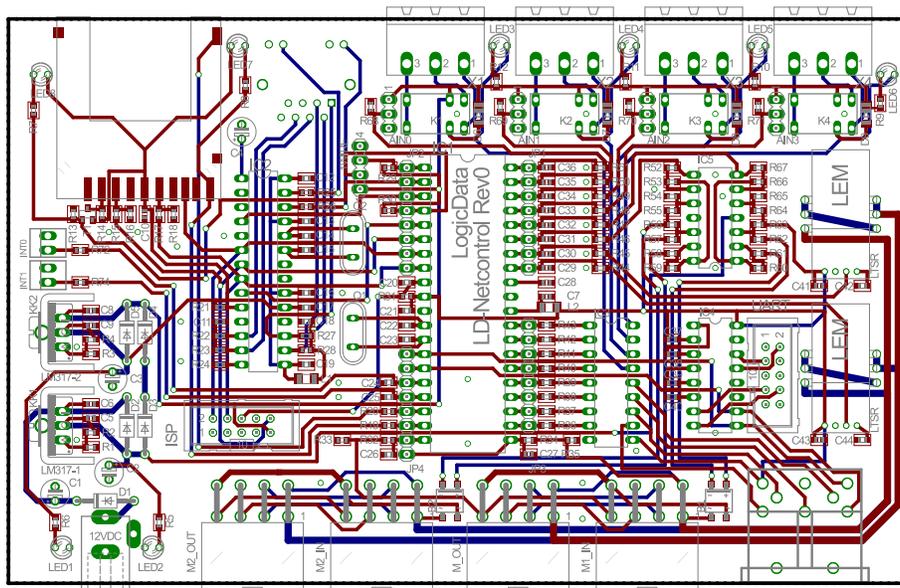


Abbildung 43 Leiterplattendesign des Zielsystems

Die Komponenten wurden so angeordnet, dass die Abstände der Verbindungen zwischen den Komponenten möglichst gering gehalten werden. So wurde Modul für Modul aufgebaut. Anschließend wurden die Module so auf der Leiterplatte verteilt, dass die Distanzen von Verbindungen zwischen den Modulen möglichst gering sind. Natürlich handelt es sich dabei um einen iterativen Prozess, wobei es laufend zu Änderungen kommt. Dennoch kann nach diesem Vorgehen ein gutes Layout gestaltet werden, wie Abbildung 43 zeigt.

Fertigung – Der Fertigungsauftrag für die PCB-Platinen wurde bei einem Fremdfertiger in Auftrag gegeben. *Variantebildung* kann auch bei der Fertigung als grundsätzliches Prinzip Anwendung finden. So können verschiedene Fertiger ausgewählt werden, wie auch unterschiedliche Fertigungsprozesse oder unterschiedliche Materialien. Hier wurde jedoch diesbezüglich kein Aufwand betrieben. Lediglich für die Endfertigung, sprich die Bestückung der Platinen, wären verschiedene Varianten angedacht. Neben der Fremdbestückung durch ein Zulieferunternehmen konnte auch eine Bestückungsschablone vom Platinen-Fertiger geordert werden. Aufgrund der geringen Stückzahl wurde sich für die Bestückung von Hand entschieden.

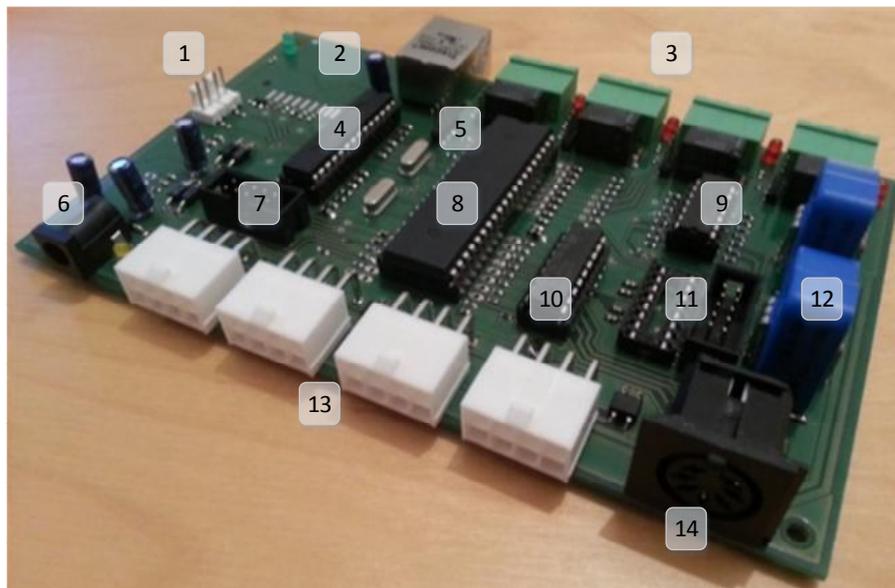


Abbildung 44: Fertig mit Bauteilen bestückte Leiterplatte des Zielsystems

Abbildung 44 zeigt die fertig bestückte Leiterplatte, wobei, durch den Modularen Aufbau, die einzelnen Systemelemente erkennbar sind:

- | | |
|----------------------------------|--|
| 1. Externe Zähler Eingänge | 8. Mikrocontroller |
| 2. Optionale Speichererweiterung | 9. Verstärker zur Motorspannungsmessung |
| 3. Pneumatik Relais Ausgänge | 10. Treiber für Relais und Hanschalterausgänge |
| 4. Ethernet Controller | 11. Optionale Serielle Schnittstelle |
| 5. Temperatur Eingänge | 12. Strom-Sensoren (LEM-Wandler) |
| 6. Stromversorgung | 13. Motor Ein- und Ausgänge |
| 7. Programmier Schnittstelle | 14. Schnittstelle zur Motorsteuerung |

5.5.3 Software

Es empfiehlt sich die Softwareentwicklung je nach Umfang als eigenes Projekt mit eigenen Projektphasen durchzuführen. Für das Fallbeispiel wurde jedoch auf einen offenen Quellcode zurückgegriffen, um das System in der begrenzten Zeit umsetzen zu können. Nach ausgiebiger Recherche wurde sich für die Software von Ulrich Radig¹ entschieden. Die Software bietet alle notwendigen Komponenten und wurde unter anderem für die verwendete Entwicklungshardware geschaffen. Um die Software für die gegebene Problemstellung verwenden zu können, müssen dennoch viele Änderungen durchgeführt werden.

Um den fremden Quellcode für die eigenen Zwecke zu nutzen, musste dieser entsprechend analysiert werden. Hierzu wurde eine Methode der zyklischen Synthese und Analyse eingesetzt.

Die Verwendung fremder Software bringt verschiedene Problematiken mit sich. Häufig ist der Quellcode schlecht dokumentiert und jeder Softwareentwickler verfügt über einen individuellen Programmierstil. Dadurch fällt es schwer, sich in einem fremden Code zurecht zu finden. Gerade nicht objektorientierte Programmiersprachen und der Verzicht auf einen modulartigen Aufbau machen eine Wiederverwendung eines Fremdcodes häufig unmöglich. Der gewählte Quellcode ist leider kaum dokumentiert, weist jedoch ausreichende modulare Struktur auf, um ihn durch Analyse wiederverwenden zu können.

Analyse und Synthese – Der Quellcode steht uns als eine bereits fertige Lösung zur Verfügung. Ausgangspunkt der Softwareentwicklung ist also schon das Detailsystem. Daraus ergibt sich somit ein Bottom-Up Vorgehen. Nachdem der gesamte Quellcode ein relativ komplexes Konstrukt darstellt, wurde dieser zunächst in seine einzelnen Module zerlegt. Anstelle die Funktion des Quellcodes in seiner Gesamtheit erfassen zu können, wurde versucht jedes einzelne Modul für sich zu verstehen. In einem iterativen Prozess wurden jene Teile des ganzen Quellcodes Schritt für Schritt entfernt, die nicht zum Modul zu gehören schienen. Der verbleibende Code wurde getestet, um festzustellen, dass das untersuchte Modul seine volle Funktion behalten hat. Dieser Prozess wurde solange durchgeführt, bis die Software soweit an Komplexität verloren hat, dass der verbleibende Rest in seiner Gesamtheit verstanden werden konnte. Anschließend wurde versucht, die einzelnen Funktionen des untersuchten Moduls zu verstehen, worauf hin Änderungen am Modul durchgeführt werden konnte. Veränderungen am Quellcode können nicht nur in Hinsicht auf die Problemstellung gemacht werden, sondern können auch helfen, einzelne Funktionen besser zu verstehen. Nach verändern des Quellcodes wurden die Module wiederum auf die Erfüllung ihrer Teilaufgabe hin getestet, um in weiterer Folge wieder in den Gesamtcode

¹ Die Open Source Software von Ulrich Radig ist zu finden auf <http://www.ulrichradig.de>

integriert und auf Interoperabilität untersucht zu werden. Dieser Prozess des wieder Zusammensetzens des Quellcodes kann als Synthese bezeichnet werden. Dieser Prozess von Analyse und Synthese wurde solange durchgeführt, bis jedes Modul, welches einer Abänderung bedarf, seine zukünftige Funktion im Gesamtsystem erfüllt.

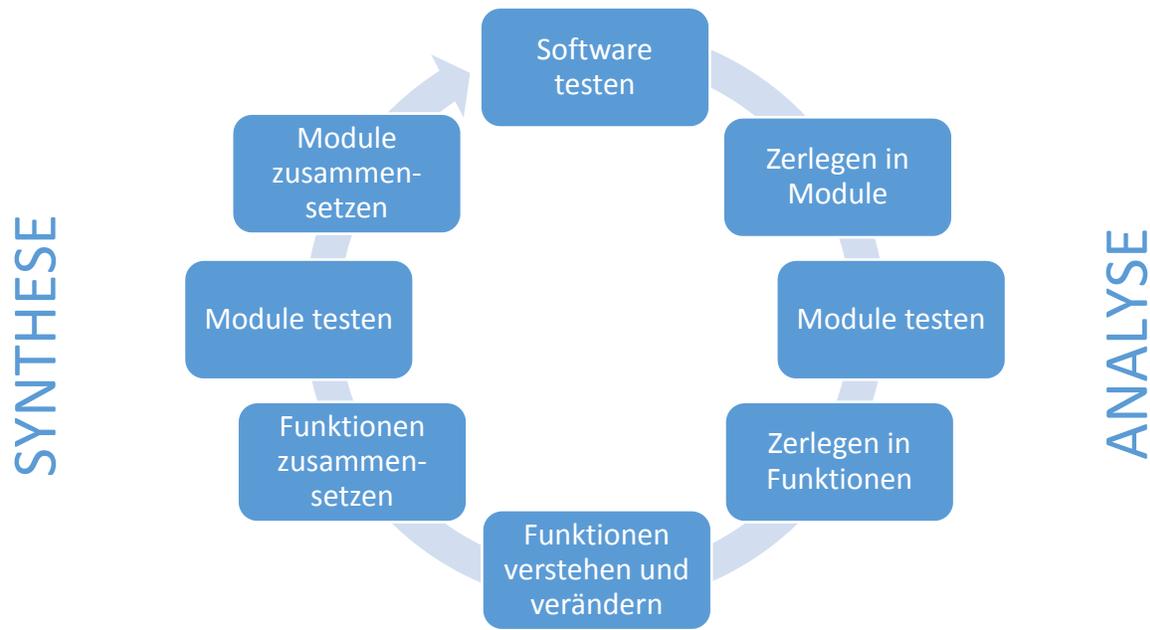


Abbildung 45: Softwareuntersuchung durch iterative Analyse und Synthese

Praktische Anwendung – Das beschriebene Vorgehen soll kurz an einem Beispiel gezeigt werden. Die im Fallbeispiel verwendete Software ist die für das von Ullrich Radig entworfene ETH_M32_EX Experimentierboard entwickelte Webserverprogramm. Die verwendete Software besteht im Wesentlichen aus siebzehn *code files* und zugehörigen *header files*.

Für die Teilfunktion der Telnet-Kommunikation über die Netzwerkschnittstelle werden jedoch nicht alle dieser Files benötigt. Deshalb werden alle Files entfernt, welche scheinbar nicht für Telnet benötigt werden. Um herauszufinden, welche Teile offensichtlich nicht benötigt werden, wird im ersten Schritt das Gesamtsystem getestet und die Funktionsweise der Telnet-Kommunikation untersucht. Diese in Abbildung 46 dargestellte Untersuchung entspricht einer Systemidentifikation (siehe Kapitel 4.3.1).

Das System tunnelt also die von der seriellen Schnittstelle gesendeten Daten und sendet diese via Telnet an einen Client sowie in umgekehrter Richtung. Die vom Client über Telnet gesendeten Daten werden an der seriellen Schnittstelle ausgegeben. Benötigt werden also alle Komponenten der seriellen Schnittstelle sowie jene des Telnet-Servers.



Abbildung 46: Systemidentifikation zur Untersuchung der Fremdsoftware (Black-Box)

Schrittweise wurden nun jene Komponenten gelöscht, welche offensichtlich nicht benötigt werden. Dabei wird der Quellcode regelmäßig kompiliert, um einzelne, zu den gelöschten Komponenten gehörende Fragmente (z.B. die so genannten *includes* der *header files*), aus den Hauptfunktionen zu entfernen. Durch dieses Vorgehen konnte die Telnet-Funktion auf sechs einzelne C-Dateien und zugehörige Header-Dateien reduziert werden. Damit hat sich die Komplexität des Programms deutlich reduziert, sodass die Untersuchung des Quellcodes wesentlich einfacher ist. Abbildung 47 zeigt ein Modell der durch den Analyseprozess vereinfachten Software.

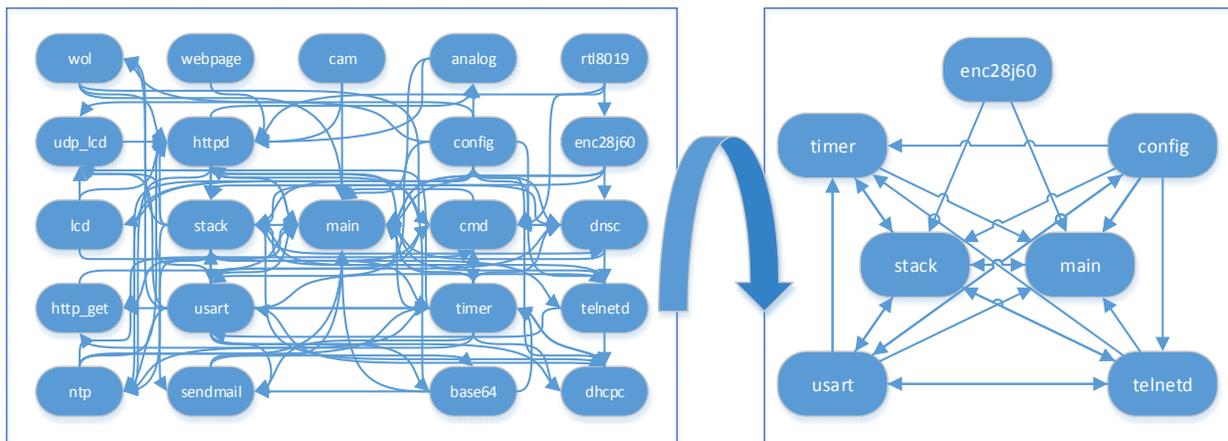


Abbildung 47: Modell der durch Analyse vereinfachten Fremdsoftware

Neben dieser Abstraktion des Gesamtsystems können zusätzlich für erste Betrachtungen einzelne Softwarekomponenten als *Black-Box* betrachtet werden, auch wenn diese eine wichtige Rolle für die Funktion der Software spielen. Im Fallbeispiel wurden die Struktur des Ethernet-Stacks (*stack.c*) sowie die Netzwerktreiber (*enc28j60.c*) für die weitere Analyse der Telnet-Funktion vernachlässigt.

Ausgehend von der Hauptschleife wird nun das vereinfachte Programm in seiner Ablauffolge betrachtet. Um die einzelnen Funktionen verstehen zu können, bedarf es natürlich speziellem Fachwissen, wie beispielsweise zum Web-Service Telnet. Bei der Analyse wurde festgestellt, dass die über den seriellen Port ankommenden Daten in eine Art Ringpuffer gespeichert werden. Bei jedem Durchlauf der Hauptschleife werden alle in diesem Puffer stehenden Daten über die aufgebaute Telnet-Verbindung

gesendet. Um die zukünftige Funktion des Systems sicher zu stellen, war es notwendig, den Ringpuffer und den Zugriff auf diesen neu zu implementieren.

Dazu wurden nach dem Prinzip der *Variantenbildung* mehrere Arten von Ringpuffern (siehe Abbildung 48) untersucht und der erfolgversprechendste Puffer implementiert.

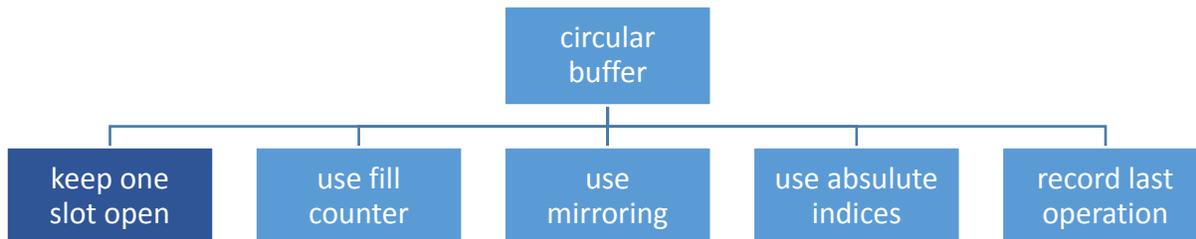


Abbildung 48: Variantenbildung in der Phase der Softwareentwicklung

Die Gewählte Variante zeichnet sich durch besonders kompakten Source aus, wobei keine Datenredundanz nötig ist. Es werden lediglich ein Empfangspuffer und drei Zeiger (*pointer*) auf diesen benötigt. Da der Programmspeicher des gewählten Mikrocontrollers stark begrenzt ist, wurde sich für diese Lösung entschieden.

Schließlich wurde die Lösung in das Telnet-Modul integriert und die Telnet-Funktion getestet. Nachdem dieser Test erfolgreich war, konnte das Telnet-Modul in die Gesamtsoftware integriert werden. Durch wiederholtes Anwenden dieses Zyklus wurde versucht, den gesamten Quellcode zu analysieren und auf die Wünsche und Problemstellungen anzupassen.

5.5.4 Tests

Einzeltests – Der größte Teil der Einzeltests der Hardware wurde bereits in der Detailstudie während der Hardwareentwicklung durchgeführt. Ein Teil der Hardwaretests wurden auch vom Fremdfertiger durchgeführt. Dieser überprüft beispielsweise die einzelnen Verbindungen auf der Leiterplatte auf Durchgang. Dennoch wurde in der Phase des Systembaus auch die Funktion einzelner Baugruppen getestet. Bevor die Leiterplatte mit den integrierten Schaltkreisen und dem Mikrocontroller bestückt wurde, ist so die Funktion der Spannungsregler überprüft worden. So wurde sichergestellt, dass eine eventuelle Überspannung teure Komponenten zerstört.

Noch bevor die Hardware zur Verfügung stand wurden einzelne Softwarekomponenten getestet. Beispielsweise wurde das gezeigte Telnet-Modul auf der Entwicklungshardware überprüft. Dazu wurde

ein einfacher Testaufbau (siehe Abbildung 49) erstellt. Über geeignete Testprogramme (siehe Tabelle 15) wurde die Übertragungsgeschwindigkeit sowie die Fehlerrate übertragener Datenpakete überprüft.

Tabelle 15: Verwendete Software für die Durchführung von Einzeltests

Testprogramm	Zweck	Quelle
HTerm	Serielles Terminal	http://www.der-hammer.info/terminal/
Realterm	Serielles Terminal	http://realterm.sourceforge.net/
Putty	SSH und Telnet Client	http://www.putty.org/
Wireshark	Netzwerkprotokoll Analyse	http://www.wireshark.org/
HW Virtual Serial Port	Virtueller COM Port	http://www.hw-group.com
SerialMon	Datenübertragungs-Analyse	http://www.serialmon.com/

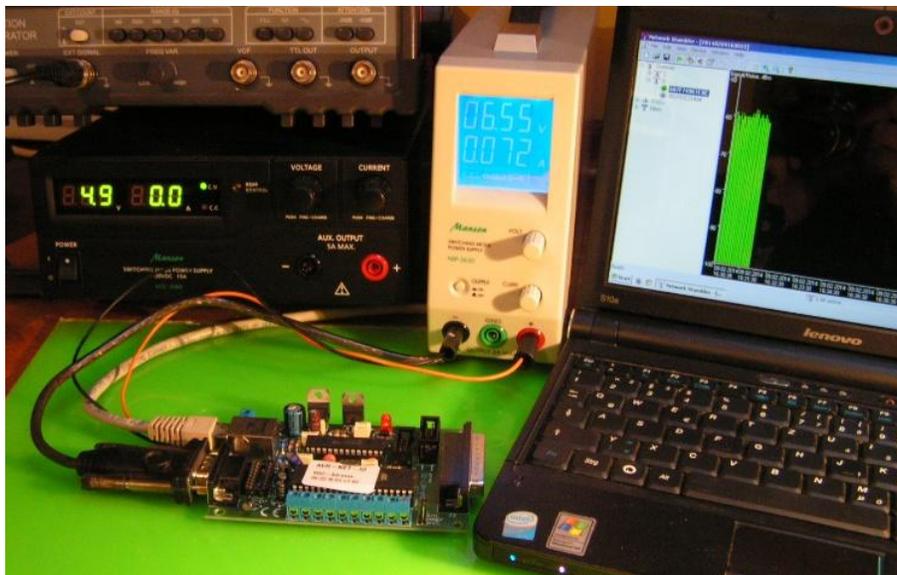


Abbildung 49: Untersuchung der Übertragungsqualität der Telnet-Kommunikation als Beispiel für Einzeltests

Bei diesem Test konnten beispielsweise Fehler in der Implementierung des Telnet-Protokolls festgestellt und ausgebessert werden. Außerdem wurden Verzögerungen der Datenübertragung bei der Verwendung des für Telnet üblichen Netzwerkports festgestellt. Durch diesen Einzeltest konnten zahlreiche Detailprobleme entdeckt und behoben werden.

Systemtests – Systemtests wurden lediglich an einem provisorischen System durchgeführt, weshalb zu diesem Zeitpunkt keine klaren Aussagen zu den Systemtests gemacht werden können. Befürchtete Störeinflüsse durch die Umgebung wie in Kapitel 5.3.2 gezeigt wurden, konnten jedoch nicht festgestellt werden.

5.6 Systemeinführung und Projektabschluss

Nachdem die Systemeinführung und der Projektabschluss zu diesem Zeitpunkt noch ausständig sind, können darüber keine Aussagen gemacht werden. Eine zwischenzeitlich durchgeführte Einschulung des Personals auf das zukünftige System, bei welchem ein provisorisches System verwendet wurde, hat jedoch gezeigt, dass der Auftraggeber mit der Lösung zufrieden ist. Die vorzeitig durchgeführte Einschulung sollte dazu dienen, Mängel am System feststellen zu können und auf Änderungswünsche des Auftraggebers eingehen zu können. Die Qualität wird sich erst nach längerfristigem Einsatz zeigen.

6 Fazit

6.1 Zusammenfassung

Anhand des Fallbeispiels eines mechatronischen Teilsystems wurden verschiedene Prinzipien und Methoden des zuvor gewählten Vorgehensmodells angewendet. Beim Fallbeispiel handelt es sich um ein technisches System, welches dazu dient, gewisse Tätigkeiten an Versuchs- bzw. Prüfständen zu vereinfachen und damit den Prüfablauf zu verbessern. Für die Entwicklung ist nach dem *Hall-BWI*-Modell vorgegangen worden.

Zunächst wurde anhand von Mitarbeitergesprächen ein gewisses Maß an Problembewusstsein geschaffen, was als Anstoß für die Systementwicklung gesehen werden kann.

In einer Vorstudie wurden im Zuge einer Situationsanalyse die derzeitigen Prüfstände analysiert und mithilfe des allgemeinen Modells für mechatronische Systeme der Untersuchungsbereich eingegrenzt. Nach der Prüfstandanalyse wurden diese in Hinblick auf die einzelnen Attribute der Definition mechatronischer Systeme untersucht. Dabei wurde festgestellt, dass diese, auch wenn sie in ihrer Gesamtheit in den Fachbereich der Mechatronik fallen, kein mechatronisches System darstellen. Da diese jedoch nachweislich mechatronischen Charakter aufweisen, kann die Systemgestaltung auf Basis des Modells mechatronischer Systeme erfolgen. Darauf hin wurde neben dem Untersuchungsbereich auch der Gestaltungsbereich abgesteckt. Nach einer näheren Begutachtung des Gestaltungsbereichs und der Frage, wonach eigentlich gesucht wird, wurden drei sehr allgemeine Konzepte erarbeitet, wovon das augenscheinlich sinnvollste Lösungsprinzip weiter verfolgt wurde.

Im Zuge der Hauptstudie wurde gemeinsam mit dem Auftraggeber ein Modell für die Systemarchitektur erstellt, welches sich als hilfreich für das Problem- und Systemverständnis zeigt. Ausgehend von diesem Architekturmodell, welches das System in dessen Systemumgebung darstellt, wurden die systembestimmenden Umsysteme wie Bussystem und Benutzerschnittstelle festgelegt. Durch Bestimmung der Prozessorarchitektur wurde die zentrale Architekturkomponente des Systems ebenso in dieser Phase definiert. Erst danach war ein eigentlicher Blick ins Systeminnere, zur Festlegung des Gesamtkonzepts, möglich. In der folgenden Detaillierungsstufe wurde ein Modell mit den elementaren Bestandteilen des zu entwerfenden Systems und dessen inneren Beziehungen erstellt. Das Modell zeigt, dass für die inneren Zusammenhänge eine genaue Festlegung der Systemkomponenten notwendig ist. Aus diesem Grund wurden in dieser Phase bereits die Sensoren und Kommunikationsports nach dem

Prinzip der *Variantenbildung* und der *stufenweisen Detaillierung* ausgewählt. Dabei konnte beispielsweise eine allgemeine Methode zur Auswahl von Sensoren identifiziert werden, welche unter Umständen eine Produktauswahl erleichtert. Durch Festlegung der verbleibenden Systemkomponenten wurde das Gesamtkonzept geschlossen. Den Abschluss der Hauptstudie stellte die Projektgliederung dar. Es wurde gezeigt, dass ein Zerteilen des Projekts in Teilprojekte anhand der Systemkomponenten nicht sinnvoll zu sein scheint. Insbesondere die Abhängigkeit einzelner Systemkomponenten, wie es auch bei mechatronischen Systemen der Fall ist, erschwert eine solche Teilprojektierung. Für das Fallbeispiel erwies sich eine Gliederung des Projekts nach einzelnen Tätigkeiten als sinnvoll.

In der Detailstudie wurde nach einer weiteren Modelldetaillierung am Beispiel eines konkreten Detailproblems der Problemlösungszyklus in iterativer Weise angewendet. Dabei wurde das Problem der Motorspannungsmessung in fünf Iterationsschritten stufenweise konkretisiert und gelöst. Anschließend wurde selbige Lösung als Beispiel für die Feststellung der Integrierbarkeit herangezogen. Dabei wurde anschaulich demonstriert, wie durch technische Modellbildung und Simulation sowie anschließendem Test des Subsystems die Interoperabilität der einzelnen Elemente festgestellt wurde. Außerdem zeigt das Beispiel die *Bottom-Up* Vorgehensweise bei der Feststellung der Integrierbarkeit im Gegensatz zum sonst üblichen *Top-Down* Vorgehen bei der Konkretisierung und Detaillierung des Systems.

Die Phase des Systembaus wurde in Hardwarerealisierung und Softwarerealisierung gegliedert. Bei der Realisierung der Hardware wurde gezeigt, welches konkrete methodische Vorgehen für den Entwurf der Elektronik angewendet wurde. Es zeigt sich, dass die Elektronikentwicklung einerseits durch ein *Meet-in-the-Middle* Prinzip geprägt ist, als auch dass die Konstruktion sich ähnlich der Modellbildung gestaltet. Schaltpläne und Leiterplatten stellen prinzipiell ein Modell von besonders hohem Detaillierungsgrad dar. Außerdem konnte für den Entwurf von Leiterplatten ein spezielles Phasenmodell identifiziert werden, welches schließlich auch zu einem sehr überschaubaren Design führten.

Nachdem bei der Softwarerealisierung ein bereits existierender Quellcode die Ausgangsbasis darstellte, wurde auch hier eine Methode angewandt, um diesen Code an die Problemstellung anzupassen. Die gefundene Methode hat sich im Zuge des Systembaus als erfolgsversprechend erwiesen. Durch zyklisches Anwenden von Analyse und Synthese wurde der Quellcode untersucht und an diverse Anforderungen angepasst. Am Beispiel der Kommunikation über das Telnet-Protokoll wurde diese Herangehensweise demonstriert. Abschließend wurden auch noch Einzeltests und Systemtests durchgeführt. Die Systemeinführung und der Projektabschluss sind jedoch noch ausständig.

6.2 Stellungnahme

Zunächst soll festgehalten werden, dass das der Arbeit zugrunde liegende Projekt weitestgehend erfolgreich abgewickelt werden konnte. Auch wenn im Fallbeispiel nicht das gesamte Projekt in allen Details dargestellt werden konnte, so wurden die wesentlichen Methoden, welche zur Problemlösung beigetragen haben, gezeigt.

Wie auch zu erkennen ist, konnten die im Vorgehensmodell hervorgehobenen Denkweisen in nahezu allen Projektphasen angewendet werden. Gerade das Vorgehen *vom Groben ins Detail* wird, wenn auch in der Praxis oft unbewusst, dennoch sehr häufig angewendet. Wenn auch in der Theorie dieses Vorgehen vom Prinzip *der Bildung von Varianten* getrennt betrachtet wird, sind diese beiden Handlungsgrundsätze in der Praxis kaum trennbar.

Wie sich zeigte, waren vor allem auch bei der Produktrecherche die *Variantenbildung* und das *Top-Down* Vorgehen die am häufigst gegangenen Wege. In einigen wenigen Fällen jedoch erwies sich diese Herangehensweise unvorteilhaft, da es durch häufiges revidieren des gewählten Lösungsprinzips zu einer verzögerten Lösungsfindung führte. Hier können Erfahrungen und damit die Anwendung eines *Bottom-Up* Vorgehens schneller zu guten Lösungen führen.

Das Fallbeispiel zeigt, dass eine Systembetrachtung wie auch die Systemmodellierung in vielen Fällen unausweichlich angewendet wird. So wäre im Fallbeispiel spätestens in der Phase des Systembaus durch die Schaltplankonstruktion ein bewusstes Detailmodell der Hardware erstellt worden. So ist die Modellbildung seit jeher Bestandteil jeder Konstruktion, auch wenn hier die Begriffe *Skizze* oder *Plan* an die Stelle des *Modells* tritt.

Auch wenn im vorliegenden Beispiel bewusst Wert auf eben beschriebene Grundprinzipien des Handelns gelegt wurde, so finden in der Lösungsfindung auch andere Prinzipien immer wieder unbewusst Anwendung. Exemplarisch möchte ich hierbei das Grundprinzip der Problemzerlegung anmerken, welches sich auch im Anwendungsbeispiel für den Problemlösungszyklus deutlich erkennen lässt. Das Beispiel zeigt, dass hier in den Iterationsschritten Teilprobleme gelöst wurden, welche in ihrer Summe das Gesamtproblem darstellen.

Dieser Umstand lässt auch an der Bezeichnung *Problemlösungszyklus* gewisse Kritik aufkommen. So muss es sich nicht zwangsläufig um einen Zyklus handeln, sondern es weist vielmehr einen spiralförmigen Charakter auf, bei dem, ausgehend von einem Gesamtproblem, durch mehrmaliges Durchlaufen der selben Schritte (Quadranten) Teilprobleme geklärt werden, um schließlich zu einer sinnvollen

Gesamtlösung zu kommen. Damit möchte ich ausdrücken, dass man am Ende eines Durchlaufs des Problemlösungszyklus nicht wieder am Anfang des Problemlösungsprozesses steht, sondern sich bereits einen Schritt auf die Lösung zubewegt hat. Dies gilt, selbst wenn sich durch den Prozess lediglich das Problem verlagert hat. Jedoch ändert diese Auffassung nichts am Prinzip des Problemlösungszyklus.

Im Zuge der Erarbeitung des praktischen Beispiels erwies sich der Problemlösungszyklus auch nicht geeignet zur Untergliederung der einzelnen Projektphasen, wie die graphische Darstellung des Vorgehensmodells es vermuten ließe. Auch wenn das Vorgehensmodell eine solche Untergliederung nicht propagiert, könnte es hier mitunter durch die Bezeichnung *Makro-* und *Mikrologik* zu Missverständnissen kommen. Aus diesem Grund wurden auch einzelne allgemeine Tätigkeiten (z.B. dem Entwurf der Systemarchitektur oder des Gesamtkonzepts) zur Untergliederung des Phasenmodells herangezogen. Der Problemlösungszyklus ist in dargestellter Form vorwiegend für die Lösung von Detailproblemen verwendet worden.

In diesem Zusammenhang zeigte sich eine weitere Auffälligkeit am gewählten Modell. In der Literatur wurde erklärt, dass die ganzheitliche Situationsanalyse einer zu stark ist- oder sollorientierten Betrachtung vorzuziehen ist. Hier zeigten sich im Problemlösungsprozess jedoch gewisse Diskrepanzen. Im Gegensatz zur Situationsanalyse stellt eine Ist-Analyse eine problemneutrale Betrachtung der Ist-Situation dar, ähnlich der lösungsneutralen Betrachtung bei der Problemidentifikation. In der Situationsanalyse findet hingegen bereits eine Problemdarstellung statt. Eine solche Betrachtung führt jedoch im folgenden Prozess der Zielformulierung zu gewisser Redundanz. Das soll heißen, dass die Zielformulierung einer Art Umformulierung der Problemstellung gleicht. Die Frage die ich damit aufbringen möchte, lautet: „Ist es nicht stets Ziel das Problem zu lösen?“.

Eine weitere Auffälligkeit zeigte sich im Prozess der Gliederung des Projekts in Teilprojekte. So erwies sich ein Gliedern des Projekts in Teilprojekte für das Fallbeispiel als schwierig. Aus diesem Grund wurde auch auf eine Detaillierung der Projektphasen anhand notwendiger Tätigkeiten zurück gegriffen. Dies war besonders dadurch hilfreich, da die Aneignung der benötigten Fachkenntnisse und Erfahrungswerte aus den einzelnen Phasen in Schritten möglich war. Aber nicht nur für das vorliegende Fallbeispiel ist die Gliederung des Projekts in Teilprojekte mit Problemen behaftet.

Hier kann auch ein allgemeiner Zusammenhang zu mechatronischen Systemen entdeckt werden. Wie gezeigt wurde, zeichnen sich mechatronische Systeme unter anderem durch ihre integrierte Struktur aus. Das bedeutet, die einzelnen Elemente sind wesentlich in ihrer Funktion von ihren Nachbarelementen

abhängig. Anders ausgedrückt, ein mechatronisches System ist kein System von Systemen, es weist kaum modularen Charakter auf. Die Funktion mechanischer Komponenten ist wesentlich durch die elektrischen und informationsverarbeitenden Teile abhängig und umgekehrt. Das Fehlen einer modularen Struktur erschwert eine Gliederung des Projekts in Teilprojekte wie auch die Analysen und Tests der einzelnen Komponenten.

Diesem Aspekt lasse ich eine besondere Bedeutung zukommen, welcher, so scheint mir, beim Entwurf von Systemen viel zu häufig außer Acht gelassen wird. Wie bereits mehrfach angemerkt, ist das Ganze mehr als nur die Summe einzelner Elemente. Hingegen propagieren die meisten SE-Konzepte die Vorteile eines modularen Aufbaus oder auch Vorteile eines Systems von Systemen. Dies wird gestärkt durch Handlungsprinzipien wie dem Grundprinzip der *Mehrfachverwendbarkeit*, dem der *Standardisierung* aber auch jenem der *Problemzerlegung*. Dies führt zwangsläufig aber dazu, dass der Blick fürs Ganze verloren gehen kann. Ein komplexes Problem kann nicht immer auf seine Teilprobleme herunter gebrochen, im Detail gelöst und anschließend wieder zur Gesamtlösung zusammen gesetzt werden. Das bedeutet auch, dass die beste Gesamtlösung nicht aus der Summe der besten Einzellösungen besteht. Die Verknüpfungen zwischen den Elementen bestimmen wesentlich die Qualität der Gesamtlösung bzw. des Gesamtsystems.

Diese Tatsache zeigt sich in gewisser Weise auch im Fallbeispiel. Hier war die Gestalt der Modelle besonders stark von den Schnittstellen und Verbindungen der Elemente bzw. Subsysteme abhängig. Ohne Festlegung der Verbindungen konnten die Systemmodelle nicht weiter detailliert werden. Aus diesem Grund wurden auch gewisse Tätigkeiten, welche man üblicherweise späteren Phasen zuordnen würde, bereits in der Vor- und Hauptstudie durchgeführt. Hier erwies sich der plangetriebene Charakter des *Hall-BWI*-Konzepts als hinderlich. Aus diesem Grund wurden in der Vorstudie bereits viele Detailfragen beantwortet, zum Beispiel um eine Aussage über die Machbarkeit treffen zu können. Auch für den Entwurf der Systemarchitektur und des Gesamtkonzepts bedarf es einer Klärung vieler Detailfragen. Dies führte wiederum dazu, dass bereits am Ende der Hauptstudie die wesentlichen Gestaltungsvarianten festgelegt waren. Auch wenn eine Rückkopplung der einzelnen Projektphasen zu vorherigen Phasen im Modell angemerkt ist, so scheint die Makrologik im Gegensatz zur Mikrologik als wenig flexibel.

Zusammengefasst ließ sich zeigen, wie das SE-Konzept auch zur Entwicklung mechatronischer Systeme verwendet werden kann, wenn auch das betrachtete System nach Definition nicht als solches einzuordnen ist. Durch das systematische Vorgehen konnte die Aufgabenstellung zur bisherigen Zufriedenheit gelöst und die gestellten Ziele, erreicht werden.

6.3 Ausblick

Wie gut sich das im Laufe des Projekts erstellte System in seinem Anwendungsgebiet als wertvoll erweist, wird sich erst in der anschließenden Systemeinführung und der laufenden Anwendung zeigen. Außerdem werden sich für die Zukunft neue ähnliche Projekte ergeben, welche vielleicht auch das erarbeitete System obsolet machen.

Tatsache ist, dass die Komplexität industrieller Produkte auch in Zukunft weiterhin zunehmen wird, wodurch auch die Anforderungen an die Entwicklungskonzepte steigen. Dadurch wird vermutlich die Flexibilität der Modelle ein wesentlicher Erfolgsfaktor sein. Aber auch die zukünftige Technik wird die Entwicklungsprozesse vermehrt unterstützen. Zukünftige Methoden, Techniken und Werkzeuge werden dabei helfen, den Architekturentwurf, das Design wie auch die Ingenieurstätigkeiten zu verbessern und zu vereinfachen. Möglicherweise lassen sich dadurch auch bessere Grenzen zwischen diesen einzelnen Rollen, Aufgaben oder Fachbereichen ziehen und diese noch nicht ganz geklärte Frage beantworten.

Auch das Fachgebiet der Mechatronik wird in Zukunft nicht an Bedeutung verlieren. Vermehrt werden uns mechatronische Systeme in unserem Alltag begegnen und diesen hoffentlich erleichtern. SE hat längst Einzug in dieses Fachgebiet gehalten. Dennoch wird sich zeigen, welche konkreten SE-Konzepte sich hier in Zukunft als besonders günstig erweisen aber auch, wie diese sich entwickeln werden. Ebenso werden sich vermutlich allgemeine Methoden speziell für die Entwicklung mechatronischer Systeme entwickeln, die sich prägend auf das SE auswirken werden. Immerhin wird sich auch in Zukunft nichts daran ändern, dass im Systems Engineering die Methode ein zentrales Bindeglied zwischen Problem und Lösung darstellen wird.

In diesem Sinne möchte ich mit einem Zitat aus dem Werk „Regeln zur Leitung des Geistes“ des französischen Philosophen, Mathematiker und Naturwissenschaftler René Descartes enden, welches lautet:

„Zur Erforschung der Wahrheit bedarf es notwendig der Methode.“ (Descartes & Buchenau, 1906, S. 15)

Literaturverzeichnis

- Bahill, T., & Gissing, B. (1998). Re-evaluating Systems Engineering Concepts Using Systems Thinking. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 28, NO. 4*, 516-527.
- Bandte, H. (2007). *Komplexität in Organisationen*. Wiesbaden: Deutscher Universitäts Verlag | GWV Fachverlage GmbH.
- Bardmann, M. (2011). *Grundlagen der Allgemeinen Betriebswirtschaftslehre*. Wiesbaden: Gabler Verlag | Springer Fachmedien Wiesbaden GmbH.
- Becker, J. (2012). *Grundsätze ordnungsmäßiger Modellierung*. Berlin Heidelberg: Springer-Verlag.
- Bertalanffy, L. v. (1950). The Theory of Open Systems in Physics and Biology. *SCIENCE*.
- Bossel, H. (2004). *Systeme, Dynamik, Simulation*. Norderstedt: BoD – Books on Demand.
- Czichos, H. (2008). *Mechatronik*. Wiesbaden: Vieweg+Teubner Verlag | GWV Fachverlage GmbH.
- Czichos, H., & Hennecke, M. (2012). *HÜTTE - Das Ingenieurwissen*. Berlin Heidelberg: Springer.
- Daenzer, W. F. (1976/77). *Systems engineering: Leitfaden zur methodischen Durchführung umfangreicher Planungsvorhaben*. Köln; Zürich: Hanstein; Verlag Industrielle Organisation.
- Descartes, R., & Buchenau, A. (1906). *Philosophische Werke: Abhandlung über die Methode. Regeln zur Leitung des Geistes. Die Erforschung der Wahrheit durch das natürliche Licht*. Leipzig: Dürr'sche Buchhandlung.
- DOD. (2001, Jänner). *SYSTEMS ENGINEERING FUNDAMENTALS*. FORT BELVOIR, VIRGINIA: DEFENSE ACQUISITION UNIVERSITY PRESS. Retrieved from [mit: ocw.mit.edu/courses/aeronautics-and-astronautics/16-885j-aircraft-systems-engineering-fall-2005/readings/sefguide_01_01.pdf](http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-885j-aircraft-systems-engineering-fall-2005/readings/sefguide_01_01.pdf)
- DUDEN. (2014, Jänner 01). *Lösung*. Retrieved from <http://www.duden.de/rechtschreibung/Loesung>
- DUDEN. (2014, Jänner 01). *System*. Retrieved from <http://www.duden.de/rechtschreibung/System>
- DUDEN. (2014a, Jänner 01). *Kybernetik*. Retrieved from <http://www.duden.de/rechtschreibung/Kybernetik>
- DUDEN. (2014b, Jänner 01). *Modell*. Retrieved from <https://www.duden.de/rechtschreibung/Modell>

- DUDEN. (2014c, Jänner 01). *Problem*. Retrieved from <http://www.duden.de/rechtschreibung/Problem>
- DUDEN. (2014d, Jänner 01). *Methode*. Retrieved from <http://www.duden.de/rechtschreibung/Methode>
- DUDEN. (2014e, Jänner 01). *Management*. Retrieved from <http://www.duden.de/rechtschreibung/Management>
- DUDEN. (2014f, Jänner 01). *Architektur*. Retrieved from <http://www.duden.de/rechtschreibung/Architektur>
- Eley, M. (2012). *Simulation in der Logistik*. Berlin Heidelberg: Springer.
- Engels, G. (2012, Oktober 31). *Modellierungssprache*. Retrieved from Enzyklopaedie der Wirtschaftsinformatik: <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/technologien-methoden/Sprache/Modellierungssprache>
- Gajski, D. D., Abdi, S., Gerstlauer, A., & Schirner, G. (2009). *Embedded System Design*. LLC: Springer Science+Business Media.
- Haberfellner, R., Weck, O. d., Fricke, E., & Vössner, S. (2012). *Systems Engineering, Grundlagen und Anwendung*. Zürich: Orell Füssli Verlag.
- Hofkirchner, W. (2005). Forerunner of Evolutionary Systems Theory. *IFSR 2005 : Proceedings of the First World Congress of the International Federation of Systems Research*, 6.
- INCOSE. (2006, März 13). *Brief History of Systems Engineering*. Retrieved from incose: <http://www.incose.org/mediarelations/briefhistory.aspx>
- INCOSE. (2010). *INCOSE Systems Engineering Handbook v. 3.2*. San Diego, CA: SE Handbook Working Group, International Council on Systems Engineering.
- INCOSE. (2013, Dezember 2013). *Home: International Council on Systems Engineering (INCOSE)*. Retrieved from International Council on Systems Engineering (INCOSE): <http://www.incose.org/>
- Isermann, R. (2008). *Mechatronische Systeme*. Berlin Heidelberg: Springer-Verlag.
- Janschek, K. (2010). *Systementwurf mechatronischer Systeme*. Berlin Heidelberg: Springer-Verlag.
- Kloth, C. (2012). *Systemgestaltung im Broadcast Engineering*. Wiesbaden: Vieweg und Teubner Verlag | GWV Fachverlage GmbH.

- Kohorst, H. (1996, Oktober 01). *Glossar: Definition zentraler Begriffe*. Retrieved from <http://www.kohorst-lemgo.de/modell/bevchina/glossar.htm>
- Liening, A. (1999). *Komplexe Systeme zwischen Ordnung und Chaos*. Münster: LIT.
- Lindemann, U. (2009). *Methodische Entwicklung technischer Produkte*. Berlin Heidelberg: Springer-Verlag.
- LOGICDATA. (2014, Jänner 01). Retrieved from logicdata.at: <http://www.logicdata.at/>
- Maier, M. W., & Rechtin, E. (2000). *THE ART OF SYSTEMS ARCHITECTING 2nd ed.* Boca Raton, London, New York, Washington D.C.: CRC Press LLC.
- Masak, D. (2005). *Moderne Enterprise Architekturen*. Springer-Verlag Berlin Heidelberg.
- Masak, D. (2010). *Der Architekturreview*. Wetzlar, Germany: Springer Berlin Heidelberg.
- Mori, T. (1969, Juli 12). *Mechatronics*. Yaskawa Internal Trademark Application Memo.
- Muller, G. (2013). *System Architecting*. Kongsberg, Norway: Buskerud University College.
- NASA. (2007). *Systems Engineering Handbook*. Washington, D.C. : NASA Headquarters .
- OMG. (2012, Juni). *OMG Systems Modeling Language V1.3*. Retrieved from Object Management Group: <http://www.omg.org/spec/SysML/1.3/PDF/>
- Patzak, G. (1982). *Systemtechnik – Planung komplexer innovativer Systeme*. Berlin : Springer.
- PONS. (2014, Jänner 01). *engineering*. Retrieved from <http://de.pons.com/übersetzung?q=engineering&l=deen&in=&lf=de>
- PONS. (2014a, Jänner 01). *design*. Retrieved from <http://de.pons.com/übersetzung?q=design&l=deen&in=&lf=de>
- Rieck, C. (1993). *Spieltheorie, Einführung für Wirtschaftsund Sozialwissenschaftler*. Wiesbaden: Gabler Verlag | Springer Fachmedien Wiesbaden GmbH.
- Roddeck, W. (2006). *Einführung in die Mechatronik*. Wiesbaden: B.G.Teubner Verlag / GWV Fachverlage GmbH.
- Rohde, T. (2008, April). *mechatronics*. Retrieved from <http://www.yaskawa.com/site/products.nsf/staticPagesNewWindow/mechatronics.html>

- Schlager, K. J. (1956). Systemas Engineering-Key to Modern Development. *IRE TRANSACTIONS ON ENGINEERING MANAGEMENT*, 64-66.
- Schneider, G., & Vecellio, S. (2011). *ICT-Systemabgrenzung, Anforderungsspezifikation und Evaluation*. Compendio Bildungsmedien.
- Schönsleben, P. (2007). *Integrales Logistikmanagement*. Berlin Heidelberg: Springer-Verlag .
- Sheard, S. A. (1996). *TWELVE SYSTEMS ENGINEERING ROLES*. Boston, Massachusetts, USA: Software Productivity Consortium, NRP, Inc.
- SMC. (2005). *SMC Systems Engineering Primer & Handbook, 3rd Edition*. USA: Space & Missile Systems Center, U.S. Air Force.
- Smith, R. J. (2013). *engineering*. Retrieved from Encyclopædia Britannica, Inc.: <http://www.britannica.com/EBchecked/topic/187549/engineering>
- Stein, B. (2003/04, 04 11). *System, Model and Deduction*. Retrieved from <http://www2.cs.uni-paderborn.de/cs/ag-klbue/de/courses/ss05/gwbs/model-ss05-slides.ps.nup.pdf>
- Thomas, O. (2005). *Das Modellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation*. Saarbrücken: Instituts für Wirtschaftsinformatik Prof. Dr. Dr. h.c. mult. August-Wilhelm Scheer.
- Tietze, U., & Schenk, C. (2002). *Halbleiter-Schaltungstechnik*. Berlin, Heidelberg: Springer-Verlag.
- TransMechatronic. (2014, Jänner 31). *Mechatronik*. Retrieved from <http://www.transmechatronic.de/index.php?id=119>
- Ulrich, H., & Probst, G. J. (1995). *Anleitung zum ganzheitlichen Denken und Handeln* . Bern: Paul Haupt.
- Winzer, P. (2013). *Generic Systems Engineering*. Berlin Heidelberg: Springer-Verlag.

Anhang

A 1. Detailansicht des Schaltplans

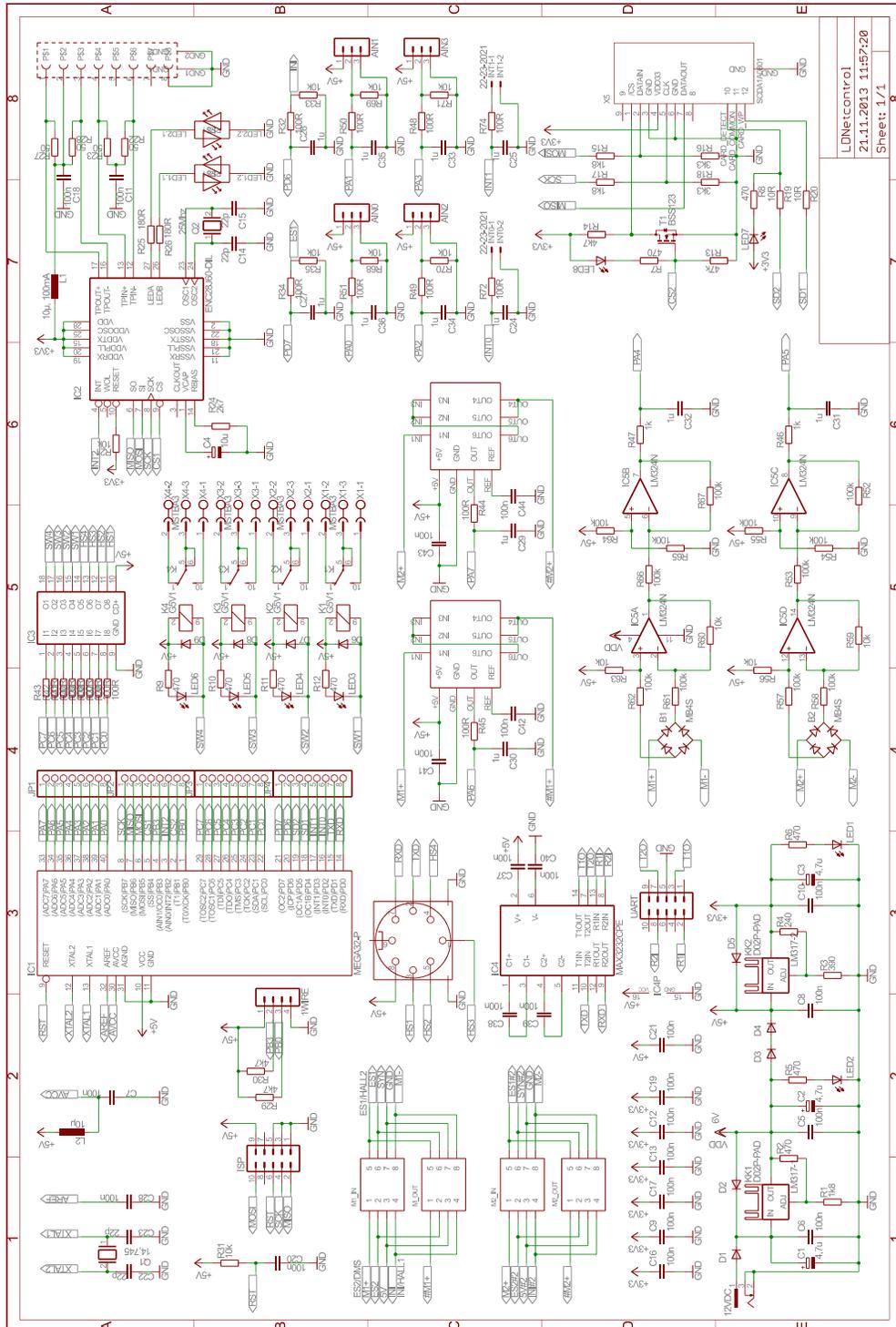


Abbildung 50: Schaltplan

A 2. Detailansicht des Designs der Leiterplatte

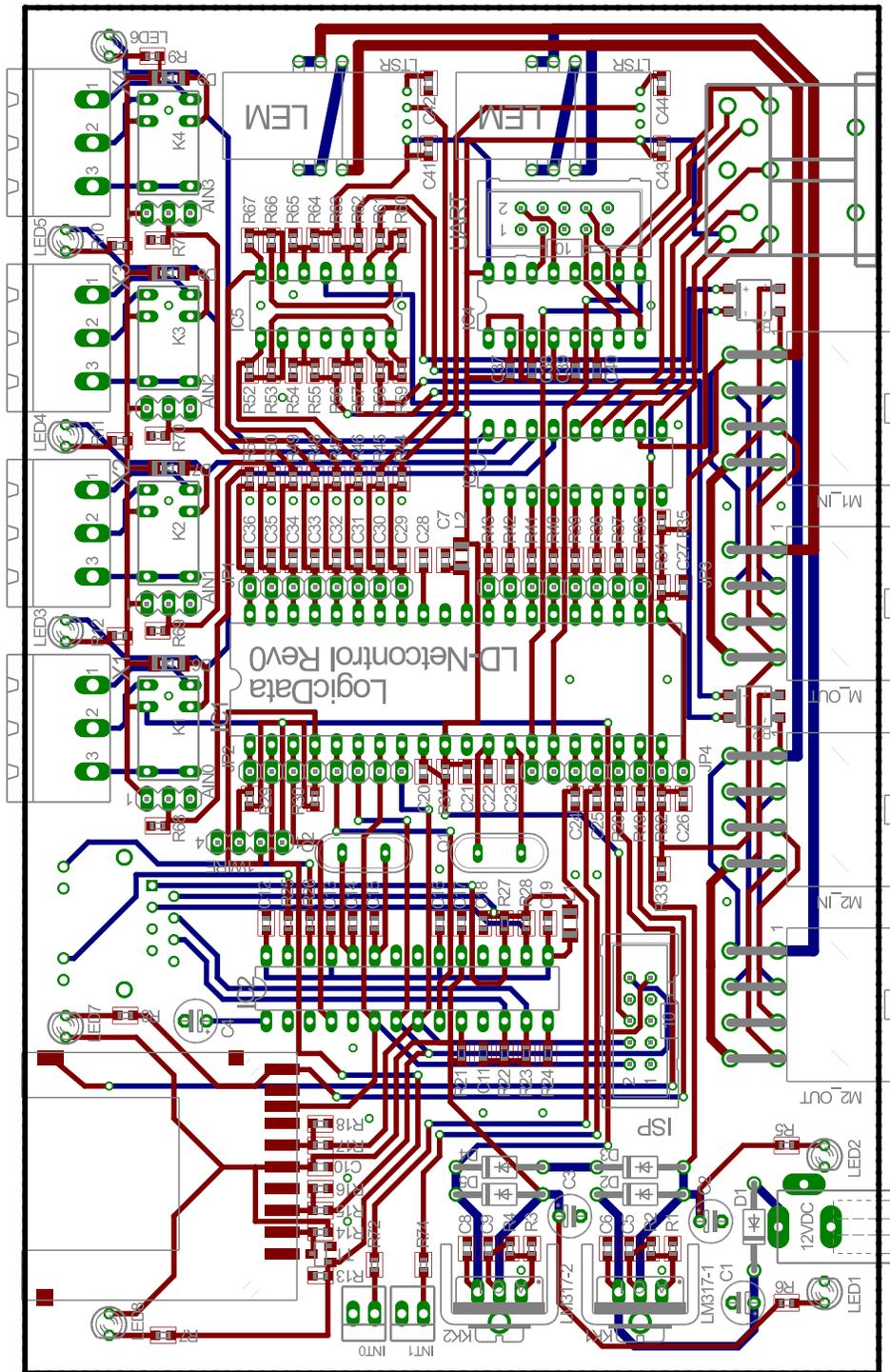


Abbildung 51: Leiterplattendesign

A 3. Design der Benutzerschnittstelle

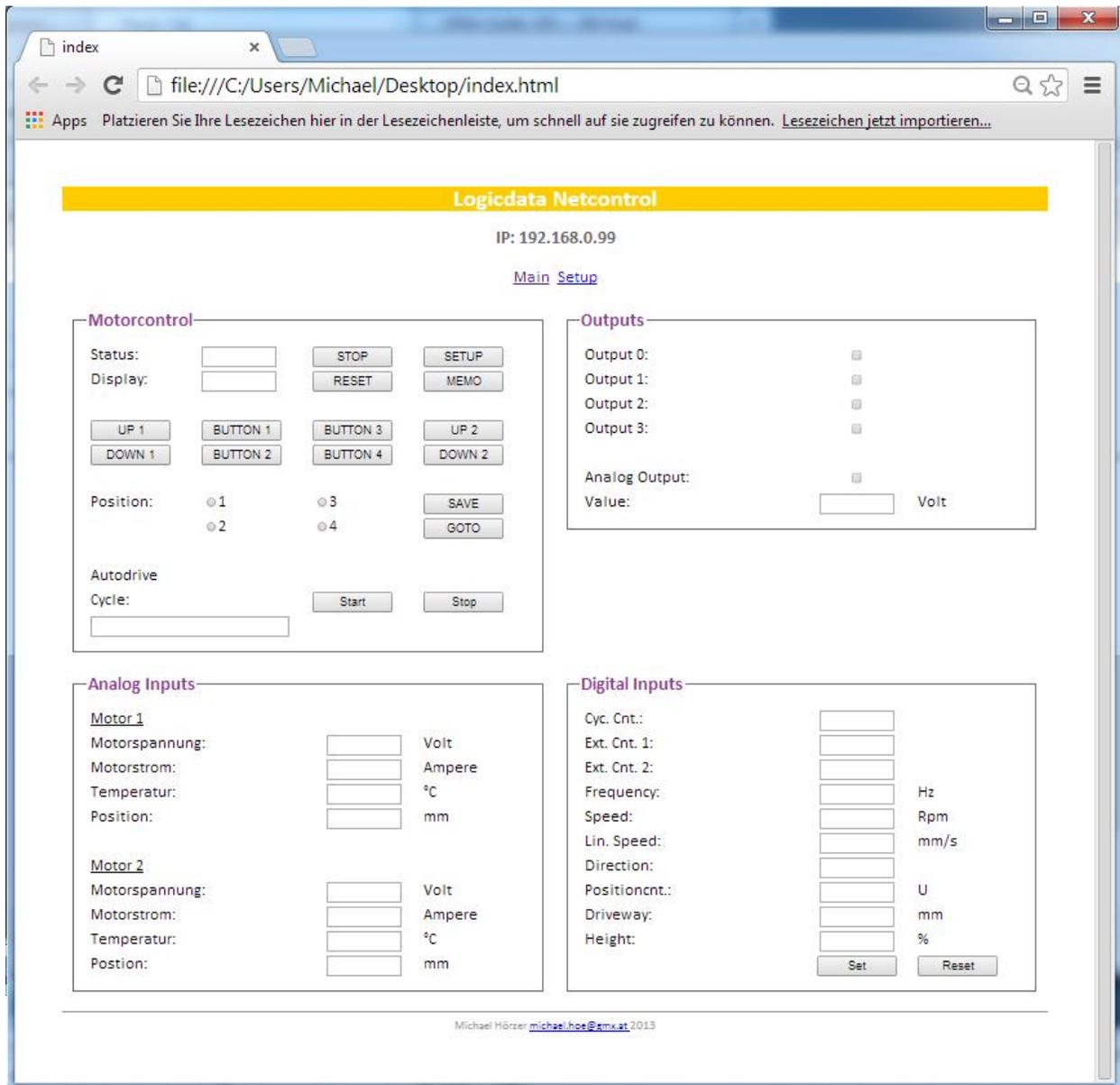


Abbildung 52: User interface design index.html

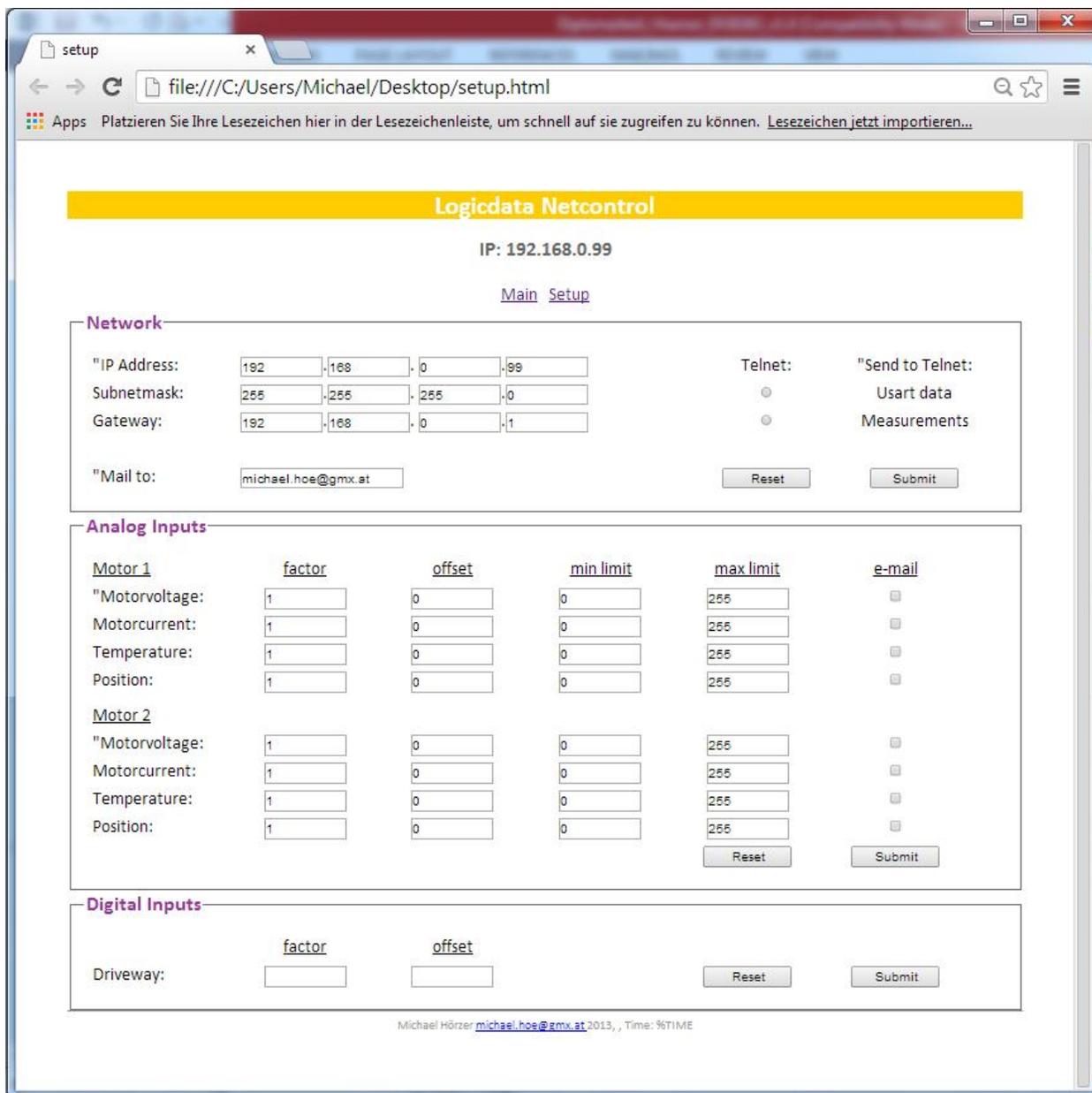


Abbildung 53: User interface design setup.html