

Masterarbeit

**MODELLGESTÜTZTE VIRTUELLE INBETRIEBNAHME
VON
KAROSSERIEBAUANLAGEN UNTER
BERÜCKSICHTIGUNG DES BESTEHENDEN
PRODUKTENTSTEHUNGSPROZESSES**

ausgeführt am



FACHHOCHSCHULE DER WIRTSCHAFT

Fachhochschul-Masterstudiengang
Automatisierungstechnik-Wirtschaft

von

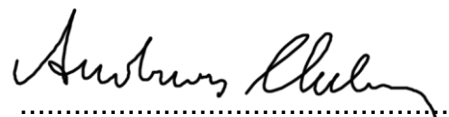
Ing. Andreas Huber, BSc

1710322024

betreut und begutachtet von

DI Johannes Fritz, BSc

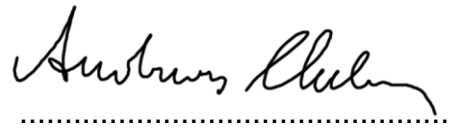
Graz, im Jänner 2019



Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

A handwritten signature in black ink, appearing to read 'Andreas Klein', is written above a horizontal dotted line.

Unterschrift

GLEICHHEITSGRUNDSATZ

Um den Lesefluss nicht durch die ständige Nennung beider Geschlechter zu stören, wird in dieser Arbeit ausschließlich die männliche Form verwendet. Dies impliziert aber immer auch die weibliche Form.

DANKSAGUNG

An dieser Stelle möchte ich mich zu allererst bei meiner lieben Frau Andrea bedanken. Sie unterstützte mich in den letzten fünf Jahren, wo es nur ging. Nur dadurch war es mir möglich, als Vater von zwei Kindern einer Berufstätigkeit in Vollzeit nachzugehen sowie zeitgleich ein Studium zu absolvieren. Ohne ihren selbstlosen Einsatz für unsere Familie wäre dieser Erfolg nicht möglich gewesen.

Ihr geduldiges Verständnis gegenüber der Tatsache, dass viele Wochenenden für das Studium zu investieren waren, gab mir den Rückhalt um dieses erfolgreich abzuschließen. Durch ihre ständige Unterstützung war es mir möglich die Zeit aufzubringen, die für das Lernen für Klausuren, das Abarbeiten von Übungen, das Verfassen von Berichten und letztendlich auch das Schreiben dieser Masterarbeit nötig war. Ebenso hat sie sich in dieser Zeit hingebungsvoll um die Erziehung unserer Kinder gekümmert, das durch die regelmäßige Abwesenheit meinerseits nicht immer einfach war. Dank dieser Hingabe haben die beiden immer ein Umfeld vorgefunden, das von Liebe, Herzlichkeit und Fürsorge geprägt war, auch wenn es ab und an mal galt stürmischere Zeiten zu durchleben. Auch für die dauernde Mehrbelastung im Haushalt und der damit verbundenen Nachsicht mir gegenüber möchte ich mich an dieser Stelle ausdrücklich bedanken. Durch diesen Einsatz habe ich stets ein Umfeld vorgefunden, das nötig war, um jene Leistungen abrufen zu können, die für das Bewältigen des Studiums erforderlich waren.

Daher verdanke ich den Erfolg, den ich in den letzten Jahren erleben durfte, größtenteils ihr.

Auch möchte ich mich bei meinem Arbeitgeber, der Magna Steyr Fahrzeugtechnik, bedanken. Durch das großzügige Entgegenkommen dieses Unternehmens war es mir erlaubt, an allen Vorlesungen im Rahmen des Studiums teilzunehmen. Ebenso stellte die thematische Zuordnung dieser Masterthesis zu meinen beruflichen Aufgaben eine große Hilfe dar. Auch möchte ich im Zuge dessen der Instandhaltung der H3/H81 ein herzliches Dankeschön für die Montage des VIBN-Testkits aussprechen.

Zu guter Letzt gilt mein Dank Herrn DI Johannes Fritz, BSc, der mir als Betreuer für diese Masterarbeit und während der Erstellung dieser stets mit Rat und Tat zur Seite stand. Nicht zuletzt durch seine konstruktive Kritik, sorgfältige Korrektur wie auch stetiges Hinterfragen der Inhalte dieser Arbeit konnte ich diese auch zu einem positiven Abschluss bringen. Durch sein zuvorkommendes und hilfsbereites Wesen war mir die Zusammenarbeit mit ihm stets ein Vergnügen.

KURZFASSUNG

Aufgrund kürzer werdender Zeiträume zur Realisierung von hochautomatisierten, robotergestützten Fertigungsanlagen im Karosseriebau, stoßen traditionelle Inbetriebnahmemethoden immer öfter an ihre Grenzen. Daher hat die vorliegende Masterarbeit die wissenschaftliche Auseinandersetzung mit dem Themenfeld der Virtuellen Inbetriebnahme zum Inhalt. Ziel der Forschungsaktivitäten ist die Entwicklung eines Prototyps, der die Anforderungen von Magna Steyr in diesem Themenbereich erfüllt. Diese sind einerseits eine Methode respektive ein Werkzeug zu schaffen, das die Verifikation und Validierung von steuerungstechnischen Entwicklungen ohne real bestehende Produktionsanlage ermöglicht. Andererseits ist mit der zu schaffenden Vorgehensweise ein System zu entwickeln, das modellgestützt die Analyse komplexer Störszenarien und Ausfälle von Produktionsanlagen im Karosseriebau erlaubt, ohne den bestehenden Fertigungsprozess zu beeinträchtigen. Das zu entwickelnde Engineeringwerkzeug ist so zu gestalten, dass weitestgehend originale Steuerungskomponenten und nativer Programmcode eingesetzt werden können. Um die forschungsleitenden Fragestellungen zu beantworten sowie die von Magna Steyr vorgegebenen Entwicklungsziele zu erreichen, werden in dieser Masterarbeit die Bereiche Modellbildung und Virtuelle Inbetriebnahme einer theoretischen Auseinandersetzung und Analyse unterzogen. Dies bildet die Grundlage für die im Praxisteil der Masterarbeit zu entwickelnde Methode zur virtuellen Absicherung von hochautomatisierten Karosseriebauanlagen.

ABSTRACT

Due to shorter periods of time which are available for the realization of automated, robot based production systems in the body shop, traditional commissioning methods are increasingly reaching their limits. Therefore, the present master's thesis analyses the field of virtual commissioning in a scientific approach. The aim of the research activities is the development of a prototype that meets the requirements of Magna Steyr in this area. On the one hand it is necessary to create a method respectively a tool, which enables the verification and validation of control technology developments without a real existing production facility. On the other hand it is demanded to develop a solution that allows a model-based analysis of complex sturgeon scenarios and failures of production facilities in body shops without adversely affecting the existing production process. The required engineering tool has to be designed in a way that ensures the use original control components and native program code as far as possible. In order to answer the research-related questions as well as to achieve the development goals set by Magna Steyr, the areas of modeling and virtual commissioning are subjected to a theoretical examination and analysis in this master's thesis. This forms the basis for the method to be developed in the practical part of the master's thesis for the virtual protection of highly automated body construction facilities.

INHALTSVERZEICHNIS

| | | |
|---------|--|----|
| 1 | Einleitung..... | 1 |
| 2 | Allgemeine Modellbildung | 3 |
| 2.1 | Modell – Begriffsbestimmung und Abgrenzung..... | 3 |
| 2.1.1 | Begriffsbestimmung nach Stachowiak..... | 3 |
| 2.1.2 | Begriffsbestimmung nach Bossel | 4 |
| 2.1.3 | Begriffsbestimmung nach Kastens und Kleine-Büning | 5 |
| 2.1.4 | Begriffsbestimmung nach VDI 3633 | 5 |
| 2.1.5 | Abgrenzung..... | 5 |
| 2.2 | Modellbildung..... | 6 |
| 2.3 | Normative Vorgaben in der Modellbildung | 8 |
| 2.3.1 | VDI 4465 Blatt 1 - Modellierung und Simulation / Modellbildungsprozess | 8 |
| 2.3.2 | VDI 2206 - Entwicklungsmethodik für mechatronische Systeme | 9 |
| 2.3.3 | VDI 3843 Blatt 1 - Modellbildung schwingungsfähiger Systeme | 10 |
| 2.4 | Methoden der Modellbildung | 11 |
| 2.4.1 | Analytische Modellbildung | 11 |
| 2.4.2 | Experimentelle Modellbildung..... | 12 |
| 2.5 | Schritte der Modellbildung | 13 |
| 2.5.1 | Systemmodellierung und Systemanalyse nach Bossel..... | 13 |
| 2.5.2 | Schritte der Modellbildung nach VDI 4465 Blatt 1 | 16 |
| 2.6 | Modellformen | 17 |
| 2.6.1 | White-Box-Modell | 17 |
| 2.6.2 | Grey-Box-Modell..... | 18 |
| 2.6.3 | Black-Box-Modell..... | 19 |
| 2.7 | Einsatzzweck von Modellen..... | 20 |
| 2.8 | Abstraktionsprinzipen | 22 |
| 2.8.1 | Begriffsbestimmung..... | 22 |
| 2.8.2 | Prinzipien der Abstraktion | 22 |
| 2.8.2.1 | Aggregation/Partizipation | 22 |
| 2.8.2.2 | Generalisierung/Spezialisierung..... | 23 |
| 2.8.2.3 | Klassifizierung/Instanziierung | 24 |
| 2.8.2.4 | Komposition/Dekomposition | 25 |
| 2.8.2.5 | Abgrenzung | 26 |
| 2.9 | Werkzeuge zur Modellbildung | 26 |
| 2.10 | Zusammenfassung | 28 |
| 3 | Virtuelle Inbetriebnahme | 29 |
| 3.1 | Inbetriebnahme - IBN..... | 29 |
| 3.2 | Virtuelle Inbetriebnahme - VIBN | 30 |
| 3.2.1 | Definition nach VDI 4499 Blatt 2..... | 30 |
| 3.2.2 | Definition VDI/VDE 3693 Blatt 1 | 31 |

| | | |
|-------|--|----|
| 3.2.3 | Definition nach Wunsch | 32 |
| 3.2.4 | Weitere Definitionen der Virtuellen Inbetriebnahme | 32 |
| 3.3 | Motivation zur Durchführung der VIBN | 33 |
| 3.4 | Steuerungsebenen zur Durchführung einer VIBN | 35 |
| 3.5 | Begriffsbestimmungen betreffend zeitlicher Abläufe in der Simulation | 39 |
| 3.5.1 | Echtzeit | 39 |
| 3.5.2 | Zeitachse zwischen Real- und Simulationssystem | 41 |
| 3.5.3 | Abtasttheorem | 42 |
| 3.6 | Methoden der Virtuellen Inbetriebnahme | 43 |
| 3.6.1 | Systemsimulation / Model in the Loop (MiL) | 44 |
| 3.6.2 | Software in the Loop (SiL) | 45 |
| 3.6.3 | Hardware in the Loop (HiL) | 47 |
| 3.7 | Modelle in der VIBN | 48 |
| 3.8 | Zusammenfassung | 49 |
| 4 | Iststandsanalyse bei Magna Steyr | 51 |
| 4.1 | Motivation und Zielsetzung der Virtuellen Inbetriebnahme | 51 |
| 4.2 | Anlagentechnik | 53 |
| 4.3 | Werkzeuge zur Virtuellen Absicherung | 57 |
| 4.4 | Produktentstehungsprozess | 58 |
| 5 | Anwendungspositionierung der Virtuellen Inbetriebnahme | 61 |
| 6 | Infrastruktur für die Virtuelle Inbetriebnahme | 63 |
| 6.1 | Übersicht HiL-Testkit | 63 |
| 6.2 | Simit Unit PN256 und Simba PNIO premium | 65 |
| 6.3 | Netzwerktopologie und Netzwerkvarianten | 67 |
| 6.3.1 | Netzwerktopologie A | 67 |
| 6.3.2 | Netzwerktopologie B | 68 |
| 6.3.3 | Netzwerktopologie C | 69 |
| 7 | Aufbau und Modellierung des Prototyps zur Virtuellen Inbetriebnahme | 70 |
| 7.1 | Setup des VIBN-Systems | 70 |
| 7.1.1 | Datenbereitstellung aus realem SPS-System | 71 |
| 7.1.2 | Projektierung Simit Unit PN256 | 72 |
| 7.1.3 | Systemvorbereitung Process Simulate 14.1 | 74 |
| 7.2 | Teamzusammensetzung und Rollenverteilung | 77 |
| 7.3 | Grundlagen der Signalaufbereitung für Process Simulate | 79 |
| 7.4 | Datenbereitstellung aus realem SPS-System | 80 |
| 7.5 | Modellaufbereitung Roboter | 82 |
| 7.5.1 | Festlegung Systemgrenze | 82 |
| 7.5.2 | Signalkorrespondenz zwischen Roboter und SPS | 84 |
| 7.5.3 | Signalimport | 86 |
| 7.5.4 | Roboterprogramme | 88 |
| 7.6 | Modellaufbereitung Anlagentechnik | 90 |
| 7.6.1 | Logikblock | 91 |

| | | |
|-------|--|-----|
| 7.6.2 | Signalimport | 93 |
| 7.6.3 | Herausforderung im Zuge der Modellaufbereitung | 95 |
| 7.7 | Herstellung Signalkorrespondenz | 96 |
| 7.8 | Materialfluss | 98 |
| 7.9 | Zusammenfassung | 99 |
| 8 | Durchführung der Virtuellen Inbetriebnahme | 101 |
| 8.1 | EA-Check | 101 |
| 8.2 | Handbetrieb | 102 |
| 8.3 | Automatikbetrieb | 103 |
| 8.4 | Zusammenfassung | 106 |
| 9 | Zusammenfassung, Fazit und Ausblick | 107 |
| 9.1 | Fazit | 107 |
| 9.2 | Ausblick | 108 |
| | Literaturverzeichnis | 109 |
| | Abbildungsverzeichnis | 115 |
| | Tabellenverzeichnis | 120 |
| | Abkürzungsverzeichnis | 121 |

1 EINLEITUNG

Aufgrund der Wachstumsstrategie von Magna Steyr und des damit stetig steigenden Automatisierungsgrades in dessen Karosseriebauten stoßen traditionelle Inbetriebnahmemethoden immer öfters an ihre Grenzen. Beharrlich kürzer werdende Zeiträume für die Inbetriebnahme dieser hochautomatisierten Fertigungsanlagen lassen es kaum mehr zu, alle erforderlichen Programmierschritte an der realen Fertigungsanlage durchzuführen. Auch die dazugehörigen Verifikations- wie auch Validierungsmaßnahmen zur Sicherstellung der geforderten Projektziele im Bereich der Steuerungstechnik sind mit der klassischen Vorgehensweise kaum mehr umsetzbar. Des Weiteren ist es für den Betreiber der Produktionseinrichtungen aufgrund anhaltend steiler werdender Anlaufkurven von Produktionsprogrammen von Bedeutung, frühestmöglich die Funktionstauglichkeit sowie Ausschreibungskonformität der beauftragten Produktionseinrichtungen sicherzustellen wie auch gegenzuprüfen. Die bei Magna Steyr betriebenen Karosseriebauten charakterisieren sich durch eine hohe Einsatzdichte von Industrierobotern. Eine durchschnittliche Fertigungszelle besteht aus vier bis zehn Robotern, deren Abläufe und Produktionssequenzen wiederum mittels einer Speicherprogrammierbaren Steuerung (SPS) synchronisiert und koordiniert werden. Ebenso werden über diese Industriesteuerung ergänzende Peripherieeinrichtungen, wie zum Beispiel pneumatische Spannvorrichtungen, Asynchronmaschinen und Sicherheitseinrichtungen gesteuert und überwacht. Für die darin eingesetzten Steuerungen und Steuerungskonzepte gilt es ein virtuelles Absicherungsmodell respektive eine Validierungsmethode zu entwickeln.

Zielsetzung der Arbeit ist die Entwicklung einer Methode beziehungsweise eines Prototyp, der die Anbindung eines virtuellen Anlagenmodells an reale SPS-Systeme ermöglicht. Im Zuge dessen ist der Fokus des Forschungsinteresses dahingehend zu konzentrieren, dass weitestgehend Originalsteuerungen und nativer Programmcode verwendet werden können. Auch die Verfolgung und das Monitoring des Entwicklungsfortschritts im Zuge der Softwareerstellung durch Dritte sind durch das zu entwickelnde Absicherungsmodell zu ermöglichen. Ebenso ist dadurch eine vom Produktionsgeschehen entkoppelte Möglichkeit zu entwickeln, komplexe Störungsszenarien und Fertigungsausfälle analysieren zu können.

Das virtuelle Anlagenmodell muss die Fähigkeit besitzen, geometrische, prozesstechnische sowie programmtechnische Abläufe abbilden und bewerten zu können. Ebenfalls ist die bereits vorhandene Systemlandschaft bei Magna Steyr zu berücksichtigen, wie auch die bestmögliche Positionierung dieses zusätzlichen Engineeringschritts im bestehenden Produktentwicklungsprozess. Die wirtschaftliche Bewertung der zu entwickelnden Alternative zur vorzeitigen Verifizierung, Validierung und Inbetriebnahme von Steuerungs- und Fertigungskonzepten ist nicht Inhalt dieser Arbeit. Das Verhältnis zwischen zusätzlich anfallendem Aufwand durch Anwendung der neuen Entwicklungs- wie auch Absicherungsmethode zum erzielten Nutzen findet jedoch Berücksichtigung im Zuge der wissenschaftlichen Auseinandersetzung.

Abb. 1 stellt die angewandte Vorgehensweise zur Beantwortung der forschungsleitenden Fragestellungen in grafischer Form dar. Daraus geht hervor, dass die Masterarbeit in zwei Hauptteile gegliedert ist. Kapitel 2 und Kapitel 3 bilden den Theorieteil dieser Arbeit. In diesen beiden Kapiteln werden die theoretischen Grundlagen erarbeitet und erörtert, die zur Entwicklung und Bearbeitung des anschließenden Praxisblocks nötig sind. Zum Praxisteil sind die Kapitel 4 bis 8 zugeordnet, mit anschließender Zusammenfassung in Kapitel 9. In diesem werden die gewonnenen Erkenntnisse und Ergebnisse abschließend zusammengefasst.

| | | |
|-------------|-----------|--|
| | Kapitel 1 | EINLEITUNG |
| Theorieteil | Kapitel 2 | ALLGEMEINE MODELLBILDUNG |
| | Kapitel 3 | VIRTUELLE INBETRIEBNAHME |
| | Kapitel 4 | ISTSTANDSANALYSE BEI MAGNA STEYR |
| Praxisteil | Kapitel 5 | ANWENDUNGSPositionIERUNG DER VIRTUELLEN INBETRIEBNAHME |
| | Kapitel 6 | INFRASTRUKTUR FÜR DIE VIRTUELLE INBETRIEBNAHME |
| | Kapitel 7 | AUFBAU UND MODELLIERUNG DES PROTOTYPEN ZUR VIRTUELLEN INBETRIEBNAHME |
| | Kapitel 8 | DURCHFÜHRUNG DER VIRTUELLEN INBETRIEBNAHME |
| | Kapitel 9 | ZUSAMMENFASSUNG, FAZIT UND AUSBLICK |

Abb. 1: Aufbau der Arbeit wie auch Vorgehensweise zur Beantwortung der forschungsleitenden Fragestellung, Quelle: Eigene Darstellung.

Das erste Kapitel im Theorieteil hat die *Allgemeine Modellbildung* zum Inhalt. In diesem werden die theoretischen Grundlagen in Bezug auf Modellbegriff, Modellentwicklung wie auch normativer Vorgaben analysiert. Ebenso stehen Methoden zur Modellbildung, Modellformen wie auch Einsatzzwecke von Modellen im Zentrum des wissenschaftlichen Interesses. Aufbauend auf dieses Wissen erfolgt die theoretische Auseinandersetzung mit dem Themenfeld der *Virtuellen Inbetriebnahme (VIBN)*. Es erfolgt eine Analyse der Begrifflichkeiten Inbetriebnahme, Virtuelle Inbetriebsetzung und Virtuelle Inbetriebnahme sowie die Abgrenzung der Begriffe zueinander. Ebenso ist die Untersuchung zu Grunde liegender Normen in diesem Themengebiet Teil dieses Kapitels. Die Untersuchung und Auswahl der geeigneten Methode zur Durchführung der Virtuellen Inbetriebnahme ist ebenfalls Inhalt von Kapitel 3.

Das erste Kapitel im Praxisteil der Masterthesis hat die Analyse der Istsituation bei Magna Steyr in Bezug auf Anlagentechnik, eingesetzte Werkzeuge zur virtuellen Absicherung sowie des vorliegenden Produktentstehungsprozesses zum Inhalt. Basierend auf den Erkenntnissen der Iststandsanalyse und der gewonnenen theoretischen Grundlagen erfolgt die Festlegung der Positionierung der VIBN bei Magna Steyr. Ebenfalls stellen das generierte Wissen aus Theorieteil und Iststandserhebung die Basis der zu schaffenden Infrastruktur zur Durchführung einer Virtuellen Inbetriebnahme dar, die in Kapitel 5 behandelt wird. Aufbauend auf die Ergebnisse und Erkenntnisse aus den vorangegangenen Forschungsaktivitäten erfolgt in Kapitel 7 die wissenschaftliche Auseinandersetzung hinsichtlich Aufbau und Modellierung des Prototyps zur Durchführung einer Virtuellen Inbetriebnahme. Die Eignung dieses Prototyps zur Durchführung einer Inbetriebnahme wird im letzten Kapitel des Praxisteils einer Analyse und Bewertung unterzogen. Abschluss findet diese Masterarbeit in der Zusammenfassung der gewonnenen Erkenntnisse und Feststellungen. Am Beginn steht jedoch die Erarbeitung der theoretischen Grundlagen für diese Arbeit, welche mit dem anschließenden Kapitel in Angriff genommen wird.

2 ALLGEMEINE MODELLBILDUNG

Kapitel 2 setzt sich in den nachfolgenden Unterkapiteln und Abschnitten mit der Begriffsbestimmung von Modell und Modellbildung auseinander. Inhalt ist die theoretische Auseinandersetzung hinsichtlich dieser Begriffe, sowie Methoden zur Erzeugung von Modellen und deren Einsatzgebiete.

2.1 Modell – Begriffsbestimmung und Abgrenzung

Inhalt von Unterkapitel 2.1 sind die Herleitung des Begriffs *Modell* sowie dessen Definition und Bedeutung in den unterschiedlichen wissenschaftlichen Disziplinen und Anwendungsgebieten. Die Begrifflichkeit *Modell* ist in unterschiedlichsten wissenschaftlichen Disziplinen in Verwendung, wobei die Bedeutung in Abhängigkeit der Fachdisziplin und des wissenschaftlichen Bereichs starken Variationen unterliegt. So versteht die Kunstwissenschaft unter Modell oftmals die Vorlage für ein Portrait oder einer Skulptur. Hingegen dienen Modelle in der Betriebswirtschaftslehre zur Darstellung quantitativer ökonomischer Größen sowie kundenspezifischer Verhaltensweisen oder der Abbildung des Mitbewerbs. In den Ingenieurwissenschaften wiederum werden unter Modell unter anderem Gleichungssysteme zur Berechnung dynamischer Systeme sowie die grafischen Darstellungen dergleichen verstanden.¹

2.1.1 Begriffsbestimmung nach Stachowiak

Nach *Stachowiak* entstammt der Begriff *Modell* ursprünglich der Anlehnung am französischen Substantiv *modèle*, das wiederum dem vulgärlateinischen Wort *modulus* entstammt. *Modulus* umfasst die Bedeutungen *Maß* und *Maßstab* und ist seinerseits die Diminutivform von *modus*, welches wiederum die Bedeutung von *Maß* und *Maßstab* umfasst. Im weiteren Sinn deckt *modus* auch Begrifflichkeiten wie Vorschrift, Form, sowie Art und Weise ab. Die Bedeutung des Ausdrucks *Modell* im deutschen Sprachraum umfasst den selben Inhalt wie sein Übersetzungsäquivalent *modèle*, wobei sich die Gültigkeit auf zwei Bedeutungen erstreckt: Erstere steht für *etwas Vorbild* und *von etwas Abbild* zu sein und zweitere für *Repräsentanz für ein Original*.²

Ebenso lässt sich der Begriff *Modell* nach *Stachowiak* in drei Hauptmerkmale unterteilen, die wie folgt beschrieben werden:³

1. Abbildungsmerkmal

Ein Modell ist immer ein Modell von etwas – etwa Repräsentation oder Abbildung künstlicher wie auch natürlicher Objekte, welche wiederum selbst Modelle verkörpern können.

2. Verkürzungsmerkmal

Im Allgemeinen er- und umfassen Modelle nicht alle Attribute des durch sie abgebildeten Originals. Es werden nur jene Attribute verwendet, welche für den Modellnutzer respektive Modellerschaffer von Relevanz sind.

¹ Vgl. Bandow (2010), S. 5 f.

² Vgl. Stachowiak (1973), S. 129.

³ Vgl. Stachowiak (1973), S. 131 ff.

3. Pragmatisches Merkmal

Original und Modell sind nicht automatisch eindeutig zuweisbar.

In ihrer Ersetzungsfunktion erfüllen Modelle bestimmte handelnde, erkennende und modellbenutzende Subjekte unter Berücksichtigung eingeordneter Zeitintervalle unter Beschränkung angegebener tatsächlicher oder gedanklicher Operationen.

Ergänzend hebt Abb. 2 den Zusammenhang der angeführten Hauptmerkmale hervor.

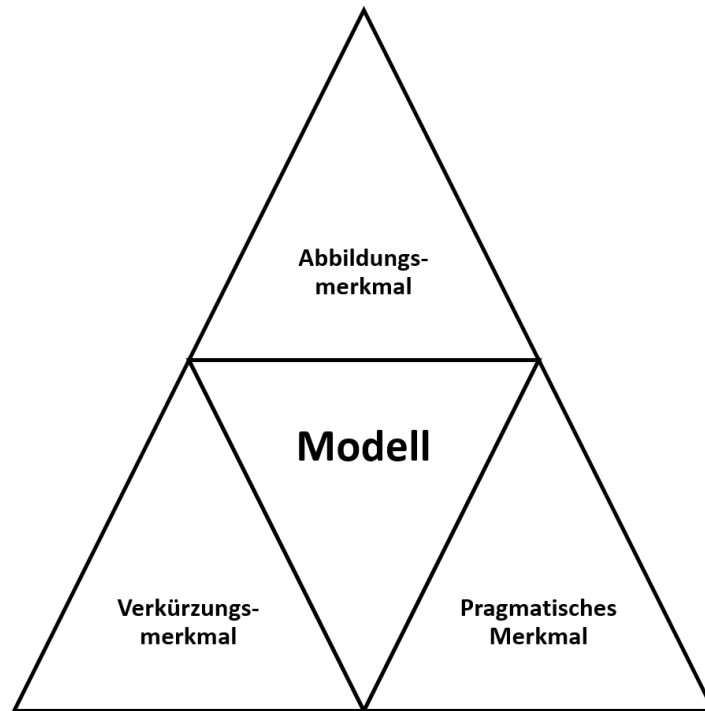


Abb. 2: Hauptmerkmale des allgemeinen Modells, Quelle: Bandow (2010), S. 9 (leicht modifiziert).

2.1.2 Begriffsbestimmung nach Bossel

Bossel beschreibt den Begriff *Modell* als einen interessierenden Realitätsausschnitt in vereinfachter Abbildung beziehungsweise reduzierter Darstellung. Eine gültige Aussage über das Originalobjekt beschränkt sich auf jenen Ausschnitt des Interesses, das zur Modellbildung dient und wird ausschließlich für diesen Ausschnitt vermittelt. Soll ein Modell die Fähigkeit besitzen Verhalten abzubilden, so muss es selbst in der Lage sein dynamisches Verhalten zu erzeugen – demzufolge muss jedes Modell prinzipiell über die gleichen Elemente verfügen wie das Originalobjekt, dessen Repräsentanz es darstellt.

Ergänzend muss es über eine Wirkungsstruktur verfügen, die es dem Modell erlaubt, auf Systemparameter und dessen Einwirkungen zu reagieren. Aufgrund des begrenzten Verhaltensausschnittes des Originals, welches durch die entsprechende Modellformulierung und den Modellzweck beschrieben wird, ist das Modell nicht das Originalsystem. Es besteht daher die Gefahr, Systemverhalten von gut funktionierenden Modellen als Systemverhalten des Originals zu interpretieren. Daher ist immer von Modellergebnissen im Zuge der Ergebnisevaluierung zu sprechen und nicht von Systemverhalten, um stets die Unterscheidung zwischen Modell und Original zu wahren.⁴

⁴ Vgl. Bossel (1992), S. 27 f.

2.1.3 Begriffsbestimmung nach Kastens und Kleine-Büning

Wie bereits *Stachowiak* erläutern *Kastens und Kleine-Büning* ebenfalls die Abstammung des Begriffs *Modell* vom lateinischen Wort *modulus* sowie dessen Bedeutung für *Maß* und *Maßstab*. Auch sie unterstreichen die unterschiedlichen Bedeutungen von *Modell* in Abhängigkeit des Einsatzgebietes und des Einsatzzweckes. So kann ein Modell zum Abbilden eines bestehenden Originals aber auch als Vorbild eines herzustellenden Originals dienen. Ebenso können das Modell sowie das Modellerte selbst abstrakt wie auch konkret aufgebaut sein. Originalgetreueheit wird bei Modellen vorsätzlich vermieden - sie verfolgen den Zweck besondere Eigenschaften wegzulassen oder hervorzuheben. Der Verwendungszweck des Modells intendiert jene Eigenschaften, die modelliert werden, und welche Herangehensweise besonders zur Beschreibung oder Nachbildung des Originals geeignet sind. Ebenfalls muss zwischen Original und Modell ein enger Bezug vorhanden sein, um den angestrebten Modellzweck erfüllen zu können, wie es Abb. 3 verdeutlicht.⁵

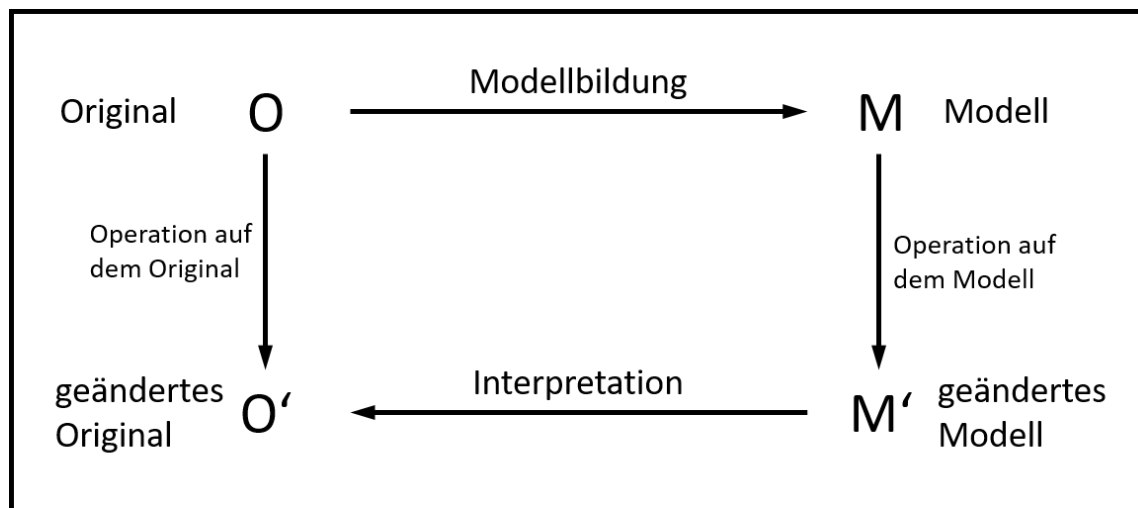


Abb. 3: Erforderlicher Bezug zwischen Modell und Original, Quelle: Kastens/Kleine-Büning (2005), S. 20 (modifiziert).

2.1.4 Begriffsbestimmung nach VDI 3633

Der Verein Deutscher Ingenieure (VDI) wiederum definiert ein *Modell* als vereinfachtes Abbild real vorkommender oder geplanter Originalsysteme in einem System anderer Begrifflich- respektive Gegenständlichkeit. Die Unterscheidung betreffend untersuchungsrelevanter Eigenschaften bezieht sich ausschließlich auf einen Toleranzrahmen, welcher vom Untersuchungsziel des Vorbilds abhängt.⁶

2.1.5 Abgrenzung

Trotz der Anwendbarkeit und Gültigkeit des Begriffs *Modell* in den unterschiedlichsten wissenschaftlichen Disziplinen wird aufgrund der ingenieurwissenschaftlichen Ausrichtung dieser Arbeit in weiterer Folge ausschließlich auf die Modellbildung in der Naturwissenschaft eingegangen.

⁵ Vgl. Kastens/Kleine-Büning (2005), S. 16-20.

⁶ Vgl. VDI 3633 Blatt 1 (Dezember 1993), S. 3.

2.2 Modellbildung

Damit der Begriff der *Modellbildung* sowie die dazugehörigen Prozesse behandelt werden können, muss zuvor der Begriff des *Systems* erörtert werden, da dieses stets den Ausgangspunkt zur Modellbildung darstellt.

Nach Definition des IEC (International Electrotechnical Commission) ist ein *System* ein Satz von miteinander in Beziehung stehenden Elementen, welche in einem definierten Kontext als Ganzes betrachtet und von ihrer Umgebung getrennt gesehen werden. Ein System wird im Allgemeinen mit dem Ziel definiert, eine bestimmte Zielsetzung zu erreichen, wie zum Beispiel die Ausführung eines bestimmten Prozesses oder einer Handlung. Des Weiteren können Elemente eines Systems natürliche oder vom Menschen geschaffene materielle Objekte sowie Denkweisen und deren Ereignisse sein, wie etwa Programmiersprachen, Organisationsformen oder mathematische Verfahren. Ebenso wird davon ausgegangen, dass ein System von der Umgebung und den anderen externen Systemen durch eine imaginäre Abzäunung abgegrenzt ist. Kann aus dem bezugnehmenden Kontext ein System nicht qualifiziert werden, so muss durch ergänzende Erläuterungen klar beschrieben sein, auf was sich die Bezeichnung *System* bezieht.⁷

Bossel definiert ein *System* als eine Struktur, die aus einzelnen oder mehreren Elementen besteht, wobei der Zustand dieser Elemente wiederum von anderen Elementen abhängt oder beeinflusst wird. Ein System wird von seiner Systemumwelt durch seine Systemgrenze getrennt. Eindeutig ist diese Grenze nur in Fällen, wo kein Transfer von Information, Energie oder Stoffen stattfindet. In der Praxis muss meist eine Systemgrenze durch eine Fläche um das zu beschreibende System definiert werden, damit dieses nicht durch externe Störgrößen beeinflusst wird. Diese Fläche stellt ebenfalls die Abgrenzung zu Komponenten dar, mit denen nur eine schwache Interaktion stattfindet, wie es Abb. 4 verdeutlicht.⁸

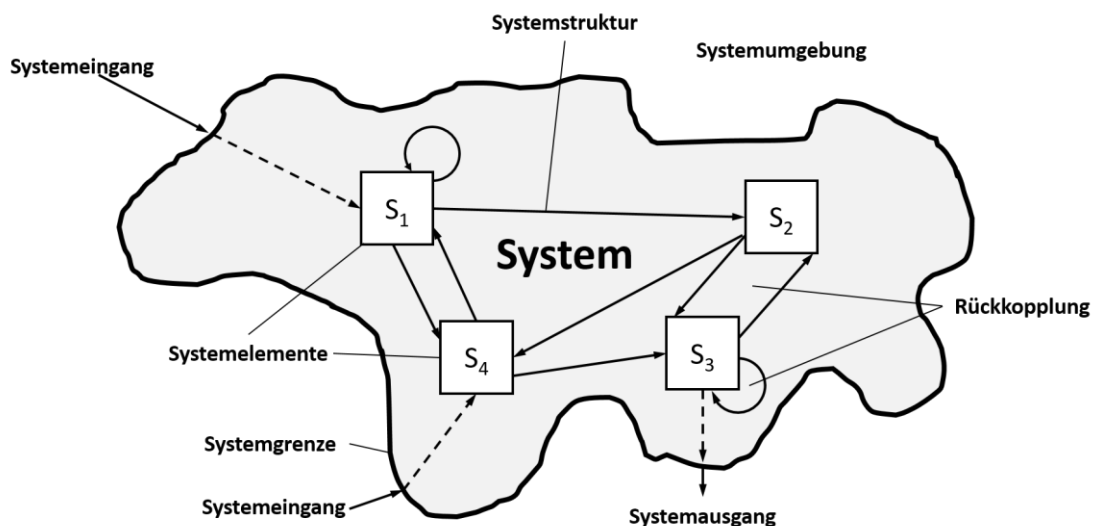


Abb. 4: Grundkonzept eines Systems, Quelle: Bossel (1989), S. 11 (leicht modifiziert).

Nachdem die Begrifflichkeit *System* einer eingehenden Analyse unterzogen wurde, kann zur wissenschaftlichen Auseinandersetzung betreffend die Modellbildung übergegangen werden.

⁷ Vgl. IEC 60050-151 (2001-07), S. 10.

⁸ Vgl. Bossel (1989), S. 10 f.

Wie im Abschnitt 2.1.5 angeführt wird ab dieser Stelle die Eingrenzung der theoretischen Auseinandersetzung hinsichtlich der Erfordernisse und Anforderungen der Ingenieursdisziplinen vollzogen. Daher bezieht sich die nachfolgende Erörterung des Begriffs *Modellbildung* ausschließlich auf diese.

Wie *Kastens und Kleine-Büning* in ihrem Grundlagenwerk anführen, ist der Bezug zwischen Modell und Original Voraussetzung, um den angestrebten Modellzweck erfüllen zu können.⁹ Diese Bezugsherstellung von Original zu Modell wird *Modellbildung* genannt, wie es auch Abb. 3 grafisch hervorhebt.

Bungratz u. a. definieren den Prozess der Modellbildung als Analyse und formale Herleitung eines Modells zur Beschreibung eines technischen Systems, eines Phänomens oder Effekts. In der Regel ist der Ausgangspunkt für die Modellbildung des Modellierungsgegenstands eine nichtformale Beschreibung desselben. Die Überleitung in ein semiformales Beschreibungsmodell erfolgt mittels Werkzeug der Anwendungsdisziplin, aus dem in weiterer Folge ein konsistentes, widerspruchsfreies – sprich streng formales Modell gebildet werden kann.¹⁰

Bossel stellt in seiner Publikation den Modellzweck in den Mittelpunkt der Modellbildung wie es Abb. 5 verdeutlicht.

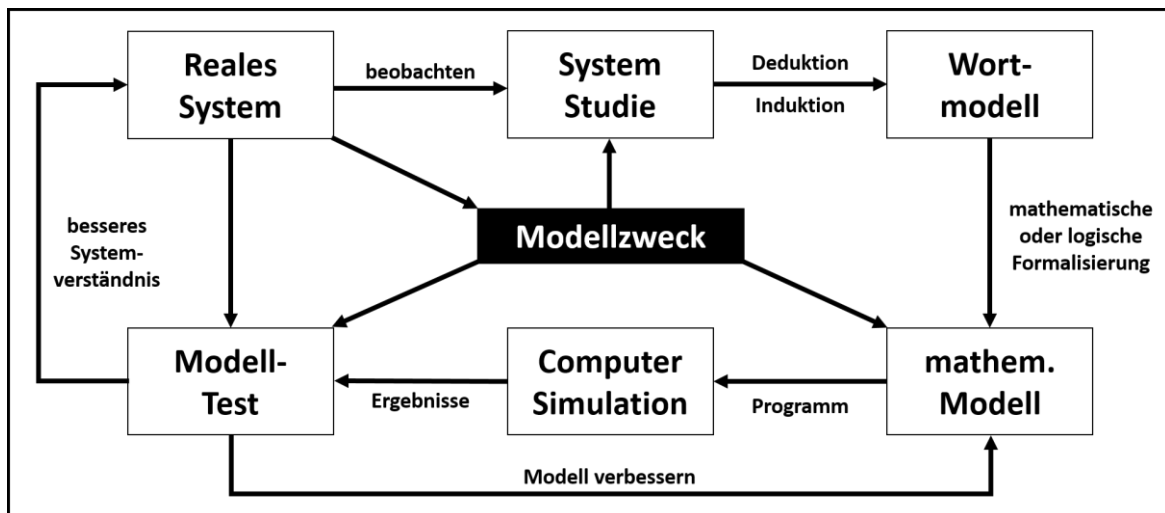


Abb. 5: Iterativer Prozess der Modellbildung und Modellentwicklung, Quelle: Bossel (1989), S. 13 (leicht modifiziert).

Ein Modell wird durch seinen Modellzweck bestimmt, wobei ein Bewirtschaftungs- oder Verständnisproblem Anlass der Modellbildung ist. Detaillierungsgrad und Umfang des Modells wird wiederum vom Modellzweck ausgehend bestimmt.¹¹

Modellzweck und Detaillierungsgrad werden auch in normativen Regelwerken als maßgebliche Einflussgrößen zur Modellbildung angeführt, welche im nachfolgenden Unterkapitel ausführlich erörtert werden.

⁹ Vgl. Kastens/Kleine-Büning (2005), S. 20.

¹⁰ Vgl. Bungratz u. a. (2009), S. 3 ff.

¹¹ Vgl. Bossel (1989), S. 12.

2.3 Normative Vorgaben in der Modellbildung

Kern von Unterkapitel 2.3 stellt die Auseinandersetzung mit den normativen Vorgaben hinsichtlich Modellbildung dar, welche Anwendung in den Ingenieurwissenschaften finden, wobei als erstes das VDI-Regelwerk 4465 einer eingehenden Betrachtung unterzogen wird.

2.3.1 VDI 4465 Blatt 1 - Modellierung und Simulation / Modellbildungsprozess

Vor Erörterung des Inhalts dieser VDI-Richtlinie (Verein Deutscher Ingenieure) sei angemerkt, dass bis zum Verfassen der Reinschrift dieser Masterthesis die VDI 4465, Blatt 1 nur als Entwurf zur Verfügung stand, obwohl die Einspruchsfrist derselben bereits mit 31.10.2016 auslief.

Nichtsdestotrotz soll der Inhalt dieses Blattes für die Masterthesis nicht unterschlagen werden, da der Schwerpunkt dieser Norm der Modellbildungsprozess von Produktions-, Materialfluss- sowie Logistiksystemen darstellt. Produktion und Materialfluss stellen wesentliche Eingangsparameter wie Einflussfaktoren für die VIBN (Virtuelle Inbetriebnahme) dar.

Nach VDI 4465 wird unter Modellbildung die „Abbildung von Systemen in Modellen, abhängig von der Zielsetzung und der subjektiven Sichtweise des Modellbildners“¹² verstanden.

Aufgabe der Modellbildung ist es, eine Vorgehensweise herzuleiten, welche über geeignete Beschreibungsmittel verfügt, um eine Aufgabenstellung oder Zielsetzung mittels geeignetem Modell darzustellen. Auf Basis dieses Modells wird es nachfolgend möglich, Leistungskennwerte simulativ oder analytisch zu ermitteln. Ebenso wird die Erfordernis mehrerer Iterationsschleifen im Zuge der Modellbildung betont. Grund dafür ist kontinuierlicher Wissenszuwachs hinsichtlich Aufgaben- respektive Problembeschreibung im Zuge der Projektarbeit sowie neue Erkenntnisse aus dem Systemverhalten durch Anwendung des Modells wie auch Abb. 6 unterstreicht.¹³

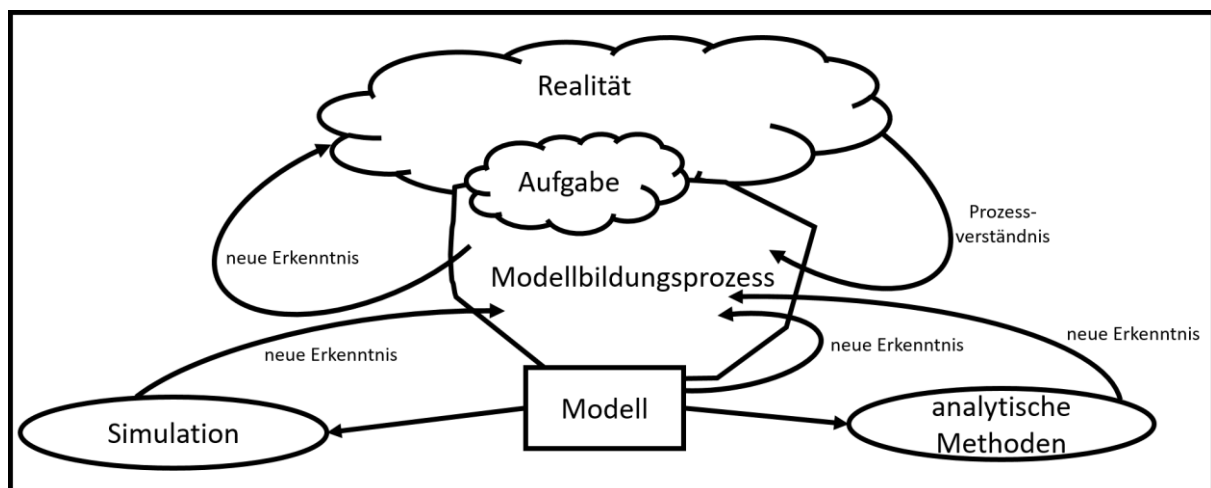


Abb. 6: Modellbildungsprozess basierend auf Erkenntnis- u. Wissenszuwachs, Quelle: VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 2 (leicht modifiziert).

Eine weitere Norm aus dem Gremium des Vereins Deutscher Ingenieure ist die Richtlinie VDI 2206, welche die Entwicklung und Modellbildung mechatronischer Systeme zum Schwerpunkt hat.

¹² VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 3.

¹³ Vgl. VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 1 f.

2.3.2 VDI 2206 - Entwicklungsmethodik für mechatronische Systeme

Zielsetzung der Richtlinie ist, den Entwicklungsprozess von mechatronischen Systemen domänenübergreifend zu unterstützen, wobei der Modellbildung in diesem Zusammenhang eine besondere Bedeutung zukommt.¹⁴

Grundlage des modellbasierten Systementwurfs ist der Makrozyklus des V-Modells, welches mechatronische Systeme im Zuge der Entwurfsphase generisch beschreibt sowie die erforderlichen Ausprägungen hinsichtlich Informationstechnik, Elektrotechnik sowie Maschinenbau ermöglicht. Der Makrozyklus kann bezüglich Modellerstellung in zwei Hauptphasen unterteilt werden, wobei der Systementwurf als erste Phase das Ziel eines festgelegten domänenübergreifenden Lösungskonzeptes verfolgt. Die zweite Phase umfasst alle Schritte der Modellbildung, welche den gesamten Systementwurf flankieren. Zweck der Modellbildung ist es, durch Unterstützung mittels Modellen die Systemeigenschaften des mechatronischen Produkts durch Simulation abzubilden, wie es Abb. 7 ergänzend hervorhebt.¹⁵

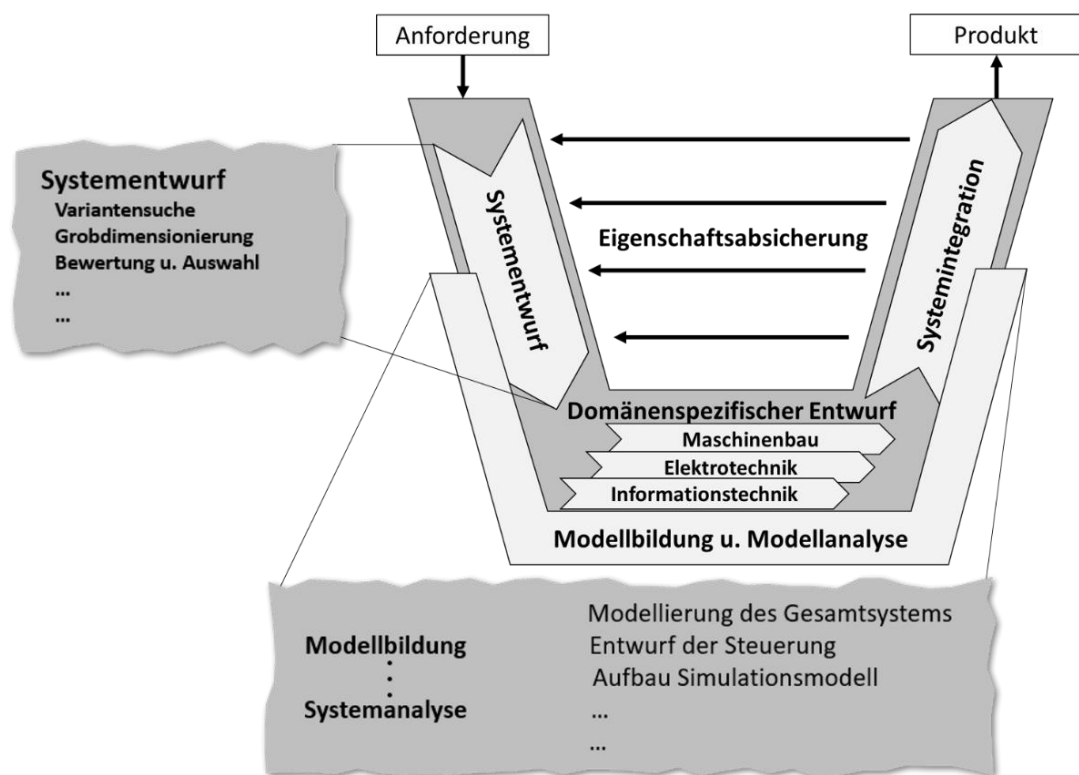


Abb. 7: Makroprozess des V-Modells zu domänenübergreifenden Entwicklung von mechatronischen Komponenten, Quelle: VDI 2206 (Juni 2004), S. 81 (leicht modifiziert).

Auch diese VDI Richtlinie unterstreicht die Notwendigkeit einer iterativen Herangehensweise für den gesamten Entwicklungsprozess. Ebenso muss während der Modellbildungsphase immer die Güte des Modells in Abhängigkeit der Problemstellung berücksichtigt werden. Hinsichtlich Entwicklungsaufgabe muss Klarheit herrschen, da für Weiterentwicklungen bestehender Komponenten andere Modellierungsansätze notwendig sind als für Neuentwicklungen. Dieses Erfordernis führt auch die letzte Richtlinie an, die im Zuge dieser Arbeit erörtert wird.

¹⁴ Vgl. VDI 2206 (Juni 2004), S. 8.

¹⁵ Vgl. VDI 2206 (Juni 2004), S. 29 f.

2.3.3 VDI 3843 Blatt 1 - Modellbildung schwingungsfähiger Systeme

Die Richtlinie VDI 3843 Blatt 1 beschreibt die Modellbildung von schwingungsfähigen Systemen. Die Modellbildung wird hierbei als wesentlicher Schritt zur Beantwortung von Fragestellungen technischer Natur betrachtet, welche es im Zuge der Untersuchung und Entwicklung technischer Systeme zu klären gilt. Ausgang für diesen Prozess sind geplante oder bereits existierende Objekte wie zum Beispiel eine technische Anlage, eine Maschine oder Baugruppe. Ergebnis der Modellbildung ist der Aufbau eines Berechnungsmodells, das ein der Aufgabenstellung nützliches Ergebnis liefert. Das aus der Modellbildung hervorgegangene Modell stellt die Grundlage für die Simulation des zu untersuchenden technischen Systems dar. Der Übergang vom realen Objekt zum Berechnungsmodell wird in diesem VDI-Blatt als Modellbildung definiert, wobei auch in diesem Regelwerk die iterative Herangehensweise im Modellentstehungsvorgang betont wird. Ebenso wird unterstrichen, dass keine eindeutigen Vorschriften betreffend der Modellbildung angewandt werden können, da das Modell stets vom Untersuchungszweck abhängt. Daher wird es in unterschiedlichen Formen mit unterschiedlichen Freiheitsgraden vom Realobjekt abgeleitet.¹⁶

Eine in der Praxis bewährte Empfehlung hinsichtlich der Herangehensweise an den Prozess der Modellerstellung wird in Abb. 8 dargestellt.

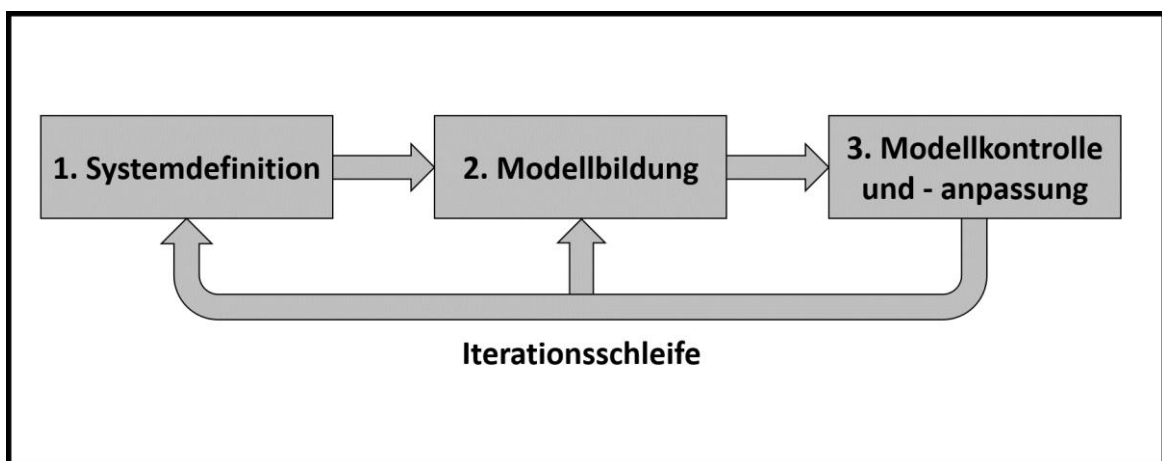


Abb. 8: Hauptblöcke des Modellentstehungsprozesses, Quelle: VDI 3843 Blatt 1 (Februar 2018), S. 6 (modifiziert).

Wie aus Abb. 8 ersichtlich, besteht der Prozess der Modellerstellung aus drei Blöcken. Block 1 und Block 2 beinhalten Regeln für die Erstellung von Erstmodellen sowie Handlungsempfehlungen für die Weiterbearbeitung von bereits bestehenden Modellen. Block 3 legt seinen Schwerpunkt auf die Kontrolle des erstellten Modells sowie dessen Aussagekraft und Güte.¹⁷ Dieser Prozessschritt wird auch im bereits behandelten VDI-Regelwerk Nummer 4465 in Form von Erkenntniszuwachs und Prozessverständnis akzentuiert, wie Abb. 6 ersichtlich macht.

Nach dem in diesem sowie im vorangegangenen Unterkapitel allgemeine Grundsätze zur Modellbildung behandelt wurden, findet an diesem Punkt die Überleitung zum nächsten Themenblock statt. Das anschließende Unterkapitel beschäftigt sich mit den methodischen Herangehensweisen zur Modellbildung.

¹⁶ Vgl. VDI 3843 Blatt 1 (Februar 2018), S. 2 ff.

¹⁷ Vgl. VDI 3843 Blatt 1 (Februar 2018), S. 30.

2.4 Methoden der Modellbildung

Die folgenden beiden Abschnitte beschäftigen sich mit den methodischen Herangehensweisen respektive Zugängen zur Bildung von Modellen. Welche Methode zur Modellbildung Anwendung findet, hängt vom Kenntnisgrad über jenes Realsystem ab, dessen Eigenschaften im Modell beschrieben und anschließend mittels rechnergestützter Simulation untersucht werden soll. Für die Abbildung mechatronischer Systeme ist das Wissen hinsichtlich der Beziehungen und Abhängigkeiten zwischen Physik, Mathematik, Elektrik, Elektronik und verwandter Fachgebiete Voraussetzung.¹⁸

2.4.1 Analytische Modellbildung

Sind Basiseigenschaften sowie entscheidende Charaktere des Realsystems bekannt, so ist die analytische Modellbildung als methodischer Zugang anzuwenden. Mittels Gleichungen werden mathematische Zusammenhänge zwischen Ausgangsgrößen und Eingangsgrößen formuliert. Meist sind hier Differenzialgleichungssysteme notwendig, die auch strukturfestlegend bezüglich des Modells sind. Besteht ein Gesamtsystem aus mehreren Teilsystemen, müssen geeignete Schnittstellen festgelegt werden. Basierend darauf erfolgt anschließend die Parametrierung des Modells. Quellen hierfür sind Daten aus der mechanischen und elektrischen Konstruktion, aus Datenblättern sowie aus Diagrammen. Ebenso kann bei Verfügbarkeit des Realsystems entscheidendes Wissen für die Modellbildung durch Vermessung generiert werden. Nicht bekannte Parameter müssen angenommen oder geschätzt werden. Dies führt zu einer Modellunsicherheit, welche sich in abweichendem Verhalten zwischen Originalsystem und simuliertem Modell widerspiegelt. Dies kann jedoch durch Erfahrungswissen sowie Parameterübernahmen aus ähnlichen Modellen kompensiert werden. Abschließend ist es erforderlich, für die Simulation Anfangswerte festzulegen. Diese hängen, wie auch Eingangs – und Ausgangsgrößen, von der Aufgabenstellung und vom Untersuchungszweck ab.¹⁹

Ein weiterer Aspekt, der für den analytischen Zugang in der Modellbildung spricht, ist die Nachvollziehbarkeit und Transparenz bei Entwurfsentscheidungen im Zuge des Entstehungsprozesses des Simulationsmodells. Mittels dieser Methode lassen sich Zusammenhänge von Verhaltenseigenschaften und physikalischen Parametern explizit darstellen. Auch Randbedingungen, welche für das Systemverhalten des Modells relevant sind, lassen sich systematisch in den Systementwurf einarbeiten. Ebenfalls zeigt sich der Nutzen dieser Herangehensweise bei der Verifikation des Modells, da hier die Nachvollziehbarkeit der angewandten mathematischen und physikalischen Eigenschaften weitestgehend gegeben ist.²⁰

Als letzten Aspekt der analytischen Modellbildung beziehungsweise eines analytischen Modells ist der geringere Aufwand bei der Bildung von Modellen gegenüber anderen Methoden, wie zum Beispiel Prototypmodellen, Pilotmodellen oder Ähnlichkeitsmodellen anzuführen.²¹

¹⁸ Vgl. Glöckler (2018), S. 117.

¹⁹ Vgl. Glöckler (2018), S. 117 f.

²⁰ Vgl. Janschek (2010), S. 13 ff.

²¹ Vgl. Roddeck (2017), S. 11.

2.4.2 Experimentelle Modellbildung

Sind grundlegende und charakterbildende Eigenschaften eines Systems nicht bekannt, so muss die Methode der experimentellen Modellbildung angewandt werden. Ziel dieser Modellbildungsmethode ist es, Eigenschaften statischer und dynamischer Natur eines Teil- oder Gesamtsystems zu ermitteln, um nachfolgend geeignete Parameter zur Beschreibung des Modells festlegen zu können. Voraussetzung hierfür ist der Zugang zum Realsystem, um die erforderlichen Messungen und Ermittlungsmethoden durchführen zu können.²²

Roddeck bezeichnet Methoden zur experimentellen Modellbildung als Identifikationsverfahren. Eine einfache Form davon repräsentiert beispielsweise die Ermittlung der Übertragungsfunktion mittels Anwendung unterschiedlicher Methoden, die aus der Regelungstechnik bekannt sind. Er führt an, dass es keine allgemeingültige Vorgehensweisen und Regeln zur Herleitung von gültigen und effizienten Modellen gibt. Ein Grundsatz in der Modellbildung hat sich jedoch vielfach bestätigt, nämlich jener, die Modellbildung so einfach wie möglich zu gestalten, ohne aber unzulässige Annahmen zu treffen, die das Systemverhalten verfälschen.²³

Bohn und Unbehauen wiederum titulieren diesen Weg der Modellbildung als Systemidentifikation oder auch nur mit Identifikation, wobei diese Herangehensweise zur Bildung eines Modells auf einem real existierenden System beruht. Oftmals lässt sich die Systemidentifikation ohne besondere Kenntnis und Spezialwissen über das zu untersuchende Realsystem einfach und schnell experimentell erheben. Ergebnis daraus ist der nachfolgende Erhalt eines parametrischen oder nichtparametrischen Modells.

Als großer Nachteil mittels Zugang durch Identifikation und Systemidentifikation wird das Vorhandensein eines bereits existierenden Realsystems bewertet. Dies steht dem Vorteil einer nicht erforderlichen Spezialkenntnis bezüglich des Realsystems gegenüber. Als Konsequenz daraus folgt, dass experimentelle Methoden zum Erstellen von Modellen in Planungs- wie auch Entwurfsstadien nicht angewandt werden können. Auch die Weitergabe und Übertragbarkeit von parametrischen Modellen, die mittels Systemidentifikation erstellt wurden, ist beschränkt, da die physikalische Bedeutung einzelner Modellparameter nicht gegeben ist.²⁴

Aufgrund der Erörterung der in diesem Unterkapitel analysierten Methoden zur Modellbildung werden im anschließenden Teil dieser Arbeit die erforderlichen und unterschiedlichen Schritte im Modellbildungsprozess behandelt. Da die zu Grunde liegenden Systeme und Untersuchungsgegenstände des praktischen Teils dieser Arbeit weitestgehend bekannt sind, wird nachfolgend der Schwerpunkt auf die einzelnen Schritte der analytischen Modellbildung gelegt.

²² Vgl. Glöckler (2018), S. 119 ff.

²³ Vgl. Roddeck (2017), S. 15 f.

²⁴ Vgl. Bohn/Unbehauen (2016), S. 6.

2.5 Schritte der Modellbildung

In den Unterkapiteln 2.2 und 2.3 wurden die grundlegenden Phasen der Modellbildung beschrieben, die sowohl aus Sicht des Modellzwecks wie auch basierend auf normativen Regelwerken Relevanz besitzen. Die nachfolgenden Seiten setzen sich eingehend mit den erforderlichen Einzelschritten auseinander, welche für die Erstellung und Bildung eines anwendbaren und aussagekräftigen Simulationsmodells notwendig sind. Aufgrund der bisherigen Erkenntnisse, die im Zuge der Literaturrecherche für diese Arbeit generiert werden konnten, basieren die nachfolgenden Detaillierungen auf den Werken von Hartmut Bossel sowie dem VDI Regelwerk 4465.

2.5.1 Systemmodellierung und Systemanalyse nach Bossel

Hartmut Bossel teilt den Prozess zur Modellbildung respektive Modellentwicklung in 13 Teilschritte auf. Diese Schritte werden in den nachfolgenden Seiten entsprechend ihrer chronologischen Abfolge im Detail erörtert:²⁵

1. Am Beginn der Systemanalyse und Modellentwicklung respektive Modellbildung steht der Schritt der *Problembeschreibung*. In diesem wird das Problem erkannt sowie genau beschrieben. Ebenso muss sichergestellt werden, dass das Problemverständnis sämtliche Systemkomponenten erfasst, welche das Problem verursachen.
2. Anschließend muss der *Modellzweck* bestimmt werden.
Auf Grund unzähliger Möglichkeiten, ein System durch ein Modell beschreiben zu können, ist es in diesem Schritt existentiell, die Zusammenfassung und Auswahl relevanter Größen stets im Hinblick und Fokus des Problemzwecks zu treffen. Deshalb muss der Modellzweck am Beginn der Systemanalyse niedergeschrieben und während der Modellbildung regelmäßig ins Bewusstsein gerufen werden. Dadurch wird ein Abdriften vom ursächlichen Problem im Laufe der Modellentwicklung verhindert.
3. Darauf basierend kann das Festlegen der *Systemgrenze* erfolgen.
Nach Spezifizierung von Problem und Modellzweck erfolgt die Definition der Systemgrenzen. In diesem wird fixiert, welche Größen im Modell eingebunden und daher Bestandteil des zu analysierenden Systems sind, sowie jene Größen, die als extern klassifiziert und einzustufen sind, und daher nicht Teil des Modells werden.
4. Anschließend wird der Schritt der Erstellung des *Wortmodells* vollzogen.
Dieser Schritt stellt sicher, dass das Modellkonzept durch alle an der Entstehung beteiligten Personen dasselbe gegengeprüft werden kann. Ebenfalls wird damit sichergestellt, das Systemwissen aller am Projekt Mitwirkenden miteinbezogen zu haben.
Vollzogen wird dies mittels umgangssprachlicher Beschreibung aller am System wirkenden Funktionen sowie dessen Einzelementen. Generiert wird das notwendige Systemwissen zur Beschreibung im Wortmodell durch Plausibilitätsprüfung, Destillation und Diskussion von wesentlichen Informationen mit jenen Menschen, welche dieses Wissen zur Verfügung stellen. Insbesondere ist

²⁵ Vgl. Bossel (1989), S. 29 - 33.

im Zuge der Informationsbeschaffung darauf zu achten, dass dieser Prozess in Zusammenarbeit mit Fachexperten und Nichtexperten sowie Personen unterschiedlicher Ausbildung und Erfahrung durchgeführt wird.

5. Aufgrund des erstellten Wortmodells lassen sich nachfolgend relevante *Systemelemente* herausdestillieren sowie passende Kategorien wie zum Beispiel Anfangsbedingungen, externe oder interne Systemparameter, Zwischengrößen sowie Zustandsgrößen.
6. Auch können im Wortmodell bereits Strukturverbindungen zu den unterschiedlichen Systemelementen erkannt werden, die nachfolgend als *Strukturverknüpfungen* dargestellt werden.
7. Die Verknüpfung der Systemelemente wird als *Wirkungsdiagramm* zusammengefasst. Dies erfolgt durch Benennung mit umgangssprachlichem Namen des einzelnen Systemelements und darauf folgender Verbindung der jeweiligen Systemelemente mittels Pfeil, wobei die Pfeilrichtung die Wirkrichtung repräsentiert.

Das Wirkungsdiagramm und die sich daraus ergebende Wirkungsstruktur des Gesamtsystems ist Basis und Grundlage für das Simulationsdiagramm und das darauf aufbauende computergestützte Simulationsmodell. Im Zuge der Modellentwicklung stellt der Schritt der Entwicklung des richtigen Wirkungsdiagramms den wichtigsten Schritt im Modellentwicklungsprozess dar.

8. Im Schritt *funktionale Beziehung* werden die Pfeilverbindungen zwischen den unterschiedlichen Systemelementen um ihre jeweilige Funktion ergänzt. Konkret bedeutet dies, dass im Zuge dieses Modellierungsschrittes analysiert und entschieden werden muss, ob zwei oder mehr auf ein Systemelement wirkende Pfeile multipliziert, subtrahiert oder durch eine höherwertige mathematische Funktion beschreiben werden müssen.
9. Anschließend erfolgt die *Quantifizierung*.

Aufgrund numerischer Darstellung dynamischer Systeme im computergestützten Simulationsmodell ist es erforderlich, sämtliche internen wie externen Systemparameter, Zustandsgrößen wie auch Anfangswerte mit entsprechenden Zahlenwerten zu versehen.

10. Das *Simulationsdiagramm* kann erstellt werden, sobald alle Systemelemente, deren Funktionen und Systemstruktur definiert und zwischen den Systemelementen alle Parameter und funktionalen Beziehungen bestimmt wurden. Im Wesentlichen stellt das Simulationsdiagramm ein Wirkungsdiagramm dar, das mit den ergänzenden Informationen aus dem Schritt der Quantifizierung angereichert wurde. Dies verfügt nun über alle Daten, die für die Erstellung des Simulationsprogramms Voraussetzung sind.
11. Im Schritt *Simulation* erfolgt das Durchrechnen des Simulationsmodells. Der zeitliche Rahmen hängt hierbei von den Anfangsbedingungen sowie den Modellparametern interner und externer Systemelemente ab.

In den seltensten Fällen wird der Modellzweck mittels einmaligem Durchlauf der Simulation erfüllt. Es sind daher in der Regel mehrere Durchläufe notwendig, damit das Systemverhalten verstanden werden kann. Ohne gründliche Prüfung der Modellgültigkeit sind die daraus gewonnenen Ergebnisse daraus jedoch wertlos.

12. Im Schritt der *Gültigkeitsprüfung* wird das Simulationsmodell dem Test der Anwendungsgültigkeit, der Verhaltensgültigkeit, der Strukturgültigkeit und der empirischen Gültigkeit unterzogen. Im Zuge dieser Testläufe wird man auf eine Vielzahl von Fehlern stoßen, die an der Gültigkeit des Simulationsmodells zweifeln lassen beziehungsweise dieses ganz widerlegen. Der Schritt der Gültigkeitsprüfung muss daher ebenfalls mehrfach durchlaufen und durchgeführt werden, bis sämtliche Verbesserungen und Umformulierungen von Teilmodellen eingeflossen sind, welche die Modellgültigkeit herstellen.
13. Bei der nachfolgenden Anwendung des Simulationsmodells im operativen Einsatz ist stets darauf Bedacht zu nehmen, dass sich dessen Gültigkeit nur über den dafür vorgesehener Modellzweck sowie innerhalb des vorgesehenen Anwendungsbereichs erstreckt.

Abschließend fasst Abb. 9 die in diesem Abschnitt erörterten Teilschritte zur Modellbildung in grafischer Form zusammen.

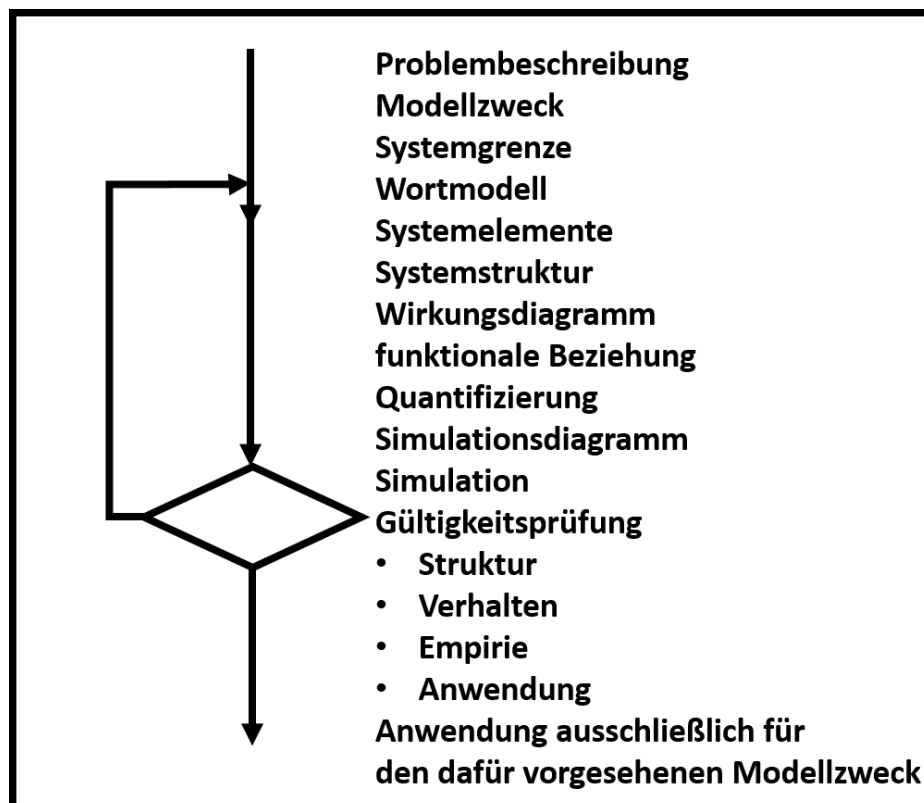


Abb. 9: Schritte zur Modellbildung und Systemanalyse, Quelle: Bossel (1989), S. 30 (leicht modifiziert).

Nachdem die Einzelschritte zur Erstellung eines Modells nach der Methodik von Hartmut Bossel ausführlich analysiert wurden, erfolgt im nachfolgenden Abschnitt die Erläuterung nach VDI 4465. Grund dafür ist, dass das VDI Regelwerk 4465 seinen inhaltlichen Schwerpunkt auf die Modellbildung von Produktionsanlagen und Materialfluss, sowie die Logistik legt.

2.5.2 Schritte der Modellbildung nach VDI 4465 Blatt 1

Dieser Abschnitt setzt sich ergänzend zu 2.5.1 mit den erforderlichen Schritten im Modellbildungsprozess nach den Sichtweisen des Blattes 1 der VDI 4465 auseinander. Ziel hierbei ist eine vervollständigende Sichtweise in Bezug auf die Vorgehensweise in der Modellbildung von Produktionsanlagen zu erlangen.

Der Modellbildungsprozess und die notwendigen Einzelschritte sind laut diesem VDI-Regelwerk in Abb. 10 grafisch dargestellt. Es handelt sich hierbei nicht um einen determinierten Prozessvorgang, dessen Schritte unumkehrbar sind und dessen Schritte ideal zueinander abgestimmt und in Folge gebracht werden können. Vielmehr handelt es sich um einen iterativen Entstehungsprozess, der durch parallele Bearbeitung unterschiedlicher Aufgaben gekennzeichnet ist.²⁶

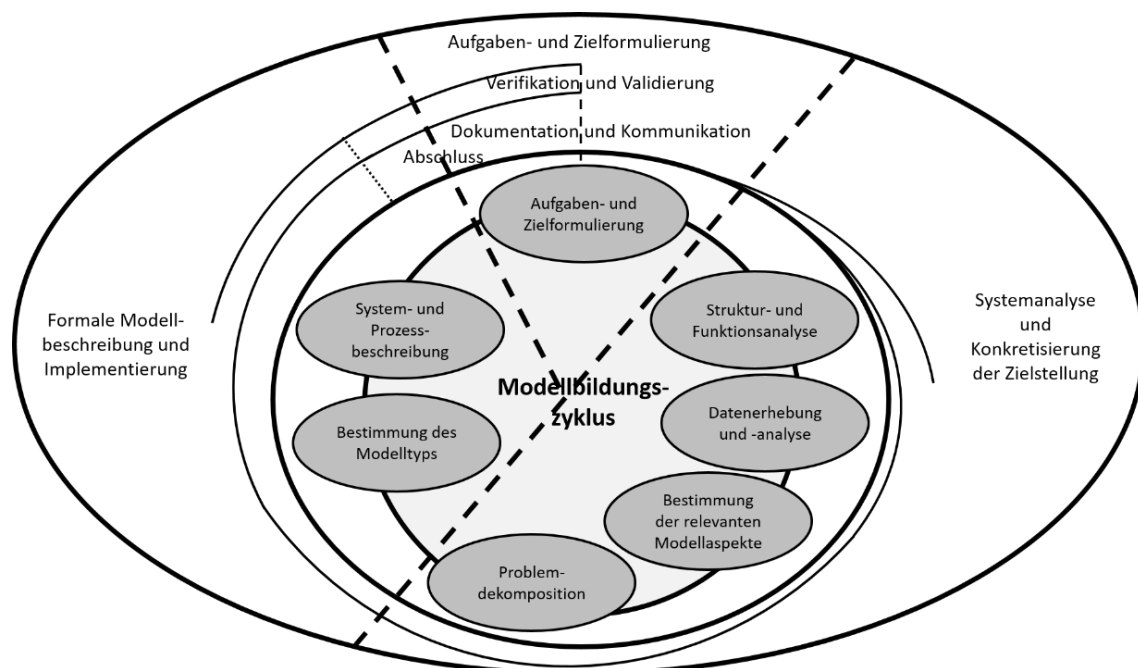


Abb. 10: Modellbildungszyklus, Quelle: VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 5 (leicht modifiziert).

Die einzelnen Schritte zur erfolgreichen Modellbildung tragen zu Bossel abweichende Überschriften respektive Titel, inhaltlich sind sie jedoch vergleichbar. Auch die Tatsache, dass eine entsprechende Teamzusammensetzung für die gelungenen Entwicklung und Erstellung eines Simulationsmodells von zentraler Bedeutung ist, wird hier betont.²⁷

Da im vorangegangenen Abschnitt ausführlich auf die Inhalte betreffend Modellbildung eingegangen wurde, wird auf eine weiterführende Auseinandersetzung mit dem Blatt 1 der VDI 4465 in dieser Arbeit verzichtet.

Nach breiter Auseinandersetzung mit den erforderlichen Schritten bezüglich des Modellbildungsprozesses in diesem Teil der Arbeit beschäftigt sich der nachfolgende mit den unterschiedlichen Grundformen von Modellen, die im Simulationsmodell zum Einsatz kommen können.

²⁶ Vgl. VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 5.

²⁷ Vgl. VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 4 - 12.

2.6 Modellformen

Die vorangegangenen beiden Abschnitte befassten sich eingehend mit den notwendigen Schritten zur Modellbildung, damit ein praxistaugliches Simulationsmodell entwickelt, erprobt und betrieben werden kann. Dieses Unterkapitel hat zum Ziel, die grundlegenden Modellformen zu analysieren, auf die im Zuge der Erstellung eines komplexen Simulationsmodells zurückgegriffen werden kann beziehungsweise muss. Die nachfolgenden Unterscheidungen beziehen sich auf den Kenntnis- und Wissensstand der Wirkstruktur und die inneren Abläufe der zur Anwendung gelangten Modellkomponenten. Der Anteil und Grad des Vorwissens über eine Systemkomponente, die zur Bildung eines Simulationsmodells herangezogen wird, kann in eine Skala von weiß (White Box) bis schwarz (Black Box) eingeteilt werden. Weiß bedeutet, dass das System komplett auf Basis des Vorwissens aufgebaut werden kann, mit schwarz wird ausgedrückt, dass keinerlei Wissen über das System vorhanden ist. Zwischenstufen werden im grauen Bereich (Grey Box) der Farbskala positioniert.²⁸

2.6.1 White-Box-Modell

Ein White-Box-Modell verfügt nach *Bohn und Unbehauen* über die Charaktereigenschaften, dass seine physikalischen Ansätze und Gesetzmäßigkeiten bestimmt sowie seine Modellstruktur bekannt ist. Auch sind alle erforderlichen Parameter mechanischer und elektrischer Natur durch die Konstruktion vorgegeben oder ganzheitlich bekannt. Es sind keinerlei Experimente oder Messungen am System erforderlich, um seine Eigenschaften zu bestimmen. Des Weiteren kann das White-Box-Modell der analytischen respektive theoretischen Modellbildung zugeordnet werden.²⁹ Abb. 11 hebt diese Charakteristik grafisch hervor.

Kojarski und Lorenz ergänzen diese Eigenschaften durch eine optimale Strukturverknüpfung sowie die Zugriffsmöglichkeit auf beliebiger Ebene des Modells sowie die Sichtbarkeit eines jeden Details desselben. Als Nachteil wird angeführt, dass diese Form des Modells schwer erweiterbar ist, sowie eine Generalisierung nur mit hohem Aufwand umsetzbar ist. Ebenso sind die Integrationskosten solcher Modellformen in Gesamtsysteme erheblich.³⁰

Zusätzlich haben White-Box-Modelle die Eigenschaft, sehr einfach vom Menschen gelesen und interpretiert werden zu können.³¹

Abschließend ist die Charaktereigenschaft der Klassifizierung und Quantifizierung anzuführen.³²

Basierend auf der Analyse und Erhebung der Eigenschaften von White-Box-Modellen erfolgt im nachfolgenden Abschnitt 2.6.2 die Auseinandersetzung mit den Charakteristika von Grey-Box-Modellen.

²⁸ Vgl. Bohn/Unbehauen (2016), S. 236.

²⁹ Vgl. Bohn/Unbehauen (2016), S. 242.

³⁰ Vgl. Kojarski/Lorenz (2006), S. 247 f.

³¹ Vgl. Zhang/Nakadai/Fukumizu (2018), S. 213.

³² Vgl. Bruni u. a. (2015), S. 164.

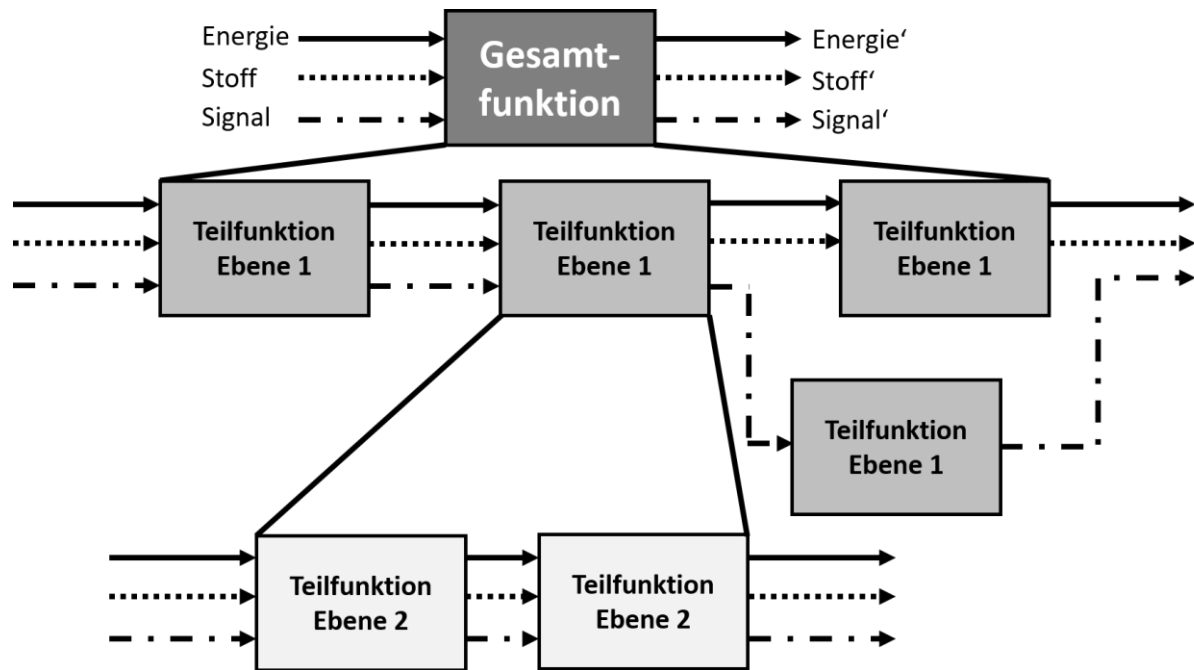


Abb. 11: Darstellung einer Funktionsstruktur mittels White-Box-Modell, Quelle: Pahl u. a. (2007), S. 45 (modifiziert).

2.6.2 Grey-Box-Modell

Sind mathematische und oder physikalische Gesetzmäßigkeiten teilweise bekannt, so wird ein Modell als Grey-Box-Modell klassifiziert. Diese unbekannt Parameter müssen mittels experimenteller Systemidentifikation beziehungsweise experimenteller Modellbildung bestimmt werden. In Abhängigkeit des vorhandenen Vorwissens über die Systemeigenschaften des zu bildenden Modells kann es in unterschiedliche Grautöne spezifiziert werden, wobei diese Einteilung einer gewissen Willkür unterliegt.³³

Grey-Box-Modell werden bewusst erzeugt, da sie die Möglichkeit bieten, die Schwächen von White-Box-Modellen wie auch Black-Box-Modellen entsprechend dem jeweiligen Modellzweck, gesteuert zu kompensieren. White-Box-Modelle haben Nachteile im Umgang mit unbekannt Umwelteinflüssen sowie zufälligen Effekten, was eine Stärke des Black-Box-Modells darstellt. Durch unterschiedliche Grauschattierungen lassen sich diese Vor- und Nachteile innerhalb des Simulationsmodells nutzbar machen.³⁴

Als weiteres Charakteristikum dieser Modellform ist die Eigenschaft zuzuordnen, dass Gesetzmäßigkeiten und Strukturaufbau, die auf äußeren Ebenen beziehungsweise Schichten positioniert sind gekannt werden, jene die weiter im inneren des Modell wirken, jedoch nicht.³⁵

Abschließend setzt sich der nachfolgende Abschnitt im Unterkapitel 2.6 mit den charakterbildenden Eigenschaften von Black-Box-Modellen auseinander.

³³ Vgl. Bohn/Unbehauen (2016), S. 242 f.

³⁴ Vgl. Bohlin (2006), S. 4.

³⁵ Vgl. Elkind u. a. (2006), S. 420.

2.6.3 Black-Box-Modell

Black-Box-Modelle stellen das Gegenüber von White-Box-Modellen dar. Es besteht keinerlei Kenntnis über Modellstruktur und Gesetzmäßigkeiten des abzubildenden Systems. Strukturbildung und anschließende Verwendung des Modells erfolgen auf Basis allgemeiner Approximationsansätze, die in der Lage sind, Systemverhalten angemessen nachbilden zu können. Keine Bedeutung haben physikalische Modellparameter – diese werden durch experimentelle Modellbildung mittels Eingangs-/ Ausgangsmessung ermittelt.³⁶

Ebenso kann ein Black-Box-Modell als funktionales Konzept definiert werden, da relevante Zusammenhänge und Zustände des Systems ausschließlich von außen beobachtet und beurteilt werden können. Durch Anlegen unterschiedlicher Eingangsgrößen kann nachfolgend durch Verallgemeinerung auf Systemzustand und Modellverhalten geschlossen werden. Dieser Systemansatz schließt ausdrücklich materielle Konkretisierung sowie Detailwissen vom inneren Aufbau des Systems aus. Er beschränkt sich ausschließlich auf das Verhalten seiner Umgebung und seiner Ganzheit.³⁷

Spezialwissen über den inneren Aufbau des Modells ist auch bei Tests nicht erforderlich. Diese leiten sich von den vorgegeben Testspezifikationen ab, wobei nur die Einhaltung dieser Spezifikationen Relevanz besitzt.³⁸

Auch eignen sich Black-Box-Modelle zur Erstellung verträglicher und sinnvoller Funktionsstrukturen in der Produkt- und Systementwicklung, da durch Anwendung dieses Entwicklungsansatzes die Möglichkeit gegeben ist, Funktionen und Fähigkeiten von Teilsystemen vorerst auszublenden, wie es Abb. 12 grafisch verdeutlicht.³⁹

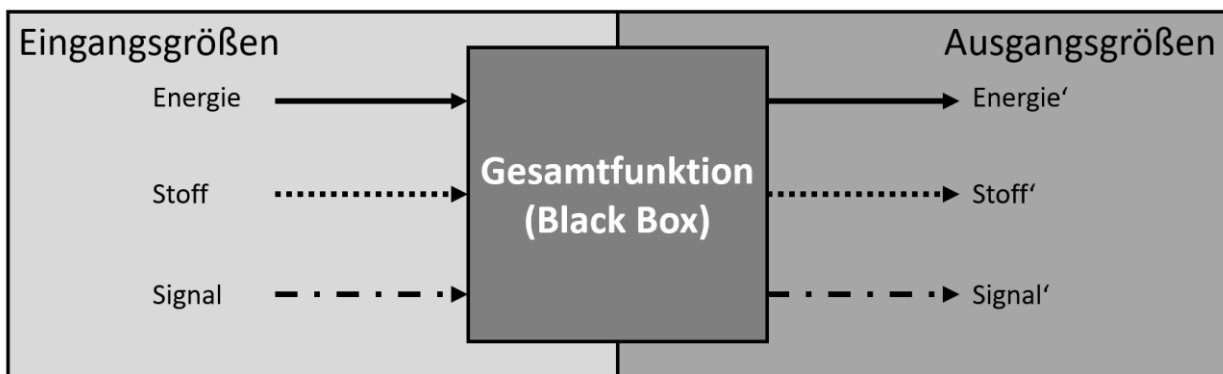


Abb. 12: Darstellung einer Gesamtfunktion als Black-Box-Modell,
Quelle: Eigner/Roubanov/Zafirov (2014), S. 17 (leicht modifiziert) in Anlehnung an Pahl/Beitz.

An dieser Stelle endet die Analyse und Betrachtung der Grundformen von Modellen und es wird auf den Einsatzzweck von Modellen in der Naturwissenschaft übergeleitet.

³⁶ Vgl. Bohn/Unbehauen (2016), S. 242.

³⁷ Vgl. Ropohl (2009), S. 75 f.

³⁸ Vgl. Gutenschwager u. a. (2017), S. 212 f.

³⁹ Vgl. Pahl u. a. (2007), S. 44.

2.7 Einsatzzweck von Modellen

Wie bereits in Abschnitt 2.1.5 abgegrenzt, fokussiert sich diese Arbeit auf Grund der angestrebten Zielsetzung einer modellbasierten virtuellen Inbetriebnahme von Karosseriebauanlagen auf jene Disziplinen der Ingenieurwissenschaften, die für die Konzeptionierung, Planung, Realisierung und Betrieb dieser Produktionseinrichtungen notwendig sind.

Besondere Berücksichtigung finden in diesem Zusammenhang die Teildisziplinen der Automatisierungstechnik wie Elektrotechnik inklusive Steuerungs-, Regelungs-, und Messtechnik, Maschinenbau sowie Informatik mit den Spezialgebieten der Informationsverarbeitung und Informationstechnik.⁴⁰ Nachfolgend wird ein Auszug unterschiedlicher Einsatzzwecke ausgewiesen.

Zur Festlegung konkreter geometrischer Eigenschaften werden *Gestaltmodelle* angewandt, welche üblicherweise in Form von 2D/3D-CAD-Modellen Beschreibung finden.⁴¹ In der Produktmodellierung werden diese als *Geometriemodell* entweder konventionell, parametrisch, featurebasierend oder wissensbasierend weiterverarbeitet, um die Geometrie des jeweiligen Produktes eindeutig zu beschreiben.⁴² Darauf aufbauend können *Kinematikmodelle* eingebunden werden, um Verhalten und Eigenschaften der ebenen Kinematik, der Raumkinematik sowie der Robotik abzubilden. Diese Modelle bilden weiterführend die Grundlage dynamischer Modelle, da Dynamik auf Kinematik aufbaut.⁴³

Eine Anwendungsform hierfür ist die Modellbildung von Mehrkörpersystemen (MKS) als *MKS-Modell*, damit komplexe Systeme, bestehend aus massebehafteten Körpern und masselosen Verbindungselementen, hinsichtlich ihrer dynamischen Ausprägungen und Verhaltensweisen simuliert werden können.⁴⁴ Weiters dienen *Reglermodelle* dazu, etwaige Stabilisierungsaufgaben, wie zum Beispiel die Erlangung und Aufrechterhaltung einer Gleichgewichtsposition, mittels Linear-Zeitinvariantem-Streckenmodell (LZI/LTI) zu erreichen.⁴⁵ Auch die automatische Generierung von *Physikmodellen* lässt der aktuelle Stand der Technik mittlerweile zu. So ist es mittels geeigneter Software möglich, Oberflächenbeschaffenheit, E-Modul und Materialwerte zu visualisieren und zu simulieren, welche dem Designer im Produktentstehungsprozess (PEP) als Entscheidungshilfe dienen.⁴⁶

In der Elektrotechnik gelangen Modelle in Gestalt von Flussdiagrammen und elektrischen Netzwerken zur Anwendung.⁴⁷ Mittels *Netzwerkmodell* lassen sich komplizierte elektrotechnische Schaltungen bestimmen, mathematisch lösen und als Simulationsmodell darstellen.⁴⁸

⁴⁰ Vgl. Reinhardt (1996), S. 5 f.

⁴¹ Vgl. Vajna u. a. (2009), S. 136.

⁴² Vgl. VDI 2209 (März 2009), S. 13.

⁴³ Vgl. Husty u. a. (1997), S. 530.

⁴⁴ Vgl. Rill/Schaeffer (2017), S. 1 f.

⁴⁵ Vgl. Reinschke (2014), S. 553.

⁴⁶ Vgl. Schenk/Schumann (2016), S. 81 f.

⁴⁷ Vgl. Vajna u. a. (2009), S. 133.

⁴⁸ Vgl. Müller (2015), S. 171.

So lassen sich die hochkomplexen chemischen und physikalischen Vorgänge innerhalb einer Batterie durch Modellbildung mittels Ersatzschaltbildern übersichtlich darstellen. Umgesetzt wird dies durch das Nachbilden elektrischen Verhaltens und Darstellung desselben durch einfache Bauteile wie beispielsweise Induktivitäten, Kapazitäten, Widerständen oder Spannungsquellen.⁴⁹

In der Informatik und Informationsverarbeitung dienen Modelle zum Beispiel beim Entwerfen von Datenstrukturen und Algorithmen.⁵⁰

Eine ergänzende Klassifizierung hinsichtlich Einsatzzweck von Modellen in den Ingenieurwissenschaften zeigt nachfolgende Tabelle Tab. 1. Zusätzlich erfolgt im Sinne der Vervollständigung eine weitere Einteilung nach Art der Information und Grad der Abstraktion.

| Einsatzzweck | Art der Information | Grad der Abstraktion |
|--|----------------------|--------------------------|
| Beschreibungsmodell | quantitatives Modell | deterministisches Modell |
| Erklärungs- oder Kausalmodell | qualitatives Modell | stochastisches Modell |
| Prognosemodell | | statisches Modell |
| Simulationsmodell | | dynamisches Modell |
| bedientheoretisches Modell | | Totalmodell |
| Entscheidungs- bzw. Optimierungsmodell | | Partialmodell |

Tab. 1: Modellklassifizierung nach Einsatzzweck, Informationsart und Abstraktionsgrad, Quelle: Arnold u. a. (2008), S. 36 (modifiziert).

Die Einteilung von Modellen nach Tab. 1 hinsichtlich Einsatzzweck findet sich auch bei Page/Kreutzer wieder, doch werden in diesem Werk Beschreibungsmodell, Erklärungsmodell, Prognosemodell und Entscheidungsmodell der Kategorie der Zielsetzung zugeordnet.⁵¹

Alle in diesem Unterkapitel angeführten Einsatzzwecke von Modellen ermöglichen in weiterer Folge die Einbindung und Eingliederung in *Anlagenmodelle*, mit deren Hilfe Produktionsmittel und Fertigungsanlagen in gesamtheitlicher Form virtuell zur Darstellung gelangen. Diese erlauben es weiterführend, Bearbeitungsstrategien und Produktionsprozesse bezüglich ihrer Auswirkungen und Tauglichkeit einer eingehenden Beurteilung zu unterziehen.⁵²

Diese gesamtheitliche Darstellung von Produktionsanlagen mittels Simulation ist auch die Basis für die virtuelle Inbetriebnahme, welche den wissenschaftlichen Kern dieser Arbeit bildet.

Zunächst sind jedoch die erforderlichen Prinzipien der Abstraktion im Zuge der Modellbildung im anschließenden Unterkapitel zu erörtern.

⁴⁹ Vgl. Böker/Paerschke/Boggasch (2017), S. 274.

⁵⁰ Vgl. Broy/Steinbrüggen (2004), S. 87.

⁵¹ Vgl. Page/Kreutzer (2005), S. 5 f.

⁵² Vgl. Botthof/Hartmann (2015), S. 118.

2.8 Abstraktionsprinzipien

Abstraktion spielt in der Modellentwicklung und Modellbildung eine wesentliche Rolle.⁵³ Dahingehend wird in den nächsten beiden Abschnitten die Definition dieses Begriffs einer Analyse unterzogen. Anschließend werden die wichtigsten Prinzipien der Abstrahierung, die im Zuge der Entwicklung von Modellen zur Anwendung gelangen, behandelt.

2.8.1 Begriffsbestimmung

Abstraktion bedeutet, konstante objektbestimmende und objektdefinierende Eigenschaften eines Gegenstandes in einen variablen Zustand zu transformieren. Es erfolgt eine schrittweise Erweiterung der Begriffsumfänge, welche diesen Gegenstand spezifizieren. Als Beispiel hierfür ist der Übergang des gleichschenkelig-rechtwinkligen Dreiecks zum gleichschenkeligen Dreieck als Veranschaulichung anzuführen, welches den Obergriff des gleichschenkelig-rechtwinkligen Dreiecks darstellt.⁵⁴

Auch kann die Abstraktion als Vorgang definiert werden, bei dem nicht das Weglassen von Merkmalen im Vordergrund steht, sondern das Variabelmachen derselben.⁵⁵

Eine weitere Erklärung des Begriffs der Abstraktion ist jene, dass Abstraktion die Trennung vom Allgemeinen und Besonderen zum Einzelnen darstellt. Ebenso erstreckt sie sich vom Besonderen auf das Allgemeine.⁵⁶

2.8.2 Prinzipien der Abstraktion

Nach durchgeführter Begriffsbestimmung setzt sich dieser Abschnitt mit jenen Prinzipien der Abstraktion auseinander, die für den praktischen Teil dieser Arbeit von Relevanz sind.

2.8.2.1 Aggregation/Partizipation

Aggregation ist eine Form der Abstraktion, dessen Wirkungsweise sich durch das Zusammenfassen von Objekten auszeichnet, welche in bestimmten Art und Weisen kooperieren. Das zusammengefasste abstrakte Objekt wird als Aggregat bezeichnet. Das Aggregat verkörpert das virtuelle oder physische Zusammenwirken der einzelnen und genauen Objekte und ersetzt diese. Abgrenzungskriterium der detaillierten zusammenfassenden Objekte ist ausschließlich das Zusammenwirken dieser detaillierten Objekte in unmittelbarer Form. Wirkt ein detailliertes Objekt mit anderen detaillierten Objekten zusammen, geht dasselbe in alle Aggregaten mit ein, an dem es mitwirkt. Als Beispiel kann hier das Zusammenfassen von Einzelteilen zu Baugruppen angeführt werden.⁵⁷ Dieses Prinzip stellt Abb. 13 in grafischer Form dar.

⁵³ Vgl. Alt (2009), S. 43.

⁵⁴ Vgl. Stachowiak (1973), S. 130 f.

⁵⁵ Vgl. Klaus (1963), S. 168.

⁵⁶ Vgl. Koch/Schick (Hrsg.) (2002), S. 198.

⁵⁷ Vgl. Winter (1991), S. 20.

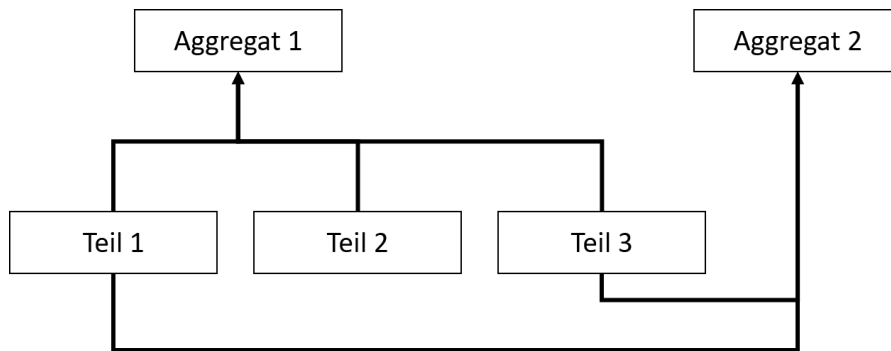


Abb. 13: Prinzip der Abstraktion durch Aggregation, Quelle: Eigene Darstellung in Anlehnung an Winter (1991), S. 20.

Ebenso wird von Aggregation gesprochen, wenn ein Mitarbeiter nicht ausschließlich Bestandteil eines Projektes ist, sondern durch eine Querschnittsfunktion Bestandteil mehrere Projekte respektive Projektaufgaben ist.⁵⁸

Auch bringt die Aggregation im Softwareentwurf zum Ausdruck, dass Instanzen einer Klasse auch Teile von Instanzen anderer Klassen repräsentieren können.⁵⁹

Abschließend stellt Abb. 14 die Wirkrichtung von Aggregation und deren Gegenüber und Umkehr, der Partizipation, bildlich dar.

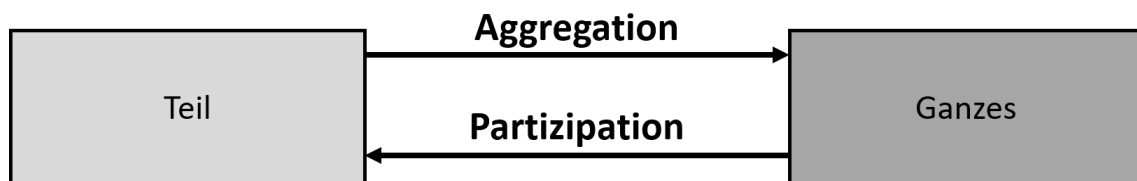


Abb. 14: Wirkrichtung des Abstraktionsprinzips der Aggregation und Partizipation, Quelle: Eigene Darstellung.

2.8.2.2 Generalisierung/Spezialisierung

Generalisierung und Spezialisierung verstärken als Modellierungswerkzeug die Modellierungspotenz semantischer Datenmodelle. Von *Generalisierung* wird gesprochen, wenn der Bearbeitungsfluss vom spezielleren System zum allgemeinen System wirkt, in umgekehrter Wirkrichtung von *Spezialisierung*.⁶⁰

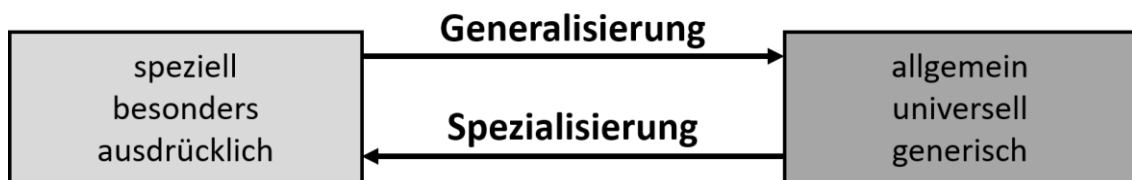


Abb. 15: Wirkrichtung des Abstraktionsprinzips der Generalisierung/Spezialisierung, Quelle: eigene Darstellung.

Abb. 15 stellt dieses Wirkprinzip in grafischer Form dar.

⁵⁸ Vgl. Kleuker (2018), S. 137.

⁵⁹ Vgl. Seidl u. a. (2015), S. 67.

⁶⁰ Vgl. Staud (2005), S. 149 ff.

Dieses Abstraktionsprinzip dient dazu, ein Modell bezüglich seiner Semantik eine hierarchische Strukturierung zu Teil werden zu lassen. Das speziellere Element fügt zusätzliche Eigenschaften hinzu, die sich verträglich zum allgemeinen Element verhalten.⁶¹ Diese Definition kann weiter in die Formulierung übergeführt werden, dass eine Spezialisierung einer Klasse X zu einer Klasse Y vorliegt, wenn jedes Element aus X auch ein Element aus Y ist.⁶² Verdeutlicht wird diese Wirkweise durch Abb. 16.

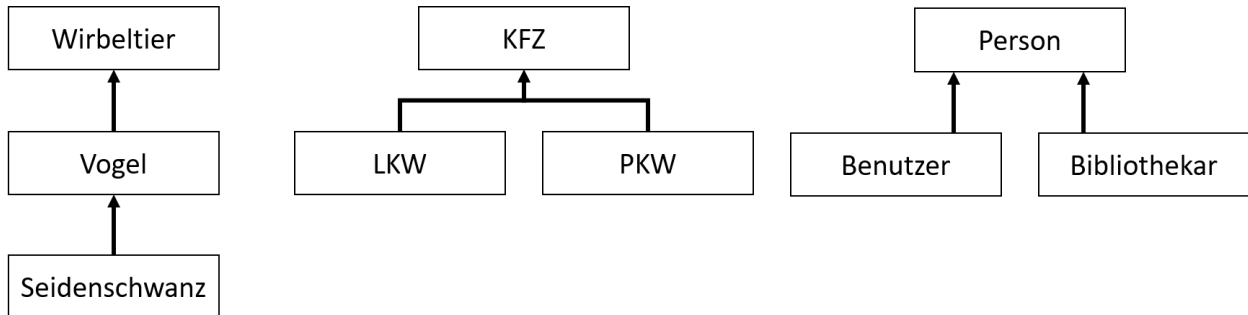


Abb. 16: Hierarchischer Strukturaufbau von Generalisierung und Spezialisierung, Quelle: Kahlbrandt (2001), S. 51 (leicht modifiziert).

Alternativ wird dieses Prinzip in der Literatur ebenfalls als *Generalisation* bezeichnet, wobei Generalisation als „die Betrachtung einer Klasse von einzelnen Objekten als ein generisches Objekt“⁶³ konkretisiert wird.

2.8.2.3 Klassifizierung/Instanziierung

Eine Klasse ist eine Vorlage für eine Sache. Eine Instanz ist eine tatsächliche Sache von Dingen, die alle die gleiche Menge von Funktionen, die selbe Struktur und das gleiche Verhalten besitzen. Die Vorlage, welche die Klasse definiert, enthält alle Elementeigenschaften, die weitervererbt werden. Das heißt nicht nur Merkmale, sondern auch prozedurale und strukturelle Beziehungen sowie Zustände werden im Zuge der Vererbung von der Klasse auf ihre Subklassen weitergegeben. Eine Instanz kann keine Funktion und keinen Zustand besitzen, die nicht durch ihre Klasse spezifiziert wird.⁶⁴ Die Wirkrichtung dieses Prinzips zeigt Abb. 17.

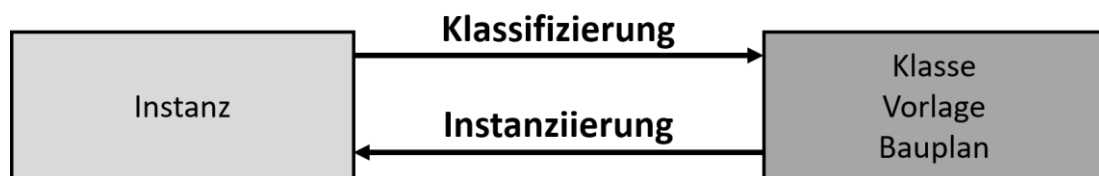


Abb. 17: Wirkrichtung des Abstraktionsprinzips der Klassifizierung/Instanziierung, Quelle: eigene Darstellung.

Eigenschaften und Verhalten einer gleichartigen Menge von Objekten wird durch ein Klasse zusammengefasst.⁶⁵ Auch können Klassen als Sammlung von Attributen und Methoden verstanden werden, die den

⁶¹ Vgl. Müller (2015), S. 19., zitiert nach: Oestereich (2002).

⁶² Vgl. Kahlbrandt (2001), S. 51.

⁶³ Vinek/Rennert/Tjoa (1982), S. 200., zitiert nach: Smith J.M./Smith D.C.P (1977), S. 105 - 133.

⁶⁴ Vgl. Dori (2016), S. 256 f.

⁶⁵ Vgl. Claussen (1998), S. 23.

Status und das Verhalten ihrer Instanzen definieren und beschreiben, wobei Klassen durch Assoziation und Vererbungsverhältnisse miteinander verbunden sind.⁶⁶

Im Gegensatz zur Generalisierung/Spezialisierung, die Strukturbeziehungen herstellt – jede Ebene der Hierarchie enthält Spezialisierungen der darüberliegenden Ebene – kann eine Instanz einer Klasse nicht weiter spezialisiert werden.⁶⁷ Dieses grundlegende unterschiedliche Verhalten verdeutlicht Abb. 18.

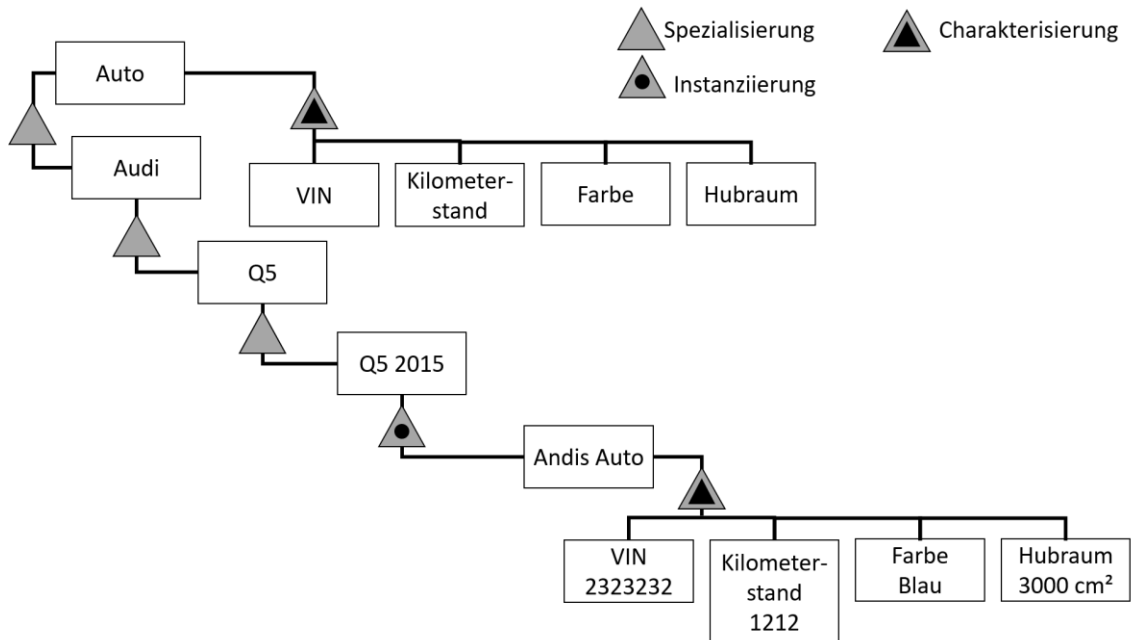


Abb. 18: Unterschied zwischen Generalisierung und Klassifizierung, Quelle: Dori (2016), S. 288 (modifiziert).

2.8.2.4 Komposition/Dekomposition

Ein weiteres Prinzip der Abstraktion wird durch das Verfahren der Komposition respektive Dekomposition dargestellt. *Komposition* beschreibt die Zusammensetzung eines Gesamtsystems durch Teilsysteme mittels eindeutig festgelegter Regeln. Umgekehrt werden durch *Dekomposition* Teilkomponenten oder Teilfunktionen eines Systems separat beschrieben.⁶⁸ Die Wirkrichtung dieses Abstraktionsprinzips hebt Abb. 19 hervor.

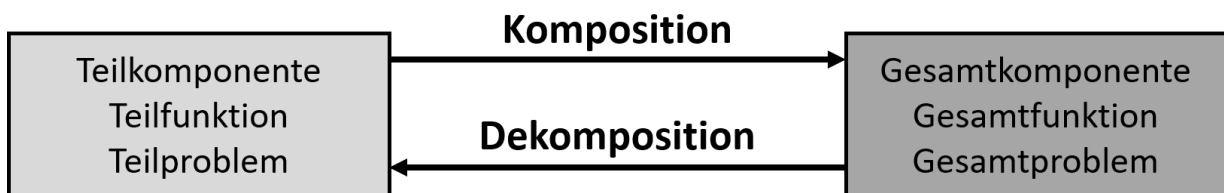


Abb. 19: Wirkrichtung des Abstraktionsprinzips der Komposition/Dekomposition, Quelle: eigene Darstellung.

⁶⁶ Vgl. Rumpe (2016), S. 17.

⁶⁷ Vgl. Dori (2016), S. 256 f.

⁶⁸ Vgl. VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 21.

Des Weiteren kann das Prinzip der Komposition als Aggregationsform beschrieben werden, deren Existenz der Teilkomponenten, Teilfunktionen oder Teilproblemen vom jeweiligen Ganzen abhängt. Teile des Ganzen existieren nur so lange, so lange auch das Ganze existiert. Wird das Ganze einer Zerstörung zugeführt, werden auch alle Teile, die zu diesem Ganzen gehören, vernichtet.⁶⁹

Ein Gesamtsystem, das durch Komposition zusammengesetzt wurde, kann auch als *Kompositum* bezeichnet werden. Die innere Struktur sowie Aufbau und Abhängigkeiten des Kompositums können durch ein Kompositionsdiagramm grafisch dargestellt werden.⁷⁰ Abb. 20 stellt dies beispielhaft dar.

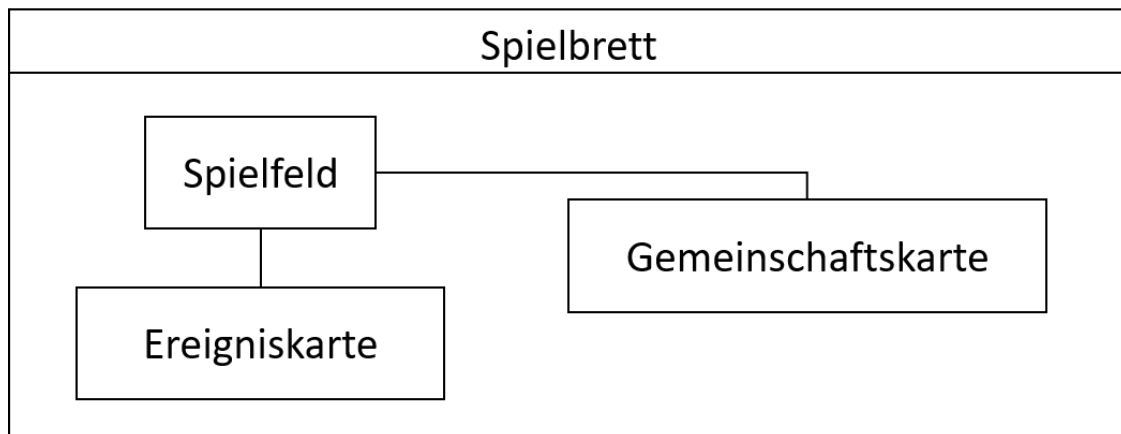


Abb. 20: Beschreibung des Abstraktionsprinzips der Komposition/Dekomposition mittels Kompositionsdiagramm anhand eines Brettspiels, Quelle: Seemann/von Gutenberg (2006), S. 129 (modifiziert).

2.8.2.5 Abgrenzung

Aufgrund des begrenzten Umfangs dieser Arbeit kann auf die verbleibenden Abstraktionsprinzipen *Assoziation* und *Selektion*,⁷¹ sowie *Exhibition* und *Charakterisierung*⁷² nicht näher eingegangen werden.

Das nachfolgende und vorletzte Unterkapitel von Kapitel 2 befasst sich mit Werkzeugen zur Modellbildung von Produktionsanlagen und deren Eignung zur Beschreibung bestimmter Systemeigenschaften.

2.9 Werkzeuge zur Modellbildung

Dieses Unterkapitel setzt sich mit den zur Verfügung stehenden Werkzeugen zur Modellbildung auseinander. Es führt und listet Werkzeuge auf, welche die Fähigkeit und Eignung besitzen, besondere Systemeigenschaften von Produktionsanlagen abzubilden. Eingrenzend ist anzuführen, dass sich dieser Teil der Arbeit mit dem allgemeinen Portfolio und Spektrum der zur Verfügung stehenden Werkzeuge im Modellentwicklungsprozess auseinandersetzt. Einen entsprechenden Auszug über diese Werkzeuge bildet Tab. 2.

⁶⁹ Vgl. Kahlbrandt (2001), S. 144.

⁷⁰ Vgl. Seemann/von Gutenberg (2006), S. 129.

⁷¹ Vgl. Winter (1991), S. 28.

⁷² Vgl. Dori (2016), S. 215.

| Beschreibungsmittel | Notation | Abbildung von Hierarchie | Komposition Dekomposition | Zeitverhalten |
|--|-------------------|--------------------------|---------------------------|---------------|
| Natürliche Sprache | textuell | | | statisch |
| Zeichnungen | grafisch | | X | statisch |
| Prozessorientierte Beschreibungsmittel | grafisch/textuell | X | X | statisch |
| Darstellungsgraphen | grafisch | X | | statisch |
| Datenflussdiagramme | grafisch | | | statisch |
| Entity-Relationship-Modelle | grafisch | | X | statisch |
| Unified Modelling Language | grafisch/textuell | X | X | statisch |
| Tabellen und Listen | grafisch/textuell | | | statisch |
| Organigramm | grafisch | X | | statisch |
| Vorranggraphen | grafisch | | X | statisch |
| Diagramme | grafisch | | | statisch |
| Mathematische Formeln | | X | X | diskret |
| Explorative Statistik | | | | statisch |
| Entscheidungstabellen | grafisch/textuell | | X | statisch |
| Constraint Programming | | | | statisch |
| Pseudocode | textuell | X | X | diskret |
| Programmiersprache | textuell | X | X | diskret |
| Simulationssprache | textuell | X | X | diskret |
| Struktogramme | grafisch/textuell | | | diskret |
| Programmablaufpläne | grafisch/textuell | | | diskret |
| Technik- und layoutorientierte Bausteine | grafisch | X | X | diskret |

Tab. 2: Modellierungssprachen kategorisiert nach Eignung zur Abbildung von Systemeigenschaften, Quelle: VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 22 (modifiziert).

Die genaue Beschreibung und Erläuterung jener Werkzeuge, die zur Erzeugung und Erstellung des Modells zur Virtuellen Inbetriebnahme verwendet wurden, erfolgt in Unterkapitel 4.3.

Den Abschluss dieses Kapitels bildet die nachfolgende Zusammenfassung der gewonnenen Erkenntnisse im Bereich der Modellbildung.

2.10 Zusammenfassung

Das abschließende Unterkapitel dient der Zusammenfassung, Reflexion und Diskussion der bis zu diesem Zeitpunkt erarbeiteten Fakten und Erkenntnisse.

Im ersten Teil dieser Arbeit wurde der Begriff *Modell* näher bestimmt. Es wurde festgestellt, dass die Begrifflichkeit *Modell* seinen Ursprung im lateinischen Wort *modulus* hat, das die Bedeutung *Maß* und *Maßstab* umfasst. Anschließend erfolgte die Begriffsbestimmung von *Modell* und dessen Beschreibungen nach *Stachowiak*, der Abbildungsmerkmal, Verkürzungsmerkmal sowie das Pragmatische Merkmal ins Zentrum seiner Definition eines Modells rückt⁷³.

Nach weiteren Begriffsbestimmungen wurde der Fokus auf den Modellbildungsprozess sowie auf die Auseinandersetzung mit einschlägigen normativen Regelwerken gelegt, die sich mit diesem Thema befassen. Hier konnte festgestellt werden, dass die Festlegung eines System und der dazugehörigen Grenzen Basis für die weiteren Schritte im Modellentwicklungsprozess naturwissenschaftlicher Modelle ist. Ebenso spielt der Modellzweck eine bedeutende Rolle für nachfolgende Entscheidungen hinsichtlich anzuwendender Methoden und Werkzeuge im Zuge der Modellerstellung.⁷⁴

Sowohl die Fachliteratur wie auch die normativen Regelwerke betonen, dass die Entwicklung eines Modells einen iterativen Vorgang darstellt. Nur durch mehrmaliges Durchlaufen der Phasen Systemdefinition, Modellbildung, Modellkontrolle und Modellanpassung kann eine Aussage hinsichtlich Güte und Aussagekraft eines Modells getroffen werden.⁷⁵ Ebenso zentral ist der Grundsatz, Modelle so einfach wie möglich zu gestalten.⁷⁶ Auch der Teamzusammensetzung zur Erstellung von Modellen wird eine wichtige Rolle beigemessen, da hier die Wissensquellen bezüglich Systemverhalten und Lösungskompetenz ansässig sind.

Anschließend wurden die Methoden der Modellbildung erörtert, wobei diese in analytische und experimentelle Modellbildung eingeteilt werden können. Erstere findet Anwendung, wenn das Systemverhalten des Realsystems bekannt ist, zweitere wenn kein Wissen über dieses vorhanden ist.⁷⁷ Ebenfalls kann der Grad über das Systemwissen respektive Modellwissen über die Modellform ausgedrückt werden, wie es in Unterkapitel 2.6 eingehend behandelt wurde.

Darauf basierend erfolgte ein Überblick betreffend der Einsatzmöglichkeiten von Modellen in den Ingenieursdisziplinen des Maschinenbaus, der Elektrotechnik sowie der Informationstechnik. Abschließend wurden die Prinzipien der Abstraktion einer Analyse unterzogen sowie ein Überblick über die Einsatzmöglichkeiten verschiedener Werkzeuge im Modellbildungsprozess gegeben.

Mit Abschluss von Kapitel 2 wurde der erste theoretische Teilbereich für die erfolgreiche Umsetzung des praktischen Teils dieser Arbeit abgedeckt. Das nachfolgende Kapitel hat den Zweck, die erforderlichen Grundlagen betreffend der VIBN zu erarbeiten und zu schaffen.

⁷³ Vgl. Stachowiak (1973), S. 129 - 133.

⁷⁴ Vgl. Bossel (1989), S. 11 - 32.

⁷⁵ Vgl. VDI 3843 Blatt 1 (Februar 2018), S. 6.

⁷⁶ Roddeck (2017), S. 15f.

⁷⁷ Vgl. Glöckler (2018), S. 117 - 121.

3 VIRTUELLE INBETRIEBNAHME

Kapitel 3 behandelt die theoretischen Grundlagen bezüglich der Durchführung einer Virtuellen Inbetriebnahme (VIBN). Zuerst erfolgt die Begriffsbestimmung betreffend dieses Ausdrucks, gefolgt von einer Analyse, die eine Zusammenfassung von Motiven zur Durchführung einer VIBN zum Inhalt hat. Danach werden jene Steuerungsebenen analysiert, welche für die Durchführung einer VIBN geeignet sind. Nachfolgend finden notwendige Definitionen von Systemeigenschaften wie Echtzeit, Virtuelle Zeitachse oder dem Abtasttheorem statt. Danach wird eine Untersuchung unterschiedlicher Grundsatzmethoden und Herangehensweisen zur Durchführung einer Virtuellen Inbetriebnahme sowie der darin eingesetzten Modellarten durchgeführt. Abschluss findet dieses Kapitel mit einer Diskussion über die erarbeiteten Erkenntnisse aus diesem Teilbereich der Arbeit.

3.1 Inbetriebnahme - IBN

Inbetriebnahme wird nach DIN 32541 als Bereitstellung von Maschinen oder vergleichbaren Arbeitsmitteln technischer Natur zur betrieblichen Nutzung definiert.⁷⁸ Weiters wird nach EN 62337 unter diesem Begriff jene Teilphase bezeichnet, unter der die Aktivitäten der Funktionsprüfung, Ausrichtung, Einstellung und Reinigung von Geräten und Maschinen einer technischen Anlage zusammengefasst werden, welche es zu errichten gilt.⁷⁹ DIN 31051 wiederum definiert diesen Begriff als „Auslösen der *Funktionserfüllung*“⁸⁰.

Zeugträger charakterisiert Inbetriebnahme als Herstellen der Funktionsbereitschaft aller zuvor installierten Einzelkomponenten inklusive des funktionalen Zusammenwirkens derselben sowie das Prüfen auf korrekte Einzelfunktion der Teilkomponenten. Voraussetzung dafür ist die davor eingeschaltete Gesamtanlage sowie das Hochfahren dieser auf das geforderte Leistungs- und Prozessniveau. Ergebnis daraus ist eine funktionsfähige Fertigungsanlage, die für die Abnahme durch den Auftraggeber bereit ist.⁸¹

Die Überleitung von der *Inbetriebnahme* zur *Virtuellen Inbetriebnahme* wird im Regelwerk VDI-4499 Blatt 1 hergestellt, in der die VIBN als Teilbereich des digitalen Fabrikbetriebs ausgewiesen wird, wie es Abb. 21 verdeutlicht. Unter dem Oberbegriff *Digitale Fabrik* werden Werkzeuge, Methoden und digitale Modelle zu einem umfassenden Netzwerk zusammengefasst. Teil davon sind unter anderem auch die dreidimensionale Visualisierung und Simulation mit der Zielsetzung, Produkte, Ressourcen, Prozesse und Strukturen der realen Fabrik einer ganzheitlichen Planung zuzuführen.⁸²

Ebenso dienen der digitale Fabrikbetrieb und seine Teilbereiche zur kontinuierlichen Verbesserung und Evaluierung der Realfabrik.⁸³

⁷⁸ Vgl. DIN 32541 (Mai 1977), S. 1.

⁷⁹ Vgl. DIN EN 62337 (September 2007), S. 5.

⁸⁰ DIN 31051 (September 2012), S. 10.

⁸¹ Vgl. Zeugträger (1998), S. 29.

⁸² Vgl. VDI 4499 Blatt 1 (Februar 2008), S. 3.

⁸³ Vgl. VDI 4499 Blatt 1 (Februar 2008), S. 3.

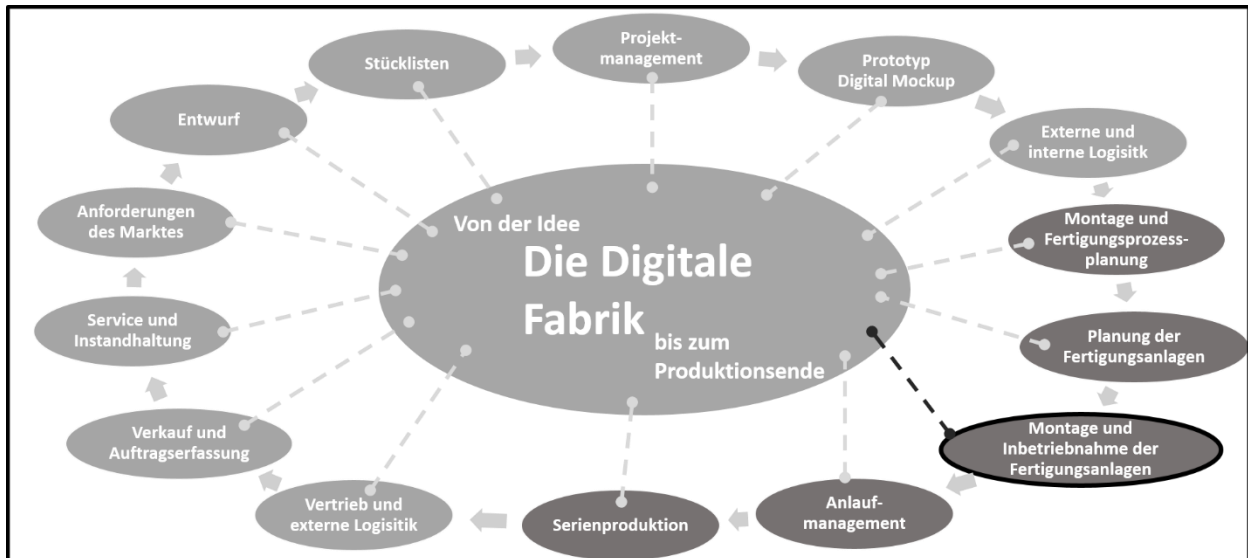


Abb. 21: Teildisziplinen der Digitalen Fabrik, Quelle: VDI 4499 Blatt 1 (Februar 2008), S. 5 (modifiziert).

3.2 Virtuelle Inbetriebnahme - VIBN

Dieses Unterkapitel befasst sich mit der Definition der Methode der Virtuellen Inbetriebnahme. In den nachfolgenden Abschnitten werden unterschiedliche Festlegungen bezüglich dieser Begrifflichkeit analysiert.

3.2.1 Definition nach VDI 4499 Blatt 2

Dieses Regelwerk der VDI beschreibt die VIBN als letzten Schritt in der Planung von Fertigungsanlagen. Voraussetzung zur Durchführung einer VIBN sind alle relevanten Daten, die für den virtuellen Betrieb einer Produktionsanlage erforderlich sind. Dazu zählt das Modell der abzubildenden Produktionsanlage, das wiederum die Teilgewerke der Elektrik, der Informationstechnik und der Mechanik beinhaltet. Sie ist vor der realen IBN durchzuführen, wie es Abb. 22 verdeutlicht. Die Virtuelle Inbetriebnahme unterscheidet sich zur Simulation dadurch, dass für die Durchführung dieser ein Modell verwendet wird, das aus realen und virtuellen Anlagenteilen besteht, und nicht wie eine Simulation auf einem reinen mathematischen Modell basiert. In Abhängigkeit des Inbetriebnahmefortschrittes werden dabei unterschiedlich intensiv reale Anlagenkomponenten wie Human Machine Interfaces (HMI) und/ oder Steuerungen sowie pneumatische, hydraulische oder elektrotechnische Realkomponenten mittels VIBN zu einem geschlossenen Regelkreis vernetzt. Einsatz findet die VIBN beim Test von Speicherprogrammierbaren Steuerungen (SPS), zur Analyse von Maschinenverhalten oder Verifikation und Validierung von Roboterprogrammen aus der Offlineprogrammierung.⁸⁴

⁸⁴ Vgl. VDI 4499 Blatt 2 (Mai 2011), S. 11 f.

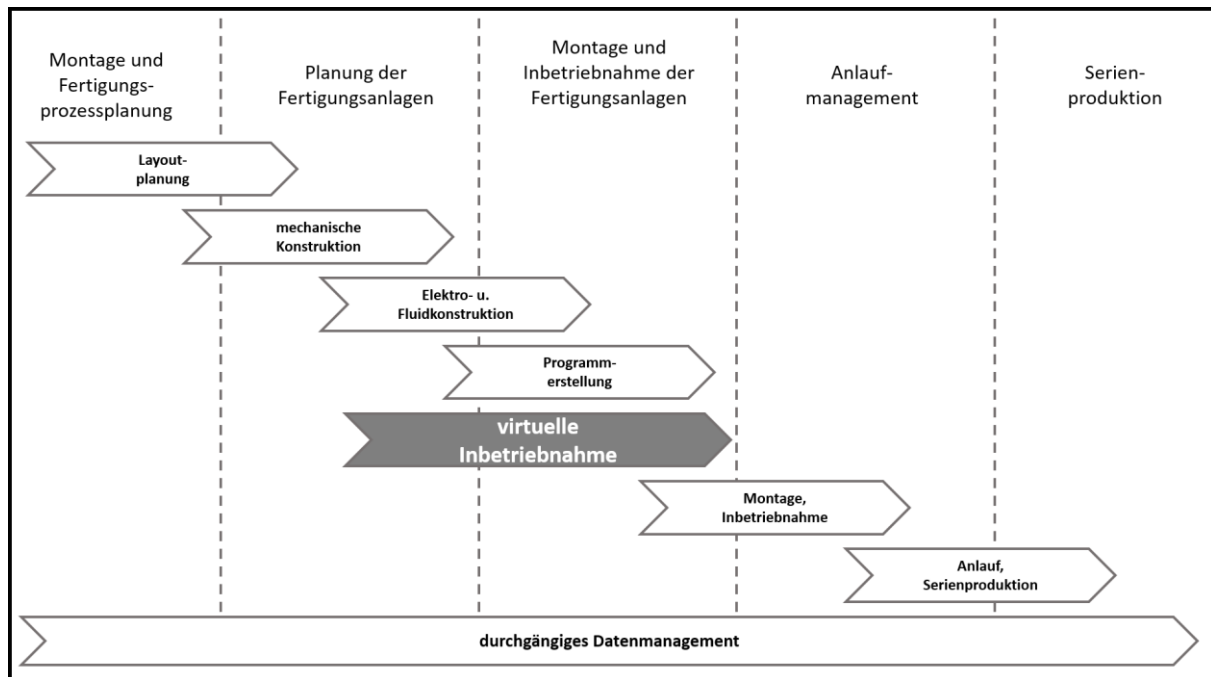


Abb. 22: Positionierung der VIBN in Bezug auf die Planungsschritte von Fertigungsanlagen, Quelle: VDI 4499 Blatt 2 (Mai 2011), S. 10 (leicht modifiziert).

3.2.2 Definition VDI/VDE 3693 Blatt 1

In der VDI/VDE 3693 Blatt 1 wird die VIBN als „Inbetriebnahme, die das entwicklungsbegleitende Testen einzelner Komponenten und Teilfunktionen des Automatisierungssystems mithilfe von auf die jeweilige Aufgabenstellung abgestimmten Simulationsmethoden und -modellen umfasst“⁸⁵ spezifiziert.

Anwendung findet die Virtuelle Inbetriebnahme mit ihrer Methodik in folgenden Domänen der Automatisierungstechnik:⁸⁶

- Prozessautomatisierung
- Fertigungsautomatisierung
- Intralogistik
- Gebäudeautomatisierung
- Energieautomatisierung
- Verkehrsautomatisierung

Von der Anwendung der VIBN sind Entscheidungsträger bezüglich VIBN-Durchführung, Inbetriebnahmeingenieure, Anlagen- und Produktionsplaner, Vertriebsingenieure, Elektroplaner, Softwareentwickler und Automatisierungstechniker unmittelbar respektive mittelbar betroffen sind.⁸⁷

⁸⁵ VDI/VDE 3693 Blatt 1 (August 2016), S. 5.

⁸⁶ Vgl. VDI/VDE 3693 Blatt 1 (August 2016), S. 2.

⁸⁷ Vgl. VDI/VDE 3693 Blatt 1 (August 2016), S. 3.

3.2.3 Definition nach Wunsch

Wunsch unterscheidet in seiner Dissertation zwischen zwei Begrifflichkeiten, der Virtuellen Inbetriebsetzung (VIBS) und der Virtuellen Inbetriebnahme. Diese Abgrenzung ist erforderlich, da sich diese beiden Methoden bezüglich ihrer Inhalte, Wesenszüge und Zielsetzungen grundlegend unterscheiden. Während die VIBS frühe Entwicklungsphasen und Verifikationen von steuerungstechnischen Abläufen mittels Simulationsmodell charakterisiert, dient die VIBN zum Steuerungstest am Ende des Entwicklungsprozesses durch Modelleinsatz. Des Weiteren ermöglicht die VIBN eine Kopplung virtueller oder realer Steuerungen an das Simulationsmodell, wobei dieses eine hinreichende Abtastrate gewährleistet, sodass alle Steuerungssignale dem Simulationsmodell zur Verfügung gestellt und von diesem verarbeitet werden können. Die VIBS dient zur Konzeptevaluierung und Überprüfung der sequentiellen Richtigkeit des Gutablaufes mittels schlichtem Simulationsmodell. Auch werden, im Gegensatz zur VIBN, keine Leistungstests an der darzustellenden Fertigungsanlage durchgeführt, sowie auch keine Überprüfung des Anlagenverhaltens in Bezug auf Störungen. Dies wiederum sind Inhalte der VIBN, wie auch die Analyse des Verhaltens der Fertigungsanlage bei Sonderabläufen wie Vollfahren und Leerfahren der Anlage, Ein- und Ausschleusen von Bauteilen aus dem Fertigungsprozess sowie das Fertigen unterschiedlicher Produktvarianten. Bei der Durchführung einer VIBN stellt sich bei nicht spezifiziertem Verhalten der virtuellen Produktionsanlage heraus, dass ein konstruktiver oder konzeptioneller Fehler in der Realanlage vorherrscht, welcher durch mangelnde mechanische, elektrotechnische oder informationstechnische Auslegung begründet ist. Vorausgesetzt wird hierbei ein korrekt arbeitendes Simulationsmodell.⁸⁸

3.2.4 Weitere Definitionen der Virtuellen Inbetriebnahme

Wie auch *Wunsch* definieren *Bracht/Geckler/Wenzel* in ihrem Sammelwerk die Virtuelle Inbetriebnahme als Absicherungsmethode zur Verifikation und Bestätigung von Anlagenverhalten unter Berücksichtigung von Störszenarien und Einfluss von Bedienereingaben. Dies erfolgt durch Zusammenspiel von Roboterprogrammen und Kopfsteuerung, welche mittels digitalem Anlagenmodell betrieben werden. Die zu steuernde Anlage wird durch ein virtuelles 3D-Modell dargestellt, das zugleich als Basis für Steuerungsdaten dient. Ebenso werden hier weitere Funktionen der Simulation abgebildet, wie zum Beispiel Kinematiken von Robotern. Die Datenübertragung wird durch einen Kommunikationsrechner übernommen, der die Verbindung zwischen realer Anlagensteuerung und Simulationsmodell darstellt.⁸⁹

Abschließend definiert *Kiefer* in seiner Dissertation die VIBN als das virtuelle Absichern realer Steuerungsabläufe im Zuge der Produktionsplanung, ohne dass reale Fertigungssysteme vorhanden sind. Ziel ist die Bewertung sowie Optimierung des gesamten Spektrums anlagentechnischen Ablaufverhaltens unter Gebrauch virtueller Ressourcen und Produktdaten sowie realer Prozessinformationen.⁹⁰

⁸⁸ Vgl. Wunsch (2007), S. 32 f.

⁸⁹ Vgl. Bracht/Geckler/Wenzel (2011), S. 303.

⁹⁰ Vgl. Kiefer (2007), S. 54.

3.3 Motivation zur Durchführung der VIBN

Nach eingehender Analyse des Begriffs der Virtuellen Inbetriebnahme befasst sich dieses Unterkapitel mit den Motivationen, eine VIBN durchzuführen.

Die unterschiedlichen Zielsetzungen und Motive zur Durchführung einer VIBN nach VDI 4499 Blatt 2 fasst Tab. 3 zusammen. Daraus geht hervor, dass die Entscheidung zur Verwirklichung einer VIBN zeitlich, qualitätstechnisch oder kostentechnisch motiviert respektive begründet sein kann.

| zeitliche Ziele | qualitätstechnische Ziele | kostentechnische Ziele |
|---|---|---|
| Parallelisierung der gesamten Planungs-, Entwicklungs-, und Anlaufphase zur gesamthaften Verkürzung der Projektlaufzeit | Absicherung von Investitionsentscheidungen durch schnelles Gegenüberstellen von Planungsszenarien | Kosteneinsparung durch vermiedene Änderungen, welche durch fehlerhafte Planung und Entwicklung ihren Ursprung haben |
| Vorverlagerung von Training und Schulungen | Reduzierung von Umsetzungs- und Planungsfehlern | durch Anzahlsenkung von Realexperimenten Kosteneinsparung bei der IBN |
| Abnahme von Teilbereichen oder Gesamtanlagen anhand des virtuellen Anlagenmodells vorverlagern | während des Anlaufs die Produktqualität steigern | Möglichkeit zur Automatisierung betriebsbegleitender Anpassungen |
| Ersetzen, Verändern und Optimieren von Prozessen und Systemen ohne Betriebsunterbrechung | Planungsfehler aufgrund vorab Testmöglichkeiten frühzeitiger identifizieren | Kosteneinsparung durch späteren Bau der Fertigungseinrichtung |
| Erhöhung der Flexibilität von Produktionsanlagen durch Anpassung von Umgebungsparametern, Anforderungen und Randbedingungen ohne Produktionsunterbrechung | Erhöhung der Transparenz von Planungs- und Fertigungsprozessen | |
| Verkürzung der Serienanlaufkurve | | |

Tab. 3: Motive zur Durchführung einer Virtuellen Inbetriebnahme, Quelle: VDI 4499 Blatt 2 (Mai 2011), S. 6 - 10 (modifiziert).

Ein weiteres Motiv zur Durchführung beziehungsweise Notwendigkeit einer VIBN begründet sich auf der Tatsache, dass 85 % der Schäden und Störungen, die bei der Durchführung einer Inbetriebnahme evident

werden, auf Mängel und Fehler aus Vorphasen der Projektumsetzung begründet sind.⁹¹

Ebenso stellt die Kostenbeeinflussung einen wesentlichen Aspekt in der Projektabwicklung dar. Die Möglichkeit diese nachhaltig günstig zu beeinflussen, ist in den Phasen der Konzeption sowie Konstruktion am größten. Fehler, die sehr früh in der Projektabwicklung entdeckt werden, lassen sich aufwandsarm und ohne höhere technische wie betriebswirtschaftliche Anstrengungen beheben. Dies bedeutet im Umkehrschluss, dass spät bemerkte Mängel sich nur unter großem technischen Aufwand beseitigen lassen, welche beträchtliche monetäre Einbußen nach sich ziehen, wie es Abb. 23 grafisch verdeutlicht.⁹²

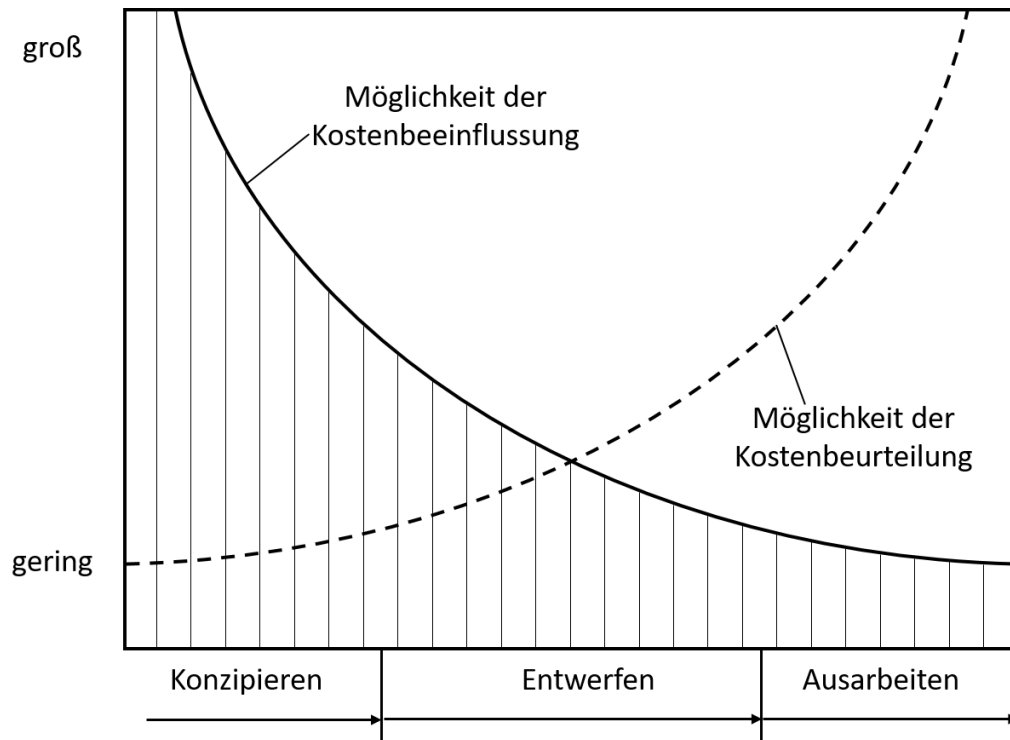


Abb. 23: Grad der Kostenbeeinflussung und Kostenbeurteilung in Abhängigkeit des Projektfortschritts, Quelle: VDI 2235 (Oktober 1987), S. 5 (leicht modifiziert).

Ein weiterer Grund, neue Methoden und Werkzeuge zur Inbetriebnahme von Karosseriebauanlagen in der Automobilindustrie zu entwickeln und anzuwenden, ist die notwendige Beschleunigung der Anlauf- und Planungsprozesse aufgrund des Anspruchs schneller Markeintritte sowie kürzer werdender Produktlebenszyklen.⁹³ Dem gegenüber steht die steigende Komplexität hochautomatisierter Produktionsanlagen aufgrund wachsender Variantenanzahl betreffend der zu fertigenden Produkte.⁹⁴ Erschwerend wirkt hierbei die Tatsache, dass Anlagenteile der Fertigungsanlagen von unterschiedlichen Herstellern beziehungsweise Lieferanten bezogen werden, die es steuerungstechnisch zu vernetzen gilt. Dies erhöht ergänzend die Notwendigkeit steuerungstechnischer Integrationstests.⁹⁵

⁹¹ Vgl. Weber (2006), S. 5 f.

⁹² Vgl. VDI 2235 (Oktober 1987), S. 5.

⁹³ Vgl. Zäh/Reinhart (Hrsg.) (2006), S. 4 (Kapitel 2).

⁹⁴ Wiendahl/ Hegenscheidt/ Winkler (2002), S. 650 - 655.

⁹⁵ Vgl. Brökelmann (2015), S. 5.

Diese Umstände führen zum Bedarf der Durchführung einer Virtuellen Inbetriebnahme, da diese Methode die Handhabe zur Parallelisierung der Planungs- und Inbetriebnahmephase ermöglicht. Diesem Nutzen steht jedoch ein zusätzlicher Aufwand gegenüber, dessen Ursache im erhöhten Modellbildungsaufwand begründet ist, wie es Abb. 24 hervorhebt. Dieser Zusatzaufwand wird in den nachfolgenden Kapiteln 5 und 7 einer eingehenden Analyse unterzogen.

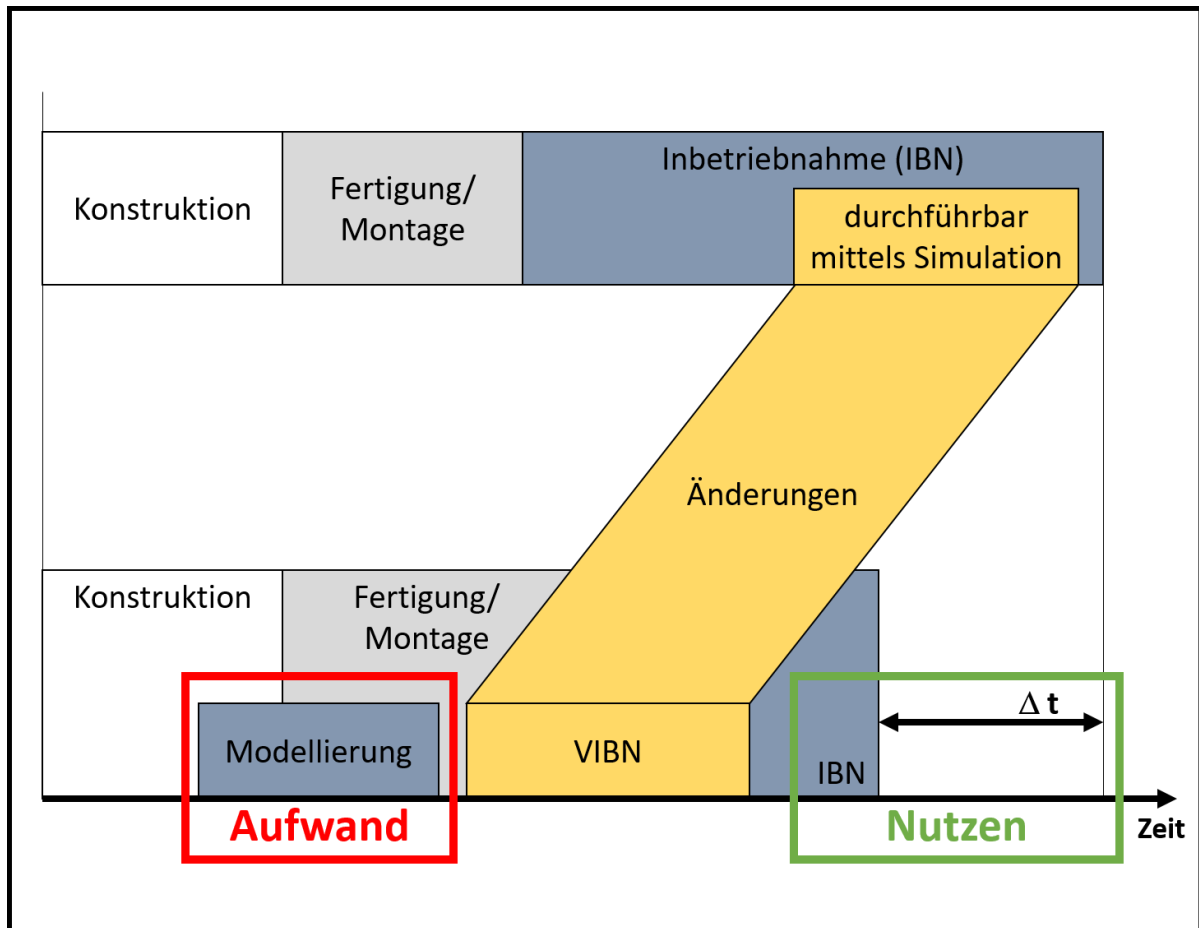


Abb. 24: Potential und Aufwand der Virtuellen Inbetriebnahme,
Quelle: Zäh/Reinhart (Hrsg.) (2006), S. 2 (Kapitel 1 / leicht modifiziert).

Nach Erhebung und Analyse der Motivlage zur Durchführung einer VIBN befasst sich der anschließende Abschnitt mit jenen Steuerungsebenen, welche zur Durchführung einer Virtuellen Inbetriebnahme geeignet sind.

3.4 Steuerungsebenen zur Durchführung einer VIBN

Inhalt von Unterkapitel 3.4 stellt die Herleitung und Definition jener Steuerungsebenen dar, die aufgrund ihrer Funktion, Wirkweise und Zielsetzung besondere Eignung zur Durchführung einer VIBN aufweisen.

Ausgangspunkt hierfür ist die Normenreihe DIN EN 62264, die ein hierarchisch aufgebautes Funktionsmodell in Bezug auf Fertigungssysteme standardisiert, welche rechnergestützt verwaltet, gesteuert, geregelt sowie überwacht werden.⁹⁶

⁹⁶ Vgl. DIN EN 62264-3 (Januar 2008), S. 12.

Das darin spezifizierte Funktionsmodell baut auf fünf Schichten beziehungsweise Strukturstufen auf die wie folgt definiert werden:⁹⁷

- Ebene 0 definiert den aktuellen physikalischen Prozess.
- Ebene 1 definiert alle Aktivitäten, die zur Erfassung und Manipulation der physikalischen Prozesse erforderlich sind. Auf dieser Ebene wird typischerweise mit Zeitintervallen von einer Sekunde und schneller operiert.
- Ebene 2 definiert die Aktivitäten zur Überwachung und Steuerung der physikalischen Prozesse. Diese Ebene arbeitet üblicherweise mit Zeitintervallen von Stunden, Minuten, Sekunden oder Teilssekunden.
- Ebene 3 definiert die Aktivitäten, Arbeitsabläufe und Prozessschritte, die zur Herstellung des gewünschten Endproduktes erforderlich sind. Diese Ebene beinhaltet ebenfalls sämtliche Aktivitäten, die zur Pflege und Wartung von Aufzeichnungen sowie zur Koordinierung der Prozesse erforderlich sind. Auf dieser Ebene finden typischerweise Zeitintervalle von Tagen, Schichten, Stunden, Minuten und Sekunden Anwendung.
- Ebene 4 definiert alle geschäftsbezogenen Aktivitäten, welche zur Verwaltung einer Fertigungsorganisation notwendig sind. Zu den produktionsbezogenen Aktivitäten gehört die Festlegung eines grundlegenden Anlagenplans wie zum Beispiel Materialeinsatz, Logistik, Lagerverwaltung und Disposition. Diese Ebene verwendet Zeitintervalle von Monaten, Wochen und Tagen. Des Weiteren ist für den fehlerfreien Betrieb der Ebene 4 eine stabile Informationsversorgung aus Ebene 3 Voraussetzung.

Diesen hierarchischen Strukturaufbau stellt Abb. 25 in grafischer Form ergänzend dar.

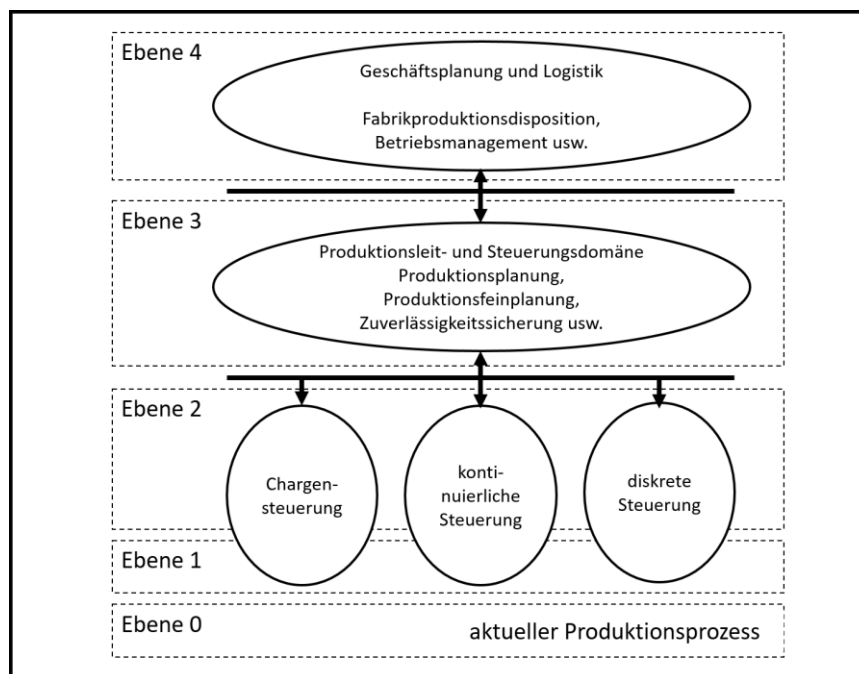


Abb. 25: Funktionales Hierarchiemodell von Steuerungen, Quelle: DIN EN 62264-1 (Juni 2008), S. 14 (modifiziert).

⁹⁷ Vgl. DIN EN 62264-3 (Januar 2008), S. 12 f.

Eine Konkretisierung des in Abb. 25 dargestellten Ebenenmodells erfolgt durch *Milberg*. Dieser gliedert die Ebenen der Informationsverarbeitung in Planungsebene, Leitebene, Zellenebene und Maschinenebene. Diese Unterteilung ist erforderlich, da dadurch komplexe und variable Aufgabenstellungen flexibler Produktionssysteme gezielt an die dafür geeignetste Steuerung übertragen werden können. Auch wird dadurch eine Systemkopplung geschaffen, welche die Verfügbarkeit der Informationsverarbeitung erhöht, sowie eine Überlastung im informationstechnischen Sinne vermeidet. Dies wird dadurch realisiert, dass jede Steuerungsebene nur mit der direkt angrenzenden im Informationsaustausch steht, wie es Abb. 26 verdeutlicht.⁹⁸

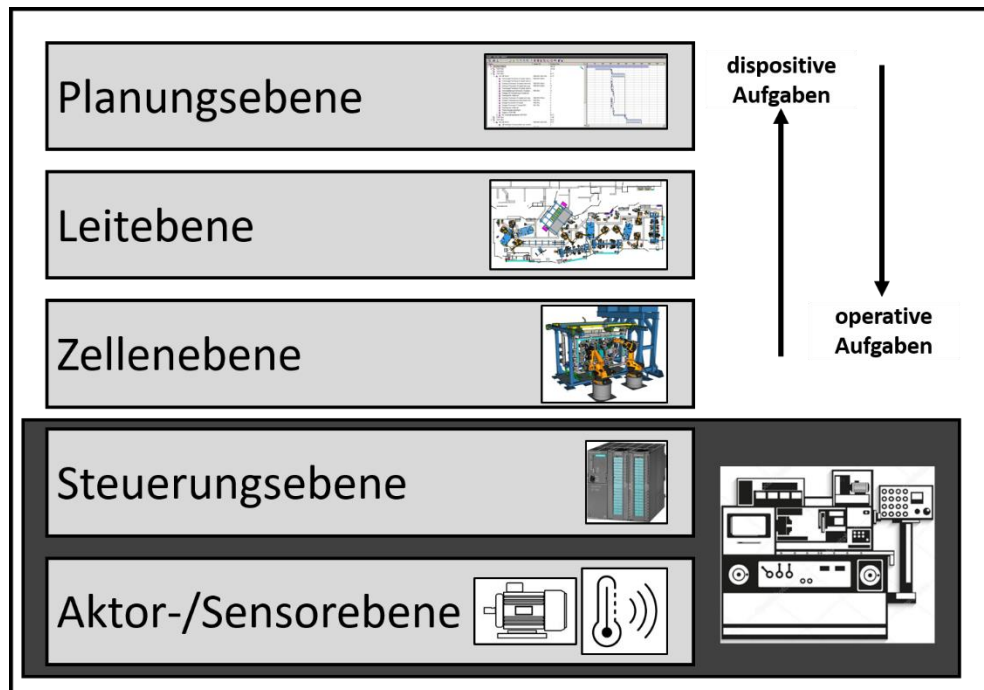


Abb. 26: Hierarchischer Aufbau informationstechnischer Verarbeitung in flexiblen Fertigungssystemen, Quelle: Milberg (Hrsg.) (1992), S. 128 (modifiziert).

Exakt die selbe Unterteilung bezüglich der Hierarchieebenen in der Informationsverarbeitung findet sich im Regelwerk VDI 4499 Blatt 2. Ergänzend werden hier die unterschiedlichen Ebenen der Steuerungshierarchie hinsichtlich ihres operativen Informationsflusses in den Kategorien Datenlänge, Ereignishäufigkeit und Zykluszeit bewertet.⁹⁹

Zur Durchführung einer Virtuellen Inbetriebnahme, welche die Untersuchung eines komplexen Fertigungssystems betreffend seiner Funktionalität als Zielsetzung verfolgt, ist eine skalierbare Auflösung der in Abb. 26 dargestellten Ebenen hinsichtlich ihrer Simulationstiefe erforderlich. Dies wird notwendig, damit Teilm Modelle mit hohem Detaillierungsgrad in anlagenübergreifende Simulationsmodelle eingebunden werden können, welche auf einem höheren Abstraktionsgrad operieren. Zu diesem Zweck erfolgt eine Umdefinierung der Leitebene in Anlagenebene, wie in Abb. 27 illustriert.¹⁰⁰

⁹⁸ Vgl. Milberg (Hrsg.) (1992), S. 127 f., zitiert nach: Duelen/Linnemann/Bernhardt (1986).

⁹⁹ Vgl. VDI 4499 Blatt 2 (Mai 2011), S. 18.

¹⁰⁰ Vgl. Wunsch (2007), S. 113., zitiert nach: Zäh/Reinhart (2004).

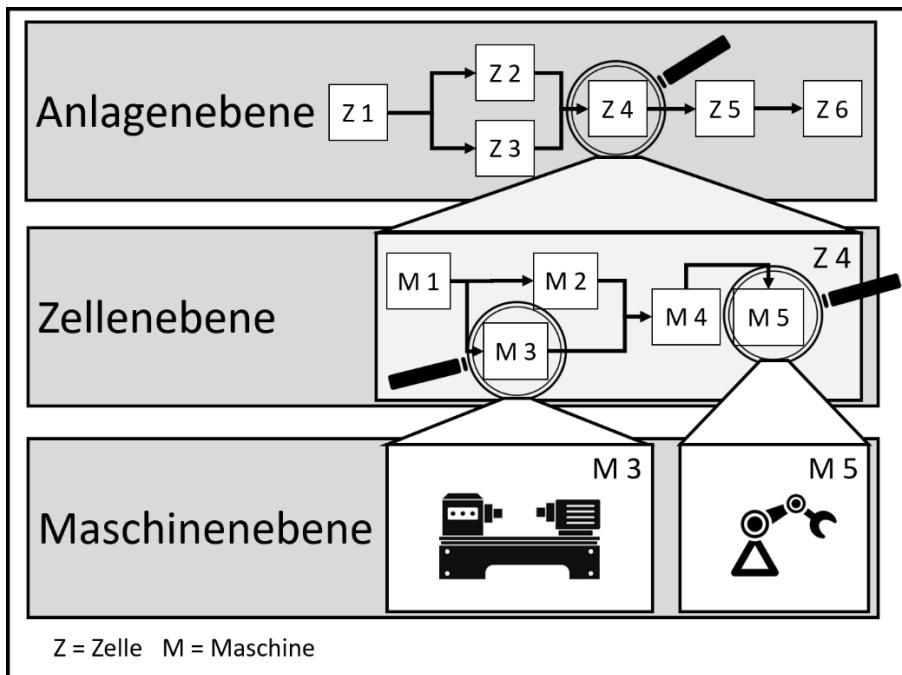


Abb. 27: Skalierbarkeit der Steuerungsebenen im Zuge der Erstellung von Simulationsmodellen, Quelle: Wunsch (2007), S. 113 (modifiziert).

Die dadurch geschaffenen Abstraktionsebenen eines Fertigungssystems ermöglichen die Strukturierung realer Automatisierungssysteme in verschiedene Modellierungsstufen. Aus diesen Abstraktionsebenen kann die Automatisierungspyramide eines vernetzten Automatisierungssystems abgeleitet werden. Aus dieser Automatisierungspyramide geht wiederum die Zuweisung unterschiedlicher Steuerungssysteme und Steuerungstechnologien hervor, welche der jeweiligen Abstraktionsebene die passende respektive geeignete Steuerungstechnologie zuweist, wie dies Abb. 28 verdeutlicht.¹⁰¹

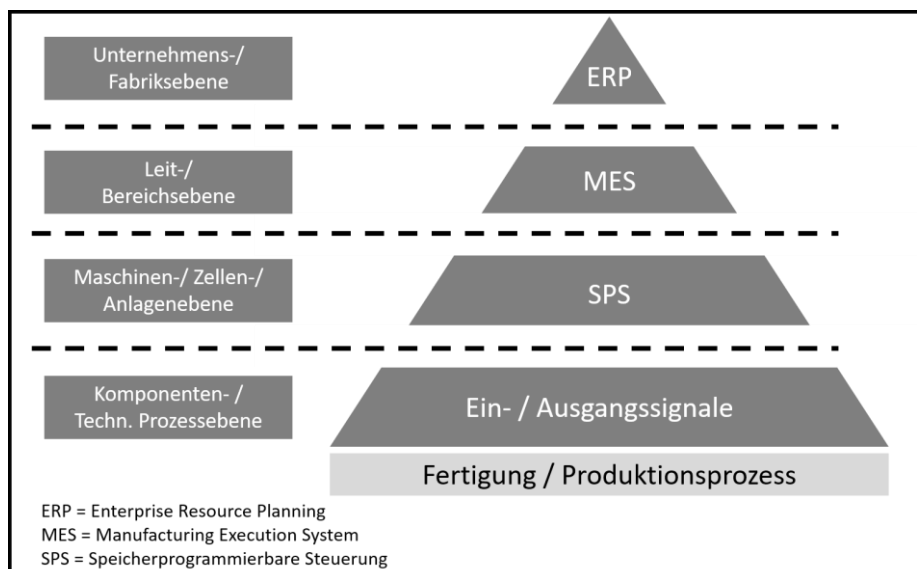


Abb. 28: Hierarchieebenen eines Automatisierungssystems sowie Zuweisung der passenden Steuerungstechnologie, Quelle: Lacour (2011), S. 12 (modifiziert).

¹⁰¹ Vgl. Lacour (2011), S. 12 f.

Aus Abb. 28 wird abgeleitet, dass für die Durchführung einer VIBN die Unternehmens- und Fabriksebene, die Leit- und Bereichsebene sowie Anlagen-, Zellen-, und Maschinenebene Eignung finden. Da die forschungsleitende Fragestellung dieser Arbeit die virtuelle Inbetriebnahme von Speicherprogrammierbaren Steuerungen zum Inhalt hat, wird in den nachfolgenden Kapiteln ausschließlich auf diese Steuerungstechnologie eingegangen. Ebenfalls erfolgt an dieser Stelle eine weitere Präzisierung hinsichtlich Abstraktionsebene des Automatisierungssystems beziehungsweise Fertigungssystems, welche im praktischen Teil dieser Masterthesis begründet ist. Diese Präzisierung erfolgt dahingehend, dass ab diesem Punkt der Arbeit ausschließlich die Abstraktionsebene der Zelle, wie in Abb. 27 dargestellt, einer weiteren wissenschaftlichen Analyse und Auseinandersetzung unterzogen wird.

Nach Definition der für die VIBN geeigneten Steuerungsebenen erfolgt im anschließenden Abschnitt die Aufarbeitung und Erörterung von Begrifflichkeiten, welche für das Verständnis zeitlicher Abläufe und Abarbeitungen innerhalb von Simulationssystemen und Steuerungen erforderlich sind.

3.5 Begriffsbestimmungen betreffend zeitlicher Abläufe in der Simulation

Dieses Unterkapitel setzt sich mit den Begrifflichkeiten zeittechnischer Abläufe innerhalb von Simulationssystemen und Steuerungen auseinander. Das Verständnis bezüglich dieser Begriffe ist Voraussetzung, damit in weiterer Folge die zeitliche Abarbeitung von Steuerungsabläufen nachvollzogen werden kann. Ebenso sind diese Begriffe notwendig, damit ein Verständnis hinsichtlich zeitlicher Interpretation sowie Darstellung und Wiedergabe von realen Steuerungsbefehlen innerhalb eines Simulationssystems geschaffen wird.

3.5.1 Echtzeit

Nach ISO/IEC 2382 wird Echtzeit als die Verarbeitung von Daten durch einen Computer in Verbindung mit einem anderen Prozess außerhalb des Computers gemäß den Zeitvorgaben des externen Prozesses definiert. Ebenfalls wird Begriff Echtzeit zur Beschreibung von Systemen verwendet, die im Konversationsmodus arbeiten wie auch für Prozesse, die durch menschliche Eingriffe während der Prozessabarbeitung beeinflusst werden.¹⁰²

Unter Echtzeitfähigkeit wird wiederum die *„Fähigkeit eines Prozessrechnensystems, die Rechenprozesse ständig derart ablaufbereit zu halten, dass sie innerhalb eines vorgegebenen Zeitintervalls auf Ereignisse im Ablauf eines technischen Prozesses reagieren können“*¹⁰³ verstanden.

Daraus lässt sich konkretisieren, dass Steuerungen zeitgebundene Rechenvorgänge, die durch Aufforderung externer Ereignisse abzuarbeiten sind, pünktlich auszuführen zu haben sowie geeignet darauf reagieren müssen. Der wichtigste Gesichtspunkt in diesem Zusammenhang ist die rechtzeitige Reaktion sowie Abarbeitung innerhalb vordefinierter Zeitgrenzen und nicht die reine Verarbeitungsgeschwindigkeit der Re-

¹⁰² Vgl. ISO/IEC 2382:2015(en) (2015).

¹⁰³ DIN IEC 60050-351 (Juli 2013).

chenoperationen. Charakteristisch für Echtzeitsysteme ist daher, dass vorliegende Verarbeitungsergebnisse nicht nur von der korrekt durchgeführten Rechenoperation abhängen, sondern auch vom Zeitpunkt der zur Verfügungstellung dieser Ergebnisse. Das entsprechende Zeitverhalten wird von äußeren Systemeinflüssen bestimmt, welche nicht gezwungen werden können, sich nach Verarbeitungsgeschwindigkeiten von Rechenoperationen und Rechenvorgängen zu richten.¹⁰⁴

In Abhängigkeit davon, wie starr die zur Abarbeitung der Rechenoperation vorgegeben Zeitgrenzen definiert sind sowie wie sich das Schadensausmaß bei Nichteinhaltung dieser zeitlichen Grenzen äußert, wird von *weicher Echtzeit* und *harter Echtzeit* gesprochen.¹⁰⁵ Das zeitliche Verhalten dieser beiden Formen der Echtzeit wird in Abb. 29 verdeutlicht.

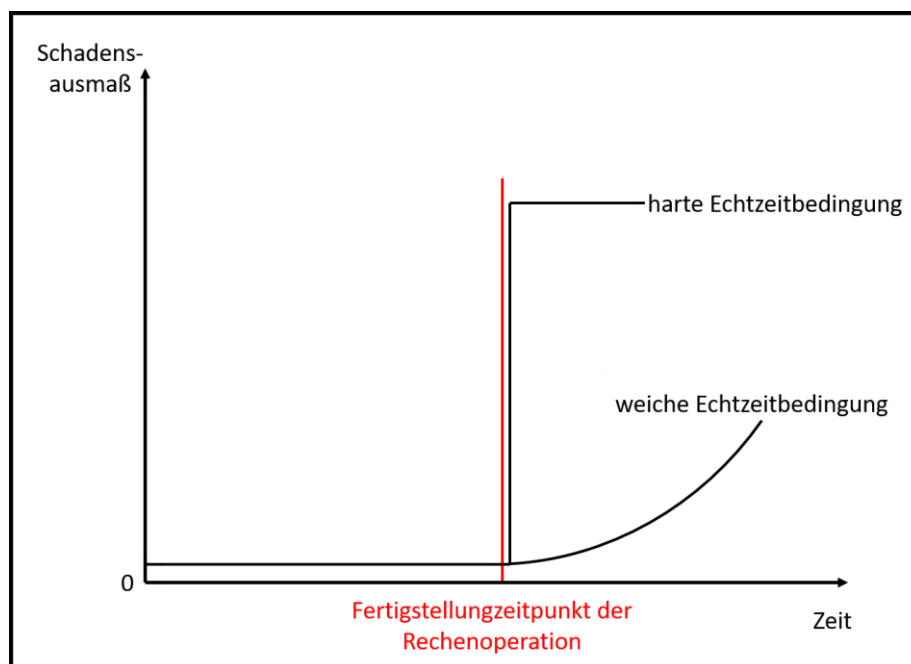


Abb. 29: Zeitliches Verhalten der unterschiedlichen Echtzeitbedingungen, Quelle: Halang/Konakovsky (2018), S. 21 (modifiziert).

Bei Systemen, die der Anforderung einer harten Echtzeit unterliegen, zieht die Nichteinhaltung einer zeitlichen Grenze einer Rechenoperation fatale Folgen nach sich. Unterliegen Systeme dieser Anforderung, so muss die Wahrscheinlichkeit gleich 100 % betragen, dass jede ausgelöste Rechenoperation zum zugesicherten Fertigstellungstermin abgeschlossen wird, und das Ergebnis der weiteren Verarbeitung zugeführt werden kann. Besteht die Wahrscheinlichkeit, dass nur gelegentlich die Einhaltung dieser Vorgabe beziehungsweise Charakteristik nicht eingehalten werden kann, so wird dies als weiche Echtzeit klassifiziert. Voraussetzung hierfür ist, dass trotz Nichteinhaltung der Bearbeitungsfrist die Funktion des Systems nicht geschädigt wird, oder die Auswirkungen des verspäteten Bearbeitungsendes toleriert werden können.¹⁰⁶ Tab. 4 fasst die Unterscheidungen bezüglich der beiden Formen der Echtzeit zusammen.

¹⁰⁴ Vgl. Halang/Konakovsky (2018), S. 20.

¹⁰⁵ Vgl. Halang/Konakovsky (2018), S. 21.

¹⁰⁶ Vgl. Scholz (2005), S. 40.

| Echtzeitbedingung | harte Echtzeit | weiche Echtzeit |
|---|--|---|
| Folgen bei nicht rechtzeitiger Reaktion | <ul style="list-style-type: none"> • nutzloses Ergebnis • Gefahr für Leib und Leben von Menschen • Gefahr fehlerhafter Ergebnisse von Prozessen | <ul style="list-style-type: none"> • steigende Kosten in Abhängigkeit der verursachten Verzögerungszeit • Güter minderer Qualität |
| Beispiele | <ul style="list-style-type: none"> • Autofahrer, der sich einer Kreuzung nähert • Temperaturüberwachung von Atomreaktoren | <ul style="list-style-type: none"> • Flugreservierung • Kontoführung • Zugverspätung |

Tab. 4: Gegenüberstellung der Auswirkungen bei Nichteinhaltung von unterschiedlichem Echtzeitverhalten, Quelle: Halang/Konakovsky (2018), S. 21 (modifiziert).

3.5.2 Zeitachse zwischen Real- und Simulationssystem

Wird zur Entwicklung und Testung realer Steuerungssoftware oder realer Steuerungen ein Simulationssystem eingebunden, tritt ständig und regelmäßig eine Wechselwirkung zwischen Steuerung und Simulation auf, welche zu einer Verzerrung und Verfälschung der Simulationserkenntnisse führen kann. Dies begründet sich in dem Umstand, dass in realen Steuerungen die Verarbeitungsgeschwindigkeit der Rechenoperationen einer systemtechnischen Begrenzung unterliegt. Wird in einem Realsystem der Abschluss einer Aktion an die betroffene reale Steuerung zurückgemeldet, benötigt diese eine bestimmte Zeit zur Abarbeitung dieser Meldung. Somit vergeht bis zum Eintreffen des nächsten abzuarbeitenden Befehls eine nicht vorhersehbare Zeitspanne, die von den eingesetzten Übertragungsmedien sowie der Leistungsfähigkeit der Steuerung abhängt. Wird das Realsystem durch eine Simulationsumgebung substituiert, tritt das Verhalten auf, dass das Zeitsystem in der Simulation nicht an die Realzeit gekoppelt ist. Da das abzubildende Fertigungssystem am Simulationsrechner dargestellt wird, werden durch den Simulationsrechner so viele diskrete Ereignisse abgearbeitet, wie es seine Rechnerleistung zulässt. Dadurch kann die Situation eintreten, dass vom Melden einer bestimmten Aktion durch das Simulationsmodell an die Realsteuerung die Simulation zeitlich wesentlich weiter fortschreitet, als das Prozessabbild der Realsteuerung. Es muss daher eine Synchronisierung zwischen Realzeit und Simulationszeit erfolgen.¹⁰⁷

Ebenso ist der Umstand zu berücksichtigen, dass der tatsächliche Zeitverbrauch einer Simulation diskreter Ereignisse von der Ereignisdichte der aufgetretenen Ereignisse abhängt. Dies begründet sich dadurch, dass in zeitkontinuierlichen Realprozessen zwei beliebige Ereigniszeitpunkte betreffend ihres Abstandes zueinander sehr unterschiedlich sein können, der selbe Prozess jedoch im Simulationssystem ohne Rücksichtnahme dieses Zeitabstandes von einem Ereigniszeitpunkt zum nächsten springt. Daraus ergibt sich ein nicht konstantes Verhältnis zwischen Realzeit und Simulationszeit. Um diese negative Beeinträchtigung zu kompensieren beziehungsweise vermeiden zu können, ist es erforderlich, dass das Simulationssystem

¹⁰⁷ Vgl. Amann (1994), S. 61 ff.

respektive die Testumgebung die Eigenschaft der Echtzeitfähigkeit aufweist, sowie mit der Realzeit synchronisiert werden kann. Dadurch verfügt das Simulationssystem über die Kompetenz, erst dann das nächste zeitdiskrete Ereignis abzuarbeiten, wenn im Realsystem die entsprechende Zeit vergangen ist. Dies ist deshalb von existentieller Relevanz, da das Simulationssystem das nächste Ereignis im Simulationsmodell berechnen kann, jedoch nicht jenes externe Ereignis, welches von der Realsteuerung exekutiert wird, wie beispielsweise die Anforderung eines Ausschleusvorgangs von Bauteilen durch den Anlagenbediener. Des Weiteren wird durch die Möglichkeit von Simulationsabläufen im Zeitraffermodus der praktische Nutzen eines Simulationssystems gesteigert, da dadurch Realabläufe, welche mehrere Sekunden oder Minuten dauern, in zeitlich verkürzten Sequenzen in der Simulation abgebildet werden können. Als Beispiele hierfür kann das Überspringen von Wartezeiten oder Bearbeitungszeiten von Betriebsmitteln angeführt werden, welche nicht im aktuellen Fokus des Untersuchungsinteresses liegen.¹⁰⁸

Damit dies technisch ermöglicht sowie stabil angewandt werden kann, ist die Gewährleistung und Einhaltung des dafür erforderlichen Abtasttheorems Voraussetzung, das im anschließenden Abschnitt erörtert wird.

3.5.3 Abtasttheorem

Unter *Abtasten* wird die Bewirkung des Übergangs eines analogen Signals in ein digitales Signal verstanden.¹⁰⁹ Damit die Rekonstruktion dieses übergeführten Signals ohne Informationsverlust gewährleistet werden kann, muss folgende Gesetzmäßigkeit eingehalten werden, die als Abtasttheorem bezeichnet wird:¹¹⁰

$$f_a > 2f_{max} \quad (3.1) \quad \begin{array}{ll} f_a/\text{Hz} & \text{Abtastfrequenz} \\ f_{max}/\text{Hz} & \text{maximale Spektralfrequenz des Signals} \end{array}$$

Sofern die maximale Spektralfrequenz des zu rekonstruierenden Signals kleiner als die halbe Abtastfrequenz ist, kann das Signal vollständig zurückgewonnen werden.¹¹¹

Zur Beurteilung binärer Steuerungsreaktionen beträgt die maximal gesuchte Spektralfrequenz die halbe Frequenz des Steuerungssignals. Wird die Anforderung des Abtasttheorems mit der Anforderung der Abtastung binärer Signale kombiniert, so ergibt sich die Erfordernis einer Abtastfrequenz, die mindestens die selbe Frequenz aufweist wie das binäre Steuersignal. Wird diese Anforderung eingehalten, so kann das binäre Steuersignal ohne Informationsverlust abgetastet und rekonstruiert werden, wie es Abb. 30 grafisch verdeutlicht.¹¹²

¹⁰⁸ Vgl. Strassacker (1997), S. 32 f.

¹⁰⁹ Vgl. Beucher (2015), S. 215.

¹¹⁰ Vgl. Beucher (2015), S. 228.

¹¹¹ Vgl. Beucher (2015), S. 229.

¹¹² Vgl. Wünsch (2007), S. 25.

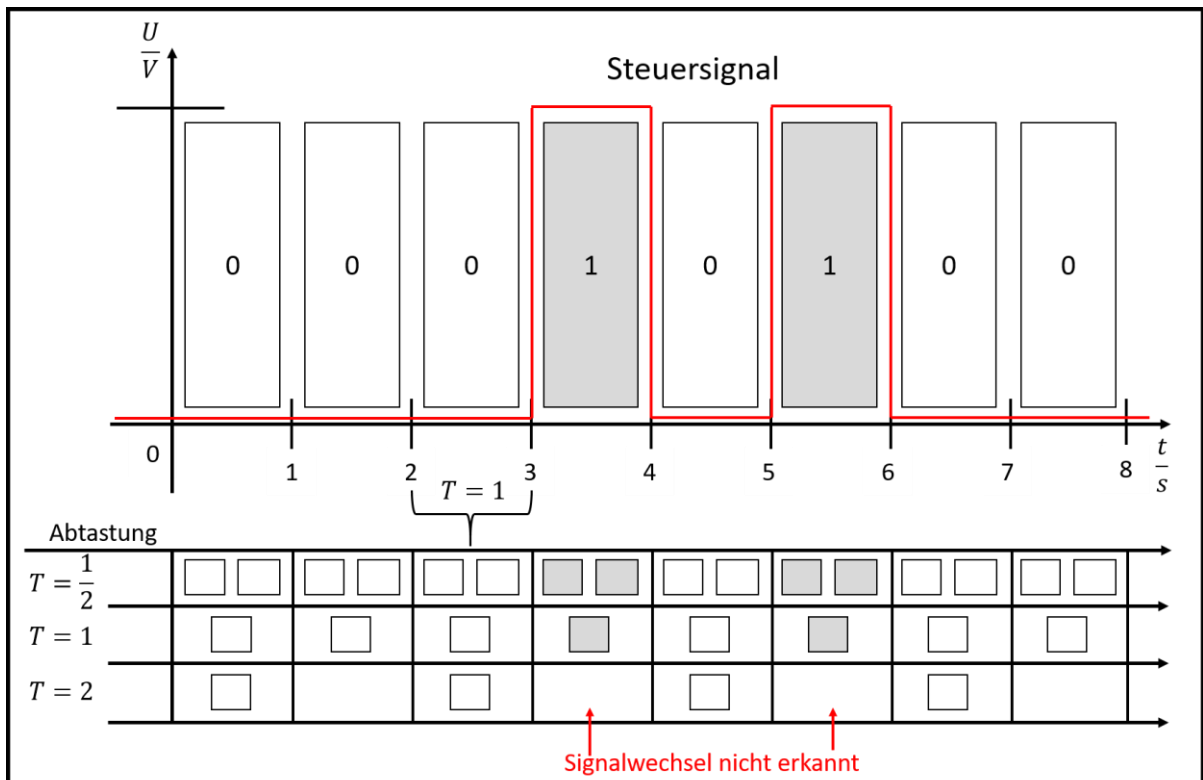


Abb. 30: Auswirkung unterschiedlicher Abtastfrequenzen hinsichtlich ihres Informationserhalts, Quelle: Wunsch (2007), S. 26 (modifiziert).

Nach Erörterung der in diesem Abschnitt behandelten Begrifflichkeiten betreffend Anforderungen an das zeitliche Verhalten von Simulationssystemen und deren Auswirkungen befasst sich der nachfolgende Abschnitt mit den unterschiedlichen Simulationismethoden einer Virtuellen Inbetriebnahme.

3.6 Methoden der Virtuellen Inbetriebnahme

Unterkapitel 3.6 setzt sich mit den unterschiedlichen Methoden und Herangehensweisen betreffend Modellaufbau und Simulationismöglichkeiten auseinander, die bei der Durchführung einer Virtuellen Inbetriebnahme zur Anwendung gelangen können. Aufgrund der erarbeitenden Erkenntnisse aus Abschnitt 3.4 beziehen sich die weiterführenden Analysen und Erörterungen auf die Steuerungsebene einer Fertigungszelle, welche mittels einer SPS gelenkt, gesteuert und geregelt wird. Tab. 5 liefert hierbei einen Überblick der Anwendungsmethoden zur Durchführung einer VIBN auf dieser Abstraktionsebene eines Automatisierungssystems (AS), die weiterführend einer vertiefenden Auseinandersetzung unterzogen werden.

| | | Produktionssystem/Produktionsanlage | |
|-----------|----------|---|----------------------------|
| | | virtuell | real |
| Steuerung | virtuell | Systemsimulation Model in the Loop (MiL) | Software in the Loop (SiL) |
| | real | Hardware in the Loop (HiL) | realer Betrieb |

Tab. 5: Methoden der VIBN einer automatisierter Produktionsanlagen, Quelle: VDI 4499 Blatt 2 (Mai 2011), S. 38 (modifiziert).

Der generische Aufbau eines solchen Automatisierungssystems respektive Automatisierungstechnik-Konfiguration (AT) wird in Abb. 31 in grafischer Form hervorgehoben und verdeutlicht. Dieser generische Aufbau stellt in weiterer Folge jene Basis und jenen Ausgangspunkt dar, von welchem die in Tab. 5 angeführten Methoden abgeleitet werden.

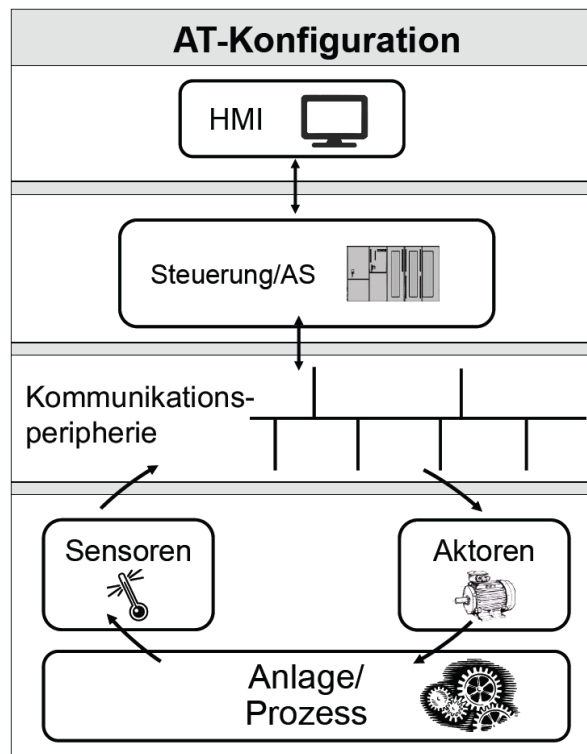


Abb. 31: Generischer Aufbau eines Automatisierungssystems auf Zellebene,
Quelle: VDI/VDE 3693 Blatt 1 (August 2016), S. 7.

Diese definiert sich durch die Bedienerschnittstelle beziehungsweise Human Machine Interface auf oberster Ebene. In der darunter angrenzenden Ebene befindet sich die SPS, welche als Steuerung des Automatisierungssystems fungiert sowie als Verbindungsglied zwischen HMI und Kommunikationsperipherie. Die Kommunikationsperipherie verbindet die Steuerung mit den Aktoren und Sensoren des Automatisierungssystems, welche Prozessinformationen aus der Produktionsanlage auslesen sowie Steuerungsanweisungen in dieser ausführen.¹¹³

3.6.1 Systemsimulation / Model in the Loop (MiL)

Als Systemsimulation wird jene Methode der Virtuellen Inbetriebnahme verstanden, bei der alle erforderlichen und beteiligten Komponenten in einer Simulationsumgebung getestet und erprobt werden, wobei diese Erprobung gesamthaft auf einem Entwicklungsrechner ausgeführt wird. Mittels dieser Methode können sehr schnell Realisierungsmöglichkeiten getestet sowie Ergebnisse erzielt werden. Zeitverhalten von simuliertem Prozess und simulierter Steuerung müssen aufeinander abgestimmt sein.¹¹⁴

¹¹³ Vgl. VDI/VDE 3693 Blatt 1 (August 2016), S. 7 f.

¹¹⁴ Vgl. VDI 4499 Blatt 2 (Mai 2011), S. 38 f.

Die Entwicklungsrechner müssen nicht mit der Zielhardware ident sein. Model-in-the-Loop stellt einen Spezialfall der Systemsimulation dar.¹¹⁵

Bei entwicklungsbegleitenden Tests von Steuerungsalgorithmen im Zuge prototypischer Implementierung findet diese Art der VIBN ebenfalls Anwendung. Implementiert werden diese Steuerungsalgorithmen meist nicht in klassischen Steuerungssprachen, wie sie zum Beispiel durch DIN EN 61131-3 spezifiziert werden, sondern durch Modellsprachen im Zusammenwirken mit Simulationswerkzeugen, welche eine grafische Orientierung zum Schwerpunkt haben. Simulierte Anlage sowie Steuerung werden im selben Simulationswerkzeug entwickelt und modelliert, wobei oftmals einfache Anlagenmodelle für den Untersuchungs- und Entwicklungszweck ausreichend sind. Abb. 32 stellt diese Methode der VIBN zur Verdeutlichung grafisch dar.¹¹⁶

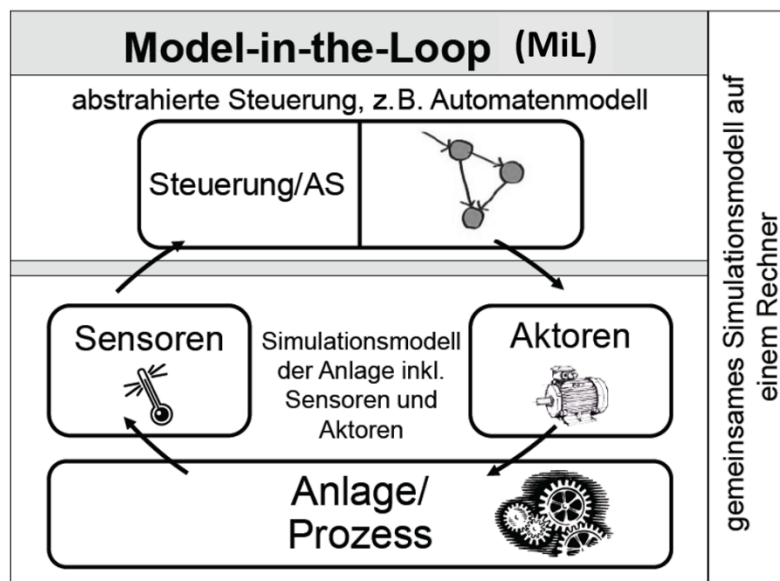


Abb. 32: Klassische Systemkonfiguration eines VIBN-Ansatzes mittels MiL, Quelle: VDI/VDE 3693 Blatt 1 (August 2016), S. 8 (leicht modifiziert).

3.6.2 Software in the Loop (SiL)

Bei der Anwendung des SiL-Ansatzes im Zuge einer Virtuellen Inbetriebnahme werden entwickelte Regelungs- und Steuerungsalgorithmen auf einem Entwicklungsrechner ausgeführt, wobei diese Algorithmen in einer Simulationsumgebung abgebildet werden. Die Simulationsumgebung in der die Steuerungsalgorithmen in Abhängigkeit ihrer Anforderung in weicher oder harter Echtzeit ausgeführt werden, ist mit der Realanlage verbunden. Der Fokus bei dieser Art der Herangehensweise liegt bei der Entwicklung von steuerungs- und regelungstechnischen Algorithmen, ohne das bereits erforderliche Hardware zur Abbildung der Steuerung vorhanden sein muss.¹¹⁷

¹¹⁵ Vgl. Abel/Bollig (2006), S. 9.

¹¹⁶ Vgl. VDI/VDE 3693 Blatt 1 (August 2016), S. 8.

¹¹⁷ Vgl. VDI 4499 Blatt 2 (Mai 2011), S. 39.

Auch *Abel/Bollig* folgen dieser Definition von Software-in-the-Loop. Sie weisen diesem Entwicklungsansatz respektive Entwicklungsmethodik die grundlegende Charaktereigenschaft zu, dass Steuerungs- und Regelungsalgorithmen am Entwicklungsrechner ausgeführt werden, welche mit Realprozessen oder Teilen davon in Verbindung stehen.¹¹⁸ Dieses Vorgehen kann bis zur vollständigen Entwicklung des endgültigen Programmcodes angewandt werden.¹¹⁹

Abweichend zu den beiden bereits angeführten Definitionen von SiL spezifiziert VDI 3693 Blatt 1 Software-in-the-Loop als weitere Detaillierung des MiL-Ansatzes. Der Unterschied zu Model-in-the-Loop besteht darin, dass der auszuführende Steuerungscode der Konfigurationssprache des Automatisierungssystems entsprechen muss, was im Falle einer abzubildenden SPS die Einhaltung der Sprachkonventionen nach DIN EN 61131-3 bedeutet. Ebenso muss der Steuerungscode auf einer Emulation der Realsteuerung ausgeführt werden, was einer gänzlichen Virtualisierung derselben entspricht. Auch hinsichtlich Abbildung des Produktionsprozesses beziehungsweise der Produktionsanlage weicht dieses Regelwerk von den vorhergehenden Definitionen ab, indem diese ebenfalls am Simulationsrechner abgebildet werden und daher die Charakteristika der Systemsimulation ausweisen.¹²⁰

Da betreffend der Spezifikation von Software-in-the-Loop kein Konsens in der Fachwelt vorherrscht, ist dahingehend auch keine eindeutige Definition in grafischer Form möglich. Abb. 33 stellt daher beide Auslegungen von SiL in Bildform dar.

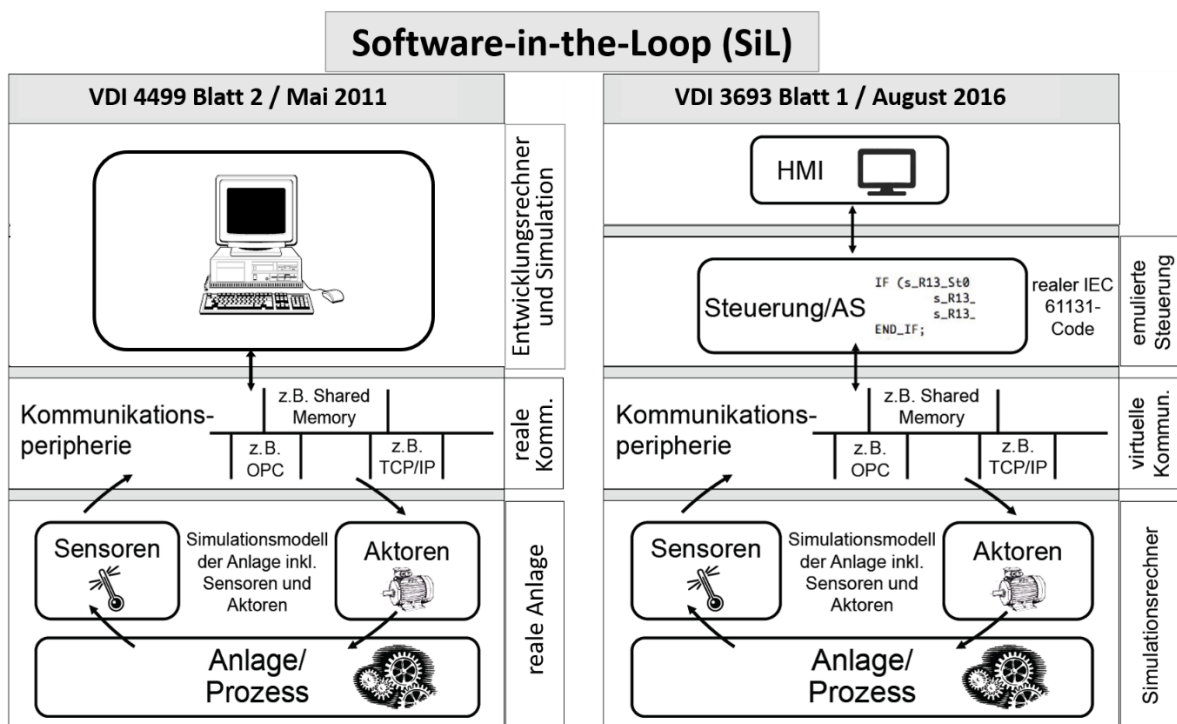


Abb. 33: Gegenüberstellung der unterschiedlichen Definitionen von Software-in-the-Loop in Anlehnung an VDI 4499 und VDI 3693, Quelle: Eigene Darstellung.

¹¹⁸ Vgl. Abel/Bollig (2006), S. 9.

¹¹⁹ Vgl. Glöckler (2018), S. 244 f.

¹²⁰ Vgl. VDI/VDE 3693 Blatt 1 (August 2016), S. 8 f.

3.6.3 Hardware in the Loop (HiL)

In Hinblick auf Sicherheit, Robustheit und Funktionsfähigkeit werden Algorithmen unter Anwendung der Hardware-in-the-Loop Methodik am virtuell abgebildeten Anlagen- und Produktionsmodell getestet und geprüft. Umfassende Tests und kritische Szenarien können ohne Eingriff und Beeinträchtigung des Realsystems durchgeführt werden. Ebenso ermöglicht diese Methodik das Abprüfen von Verhalten der realen Steuerungshardware gegenüber äußerer Störeinflüsse wie beispielsweise Erschütterungen, Hitze, Kälte oder Feuchtigkeit.¹²¹

Weiters spricht man von HiL, wenn unter Ausnutzung eines Simulationsmodells der reale Prozess oder die reale Fertigungsanlage abgebildet und mittels dieser Steuerungs- und Regelungsalgorithmen verifiziert und validiert werden, welche auf der jeweiligen Zielhardware implementiert sind.¹²² Sowohl die reale Steuerung als auch das mittels Simulation abgebildete Prozessmodell sind unter Echtzeitbedingung miteinander verbunden.¹²³ In regelungstechnischen Anwendungen wird die Regelstrecke in Echtzeit simuliert und die dafür vorgesehenen Algorithmen zur Steuerung und Regelung auf der betreffenden Realhardware ausgeführt.¹²⁴

Im Gegensatz zu SiL führen die VDI-Regelwerke 4499 Blatt 2 und 3693 Blatt 1 keine widersprüchlichen Definitionen von Hardware-in-the-Loop an. Dahingehend stellt Abb. 34 die Methodik von HiL eindeutig dar.

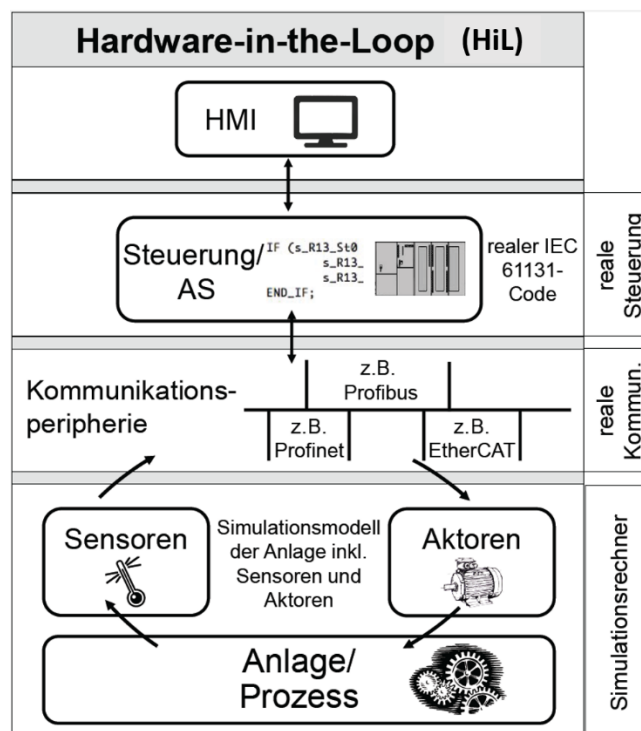


Abb. 34: Methode der VIBN mittels Hardware-in-the-Loop, Quelle: VDI/VDE 3693 Blatt 1 (August 2016), S. 10 (leicht modifiziert).

¹²¹ Vgl. VDI 4499 Blatt 2 (Mai 2011), S. 39.

¹²² Vgl. Abel/Bollig (2006), S. 9.

¹²³ Vgl. Abel/Bollig (2006), S. 304.

¹²⁴ Vgl. Glöckler (2018), S. 244.

Wie auch in VDI 4499 Blatt 2 definiert VDI 3693 Blatt 1 HiL als Methode zur Virtuellen Inbetriebnahme, bei der auf einer Realsteuerung der reale Steuerungscode des Zielsystems ausgeführt und anhand eines virtuellen Anlagenmodells getestet wird. Überwiegend wird diese Methode für Gesamt- und Integrationstests von Automatisierungssystemen mit der Zielsetzung angewandt, das Zusammenspiel der im Projekt eingesetzten Steuerungskomponenten abzusichern.¹²⁵

3.7 Modelle in der VIBN

In Abhängigkeit des Industriesektors erfordert die VIBN unterschiedliche Modellierungen von Anlagentypen. Dahingehend unterscheiden sich die eingesetzten Modellarten hinsichtlich ihrer Modellierungstiefe sowie gegebenenfalls bestehender Relationen zueinander.¹²⁶

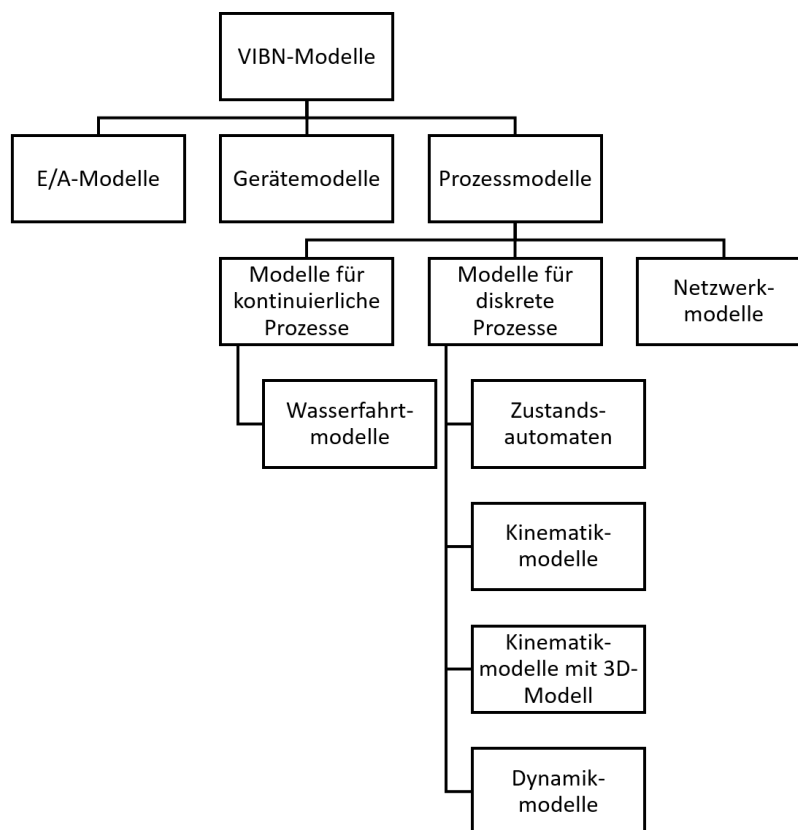


Abb. 35: Übersicht zur Anwendung kommender Modelle im Zuge einer VIBN, Quelle: VDI/VDE 3693 Blatt 1 (August 2016), S. 16.

Abb. 45 bietet diesbezüglich einen Überblick der zum Einsatz gelangenden Modellarten im Zuge einer Virtuellen Inbetriebnahme.

Aufgrund des beschränkten Umfangs dieser Arbeit kann an dieser Stelle nicht im Detail auf die angeführten Modellarten eingegangen werden. Jene Modellarten, welche besondere Relevanz zur Umsetzung des praktischen Teils der Masterthesis aufweisen, werden an ihrer jeweiligen Bedarfsstelle einer vertiefenden Auseinandersetzung unterzogen. Abschließend erfolgt die Zusammenfassung der erarbeiteten Erkenntnisse und Schlussfolgerungen, die aus Kapitel 3 abgeleitet werden können.

¹²⁵ Vgl. VDI/VDE 3693 Blatt 1 (August 2016), S. 9.

¹²⁶ Vgl. VDI/VDE 3693 Blatt 1 (August 2016), S. 16.

3.8 Zusammenfassung

Abschließend fasst Unterkapitel 3.8 die gewonnenen Erkenntnisse und Schlussfolgerungen, welche aus der Untersuchung und Analyse im Zuge der theoretischen Auseinandersetzung im Themenfeld der Virtuellen Inbetriebnahme gewonnen werden konnten, zusammen.

Am Beginn dieser Auseinandersetzung stand die Herleitung der Begriffe *Inbetriebnahme* sowie dessen Überleitung in die Methodik der *Virtuellen Inbetriebnahme*. Es konnte festgestellt werden, dass unter Inbetriebnahme unter anderem das Auslösen zur Erfüllung einer Funktion¹²⁷ verstanden wird, sowie die Bereitstellung von Arbeitsmitteln technischer Natur zur betrieblichen Nutzung¹²⁸. Die Überleitung vom Begriff der Inbetriebnahme zur Virtuellen Inbetriebnahme wurde durch das VDI Regelwerk 4499 Blatt 1 im Rahmen der Definition und Konkretisierung der Teilbereiche des digitalen Fabrikbetriebs hergestellt.¹²⁹ Darauf folgend wurde die Bedeutung und Zielsetzung dieser Methode eingehend analysiert, wobei diagnostiziert werden konnte, dass sich die VIBN zur Simulation dadurch unterscheidet, dass die Virtuelle Inbetriebnahme immer reale Teile respektive Teilsegmente im Modellbildungsprozess des Produktionssystems implementiert und einbindet¹³⁰. Eine umfassende Begriffsbestimmung und Definition der Virtuellen Inbetriebnahme wurde in Abschnitt 3.2 durchgeführt. Ebenso wurden die Motive zur Durchführung einer VIBN erörtert und festgestellt, dass die Beweggründe zur Durchführung zeitlich, qualitätstechnisch sowie kostentechnisch motiviert sein können¹³¹. Auch die Fehlervermeidung im Entstehungsprozess von Automatisierungssystemen sowie die Mängelaufdeckung aus vorangegangenen Entwicklungsphasen wurden als Auslöser zur Anwendung der VIBN identifiziert.¹³²

Im Anschluss erfolgte die Untersuchung jener Steuerungsebenen, die in der VIBN Anwendung finden. Im Zuge dessen wurde festgestellt, dass die Anlagenebene, die Zellenebene sowie die Maschinenebene zur Durchführung einer VIBN geeignet sind¹³³. Nachdem alle drei genannten Ebenen durch Einsatz einer Speicherprogrammierbaren Steuerung gelenkt, gesteuert und geregelt werden können, wurde eine Konkretisierung hinsichtlich der Zellenebene vorgenommen. Diese begründet sich dadurch, dass der praktische Teil dieser Arbeit die Zellenebene in den Mittelpunkt der forschungsleitenden Fragestellung rückt.

Danach wurden die zeitlichen Verhaltensweisen der Echtzeit, des Abtasttheorems sowie der Zeitachse zwischen Real- und Simulationssystem untersucht. Es wurde erkannt, dass eine Synchronisierung der

¹²⁷ Vgl. DIN 31051 (September 2012), S. 10.

¹²⁸ Vgl. DIN 32541 (Mai 1977), S. 1.

¹²⁹ Vgl. VDI 4499 Blatt 2 (Mai 2011), S. 3.

¹³⁰ Vgl. VDI 4499 Blatt 2 (Mai 2011), S. 11 f.

¹³¹ Vgl. VDI 4499 Blatt 2 (Mai 2011), S. 6 - 10.

¹³² Vgl. Weber (2006), S. 5 f.

¹³³ Vgl. Wünsch (2007), S. 113.

Zeitsysteme zwischen Realsystem und Simulationssystem notwendig ist, sowie das Wissen über die eingesetzten und angewandten Zeiteinheiten.¹³⁴ Ebenso ist die Fähigkeit und Kompetenz eines VIBN-Systems zur Darstellung eines Simulationsvorganges im Zeitraffer sowie in Slow Motion von Relevanz, damit die Effizienz der Methode der VIBN gesteigert werden kann.¹³⁵

In Unterkapitel 3.6 wurden die unterschiedlichen Methoden zur Durchführung einer Virtuellen Inbetriebnahme eingehend behandelt. Hier wurde ermittelt, dass ausschließlich bei der Methode Hardware-in-the-Loop Konsens hinsichtlich der wissenschaftlichen Definition vorliegt. Bei den Methoden der Systemsimulation, Model-in-the-Loop sowie Software-in-the-Loop bestehen unterschiedliche Definitionen durch die Ingenieurwissenschaften, wobei im Zuge dieser Arbeit der Definition nach VDI 4499 Blatt 2 gefolgt wird. Abb. 36 stellt diese Definition in grafischer Form zur Verdeutlichung dar, wobei in dieser auch der Zusammenhang zwischen Modellbildung und Codegenerierung Berücksichtigung findet.

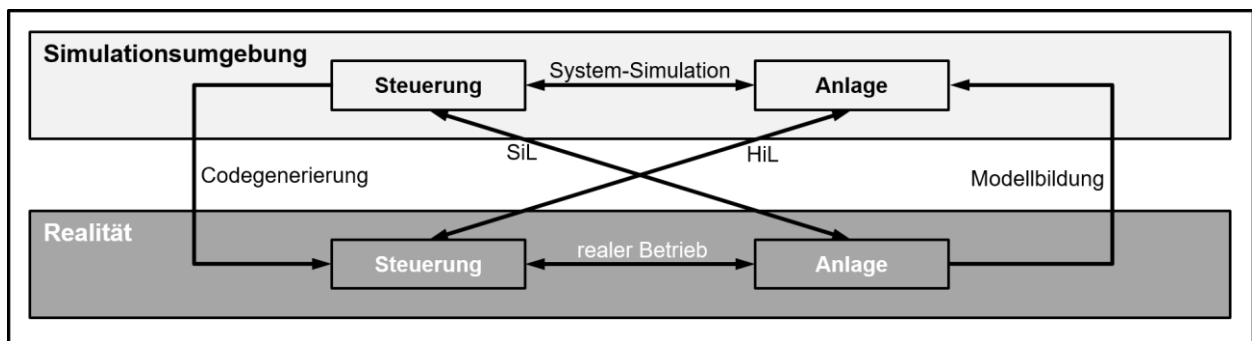


Abb. 36: Zusammenhang unterschiedlicher VIBN-Methoden in Bezug auf Modellbildung und Codegenerierung, Quelle: VDI 4499 Blatt 2 (Mai 2011), S. 39 (leicht modifiziert).

Abschließend erfolgte die Erörterung der unterschiedlichen Modellarten, die bei der Durchführung einer VIBN zur Anwendung gelangen; eine Übersicht dieser gibt Unterkapitel 3.7.

Die Zusammenfassung von Kapitel 3 stellt zugleich den Abschluss des theoretischen Teils dieser Masterthesis dar. Weitere theoretische Grundlagen, die für das Verständnis der praktischen Bereiche der Arbeit erforderlich sind, werden in weiterer Folge direkt an der jeweiligen Bedarfsstelle eingearbeitet. Das nachfolgende Kapitel hat die Ausgangssituation und Iststandsanalyse im Karosseriebau bei Magna Steyr zum Inhalt. Auf Basis dieser wird das zur erarbeitende VIBN-Modell abgeleitet. Zum einen werden die gegenwärtig eingesetzten Betriebsmittel, Maschinen und Fertigungsanlagen erhoben, zum anderen der gegenwärtig angewandte Produktentstehungsprozess.

¹³⁴ Vgl. Amann (1994), S. 61 ff.

¹³⁵ Vgl. Strassacker (1997), S. 33.

4 ISTSTANDSANALYSE BEI MAGNA STEYR

Kapitel 4 hat die Analyse der Ist- sowie der Ausgangssituation bezüglich Einführung und Anwendung der Methode der Virtuellen Inbetriebnahme im Karosseriebau zum Inhalt. In den nachfolgenden Unterkapiteln werden die Aspekte und Einflussgrößen des Produktentstehungsprozesses, der eingesetzten Anlagentechnik in den Karosseriebauten sowie des eingesetzten Werkzeugs zur Durchführung virtueller Absicherungen behandelt. Ebenso wird auf die Motivation zur Anwendung einer VIBN eingegangen. Die Auseinandersetzung mit den genannten Themen erfolgt jedoch nur insoweit, wie es für das Verständnis dieser Arbeit erforderlich ist, da diese bereits ausführlich in *Huber (2017)* behandelt wurden.¹³⁶

4.1 Motivation und Zielsetzung der Virtuellen Inbetriebnahme

Die Magna Steyr Fahrzeugtechnik AG & Co KG (MSF) betreibt an ihrem Standort in Graz mehrere Karosseriebauten zur Fertigung und Herstellung von ganzen Rohkarossen. Diese werden nachfolgend in den Technologiebereichen Lackiererei und Endmontage zu Gesamtfahrzeugen unterschiedlicher namhafter Fahrzeughersteller komplettiert. Gemeinsames Attribut der Karosseriebauten ist der hohe Automatisierungsgrad hinsichtlich Fügetechnik und Geometriebildung unabhängig von Stückzahlauslegung, Varianz der zu fertigenden Produkte, sowie Fertigungstiefe. So haben Fügetechniken wie beispielsweise Widerstandspunktschweißen, Laserschweißen, Bahnschweißen, Nieten, Schrauben und Kleben einen Automatisierungsgrad von annähernd 100 % erreicht. Ebenso weist die Geometriebildung von Einzelteilen zu Baugruppen und in weiterer Folge zur Gesamtkarosserie den selben Grad der Automatisierung auf, wie auch der damit verbundene Bauteiltransport innerhalb der unterschiedlichen Karosseriebauanlagen.

Wie bereits in Unterkapitel 3.3 beschrieben, stellt die Erfordernis schnellerer Markteintritte¹³⁷ in Kombination mit steigender Komplexität der Anlagentechnik und Fertigungssysteme¹³⁸ eine neue Herausforderung für Anlagenbauer und Anlagenbetreiber dar. Diese Herausforderung gilt es auch für Magna Steyr zu bewältigen, da traditionelle Planungs- und Absicherungsmethoden in der Anlagen- und Steuerungstechnik stetig an Wirkung verlieren. Auch die klassische Inbetriebnahme stößt im Zuge der Realisierungsphase von Produktionsanlagen im Karosseriebau aufgrund zeitlicher Rahmenbedingungen, Produktvarianz und Automatisierungsgrad an ihre Grenzen. Dementsprechend verfolgt Magna Steyr die Zielsetzung, mittels Einführung und Anwendung der VIBN den vorhin genannten Einflüssen und geänderten Rahmenbedingungen im Entstehungsprozess von Produkt und Produktionsanlage gerecht zu werden.

Als erstes Hauptziel im Zuge der Implementierung der Virtuellen Inbetriebnahme wurde die frühestmögliche Bewertung, Absicherung und Bestätigung von steuerungstechnischen Abläufen bei MSF identifiziert. Diese beinhaltet unter anderem die Koordination des Materialflusses durch die Karosseriebauanlagen, welche mittels Förderbändern, Stauförderern und Industrierobotern praktiziert wird. Neben dem Gutablauf müssen in diesem Zusammenhang auch Szenarien wie Ein- und Ausschleusvorgänge von Bauteilen abgesichert werden. Auch der Einfluss unterschiedlicher Produktvarianten wie Linkslenker-/ Rechtslenkerfahrzeuge,

¹³⁶ Vgl. Huber (2017).

¹³⁷ Vgl. Zäh/Reinhart (Hrsg.) (2006), S. 4 (Kapitel 2).

¹³⁸ Wiendahl/ Hegenscheidt/ Winkler (2002), S. 650 - 655.

Normaldach-/ Panoramadachaustattung oder Ersatzteilproduktion muss untersucht und analysiert werden. Ebenso gilt es zeitnah die Synchronisierung der Fertigungsschritte und Fertigungssequenzen der eingesetzten Roboter zu verifizieren und zu validieren, damit diese trotz Ausnutzung gemeinsamer Arbeitsbereiche und Arbeitsräume ungehindert ihre jeweiligen Arbeitsschritte ausführen können. Besonderes Augenmerk in diesem Zusammenhang gilt der Bestätigung des Verriegelungskonzeptes der Industrieroboter zueinander. Dieses dient zur Sicherstellung der Maschinensicherheit zwischen den eingesetzten Robotern unter Berücksichtigung beziehungsweise Gewährleistung einer ungehinderten Abarbeitung der auszuführenden Fertigungssequenzen. Dieses erste Teilziel soll mittels Anwendung einer geeigneten VIBN-Methode erreicht werden, da damit programmierte Fertigungsabläufe auf Speicherprogrammierbaren Steuerungen abgesichert werden können. Wie die Analysen in Unterkapitel 3.4 ergaben, kann eine Absicherung sowohl auf Anlagenebene, Zellenebene sowie Maschinenebene in Kombination mit einem Simulationsmodell durchgeführt werden. Inhalt des Praxisteils ist daher die Entwicklung einer Hardware-in-the-Loop Lösung zur Durchführung einer VIBN bei Magna Steyr. Abstraktionsebene soll die Ebene der Zelle sein, da diese die kleinste steuerungstechnische Einheit im Karosseriebau bei MSF darstellt.

Zweite Zielsetzung im Zusammenhang mit der Einführung und Anwendung der Virtuellen Inbetriebnahme ist die Vorbereitung, Projektierung und Absicherung von Umbauten bestehender Fertigungseinrichtungen und Anlagen. Auslöser dafür sind zum Beispiel Optimierungs- und Rationalisierungsmaßnahmen bestehender Fertigungsabläufe, sowie Integration zusätzlicher Produkte oder Produktvarianten in bestehende Produktionsanlagen. Für beide Anlassfälle stehen aufgrund ambitionierter Produktionsprogramme nur begrenzte Zeitfenster zu Verfügung, in der diese Änderungen und Erweiterungen an den Bestandsanlagen umsetzbar sind. Die Methode der VIBN soll zukünftig ermöglichen, programmierte Fertigungsabläufe, die durch eine reale SPS gesteuert und geregelt werden, sowohl an einem virtuellen Anlagenmodell als auch an seinem realen Äquivalent zu testen, zu prüfen und in Betrieb zu nehmen.

Für beide genannten Zielsetzungen ist auch die organisatorische Anpassung am Produktentstehungsprozess von Fertigungsanlagen bei MSF Voraussetzung. Diese wurde, neben der technischen Aufgabenstellung, ebenfalls im Zuge dieser Masterthesis erarbeitet. Es wird in weiterer Folge nur auf das erarbeitete Ergebnis der erforderlichen Reorganisation im Produktentstehungsprozess (PEP) von Magna Steyr eingegangen, da diese in Bezug auf die Anwendungspositionierung der VIBN im PEP unerlässlich ist. Details zu den dafür angewandten Methoden und Werkzeugen werden in dieser Arbeit nicht weiter behandelt, da dies den Umfang derselben erheblich überschreiten würde.

Dahingehend erfolgt im nachfolgenden Unterkapitel die Analyse der Istsituation von Anlagentechnik, virtueller Absicherung und Produktentstehungsprozess, welche als Grundlage und Ausgangspunkt des Praxisteils dieser Arbeit dient. Aufgrund der genannten Rahmenbedingung betreffend Umfang der Masterthesis wird auf den Entstehungsprozess von Produktionsanlagen nur insoweit eingegangen, wie es für das Verständnis der Arbeit erforderlich ist.

4.2 Anlagentechnik

Wie in der Einführung dieses Kapitels bereits angeführt, ist die Fertigungszelle jene Abstraktionsebene der Steuerungstechnik, die den Ausgangspunkt der nachfolgenden Forschungstätigkeiten bildet. Eine typische Fertigungszelle beinhaltet Industrieroboter, Vorrichtungen zur Geometriebildung und Arretierung von Bauteilen während des Fügeprozesses, sowie Zwischenablagen von Bauteilen. Ebenso sind erforderliche Peripherieeinrichtungen Bestandteil der Fertigungszelle, welche zum Beispiel zur automatischen Wartung von Elektrodenkappen von Schweißzangen, oder im Anlassfall als Bahnhöfe zur Ablage von robotergeführten Werkzeugen wie Schweißzangen oder Bauteilgreifern dienen. Angetrieben werden die genannten Betriebsmittel mittels elektrischer oder pneumatischer Energie, wobei diese Betriebsmittel über eine eigene Steuerung verfügen oder über die Kopfsteuerung der Zelle - der SPS - betrieben werden. Zur Vermeidung von unzulässigen Betretungen durch Bedien- und Wartungspersonal sind die Fertigungszellen durch einen Schutzzaun eingehaust sowie Automatikbereiche dadurch zueinander abgegrenzt. Abb. 37 verdeutlicht den typischen Aufbau einer Fertigungszelle in grafischer Form anhand der Zelle HB4-BA1. An dieser Zelle wird in weiterer Folge das VIBN-Modell aufgebaut und erprobt. Daher dient diese Fertigungszelle weiterführend als Grundlage zur Istandserhebung der Anlagentechnik.

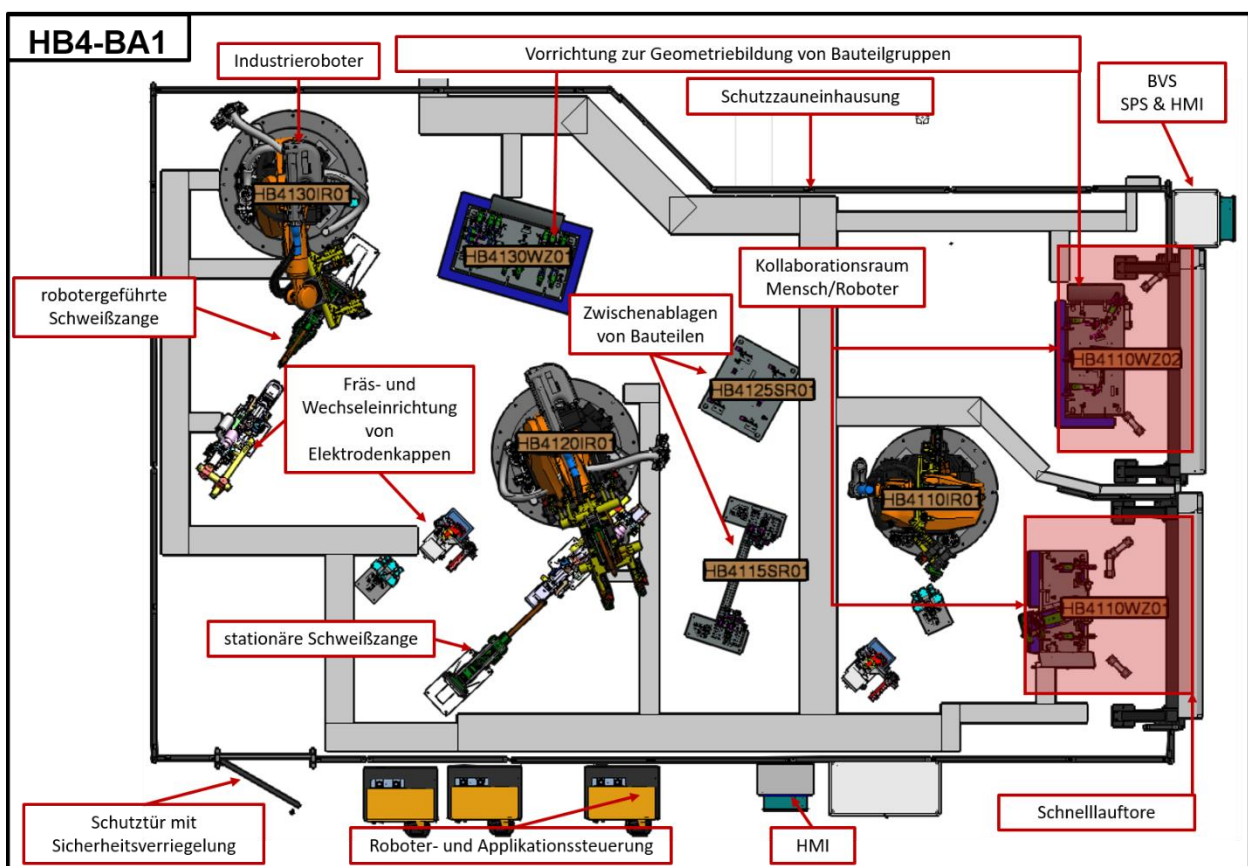


Abb. 37: Typischer Aufbau einer Karosseriebauzelle bei MSF, Quelle: Eigene Darstellung.

Betreten kann die Zelle ausschließlich über die dafür vorgesehenen Schutztüren werden. Kollaborationsbereiche zwischen Werker und Roboter sind durch trennende Schutzeinrichtungen wie Schnellaufitore oder durch optische Schutzeinrichtungen wie Laserscanner oder Lichtgitter überwacht und geschützt, wie dies ebenfalls aus Abb. 37 ersichtlich wird.

Steuerungstechnisch ist die Fertigungszelle als Speicherprogrammierbares Steuerungssystem (SPS-System) aufgebaut. Kernelement ist die SPS, sowie die vom Elektrokonstrukteur konfigurierten Peripheriegeräte, die für das Automatisierungssystem notwendig sind.¹³⁹ Vom strukturellen Aufbau entspricht dieses SPS-System dem generischen Automatisierungssystem, wie es in Unterkapitel 3.2 dargestellt wird. Aus Sicht der Steuerungstechnik wird bei MSF eine Fertigungszelle als Betriebsartenbereich (BA) bezeichnet. Der BA stellt sicherheitstechnisch einen Wirkungsraum dar, von dem die SPS im Fehlerfall alle Energien wegschaltet, von denen eine Gefahr für Leib und Leben ausgehen kann. Eingegrenzt wird der jeweilige BA durch eine Schutzzaunumhausung, die als mechanische Abgrenzung denselben umschließt. Zur Detektion von Bauteilen in den Spannvorrichtungen werden optische oder induktive Sensoren eingesetzt. Induktive Sensorik findet ebenfalls bei der Detektion von Stellgliedpositionen von Bauteilspannern, Stiftziehzylindern und Schwenkarmen Anwendung. Elektrisch angetriebene Asynchronmotoren werden zum Antreiben von Fördertechnik für den Bauteiltransport oder zum Positionieren von Dreh- und Schwenkeinheiten im Zuge des Fertigungsprozesses eingesetzt. Geregelt und gesteuert werden die Asynchronantriebe durch passende Frequenzumrichter. Zur Übertragung der durch die genannte Aktuatorik und Sensorik erzeugten sowie benötigten Eingangs- und Ausgangssignale werden geeignete Feldgeräte eingesetzt, welche die erforderlichen Informationen der SPS zur Verfügung stellen oder Anweisungen von der SPS verarbeiten. Vernetzt und verbunden werden die Feldgeräte zur SPS mittels Feldbus, wobei MSF das auf Industrial Ethernet basierende Feldbusprotokoll Profinet IO/Profisafe einsetzt. Einen Überblick der angewandten Systemarchitektur der Steuerungstechnik gibt Abb. 38.

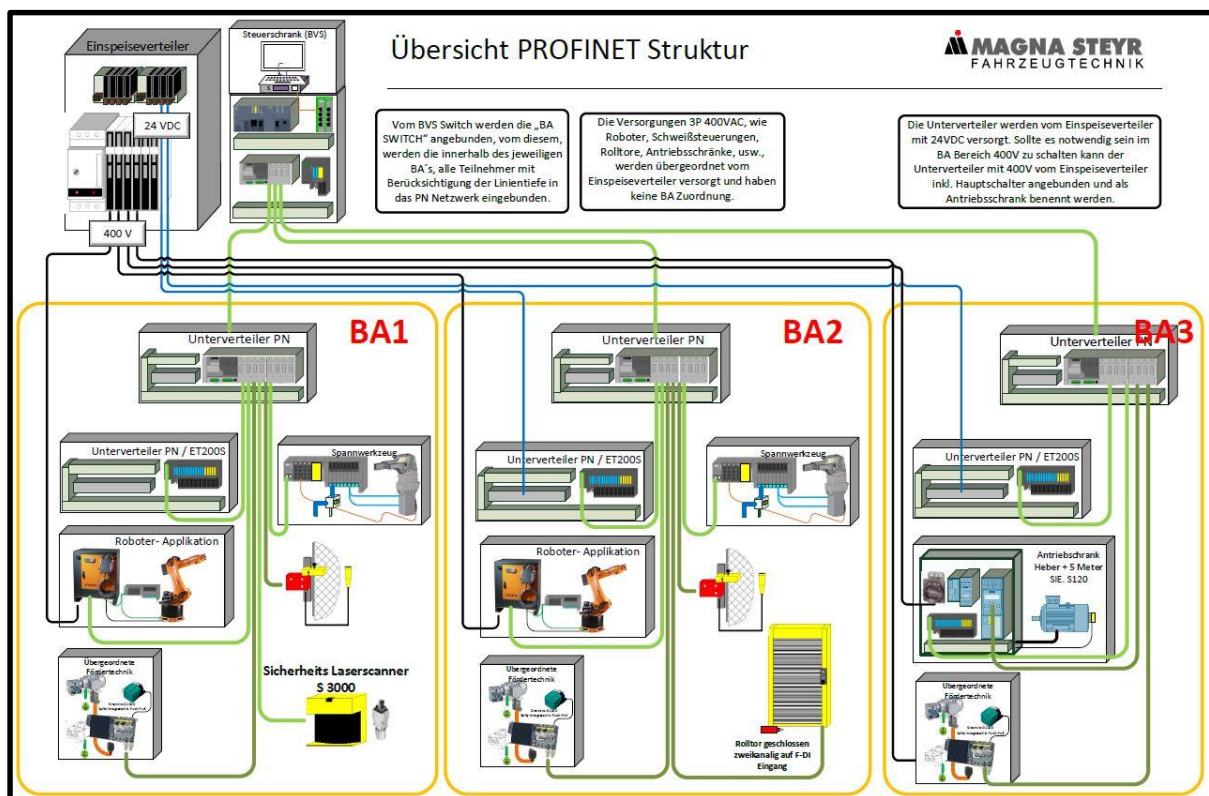


Abb. 38: Steuerungsarchitektur eines SPS-Systems bei MSF, Quelle: Magna Steyr.

¹³⁹ Vgl. DIN EN 61131-1 (März 2004), S. 6.

Zur Abarbeitung der für die Speicherprogrammierbare Steuerung vorgesehenen Aufgaben wird eine Central Processing Unit (CPU) der Marke Siemens vom Typ *CPU 319F-3 PN/DP* verwendet.

Diese verfügt über die Funktionalität eines Profinet IO-Controllers und unterstützt sowohl Realtime (RT) wie auch Industrial Realtime (IRT) Kommunikation. Des Weiteren verfügt diese CPU über Anschlussports für Datenübertragungsprotokolle wie TCP/IP und UDP zur Übertragung von nicht zeitkritischen Informationen und Daten.¹⁴⁰

Ebenso ist diese CPU dafür ausgelegt, Programme zur Sicherstellung der Personensicherheit nach DIN EN ISO 13849-1 auszuführen.¹⁴¹ Details zu allen weiteren eingesetzten Feldgeräten können aus der im Anhang beigefügten Datei *Th03rhb4-Hardware-config.cfg* entnommen werden. Diese Datei enthält den vollständigen Export der Hardwarekonfiguration des SPS-Systems sowie der darin eingesetzten Feldgeräte und angewandte Feldbustopologie des SPS-Bereichs. Die im Zuge der Projektumsetzung verwendete Software zur Projektierung und Programmierung der SPS-Software, des Feldbussystems sowie der Anlagenvisualisierung ist der im Anhang befindlichen Datei *Th03rhb4-Systeminfo-SPS.docx* zu entnehmen. Abb. 39 stellt die projektierte Feldbustopologie des SPS-Systems TH03RHB4 grafisch dar, wobei der Betriebsartenbereich HB4-BA1 durch eine rote Umrandung hervorgehoben ist.

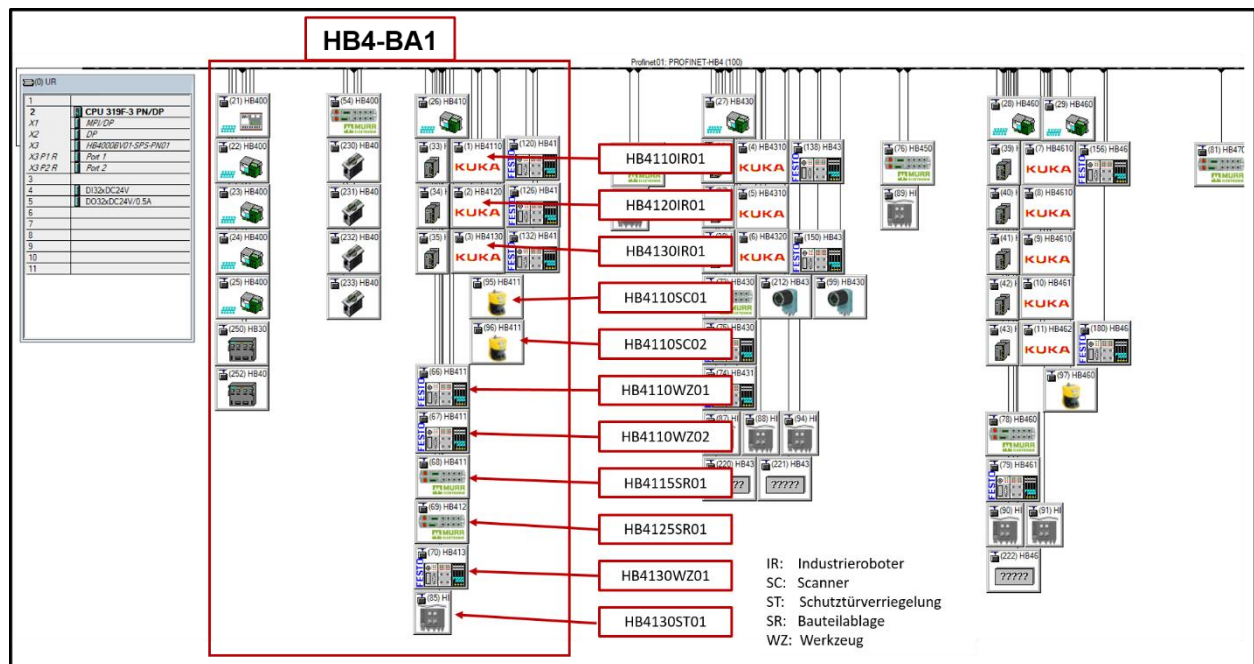


Abb. 39: Feldbustopologie des SPS-Systems TH03RHB4, Quelle: Eigene Darstellung.

Aus Sicht der Feldbustopologie und der darin angeführten Feldgeräte spiegelt Abb. 39 den Aufbau der Fertigungszelle HB4-BA1 wider und stellt somit das steuerungstechnische Äquivalent zu Abb. 37 dar. Aus beiden Darstellungsformen geht hervor, dass besagter Zelle drei Industrieroboter der Marke Kuka zugeordnet sind. Die drei Roboter verfügen jeweils über eine eigene Steuerung, die im Profinet IO Netz des SPS-Systems eingebunden ist und auf dieser Steuerungsebene als Slave im Feldbus des SPS-Systems fungiert.

¹⁴⁰ Vgl. Siemens AG (2018), Online-Quelle [15.09.2018].

¹⁴¹ Vgl. Siemens AG (2018), Online-Quelle [15.09.2018].

Im Gegensatz zu den restlichen Betriebsmitteln verfügt jeder Roboter zusätzlich über ein eigenes Feldbusnetz, in dem der Roboter als Master auftritt.

Ebenfalls sind drei Spannvorrichtungen zur Geometriebildung und Bauteilarretierung Bestandteil der Zelle. Die Spannvorrichtungen sind mit einer Ventilinsel des Fabrikats Festo ausgestattet. Die Ventilinsel dient einerseits zum Antreiben der pneumatischen Stellglieder, andererseits zum Erfassen der jeweiligen Stellgliedposition respektive Endlagen und des Bauteilstatus durch die dafür eingesetzte Sensorik. Das technische Umsetzungskonzept wird anhand Abb. 40 an Vorrichtung HB4110WZ02 verdeutlicht.

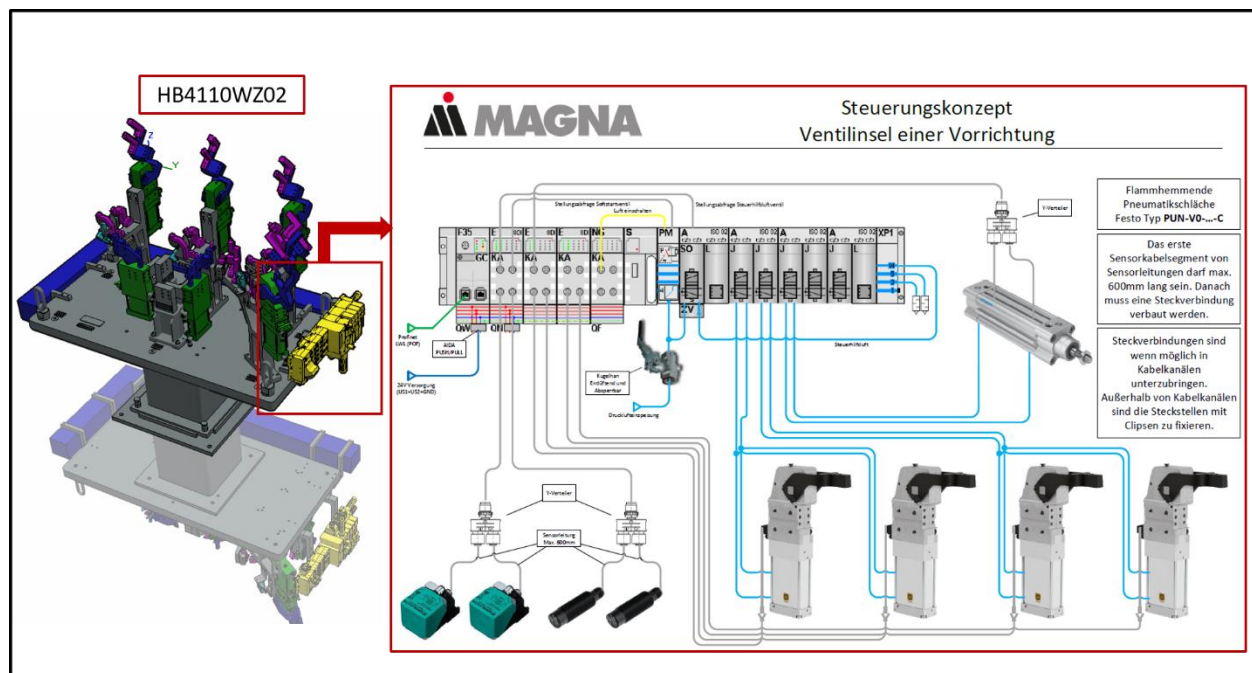


Abb. 40: Steuerungskonzept mittels Ventilinsel anhand der Vorrichtung HB4110WZ01, Quelle: Eigene Darstellung.

Die Sensorik der beiden Bauteilablagen wird durch Feldbusmodule der Marke Murr zusammengefasst und in das Profinet IO-Netz eingebunden. Die beiden Scanner zur sicherheitstechnischen Überwachung der beiden Kollaborationsbereiche zwischen Mensch und Roboter verfügen über eigene Controller zur Teilnahme im Feldbus. Hervorzuheben ist hierbei, dass diese beiden Feldgeräte aufgrund ihrer Sicherheitsrelevanz mit dem sicherheitsgerichteten Teil der SPS kommunizieren. Betreten kann die Zelle über eine Schutztür werden, die auf Höhe des Roboters HB4130IR01 positioniert ist. Diese ist mit einer Sicherheitsverriegelung ausgestattet, die ebenfalls durch den sicherheitsgerichteten Teil der SPS überwacht und verwaltet wird.

Auf weitere Komponenten und Betriebsmittel wird in diesem Unterkapitel nicht weiter eingegangen, da diese für die Erstellung und Bildung des VIBN-Modells von untergeordneter Bedeutung sind. Das anschließende Unterkapitel hat jene Werkzeuge zum Inhalt, die gegenwärtig zur Virtuellen Absicherung von Karosseriebauanlagen bei Magna Steyr eingesetzt werden.

4.3 Werkzeuge zur Virtuellen Absicherung

Zur Virtuellen Absicherung des Fertigungsprozesses auf Ebene der Produktionsanlage, der Fertigungszelle, sowie der Robotik wird das datenbankbasierende Softwaresystem *Process Designer/Process Simulate on emServer* von Siemens PLM eingesetzt. Das Softwaresystem besteht aus der Applikation *Process Designer (PD)* als Werkzeug zur Prozessplanung in einer 3D-Umgebung¹⁴², sowie der Applikation *Process Simulate (PS)*.

Process Simulate dient zur virtuellen Absicherung mittels Offlineprogrammierung von Industrierobotern, sowie zeitbasierender wie auch ereignisbasierender Simulation von Einzelkomponenten bis hin zu ganzen Fertigungszellen und Anlagenbereichen.¹⁴³ Planungswerkzeug/PD und Engineeringwerkzeug/PS stehen zueinander in einem bidirektionalen Daten- und Informationsaustausch, welcher über die Softwareapplikation emServer ermöglicht wird. Ebenfalls ist es durch Einsatz von emServer möglich, mehrere Anwendungsinstanzen von Process Designer und Process Simulate auf einem Projekt arbeiten zu lassen. Dies erlaubt zeitgleiches und paralleles Arbeiten von ganzen Teams am selben Projekt. Metadaten werden auf einer Oracle Datenbank gespeichert, Geometriedaten in 2D oder 3D auf einem geschützten Filespace im Dateiformat Jupiter Tessellation (JT). Die Architektur dieses Softwaresystems stellt Abb. 41 in schematischer Form dar.

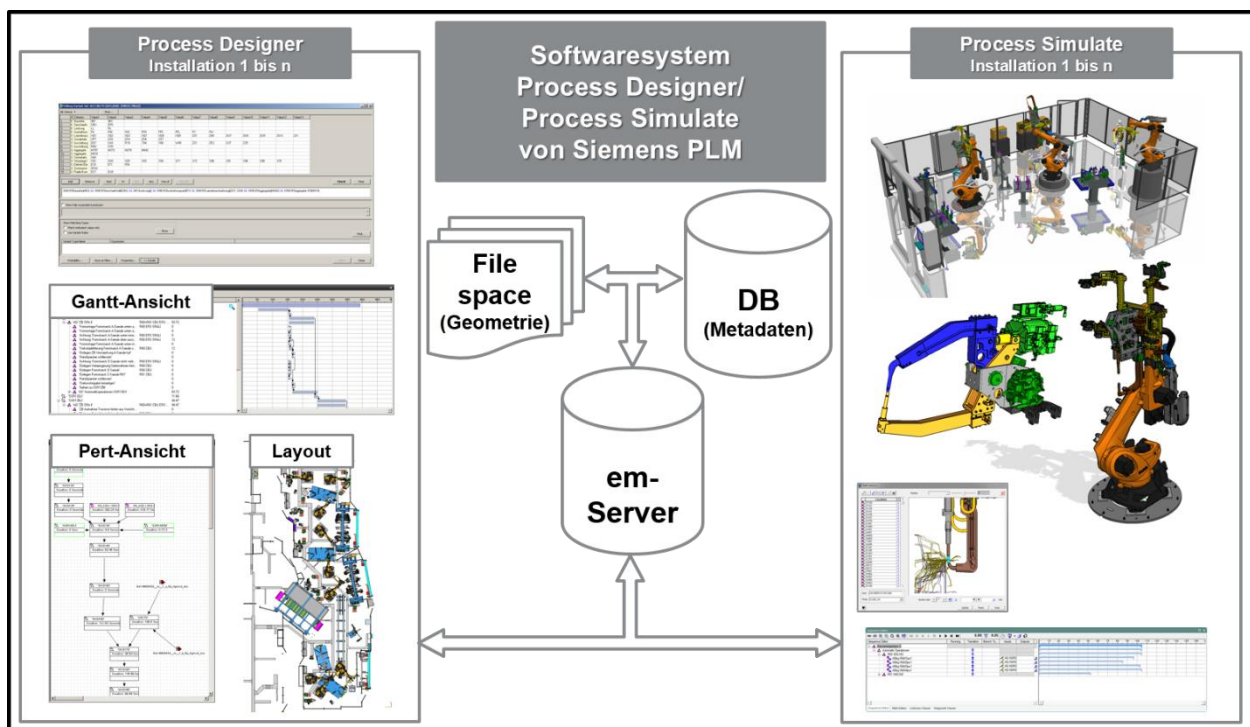


Abb. 41: Architektur des Softwaresystems Process Designer/Process Simulate, Quelle: Eigene Darstellung.

Aufgrund der Möglichkeit, Process Simulate sowohl an eine Soft-SPS wie auch an eine reale SPS anbinden zu können, ist PS jenes Werkzeug, das zur Modellbildung im Zuge der VIBN eingesetzt wird.

¹⁴² Vgl. Siemens Product Lifecycle Management Software (2018), Online-Quelle [15.09.2018].

¹⁴³ Vgl. Siemens Product Lifecycle Management Software (2018), Online-Quelle [15.09.2018].

Gegenwärtig wird bei MSF die Version 13.1 von Process Designer/Process Simulation emServer im Produktiveinsatz verwendet. Aufgrund der erweiterten Funktionalitäten und der verbesserten Rechenperformance wird zur Modellbildung und VIBN-Erprobung die Version 14.1 eingesetzt. Detaillierte Informationen hinsichtlich der verwendeten Softwareversion, der darin angewandten Zusatzinstallationen wie Robotercontroller und des installierten Hotfix-Levels sind der Datei *eMPowerAnalysis_14.1.html* zu entnehmen. Diese enthält ebenfalls eine ausführliche Aufstellung der Systeminformationen der eingesetzten Rechnerhardware, auf der Process Simulation installiert ist.

Der letzte Teil von Kapitel 4 hat den gegenwärtigen Produktentstehungsprozess bei MSF zum Inhalt. Die Analyse ist von Relevanz, da das abschließende Unterkapitel 4.4 den Ausgangspunkt zur anschließenden Anwendungspositionierung der VIBN im neu zu gestaltenden PEP darstellt. Ebenso beinhaltet dieses Unterkapitel eine Übersicht über alle Phasen der Virtuellen Absicherung, die bis zum Start der VIBN bereits durchgeführt und abgeschlossen sind.

4.4 Produktentstehungsprozess

Grundlage der Produktentstehung bei MSF ist das *Magna Steyr Development System (MSDS)*, das den Entwicklungsprozess von Gesamtfahrzeugen bei Magna Steyr steuert und regelt. Von den darin angeführten Phasen der Produktentstehung sowie den dazugehörigen Gates leiten sich nachfolgend alle weiteren Phasen der erforderlichen Teildisziplinen ab, die zur Serienfertigung eines Gesamtfahrzeugs notwendig sind. Dazu gehören die Teildisziplinen *Engineering, Styling, Supply Chain, Quality, Finance, Marketing, After Sales* sowie *Production*. In Abb. 42 werden die Gates und Meilensteine, wie auch die dazugehörigen Phasen des MSDS dargestellt.

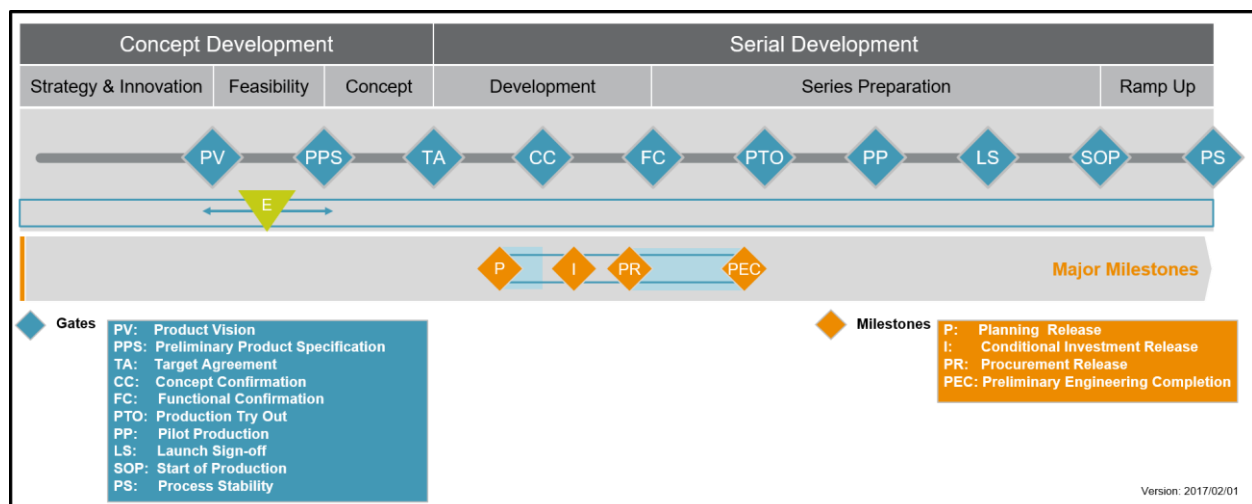


Abb. 42: Gates und Meilensteine des Magna Steyr Development System, Quelle: Magna Steyr (leicht modifiziert).

Darauf basierend wird die Teildisziplin *Production* einer Detaillierung unterzogen, die weiterführend alle relevanten Phasen in Bezug auf den Entstehungsprozess einer Produktionsanlage enthält. Diese sind in chronologischer Reihenfolge die Abschnitte *Production Process Definition, Strategic Planning Simulation/Detail Planning, Manufacturing Design, Series Devt. Design, Line Build, Training* und abschließend die Phase der *Process Verification*, wie es Abb. 43 in grafischer Form verdeutlicht.

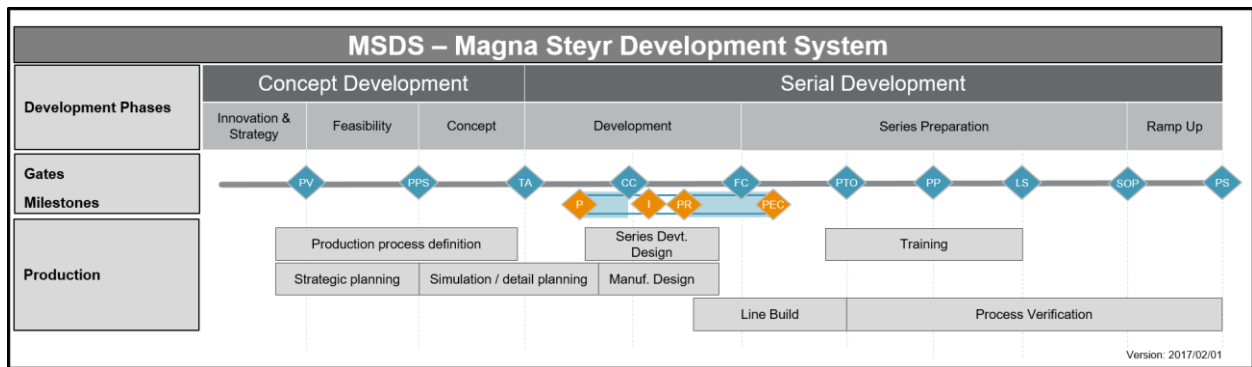


Abb. 43: Entstehungsprozess von Produktionsanlagen basierend auf MSDS, Quelle: Magna Steyr (modifiziert).

Auf Basis der Entwicklungsphasen für den Bereich *Production* leiten sich weiterführend die Maßnahmen zur *Prozessentwicklung und Absicherung* ab, die die Ingenieursdisziplinen Maschinenbau, Elektrotechnik und Programmierung umfassen. In Abhängigkeit des Projektfortschrittes lassen sich die in Abb. 44 dargestellten Teilbereiche der Virtuellen Absicherung hinsichtlich ihres Schwerpunkts einer der vorhin genannten Ingenieursdisziplinen zuordnen.

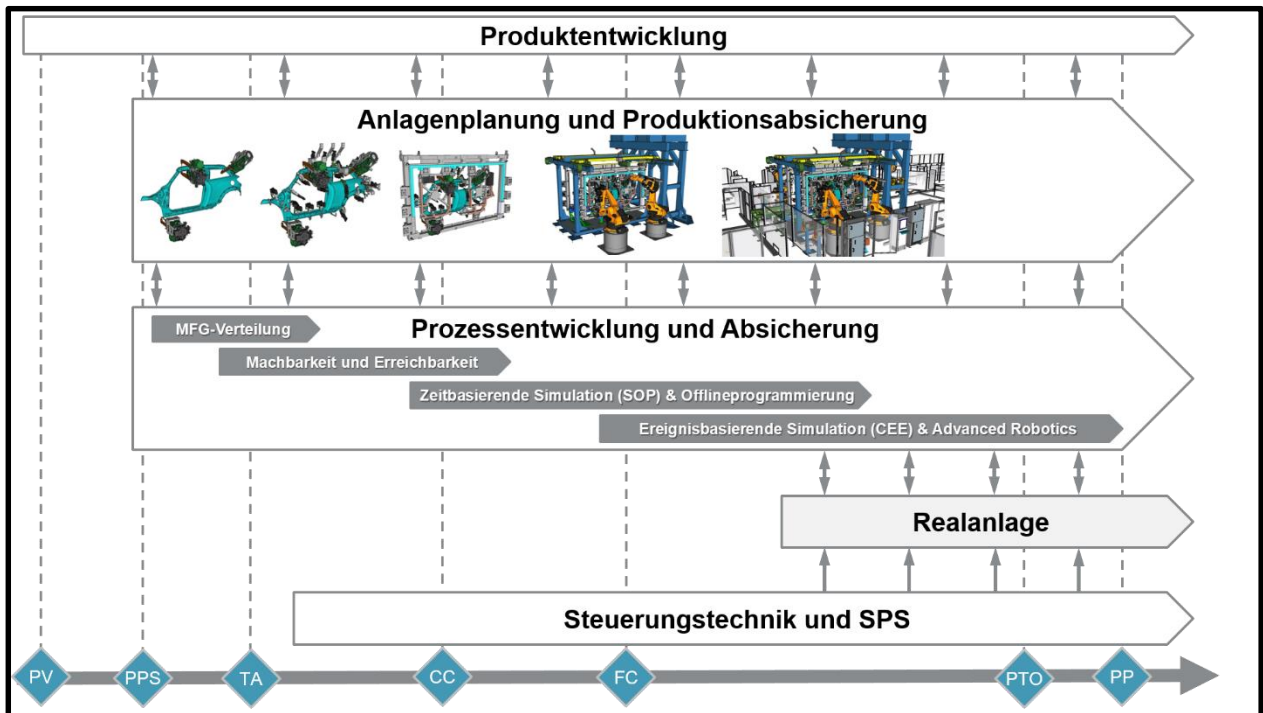


Abb. 44: Phasen der Virtuellen Absicherung von Karosseriebauanlagen bei MSF, Quelle: Eigene Darstellung.

So werden *MFG-Verteilung* (Manufacturing Feature), welche die Distribution von Verbindungselementen wie Schweißpunkte, Schrauben, Nieten und so weiter zum Inhalt hat, dem Maschinenbau zugeordnet, ebenso wie *Machbarkeit und Erreichbarkeit*. In diesem Projektabschnitt finden neben den Machbarkeitsanalysen ebenfalls die erforderlichen Tätigkeiten zum Konzeptentwurf der Vorrichtungstechnik statt, sowie die Festlegung des Spann- und Fixierkonzepts zur Geometriebildung der Karosseriesegmente. Abschluss findet diese Projektphase mit der Detailkonstruktion der Vorrichtungstechnik.

Zeitbasierende Simulation und Offlineprogrammierung bildet den Übergang vom maschinenbautechnischen Schwerpunkt zur elektrotechnischen Schwerpunktsetzung. Während die zeitbasierende Simulation

von Experten des Maschinenbaus durchgeführt wird, ist die Offlineprogrammierung der Industrieroboter Domäne der Elektrotechnik. Resultat der zeitbasierenden Simulation ist die Sequence Of Operation (SOP), welche den Fertigungsprozess der Roboterzelle mittels eines dreidimensionalen, auf zeitlichen Abfolgen basierenden Simulationsmodell bestätigt. Interaktionen zwischen Robotern untereinander oder von Robotern zu Schweißvorrichtungen werden durch generische Simulationsbefehle erzeugt. Den einzelnen Fertigungsschritten sind fixe Zeitwerte zugeordnet, welche nach der vorgesehenen Abfolge des Materialflusses durch die Fertigungszelle abgearbeitet werden. Hingegen werden in der Offlineprogrammierung die Abfolgen der Robotertätigkeiten durch Verwendung herstellerbezogener Roboterbefehle und Robotersyntax angereichert. Ebenso findet die Bahnplanung und Bewegungssteuerung durch Einsatz der *Realistic Controller Simulation* (RCS) ihre Umsetzung. Das dazu erforderliche RCS-Interface wird vom jeweiligen Roboterhersteller zur Verfügung gestellt und erlaubt die Anwendung der Bahnplanungsalgorithmen basierend auf dem mathematischen Originalmodell des Roboterherstellers.¹⁴⁴ Des Weiteren verfügen die Roboterprogramme nach Abschluss der Offlineprogrammierung über alle erforderlichen Applikationspunkte wie auch Umfahrpunkte. Auch muss nach Definition von Magna Steyr das fertiggestellte Offlineprogramm mit sämtlichen Logikbefehlen versehen sein, die für die ordnungsgemäße Ausführung des Programms am realen Roboter notwendig sind. Nach Abschluss der Offlineprogrammierung kann das virtuell erstellte Programm durch einen Exportvorgang aus dem Simulationssystem auf dem Realroboter ausgeführt werden. Abschluss findet die Phase der Prozessentwicklung und Absicherung in der Ereignisbasierenden Simulation. Diese Methodik beruht auf der Grundlage, dass Prozessabläufe nicht auf fixen Zeitwerten beruhen, sondern von einer im Simulationssystem eingebetteten Kopfsteuerung gesteuert, geregelt und verwaltet werden. Die Funktionsweise der Kopfsteuerung ist mit der Verknüpfungssteuerung einer SPS vergleichbar. Durch Anwendung dieser Simulationsmethode ist es möglich, Fertigungsabläufe von Industrierobotern im Simulationsmodell zu synchronisieren. Auch Aktuatoren von Spannvorrichtungen lassen sich dadurch ansteuern sowie Eingangssignale von Endlagen derselben auswerten. Unter Ausnutzung des vollen Funktionsumfangs der Ereignisbasierenden Simulation ist es möglich, den gesamten Produktionsablauf einer Fertigungszelle unter realitätsnahen Rahmenbedingungen abzubilden. Vom Materialfluss bis zur Interaktion zwischen Roboter und Fertigungsvorrichtung wird die Fertigungszelle mittels Model-in-the-Loop Ansatz abgesichert.

Parallel dazu finden die erforderlichen Planungs- und Entwicklungsschritte in der Domäne der Steuerungstechnik und SPS statt, wie es Abb. 44 grafisch verdeutlicht. Daraus ist ebenso ersichtlich, dass Mechanik und Robotik bereits während der Prozessentwicklung und Absicherung parallel beziehungsweise überschneidend behandelt werden. Ebenso finden beide Fakultäten Abbildung in einem gemeinsamen Simulationsmodell. Steuerungstechnik und SPS hingegen treffen erst in der Realanlagen mit Mechanik und Robotik zusammen. Dadurch ist es gegenwärtig nicht möglich, die in Unterkapitel 3.3 angeführten Vorteile und Ziele zu erreichen, die durch eine frühzeitige Zusammenführung aller Disziplinen möglich wären.

Um diesen Wettbewerbsnachteil zu beseitigen, ist eine VIBN-Methode bei Magna Steyr zu entwickeln, deren Entstehung und Entwicklung in den nachfolgenden Kapiteln erläutert wird.

¹⁴⁴ Vgl. Weber (2009), S. 122.

5 ANWENDUNGSPositionIERUNG DER VIRTUELLen INBETRIEBNAHME

Die Virtuelle Inbetriebnahme stellt eine geeignete Engineering- wie auch Absicherungsmethode dar, die es ermöglicht, gegenwärtige und zukünftige Herausforderungen im Zuge von Karosseriebauprojekten zu beherrschen. Dementsprechend ist die richtige zeitliche Positionierung dieser Maßnahme innerhalb des Projektentwicklungsprozesses, der bei Magna Steyr Anwendung findet, von Bedeutung. Wegen der Relevanz dieser Thematik soll die Anwendungspositionierung der VIBN innerhalb der Gesamtprojektentwicklung hervorgehoben werden. Das Ergebnis mehrerer Workshops stellt Abb. 45 in grafischer Form dar, die auch die richtige zeitliche Positionierung der Virtuellen Inbetriebnahme bei Magna Steyr zum Inhalt hatten.

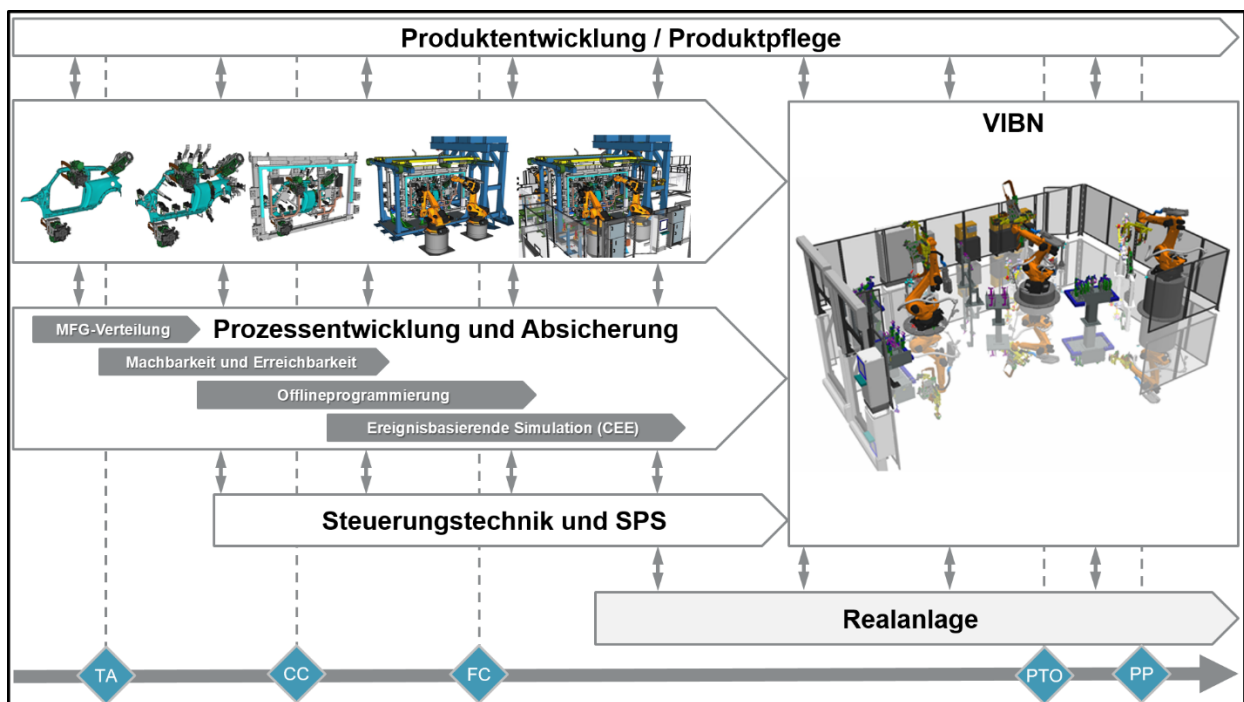


Abb. 45: Positionierung der VIBN im Rahmen der virtuellen Absicherung von Produktionsanlagen, Quelle: Eigene Darstellung.

Des Weiteren hat Abb. 45 den Zweck, die VIBN in Abhängigkeit der virtuellen Absicherungsschritte zu positionieren, die im Zuge der Projektentwicklung im Karosseriebau bei Magna Steyr auszuführen sind. Ebenso hat Abb. 45 die Aufgabe, das Erfordernis der interdisziplinären Zusammenarbeit zwischen Planung, mechanischer und elektrischer Konstruktion sowie der Softwareentwicklung zu verdeutlichen. Diese ist Voraussetzung für die Durchführung der Virtuellen Inbetriebnahme, da die rechtzeitige Bereitstellung der nötigen Daten und Informationen aus den voran genannten Fakultäten der Ingenieurwissenschaften grundlegend sind. Auch dient Abb. 45 dazu, die Notwendigkeit mechatronischer Verhaltensmodelle für die erfolgreiche Realisierung der Virtuellen Inbetriebnahme hervorzuheben. Erst diese stellen eine fakultätenübergreifende Interaktion zwischen Mechanik, Elektrotechnik sowie der Steuerungstechnik sicher. Eine weitere Darstellungsform bezüglich der Positionierung der Virtuellen Inbetriebnahme stellt Abb. 46 dar.

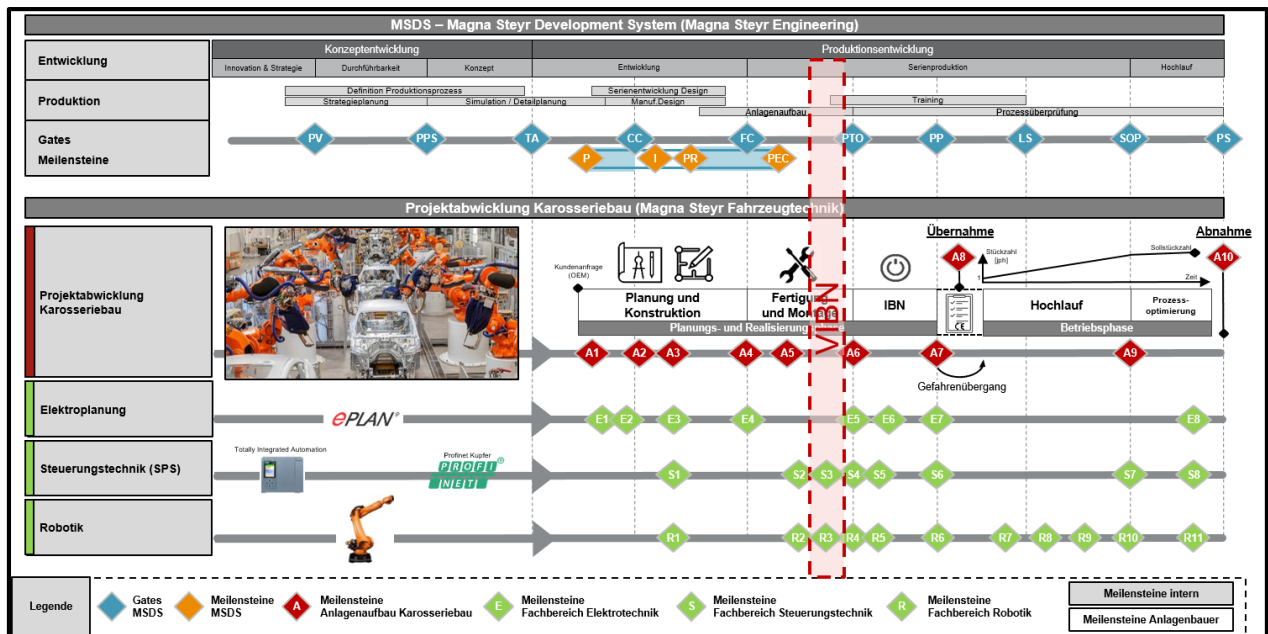


Abb. 46: Zeitliche Positionierung des Meilensteins der Virtuellen Inbetriebnahme im Anlagenentstehungsprozess, Quelle: Magna Steyr.

Aus Abb. 46 geht hervor, dass als Grundlage für alle Meilensteine zur Projektentwicklung im Karosseriebau das Magna Steyr Development System dient. Von diesem werden in weiterer Folge die Meilensteine A1 bis A10 abgeleitet. Diese beschreiben inhaltlich die allgemeinen Anforderungen der Projektentwicklung und des Anlagenaufbaus. Von diesen werden wiederum die Meilensteine der Fachdisziplinen Elektroplanung, Steuerungstechnik sowie Robotik abgeleitet. Der Elektroplanung sind die Meilensteine E1 bis E8 zugeordnet, der Steuerungstechnik die Meilensteine S1 bis S8 sowie der Robotik die Meilensteine R1 bis R11. Im Zusammenhang mit der VIBN sind die Meilensteine S3 und R3 von entscheidender Bedeutung. Diese markieren als Meilensteine den Abschluss der Virtuellen Inbetriebnahme in den Disziplinen Steuerungstechnik sowie Robotik. Somit wird die VIBN in Bezug auf die allgemeine Projektentwicklung nach dem Meilenstein A5 positioniert, der den Abschluss der Planungsaktivitäten repräsentiert, und vor dem Meilenstein A6, welcher das Ende aller Anlieferungstätigkeiten von Hardwarekomponenten darstellt. Hinsichtlich der Disziplinen Steuerungstechnik und Robotik wird die VIBN nach Abschluss der Offlineprogrammierung sowohl auf Seiten der SPS-Programmierung wie auch auf Seiten der Robotik positioniert. Damit wird für zukünftige Projekte die Möglichkeit geschaffen, mittels Anwendung der Virtuellen Inbetriebnahme sicherzustellen, dass vor Abschluss der steuerungstechnischen wie auch robotertechnischen Installationstätigkeiten fehlerfreie Softwareprogramme zur weiteren Verwendung zur Verfügung stehen. Dies wird durch Positionierung der VIBN vor den Meilensteinen S4 und R4 in Abb. 46 verdeutlicht.

An dieser Stelle findet das Ende der Auseinandersetzung in Bezug auf die Anwendungspositionierung der Virtuellen Inbetriebnahme bei Magna Steyr statt. Ebenfalls stellt der Abschluss der Anwendungspositionierung der VIBN den Startpunkt der technischen Entwicklungsaktivitäten des praktischen Teils der Masterthesis dar. Das nachfolgende Kapitel dieser Masterarbeit hat die erforderliche Infrastruktur zum Inhalt, die zur Durchführung einer Virtuellen Inbetriebnahme bei Magna Steyr nötig ist.

6 INFRASTRUKTUR FÜR DIE VIRTUELLE INBETRIEBNAHME

Kapitel 5 befasst sich mit dem Aufbau der notwendigen Infrastruktur, die für die Durchführung einer Virtuellen Inbetriebnahme im Karosseriebau bei Magna Steyr erforderlich ist. Aufgrund der Anforderungen und Zielsetzungen, die durch Anwendung dieser Methode bei MSF verfolgt werden, ist der Aufbau einer geeigneten Hardware-in-the-Loop Lösung notwendig. Die Beschreibung und Erläuterung dieser ist Inhalt der anschließenden Unterkapitel, wobei diese nachfolgend als HiL-Testkit bezeichnet wird.

6.1 Übersicht HiL-Testkit

Der HiL-Testkit, der neben dem virtuellen Anlagenmodell die technische Grundlage zur Durchführung einer VIBN im Karosseriebau bei MSF darstellt, muss allen Anforderungen aus Unterkapitel 4.2 gerecht werden. Zum einen ist es erforderlich, die reale SPS mit der darauf konfigurierten Feldbustopologie darzustellen, zum anderen muss die Möglichkeit geschaffen werden, unter Verwendung realer Feldbustelegramme den Informationsaustausch zwischen Real-SPS und virtuellem Anlagenmodell zu gewährleisten. Zusätzlich ist sicherzustellen, dass alle an der VIBN beteiligten Systemkomponenten geeignet miteinander vernetzt werden. Ebenfalls muss bei der informationstechnischen Vernetzung der einzelnen Systemkomponenten gewährleistet sein, dass sämtliche Datenpakete, die zwischen diesen ausgetauscht werden, ohne Einschränkung der Informationsgüte und Informationsmenge ihre Zieldestinationen erreichen. Auch die zur Bedienung durch das Vorortpersonal angewandten Eingabeeinrichtungen wie Schlüsselschalter, Pilz- und Leuchtdrucktaster müssen am HiL-Testkit vorhanden sein. Diese dienen zur Anwahl der Betriebsarten wie Handbetrieb und Automatikbetrieb, sowie zum Auslösen eines Not-Halts. Mittels Leuchtsignalen unterschiedlicher Frequenz und Farbe werden dem Bediener Anlagenzustände und Anforderungen der Produktionsanlage rückgemeldet. Resultierend aus diesen Anforderungen wurde der in Abb. 47 und Abb. 48 dargestellte HiL-Testkit entwickelt, welcher von der Instandhaltung des Karosseriebaus bei Magna Steyr angefertigt wurde.



Abb. 47: HiL-Testkit zur Durchführung einer VIBN bei MSF (Seitenansicht), Quelle: Eigene Darstellung.

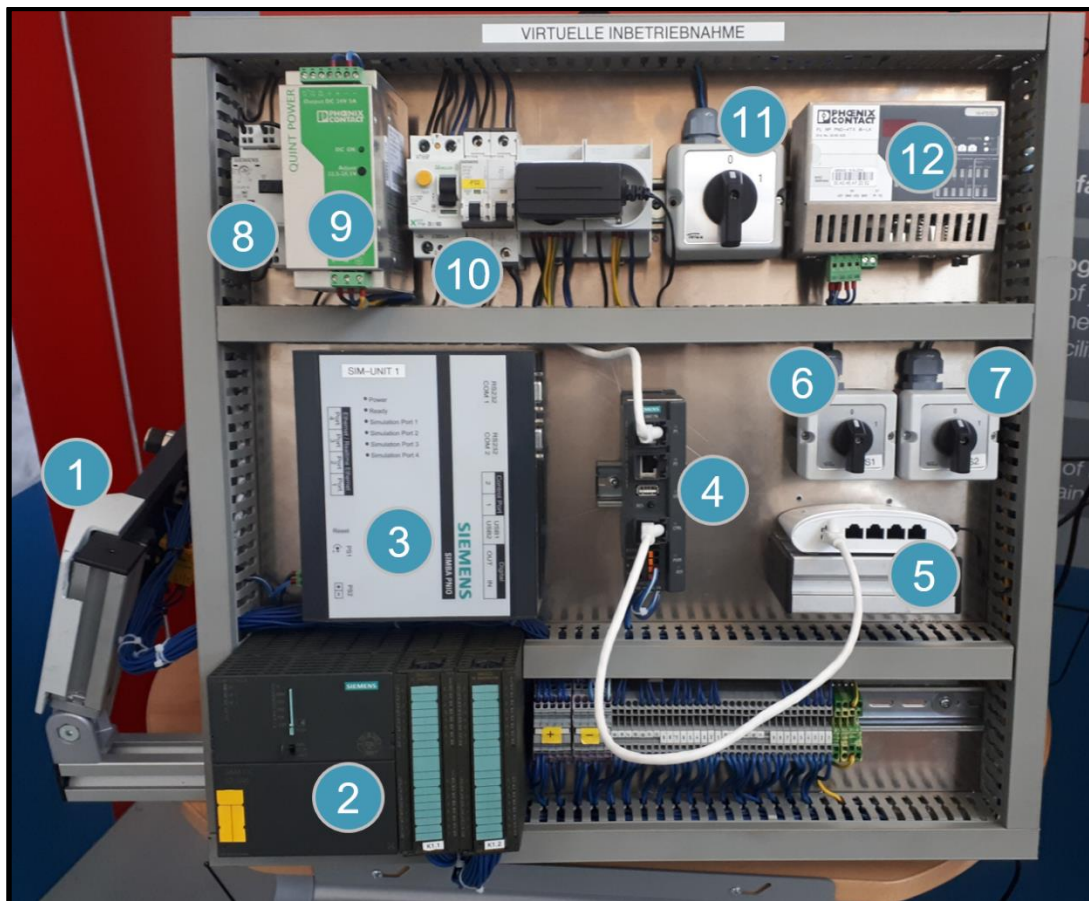


Abb. 48: HiL-Testkit zur Durchführung einer VIBN bei MSF (Frontansicht), Quelle: Eigene Darstellung.

Referenzierend auf Abb. 47 und Abb. 48 werden in nachfolgender Aufzählung die darin nummerierten Komponenten aufgelistet und bezeichnet:

- (1) Bedienfeld zur Betriebsartenwahl und Anlagenbedienung
- (2) Speicherprogrammierbare Steuerung der Marke Siemens vom Typ CPU 319F 3 PN/DP
- (3) Simba PNIO premium vom Hersteller Siemens zur Simulation von Profinet IO Devices
- (4) Simit Unit PN256 vom Hersteller Siemens zur Simulation von Profinet IO Devices
- (5) DGS-1005D 5-Port Layer2 Gigabit Switch zum Verbinden der erforderlichen Netzwerke
- (6) Schalter zum separaten Trennen der Simba PNIO premium von der Spannungsversorgung
- (7) Schalter zum separaten Trennen der Simit Unit PN 256 von der Spannungsversorgung
- (8) Motorschutzschalter
- (9) Stromversorgung - QUINT-PS-100-240AC/24DC/ 5 von Phoenix Contact zur Spannungsversorgung von SPS, Simba PNIO und Simit Unit mit 24 V Gleichspannung
- (10) Fehlerstromschutzschalter und Leitungsschutzschalter zur Gewährleistung des Fehlerschutzes und Zusatzschutzes nach ÖVE8001
- (11) Hauptschalter für das 24 V Gleichspannungsnetz
- (12) Zusätzlicher Proxy - FL NP PND-4TX IB-LK zur Einbindung optionaler Netzwerke

Aufgrund ihrer Bedeutung für die Funktionserfüllung des HiL-Testkits wird in Unterkapitel 6.2 vertiefend auf die Komponenten zur Simulation von Profinet-Teilnehmern eingegangen.

6.2 Simit Unit PN256 und Simba PNIO premium

Simit Unit und Simba PNIO sind Systemkomponenten, die zum schnellen Einrichten von Testumgebungen zur Durchführung von Factory Acceptance Tests (FAT-Tests) und Funktionstests von S7/PCS7-Anlagen dienen.¹⁴⁵ FAT-Tests stellen sicher, dass Produktfunktionalitäten den Kundenanforderungen entsprechen, und bieten die Möglichkeit, umgesetzte Kommunikationslösungen vor Aufbau beim Endkunden diesem zu präsentieren.¹⁴⁶ S7-Anlagen bilden den Überbegriff von Fertigungssystemen, die von einer Siemens-SPS gesteuert werden, PCS7-Anlagen den von Prozessleitsystemen in Prozess- und Fertigungsindustrien¹⁴⁷. Sowohl mit der Simit Unit PN256 wie auch mit der Simba PNIO premium lassen sich bis zu 256 Profinet IO Devices simulieren, die auf einem Profinet-IO Strang projektiert sind. Es können beliebige Profinet-IO Controller in die Profinet-IO Simulation eingebunden werden. Unter anderem sind mit diesen Simulationskomponenten auch Profinet-CPU's der Baureihe SIMATIC S7-300 simulierbar, denen auch die bei Magna Steyr eingesetzte Speicherprogrammierbare Steuerung CPU 319F 3 PN/DP zugeordnet wird.¹⁴⁸ Die Datenübertragung zwischen Simulations-PC und Simit Unit/Simba PNIO erfolgt mittels TCP/IP. Auf Seiten des Profinet-Netzwerkes findet der Datenaustausch zwischen den Teilnehmern im Profinet-Verbund durch Echtzeit-Kommunikation statt. Beide Simulations-Devices werden nicht in die Hardwarekonfiguration des SPS-Projekts SIMATIC-S7 eingebaut. Auch findet keine Anmeldung der Boxen als eigene Teilnehmer am Profinet-System statt – diese stellen unabhängige Hardware-Module dar und verhalten sich im Automatisierungssystem neutral.¹⁴⁹

Abb. 49 verdeutlicht die notwendige Systemarchitektur sowie das Systemverhalten von Simit Unit/Simba PNIO in grafischer Form.

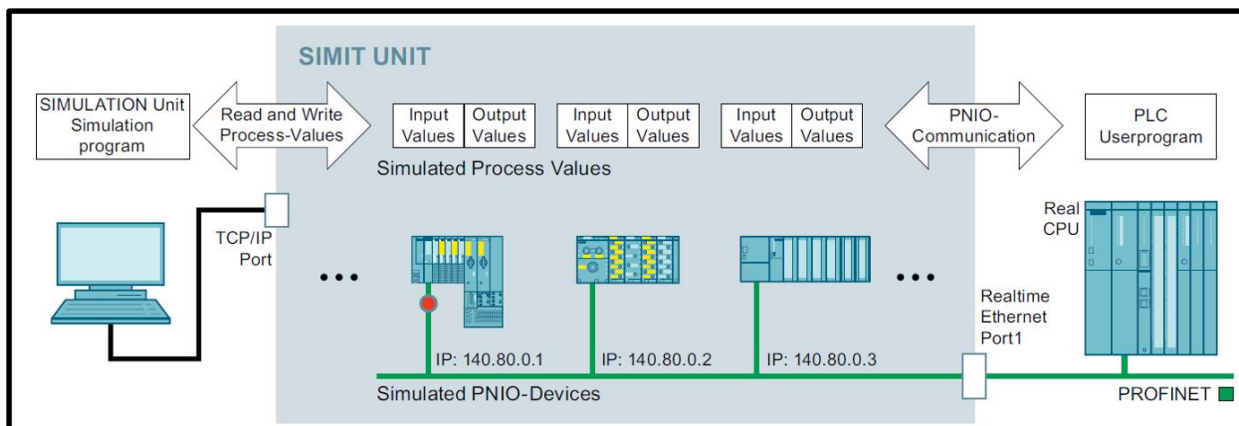


Abb. 49: Systemarchitektur und Systemverhalten von SIMIT Unit/Simba PNIO, Quelle: Siemens AG (Hrsg.) (07/2017), S. 57.

Ebenso werden aus Abb. 49 die angewandten Kommunikationsprotokolle auf Seiten der Simulation wie auch auf Seiten der realen Komponenten des Automatisierungssystems hervorgehoben.

¹⁴⁵ Vgl. Siemens AG (Hrsg.) (07/2017), S. 9.

¹⁴⁶ Vgl. Siemens AG (2018), Online-Quelle [28.09.2018].

¹⁴⁷ Vgl. Siemens AG (2018), Online-Quelle [28.09.2018].

¹⁴⁸ Vgl. Siemens AG (Hrsg.) (07/2017), S. 9.

¹⁴⁹ Vgl. Siemens AG (Hrsg.) (07/2017), S. 57.

Wie in Unterkapitel 4.2 bereits angeführt und in Abb. 38 grafisch verdeutlicht, werden bei Magna Steyr Feldgeräte unterschiedlicher Funktion innerhalb eines Feldbusstrangs eingesetzt. Speziell der gemischte Einsatz von Feldgeräten zur bestimmungsgemäßen Verwendung der Produktionsanlage mit Feldgeräten, die sicherheitsbezogene Aufgaben zur Sicherstellung der körperlichen Unversehrtheit nach EN ISO 13849-1 gewährleisten, ist hier hervorzuheben. Da Simit Unit PN256 und Simba PNIO premium das Protokoll Profisafe unterstützen, ist auch die Simulation von sicherheitsgerichteten Feldbusteilnehmern sichergestellt.¹⁵⁰

Eine Übersicht weiterer wichtiger Funktions- und Systemeigenschaften von Simit Unit PN256 und Simba PNIO premium gibt Tab. 6.

| Eigenschaft | SIMBA PNIO premium | SIMIT UNIT PN256 |
|--------------------------------|--------------------|------------------|
| Bestell Nr. | 9AE4120-1AB00 | 9AE4120-2AB00 |
| Max Anzahl Devices | 256 | 256 |
| Profinet Schnittstellen | 1 | 2 |
| minimale Zykluszeit | 250 µs | 250 µs |
| Device und Modulausfall | ✓ | ✓ |
| Bus Ausfall | ✓ | ✓ |
| Kanaldiagnosen | ✓ | ✓ |
| IRT class 2 | ✓ | ✓ |
| IRT class 3 | | ✓ |
| Profisafe V1.0 (PN/PN Koppler) | ✓ | ✓ |
| Profisafe V2.4 | ✓ | ✓ |
| Profisafe V2.6 | | ✓ |
| Shared Device | ✓ | ✓ |
| Redundanz MRP | | ✓ |
| System Redundanz S2 | | ✓ |
| System Redundanz R1 | | ✓ |
| integrierter Switch | | ✓ |

Tab. 6: Übersicht der Systemeigenschaften von Simba PNIO premium und Simit Unit PN256, Quelle: Siemens AG (Hrsg.) (07/2017), S. 58 (leicht modifiziert).

¹⁵⁰ Vgl. Siemens AG (Hrsg.) (07/2017), S. 79.

6.3 Netzwerktopologie und Netzwerkvarianten

Der Aufbau des HiL-Testkits und die darin verwendeten Komponenten zur Simulation des Profinet-Systems lassen unterschiedliche Netzwerkkonfigurationen und Netzwerksysteme zu. Diese Netzwerksysteme dienen zur Verbindung der in der VIBN eingesetzten Teilsysteme wie virtuelles Anlagenmodell, Speicherprogrammierbare Steuerung sowie Feldbusteilnehmer und Anlagenperipherie.

Aufgrund der geplanten Vorgehensweise zum Aufbau eines funktionierenden VIBN-Systems werden im Zuge des Praxisteils zwei Netzwerkkonfigurationen angewandt. Auf eine dritte Konfiguration wird verwiesen, da diese nach entsprechendem Know-How-Aufbau im Gebiet der Virtuellen Inbetriebnahme bei MSF die Möglichkeit bietet, Produktionsanlagen mit mehr als einem SPS-System abzubilden. Auch der Betrieb eines Profinet-Systems mit mehr als 256 Busteilnehmern respektive Feldgeräten wird dadurch ermöglicht. Ebenso ist an dieser Stelle festzuhalten, dass in weiterer Folge der Praxisarbeit ausschließlich das Hardware-Modul Simit Unit PN256 eingesetzt wird, da dieses über ein größeres Funktionsspektrum als die Komponente Simba PNIO premium verfügt.

6.3.1 Netzwerktopologie A

Netzwerktopologie A stellt den Idealfall hinsichtlich Betriebssicherheit und Datenübertragungsgeschwindigkeit dar. In diesem sind das Anlagennetz/Profinet und der Simulations-PC voneinander getrennt. Ebenso findet in diesem eine Trennung von Profinet-Netzwerk zu allen weiteren an der VIBN eingebundenen Netzwerken statt, wie es Abb. 50 verdeutlicht.¹⁵¹

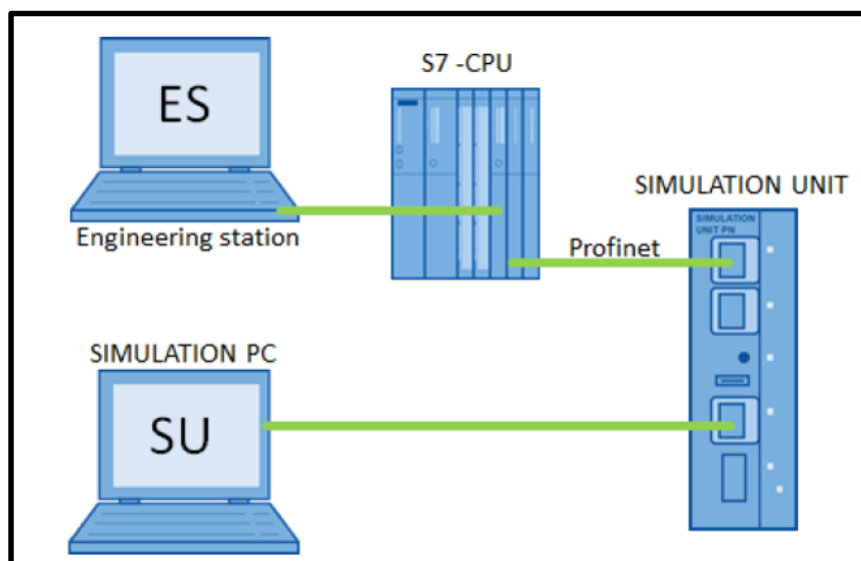


Abb. 50: Netzwerktopologie A des VIBN-Systems, Quelle: Siemens AG (Hrsg.) (07/2017), S. 70.

Auf der in Abb. 50 abgebildeten Engineering-Station werden die Werkzeuge und Softwaretools zur Entwicklung der SPS-Programme und Anlagensvisualisierung betrieben. Am dargestellten Simulations-PC wird die virtuelle Anlage abgebildet.

¹⁵¹ Vgl. Siemens AG (Hrsg.) (07/2017), S. 69.

6.3.2 Netzwerktopologie B

Trotz der Empfehlung durch den Hersteller der Simit Unit, Netzwerktopologie A im Gesamtverbund der Virtuellen Inbetriebnahme zu verwenden, wird nach Projektierung der Simit Unit im Praxisteil Netzwerkkonfiguration B verwendet. Diese Netzwerktopologie wird für die Phasen des Setup des VIBN-Systems, Modellierung der virtuellen Anlage, sowie Hand- und Automatikbetrieb der virtuellen Produktionsanlage beibehalten. Die vom Hersteller angeführten potentiellen Komplikationen wie Störung und Beeinträchtigung der I/O-Kommunikation auf Seiten des Profinet-Netzwerks konnten während des Praxisbetriebs nicht festgestellt werden.¹⁵² Eine mögliche Reduzierung der Kommunikationsgeschwindigkeit zwischen Simulations-PC und Simit Unit¹⁵³ wird bewusst in Kauf genommen, da das Forschungsinteresse der Grundfunktionalität der VIBN gilt. Abb. 51 stellt die gewählte Netzwerktopologie grafisch dar.

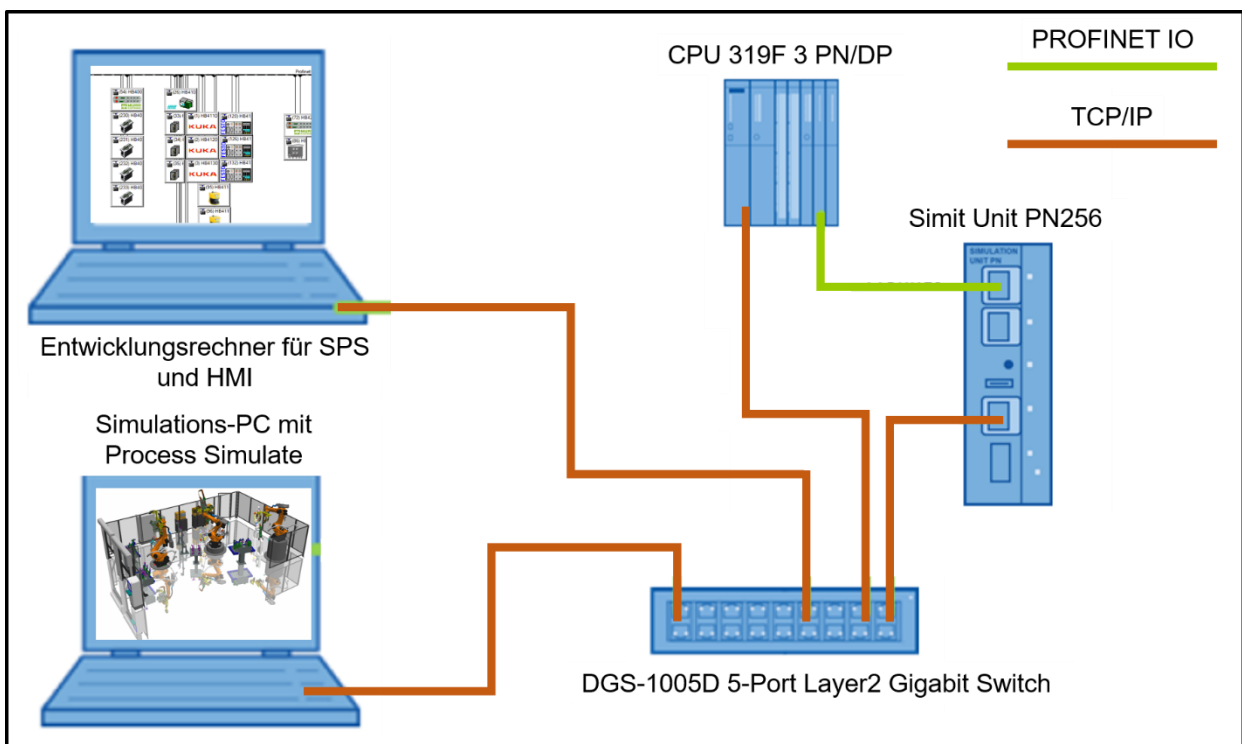


Abb. 51: Netzwerktopologie B des VIBN-Systems, Quelle: Eigene Darstellung.

Ebenfalls unterstützt die in Abb. 51 dargestellte Netzwerktopologie das parallele Arbeiten unterschiedlicher Fachdisziplinen wie SPS-Programmierung, Roboterprogrammierung, Prozessvisualisierung und Simulation. So ist es möglich, gleichzeitig Änderungen am SPS-Programm vorzunehmen, während Anpassungen am virtuellen Anlagenmodell vorgenommen werden. Auch paralleles Überwachen von Prozessinformationen ist dadurch sichergestellt. Zum Beispiel kann der Status vom selben Signal sowohl auf Seiten des Entwicklungsrechners für des SPS-Programm beobachtet werden, wie auch der Status desselben im virtuellen Anlagenmodell.

¹⁵² Vgl. Siemens AG (Hrsg.) (07/2017), S. 70.

¹⁵³ Vgl. Siemens AG (Hrsg.) (07/2017), S. 70.

6.3.3 Netzwerktopologie C

Netzwerktopologie C wird aufgrund der dadurch geschaffenen Möglichkeit zur Darstellung des Potentials der VIBN in dieser Arbeit ergänzend behandelt. Obwohl diese Topologie im Zuge der Forschungstätigkeiten im Praxisteil dieser Masterthesis nicht zur Anwendung kommt, ist sie geeignet, die Leistungsfähigkeit der Virtuellen Inbetriebnahme zu verdeutlichen. Abb. 52 erläutert in grafischer Form, dass eine Vermischung von realen Profinet-Feldgeräten mit simulierten Profinet Devices technisch möglich ist.¹⁵⁴

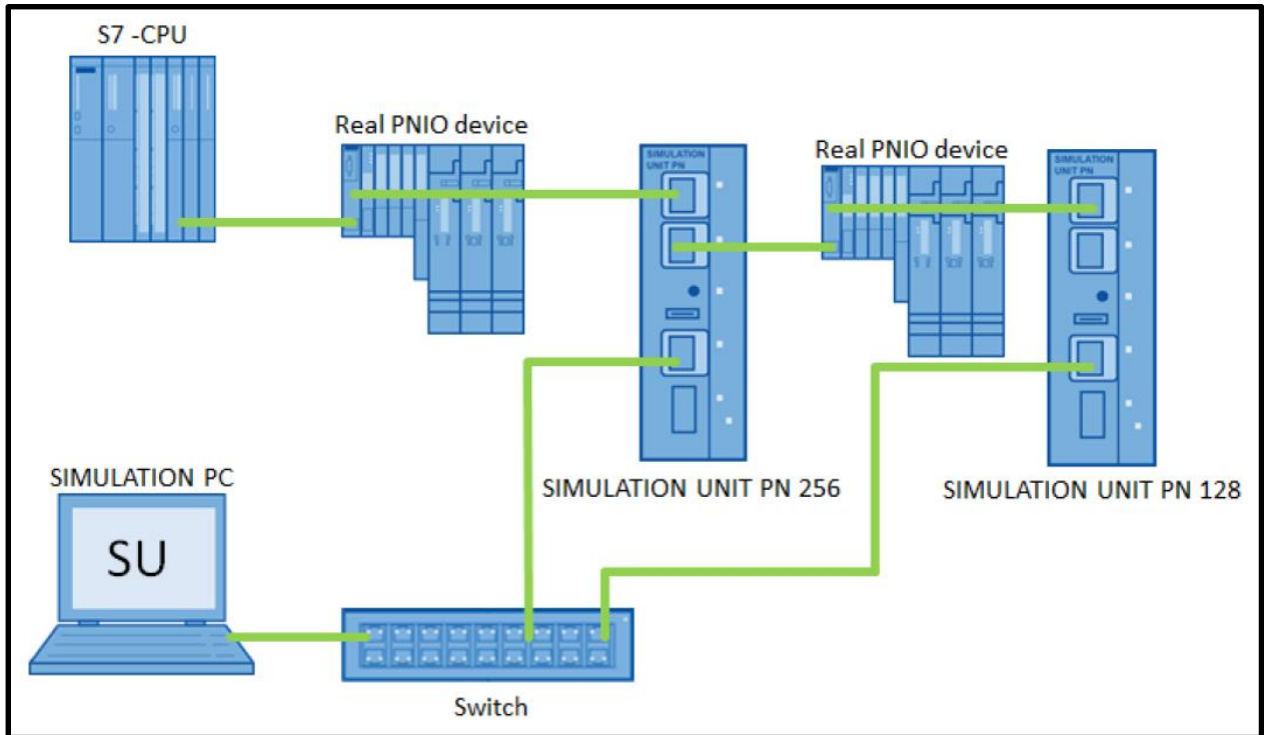


Abb. 52: Netzwerktopologie C eines VIBN-Systems, Quelle: Siemens AG (Hrsg.) (07/2017), S. 72.

Ebenfalls geht aus Abb. 52 die Möglichkeit hervor, mehrere Simulation Units durch einen geeigneten Switch mit einem Simulations-PC zu verbinden.¹⁵⁵ Dadurch wird ermöglicht, mehrere SPS-Systeme in einem virtuellen Anlagenmodell zum Zwecke der VIBN abzubilden. Auch können mittels dieser Netzwerktopologie mehrere Profinet-Stränge simuliert werden. Damit wird die Möglichkeit geschaffen, mehr als 256 Teilnehmer am Feldbussystem zu simulieren. Voraussetzung dafür ist die Verwendung der im HiL-Testkit verbauten Hardware-Komponenten zur Feldbussimulation. So verfügen ausschließlich die Modelle Si mit Unit PN256 und Simba PNIO premium über zwei aktive Ports zum Anschließen von realen Feldgeräten oder zum Kaskadieren von Simulation Units.¹⁵⁶

Nachdem in diesem Kapitel die erforderliche Infrastruktur zur Durchführung einer VIBN mittels Hardware-in-the-Loop Methodik erläutert und beschrieben wurde, befasst sich das nachfolgende Kapitel mit dem Aufbau des prototypischen VIBN-Modells.

¹⁵⁴ Vgl. Siemens AG (Hrsg.) (07/2017), S. 72.

¹⁵⁵ Vgl. Siemens AG (Hrsg.) (07/2017), S. 72.

¹⁵⁶ Vgl. Siemens AG (Hrsg.) (07/2017), S. 72.

7 AUFBAU UND MODELLIERUNG DES PROTOTYPUS ZUR VIRTUELLEN INBETRIEBNAHME

Kapitel 7 hat den Aufbau des Prototypen zur Durchführung der Virtuellen Inbetriebnahme zum Inhalt. Die VIBN wird auf Basis des Hardware-in-the-Loop Ansatzes durchgeführt. Die dafür erforderliche Infrastruktur wurde im vorangegangenen Kapitel erläutert und beschrieben. Inhalt der nachfolgenden Unterkapitel und Abschnitte sind System Setup und Projektierung der Simulation Unit zur Simulation des Feldbussystems. Im Anschluss daran wird auf den Kommunikationsaufbau zwischen den unterschiedlichen Teilsystemen im VIBN-Verbund eingegangen. Dieser repräsentiert die Grundlage und Voraussetzung eines fehlerfreien und funktionierenden Informationsaustausches zwischen HMI, SPS, Feldgeräten sowie virtuellem Produktionssystem. Im Anschluss daran werden die handelnden Personen respektive Funktionen und Fachdisziplinen erörtert, die entscheidende Rollen im Zuge eines VIBN-Projekts einnehmen. Danach wird eine Analyse der notwendigen Informations- und Datenaufbereitung durchgeführt, welche zur Bildung eines homogenen und holistischen VIBN-Modells vorausgesetzt wird. Im Zuge dessen wird auf die erforderliche Festlegung von Systemgrenzen für die in der VIBN beteiligten Anlagenmodelle und Kommunikationsnetzwerke eingegangen. Auch muss in diesem Zusammenhang eine Auseinandersetzung der historisch bedingten unterschiedlichen Entwicklungsmethoden von Mechanik und Elektrotechnik geführt werden. Speziell die unterschiedlichen Zugänge dieser beiden Ingenieursdisziplinen in Bezug auf Bezeichnungssystematik sowie Konstruktionsmethodik stellen gegenwärtig eine Herausforderung für einen ressourcenschonenden Aufbau des virtuellen Anlagenmodells dar. Im Anschluss daran erfolgt eine Erörterung der erforderlichen Modellierungsschritte, die zur Qualifizierung des übernommenen Simulationsmodells für die Virtuelle Inbetriebnahme notwendig sind. Insbesondere wird in diesem Zusammenhang auf die Modellbildung von Sensorik, Aktuatorik und Robotik eingegangen. Auch die Modellierung des Materialflusses zur Darstellung des Handbetriebs und Automatikbetriebs mit Bauteilen findet Berücksichtigung. Danach findet eine Bewertung des durchgeführten I/O-Checks statt, sowie die im Zuge der VIBN getesteten Inbetriebnahmeschritte wie Handbetrieb und Automatikbetrieb. Abschluss findet dieses Kapitel mit einer Diskussion über die erreichten Ziele sowie der gewonnenen Erkenntnisse, die während des VIBN-Betriebs generiert werden konnten.

Da der Fokus des Forschungsinteresses am Aufbau einer geeigneten VIBN-Methodik liegt, werden alle nachfolgenden Ausführungen an einem bestehenden und funktionierenden Produktionssystem getätigt.

7.1 Setup des VIBN-Systems

Unterkapitel 7.1 hat das Setup des VIBN-Systems zum Inhalt. In diesem werden die notwendige Schritte zum Kommunikationsaufbau und Vernetzung von HMI, SPS, Profinet-Feldgeräten sowie dem virtuellen Modell der Produktionsanlage dargelegt. Die dafür eingerichtete Netzwerktopologie wurde bereits in Abschnitt 6.3.2 sowie mittels Abb. 51 beschrieben. Grundlage des VIBN-Prototyps bildet der Betriebsartenbereich 1 des Anlagenbereichs TH03RHB4, kurz HB4-BA1. An diesem werden im Weiteren alle Arbeiten vorgenommen, die zum Aufbau und Anwendung einer geeigneten VIBN-Lösung für den Karosseriebau bei Magna Steyr führen. HB4-BA1 dient als Forschungsgrundlage, da dieser Bereich einen Großteil der bei MSF eingesetzten Anlagen- und Applikationstechnik beinhaltet und daher als Referenzanlage geeignet ist.

7.1.1 Datenbereitstellung aus realem SPS-System

In der Hardwarekonfiguration des realen SPS-Projekts *TH03RHB4* des Anlagenbereichs HB4 ist das angewandte Profinet IO-Netzwerk projektiert. Diese dient als Grundlage für den weiteren Kommunikationsaufbau innerhalb des VIBN-Systems. Befinden sich die Entwicklungssoftware für SPS sowie die Projektierungssoftware für die Simit Unit PN256 am selben Rechner, sind keine weiteren Installationstätigkeiten am Entwicklungsrechner vorzunehmen.¹⁵⁷ Nach Ausführen des Befehls *Speichern und übersetzen* steht das im SPS-System projektierte Profinet-Netzwerk in Form von Systemdatenbausteinen (sdb-Dateien) zur weiteren Verwendung zur Verfügung,¹⁵⁸ wie dies ergänzend aus Abb. 53 ersichtlich wird.

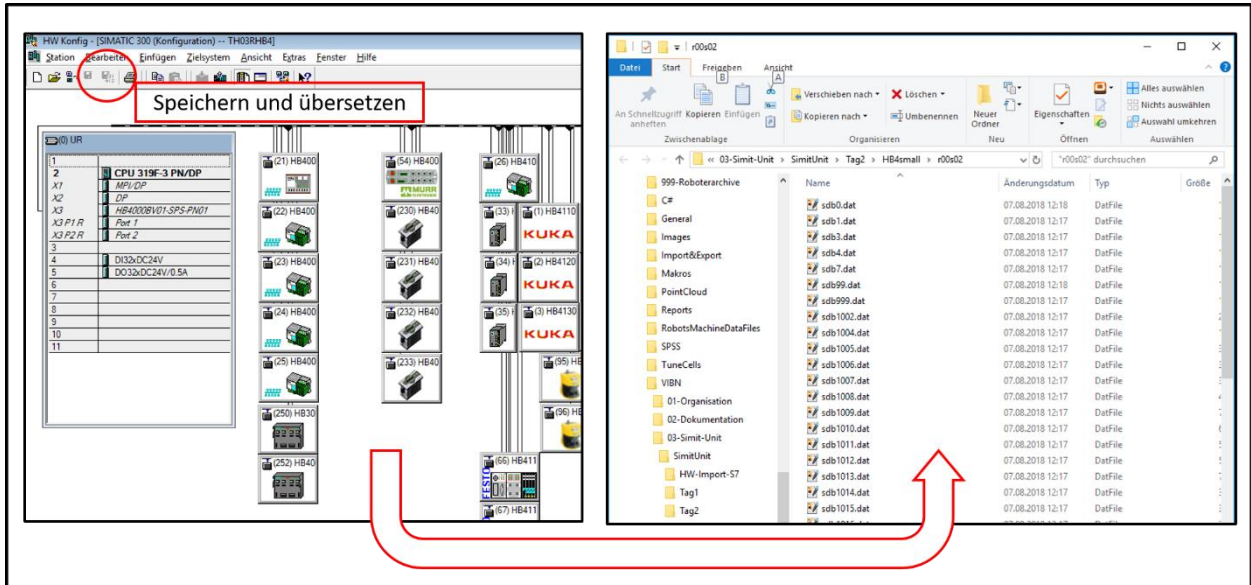


Abb. 53: Export der Hardwarekonfiguration aus SPS-System, Quelle: Eigene Darstellung.

Die durch das Speichern und Übersetzen der Hardwarekonfiguration (HW-Konfiguration) aus dem SPS-System generierten Dateien stellen wiederum die Basis zur Erstellung der zu simulierenden Profinet-Feldgeräte mittels Simit Unit PN256 dar. Zu finden sind diese unter dem Dateipfad `<STEP7>/S7tmp/SDBData/program/DOWN/rXXsXX`.¹⁵⁹

Im Falle einer echten Virtuellen Inbetriebnahme wird an dieser Stelle der Datenbereitstellung empfohlen, Revisionsordner für den jeweiligen Speichervorgang der HW-Konfiguration anzulegen. Dies ermöglicht ein geordnetes und strukturiertes Verwalten der sdb-Dateien bei nachträglicher Änderungen der Feldbustopologie im SPS-System.

Ergänzend ist die Hardwarekonfiguration der SPS als *.cfg Datei zu exportieren, da diese die Übertragung der gewählten Symbolik von SPS zu Simit Unit erlaubt.¹⁶⁰

¹⁵⁷ Vgl. Siemens AG (Hrsg.) (07/2017), S. 14.

¹⁵⁸ Vgl. Siemens AG (Hrsg.) (07/2017), S. 42.

¹⁵⁹ Vgl. Siemens AG (Hrsg.) (07/2017), S. 42.

¹⁶⁰ Vgl. Siemens AG (Hrsg.) (07/2017), S. 43.

7.1.2 Projektierung Simit Unit PN256

Nach erfolgter Bereitstellung der notwendigen Systemdatenbausteine aus dem realen SPS-System, kann mit der Projektierung der Simit Unit begonnen werden. Die bereits installierte Projektierungssoftware SIMULATIONUnit (Version 9.1 Upd1, Release 09.01.00.01_00.00.00.06) ist am Entwicklungsrechner für SPS und HMI zu öffnen. Zu Beginn ist die zu projektierende Simit Unit PN256 im VIBN-Netzwerk aufzufinden. Dafür steht die Funktion *Scan network* in der Projektierungssoftware zur Verfügung. Nach detektierter und aufgefundener Simit Unit muss dieser eine geeignete IP-Adresse und Subnetzmaske zugewiesen werden. In diesem Zusammenhang ist zu betonen, dass die Simit Unit und die zu simulierenden Profinet IO-Devices nicht im selben IP-Subnetz adressiert sein dürfen.¹⁶¹ Wird dies nicht beachtet, so kommt es zu teilweisen Ausfällen einzelner Feldgeräte während der Feldbussimulation beziehungsweise der gesamte Profinet-Strang ist nicht simulierbar. Abb. 54 gibt eine Übersicht der angewendeten und funktionierenden Netzwerkkonfiguration. Diese ermöglicht eine erfolgreiche Projektierung der Simit Unit PN256 zum Zwecke des Setups des VIBN-Systems.

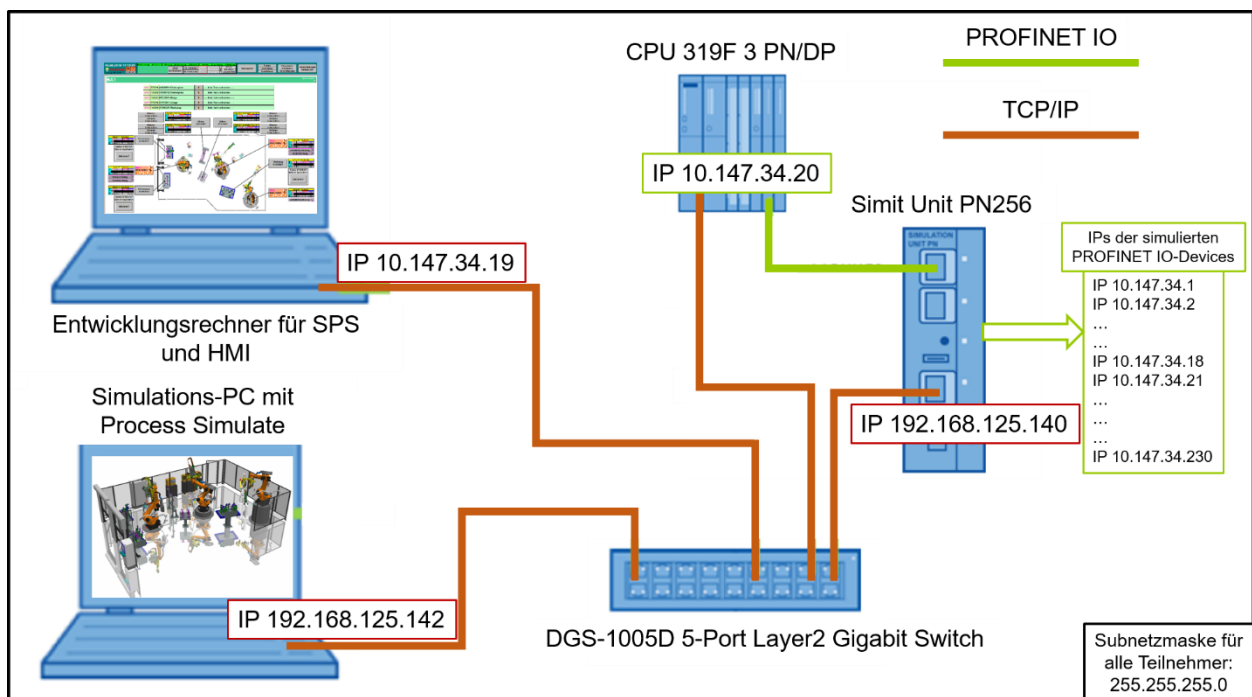


Abb. 54: IP-Adressierung des VIBN-Netzwerkes während Projektierung der Simit Unit, Quelle: Eigene Darstellung.

Anschließend erfolgt der Import der zur simulierenden Profinet-Feldgeräte in die Projektierungssoftware durch Einlesen des Systemdatenbausteins *sdb3000.dat*. Alle weiteren *sdb*-Dateien haben im Zusammenhang mit der VIBN keine Relevanz und werden für die weitere Forschungstätigkeit nicht weiterführend betrachtet.

¹⁶¹ Vgl. Siemens AG (Hrsg.) (07/2017), S. 17.

Nach erfolgreichem Import sind die zu simulierenden Profinet-Feldgeräte in der Hardwareansicht der Projektierungssoftware auffindbar. Entsprechend ihrer IP-Adresse sind diese in aufsteigender Reihenfolge sortiert, wie aus Abb. 55 ersichtlich.

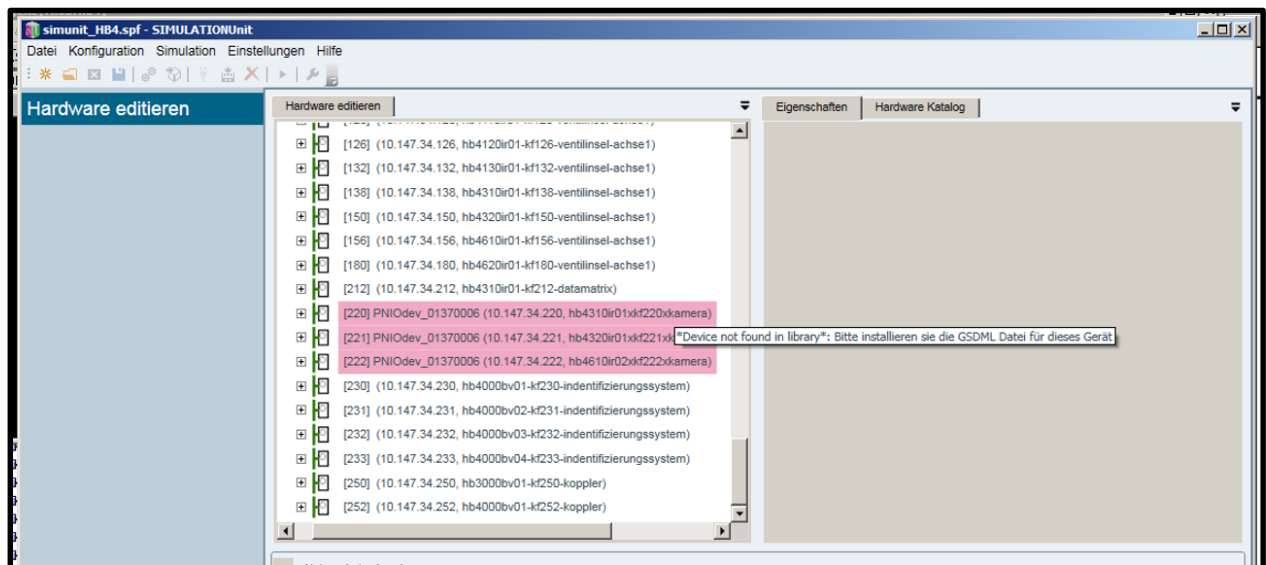


Abb. 55: Auszug des Importergebnisses der Profinet-Feldgeräte des SPS-Systems TH3RHB04, Quelle: Eigene Darstellung.

Werden Profinet-Devices, wie in Abb. 55 verdeutlicht, rot hervorgehoben, so stehen für diese Feldgeräte keine gültigen Gerätestammdateien (GSDML) zur Weiterverwendung in der Simit Unit zur Verfügung. Diese müssen vom jeweiligen Gerätehersteller beschafft und nachimportiert werden. Sind diese Mängel beseitigt, kann die Simulation der Profinet-Devices durch Ausführen der Funktionen *Connect all* und *Start run time system* gestartet werden. Werden alle Profinet-Devices nachfolgend mittels grünem Quadrat hinterlegt, so ist die Simulation aller Feldgeräte fehlerfrei aktiv und ein I/O-Datenaustausch vorhanden.¹⁶² Eine detaillierte Beschreibung der Gerätesymbolik und Farbcodierung ist dem Handbuch *SIMATIC- SIMULATIONUnit V9.1* zu entnehmen. Ist sichergestellt, dass alle notwendigen Profinet-Feldgeräte fehlerfrei simuliert werden, darf die Projektierung der Simit Unit vom Entwicklungsrechner auf die Simit Unit PN256 übertragen werden. Dafür ist der Befehl *Download all* auszuführen. Wird dies durch eine positive Rückmeldung aus dem System-Log der Projektierungssoftware bestätigt, muss das Simit Unit-Projekt mittels des Befehls *Save project as* abgespeichert werden. Im Zuge dessen wird der Dateiordner *simitunit_HB4.spf* zur weiteren Verwendung im Setup des VIBN-Systems erzeugt. Sowohl für den Downloadvorgang vom Entwicklungsrechner auf die Simit Unit PN256, wie auch für das Erstellen des Dateiordners *simitunit_HB4.spf* ist die Spracheinstellung in der Projektierungssoftware entscheidend. Für beide Vorgänge muss sichergestellt sein, dass diese in englischer Spracheinstellung ausgeführt werden. Wird dies nicht berücksichtigt, so ist in weiterer Folge ein Kommunikationsaufbau und Informationsaustausch zur virtuellen Produktionsanlage nicht gegeben, welche in Process Simulate 14.1 abgebildet wird. Sind beide Vorgänge in der richtigen Spracheinstellung abgeschlossen, muss die Projektierungssoftware *SIMULATIONUnit V9.1* geschlossen werden. Andernfalls verhindert die geöffnete Software den Datenverkehr zwischen PS 14.1 und Simit Unit PN256.

¹⁶² Vgl. Siemens AG (Hrsg.) (07/2017), S. 26.

7.1.3 Systemvorbereitung Process Simulate 14.1

Der letzte Schritt im Zuge des System Setups zur Durchführung der Virtuellen Inbetriebnahme sind die erforderlichen Vorbereitungsaktivitäten auf Seiten der Simulationssoftware, die das virtuelle Anlagenmodell abbildet. Für den verfolgten Forschungszweck im Rahmen dieser Masterthesis findet die Software Process Simulate, Version 14.1, Anwendung. Systemeigenschaften und Funktionsportfolio der Software wurden bereits in Unterkapitel 4.3 behandelt. An dieser Stelle sind ergänzend die dafür nötigen Softwarelizenzen zu betonen. Ohne diese ist die Nutzung der erforderlichen Funktionalitäten in PS 14.1 nicht möglich. Eine Übersicht der notwendigen Lizenzen gibt die Datei *License_Ugslmd-20180830.txt*, die dem Anhang beigelegt ist. Insbesondere die darin angeführte Lizenz *FEA_PS_COMMISSIONING ugslmd 14.0* ist hervorzuheben, da diese alle Funktionen innerhalb von Process Simulate freigibt, die zur Durchführung der VIBN vorausgesetzt werden.

Sind Process Simulate 14.1 und alle relevanten Zusatzmodule am Simulationsrechner installiert, sowie die Lizenzvoraussetzungen gegeben, sind die Voraussetzungen für das Setup zur Virtuellen Inbetriebnahme erfüllt.

Zu Beginn ist das gesamte Installationsverzeichnis der Projektierungssoftware SIMULATIONUnit 9.1 Upd1 in das Process Simulate Verzeichnis `<#\Program Files\Tecnomatix_Version_xx\MPower\SIMBAPro>` zu übertragen. Im Zuge des Setups von Process Simulate 14.1 erfolgt die Übergabe der geforderten Inhalte vom Installationsverzeichnis `<C:\Program Files\SIMULATIONUnit_V9.1Upd1>` nach `<C:\Program Files\Tecnomatix_Version_14.1\MPower\SIMBAPro>`. Damit wird sichergestellt, dass PS alle erforderlichen Informationen über die im VIBN-System eingesetzte Simulation Unit erhält. Dazu zählt beispielsweise eingesetzte Hardware und angewandter Firmwarestand. Im konkreten Anwendungsfall ist dies Simit Unit PN256 mit Firmware der Version 2.1.00.14. Der Kopiervorgang ist parallel zu einer gestarteten Simulation in Process Simulate möglich. Ist dies der Fall, so ist ein Neustart von PS 14.1 nach Abschluss der Dateiübertragung nötig. Nach erfolgtem Start respektive Neustart von PS 14.1 kann mit dem Setup fortgefahren werden.

Zur Durchführung der VIBN wird die HiL-Methodik eingesetzt, dahingehend sind in Process Simulate die in Abb. 56 hervorgehobenen Verbindungseinstellungen vorzunehmen. Werden diese nicht vorgenommen, so befindet sich PS im MiL-Modus, welche die Grundeinstellung darstellt. Alternativ besteht die Möglichkeit, Process Simulate im Modus der Systemsimulation zu betreiben, respektive Software-in-the-Loop.

Sind die Einstellungen im Optionsmenü von PS 14.1 für den Simulationsbetrieb mittels Hardware-in-the-Loop Modus richtig gewählt, kann mit dem Setup fortgefahren werden. Als nächstes ist die Funktion *Connection Setting* auszuführen, welche nach Setzen der richtigen HiL-Einstellungen aktiv wird. Die Funktion *Connection Setting* ist ebenfalls in Abb. 56 hervorgehoben. Durch das Ausführen der Funktion wird ein zusätzliches Menüfenster geöffnet, das durch Abb. 57 grafisch dargestellt wird. In diesem sind weitere Verbindungsoptionen vorzunehmen, die für den geforderten Simulationsmodus notwendig sind. Durch wählen der Funktion *Add...* öffnet sich ein Popup-Fenster, in dem die Option *Simulation Unit* auszuwählen ist, wie es in Abb. 57 in bildlicher Form hervorgehoben wird.

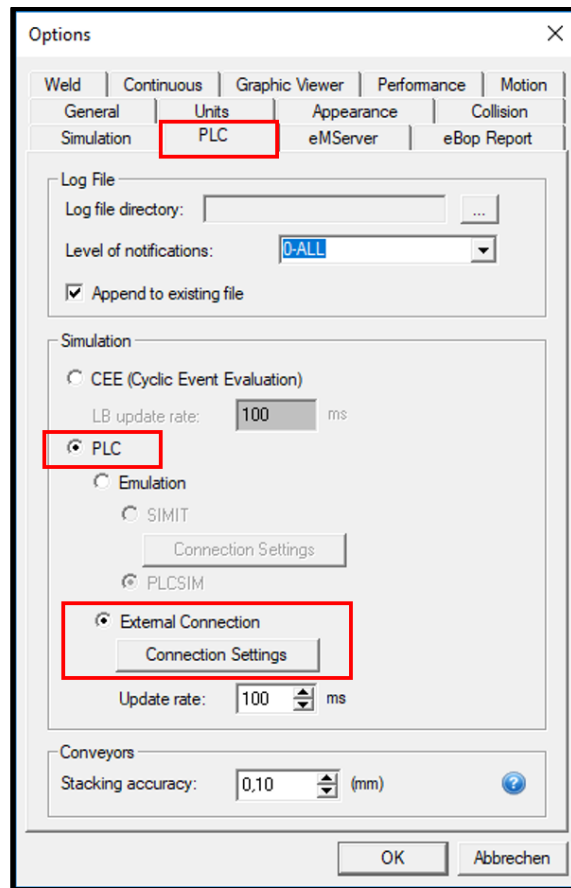


Abb. 56: Erforderliche Einstellwerte zur Einbindung von PS in ein HiL-System, Quelle: Eigene Darstellung.

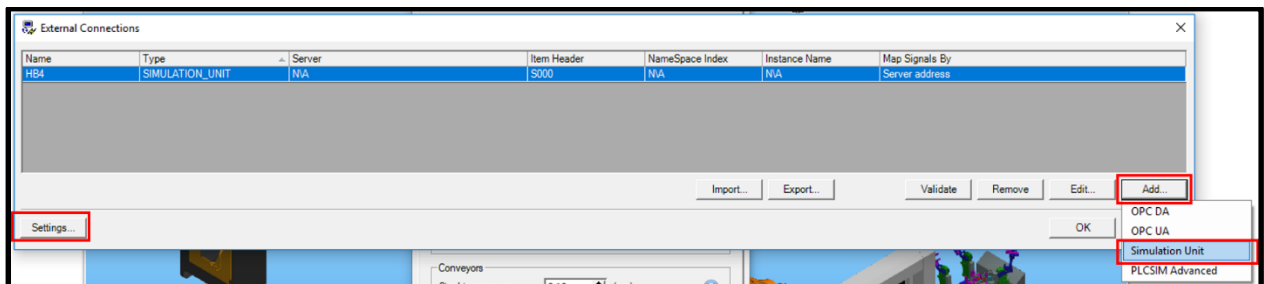


Abb. 57: Konfiguration der Kommunikationseinstellung für den HiL-Betrieb durch Einbindung der Simulation Unit, Quelle: Eigene Darstellung.

Dies ermöglicht in weiterer Folge, die Kommunikationseinstellungen für die im VIBN-System eingebundene Simulation Unit zu konkretisieren. Da im Rahmen der Masterthesis der Anlagenbereich TH03RHB4 Basis der Forschungstätigkeit ist, wird der Kommunikationskanal *HB4* benannt sowie der Hardwareindex für diesen mit dem Wert Null parametrisiert, wie dies ergänzend in Abb. 58 dargestellt wird.

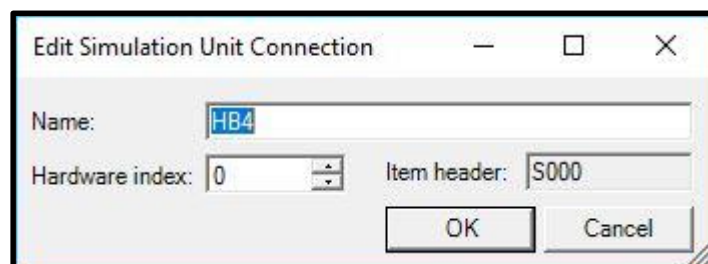


Abb. 58: Einstellwerte zum Kommunikationsaufbau zwischen PS 14.1 und Simulation Unit, Quelle: Eigene Darstellung.

Der in Abb. 58 eingestellte Hardwareindex darf im Falle dieser Systemkonfiguration den Wert Null einnehmen, da nur eine Simulation Unit im VIBN-System eingebunden ist. Welche Hardwareindizes bei Verwendung mehrere Simulation Units einzustellen sind, konnte zum Zeitpunkt der Erstellung dieser Arbeit nicht ermittelt werden. Als letzter Schritt im Zuge des Setups auf Seiten von Process Simulate sind jene Einstellungen vorzunehmen, die durch Ausführen der Funktion *Setting...* zu parametrieren sind. Diese Funktion ist ebenfalls im Menüfenster *Connection Settings* vorzufinden, die in Abb. 57 dargestellt ist. Durch Betätigen des Befehls *Settings...* öffnet sich ein weiteres Eingabefenster. In diesem ist der Ablageort des zuvor erstellten und abgespeicherten Simit Unit Projekt bekanntzugeben, wie in Abb. 59 dargestellt.

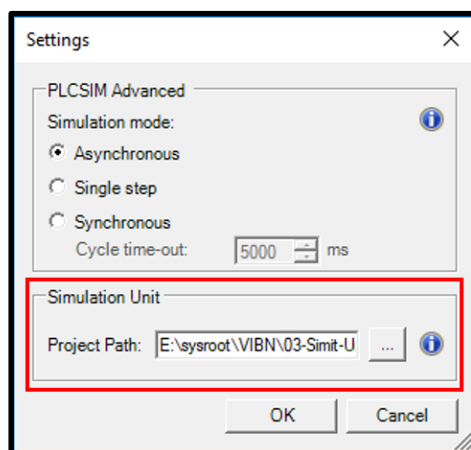


Abb. 59: Bekanntgabe des Ablageorts der Simit Unit Projektierung für Process Simulate, Quelle: Eigene Darstellung.

Unter *Project Path* sind die im Rahmen der Projektierung der Simit Unit PN256 erzeugten Systemkonfigurationen sowie die generierten Projektdateien bekanntzugeben. Diese sind im Ordner *simitunit_HB4.spf* abgespeichert. Im Zuge des Setups für diese Arbeit ist der Ordner unter `<E:\sysroot\VIBN\03-Simit-Unit\SimitUnit\Tag2\HB4all\HB4all.spf>` am Simulationsrechner abgelegt. Die darin befindlichen Dateien enthalten alle nötigen und relevanten Informationen über das zuvor projektierte Profinet System. Aus diesen wird für Process Simulate ersichtlich, welche Topologie das Feldbus-System aufweist. Ebenso wird darin die Art und Anzahl der eingebunden Feldgeräte PS 14.1 mitgeteilt. Auch werden darin die eingesetzten Profinet IO- und Profisafe-Protokolle Process Simulate bekanntgegeben, das im realen SPS-System eingesetzt wird und von der Simulation Unit zur simulieren ist. Ist der letzte Parametrierungsschritt zur Einstellung der Kommunikationsparameter zwischen Process Simulate und der eingesetzten Simulation Unit vollzogen, ist dieser durch Anstoß der Funktion *Validate* zu prüfen.

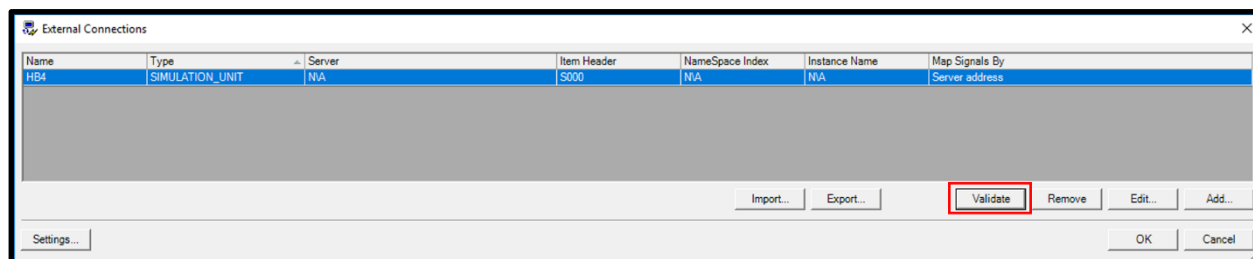


Abb. 60: Validierung fehlerfreier Kommunikation zwischen Process Simulate und Simulation Unit, Quelle: Eigene Darstellung.

Abschließend hebt Abb. 60 dies hervor. Nach positiver Bestätigung von Kommunikationsaufbau und Informationsaustausch kann mit der Durchführung der Virtuellen Inbetriebnahme begonnen werden.

7.2 Teambzusammensetzung und Rollenverteilung

Nach Aufbau der erforderlichen Infrastruktur sowie des System-Setups der im VIBN-System eingebundenen Komponenten wird mit der Durchführung der Virtuellen Inbetriebnahme begonnen. Wie bereits in Abschnitt 2.5.2 und in Unterkapitel 2.10 analysiert und erarbeitet wurde, spielt die Zusammensetzung des Entwicklungsteams im Rahmen des Modellbildungsprozesses eine bedeutende Rolle. Die Zusammensetzung der an der Modellbildung beteiligten Personen beeinflusst maßgeblich den Erfolg des zu bildenden Modells. Dieser Anforderung wird auch im Zuge der Durchführung der VIBN Rechnung getragen. Einerseits ist im Rahmen der Erarbeitung des VIBN-Prototypen ein hohes Maß an Modellbildung nötig, andererseits erfordert die Durchführung der VIBN die Beteiligung mehrere Ingenieursdisziplinen. Am Aufbau des nachfolgend beschriebenen VIBN-Prototypen sind dahingehend die in Abb. 61 angeführten Rollen respektive Disziplinen beteiligt.

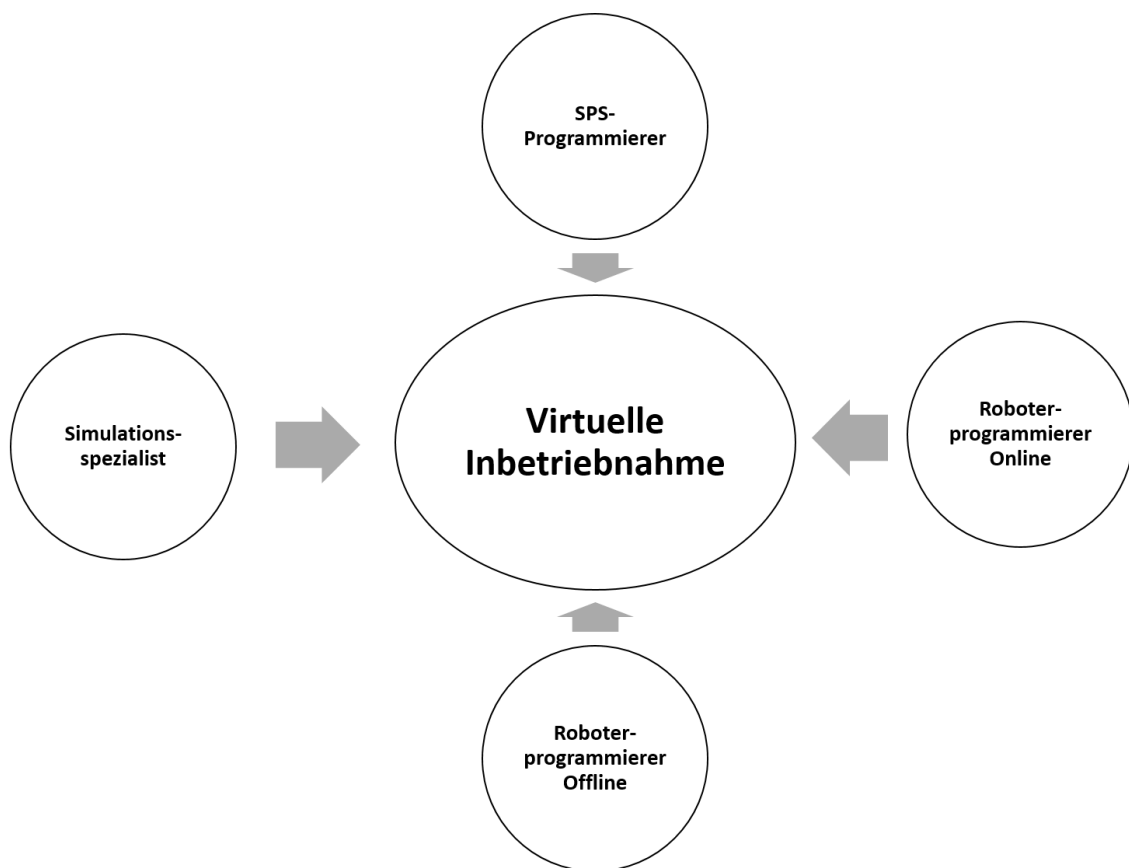


Abb. 61: Teambzusammensetzung im Rahmen des Aufbaus des VIBN-Prototypen, Quelle: Eigene Darstellung.

Der SPS-Programmierer stellt alle relevanten Informationen und Daten aus dem SPS-System bereit, die für den Aufbau und Betrieb des VIBN-Prototypen notwendig sind. Des Weiteren bedient dieser die HMI des Anlagenbereichs HB4-BA1. Im Rahmen der Entwicklungstätigkeit erweitert dieser die bestehenden Programmbausteine auf Seiten des SPS-Systems. Ebenfalls analysiert der SPS-Programmierer die Informationen und Daten, welche von der virtuellen Produktionsanlage an das SPS-System übermittelt werden. Er analysiert im Hand- und Automatikbetrieb das Prozessabbild in der Speicherprogrammierbaren Steuerung. Dies führt er durch Anwendung der Analysewerkzeuge durch, die am Entwicklungsrechner für HMI und SPS installiert sind und über die SPS mit dem VIBN-System verbunden sind.

Der Roboterprogrammierer mit Schwerpunkt Onlineprogrammierung hat ebenso die Aufgabe, alle erforderlichen Informationen und Daten dem VIBN-Prototypen zur Verfügung zu stellen, die im Zuge der Virtuellen Inbetriebnahme von Seiten der Roboterprogrammierung nötig sind. Er ist im Rahmen der Erstellung des Prototypen für die Analyse des Roboter-codes verantwortlich. Diese stehen entweder in Interaktion mit Komponenten des virtuellen Anlagenmodells oder im Austausch mit dem SPS-System. Ebenfalls hat er die Aufgabe, notwendige Korrekturen an Codeteilen der Roboterprogramme vorzunehmen, falls diese mit dem VIBN-System nicht kompatibel sind. Auch hat er im Rahmen der VIBN die Aufgabe, Optimierungspotentiale im Segment der Onlineprogrammierung zu prüfen. Sind Programmteile nur bedingt zur Einbindung in die VIBN tauglich, so sind diese durch ihn zu dokumentieren und anzupassen. Aus den gewonnenen Erkenntnissen wird der Programmierstandard zur Spezifizierung der Onlineprogrammierung dahingehend angepasst, so dass dieser allen Anforderungen gerecht wird, welcher der Produktlebenszyklus eines Roboterprogramms im Karosseriebau bei Magna Steyr erfordert.

Der Roboterprogrammierer mit Schwerpunkt Offlineprogrammierung bereitet die virtuellen Robotermodelle soweit vor, dass diese im Rahmen der VIBN benutzt werden können. Ebenso hat er die Aufgabe, alle Bewegungsprogramme dem Simulationsmodell zur Verfügung zu stellen, die der jeweilige virtuelle Roboter im Rahmen seiner Fertigungsaufgaben abzuarbeiten hat. Diese müssen zum Startzeitpunkt der VIBN fehlerfrei abfahrbar und aufrufbar sein. Während der Entwicklung des VIBN-Prototypen hat er die Aufgabe, mögliche Potentiale beim Modellaufbau zu erkennen. Auch hat er den Auftrag, mangelhafte Funktionalitäten innerhalb des virtuellen Robotermodells zu beseitigen, die im Zusammenhang mit der Virtuellen Inbetriebnahme stehen. Werden diese evident, so hat er jene Programmschnittstellen anzupassen und zu verbessern, welche den bidirektionalen Informationstransfer zwischen virtuellem Roboter, virtuellem Anlagenmodell und SPS sicherstellt.

Der Simulationsspezialist hat die Aufgabe das Simulationsmodell so vorzubereiten beziehungsweise zu übernehmen, dass ohne Verzögerung mit dem Modellaufbau respektive Modellerweiterung zur Durchführung der VIBN begonnen werden kann. Durch den bei MSF definierten Produktentstehungsprozess wird dies im Normalfall sichergestellt. Wie in Abb. 44, Unterkapitel 4.4 beschrieben, endet die virtuelle Absicherung mit Abschluss der Offlineprogrammierung und im Idealfall mit Beendigung der ereignisbasierenden Simulation. Der Simulationsspezialist hat die virtuelle Produktionsanlage dahingehend weiter zu entwickeln, dass sämtliche Modellvoraussetzungen zur Durchführung der VIBN erfüllt sind. So hat er die Modelle zur Abbildung von Sensorik und Aktuatorik zu entwickeln und den Verwendungsstellen innerhalb des virtuellen Anlagenmodells zuzuweisen. Er hat die Programmstruktur im Simulationssystem aufzubauen, damit Roboter, Vorrichtungstechnik, Sicherheitstechnik sowie Peripherie über die Simulation Unit mit dem SPS-System fehlerfrei interagiert. Auch hat er die Aufgabe, alle relevanten Daten und Informationen ins Simulationssystem zu importieren, die von der SPS-Programmierung und der Roboterprogrammierung zur Durchführung der VIBN zur Verfügung gestellt werden. Diese Vorbereitungstätigkeiten stellen den Inhalt des nachfolgenden Unterkapitels dar.

7.3 Grundlagen der Signalaufbereitung für Process Simulate

Unterkapitel 7.3 setzt sich mit der nötigen Datenaufbereitung aus SPS-System und Industrieroboter auseinander. Diese ist notwendig, da einerseits ohne Einhaltung vorgegebener Formvorschriften ein Import in das Simulationssystem respektive virtuelle Anlagenmodell nicht möglich ist. Andererseits bietet der Schritt der Datenaufbereitung die Handhabe, effiziente und zeitschonende Aufbereitungsmethoden zu entwickeln. Mit diesen wird nach erfolgter Entwicklung eine weitestgehend automatisierte Aufbereitung und automatischer Import von Projektierungsinformationen aus SPS und Roboter möglich.

Basierend auf dem in Unterkapitel 7.1 dargelegten Setup des VIBN-Systems ist die Vernetzung der an der VIBN beteiligten Hauptkomponenten abgeschlossen. Die Kommunikation zwischen realer HMI, Real-SPS, Feldgeräten des Profinet IO-Netzwerkes sowie der virtuellen Produktionsanlage ist aufgebaut und funktioniert fehlerlos. Abb. 62 stellt den Verlauf der aufgebauten Kommunikation zwischen den Hauptkomponenten des VIBN-System symbolisch dar.

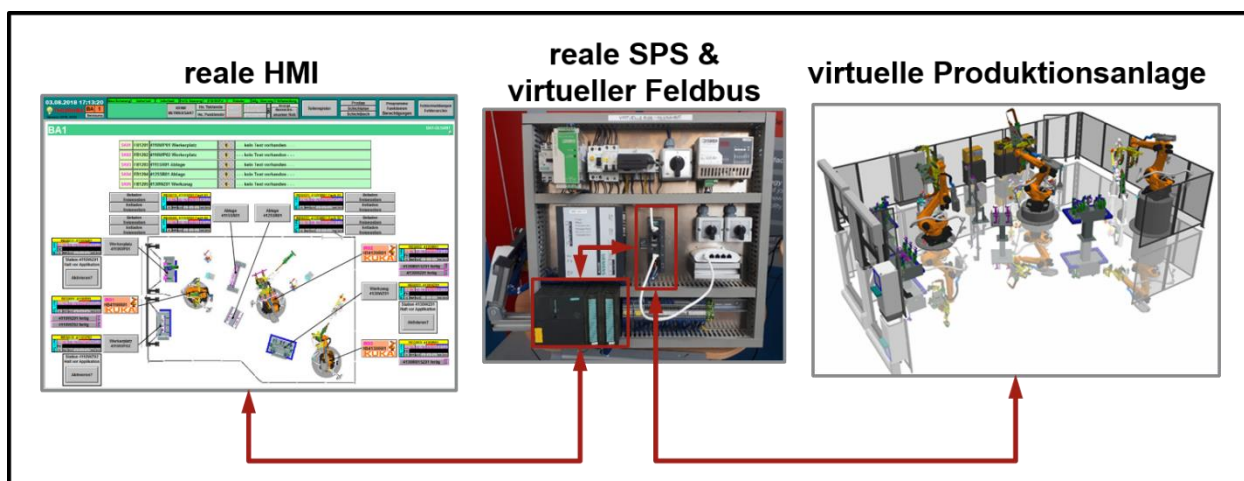


Abb. 62: Kommunikationsverlauf zwischen den Hauptkomponenten des VIBN-Systems, Quelle: Eigene Darstellung.

Ebenso wird aus Abb. 62 ersichtlich, dass die virtuelle Produktionsanlage, die in Process Simulate 14.1 abgebildet ist, über die Simit Unit PN256 mit der Real-SPS kommuniziert. Das heißt, dass die Feldgeräte des Profinet IO-Netzwerkes ebenfalls mit ihren Äquivalenten in Process Simulate in Verbindung stehen und deren Signale entsprechend auf den Feldbus schreiben beziehungsweise von diesem auslesen. Damit die in der virtuellen Produktionsanlage abgebildeten Feldgeräte wie Roboter, Ventilinseln, Schutzeinrichtungen und so weiter im Profinet IO-Netzwerk richtig adressiert werden können, sind eindeutige Signalbezeichnungen in PS 14.1 Voraussetzung. Diese stellen sicher, dass Ein- und Ausgangssignale der in Process Simulate abgebildeten Feldbusteilnehmer respektive Profinet IO-Devices richtig adressiert werden. Diese Adressierung muss so erfolgen, dass ein Zusammenfinden zwischen virtuellem Feldgerät und dessen realem Gegenüber sichergestellt wird. Aus diesem Grund muss ein Signalmapping zwischen SPS, Profinet IO-System und Process Simulate erfolgen, welches nachfolgend eingehend beschrieben wird.

Für das weitere Verständnis bezüglich der Vorgehensweise während der Datenaufbereitung ist die richtige Interpretation der Datenrichtung von Relevanz. Im Simulationswerkzeug zur Darstellung der virtuellen Produktionsanlage ist stets die Datenrichtung aus Sichtweise der SPS maßgebend.

Dies spielt beim Import von Prozessdatenschnittstellen von Industrierobotern eine wichtige Rolle, da die Steuerung der Roboter durch ein eigenes Programmmodul innerhalb des Simulationssystems abgebildet wird.

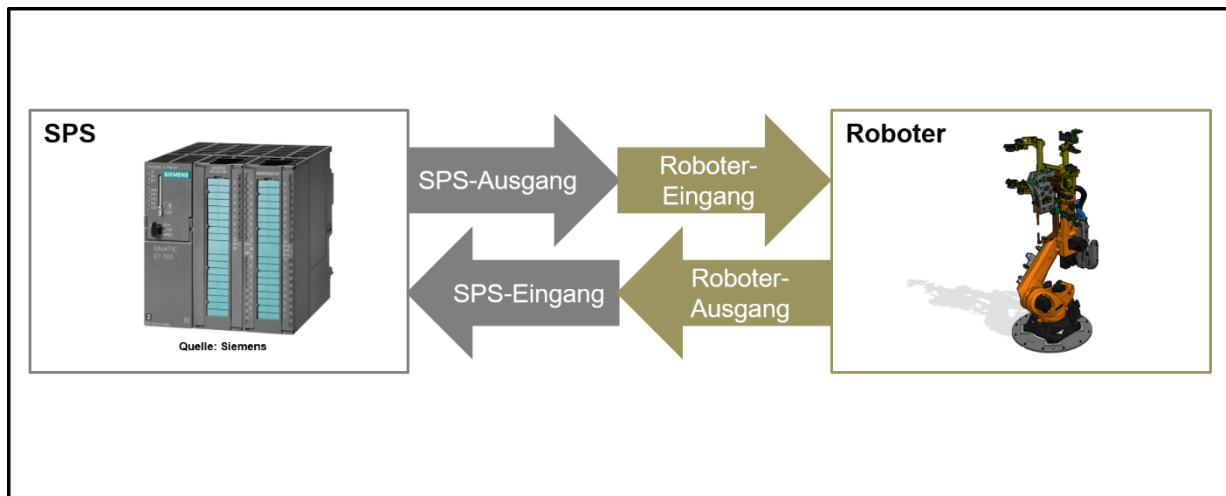


Abb. 63: Interpretation der Datenrichtung von Eingangs- und Ausgangssignalen im virtuellen Anlagenmodell, Quelle: Eigene Darstellung.

Wie in Abb. 63 verdeutlicht, sind dementsprechend bei der Festlegung der Roboterschnittstelle Eingangssignale der Roboter als SPS-Ausgangssignale zu deklarieren und umgekehrt. Diese Direktive gilt im Rahmen der Daten- respektive Signalaufbereitung für alle weiteren Betriebsmittel der virtuellen Produktionsanlage, die ebenso über eine eigene Steuerung und ein eigenes Feldbussystem verfügen. Für alle Feldgeräte beziehungsweise Profinet IO-Devices, die direkt am Feldbussystem der Speicherprogrammierbaren Steuerung angebunden sind, gilt diese Konvention nicht.

Damit mit der Signalaufbereitung für Process Simulate begonnen werden kann, sind die dafür erforderlichen Quelldaten bereitzustellen. Diese werden in den anschließenden Unterkapiteln erörtert.

7.4 Datenbereitstellung aus realem SPS-System

Unterkapitel 7.4 hat die Signalbereitstellung des SPS-Systems nach Process Simulate zum Inhalt, in dem das virtuelle Anlagenmodell abgebildet wird. Nach Aufbau und Sicherstellung der fehlerfreien Kommunikation zwischen Speicherprogrammierbarer Steuerung, Profinet IO-System und PS 14.1 ist dies der nächste Schritt im Rahmen der Datenaufbereitung. Im Zuge dessen werden alle Eingangs- und Ausgangssignale des SPS-Systems in das Simulationsmodell zur Abbildung der virtuellen Produktionsanlage importiert. Dieser Import beinhaltet alle Signale, die zum Steuern, Regeln und Überwachen von Sicherheitseinrichtungen sowie Bedien- und Fertigungskomponenten erforderlich sind. Auf die Aufbereitung des Signalimports für Industrieroboter wird gesondert in Abschnitt 7.4 eingegangen. Auf Seiten der Sicherheitseinrichtungen sind dies zum Beispiel Not Halt-Einrichtungen, Zutrittsüberwachungen von Schutztüren, Laserscanner zur Überprüfung von Hintertrittschutzbereichen und Kollaborationsräumen zwischen Mensch und Roboter sowie Werkerschutztore. Unter dem Sammelbegriff Bedien- und Fertigungskomponenten sind Taster zur Anlagenbedienung und Leuchteinrichtungen zum Wiedergeben von Anlagenzuständen zusammengefasst.

Ebenfalls zählen Sensoren zur Endlagenkontrolle von Stellgliedern dazu sowie Pneumatikventile zum Durchschalten von Antriebsenergie. Mit dieser werden wiederum Spanner zum Arretieren von Bauteilen in entsprechende Position gebracht. Grundlage hierfür ist die aus der Entwicklungssoftware des SPS-Systems exportierte Zuweisungsliste des Anlagenbereichs TH03RHB4. Diese ist wie aus Abb. 64 ersichtlich, durch die Exportfunktion der Symboltabelle aus der Entwicklungsumgebung des SPS-Systems zu exportieren.

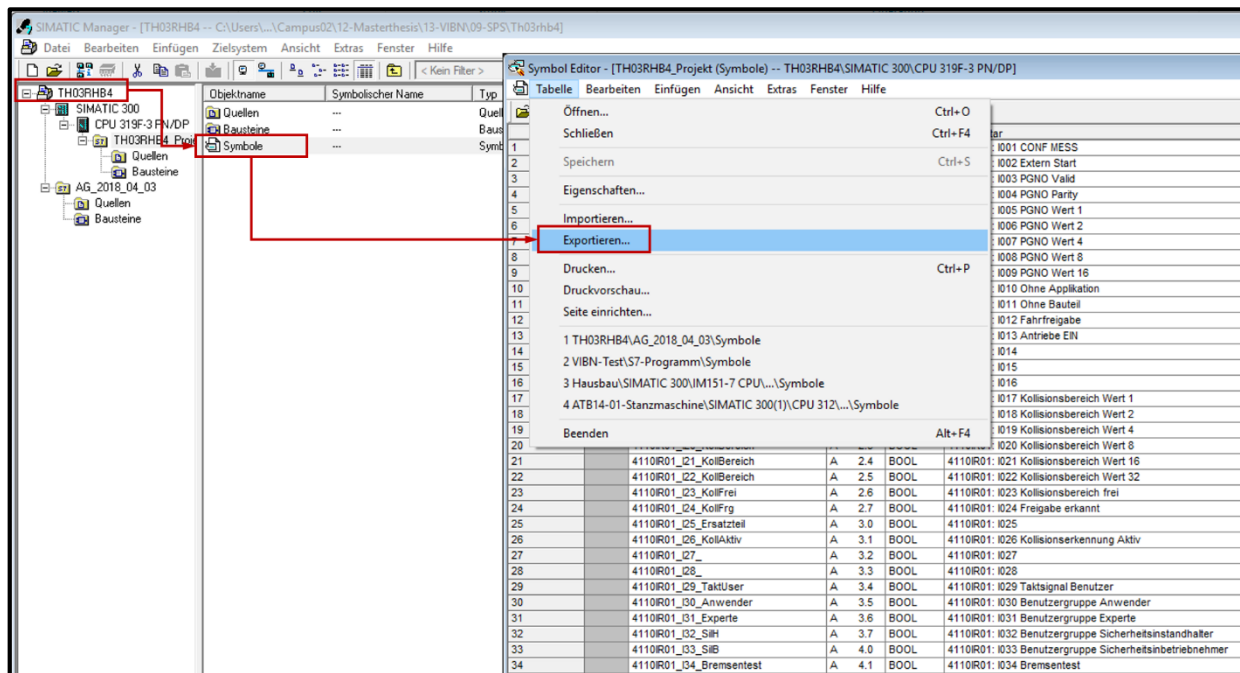


Abb. 64: Export der Symboltabelle respektive Signale des SPS-Systems zur weiteren Aufbereitung, Quelle: Eigene Darstellung.

Im Anschluss daran steht die exportierte Symboltabelle in Form der *export_S7_Zuli_HB4.sdf* zur weiteren Verarbeitung zur Verfügung. Diese lässt sich durch einen handelsüblichen Texteditor öffnen. Nachfolgend sind die Inhalte der Datei *export_S7_Zuli_HB4.sdf* in die für den Import nach Process Simulate notwendige Form überzuführen. Im Zuge dessen sind nur Signale von Belang und Interesse, welche den Datentyp *BOOL* aufweisen sowie den Operanden *E* für Eingangssignal und den Operanden *A* für Ausgangssignal aufweisen.

Ausschließlich diese Datenoperanden respektive Signale stehen in Korrespondenz mit Sicherheitseinrichtungen, Fertigungskomponenten und Industrierobotern, welche im virtuellen Anlagenmodell abgebildet werden. Alle weiteren Datentypen wie zum Beispiel Merker, Timer, Zähler, FCs, FBs, UDTs et cetera, dienen zur internen Verarbeitung von Programmsequenzen sowie internen Speicherung von Daten und Werten. Diese werden weiterhin von der realen SPS im Rahmen des Hardware-in-the-Loop Aufbaus verarbeitet und stehen in Interaktion mit dem Peripherieeingangsbereich der CPU sowie dem Peripherieausgangsbereich derselben. Um die Inhalte der Datei *export_S7_Zuli_HB4.sdf* für die weitere Verarbeitung hinsichtlich der nötigen Signalaufbereitung einfach und übersichtlich zu verwalten, werden diese in die Datei *HB4-Zuli.xlsx* überführt. Damit stehen in weiterer Folge die Filter- und Sortierfunktionen aus Microsoft Excel zur Verfügung. Abb. 65 stellt die Datenüberführung in grafischer Form dar und dient zugleich als Formempfehlung zur weiteren Verwaltung der aus dem SPS-System zur Verfügung gestellten Signale beziehungsweise Variablen.

The screenshot shows an Excel spreadsheet titled 'HB4-Zuli.xlsx'. The spreadsheet contains a table with the following columns: Name, Adresse, Datentyp, and Kommentar. Red callout boxes are placed over the spreadsheet to highlight specific elements: 'Name des Signals/Variablen' points to the 'Name' column; 'Adresse des Signals/Variablen' points to the 'Adresse' column; 'Datentyp des Signals/Variablen' points to the 'Datentyp' column; and 'Kommentars des Signals/Variablen' points to the 'Kommentar' column. The table data is as follows:

| | Name | Adresse | Datentyp | Kommentar |
|----|--------------------------|---------|----------|---------------------------------|
| 2 | 4110IR01_I1_ConfMess | A 0.0 | BOOL | 4110IR01: I001 CONF MESS |
| 3 | 4110IR01_I2_ExtStart | A 0.1 | BOOL | 4110IR01: I002 Extern Start |
| 4 | 4110IR01_I3_PgnoValid | A 0.2 | BOOL | 4110IR01: I003 PGNO Valid |
| 5 | 4110IR01_I4_PgnoParity | A 0.3 | BOOL | 4110IR01: I004 PGNO Parity |
| 6 | 4110IR01_I5_Pgno | A 0.4 | BOOL | 4110IR01: I005 PGNO Wert 1 |
| 7 | 4110IR01_I6_Pgno | A 0.5 | BOOL | 4110IR01: I006 PGNO Wert 2 |
| 8 | 4110IR01_I7_Pgno | A 0.6 | BOOL | 4110IR01: I007 PGNO Wert 4 |
| 9 | 4110IR01_I8_Pgno | A 0.7 | BOOL | 4110IR01: I008 PGNO Wert 8 |
| 10 | 4110IR01_I9_Pgno | A 1.0 | BOOL | 4110IR01: I009 PGNO Wert 16 |
| 11 | 4110IR01_I10_OhneApp | A 1.1 | BOOL | 4110IR01: I010 Ohne Applikation |
| 12 | 4110IR01_I11_OhneBauteil | A 1.2 | BOOL | 4110IR01: I011 Ohne Bauteil |
| 13 | 4110IR01_I12_Fahrfrg | A 1.3 | BOOL | 4110IR01: I012 Fahrfreigabe |
| 14 | 4110IR01_I13_AntriebeEIN | A 1.4 | BOOL | 4110IR01: I013 Antriebe EIN |

Abb. 65: Überführte Symboltabelle respektive Signale des SPS-Systems zur weiteren Aufbereitung für Process Simulate, Quelle: Eigene Darstellung.

Beide Dateien, sowohl *export_S7_Zuli_HB4.sdf* als auch *HB4-Zuli.xlsx*, sind dem Anhang der Masterarbeit beigelegt. Die Datei *HB4-Zuli.xlsx* enthält ab diesem Zeitpunkt den Datenvorrat aller nachfolgend benötigten Signale zum weiteren Aufbau des VIBN-Prototypen. Diese stellen die Basis der anschließend behandelten Aufbereitung der Signalschnittstelle zwischen virtuellem Roboter, Process Simulate und der realen SPS dar.

7.5 Modellaufbereitung Roboter

Das vorangegangene Unterkapitel hatte die Datenbereitstellung aus dem SPS-System zum Inhalt. Unterkapitel 7.5 setzt sich eingehend mit der Modellaufbereitung der Industrieroboter auseinander, welche zur Qualifizierung derselben für den Einsatz in Rahmen der VIBN vorausgesetzt wird. Ebenso wird die Festlegung der Systemgrenze in Bezug auf jene Robotersignale analysiert, welche zur Kommunikation mit dem SPS-System von Bedeutung sind. Auch wird der Kommunikationsweg der zum Einsatz kommenden Anlagensignale behandelt, wie auch der Korrespondenzverlauf derselben. Des Weiteren erfolgt eine Analyse der vorzunehmenden Schritte, die ein fehlerfreier Import der Roboterschnittstelle in das virtuelle Anlagenmodell erfordert. Abschließend wird die Bereitstellung und Import der Roboterprogramme behandelt, die zur Aufgabenerfüllung derselben Anwendung finden.

7.5.1 Festlegung Systemgrenze

Der Anlagenbereich HB4-BA1 verfügt über drei Roboter, für die eine Schnittstelle aufzubauen ist. Über diese kommuniziert in weiterer Folge der jeweilige virtuelle Industrieroboter mit der Real-SPS. An dieser Stelle ist auch die Festlegung der Systemgrenze pro virtuellem Robotermodell vorzunehmen, da jeder reale Roboter als Teilnehmer im Profinet IO-Netzwerk des SPS-Systems fungiert, selbst jedoch auch über ein eigenes Feldbusnetzwerk verfügt.

Im Zuge der VIBN haben nur jene Eingangs- und Ausgangssignale Relevanz, die in Interaktion mit der realen SPS stehen. Daher sind diese aus der Gesamtschnittstelle herauszufiltern. Ein vollständiger Import der gesamten Roboterschnittstelle, ohne die Funktion des Simulationswerkzeugs zu beeinträchtigen, ist technisch möglich. Er wirkt sich jedoch auf die Übersichtlichkeit und die Verwaltbarkeit der Eingangs- und Ausgangssignale in PS 14.1 aus, wie aus Tab. 7 ersichtlich wird.

| Anzahl Roboter | Anzahl Signale ohne Systemgrenze | Anzahl Signale mit Systemgrenze |
|----------------|----------------------------------|---------------------------------|
| 3 | 53469 | 1641 |

Tab. 7: Vergleich der Signalanzahl mit und ohne Festlegung der Systemgrenze an der Roboterschnittstelle, Quelle: Eigene Darstellung.

Im Rahmen der Festlegung der Systemgrenze pro Roboter werden alle Timer, Counter, Flags und zyklischen Flags entfernt, da diese ausschließlich roboterintern Anwendung finden. Ebenso werden alle Eingangssignale und Ausgangssignale gelöscht, die nicht zur Kommunikation mit der SPS eingesetzt werden. Abb. 66 stellt die Festlegung der Systemgrenze in grafischer Form beispielgebend für alle weiteren Robotersysteme dar.

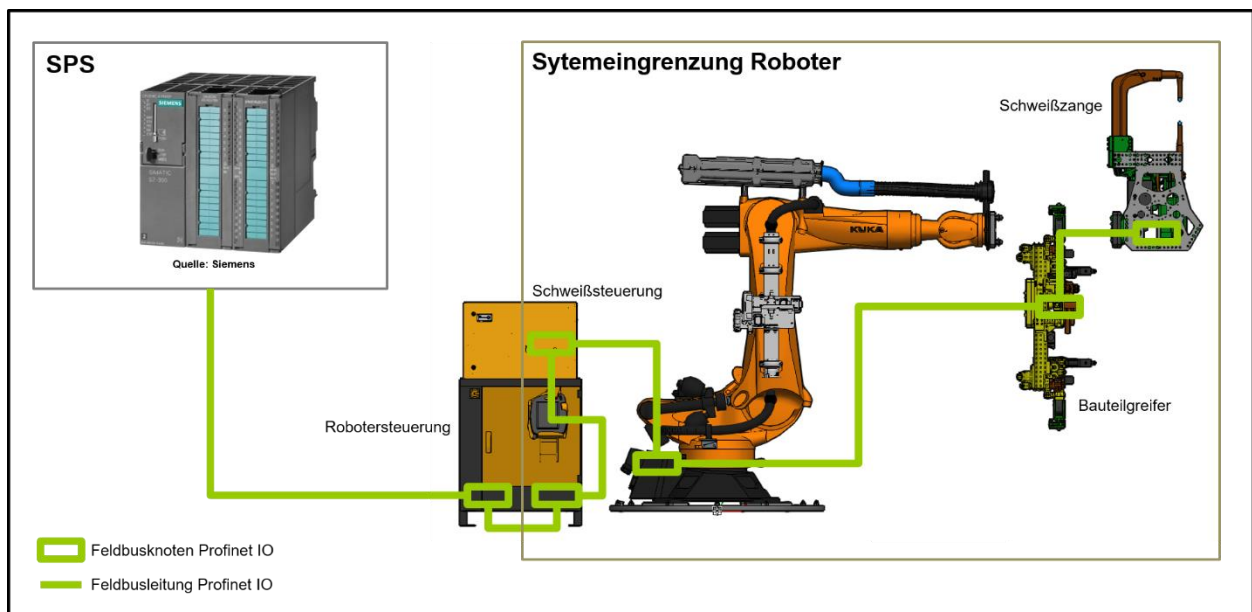


Abb. 66: Festlegung der Systemgrenze des Roboters hinsichtlich anzuwendender Signale im Zuge der VIBN, Quelle: Eigene Darstellung.

Aus dieser wird die Trennung zwischen dem SPS-seitigen Feldbussystem und jenem Feldbussystem deutlich, in dem der Roboter die Rolle des *Masters* einnimmt. Im Zuge der Virtuellen Inbetriebnahme werden demzufolge ausschließlich jene Signale für den Import in das virtuelle Anlagenmodell aufbereitet, die für einen fehlerfreien Datenaustausch zwischen Real-SPS und virtuellem Roboter nötig sind. Robotersignale, die zum Steuern und Regeln von robotergeführten Applikationsprozessen, Applikationswerkzeugen, Greifersystemen und Peripheriesystemen benötigt werden, werden deshalb vor Import in das Simulationsmodell entfernt. Zum Abbilden dieser Vorgänge und Komponenten werden weiterhin generische Simulationsbefehle verwendet. Diese dienen zur Kommunikation zwischen den virtuellen Komponenten sowie zum Nachbilden von physikalischen Vorgängen wie kinematische Abläufen oder dem Fixieren von Bauteilen in Bauteilgreifern.

7.5.2 Signalkorrespondenz zwischen Roboter und SPS

In Unterkapitel 7.3 wurde bereits festgestellt, dass Industrieroboter über eine eigene Steuerung verfügen. Diesem Umstand wird in Process Simulate Rechnung getragen. Zur Abbildung eines Roboters in herstellerbezogener beziehungsweise nativer Form ist die Installation eines zusätzlichen Programmmoduls erforderlich. Auf diese Voraussetzung wurde vorangehend in Abschnitt 7.1.3 eingegangen.

Jeder Roboter, der im virtuellen Anlagenmodell eingesetzt wird, stellt eine Instanz der Klasse Roboter der Marke KUKA KR270R2700 ULTRA dar. Dies gilt ebenso für jenen Bereich des Robotermodells, der zur Darstellung der steuerungstechnischen Schnittstelle dient. Ebenfalls ist das Verständnis bezüglich der angewandten und ausgeführten Wege im Rahmen des Signalaustauschs zwischen virtuellem Roboter, virtueller Produktionsanlage und realer SPS bedeutend. Diese Eigenschaften werden anhand der in Tab. 8 angeführten Signale hervorgehoben und nachfolgend analysiert.

| Signalzugehörigkeit | Signalname | Signalart |
|-----------------------------|--|----------------|
| HB4110IR01 | \$IN[5] | Eingangssignal |
| HB4110IR01 | \$OUT[8] | Ausgangssignal |
| HB4120IR01 | \$IN[5] | Eingangssignal |
| HB4120IR01 | \$OUT[8] | Ausgangssignal |
| HB4130IR01 | \$IN[5] | Eingangssignal |
| HB4130IR01 | \$OUT[8] | Ausgangssignal |
| virtuelle Produktionsanlage | HB4110IR01.\$IN[5];PGNO WERT 1 | Ausgangssignal |
| virtuelle Produktionsanlage | HB4110IR01.\$OUT[8];A_IN_HOME 1:\$IN_HOME1 | Eingangssignal |
| virtuelle Produktionsanlage | HB4120IR01.\$IN[5];PGNO WERT 1 | Eingangssignal |
| virtuelle Produktionsanlage | HB4120IR01.\$OUT[8];A_IN_HOME 1:\$IN_HOME1 | Ausgangssignal |
| virtuelle Produktionsanlage | HB4130IR01.\$IN[5];PGNO WERT 1 | Eingangssignal |
| virtuelle Produktionsanlage | HB4130IR01.\$OUT[8];A_IN_HOME 1:\$IN_HOME1 | Ausgangssignal |
| SPS | A 0.4 | Ausgangssignal |
| SPS | E 0.7 | Eingangssignal |
| SPS | A 44.4 | Eingangssignal |
| SPS | E 44.7 | Ausgangssignal |
| SPS | A 88.4 | Eingangssignal |
| SPS | E 88.7 | Ausgangssignal |

Tab. 8: Wertevorrat zur Veranschaulichung der Signalkorrespondenz zwischen virtuellen Robotern und realer SPS, Quelle: Eigene Darstellung.

Der Wertevorrat jener Signale, die nachfolgend zur Verdeutlichung der Instanziierung der zur Anwendung gelangenden Roboter dient, wird in Tab. 8 dargestellt. Auch dient diese als Wertevorrat zur Veranschaulichung des Kommunikationsverlaufs und der Signalkorrespondenz zwischen Roboterinstanz, virtueller Produktionsanlage, Simit Unit PN256 und realer SPS.

Wie bereits in der Einleitung dieses Abschnitts erwähnt, stellen aufgrund des datentechnischen Aufbaus die in der virtuellen Produktionsanlage eingesetzten Roboter, Instanzen der Klasse KUKA KR270R2700 ULTRA dar. Das heißt, dass jeder der drei Roboter die im Simulationsmodell zur Durchführung der VIBN eingesetzt sind, über ein vom Klassenobjekt abgeleitetes geometrisches, kinematisches sowie steuerungstechnisches Modell verfügt. Somit verfügen die Roboter HB4110IR01, HB4120IR01, HB4130IR01 jeweils über eine eigene virtuelle Robotersteuerung, die intern jeweils über die selbe Signalschnittstelle verfügt. Demzufolge besitzt jeder der drei eingesetzten Roboter unter anderem das Eingangssignal $\$IN[5]$ und das Ausgangssignal $\$OUT[8]$, wie auch in Abb. 67 hervorgehoben wird.

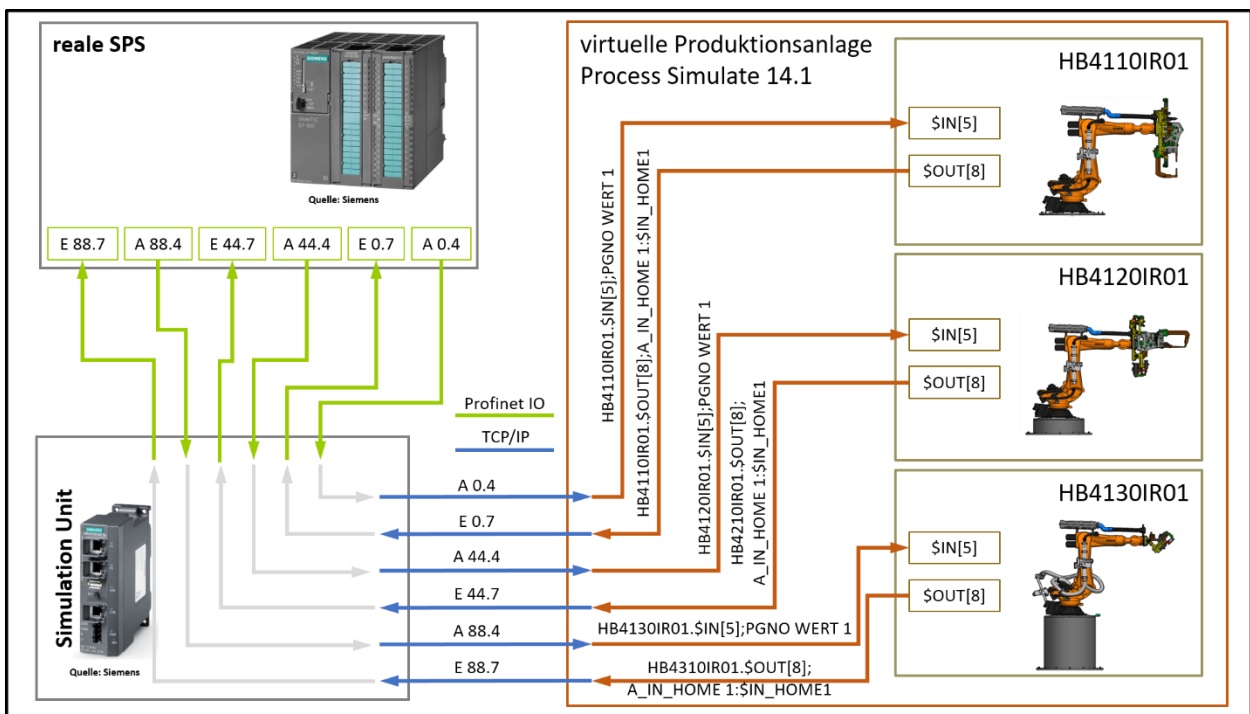


Abb. 67: Signalkorrespondenz zwischen Roboterinstanz, virtueller Produktionsanlage, Simit Unit PN256 und realer SPS, Quelle: Eigene Darstellung.

Diese Signale dienen zum einen zur weiteren Verwendung innerhalb eines jeden Roboters im Rahmen seiner Aufgabenerfüllung, zum anderen zur Kommunikation mit der Außenwelt, im konkreten Anwendungsfall als Feldgerät respektive Profinet IO-Device im Felsbussystem der Produktionsanlage. Auf Grund der virtuellen Abbildung im Simulationsmodell müssen diese roboterzugehörigen Signale durch Process Simulate eindeutig dargestellt werden. Dies ist notwendig, da Process Simulate das gesamte virtuelle Anlagenmodell darzustellen hat, und als Kommunikationspartner zur Simit Unit PN256 agiert. Technisch umgesetzt wird dies durch das ebenfalls in Abb. 67 dargestellte Signalmapping zwischen Roboterinstanz und virtueller Produktionsanlage. Da Process Simulate auch die Rolle innehat, die Außenwelt der Roboterinstanzen darzustellen, gilt ab dem Zeitpunkt des Signalmappings zwischen PS 14.1 und dem jeweiligen Roboter die Signalrichtung des SPS-Systems. Demzufolge muss das Signal $HB4110IR01.\$IN[5];PGNO WERT 1$ des Roboters HB4110IR01 als Ausgangssignal in Process Simulate deklariert werden und das Signal

HB4110IR01.\$OUT[8];A_IN_HOME 1:\$IN_HOME1 des selben Roboters als Eingangssignal. Diese Konvention ist sinngemäß für das gesamte Signalmapping zwischen Roboter und virtueller Produktionsanlage anzuwenden.

Die Länge der Signalbezeichnungen ist durch das Einhalten der Eindeutigkeit innerhalb von PS 14.1 geschuldet sowie der Erfordernis der Zuweisbarkeit des Signals zu dessen Betriebsmittel. Zur Weiterreichung der Robotersignale an die reale SPS ist ein weiterer Mappingvorgang der Signale notwendig. Dieser stellt die Korrespondenz zwischen Process Simulate und der Simulation Unit her. Dies ist zugleich auch jene Stelle, an der die Signalverknüpfung zwischen virtuellem Anlagenmodell und realem SPS-System vollzogen wird. Veranschaulicht wird dies durch den in Abb. 67 dargestellten Mappingvorgang zwischen den Signalen *HB4110IR01.\$IN[5];PGNO WERT 1* und *A 0.4* sowie den Signalen *HB4110IR01.\$OUT[8];A_IN_HOME 1:\$IN_HOME1* und *E 0.7*. Diese Darstellung gilt sinngemäß für das gesamte Signalmapping zwischen virtueller Produktionsanlage und realer SPS.

Nach eingehender Auseinandersetzung und Analyse betreffend des Kommunikationsverlaufs und der Signalkorrespondenz zwischen virtuellem Roboter, virtueller Produktionsanlage und realem SPS-System wird im nachfolgenden Abschnitt ausführlich auf die Datenaufbereitung der erforderlichen Robotersignale eingegangen.

7.5.3 Signalimport

Nach Festlegung der Systemgrenzen und Analyse der Signalkorrespondenz kann mit der Aufbereitung der Datei zum Import der Roboterschnittstelle begonnen werden. Pro Roboter ist hierfür eine Importdatei zu erstellen. Die nachfolgend beschriebenen Schritte werden am Beispiel des Roboters HB4110IR01 erläutert und gelten für alle weiteren Roboterschnittstellen, die es zu importieren gilt. Ausgenommen hiervon sind die korrespondierenden Zieladressen der Robotersignale auf SPS-Seite. Ausgangspunkt ist die zuvor aus dem Realroboter exportierte Datei *HB4110IR01-Langtexte.csv*, welche dem Anhang dieser Arbeit beigelegt ist. Aus dieser sind die relevanten Signale zu extrahieren und der weiteren Bearbeitung zuzuführen. Jedes Signal besteht aus der Signaladresse in nativer Robotersyntax, sowie einer textuellen Beschreibung des Signals, wie es Tab. 9 erläutert.

| Signalsyntax | Signalrichtung | roboterinterne Adresse | Beschreibung des Signals |
|--------------|----------------|------------------------|--------------------------|
| \$IN[5] | Eingangssignal | 5 | PGNO WERT 1 |
| \$OUT[8] | Ausgangssignal | 8 | A_IN_HOME 1:\$IN_HOME1 |

Tab. 9: Erläuterung der nativen Robotersyntax zur Beschreibung von Signalen, Quelle: Eigene Darstellung.

Laut gültigem Programmierstandard zur Programmierung von Robotern der Marke Kuka bei Magna Steyr liegt der Adressbereich pro Roboter, der für die Kommunikation zur SPS verwendet wird, zwischen 1 und 254. Demnach werden alle Eingangs- und Ausgangssignale, die in diesem Adressbereich liegen, der weiteren Datenaufbereitung zugeführt. Die vollständig aufbereitete Importdatei *RobotSignals_HB4110IR01.xls* des Roboters HB4110IR01 ist dem Anhang in digitaler Form beigelegt. Die zu befolgenden Formvorgaben hebt Abb. 68 grafisch hervor. Ebenfalls erfolgt darin eine Funktionsbeschreibung der zu befüllenden Spalten. Die Zeichenfolge *<Robot>* am Beginn eines jeden Signalnamens der in Abb. 68 dargestellten Tabelle dient dazu, automatisch den Modellnamen der jeweiligen Instanz des virtuellen Roboters zuzuweisen. Damit wird die Eindeutigkeit des Signalnamens im virtuellen Anlagenmodell sichergestellt.

| 1 | Signal Name | Electrical Name | Internal Name | Function | Type | Address | OPC Connection | PLC Connection | Comment |
|-----|--|-----------------|---------------|----------|------|---------|----------------|----------------|---------|
| 332 | <Robot>.\$OUT[40];Freigabe der SGT:AP_SGT_FRG | | \$OUT[40] | | BOOL | I4.7 | | | |
| 333 | <Robot>.\$OUT[41];FK von der Steuerung fehlt:AP_FK_FEHLT | | \$OUT[41] | | BOOL | I5.0 | | | |
| 334 | <Robot>.\$OUT[42];Bauteil aussteuern:AP_BTL_AUSSTEUERN | | \$OUT[42] | | BOOL | I5.1 | | | |
| 335 | <Robot>.\$OUT[43];Reserve | | \$OUT[43] | | BOOL | I5.2 | | | |
| 336 | <Robot>.\$OUT[44];Reserve | | \$OUT[44] | | BOOL | I5.3 | | | |
| 337 | <Robot>.\$OUT[45];Reserve | | \$OUT[45] | | BOOL | I5.4 | | | |
| 338 | <Robot>.\$OUT[46];Reserve | | \$OUT[46] | | BOOL | I5.5 | | | |
| 339 | <Robot>.\$OUT[47];Reserve | | \$OUT[47] | | BOOL | I5.6 | | | |
| 340 | <Robot>.\$OUT[48];Reserve | | \$OUT[48] | | BOOL | I5.7 | | | |
| 341 | <Robot>.\$OUT[49];IM Bereich WERT 1:APS_BEREICH | | \$OUT[49] | | BOOL | I6.0 | | | |
| 342 | <Robot>.\$OUT[50];IM Bereich WERT 2:APS_BEREICH | | \$OUT[50] | | BOOL | I6.1 | | | |
| 343 | <Robot>.\$OUT[51];IM Bereich WERT 4:APS_BEREICH | | \$OUT[51] | | BOOL | I6.2 | | | |

Abb. 68: Form und Inhalt der importfähigen Signalschnittstelle für die Instanz des virtuellen Roboters HB4110IR01, Quelle: Eigene Darstellung.

Sind Eingangssignale zur Darstellung von Ganzzahlen auf Seiten des realen Roboters zu einer Variablen zusammengefasst, so kann diese Variable ebenfalls durch Anlegen in der Signalschnittstelle des virtuellen Roboters diesem bekanntgegeben werden. Voraussetzung hierfür ist eine exakte Prozessdatenbreite von 8, 16, oder 32 Bit. Am Beispiel des Signals <Robot>_APS_VARIANTE wird dies verdeutlicht.

Das Signal <Robot>_APS_VARIANTE repräsentiert in der virtuellen Produktionsanlage die roboterintern verwendete Variable APS_VARIANTE der jeweiligen Instanz eines virtuellen Roboters. Die Variable APS_VARIANTE fasst die Robotersignale \$OUT[169] bis \$OUT[184] zur einer Ganzzahl zusammen und besitzt somit eine Prozessdatenbreite von 16 Bit. Damit der zur Programmlaufzeit der virtuellen Roboterinstanz gebildete Aktualwert der Robotervariable APS_VARIANTE der SPS richtig übergeben werden kann, muss APS_VARIANTE als Signal in der Roboterschnittstelle angelegt werden. Nach Anlegen von APS_VARIANTE muss dieser mit dem Type WORD deklariert werden, da dieser eine Prozessdatenbreite von 16 Bit besitzt. Anschließend ist APS_VARIANTE die Startadresse jenes SPS-Signals zuzuweisen, welches mit der Startadresse auf Seiten der Instanz des virtuellen Roboters korrespondiert. Im konkreten Anwendungsfall ist dies I21, da das SPS-Eingangssignal I21.0 mit dem Roboter Ausgangssignal \$OUT[169] korrespondiert. Abb. 69 verdeutlicht das beschriebene Signalmapping in grafischer Form.

| 1 | Signal Name | Internal Name | Type | Address |
|----|---|---------------|------|---------|
| 2 | | | | |
| 3 | <Robot>_APS_VARIANTE | APS_VARIANTE | WORD | I21 |
| 4 | | | | |
| 5 | <Robot>.\$OUT[169];VARIANTE WERT 1:APS_VARIANTE | \$OUT[169] | BOOL | I21.0 |
| 6 | <Robot>.\$OUT[170];VARIANTE WERT 2 | \$OUT[170] | BOOL | I21.1 |
| 7 | <Robot>.\$OUT[171];VARIANTE WERT 4 | \$OUT[171] | BOOL | I21.2 |
| 8 | <Robot>.\$OUT[172];VARIANTE WERT 8 | \$OUT[172] | BOOL | I21.3 |
| 9 | <Robot>.\$OUT[173];VARIANTE WERT 16 | \$OUT[173] | BOOL | I21.4 |
| 10 | <Robot>.\$OUT[174];VARIANTE WERT 32 | \$OUT[174] | BOOL | I21.5 |
| 11 | <Robot>.\$OUT[175];VARIANTE WERT 64 | \$OUT[175] | BOOL | I21.6 |
| 12 | <Robot>.\$OUT[176];VARIANTE WERT 128 | \$OUT[176] | BOOL | I21.7 |
| 13 | <Robot>.\$OUT[177];VARIANTE WERT 256 | \$OUT[177] | BOOL | I22.0 |

Abb. 69: Signalmapping von Robotersignalen, die roboterintern zu Variablen zusammengefasst sind, Quelle: Eigene Darstellung.

Abweichende Prozessdatenbreiten müssen mittels Signalmanipulation durch einen Logikblock aufbereitet werden. Dies kann in Form einer White-Box-Lösung durch Ausprogrammierung im Logikblock erfolgen, wie

auch als Grey-Box-Lösung durch Aufruf einer DLL-Datei innerhalb des Logikblocks, die die erforderliche Signalmanipulation sicherstellt.

Danach ist die Instanz des virtuellen Roboters im Simulationssystem auszuwählen und das Funktionsmenü *Robot Signals* aufzurufen. Durch Ausführung des Befehls *Import Signal* wird die Signalschnittstelle des Roboters HB4110IR01 importiert. Ergänzend stellt dies Abb. 70 in bildlicher Form dar.

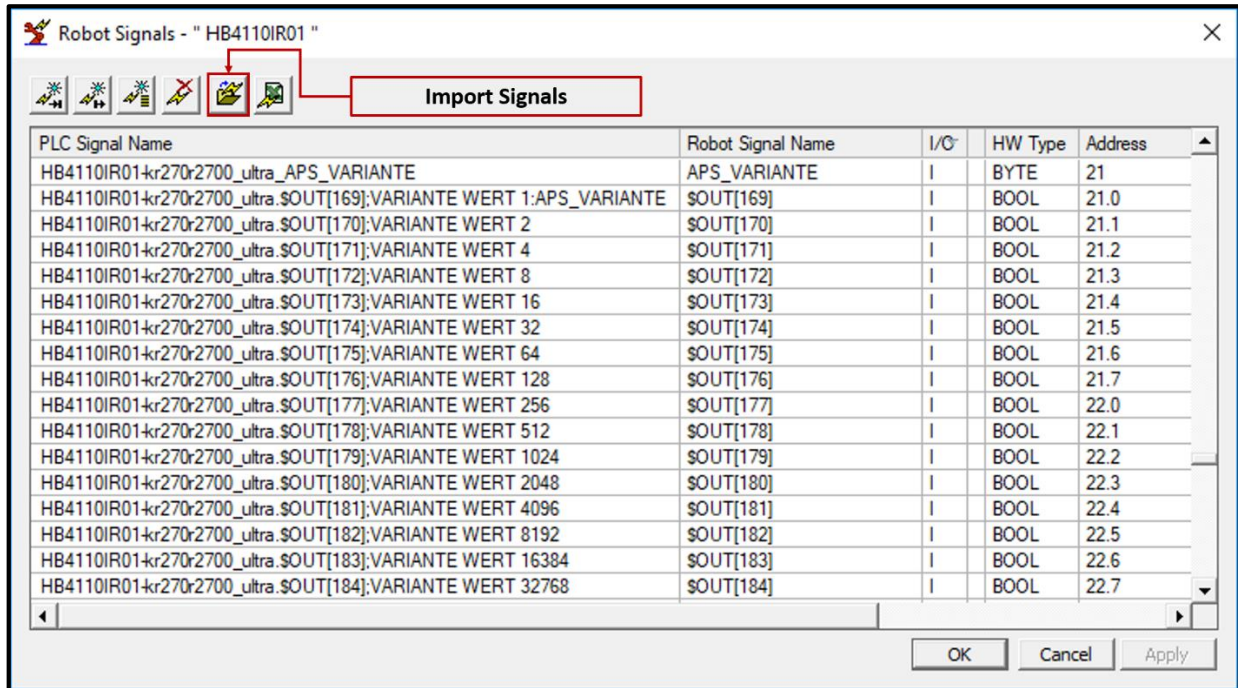


Abb. 70: Erfolgreicher Import der Signalschnittstelle des Roboters HB4110IR01, Quelle: Eigene Darstellung.

Nach Fertigstellung und Abschluss der Signalaufbereitung der an der VIBN beteiligten Roboter erfolgt die erforderliche Bereitstellung der Applikationsprogramme der Roboter, die zu deren Funktionserfüllung im Zuge der Karossenfertigung notwendig sind.

7.5.4 Roboterprogramme

Abschnitt 7.5.4 hat die Bereitstellung der Roboterprogramme zum Inhalt. Durch diese wird die Aufgabenerfüllung des jeweiligen Roboters im Rahmen des Fertigungsprozesses sichergestellt. Im Rahmen der Bereitstellung für die VIBN ist zwischen zwei grundlegenden Importmethoden zu unterscheiden. Ausschlaggebend hierfür ist die Eigenschaft des Roboterprogrammes. Roboterprogramme, die zur Aufgabenerfüllung verschiedene Raumpunkte anzufahren haben, sind als Objekt der generischen Datenklasse *Process* abzubilden. Passend zur jeweiligen Aufgabe, werden diese Programme in spezialisierten Subklassen abgebildet. Eine sehr häufig verwendete Subklasse ist die Klasse *PmWeldOperation*, welche zur Darstellung von robotergeführten Schweißoperationen dient. Anhand des in Abb. 71 dargestellten Schweißprogrammes *g30hb4110r01ps01* wird diese Form der Abbildung von Roboterprogrammen im Simulationsmodell verdeutlicht. Basierend auf den in Unterkapitel 4.4 erörterten Vorgehensweisen und Methoden zur virtuellen Absicherung von Produktionsanlagen ist davon auszugehen, dass diese Form der Roboterprogramme bereits zu Beginn der VIBN-Aktivitäten Inhalt des Simulationsmodells sind. Andernfalls sind diese Programme mittels Uploadmechanismus in die virtuelle Produktionsanlage zu importieren. Dazu stellt *Process Simulate* die Funktion *Upload Program* zur Verfügung.

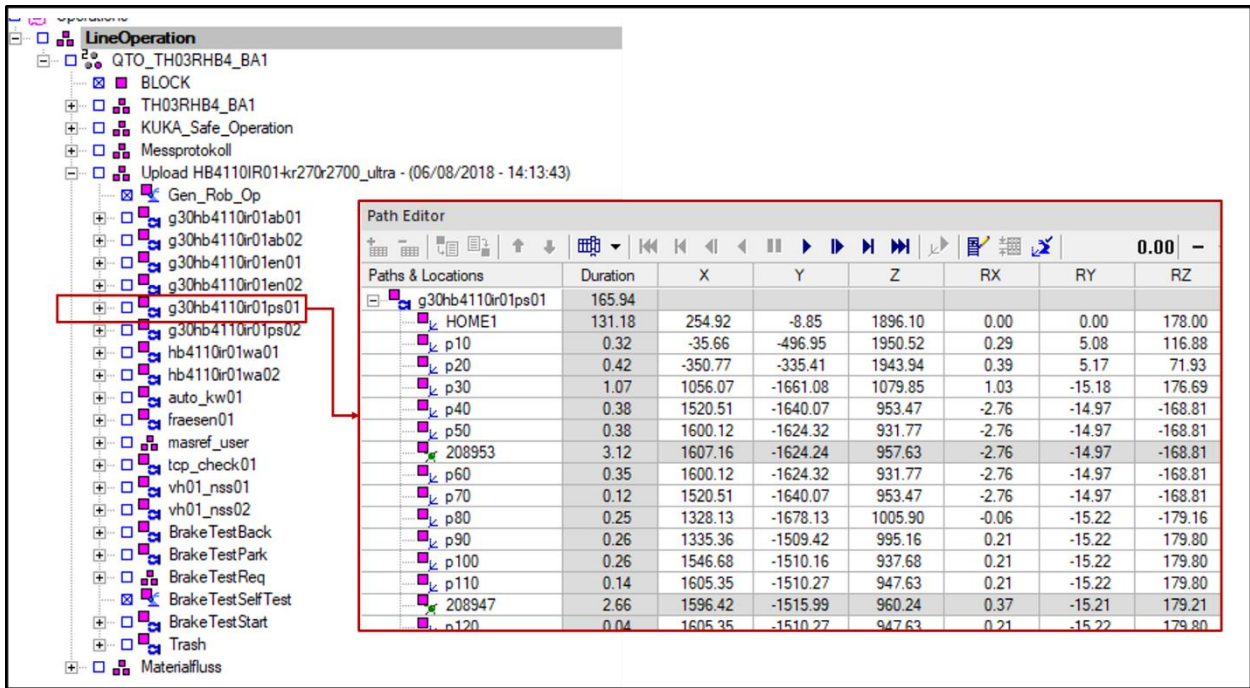


Abb. 71: Abbildung von Roboterprogrammen mit Positionspunkten im kartesischen Raum im VIBN-Modell, Quelle: Eigene Darstellung.

Die zweite Form von Roboterprogrammen ist jene, die ausschließlich aus Programmteilen bestehen, die zur Kommunikation, zur Steuerung und Überwachung von Applikationsprozessen sowie zur Ermittlung und Speicherung von Werten dienen. Diese werden der Instanz der jeweiligen virtuellen Robotersteuerung mittels der Funktionen zugewiesen, die unter der Funktionsgruppe *Robot Modules* zugewiesen sind. Abb. 72 hebt diese Funktionsgruppe in bildlicher Darstellungsform hervor.

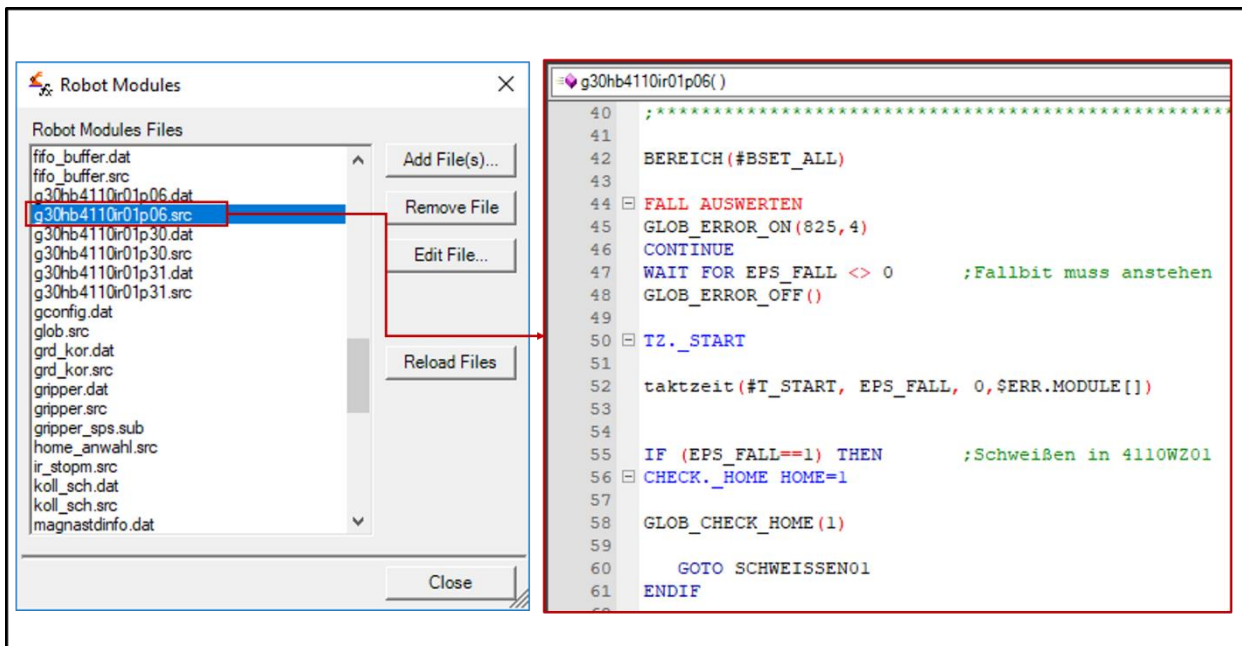


Abb. 72: Einbindung von Roboterprogrammen in das VIBN-Modell, die ausschließlich aus Programmlogik bestehen, Quelle: Eigene Darstellung.

Aus Abb. 72 ist ersichtlich, dass das Roboterprogramm *g30hb4110ir01p06.src* Teil respektive Inhalt der Instanz der virtuellen Robotersteuerung des Roboters HB4110IR01 ist. Demzufolge besteht dieses Roboterprogramm ausschließlich aus Programmteilen zur Abarbeitung von Anweisungen, die keine Interaktion mit dem Achsleitreechner des Roboters aufweisen. Mit der Bereitstellung der relevanten Programme verfügen die abzubildenden Roboter über alle Informationen und Daten, die zur ihrer Funktionserfüllung im Rahmen der VIBN notwendig sind. Das anschließende Unterkapitel hat die Modellaufbereitung jener Anlagenkomponenten zum Inhalt, die nicht Teil des Roboters sind.

7.6 Modellaufbereitung Anlagentechnik

Unterkapitel 7.6 befasst sich mit der nötigen Signalaufbereitung der im virtuellen Anlagenmodell darzustellenden Anlagentechnik. Dazu zählen Spannvorrichtungen zur Arretierung und Geometriebildung von Karosserieteilen, Sicherheitseinrichtungen wie Laserscanner und Werkerschutztore, Bedien- und Meldeeinrichtungen für das Vorort-Personal, sowie Komponenten der elektrischen Antriebstechnik. Ausgenommen hiervon ist die Behandlung von Industrierobotern, welche bereits im vorangegangenen Unterkapitel eingehend analysiert wurde.

Wie bei der zuvor erörterten Signalaufbereitung für Industrieroboter stellt der Abschluss der Offlineprogrammierung den Ausgangspunkt zur Aufbereitung der Signale für die Anlagentechnik dar. Dementsprechend ist davon auszugehen, dass alle Anlagenkomponenten der virtuellen Produktionsanlage, welche Teil des VIBN-Systems sind, folgenden Eigenschaften aufweisen:

- mechanische Konstruktion abgeschlossen
- vollständig in Ausführung
- vollständig hinsichtlich Anzahl und Vorkommen
- Kinematikmodell für Vorrichtungen erzeugt
- Kinematikmodell für Anlagenkomponenten erzeugt
- Kinematikmodell für Sicherheitskomponenten erzeugt
- Kinematik für Fördertechnik erzeugt
- Kinematikmodell den Vorrichtungen zugewiesen
- Kinematikmodell den Anlagenkomponenten zugewiesen
- Kinematikmodell den Sicherheitskomponenten zugewiesen
- Kinematik der Fördertechnik zugewiesen
- alle Betriebsmittel und Maschinen haben richtige Position im Raum eingenommen

Sind die Voraussetzungen erfüllt, kann mit der Aufbereitung der elektrischen Signale sowie der Steuerungslogik für die jeweilige Anlagenkomponente begonnen werden. Die Abbildung der Steuerungslogik in Process Simulate erfordert die Modellierung eines zusätzlichen Objekts im Simulationssystem, welches nachfolgend als *Logikblock (LB)* bezeichnet wird. Zum Verständnis hinsichtlich Funktion und Rolle eines *Logikblockes* innerhalb der virtuellen Produktionsanlage wird dieser in Abschnitt 7.6.1 eingehend analysiert.

7.6.1 Logikblock

Dieser Abschnitt hat die Analyse der Funktion und Rolle eines Logikblockes innerhalb der virtuellen Produktionsanlage zum Inhalt. Nach Fertigstellung der Offlineprogrammierung steht das Simulationsmodell der weiteren Aufbereitung bereit, um dieses zur Einbindung in ein VIBN-System zu qualifizieren. Dazu ist es erforderlich, die bereits bestehenden Modelleigenschaften wie geometrische Form und Kinematik um das elektrische Modell respektive steuerungstechnisches Verhalten zu ergänzen. Dazu werden in Process Simulate sogenannte Logikblöcke eingesetzt. Ein Logikblock besitzt keine eigene grafische Repräsentanz im virtuellen Anlagenmodell, sowie keine eigene Kinematik. Er dient ausschließlich zur Abbildung des logischen Verhaltens einer Anlagenkomponente sowie der Verschaltung der elektrischen Signale derselben. Ein Logikblock verfügt über Eingangsparameter, kann intern über temporäre und statische Variablen verfügen sowie mathematische Formeln beinhalten. Ermittelte Werte werden wieder über Ausgangsparameter an das zu übergebende Objekt übergeben. Es können Prototypen respektive Klassen zur Abbildung bestimmter Logik entwickelt werden, welche mittels Instanziierung mehrfach genutzt werden können. Aus programmierertechnischer Sicht ist der Funktionsumfang eines Logikblocks mit dem eines Funktionsbausteins nach DIN EN 61131-3 vergleichbar. Jedoch hat der Logikblock neben dem Abbilden des steuerungstechnischen Verhaltens die grundlegende Aufgabe zu erfüllen, mechanisches Modell, kinematisches Modell sowie Logikmodell von Anlagenkomponenten zu verbinden. Das Funktionsportfolio des Logikblocks wird nachfolgend am Beispiel der Vorrichtung HB4110WZ02 analysiert, welche in Abb. 73 grafisch dargestellt wird.

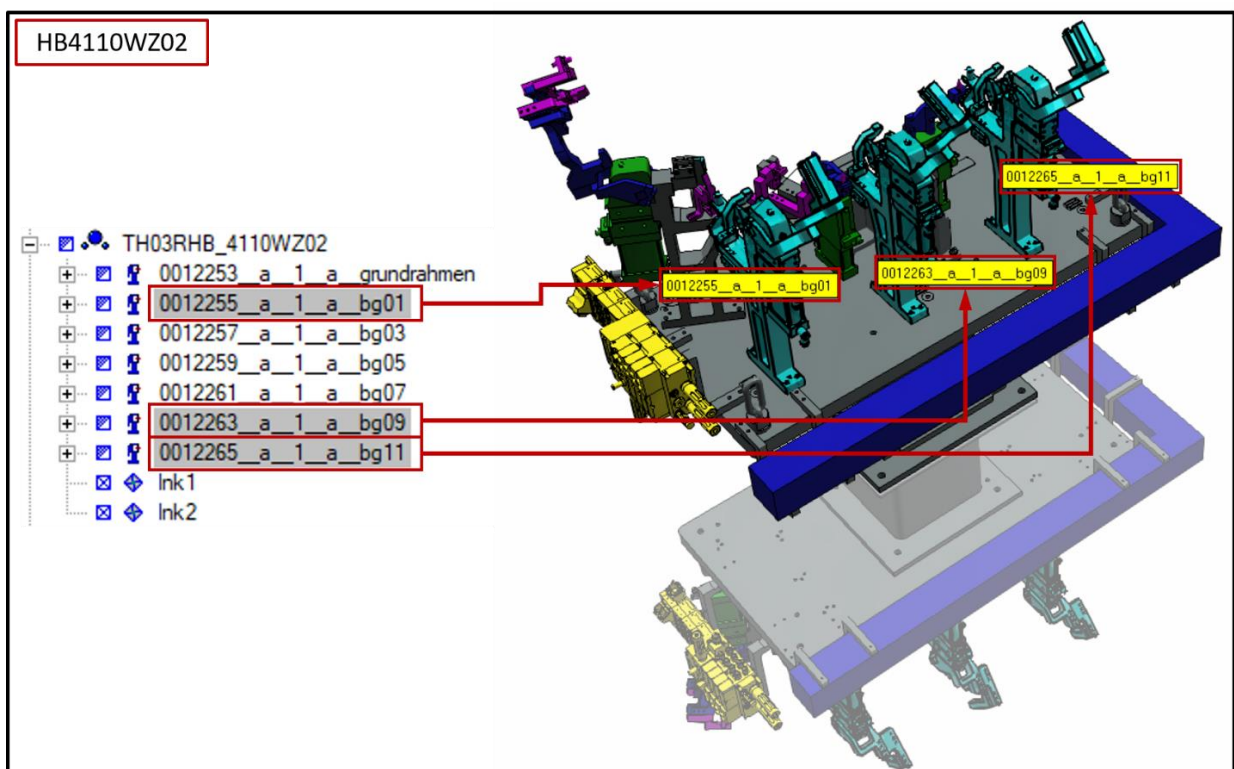


Abb. 73: Mechanisch- / kinematisches Modell der Spannvorrichtung HB4110WZ02, Quelle: Eigene Darstellung.

Die in Abb. 73 dargestellte Spannvorrichtung HB4110WZ02 besteht aus einem Grundrahmen sowie insgesamt sechs Stellgliedern. Alle darin abgebildeten Stellglieder verfügen bereits über die richtige geometrische Form sowie über ein funktionierendes Kinematikmodell. Jedes dieser Stellglieder wird in Realität mittels pneumatischer Energie in die für den jeweiligen Fertigungsschritt gewünschte Endlage gebracht. Dies erfolgt durch Schalten impulsgesteuerter 5/2-Wegeventile, wobei die Steuerleitungen des Ventils indirekt durch Beaufschlagung elektrischer Energie geschaltet werden. Ebenso ist jedes der sechs Stellglieder mit Endlagenkontrollen ausgestattet, damit der Anlagensteuerung das Erreichen der geforderten Endlage rückgemeldet werden kann. Das wird in der realen Vorrichtung durch induktive Sensoren umgesetzt, die im jeweiligen Stellglied verbaut sind. Diese Eigenschaften fehlen dem Simulationsmodell der Vorrichtung HB4110WZ02. Durch Zuweisung von Logikblöcken mit entsprechend modelliertem logischem Verhalten wird die Vorrichtung HB4110WZ02 respektive dessen Stellglieder mit diesen fehlenden Eigenschaften vervollständigt. Auch wird aus dem in Abb. 73 dargestellten Vorrichtungsmodell nicht ersichtlich, dass die Stellglieder 0012255_a_1_a_bg01, 0012263_a_1_a_bg09, 0012265_a_1_a_bg11 in der Realanlage durch ein Pneumatikventil 22QM angetrieben werden. Diese Eigenschaft geht aus dem Elektroplan des Anlagenbereichs TH03RHB4 hervor, wie aus Abb. 74 verdeutlicht wird.

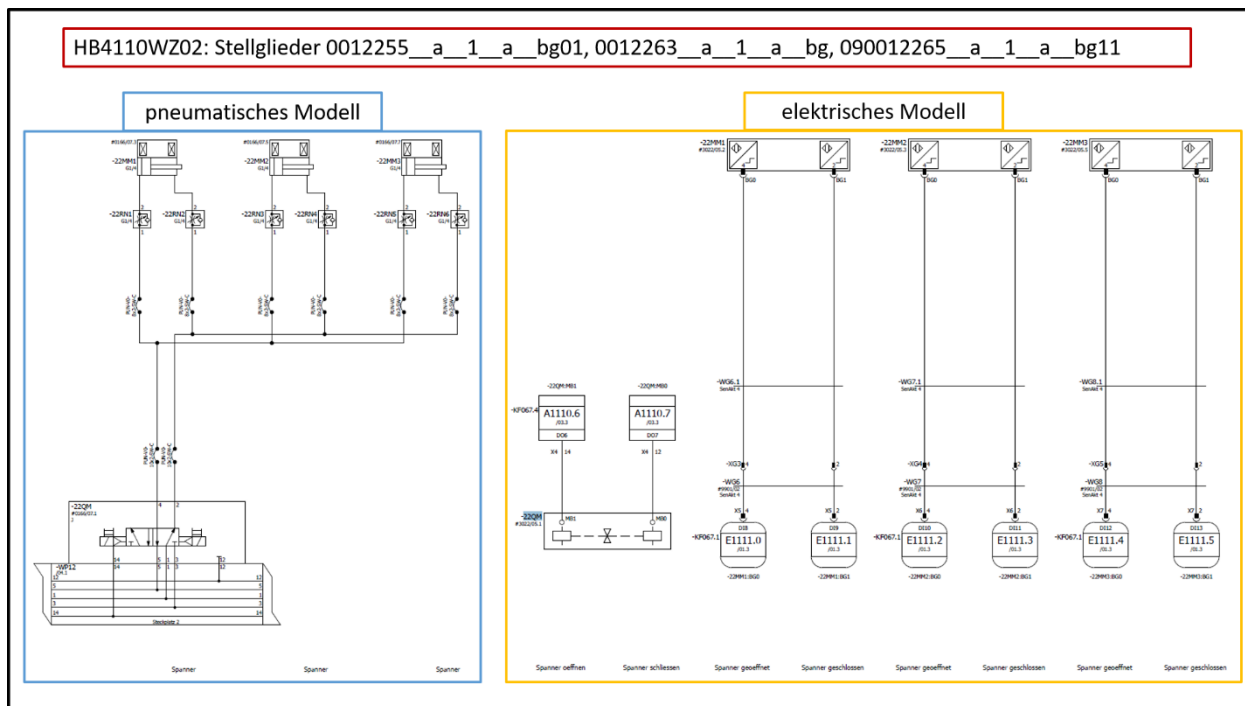


Abb. 74: Pneumatisches und elektrisches Modell der Ventilgruppe 22QM der Vorrichtung HB4110WZ02, Quelle: Eigene Darstellung.

Zur Durchführung einer Virtuellen Inbetriebnahme muss das Simulationsmodell die Fähigkeit besitzen, geometrisches, mechanisches, kinematisches, pneumatisches und steuerungstechnisches Verhalten abzubilden. Diese Aufgabe übernimmt der Logikblock innerhalb des virtuellen Anlagenmodells. Process Simulate 14.1 bietet hierzu unterschiedliche Funktionen, um Logikverhalten manuell, teilautomatisch oder vollautomatisch an der jeweiligen Zielkomponenten zu erzeugen und mit dieser zu verknüpfen. Zur Erzeugung von Logikblöcken zur Lösung des konkreten Anwendungsbeispiels ist die Funktion *Create LB Pose Action and Sensors* an allen drei Stellgliedern der Vorrichtung HB4110WZ02 auszuführen. Dadurch wird durch PS 14.1 jeweils ein Logikblock pro Stellglied erzeugt.

Da die Stellglieder bereits über ein Kinematikmodell verfügen, wird die Logik zur Abbildung der Zylinderansteuerung und der Endlagen automatisch erzeugt. Ebenso wird automatisch die dafür erforderliche Sensorik im Kinematikmodell angelegt und mit dem dazugehörigen Logikblock verknüpft. Am Beispiel des Stellgliedes 0012255_a_1_a_bg01 hebt Abb. 75 dies in grafischer Form hervor.

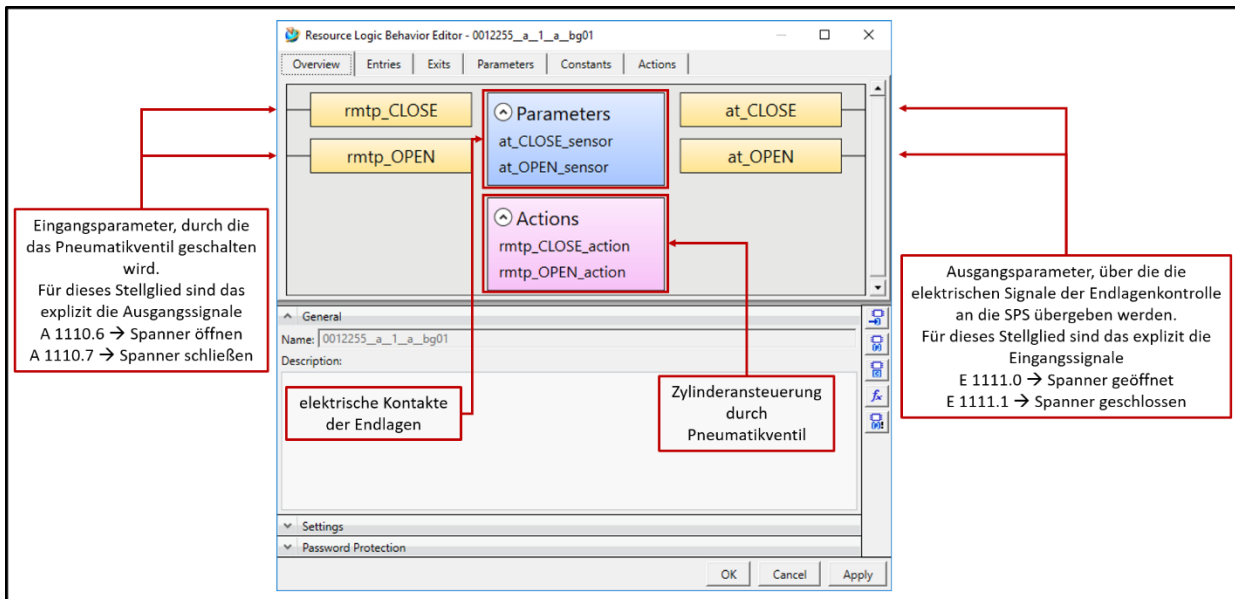


Abb. 75: Logikblock des Stellgliedes 0012255_a_1_a_bg01, Quelle: Eigene Darstellung.

Durch Beschalten der Eingangs- und Ausgangsparameter des jeweiligen Logikblocks mit den Korrespondenzsignalen zu Process Simulate wird der Signalaustausch zum SPS-System hergestellt. Dies geschieht analog zu dem in Abb. 67 erläuterten Kommunikationskonzept von Steuerungssignalen. Das dafür erforderliche Mapping der Steuersignale zwischen Logikblock, Process Simulate sowie dem realen SPS-System wird im anschließenden Abschnitt im Zuge der Erstellung von Importdateien erläutert.

7.6.2 Signalimport

Die Feldgeräte des SPS-Systems der Produktionsanlage TH3RHB04 stellen der Speicherprogrammierbaren Steuerung 4391 Eingangssignale zur Verfügung und nehmen 3806 Ausgangssignale von dieser entgegen. In Summe muss die im HiL-Testkit verbaute SPS 8197 Signale verarbeiten und das virtuelle Anlagenmodell diese Signale zur Durchführung der VIBN abbilden. Die zu importierenden Signale werden durch die Datei *HB4-Zuli.xlsx* bereitgestellt. Um den manuellen Aufwand für den Import dieser Signale nach Process Simulate auf ein Minimum zu beschränken, steht hierfür die Funktion *Connection Mapping* zur Verfügung. Unter Einhaltung der erforderlichen Formvorschriften werden durch Anwendung dieser Funktion die nötigen Eingangs- und Ausgangssignale in das Simulationsmodell importiert und automatisch mit dem richtigen Logikblock verknüpft. Stehen zum Zeitpunkt des Signalimports keine geeigneten Logikblöcke zur Verfügung, so ist ein Dummy-LB im Simulationsmodell anzulegen, der stellvertretend die Funktion der Zieldestination während des Signalimports übernimmt. Im Rahme des Aufbaus des VIBN-Prototypen wird für diese Aufgabe ein Logikblock mit der Benennung *LB* erzeugt.

Am Beispiel der im Anhang beigefügten Datei *HB4110WZ02_import_Signale.xlsx* werden die nötigen Schritte zur Herstellung der Importfähigkeit von Signalen der Anlagentechnik erörtert. Den dafür eingesetzten Logikblock stellt Abb. 76 in grafischer Form dar.

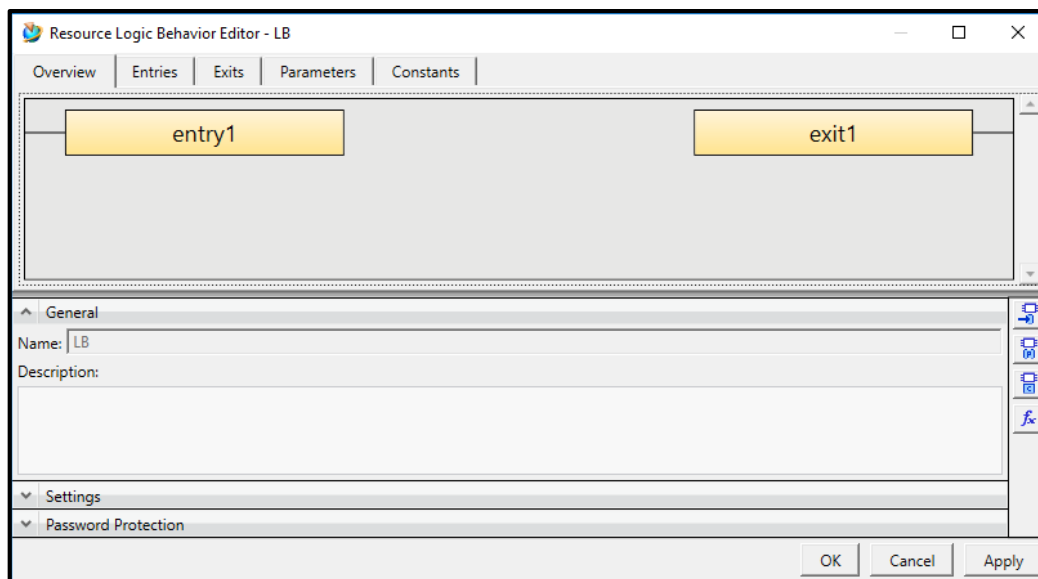


Abb. 76: Temporärer Logikblock mit der Bezeichnung LB zum Zwecke des Signalimports, Quelle: Eigene Darstellung.

Aus Abb. 76 wird ersichtlich, dass für den initialen Signalimport von Sensorik und Aktuatorik des Anlagenbereichs TH3RHB4 die Bezeichnung *LB* verwendet wird. Auch geht daraus hervor, dass als Zielkontakt für SPS-Ausgangssignale der Pin *entry1* angelegt ist, sowie als Zielkontakt von SPS-Eingangssignalen der Pin *exit1*. An diese beiden Pins werden im Zuge des Importvorgangs alle Signale des SPS-Systems verknüpft. Ein systemtechnisch fehlerfreier Signalimport funktioniert ebenso, falls alle Signale beispielsweise auf Pin *entry1* verknüpft werden. Die im Rahmen des Modellaufbaus für den VIBN-Prototypen eingesetzte Methode hat jedoch den Vorteil, bereits eine Vorselektion zwischen Eingangs- und Ausgangssignalen vorgenommen zu haben. Dies ermöglicht eine wesentlich einfachere Verwaltung der Signale im weiteren Verlauf des Modellaufbaus der virtuellen Produktionsanlage. Abb. 77 stellt am Beispiel der Importdatei *HB4110WZ02_import_Signale.xlsx* die gültige Formvorschrift zum Signalimport dar.

| ResourceName | CategoryNameProvider | PinName | SignalWireName | SignalType | Address | Comment | External Connection | PLC Resource |
|--------------|----------------------|---------|-------------------------|------------|---------|---|---------------------|--------------|
| LB | | exit1 | X00+4110WZ02_RES_E13_1 | I | 1607.5 | Reserve | | |
| LB | | exit1 | X00+4110WZ02_RES_E14_1 | I | 1607.6 | Reserve | | |
| LB | | exit1 | X00+4110WZ02_PASSIrfk_1 | I | 1607.7 | Rueckmeldung Passivierungsart (1..Kanalarweise) | | |
| LB | | entry1 | X00+4110WZ02-20QM1:MB1 | Q | 1110.0 | Steuerluftventil ein | | |
| LB | | entry1 | X00+4110WZ02-21QM:MB1 | Q | 1110.4 | Stiftziehzylinder ausfahren | | |
| LB | | entry1 | X00+4110WZ02-21QM:MB0 | Q | 1110.5 | Stiftziehzylinder einfahren | | |

Abb. 77: Auszug der Aufbereitung der Eingangs- und Ausgangssignale der Vorrichtung HB4110WZ02, Quelle: Eigene Darstellung.

Durch Abb. 77 erfolgt die Verdeutlichung der beschriebenen Vorgehensweise für den Signalimport. Durch Zuweisung von Logikblock-Bezeichnung und Pin-Namen wird die Zieldestination im virtuellem Anlagenmodell dem entsprechenden Signal zugewiesen. Durch Festlegung des Signalnamens sowie des Signaltyps

und der Korrespondenzadresse im realen SPS-System erfolgt das Mapping der Signale. Damit wird der durchgängige Kommunikationspfad zwischen Logikblock respektive virtuellem Stellglied, Process Simulate 14.1, Simit Unit PN256 sowie der Real-SPS sichergestellt. Die Signalkorrespondenz erfolgt analog der in Abschnitt 7.5.2 vorgestellten Methodik, wobei die Rolle des Roboters von der jeweiligen Instanz eines Logikblockes übernommen wird. Nach erfolgtem Import der Eingangs- und Ausgangssignale in das Simulationssystem stehen diese zur weiteren Verwendung zur Verfügung. Abschließend müssen die Signale manuell dem jeweiligen Logikblock zur Abbildung logischen Verhaltens zugewiesen werden. Die Gründe dafür sind Inhalt des nachfolgenden Abschnitts.

7.6.3 Herausforderung im Zuge der Modellaufbereitung

Die im vorangegangenen Abschnitt erörterte Methode zur Signalaufbereitung von Anlagenkomponenten und anschließenden Import in das virtuelle Anlagenmodell ist darauf ausgerichtet, Signale während des Importvorgangs direkt mit dem dafür vorgesehenen Logikblock zu verknüpfen. Dies setzt voraus, dass zum Zeitpunkt des Signalimports der entsprechende Logikblock bereits existiert. Bezug nehmend auf das in Abschnitt 7.6.1 erläuterte Beispiel hinsichtlich automatischer Erzeugung von Logikblöcken wird dieser methodische Ansatz zum Mapping von Anlagensignalen mit Logikblöcken durch Process Simulate durchaus unterstützt. Durch Bereitstellung von Systemfunktionen zur automatischen Erzeugung von Logikblöcken durch PS können diese ressourcenschonend generiert werden. Am Beispiel des Stellgliedes *0012255_a_1_a_bg01* wurde dies ausführlich analysiert. In Abschnitt 7.6.2 wurde aufgezeigt, dass für das Mapping zwischen Logikblock und Anlagensignalen der Name des Logikblocks in der Importdatei anzuführen ist. Hier besteht gegenwärtig eine bedeutende Diskrepanz, die einen vollautomatischen Signalmapping entgegensteht und durch Tab. 10 ergänzend verdeutlicht wird.

| Ingenieursdisziplin | Bezeichnungsmethodik des Stellglieds |
|--------------------------|--------------------------------------|
| mechanische Konstruktion | 0012255_a_1_a_bg01 |
| elektrische Konstruktion | X00+4110WZ02-22 |

Tab. 10: Vergleich der Bezeichnungsmethodik von Anlagenkomponenten zwischen mechanische- und elektrischer Konstruktion, Quelle: Eigene Darstellung.

Aus Tab. 10 wird ersichtlich, dass ein und dieselbe Komponente, die in der realen Anlage verbaut ist, durch die am Entstehungsprozess beteiligten Ingenieursdisziplinen einer unterschiedlichen Bezeichnungsmethodik zugeführt wird. Dies hat zur Konsequenz, dass ohne Angleichung respektive Harmonisierung der Bezeichnungsmethodik zwischen den beiden Fakultäten kein effizientes und ressourcenschonendes Signalmapping möglich sein wird. Ein einheitliches und fakultätenübergreifendes Bezeichnungssystem ist daher zukünftig zu erarbeiten. Auf diese Herausforderung wird im Rahmen dieser Arbeit nicht weiter eingegangen – aufgrund ihrer Bedeutsamkeit darf sie jedoch nicht unerwähnt bleiben.

Das nachfolgende Unterkapitel hat die Analyse der abschließenden Schritte zum Inhalt, die zur Erlangung der vollständigen und durchgängigen Signalkorrespondenz zwischen virtueller Produktionsanlage und realen Steuerungskomponenten erforderlich sind.

7.7 Herstellung Signalkorrespondenz

Unterkapitel 7.7 hat die Analyse jener abschließenden Aktivitäten zum Inhalt, die zur Erlangung der durchgängigen Signalkorrespondenz und Kommunikation zwischen virtuellem Anlagenmodell und Real-SPS nötig sind. Auf Seiten der Real-SPS sowie der Simulation Unit sind diese Tätigkeiten abgeschlossen, auf Seite der Simulationssoftware ist diese Maßnahme als letzter Konfigurationsschritt im Zuge des Informations- und Datenaustausches durchzuführen.

Alle relevanten Eingangs- und Ausgangssignale stehen dem virtuellen Anlagenmodell zur Verfügung. Diese sind durch die entsprechenden Importmethoden für Robotik und Logikblöcke beziehungsweise Anlagenkomponenten der virtuellen Produktionsanlage zugeführt worden. Um allen Signalen, welche zur Kommunikation mit der realen SPS definiert sind, den richtigen Kommunikationskanal zuzuweisen, ist die in Abb. 78 hervorgehobene Signaldeklaration im *Signal Viewer* von Process Simulate vorzunehmen.

| Signal Name | Comment | PLC Connection | External Connection | Type | Address | IEC Format | Resource |
|------------------------|---------------------------------------|-------------------------------------|---------------------|------|---------|------------|-------------------------------|
| X00+4110WZ01-15BG11:4 | Schwenker geschlossen externe Abfrage | <input checked="" type="checkbox"/> | HB4 | BOOL | 1102.3 | I1102.3 | |
| X00+4110WZ01-15BG0.1:4 | Bremse aktiviert | <input checked="" type="checkbox"/> | HB4 | BOOL | 1102.4 | I1102.4 | |
| X00+4110WZ01-15BG0.2:4 | Bremse aktiviert | <input checked="" type="checkbox"/> | HB4 | BOOL | 1102.5 | I1102.5 | |
| X00+4110WZ01-BG01:4 | TK-Verstaerkungsplatte QTR Trennwand | <input checked="" type="checkbox"/> | HB4 | BOOL | 1102.6 | I1102.6 | ● 4110WZ01__LB_Sensor_Mapping |
| X00+4110WZ01-BG02:4 | TK-Verstaerkungsplatte QTR Trennwand | <input checked="" type="checkbox"/> | HB4 | BOOL | 1102.7 | I1102.7 | ● 4110WZ01__LB_Sensor_Mapping |
| X00+4110WZ01-BG03:4 | TK-Schottblech Quertraeger Trennwand | <input checked="" type="checkbox"/> | HB4 | BOOL | 1103.0 | I1103.0 | ● 4110WZ01__LB_Sensor_Mapping |
| X00+4110WZ01-BG04:4 | TK-Quertraeger Trennwand Unterteil | <input checked="" type="checkbox"/> | HB4 | BOOL | 1103.1 | I1103.1 | ● 4110WZ01__LB_Sensor_Mapping |
| X00+4110WZ01-BG05:4 | TK-Quertraeger Trennwand Unterteil | <input checked="" type="checkbox"/> | HB4 | BOOL | 1103.2 | I1103.2 | ● 4110WZ01__LB_Sensor_Mapping |
| X00+4110WZ02-01QM1:BP1 | Druckaufbauventil Druck iO | <input checked="" type="checkbox"/> | HB4 | BOOL | 1110.0 | I1110.0 | ● LB |
| X00+4110WZ02-20QM1:BP1 | Steuerluftventil Druck iO | <input checked="" type="checkbox"/> | HB4 | BOOL | 1110.2 | I1110.2 | ● LB |
| X00+4110WZ02-21MM1:BG0 | Stiftziehzylinder eingefahren | <input checked="" type="checkbox"/> | HB4 | BOOL | 1110.4 | I1110.4 | ● (2) |
| X00+4110WZ02-21MM1:BG1 | Stiftziehzylinder ausgefahren | <input checked="" type="checkbox"/> | HB4 | BOOL | 1110.5 | I1110.5 | ● (2) |
| X00+4110WZ02-21MM2:BG0 | Stiftziehzylinder eingefahren | <input checked="" type="checkbox"/> | HB4 | BOOL | 1110.6 | I1110.6 | ● (2) |
| X00+4110WZ02-21MM2:BG1 | Stiftziehzylinder ausgefahren | <input checked="" type="checkbox"/> | HB4 | BOOL | 1110.7 | I1110.7 | ● (2) |
| X00+4110WZ02-22MM1:BG0 | Spanner geoeffnet | <input checked="" type="checkbox"/> | HB4 | BOOL | 1111.0 | I1111.0 | ● (2) |

Abb. 78: Signaldeklaration zur Einbindung in externen Kommunikationskanal, Quelle: Eigene Darstellung.

Mittels Aktivierung der unter der Spalte *PLC Connection* zugeordneten Checkbox wird dem jeweiligen Signal mitgeteilt, dass es für die Kommunikation mit einer Steuerung deklariert ist, die sich außerhalb des virtuellen Produktionsmodells beziehungsweise Process Simulate befindet. Durch weitere Zuweisung des in Abschnitt 7.1.3 projektierten Kommunikationskanals *HB4*, wird dem Signal angegeben, dass es zur Kommunikation mit der Simit Unit PN256 deklariert wird. Diese Deklaration erfolgt durch Auswahl des in der Spalte *External Connection* vorrätigen Parameters *HB4*.

Weiters ist die Unterscheidung zwischen Signalen, die zur externen Kommunikation vorgesehen sind, und jenen zum exklusiven internen Gebrauch von Relevanz. Sie sind für die weitere erfolgreiche Durchführung der Virtuellen Inbetriebnahme von Bedeutung. Diese dienen zur Sicherstellung von Ereignissen und Eigenschaften, die auf Grund fehlender realphysikalischer Objekte innerhalb des Simulationsmodells nachgebildet werden müssen. Dies soll am Beispiel eines Sensors zur Präsenzdetection eines Bauteils verdeutlicht werden, welcher in der Vorrichtung *HB4110WZ02* verbaut ist.

In der Realanlage wird ein Sensorsignal, das für die Präsenzdetection eines Bauteils vorgesehen ist, durch Eintritt des Bauteils in das Detektionsfeld des Sensors ausgelöst. Das Auslösen des Sensors wird über das Schalten des entsprechenden Eingangssignals und nachfolgender Übertragung über das Profinet-System an die reale SPS zurückgemeldet. Dieses Verhalten muss mittels virtueller Bauteile und Sensoren substituiert werden, damit der Eintritt von Bauteilen in den Detektionsbereich von Sensoren im virtuellen Anlagenmodell nachgebildet und der Real-SPS mitgeteilt werden kann. Zur Verdeutlichung dieses Vorgehensweise dienen die in Tab. 11 angeführten Sensorsignale.

| Signalzugehörigkeit | Signalname | PLC Connection |
|-----------------------------|---------------------|----------------|
| virtueller Sensor | LS_100mm_8 | nein |
| virtuelle Produktionsanlage | X00+4110WZ02-BG03:4 | ja |
| reale SPS | E 1112.2 | ja |

Tab. 11: Signalmapping von Sensorsignalen zur Substituierung im virtuellen Anlagenmodell, Quelle: Eigene Darstellung.

Mittels virtuellem Sensor und gleichnamigem Signalnamen *LS_100mm_8* wird die Präsenz eines virtuellen Bauteils detektiert. Dieses Signal dient zur Nachbildung der Präsenzüberwachung in der virtuellen Produktionsanlage und ist ausschließlich innerhalb des Simulationsmodells bekannt. Dementsprechend ist dem Signal *LS_100mm_8* keine aktive Kommunikation mit der Real-SPS zugewiesen. In der realen Produktionsanlage ist diese Aufgabe dem Signal *E 1112.2* zuerkannt. Damit der Schaltvorgang des virtuellen Sensors der Real-SPS mitgeteilt werden kann, ist es erforderlich, das Sensorsignal *LS_100mm_8* durch Einsatz eines Logikblockes zu mappen. Damit wird die Korrespondenz zwischen internem Sensorsignal und jenem Signal hergestellt, welches die Kommunikation zur realen SPS durchführt, wie in Abb. 79 verdeutlicht.

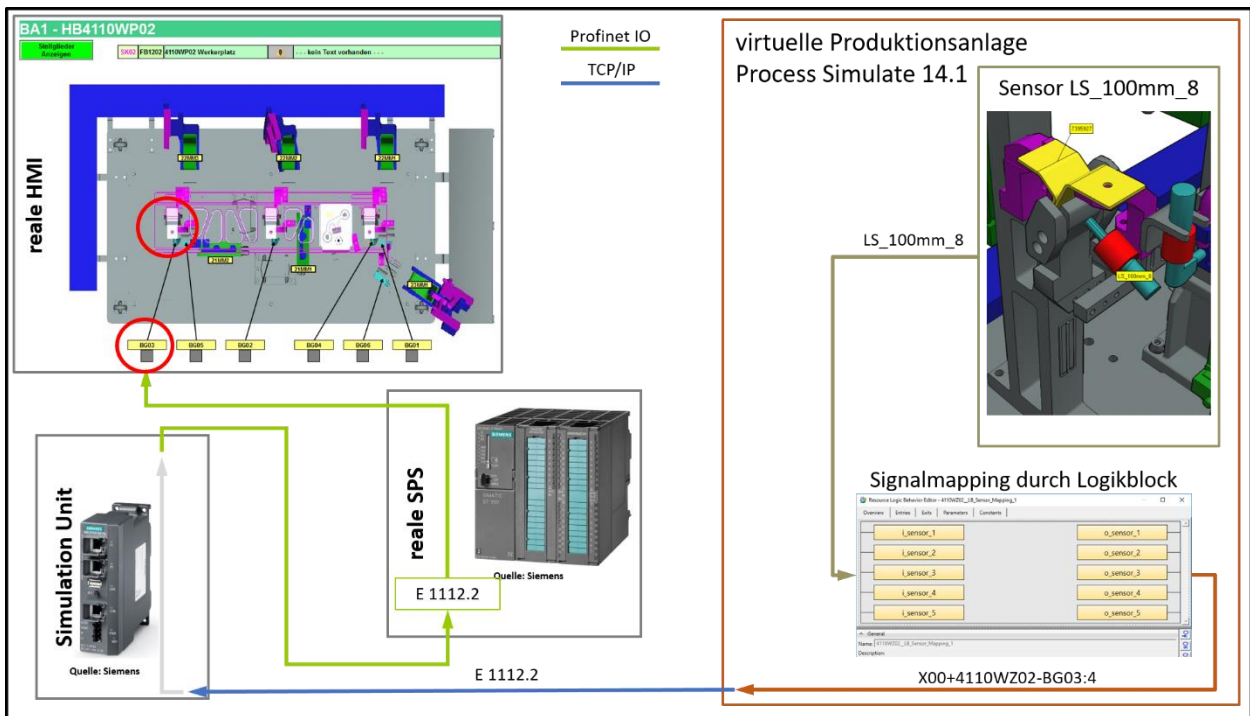


Abb. 79: Signalkorrespondenz zwischen virtuellem Sensor und realer HMI, Quelle: Eigene Darstellung.

Aus Abb. 79 wird ersichtlich, dass diese Aufgabe dem Eingangssignal *X00+4110WZ02-BG03:4* zugewiesen ist, welches nachfolgend die Signalkorrespondenz zur Real-SPS und der realen HMI sicherstellt. Durch flächendeckende Anwendung dieser Methode zum Mappen von Signalen ist das virtuelle Anlagenmodell in Bezug auf durchgängige Signalkorrespondenz vollständig aufgebaut und für den Einsatz im Rahmen der Virtuellen Inbetriebnahme aus Sicht der Signalaufbereitung bereit. Der letzte Modellierungsschritt im Zuge des Modellaufbaus zur Einbindung in die VIBN ist die Definition des Materialflusses. Dieser wird im nachfolgenden Unterkapitel einer Auseinandersetzung unterzogen.

7.8 Materialfluss

Die Definition des Materialflusses im Rahmen des Modellaufbaus zur Durchführung einer Virtuellen Inbetriebnahme stellt den letzten Schritt der Modellierung des virtuellen Anlagenmodells dar. Dieser wird in Unterkapitel 7.8 einer Analyse unterzogen. Die Darstellung des Materialflusses im Zuge der Virtuellen Inbetriebnahme dient zur Wahrung zweierlei Zwecke. Einerseits ermöglicht die visuelle und dynamische Abbildung des aus der Planungsphase vorgegebenen Materialflusses dem Inbetriebnahme- und Bedienpersonal, diesen bereits zu einem früheren Zeitpunkt der Projektrealisierung kennenzulernen. Abb. 80 stellt die planungsseitige Vorgabe des Materialflusses in grafischer Form dar.

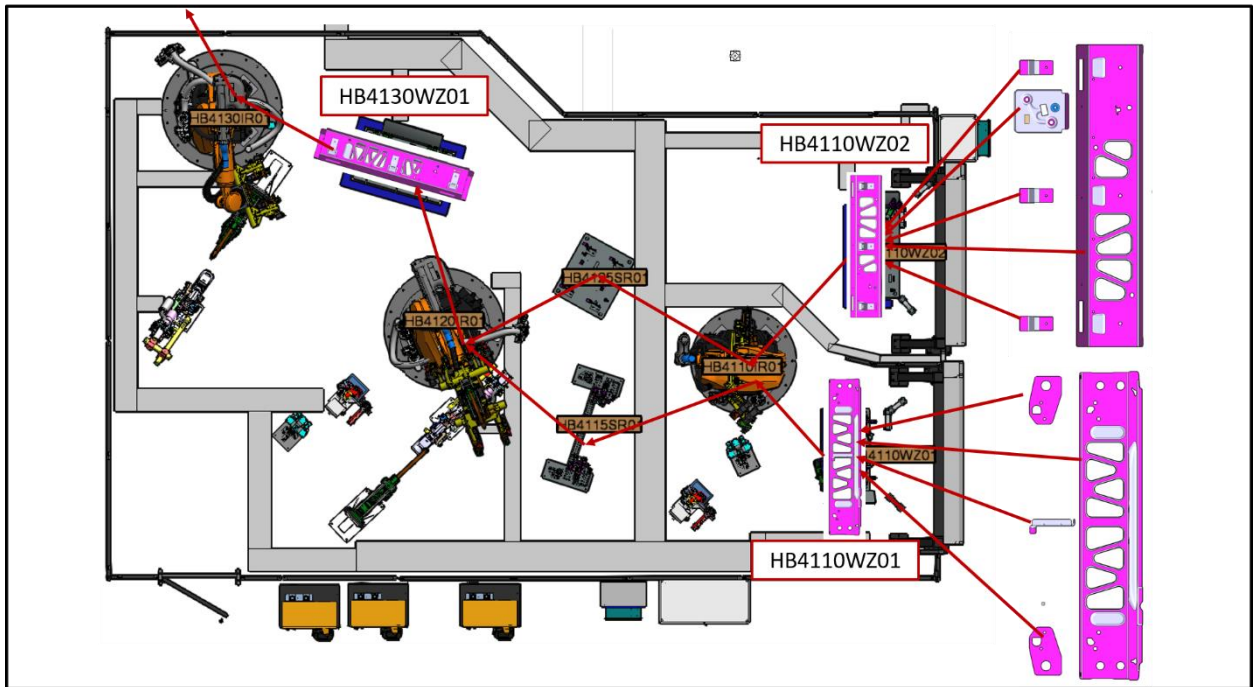


Abb. 80: Planungsvorgabe des Materialflusses des Betriebsartenbereiches HB4-BA1, Quelle: Eigene Darstellung.

Andererseits wird in visueller und dynamischer Form überprüfbar, ob die in der Planungsphase vorgenommenen Abarbeitungsfolgen der Bauteilsequenzen in Bezug auf die Steuerungstechnik umsetzbar sind beziehungsweise richtig umgesetzt wurden. Dazu ist es erforderlich, die Planungsvorgaben in das virtuelle Anlagenmodell einzugliedern und die Materialfolge aus Sichtweise des Fertigungsprozesses darzustellen, wie aus Abb. 81 ersichtlich wird.

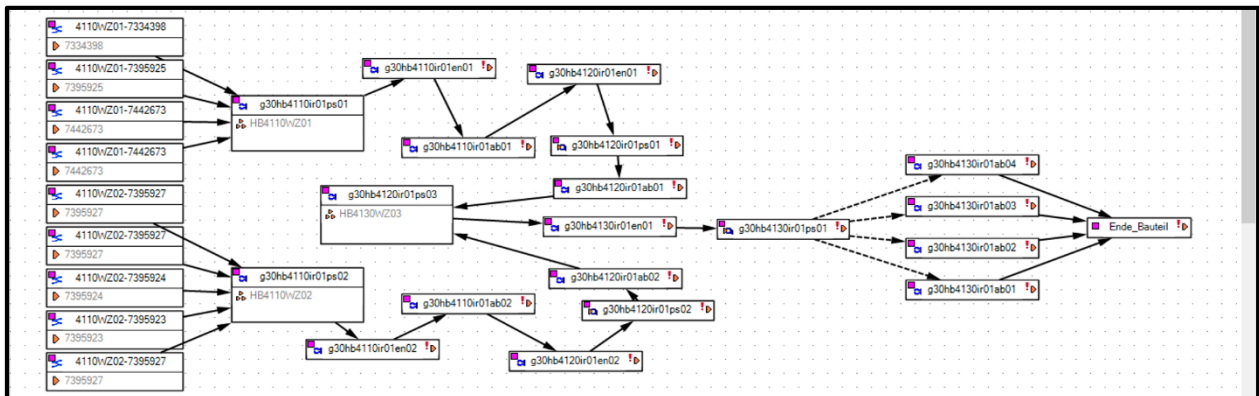


Abb. 81: Prozessbezogene Abbildung des Materialflusses des Betriebsartenbereiches HB4-BA1, Quelle: Eigene Darstellung.

Daraus geht hervor, dass die Zuführung der zu verbauenden Einzelteile manuell in den Stationen HB4110WZ01 sowie HB4110WZ02 erfolgt. Da im virtuellen Anlagenmodell diese Werkertätigkeiten ebenfalls im Rahmen des Materialflusses abgebildet werden müssen, werden diese durch Prozessfolgen substituiert, welche im virtuellen Anlagenmodell durch die Datenklasse *PmObjectFlowOperation* dargestellt werden. In den beiden Einlegestationen erfolgen die ersten Fügestufen, wobei aus den Einzelteilen jeweils eine Baugruppe verschweißt wird. Dies erfolgt durch die robotergeführten Schweißoperationen *g30hb4110ir01ps01* und *g30hb4110ir01ps02*. Anschließend werden die beiden Baugruppen durch Weitertransport mittels Roboter in die nachfolgenden Schweißwerkzeuge manipuliert und den weiteren Schweißoperationen zugeführt. Durch die Schweißoperation *g30hb4120ir01ps03* in Station HB4130WZ01 erfolgt die letzte Fügstufe, die dem Betriebsartenbereich HB4-BA1 zugewiesen ist. Danach erfolgen weitere Schweißoperationen mit anschließender Übergabe in den nächsten Betriebsartenbereich. Diese Übergabe erfolgt durch vier robotergeführte Handlingoperationen, wobei jede dieser vier Roboteroperationen alternierend ausgeführt wird. Dies wird in der prozessbezogenen Darstellung des Materialflusses durch gestrichelte Pfeillinien dargestellt, wie aus Abb. 81 ersichtlich. Das prozesstechnische Ende des Materialflusses wird mit der Operation *Ende_Bauteil* umgesetzt.

Durch die prozessbezogene Darstellung der Materialfolge und programmtechnische Umsetzung im Steuerungsprogramm der SPS ist eine Verifizierung und Validierung des Materialflusses im Rahmen der VIBN frühzeitig möglich. Dies ist insofern bedeutend, da programmtechnisch Normalablauf, Vollfahren, Leerfahren, Ein-/ Ausschleusvorgänge sowie Produktvarianz sicherzustellen sind. Diese gehen aus den planungsbezogenen Darstellungsformen des Materialflusses selten hervor. Dahingehend erschließt sich die Möglichkeit, bei erkannten Abweichungen in Bezug auf vorgegebene Zieleigenschaften der Produktionsanlage ehest Gegenmaßnahmen einzuleiten.

Nach Erstellung des Materialflusses ist die Modellbildung des virtuellen Anlagenmodells abgeschlossen und es kann mit der operativen Durchführung der VIBN begonnen werden. Dies ist Inhalt von Kapitel 8, welches der abschließenden Zusammenfassung der gewonnenen Erkenntnisse aus dem Modellaufbau des virtuellen Anlagenmodells folgt.

7.9 Zusammenfassung

Unterkapitel 7.9 fasst die gewonnenen Erkenntnisse aus den vorangegangenen Analysen und Auseinandersetzungen in Bezug auf Aufbau und Modellierung des Prototyps zur virtuellen Inbetriebnahme zusammen. Durch das nötige Setup des VIBN-Systems konnte die erforderliche Vorgehensweise und Umsetzung hinsichtlich der Datenbereitstellung aus der realen SPS abgeleitet werden. Mittels Projektierung der Si mit Unit 256 konnte eine Methode geschaffen werden, die das Einbinden der dezentralen Peripherie des SPS-Systems in den VIBN-Prototypen erlaubt. Ergebnis daraus war die fehlerfreie Integration der Profinet IO-Feldgeräte in das VIBN-Gesamtsystem. Im Rahmen der Setup-Aktivitäten des Simulationssystems Process Simulate 14.1 kann dieses fehlerfrei in den Kommunikationsverbund zwischen realer SPS und Feldbus-System eingebunden werden. Durch eingehende Auseinandersetzung bezüglich der notwendigen Teamzusammensetzung, die für den Systemaufbau und die Durchführung einer Virtuellen Inbetriebnahme nötig ist, wurden vier Schlüsselrollen identifiziert die für den Erfolg einer VIBN maßgeblichen Einfluss nehmen. Dies sind SPS-Programmierer, Roboter-Programmierexperten in den Bereichen Onlineprogrammierung und Offlineprogrammierung sowie der Simulationsspezialist. Letzterer ist für das Zusammenführen

der steuerungstechnischen Teildisziplinen in das virtuelle Anlagenmodell der VIBN verantwortlich. Durch Analyse des Systemverhaltens der Simulationssoftware hinsichtlich Signalaufbereitung und Datenrichtung konnte die Erkenntnis gewonnen werden, dass betreffend der Datenrichtung stets die Sichtweise der SPS maßgebend ist. Auf Grund der Untersuchungsaktivitäten im Bereich der Bereitstellung der nötigen Daten und Informationen aus der Real-SPS wird die Feststellung abgeleitet, dass ausschließlich Eingangssignale und Ausgangssignale für die Weiterverwendung im VIBN-Modell von Relevanz sind. Betreffend der Aufbereitung der Robotermodelle wird die Empfehlung getroffen, Systemgrenzen bei den in das VIBN-System zu integrierenden Robotersteuerungen festzulegen. Damit wird eine effiziente Datenadministration im virtuellen Anlagenmodell gewährleistet sowie der Fokus auf den zu erfüllenden Modellzweck im Rahmen der VIBN beibehalten. Ebenso ist das Wissen über die Instanziierung der virtuellen Robotersteuerungen von wesentlicher Bedeutung für eine funktionierende Signalkorrespondenz zwischen virtuellem Roboter und realer SPS. Ergebnis aus der Analyse der Methoden zum Import relevanter Robotersignale ist die Erstellung eines Templates für den Signalimport von Robotersignalen. Ebenfalls konnte das Wissen aufgebaut werden, dass Signalimporte der Datentypen Bit, Byte, Wort und Doppelwort ohne erhöhte Aufwände durchführbar sind. Datentypen, welche von diesen Prozessdatenbreiten abweichen, müssen einer nachträglichen Bearbeitung zugeführt werden. Im Rahmen des Imports von Roboterprogrammen konnte die Erkenntnis generiert werden, dass eine Unterscheidung zwischen Bewegungsprogrammen und Logikprogrammen im Zuge der Einbindung in das virtuelle Anlagenmodell nötig ist. Bewegungsprogramme werden mittels Upload-Mechanismus in das virtuelle Anlagenmodell übergeführt, Logikprogramme müssen in der Instanz der jeweiligen virtuellen Robotersteuerung hinterlegt werden. Im Zuge der Modellaufbereitung der Anlagentechnik wurde das Objekt des Logikblocks als entscheidende Komponente identifiziert. Dieser dient zur Verknüpfung von mechanischem Modellverhalten mit den Verhaltensmodellen von Elektro- und Steuerungstechnik. Ebenso dient dieser zur Umsetzung steuerungstechnischer Aufgaben sowie als Schnittstelle zur Wahrung der Signalkorrespondenz zwischen Komponenten der virtuellen Produktionsanlage und der Real-SPS. Auch wurde das Erfordernis der Angleichung der Bezeichnungsmethodik in den Ingenieursdisziplinen mechanischer sowie elektrischer Konstruktion im Zuge der Modellaufbereitung erkannt. Ohne diese Harmonisierung wird ein zukünftiges effizientes, ressourcenschonendes sowie teilautomatisches Signalmapping nicht möglich sein.

Auf Basis der gewonnenen Erkenntnisse sowie des in diesem Kapitel analysierten Aufbaus und Modellierung des VIBN-Prototyps kann mit der operativen Durchführung der Virtuellen Inbetriebnahme begonnen werden. Diese ist Inhalt des anschließenden Kapitels.

8 DURCHFÜHRUNG DER VIRTUELLEN INBETRIEBNAHME

Kapitel 8 hat die operative Durchführung der Virtuellen Inbetriebnahme zum Inhalt. Die dafür notwendigen Voraussetzungen in Bezug auf Infrastruktur, Modellentwicklung und Modellaufbau wurden in den vorangegangenen Kapiteln ausführlich behandelt. Im Rahmen der Durchführung der VIBN wurden die Phasen EA-Check, Handbetrieb sowie Automatikbetrieb analysiert. Aufgrund der Tatsache, dass das VIBN-System im Zentrum des Forschungsinteresses stand, wurden die nachfolgend behandelten Schritte an der bereits bestehenden Produktionsanlage TH03RHB4/Betriebsartenbereich 1 vorgenommen. Abb. 82 stellt die dafür aufgebaute und eingesetzte Infrastruktur in grafischer Form dar.

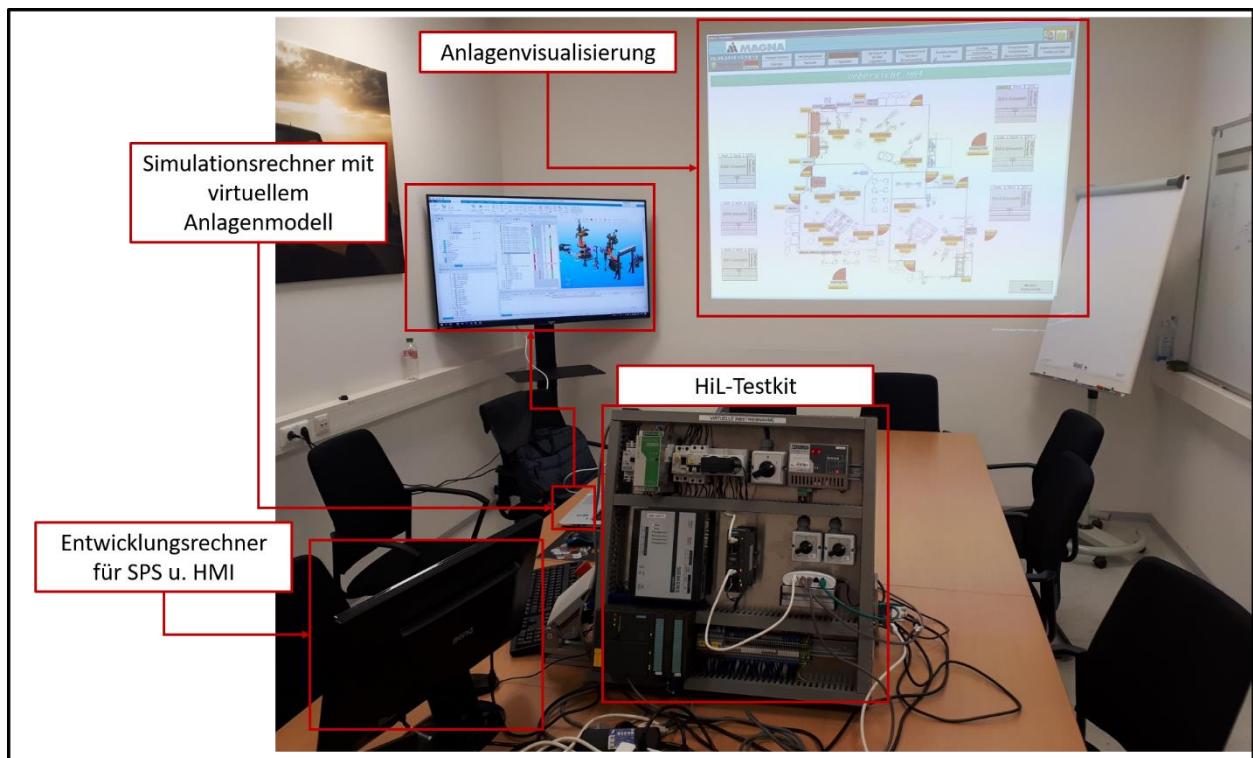


Abb. 82: Systemaufbau zur Durchführung der operativen Tätigkeiten im Rahmen der VIBN, Quelle: Eigene Darstellung.

Die Netzwerktopologie, die im Zuge der operativen Tätigkeiten angewendet wurde, kann dem Abschnitt 7.1.2 entnommen werden.

Am Beginn der operativen Aktivitäten im Rahmen der VIBN stand die Sicherstellung der Signalkorrespondenz zwischen der in der Anlage verbauten Sensorik und Aktuatorik mit der Speicherprogrammierbaren Steuerung. Diese wurde durch den Eingangs-/ Ausgangs Check (EA-Check) verifiziert und validiert.

8.1 EA-Check

Der EA-Check diente zu Sicherstellung der Erreichbarkeit der im SPS-System verbauten Sensorik und Aktuatorik durch die Speicherprogrammierbare Steuerung. Im Zuge dessen wurde das Ankommen eines jeden Eingangssignals an der SPS durch Abarbeiten einer Checkliste sichergestellt. Diese enthielt alle in der elektrischen Anlage benutzten Signale. Ebenso wurde im Rahmen dieser Validierungsmaßnahme die Erreichbarkeit eines jeden Aktuators durch die SPS überprüft sowie das gewünschte Verhalten durch Ansteuern des jeweiligen Ausgangssignals. Aufgrund der Tatsache, dass sämtliche Sensorik und Aktuatorik

durch das virtuelle Anlagenmodell abgebildet wurde, erfolgten die genannten Validierungsschritte zwischen realer SPS und virtueller Produktionsanlage. Während der Durchführung des EA-Checks am VIBN-System konnte festgestellt werden, dass die Adressierung der Eingangssignale sowie die Durchgängigkeit vom virtuellen Sensor bis zur realen HMI gegeben war. Das Eintreten von Bauteilen in den Detektionsbereich der dafür vorgesehenen Sensoren im virtuellen Anlagenmodell löste den entsprechenden Schaltvorgang des Signals aus. Dies wurde nachfolgend an der realen HMI dargestellt. Das selbe Verhalten konnte an der Sensorik zur Endlagenüberwachung von Stellgliedern wie Bauteilspanner und Schwenkeinheiten festgestellt werden. Auch das Ansteuern von pneumatischen Ventilen an der Real-HMI zur Positionsänderung von Stellgliedern hatte das gewünschte Ergebnis erzielt. Schwenk- und Spannergruppen änderten wie vorgesehen ihre Position durch Verfahren in die gewünschte Endlage. Das gleiche positive Verhalten und Ergebnis konnte bei Komponenten und Einrichtungen der Sicherheitstechnik festgestellt werden. Nach Sicherstellung der Kommunikationsdurchgängigkeit, konnte zu den Aktivitäten des Handbetriebs übergegangen werden.

8.2 Handbetrieb

Der Handbetrieb erfolgte nach Abschluss des EA-Checks. Dieser hatte den Zweck, Einzelsequenzen des Fertigungsprozesses durch manuelles Setzen von Programmeinsprünge in der jeweiligen Schrittkette des SPS-Programms auszuführen. Ebenso diente dieser Schritt dazu, die Interaktion zwischen Real-SPS, virtueller Vorrichtungstechnik und virtuellem Roboter zu prüfen und sicherzustellen. Dazu war es notwendig, die jeweiligen Roboter in die Fertigungsaufgaben einzubinden. Abb. 83 gibt die dafür erforderlichen Programmanweisungen in grafischer Form wieder.

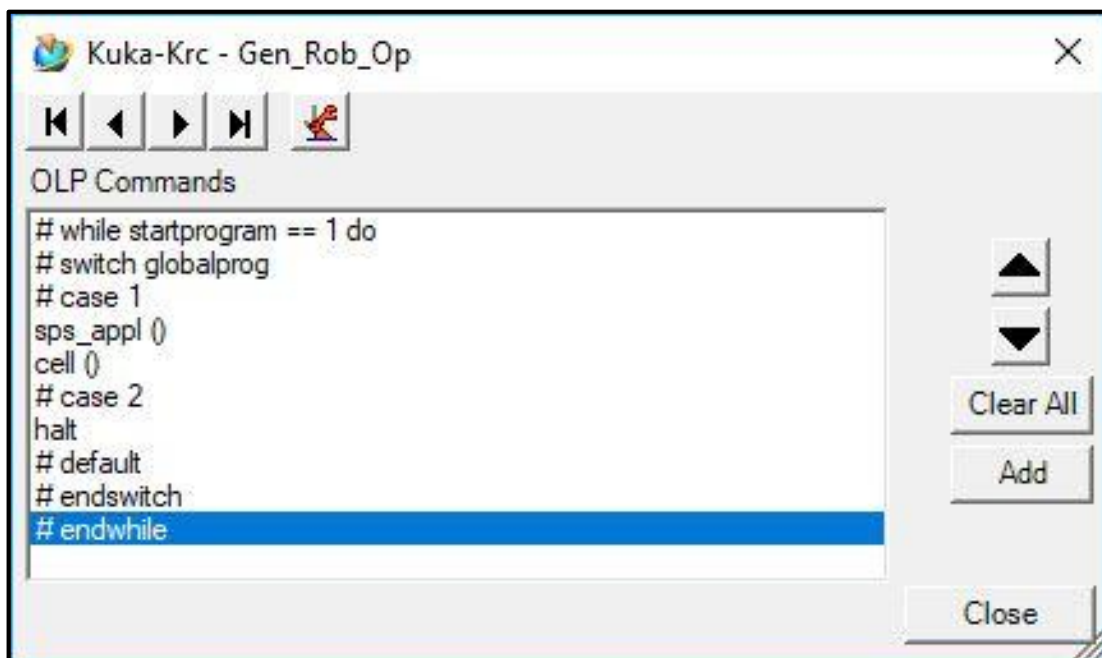


Abb. 83: Startroutine des virtuellen Roboters zur programmtechnischen Einbindung in Hand- und Automatikbetrieb, Quelle: Eigene Darstellung.

Durch die in Abb. 83 dargestellten Programmanweisungen wurde am jeweiligen virtuellen Roboter dessen Startroutine ausgeführt und der Kommunikationsaustausch mit der realen SPS aufgenommen. Danach war

es möglich, Teilprogramme am Roboter durch die Real-SPS aufzurufen. Im Zuge dessen musste festgestellt werden, dass beispielsweise die Variable *HOME* vom Simulationssystem nicht interpretiert werden konnte. Diese Variable enthält die Positionswerte der Grundstellung des Roboters. Erst durch programmübergreifendes Umbenennen der Variable in *YYHOME* konnte dies behoben werden. Diese Vorgehensweise galt es ebenfalls für die Kontrollanweisung *HALT* anzuwenden. Auf Seiten der EA-Kommunikation zwischen Roboter und SPS wurde identifiziert, dass die Anweisung *\$OUT_C[xxx]* vom virtuellen Roboter nicht ausgeführt werden konnte. Diese Anweisung zum Setzen von Ausgangssignalen hat die Besonderheit, dass diese bereits nach Erreichen des Vorlaufzeigers in der Programmbearbeitung aufbereitet, jedoch erst bei Erreichen des Hauptlaufzeigers vom Roboter ausgeführt wird. Diese musste im Zuge der Phase des Handbetriebs in eine konventionelle Ausgangsanweisung am Roboter abgeändert werden. Anschließend konnten mittels manueller Steuerung der Schrittkette im SPS-Programm Teilsequenzen des Fertigungsprozesses abgefahren werden. Dazu zählten beispielsweise das Einlegen von Bauteilen durch das Bedienpersonal, automatisches Schweißen oder Entnehmen von Baugruppen aus Schweißvorrichtungen und Bauteilablagen. Im Rahmen des Handbetriebes wurden sämtliche Programmanweisungen auf Seiten der realen SPS, der virtuellen Anlagenkomponenten sowie der virtuellen Roboter ausgeführt, die zur Erfüllung der auszuführenden Teilaufgabe nötig waren. Nach Anpassung der vorhin genannten Variablen und Anweisungen auf Seiten des virtuellen Roboters konnten diese ohne jegliche Veränderung des Steuerungsprogrammes in der Real-SPS am virtuellen Anlagenmodell ausgeführt werden. Nachfolgend und abschließend erfolgte die letzte Phase der Inbetriebnahme, der Automatikbetrieb. Dieser ist Inhalt des nachfolgenden Unterkapitels.

8.3 Automatikbetrieb

Nach erfolgreichem Durchlaufen und Abschluss des EA-Checks sowie des Handbetriebs stellte die Erreichung des fehlerfreien Automatikbetriebs den Abschluss der Inbetriebnahmeaktivitäten dar. Nach diesem ist sichergestellt, dass die Betriebsformen des Normalbetriebs, des Voll- und Leerfahrens der Produktionsanlage im Automatikbetrieb funktionieren. Auch der bauteillose Betrieb der Fertigungsbereiche sowie sämtliche Produktvarianten können nach erfolgreicher Testung des Automatikbetriebs durch das Bedienpersonal ausgeübt werden. Dafür war auf steuerungstechnischer Seite notwendig, dass sämtliche Programmteile vollständig und fehlerfrei zur Verfügung stehen. Auch war sicherzustellen, dass die Interaktion zwischen allen am Fertigungsgeschehen beteiligten Komponenten und Einrichtungen abgesichert ist. Die Erkenntnisse, die im Zuge der operativen VIBN in der Betriebsart Automatik gewonnen werden konnten, werden nachfolgend am Beispiel der Fertigungsaufgabe in Vorrichtung HB4110WZ01 veranschaulicht. Durch Aktivierung des Automatikbetriebs am Bedienpult des HiL-Testkits sowie durch Einlegen der virtuellen Bauteile in Vorrichtung HB4110WZ01 wurde die vollautomatisch durchzuführende Fertigungssequenz eingeleitet und gestartet. Nachdem die Bauteile ihre Endposition in der Vorrichtung zur Geometriebildung und Widerstandspunktschweißen eingenommen hatten, wurde dies durch die entsprechende virtuelle Sensorik an die Real-SPS rückgemeldet. Ebenso wurde die Personensicherheit des Kollaborationsraums zwischen Mensch und Maschine bei HB4110WZ01 mittels virtueller Sicherheitssensorik bestätigt. Darauf folgend aktivierte die reale SPS die erforderlichen Ausgangssignale zum Schließen des Werkerschuttores. Das Erreichen der gewünschten Endlage des Tores wurde durch die modellierte Sicherheitssensorik am Werkerschuttor dem sicherheitsgerichteten Teil der Speicherprogrammierbaren Steuerung rückgemeldet.

Nach Erfüllung der nötigen Sicherheitsbedingung wurde durch die reale SPS die Sequenz zum Schließen der pneumatisch angetriebenen Schwenker und Spanner in der richtigen Reihenfolge eingeleitet, durchgeführt und überwacht. Dies geschah durch Beschaltung virtueller Magnetspulen von Pneumatikventilen, anschließendes Ausführen kinematischer Abläufe im Simulationsmodell sowie Rückmelden virtueller Sensorsignale an die Real-SPS. Durch Interaktion zwischen realer SPS und virtuellem Anlagenmodell wurden die Stellglieder in richtiger Reihenfolge geschlossen. Nach Erreichen der gewünschten Endlagen wurden die Startbedingungen für den nächsten Fertigungsschritt gebildet. Im konkreten Anwendungsfall waren dies die Generierung der auszuführenden Programmnummer durch den Roboter HB4110IR01 sowie die Bildung der abzuarbeitenden Teilsequenznummer desselben. Dementsprechend bildete die reale SPS die Programmnummer *sechs* sowie die Fallnummer *eins*. Durch Übergabe der Programmnummer wurde dem Roboter mitgeteilt, dass dieser das Hauptprogramm *g30hb4110ir01p06* aufzurufen hat. Die Fallnummer gab an, die in *g30hb4110ir01p06* vorzufindenden Teilsequenz *SCHWEISSEN01* abzuarbeiten. Im Zuge des Zusammenspiels zur Übermittlung von Programm- und Fallnummer zwischen realer SPS und virtuellem Roboter wurde nachfolgend beschriebenes Problem festgestellt. Die Spektralfrequenz zur Detektion der Signalwerte auf Seiten des Simulationssystems zur Abbildung der virtuellen Produktionsanlage war nicht ausreichend. Erst durch Einfügen der in Abb. 84 dargestellten Codesequenz im Programm *p00_bmw* konnte die Programmnummer erfolgreich zwischen realer SPS und virtuellem Roboter in der Betriebsart *Automatik* übertragen werden.

| originale Codesequenz | Codesequenz für VIBN |
|---|--|
| <pre> CASE 1 ;***** FOR W=0 TO PGNO_LENGTH-1 IF \$IN[PGNO_FBIT+W] THEN PGNO=PGNO+J ENDIF J=J*2 ENDFOR ;***** </pre> | <pre> CASE 1 ;***** FOR W=0 TO PGNO_LENGTH-1 IF \$IN[PGNO_FBIT+W] THEN PGNO=PGNO+J WAIT FOR PGNO > 1 ENDIF J=J*2 ENDFOR ;***** </pre> |

Abb. 84: Notwendige Programmänderung zur Erreichung der fehlerfreien Interaktion zwischen Real-SPS und virtuellem Roboter, Quelle: Eigene Darstellung.

Nach erfolgreicher Übertragung der Programmnummer von der SPS an den Roboter HB4110IR01 führte dieser die abzuarbeitenden Programme entsprechend der vorgegeben Auftragserfüllung aus. Im Zuge des Zusammenspiels zwischen realer SPS und virtuellem Roboter während dessen Aufgabenerfüllung in Vorrichtung HB4110WZ01 traten keine Auffälligkeiten auf. Der virtuelle Roboter meldete an die reale SPS, dass er seine erste Teilschweißung abgeschlossen hatte, die reale SPS stellte nachfolgend die Stellglieder der Vorrichtung HB4110WZ01 in jene Positionen, die für das ordnungsgemäße Verschweißen der Bauteile erforderlich war. Auch die anschließende Entnahme der Bauteile aus der Vorrichtung HB4110WZ01 erfolgte ohne Komplikationen wie auch das Ablegen der Baugruppe in die Zwischenablage HB4115SR01. Dieser Schritt war zugleich die letzte Teilaufgabe, die der virtuelle Roboter im Rahmen des Programm-

aufrufs von g30hb4110ir01p06 abzuarbeiten hatte. Im Zuge des letzten Abarbeitungsschrittes des Roboters war jene Problematik festzustellen, die im abschließenden Absatz erörtert wird.

Die Bauteilablage HB4115SR01 stellte einen Kollaborationsraum zwischen den Robotern HB4110IR01 und HB4120IR01 dar. Zur Sicherstellung des Maschinenschutzes respektive zur Vermeidung einer Kollision zwischen den beiden Robotern im Zuge ihrer Auftragserfüllung sind in den Roboterprogrammen Verriegelungszonen an geeigneter Stelle vorzusehen. Diese Verriegelungszonen werden von der SPS verwaltet. Der Roboter hat der SPS mitzuteilen, dass er in eine bestimmte Verriegelungszone einfahren möchte. Die SPS erteilt entweder die Freigabe zum Einfahren oder verweigert diese, falls bereits ein anderer Roboter diese freigegeben bekommen hat. Diese Methodik beziehungsweise Prozedere zur Wahrung des Maschinenschutzes ist in allen Schritten der virtuellen Absicherung vorzunehmen, jedoch in unterschiedlicher programmtechnischer und syntaktischer Umsetzungsform, wie aus Abb. 85 ersichtlich ist.

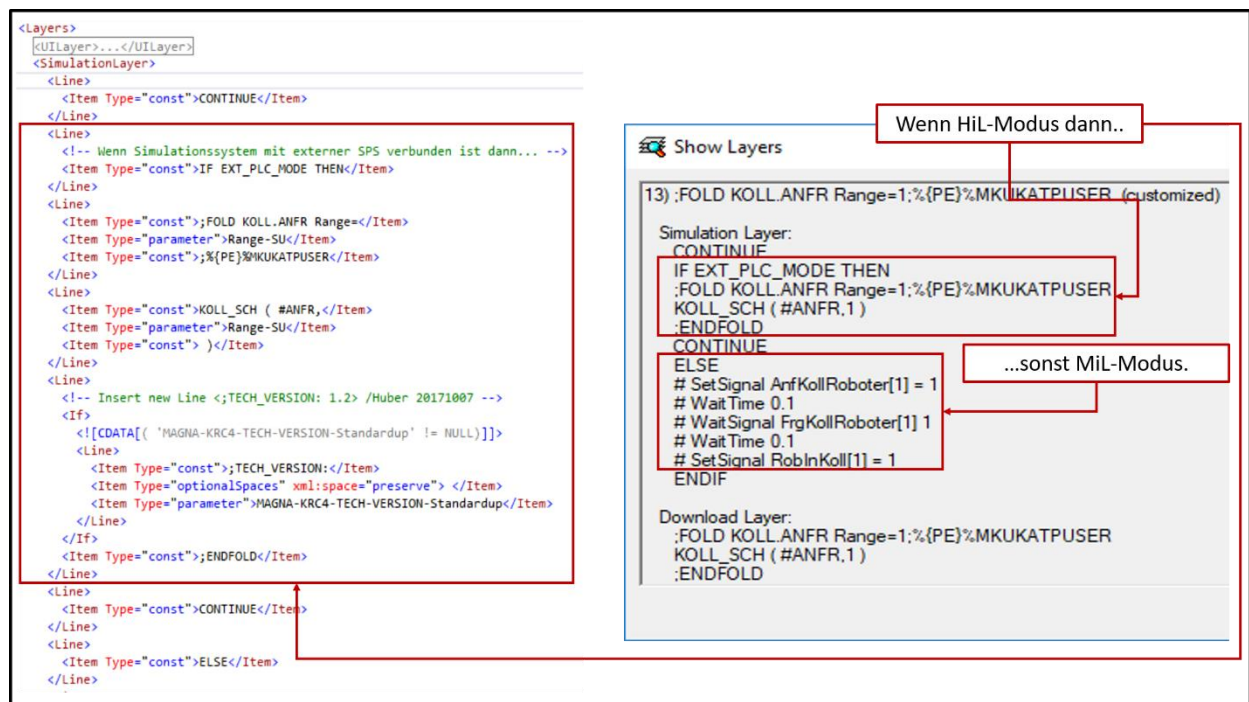


Abb. 85: Programmtechnische Lösung zur Unterscheidung zwischen HiL u. MiL am virtuellen Roboter, Quelle: Eigene Darstellung.

Daher musste dem virtuellen Roboter mitgeteilt werden, in welchen Absicherungsmodus er eingebunden ist. Aus Abb. 85 wird ersichtlich, dass im Zuge der Durchführung der operativen VIBN das Ausgangssignal *EXT_PLC_MODE* an der realen SPS hinzugefügt wurde. Damit wurde dem virtuellen Roboter mitgeteilt, ob er in ein MiL-System eingebunden ist, oder mit einem HiL-System interagiert. Dies ermöglichte nachfolgend der virtuellen Robotersteuerung zu unterscheiden, ob sie die native Codesequenz zum Kollisionschutz auszuführen hat, oder jene Befehle, die der MiL-Modus erfordert. Diese Vorgehensweise wurde für alle weiteren Programmteile vorgenommen, welche syntaktische Unterschiede zwischen MiL und HiL aufwiesen. Nach Anpassung der in Abb. 85 dargestellten Codesequenz konnte die Fertigungsaufgabe in der Betriebsart *Automatik* abgeschlossen werden, ohne weitere Anpassungen an der Real-SPS vorzunehmen. Weitere Erkenntnisse im Rahmen der operativen Durchführung der VIBN werden in der abschließenden Zusammenfassung dieses Kapitels behandelt.

8.4 Zusammenfassung

Durch die operative Durchführung der Virtuellen Inbetriebnahme konnte festgestellt werden, dass die Interaktion zwischen realer SPS, virtueller Sensorik und virtueller Aktuatorik einfach herzustellen ist. Ebenso konnte die Erkenntnis gewonnen werden, dass von Eingangs- und Ausgangsignalen dieser virtuellen Anlagenkomponenten kein Störverhalten ausgeht. Sämtliche Änderungen am Signalstatus im virtuellen Anlagenmodell werden zuverlässig detektiert und an die Real-SPS kommuniziert. Ebenso werden Anweisungen aus der realen HMI ordnungsgemäß im virtuellen Anlagenmodell ausgeführt. Es sind auf Seiten der realen SPS sowie der realen HMI keine Anpassungen respektive Änderungen nötig, damit diese ohne Einschränkung mit der virtuellen Produktionsanlage interagieren können.

Die Einbindung von virtuellen Robotersystemen in das VIBN-System setzt wesentlich mehr Anpassungsaufwand voraus. Ursache für diesen Aufwand ist der Umstand, dass der Programmcode des realen Roboters teilweise von dessen virtuellem Äquivalent nicht ausgeführt werden kann. Dies hat zur Folge, dass beispielsweise Variablen im Robotercode umbenannt werden müssen, damit die virtuelle Robotersteuerung respektive das Simulationssystem die Programmanweisung ordnungsgemäß ausführen kann. Ebenso sind bestimmte Formen von Anweisungen zum Setzen von Ausgangssignalen von der virtuellen Robotersteuerung nicht möglich. Auch die Tatsache, dass teilweise die Voraussetzungen zur Wahrung des nötigen Abtasttheorems für binäre Signale vom Simulationssystem nicht gewährleistet werden, führt zu Mehraufwänden. Einerseits sind diese Mehraufwände durch die Ursachenanalyse bedingt, andererseits in der erforderlichen Entwicklung von alternativem ausführbarem Robotercode. Aufgrund dieser Tatsache sind alle Roboterprogramme einer tiefgreifenden Analyse hinsichtlich Einbindungsnotwendigkeit in die Virtuelle Inbetriebnahme sowie VIBN-Tauglichkeit zu unterziehen. Die Einbindungsnotwendigkeit ist dahingehend zu untersuchen, da jedes Roboterprogramm, welches der virtuellen Robotersteuerung zugeordnet wird, im Zuge der VIBN vom Simulationssystem Bit für Bit abgearbeitet und ausgeführt wird. Hier ist zu hinterfragen, ob dieses Roboterprogramm oder Teile davon für die erfolgreiche Zielerreichung der Virtuellen Inbetriebnahme nötig sind. Als Beispiel hierzu können Programmteile angeführt werden, die Informationen, Betriebsmeldungen und Störmeldungen an das Bedienhandgerät der Robotersteuerung ausgeben. Dementsprechend ist auch in Bezug auf die Einbindungsnotwendigkeit von Roboterprogrammen die Festlegung entsprechender Systemgrenzen zu empfehlen, wie sie auch bereits in Abschnitt 7.5.1 vorgenommen wurde. Betreffend VIBN-Tauglichkeit der Roboterprogramme wird empfohlen, diese so umzugestalten, dass sie sowohl vom realen Roboter fehlerfrei ausgeführt werden können wie auch von dessen virtuellem Substitut. Alternativ dazu besteht die Möglichkeit, den bestehenden Programmierstandard für die Roboterprogrammierung zu duplizieren und gezielt für die VIBN zu optimieren. Damit wird sichergestellt, dass die Virtuelle Inbetriebnahme effizient durchgeführt werden kann. Ebenso kann damit das volle Potential der VIBN in Bezug auf das Verhältnis von Aufwand und Nutzen generiert werden.

Das nachfolgende und letzte Kapitel dieser Masterarbeit fasst die gewonnen Erkenntnisse zusammen, die im Zuge der wissenschaftlichen Auseinandersetzung mit dem Themenfeld der Virtuellen Inbetriebnahme gewonnen werden konnten.

9 ZUSAMMENFASSUNG, FAZIT UND AUSBLICK

Kapitel 9 fasst die gewonnenen Erkenntnisse und Feststellungen aus den vorangegangenen Kapiteln, Unterkapiteln und Abschnitten zusammen. Im Zuge der wissenschaftlichen Auseinandersetzung zur Beantwortung der forschungsleitenden Fragestellungen wie auch den von Magna Steyr vorgegebenen Zielsetzungen wurden theoretische Analysen durchgeführt und ein praktischer Versuchsaufbau installiert. Im ersten Block dieser Masterarbeit wurden die Themen *Modellbildung* und *Virtuelle Inbetriebnahme* einer theoretischen Auseinandersetzung unterzogen. Die Ergebnisse daraus bildeten neben den Erkenntnissen aus der Iststandsanalyse bei Magna Steyr die Grundlage der zu lösenden praktischen Aufgabenstellungen. Einerseits galt es die Frage zu beantworten, in welcher Phase der Projektabwicklung die Methode respektive das Werkzeug der Virtuellen Inbetriebnahme zu positionieren ist. Diese Fragestellung wurde in Kapitel 5 untersucht und beantwortet. Andererseits galt es eine Methode und ein Werkzeug zu entwickeln, das den formulierten Zielvorgaben von Magna Steyr entsprach. Die erste Zielvorgabe beinhaltete die Aufgabe ein System zu entwickeln, das die frühzeitige Verifizierung und Validierung von Steuerungskonzepten und Programmcode von Steuerungen ermöglicht, die hochautomatisierte Fertigungsanlagen von Karosseriebauteilen steuern, regeln und überwachen. Die zweite Zielvorgabe war ein Werkzeug zu entwickeln, das die Analyse von komplexen Störungsszenarien und Ausfällen von Produktionsanlagen ermöglicht, ohne den Fertigungsprozess zu beeinträchtigen. Für beide Fragestellungen galt es die Prämisse einzuhalten, weitestgehend originale Steuerungskomponenten und nativen Steuerungscode im Rahmen der Lösungsfindung einzusetzen. Die Fragestellungen konnten durch die in den Kapiteln 6 bis 8 erläuterten Forschungs- und Entwicklungsaktivitäten beantwortet werden. Konkret wurden diese Fragen durch die Schaffung einer geeigneten Infrastruktur beantwortet, die die modellgestützte Virtuelle Inbetriebnahme von Produktionsanlagen erlaubt, wie sie bei Magna Steyr betrieben werden. Basierend darauf wurde ein Prototyp entwickelt und errichtet, der die frühzeitige Verifizierung und Validierung von Steuerungskonzepten sicherstellt, die zur Aufgabenerfüllung eine Speicherprogrammierbare Steuerung einsetzen. Dieser Prototyp kann ebenfalls dazu herangezogen werden, komplizierte Störungen und Ausfälle von Produktionsanlagen im Nachhinein zu analysieren.

Das aus den theoretischen wie auch praktischen Forschungstätigkeiten resultierende Fazit wird im anschließenden Unterkapitel erläutert.

9.1 Fazit

Modellbildung und der Entstehungsprozess von Modellen ist eine Teamaufgabe. Dadurch wird sichergestellt, dass der zu erzielende Modellzweck im Fokus der Modellentwicklung steht und bestehen bleibt. Ebenso ist das Festlegen geeigneter Systemgrenzen im Rahmen der Modellbildung hinsichtlich Effizienz während der Modellentstehung wie auch Leistungsfähigkeit des Modells im operativen Einsatz entscheidend. Dies gilt ebenfalls hinsichtlich der geeignetsten Methode zu Bildung von Modellen. Ebenso ist es durch Einsatz und gezielte Anwendung unterschiedlicher Modellformen möglich, die Leistungsfähigkeit des Modells positiv zu beeinflussen. Dies wird durch Anwendung geeigneter Abstraktionsprinzipien zur Formulierung modellspezifischer Zusammenhänge zusätzlich verstärkt. Durch Analyse und Vergleich der Begriffe Inbetriebnahme, Virtuelle Inbetriebsetzung und Virtuelle Inbetriebnahme kann die Erkenntnis abgeleitet werden, dass ausschließlich der Begriff der Virtuellen Inbetriebnahme modellgestützte Steuerungstests am

Ende der Produktentstehungsphase repräsentiert¹⁶³. Auch das Wissen über unterschiedliche Methoden zur Durchführung einer Virtuellen Inbetriebnahme ist von Bedeutung. Dadurch ist die Möglichkeit gegeben, die zur Erfüllung des Untersuchungsinteresses geeignetste VIBN-Methode auszuwählen. Hinsichtlich der dazu anzuwendenden Auswahlkriterien sind das Wissen und die richtige Einordnung von Echtzeit, Abtasttheorem wie auch der virtuellen Zeitachse ausschlaggebend. Das entwickelte VIBN-System, das auf der Methode Hardware-in-the-Loop basiert, erfüllt die von Magna Steyr formulierten Zielvorgaben. Des Weiteren ist die Erkenntnis darüber, dass virtuelle Anlagenkomponenten einfach für den VIBN-Einsatz zu qualifizieren sind, bedeutend. Ausgenommen hiervon sind jedoch Modelle zur Abbildung virtueller Roboter. Ebenso stellen die unterschiedlichen Bezeichnungsmethodiken von Anlagenkomponenten zwischen Mechanik und Elektrotechnik bei Magna Steyr eine Herausforderung dar. Ohne Harmonisierung dieser wird eine Verschiebung im Verhältnis zwischen Aufwand und Nutzen einer Virtuellen Inbetriebnahme zu Gunsten deren Durchführung uneindeutig bleiben. Ist das Modell zur Abbildung der virtuellen Anlage jedoch aufgebaut, erfüllt das Instrument der Virtuellen Inbetriebnahme alle Forderungen, die von Seiten Magna Steyr zur Zielerreichung vorgegeben wurden.

9.2 Ausblick

Basierend auf dem in dieser Arbeit gewonnenen Wissen, wird die Virtuelle Inbetriebnahme in naher Zukunft als Standardwerkzeug im Entstehungsprozess von Produktionsanlagen bei Magna Steyr eingesetzt werden. Dies lässt sich dahingehend begründen, dass durch Angleichung der Bezeichnungsmethodik zwischen mechanischer und elektrischer Konstruktion ein effizienter Modellaufbau möglich sein wird. Diese Anpassung stellt sicher, dass zukünftig Anlagenmodelle zur Durchführung einer Virtuellen Inbetriebnahme teilautomatisiert erstellt werden. Neben dem Aufbau einer interdisziplinären Komponentenbibliothek zur Abbildung mechatronischen Verhaltens stellt dies jene Maßnahme dar, die das Verhältnis zwischen Aufwand und Nutzen betreffend Durchführung einer Virtuellen Inbetriebnahme eindeutig zu deren Gunsten verschieben wird. Die Leistungsfähigkeit wie auch das Potential des Werkzeugs der Virtuellen Inbetriebnahme kann im Rahmen der operativen Anwendung in dieser Masterarbeit bestätigt werden. Dieses Potential lässt sich nicht nur durch weiter steigende Rechenleistung der zur Abbildung der virtuellen Produktionsanlage heranzuziehenden Rechnersysteme begründen. Diese Annahme beruht ebenso auf der Tatsache, dass von Seiten der Anbieter von Simulationssoftware wie auch der Hersteller von Speicherprogrammierbaren Steuerungen intensiv an der Weiterentwicklung von Möglichkeiten zur VIBN gearbeitet wird. So ist es zum Beispiel bei Einsatz von SPS-Systemen neuester Generation bereits möglich, die SPS selbst vollständig virtuell abzubilden und mittels PLCSim-Advanced an Systeme zur Abbildung virtueller Produktionseinrichtungen anzubinden.¹⁶⁴ Auch die Tatsache, dass die Methode der Virtuellen Inbetriebnahme auf unterschiedlichsten Steuerungsebenen angewandt werden kann, wird den zukünftigen Einsatz dieses Instruments zusätzlich beschleunigen und intensivieren.

¹⁶³ Vgl. Wunsch (2007), S. 32 f.

¹⁶⁴ Vgl. WEKA FACHMEDIEN GmbH (2018), Online-Quelle [03.11.2018].

LITERATURVERZEICHNIS

Gedruckte Werke (63)

- Abel, Dirk; Bollig, Alexander (2006): *Rapid Control Prototyping - Methoden und Anwendungen*, Springer-Verlag, Berlin
- Alt, Oliver (2009): *Car Multimedia Systeme Modell-basiert testen mit SysML*, 1. Auflage, Vieweg+Teubner | GWV Fachverlage GmbH, Wiesbaden
- Amann, Wolfgang (1994): *IWB Forschungsbericht Band 71 - Eine Simulationsumgebung für Planung und Betrieb von Produktionssystemen*, Springer-Verlag GmbH, Berlin
- Arnold, Dieter u. a. (2008): *Handbuch Logistik*, 3. Auflage, Springer-Verlag, Berlin
- Bandow, Gerhard (2010): „Das ist gar kein Modell!“ *Unterschiedliche Modelle und Modellierungen in Betriebswirtschaftslehre und Ingenieurwissenschaften*, 1. Auflage, Gabler | GWV Fachverlage, Wiesbaden
- Beucher, Ottmar (2015): *Signale und Systeme: Theorie, Simulation, Anwendung*, 2. Auflage, Springer-Verlag, Heidelberg
- Bohlin, Torsten (2006): *Practical Grey-box Process Identification*, 1 Auflage, Springer-Verlag, London
- Bohn, Christian; Unbehauen, Heinz (2016): *Identifikation dynamischer Systeme*, 1. Auflage, Springer Vieweg, Wiesbaden
- Böker, Andreas; Paerschke, Hartmuth; Boggasch, Ekkehard (2017): *Elektrotechnik für Gebäudetechnik und Maschinenbau*, Springer Vieweg, Wiesbaden
- Bossel, Hartmut (1989): *Simulation dynamischer Systeme*, 1. Auflage, Springer Fachmedien, Wiesbaden
- Bossel, Hartmut (1992): *Modellbildung und Simulation*, 1. Auflage, Vieweg, Braunschweig / Wiesbaden
- Botthof, Alfons; Hartmann, Ernst Andreas (2015): *Zukunft der Arbeit in Industrie 4.0*, Springer, Berlin
- Bracht, Uwe; Geckler, Dieter; Wenzel, Sigrid (2011): *Digitale Fabrik*, 1. Auflage, Springer-Verlag, Berlin
- Brökelmann, Jan (2015): *Systematik der virtuellen Inbetriebnahme von automatisierten Produktionssystemen*, Verlagshaus Monsenstein und Vannerdat OHG, Paderborn
- Broy, Manfred; Steinbrüggen, Ralf (2004): *Modellbildung in der Informatik*, Springer-Verlag, Heidelberg
- Bungartz u. a., Hans-Joachim (2009): *Modellbildung und Simulation - Eine anwendungsorientierte Einführung*, 1. Auflage, Springer-Verlag, Berlin Heidelberg
- Claussen, Ute (1998): *Objektorientiertes Programmieren*, 2. Auflage, Springer, Berlin
- Dori, Dov (2016): *Model-Based Systems Engineering with OPM and SysML*, Springer Science+Business Media, New York
- Duelen, G.; Linnemann, H.; Bernhardt, R. (1986): *Die Informationsarchitektur in datengetriebenen Fabriken*, Produktionstechnisches Kolloquium, Berlin

-
- Eigner, Martin; Roubanov, Daniil; Zafirov, Radoslav (2014): *Modellbasierte Virtuelle Produktentwicklung*, Springer-Verlag, Heidelberg 2014
- Glöckler, Michael (2018): *Simulation mechatronischer Systeme*, 2. Auflage, Springer Vieweg, Wiesbaden
- Gutenschwager, Kai u. a. (2017): *Simulation in Produktion*, 1. Auflage, Springer Vieweg, Berlin
- Halang, Wolfgang A.; Konakovsky, Rudolf M. (2018): *Sicherheitsgerichtete Echtzeitsysteme*, 3. Auflage, Springer-Verlag, Hagen
- Huber, Andreas (2017): *Systemauswahl und Konzeptfestlegung für die Einführung der Virtuellen Inbetriebnahme von hochautomatisierten Karosseriebauanlagen*, Campus02 - Fachhochschule der Wirtschaft, Graz, Bachelorarbeit
- Husty, Manfred u. a. (1997): *Kinematik und Robotik*, Springer-Verlag Berlin Heidelberg, Heidelberg
- Janschek, Klaus (2010): *Systementwurf mechatronischer Systeme*, 1. Auflage, Springer, Berlin / Heidelberg
- Kahlbrandt, Bernd (2001): *Software-Engineering mit der Unified Modeling Language*, 2. Auflage, Springer-Verlag, Berlin
- Kastens, Uwe; Kleine-Büning, Hans (2005): *Modellierung - Grundlagen und formale Methoden*, 1. Auflage, Carl Hanser Verlag, München / Wien
- Kiefer, Jens (2007): *Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau*, Universität des Saarlandes, Saarbrücken, Dissertation
- Klaus, Georg (1963): *Kybernetik in philosophischer Sicht*, Bern
- Kleuker, Stephan (2018): *Grundkurs Software-Engineering mit UML*, 4. Auflage, Springer Fachmedien, Wiesbaden
- Koch, Anton Friedrich; Schick, Friederike (Hrsg.) (2002): *Wissenschaft und Logik*, Akademie Verlag, Berlin
- Lacour, Frédéric-Felix Robert (2011): *Modellbildung für die physikbasierte Virtuelle Inbetriebnahme materialflussintensiver Produktionsanlagen*, Herbert Utz Verlag, München
- Milberg, Joachim (Hrsg.) (1992): *Von CAD/CAM zu CIM*, Springer-Verlag, Heidelberg
- Müller, Jörg Rudolf (2015): *Die Formalisierte Terminologie der Verlässlichkeit Technischer Systeme*, Springer-Verlag, Berlin
- Müller, Rolf (2015): *Modellierung, Analyse und Simulation elektrischer und mechanischer Systeme mit Maple™ und MapleSim™*, Springer Vieweg, Wiesbaden
- Oestereich, Bernd (2002): *Die UML Kurzreferenz – kurz, bündig, ballastfrei.*, Oldenburgverlag, München / Wien
- Page, Bernd; Kreutzer, Wolfgang (2005): *The Java Simulation Handbook*, Shaker Verlag, Aachen
- Pahl, Gerhard u. a. (2007): *Konstruktionslehre*, 7. Auflage, pringer-Verlag Berlin Heidelberg, Heidelberg

-
- Reinhardt, Helmut (1996): *Automatisierungstechnik - Theoretische und gerätetechnische Grundlagen*, SPS, Springer-Verlag, Heidelberg
- Reinschke, Kurt (2014): *Lineare Regelungs- und Steuerungstheorie*, 2. Auflage, Springer Vieweg, Heidelberg
- Rill, Georg; Schaeffer, Thomas (2017): *Grundlagen und Methodik der Mehrkörpersimulation*, 3. Auflage, Springer Vieweg, Wiesbaden
- Roddeck, Werner (2017): *Grundprinzipien der Mechatronik*, 2. Auflage, Wiesbaden
- Ropohl, Günter (2009): *Allgemeine Technologie - Eine Systemtheorie der Technik*, 3. Auflage, Universitätsverlag Karlsruhe, Karlsruhe
- Rumpe, Bernhard (2016): *Modeling with UML - Language, Concepts, Methods*, Springer International Publishing Switzerland, Heidelberg
- Schenk, Michael; Schumann, Marco (2016): *Angewandte Virtuelle Techniken im Produktentstehungsprozess*, Springer Vieweg, Berlin
- Scholz, Peter (2005): *Softwareentwicklung eingebetteter Systeme - Grundlagen, Modellierung, Qualitätssicherung*, Springer-Verlag Berlin Heidelberg, Berlin
- Seemann, Jochen; von Gudenberg, Jürgen Wolff (2006): *Software-Entwurf mit UML 2*, 2. Auflage, Springer Berlin Heidelberg New York, Berlin
- Seidl, Martina u. a. (2015): *UML @ Classroom - An Introduction to Object-Oriented Modeling*, Springer International Publishing Switzerland, Heidelberg,
- Siemens AG (Hrsg.) (07/2017): *SIMATIC SIMULATIONUnit V9.1 Systemhandbuch*, Nürnberg
- Smith J.M.; Smith D.C.P (1977): *Data base abstractions: aggregation and generalization*
- Stachowiak, Herbert (1973): *Allgemeine Modelltheorie*, 1. Auflage, Springer-Verlag, Wien
- Staud, Josef L. (2005): *Datenmodellierung und Datenbankentwurf - Ein Vergleich aktueller Methoden*, Springer, Berlin
- Strassacker, Dirk (1997): *ISW Forschung und Praxis Band 117 - Testumgebung und Inbetriebnahme eines adaptierbaren Leitsteuerungssystems*, Springer-Verlag, Berlin
- Vajna, Sandor u. a. (2009): *CAX für Ingenieure - Eine praxisbezogene Einführung*, 2. Auflage, Springer-Verlag Berlin Heidelberg, Heidelberg
- Vinek, Gunther; Rennert, Paul Frederick; Tjoa, A Min (1982): *Datenmodellierung: Theorie und Praxis des Datenbankentwurfs*, Physica-Verlag, Würzburg-Wien
- Weber, Klaus H. (2006): *Inbetriebnahme verfahrenstechnischer Anlagen*, 3. Auflage, Springer-Verlag, Heidelberg
- Weber, Wolfgang (2009): *Industrieroboter - Methoden der Steuerung und Regelung*, 2 Auflage, Carl Hansa Verlag, München

Winter, Robert (1991): *Mehrstufige Produktionsplanung in Abstraktionshierarchien auf der Basis relationaler Informationsstrukturen*, Springer-Verlag, Berlin

Wünsch, Georg (2007): *Methoden der virtuellen Inbetriebnahme automatisierter Produktionssysteme*, Technische Universität München, München

Zäh, M. F.; Reinhart, G. (2004): *Virtuelle Produktionssystem-Planung: Virtuelle Inbetriebnahme und Digitale Fabrik*, Herbert Utz Verlag, München

Zäh, Michael; Reinhart, Gunter (Hrsg.) (2006): *Seminarberichte Virtuelle Inbetriebnahme - Von der Kür zur Pflicht?*, Herbert Utz Verlag GmbH, Garching

Zeugträger, Karsten (1998): *Anlaufmanagement für Großanlagen. Dissertation Universität Hannover*, VDI Verlag, Hannover

Wissenschaftliche Artikel (5)

Bruni, Roberto u. a. (2015): *Reconciling White-Box and Black-Box Perspectives on Behavioral Self-adaptation*, in: Springer International Publishing Switzerland, 1/2015, Department of Computer Science, University of Pisa, S. 163-184

Elkind, Edith u. a. (2006): *Grey-Box Checking*, in: International Federation for Information Processing, 1/2006, Department of Computer Science, Warwick, S. 420-435

Kojarski, Sergei; Lorenz, David H. (2006): *Comparing White-Box, Black-Box, and Glass-Box*, in: Springer-Verlag Berlin Heidelberg, 1/2006, University of Virginia, Charlottesville, S. 246-259

Wiendahl, Hans-Peter; Hegenscheidt, M.; Winkler, H. (2002): *Anlaufrobuste Produktionssysteme*, in: wt Werkstattstechnik 92, 11/12/2002, Springer-VDI-Verlag, S. 650-655

Zhang, Hao; Nakadai, Shinji; Fukumizu, Kenji (2018): *From Black-Box to White-Box: Interpretable Learning with Kernel Machines*, in: Springer International Publishing AG, part of Springer Nature, 1/2018, NEC Corporation, Tokyo, S. 213-227

Online-Quellen (7)

Siemens AG (2018): *Factory Acceptance Test*

<https://w3.siemens.com/smartgrid/global/en/products-systems-solutions/smart-communication/building-services/Pages/factory-acceptance-test.aspx> [Stand: 28.09.2018]

Siemens AG (2018): *Siemens Industrie Mall*

<https://mall.industry.siemens.com/mall/de/WW/Catalog/Product/6ES7318-3FL01-0AB0>
[Stand: 15.09.2018]

Siemens AG (2018): *Siemens Industry Online Support Produkt Support*

<https://support.industry.siemens.com/cs/document/12996906/simatic-s7-300-cpu-31xc-und-cpu-31x%3A-technische-daten?dti=0&lc=de-WW> [Stand: 15.09.2018]

Siemens AG (2018): *SIMATIC PCS 7 Tutoria I*
https://w3.siemens.com/mcms/process-control-systems/de/simatic-pcs-7/simaticpcs7tutorial/pages/default.aspx#Grundlagen_20zu_20SIMATIC_20PCS_207
[Stand: 28.09.2018]

Siemens Product Lifecycle Management Software (2018): *Process Designer*
https://www.plm.automation.siemens.com/de/products/tecnomatix/manufacturing-planning/process-design/designer.shtml#lightview%26url=/de_de/Images/7456_tcm73-4941.pdf%26title=Process%20Designer%26description=Process%20Designer%20Fact%20Sheet%26doctype=pdf [Stand: 15.09.2018]

Siemens Product Lifecycle Management Software (2018): *Process Simulate for Robotics and Automation*
https://www.plm.automation.siemens.com/de/products/tecnomatix/manufacturing-simulation/robotics/process-simulate.shtml#lightview%26url=/de_de/Images/7457_tcm73-80351.pdf%26title=Process%20Simulate%26description=Process%20Simulate%20Fact%20Sheet%26doctype= [Stand: 15.09.2018]

WEKA FACHMEDIEN GmbH (2018): *smarterworld.de*
<https://www.smarterworld.de/smart-automation-iot/iot-industrie-40/artikel/129265/1/> [Stand: 03.11.2018]

Normen (18)

Deutsche Kommission Elektrotechnik im DIN und VDE (Hrsg.) (Januar 2008): *DIN EN 62264-3: Integration von Unternehmens-EDV und Leitsystemen - Teil 3: Aktivitätsmodelle für das operative Produktionsmanagement*

Deutsche Kommission Elektrotechnik im DIN und VDE (Hrsg.) (Juli 2013): *DIN IEC 60050-351: Internationales Elektronisches Wörterbuch - Teil 351: Leittechnik*

Deutsche Kommission Elektrotechnik im DIN und VDE (Hrsg.) (Juni 2008): *DIN EN 62264-1: Integration von Unternehmensführungs- und Leitsystemen - Teil 1: Modelle und Terminologie*

Deutsche Kommission Elektrotechnik im DIN und VDE (Hrsg.) (März 2004): *DIN EN 61131-1: DIN EN 61131-1 Speicherprogrammierbare Steuerungen – Teil 1: Allgemeine Informationen*

Deutsches Institut für Normung e. V. (Hrsg.) (September 2012): *DIN 31051: Grundlagen der Instandhaltung*

Deutsches Institut für Normung e. V. (Hrsg.) (Mai 1977): *DIN 32541: Betreiben von Maschinen und vergleichbaren technischen Arbeitsmitteln*

Europäisches Komitee für Elektrotechnische Normung (Hrsg.) (September 2007): *DIN EN 62337: Inbetriebnahme elektrischer und leittechnischer Systeme in der Prozessindustrie - Phasen und Meilensteine*

International Commission Electrotechnical (Hrsg.) (2001-07): *IEC 60050-151: International Electrotechnical Vocabulary*

International Organization for Standardization; International Electrotechnical Commission (Hrsg.) (2015): *ISO/IEC 2382:2015(en): Information technology — Vocabulary*

Verein Deutscher Ingenieure (Hrsg.) (Dezember 1993): *VDI 3633 Blatt 1: Simulation von Logistik-, Materialfluß und Produktionssystemen*

Verein Deutscher Ingenieure (Hrsg.) (Februar 2018): *VDI 3843 Blatt 1: Modellbildung schwingungsfähiger Systeme*

Verein Deutscher Ingenieure (Hrsg.) (Juni 2004): *VDI 2206: Entwicklungsmethodik für mechatronische Systeme*

Verein Deutscher Ingenieure (Hrsg.) (Mai 2016): *VDI 4465 Blatt 1 (Entwurf): Modellierung und Simulation - Modellbildungsprozess*

Verein Deutscher Ingenieure (Hrsg.) (Oktober 1987): *VDI 2235: Wirtschaftliche Entscheidungen beim Konstruieren - Methoden und Hilfen*

Verein Deutscher Ingenieure, Verein (Hrsg.) (März 2009): *VDI 2209: 3-D-Produktmodellierung*

Verein Deutscher Ingenieure (Hrsg.) (August 2016): *VDI/VDE 3693 Blatt 1: Virtuelle Inbetriebnahme - Modellarten und Glossar*

Verein Deutscher Ingenieure (Hrsg.) (Februar 2008): *VDI 4499 Blatt 1: Digital Fabrik - Grundlagen*

Verein Deutscher Ingenieure (Hrsg.) (Mai 2011): *VDI 4499 Blatt 2: Digital Fabrik - Digitaler Fabrikbetrieb*

ABBILDUNGSVERZEICHNIS

| | |
|---|----|
| Abb. 1: Aufbau der Arbeit wie auch Vorgehensweise zur Beantwortung der forschungsleitenden Fragestellung, Quelle: Eigene Darstellung..... | 2 |
| Abb. 2: Hauptmerkmale des allgemeinen Modells, Quelle: Bandow (2010), S. 9 (leicht modifiziert). | 4 |
| Abb. 3: Erforderlicher Bezug zwischen Modell und Original, Quelle: Kastens/Kleine-Büning (2005), S. 20 (modifiziert). | 5 |
| Abb. 4: Grundkonzept eines Systems, Quelle: Bossel (1989), S. 11 (leicht modifiziert). | 6 |
| Abb. 5: Iterativer Prozess der Modellbildung und Modellentwicklung, Quelle: Bossel (1989), S. 13 (leicht modifiziert). | 7 |
| Abb. 6: Modellbildungsprozess basierend auf Erkenntnis- u. Wissenszuwachs, ... Quelle: VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 2 (leicht modifiziert). | 8 |
| Abb. 7: Makroprozess des V-Modells zu domänenübergreifenden Entwicklung von mechatronischen Komponenten, Quelle: VDI 2206 (Juni 2004), S. 81 (leicht modifiziert). | 9 |
| Abb. 8: Hauptblöcke des Modellentstehungsprozesses, Quelle: VDI 3843 Blatt 1 (Februar 2018), S. 6 (modifiziert). | 10 |
| Abb. 9: Schritte zur Modellbildung und Systemanalyse, Quelle: Bossel (1989), S. 30 (leicht modifiziert). | 15 |
| Abb. 10: Modellbildungszyklus, Quelle: VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 5 (leicht modifiziert). . | 16 |
| Abb. 11: Darstellung einer Funktionsstruktur mittels White-Box-Modell, Quelle: Pahl u. a. (2007), S. 45 (modifiziert). | 18 |
| Abb. 12: Darstellung einer Gesamtfunktion als Black-Box-Modell, Quelle: Eigner/Roubanov/Zafirov (2014), S. 17 (leicht modifiziert) in Anlehnung an Pahl/Beitz. | 19 |
| Abb. 13: Prinzip der Abstraktion durch Aggregation, Quelle: Eigene Darstellung in Anlehnung an Winter (1991), S. 20. | 23 |
| Abb. 14: Wirkrichtung des Abstraktionsprinzips der Aggregation und Partizipation, Quelle: Eigene Darstellung. | 23 |
| Abb. 15: Wirkrichtung des Abstraktionsprinzips der Generalisierung/Spezialisierung, Quelle: eigene Darstellung. | 23 |
| Abb. 16: Hierarchischer Strukturaufbau von Generalisierung und Spezialisierung, Quelle: Kahlbrandt (2001), S. 51 (leicht modifiziert). | 24 |
| Abb. 17: Wirkrichtung des Abstraktionsprinzips der Klassifizierung/Instanziierung, Quelle: eigene Darstellung. | 24 |
| Abb. 18: Unterschied zwischen Generalisierung und Klassifizierung, Quelle: Dori (2016), S. 288 (modifiziert). | 25 |

| | |
|--|----|
| Abb. 19: Wirkrichtung des Abstraktionsprinzips der Komposition/Dekomposition, Quelle: eigene Darstellung. | 25 |
| Abb. 20: Beschreibung des Abstraktionsprinzips der Komposition/Dekomposition mittels Kompositionsdiagramm anhand eines Brettspiels, Quelle: Seemann/von Gutenberg (2006), S. 129 (modifiziert). | 26 |
| Abb. 21: Teildisziplinen der Digitalen Fabrik, Quelle: VDI 4499 Blatt 1 (Februar 2008), S. 5 (modifiziert). | 30 |
| Abb. 22: Positionierung der VIBN in Bezug auf die Planungsschritte von Fertigungsanlagen, Quelle: VDI 4499 Blatt 2 (Mai 2011), S. 10 (leicht modifiziert). | 31 |
| Abb. 23: Grad der Kostenbeeinflussung und Kostenbeurteilung in Abhängigkeit des Projektfortschritts, Quelle: VDI 2235 (Oktober 1987), S. 5 (leicht modifiziert). | 34 |
| Abb. 24: Potential und Aufwand der Virtuellen Inbetriebnahme, Quelle: Zäh/Reinhart (Hrsg.) (2006), S. 2 (Kapitel 1 / leicht modifiziert). | 35 |
| Abb. 25: Funktionales Hierarchiemodell von Steuerungen, Quelle: DIN EN 62264-1 (Juni 2008), S. 14 (modifiziert). | 36 |
| Abb. 26: Hierarchischer Aufbau informationstechnischer Verarbeitung in flexiblen Fertigungssystemen, Quelle: Milberg (Hrsg.) (1992), S. 128 (modifiziert). | 37 |
| Abb. 27: Skalierbarkeit der Steuerungsebenen im Zuge der Erstellung von Simulationsmodellen, Quelle: Wünsch (2007), S. 113 (modifiziert). | 38 |
| Abb. 28: Hierarchieebenen eines Automatisierungssystems sowie Zuweisung der passenden Steuerungstechnologie, Quelle: Lacour (2011), S. 12 (modifiziert). | 38 |
| Abb. 29: Zeitliches Verhalten der unterschiedlichen Echtzeitbedingungen, Quelle: Halang/Konakovsky (2018), S. 21 (modifiziert). | 40 |
| Abb. 30: Auswirkung unterschiedlicher Abtastfrequenzen hinsichtlich ihres Informationserhalts, Quelle: Wünsch (2007), S. 26 (modifiziert). | 43 |
| Abb. 31: Generischer Aufbau eines Automatisierungssystems auf Zellenebene, Quelle: VDI/VDE 3693 Blatt 1 (August 2016), S. 7. | 44 |
| Abb. 32: Klassische Systemkonfiguration eines VIBN-Ansatzes mittels MiL, Quelle: VDI/VDE 3693 Blatt 1 (August 2016), S. 8 (leicht modifiziert). | 45 |
| Abb. 33: Gegenüberstellung der unterschiedlichen Definitionen von Software-in-the-Loop in Anlehnung an VDI 4499 und VDI 3693, Quelle: Eigene Darstellung. | 46 |
| Abb. 34: Methode der VIBN mittels Hardware-in-the-Loop, Quelle: VDI/VDE 3693 Blatt 1 (August 2016), S. 10 (leicht modifiziert). | 47 |
| Abb. 35: Übersicht zur Anwendung kommender Modelle im Zuge einer VIBN, Quelle: VDI/VDE 3693 Blatt 1 (August 2016), S. 16. | 48 |

| | |
|--|----|
| Abb. 36: Zusammenhang unterschiedlicher VIBN-Methoden in Bezug auf Modellbildung und Codegenerierung, Quelle: VDI 4499 Blatt 2 (Mai 2011), S. 39 (leicht modifiziert)..... | 50 |
| Abb. 37: Typischer Aufbau einer Karosseriebauzelle bei MSF, Quelle: Eigene Darstellung..... | 53 |
| Abb. 38: Steuerungsarchitektur eines SPS-Systems bei MSF, Quelle: Magna Steyr. | 54 |
| Abb. 39: Feldbustopologie des SPS-Systems TH03RHB4, Quelle: Eigene Darstellung..... | 55 |
| Abb. 40: Steuerungskonzept mittels Ventilinsel anhand der Vorrichtung HB4110WZ01, Quelle: Eigene Darstellung. | 56 |
| Abb. 41: Architektur des Softwaresystems Process Designer/Process Simulate, Quelle: Eigene Darstellung. | 57 |
| Abb. 42: Gates und Meilensteine des Magna Steyr Development System, Quelle: Magna Steyr (leicht modifiziert). | 58 |
| Abb. 43: Entstehungsprozess von Produktionsanlagen basierend auf MSDS, Quelle: Magna Steyr (modifiziert). | 59 |
| Abb. 44: Phasen der Virtuellen Absicherung von Karosseriebauanlagen bei MSF, Quelle: Eigene Darstellung. | 59 |
| Abb. 45: Positionierung der VIBN im Rahmen der virtuellen Absicherung von Produktionsanlagen, Quelle: Eigene Darstellung. | 61 |
| Abb. 46: Zeitliche Positionierung des Meilensteins der Virtuellen Inbetriebnahme im Anlagenentstehungsprozess, Quelle: Magna Steyr. | 62 |
| Abb. 47: HiL-Testkit zur Durchführung einer VIBN bei MSF (Seitenansicht), Quelle: Eigene Darstellung. | 63 |
| Abb. 48: HiL-Testkit zur Durchführung einer VIBN bei MSF (Frontansicht), Quelle: Eigene Darstellung. | 64 |
| Abb. 49: Systemarchitektur und Systemverhalten von SIMIT Unit/Simba PNIO, Quelle: Siemens AG (Hrsg.) (07/2017), S. 57..... | 65 |
| Abb. 50: Netzwerktopologie A des VIBN-Systems, Quelle: Siemens AG (Hrsg.) (07/2017), S. 70..... | 67 |
| Abb. 51: Netzwerktopologie B des VIBN-Systems, Quelle: Eigene Darstellung..... | 68 |
| Abb. 52: Netzwerktopologie C eines VIBN-Systems, Quelle: Siemens AG (Hrsg.) (07/2017), S. 72..... | 69 |
| Abb. 53: Export der Hardwarekonfiguration aus SPS-System, Quelle: Eigene Darstellung..... | 71 |
| Abb. 54: IP-Adressierung des VIBN-Netzwerkes während Projektierung der Simit Unit, Quelle: Eigene Darstellung. | 72 |
| Abb. 55: Auszug des Importergebnisses der Profinet-Feldgeräte des SPS-Systems TH3RHB04, Quelle: Eigene Darstellung. | 73 |
| Abb. 56: Erforderliche Einstellwerte zur Einbindung von PS in ein HiL-System, Quelle: Eigene Darstellung. | 75 |

| | |
|---|----|
| Abb. 57: Konfiguration der Kommunikationseinstellung für den HiL-Betrieb durch Einbindung der Simulation Unit, Quelle: Eigene Darstellung. | 75 |
| Abb. 58: Einstellwerte zum Kommunikationsaufbau zwischen PS 14.1 und Simulation Unit, Quelle: Eigene Darstellung. | 75 |
| Abb. 59: Bekanntgabe des Ablageorts der Simit Unit Projektierung für Process Simulate, Quelle: Eigene Darstellung. | 76 |
| Abb. 60: Validierung fehlerfreier Kommunikation zwischen Process Simulate und Simulation Unit, Quelle: Eigene Darstellung. | 76 |
| Abb. 61: Teamzusammensetzung im Rahmen des Aufbaus des VIBN-Prototypen, Quelle: Eigene Darstellung. | 77 |
| Abb. 62: Kommunikationsverlauf zwischen den Hauptkomponenten des VIBN-Systems, Quelle: Eigene Darstellung. | 79 |
| Abb. 63: Interpretation der Datenrichtung von Eingangs- und Ausgangssignalen im virtuellen Anlagenmodell, Quelle: Eigene Darstellung. | 80 |
| Abb. 64: Export der Symboltabelle respektive Signale des SPS-Systems zur weiteren Aufbereitung, Quelle: Eigene Darstellung. | 81 |
| Abb. 65: Überführte Symboltabelle respektive Signale des SPS-Systems zur weiteren Aufbereitung für Process Simulate, Quelle: Eigene Darstellung. | 82 |
| Abb. 66: Festlegung der Systemgrenze des Roboters hinsichtlich anzuwendender Signale im Zuge der VIBN, Quelle: Eigene Darstellung. | 83 |
| Abb. 67: Signalkorrespondenz zwischen Roboterinstanz, virtueller Produktionsanlage, Simit Unit PN256 und realer SPS, Quelle: Eigene Darstellung. | 85 |
| Abb. 68: Form und Inhalt der importfähigen Signalschnittstelle für die Instanz des virtuellen Roboters HB4110IR01, Quelle: Eigene Darstellung. | 87 |
| Abb. 69: Signalmapping von Robotersignalen, die roboterintern zu Variablen zusammengefasst sind, Quelle: Eigene Darstellung. | 87 |
| Abb. 70: Erfolgreicher Import der Signalschnittstelle des Roboters HB4110IR01, Quelle: Eigene Darstellung. | 88 |
| Abb. 71: Abbildung von Roboterprogrammen mit Positionspunkten im kartesischen Raum im VIBN-Modell, Quelle: Eigene Darstellung. | 89 |
| Abb. 72: Einbindung von Roboterprogrammen in das VIBN-Modell, die ausschließlich aus Programmlogik bestehen, Quelle: Eigene Darstellung. | 89 |
| Abb. 73: Mechanisch- / kinematisches Modell der Spannvorrichtung HB4110WZ02, Quelle: Eigene Darstellung. | 91 |

| | |
|---|-----|
| Abb. 74: Pneumatisches und elektrisches Modell der Ventilgruppe 22QM der Vorrichtung HB4110WZ02, Quelle: Eigene Darstellung | 92 |
| Abb. 75: Logikblock des Stellgliedes 0012255__a__1__a__bg01, Quelle: Eigene Darstellung. | 93 |
| Abb. 76: Temporärer Logikblock mit der Bezeichnung LB zum Zwecke des Signalimports, Quelle: Eigene Darstellung. | 94 |
| Abb. 77: Auszug der Aufbereitung der Eingangs- und Ausgangssignale der Vorrichtung HB4110WZ02, Quelle: Eigene Darstellung | 94 |
| Abb. 78: Signaldeklaration zur Einbindung in externen Kommunikationskanal, Quelle: Eigene Darstellung. | 96 |
| Abb. 79: Signalkorrespondenz zwischen virtuellem Sensor und realer HMI, Quelle: Eigene Darstellung. | 97 |
| Abb. 80: Planungsvorgabe des Materialflusses des Betriebsartenbereiches HB4-BA1, Quelle: Eigene Darstellung. | 98 |
| Abb. 81: Prozessbezogene Abbildung des Materialflusses des Betriebsartenbereiches HB4-BA1, Quelle: Eigene Darstellung. | 98 |
| Abb. 82: Systemaufbau zur Durchführung der operativen Tätigkeiten im Rahmen der VIBN, Quelle: Eigene Darstellung. | 101 |
| Abb. 83: Startroutine des virtuellen Roboters zur programmtechnischen Einbindung in Hand- und Automatikbetrieb, Quelle: Eigene Darstellung..... | 102 |
| Abb. 84: Notwendige Programmänderung zur Erreichung der fehlerfreien Interaktion zwischen Real-SPS und virtuellem Roboter, Quelle: Eigene Darstellung. | 104 |
| Abb. 85: Programmtechnische Lösung zur Unterscheidung zwischen HiL u. MiL am virtuellen Roboter, Quelle: Eigene Darstellung..... | 105 |

TABELLENVERZEICHNIS

| | |
|---|----|
| Tab. 1: Modellklassifizierung nach Einsatzzweck, Informationsart und Abstraktionsgrad, Quelle: Arnold u. a. (2008), S. 36 (modifiziert). | 21 |
| Tab. 2: Modellierungssprachen kategorisiert nach Eignung zur Abbildung von Systemeigenschaften, Quelle: VDI 4465 Blatt 1 (Entwurf) (Mai 2016), S. 22 (modifiziert). | 27 |
| Tab. 3: Motive zur Durchführung einer Virtuellen Inbetriebnahme, Quelle: VDI 4499 Blatt 2 (Mai 2011), S. 6 - 10 (modifiziert)..... | 33 |
| Tab. 4: Gegenüberstellung der Auswirkungen bei Nichteinhaltung von unterschiedlichem Echtzeitverhalten, Quelle: Halang/Konakovsky (2018), S. 21 (modifiziert)..... | 41 |
| Tab. 5: Methoden der VIBN einer automatisierter Produktionsanlagen, Quelle: VDI 4499 Blatt 2 (Mai 2011), S. 38 (modifiziert). | 43 |
| Tab. 6: Übersicht der Systemeigenschaften von Simba PNIO premium und Simit Unit PN256, Quelle: Siemens AG (Hrsg.) (07/2017), S. 58 (leicht modifiziert). | 66 |
| Tab. 7: Vergleich der Signalanzahl mit und ohne Festlegung der Systemgrenze an der Roboterschnittstelle, Quelle: Eigene Darstellung. | 83 |
| Tab. 8: Wertevorrat zur Veranschaulichung der Signalkorrespondenz zwischen virtuellen Robotern und realer SPS, Quelle: Eigene Darstellung. | 84 |
| Tab. 9: Erläuterung der nativen Robotersyntax zur Beschreibung von Signalen, Quelle: Eigene Darstellung. | 86 |
| Tab. 10: Vergleich der Bezeichnungsmethodik von Anlagenkomponenten zwischen mechanische- und elektrischer Konstruktion, Quelle: Eigene Darstellung. | 95 |
| Tab. 11: Signalmapping von Sensorsignalen zur Substituierung im virtuellen Anlagenmodell, Quelle: Eigene Darstellung. | 97 |

ABKÜRZUNGSVERZEICHNIS

| | |
|---------|--|
| AS | Automatisierungssystem |
| AT | Automatisierungstechnik |
| BA | Betriebsartenbereich |
| CC | Concept Confirmation |
| CEE | Cyclic Event Evaluation |
| CNC | Computerized Numerical Control |
| CPU | Central Processing Unit |
| DB | Datenbank |
| DLL | Dynamic Link Library / Dynamische Verbindungs-Bibliothek |
| EOP | End of Production |
| ERP | Enterprise Resource Planning |
| FAT | Factory Acceptance Tests |
| FC | Functional Confirmation |
| GSDML | General Station Description Markup Language |
| HiL | Hardware in the Loop |
| HMI | Human Machine Interface |
| IEC | International Electrotechnical Commission |
| IRT | Industrial Real-Time |
| JT | Jupiter Tessellation |
| LB | Logikblock |
| LS | Launch Sign Off |
| LZI/LTI | Lineares zeitinvariantes System |
| MES | Manufacturing Execution System |
| MFG | Manufacturing Feature |
| MiL | Model in the Loop |
| MKS | Mehrkörpersystem |
| MRP | Media Redundancy Protocol |
| MSDS | Magna Steyr Development System |
| MSF | Magna Steyr Fahrzeugtechnik AG & Co KG |
| NC | Numerical Control |
| OLP | Offlineprogrammierung |

| | |
|--------|---|
| PD | Process Designer |
| PEP | Produktentstehungsprozess |
| PLC | Programmable Logic Controller |
| PN | Profinet |
| PPS | Preliminary Product Specification |
| PS | Process Stability, Process Simulate |
| PTO | Production Try Out |
| PV | Product Vision |
| RCS | Realistic Controller Simulation |
| RT | Real Time |
| SiL | Software in the Loop |
| SOP | Start of Production, Sequence of Operation |
| SPLM | Siemens Product Lifecycle Management Software |
| SPS | Speicherprogrammierbare Steuerung |
| SysML | Systems Modeling Language |
| TA | Target Agreement |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| UDP | User Datagram Protocol |
| UML | Unified Modeling Language |
| VC | Virtual Commissioning |
| VDA | Verband der Automobilindustrie |
| VDE | Verband deutscher Elektrotechnik, Elektronik und Informationstechnik. |
| VDI | Verein Deutscher Ingenieure |
| VIBN | Virtuelle Inbetriebnahme |
| VIBS | Virtuelle Inbetriebsetzung |