

Virtuelle Inbetriebnahme mechatronischer Systeme unter Einbeziehung realer Industriesteuerungen von Bosch Rexroth

Virtual commissioning of mechatronic systems using motion controllers from Bosch Rexroth

Andreas Hofmann, Bosch Rexroth, 97816 Lohr am Main, Deutschland, andreas.hofmann7@boschrexroth.de

Stephan Schweig, Uni Duisburg-Essen, Lehrstuhl für Mechatronik, 47057 Duisburg, Deutschland, schweig@mechatronik.uni-duisburg.de

Lars Mikelsons, Bosch Rexroth, 97816 Lohr am Main, Deutschland, lars.mikelsons@boschrexroth.de

Kurzfassung

Die Projektdauer eines technischen Systems wird signifikant von der Inbetriebnahme beeinflusst, wobei hier das Beseitigen von Fehlern in der Steuerungsapplikationen den dominierenden Aufwand darstellt. Durch die Entwicklung immer komplexerer Anlagen, die horizontal und vertikal vernetzt sind, wird der Anteil an Software innerhalb eines technischen Systems deutlich steigen und somit erheblichen Einfluss auf die Inbetriebnahmezeiten ausüben. Um die Inbetriebnahmezeiten einzuschränken, stellt die virtuelle Inbetriebnahme ein geeignetes Mittel dar. Darüber hinaus kann mit dieser auch die Softwarequalität von Steuerungsapplikationen verbessert werden. Gängige Methoden der virtuellen Inbetriebnahme können der steigenden Komplexität eines technischen Systems sowie dem erhöhten Bedarf an Software aber nur eingeschränkt Herr werden. In der vorliegenden Arbeit wird daher ein Verfahren vorgestellt, mit welchem beliebig komplexe technische Systeme unter der Verwendung von Bosch Rexroth Industriesteuerungen virtuell in Betrieb genommen werden können, was abschließend an der virtuellen Inbetriebnahme eines Delta-Roboters demonstriert wird.

Abstract

Up to 25 % of the project period of a plant are needed for commissioning. Debugging of error-prone controller applications dominate this operation. The rise of connected, highly automated and cyber-physical systems will increase the number of software parts within a plant and thus have a significant impact on commissioning times. In order to deal with this impact, the technique of virtual commissioning is practical. Using virtual commissioning can not only reduce the commissioning time but also increase quality of software parts within a technical system. However, common methods of virtual commissioning are limited in complexity of a technical system and in handling the rising demand on software. Industrial controllers from Bosch Rexroth allow performing virtual commissioning for any complexity of technical system by using an enhanced hardware-in-the-loop approach, which is presented in this paper. The simplicity of this method, which requires no expensive infrastructure, is shown by virtual commissioning of a delta-robot system.

1 Einleitung

1.1 Motivation

Die Inbetriebnahme (IBN) technischer Systeme, welche im Folgenden nach [1] als „funktionsgerechtes Einschalten des Systems in Verbindung mit dem zugehörigen Prozess“ verstanden werden soll, stellt einen wesentlichen Bestandteil der Entwicklung eines technischen Produktes dar. Hierbei beansprucht die IBN einer Anlage bis zu einem Viertel der kompletten Projektdauer, vgl. **Bild 1**. Innerhalb des technischen Systems wird diese Zeit vor allem für die Einrichtung der Steuerungstechnik und die Beseitigung von Softwarefehlern benötigt.

Sind heute schon ein Großteil der Probleme der IBN einer technischen Anlage softwareseitig verursacht, so ist anzunehmen, dass die Inbetriebnahmezeiten durch die vierte industrielle Revolution (I 4.0) auf Basis cyber-physikalischer Systeme steigen wird.

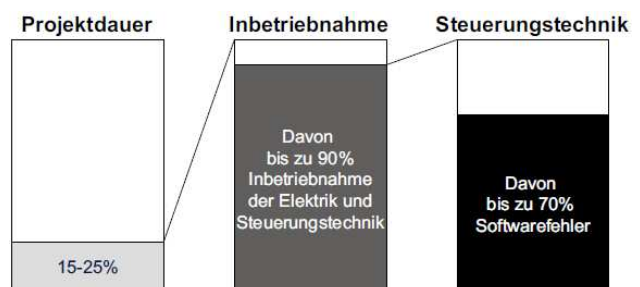


Bild 1 Anteil der Inbetriebnahmezeit an der Projektdauer, [2]

Dies wird zum einen durch den deutlich ansteigenden Anteil an Elektronik und Software im Maschinenbau und die horizontale wie vertikale Vernetzung, vgl. [3] und **Bild 2**, verursacht, aber auch durch die steigende Komplexität der Softwareprojekte selbst.

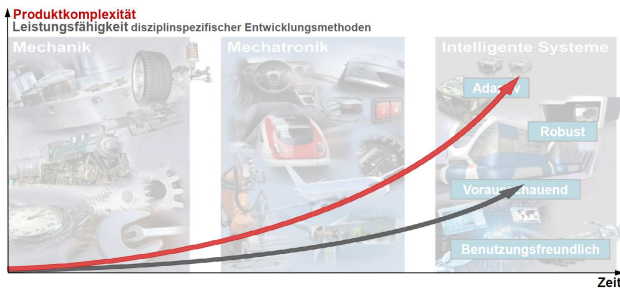


Bild 2 Anstieg der Produktkomplexität und Leistungsfähigkeit disziplinspezifischer Entwicklungsmethoden, [4]

Dementsprechend müssen auch Methoden, die schon jetzt für eine deutliche Reduzierung von Inbetriebnahmezeiten eingesetzt werden, für diese neuen Herausforderungen weiterentwickelt werden. Besonders die virtuelle Inbetriebnahme bietet hier gute Einsparpotentiale bei der Inbetriebnahmezeit, aber darüber hinaus auch beim Finden von Fehlern und der Steigerung der Softwarequalität [5], vgl. **Bild 3**.

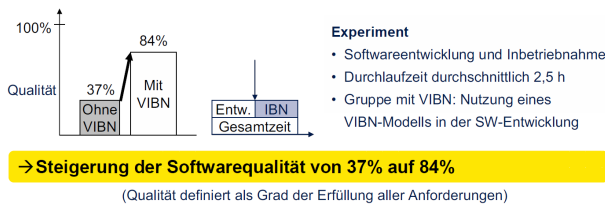


Bild 3 Steigerung der Softwarequalität durch virtuelle Inbetriebnahme, [5]

Allerdings müssen auch die Verfahren der virtuellen Inbetriebnahme, wie beispielsweise die Hardware-In-The-Loop-Simulation, an die neuen Herausforderungen angepasst werden.

1.2 Aufbau der Arbeit

In Kapitel 2 wird der Begriff virtuelle Inbetriebnahme näher beleuchtet, Grundprinzip und Anforderungen untersucht und auf geläufige Verfahren eingegangen. Im nachfolgenden Kapitel wird die virtuelle Inbetriebnahme konkret für Rexroth Industriesteuerungen und ihre Besonderheiten, welche eine Erweiterung der Hardware-In-The-Loop-Simulation darstellt, präsentiert. Im vierten Kapitel wird die virtuelle Inbetriebnahme mit Rexroth Steuerungen am konkreten Beispiel eines Delta-Roboters dargestellt, bevor im letzten Kapitel der Inhalt des Beitrages noch einmal zusammengefasst und ein Ausblick über mögliche Erweiterungen gegeben wird.

2 Virtuelle Inbetriebnahme

2.1 Definition, Grundprinzip und Anforderungen

Unter virtueller Inbetriebnahme (VIBN) wird im Folgenden die Überprüfung der Funktionsfähigkeit bzw. des Funktionsablaufes einer Anlage mit Hilfe eines virtuellen Simulationsmodells der Anlage bezeichnet. Dabei wird davon ausgegangen, dass die Steuerung analog zur Anlage als

virtuelles Modell vorliegt oder als reale Steuerungshardware mit darauf laufender Steuerungsapplikation zur Verfügung steht.

Grundprinzip der VIBN ist die Kopplung einer virtuellen Anlage mit einer virtuellen oder realen Steuerung. Durch die Simulation kann bereits in frühen Stadien der Entwicklung die Funktionsfähigkeit von Steuerungsapplikationen überprüft werden. Wenn die virtuelle Inbetriebnahme parallel zur Fertigung der Anlage erfolgt, können nach [5] speziell Fehler der Steuerungsapplikation gefunden und behoben werden. Dadurch können bei der IBN des technischen Systems erhebliche Inbetriebnahmezeiten und dadurch Kosten eingespart werden, vgl. **Bild 4**.

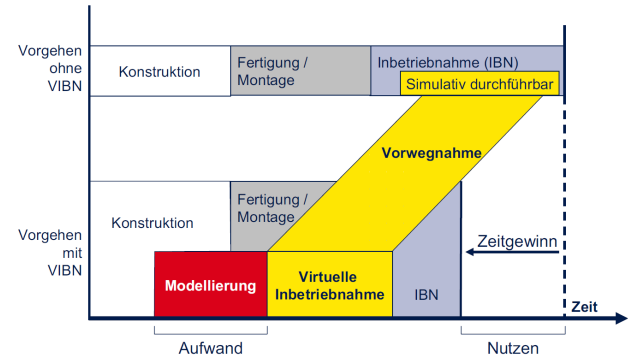


Bild 4 Grundidee der virtuellen Inbetriebnahme nach Wunsch und Zäh, [5]

Darüber hinaus unterstützt die VIBN zusätzlich die Qualität der Steuerungsprogramme zu verbessern. Wird die Steuerungsapplikation erst während der tatsächlichen IBN entwickelt und getestet, werden üblicherweise nicht ausgereifte Steuerungsapplikationen auf der Steuerung implementiert, was zu unerwartetem Verhalten führen kann.

Als zentrales Ziel der VIBN soll daher im Folgenden das frühzeitige Auffinden von Fehlern in der Steuerungsapplikation eines technischen Systems definiert werden, um Kosten zu sparen und die Qualität des Anlagenverhaltens zu verbessern.

Durch die Integration von Simulation in den Prozess der (virtuellen) Inbetriebnahme ergeben sich zusätzlich noch weitere Möglichkeiten. Dies beinhaltet zum einen die Durchführung gefahrloser Tests unter einstellbaren äußeren Einflüssen, die mit geringen Kosten verbunden sind. Zum anderen können problemlos Fehler injiziert werden und so die Reaktion der Anlage und der Steuerung auf Fehler untersucht werden, lange bevor die echte Anlage überhaupt aufgebaut ist.

Allerdings ist für die VIBN zusätzlicher Aufwand in Form der Modellbildung der Anlage erforderlich. Dies stellt vor allem für viele kleine und mittelständische Unternehmen einen unbekanntem Schritt dar, der ein Umdenken in der Entwicklung erfordert. Im Zuge des Zukunftsprojektes Industrie 4.0 des Bundesministeriums für Bildung und Forschung ist dieses Umdenken zur modellbasierten Entwicklung aber unabhängig von der VIBN nötig, um die Wettbewerbsfähigkeit deutscher Unternehmen zu erhalten, vgl. [3]. Nichtsdestotrotz ergibt sich daraus eine Anforderung

an die VIBN technischer Systeme, sich möglichst nahtlos in die durchgängige modellbasierte Entwicklung zu integrieren, damit keine neuen Modelle erstellt, sondern virtuelle Abbilder der Anlage aus früheren Produktentstehungsphasen verwendet werden können.

2.2 Kopplungsstrategien der VIBN

Die Kopplung zwischen Steuerung und virtueller Anlage stellt einen zentralen Punkt der VIBN dar und hat wesentlichen Einfluss auf die Aussagekraft der Ergebnisse. Die beiden Kopplungsstrategien Hardware-In-The-Loop (HiL) und Software-In-The-Loop (SiL) sind dabei die wichtigsten Verfahren und sollen näher beschrieben werden. Da die Begriffe HiL und SiL nicht zwangsweise eindeutig definiert sind, wird an dieser Stelle zunächst eine kurze Definition gegeben. Des Weiteren werden die Verfahren für die eingangs gestellten Anforderungen, das Finden von Fehlern in der Steuerungsapplikation und die Integration in eine durchgängige modellbasierte Entwicklung, bewertet.

Software-In-The-Loop-Kopplung

Unter einer SiL-Kopplung wird im Folgenden die Strategie der Verknüpfung des virtuellen Modells der Anlage mit einem virtuellen Abbild der Steuerung verstanden.

Dadurch, dass die Steuerung nur in Form eines Modells vorliegt, kann das komplette System auf dem gleichen PC und häufig in der gleichen Software simuliert werden. Da keine reale Steuerungshardware verwendet wird, sind folglich auch keine Echtzeitanforderungen zu erfüllen. Daraus ergibt sich, dass beliebig komplexe virtuelle Modelle der Anlage für die VIBN verwendet werden können und somit eine hohe Durchgängigkeit durch die Verwendung von Modellen aus früheren Entwicklungsphasen ermöglicht ist. Da bei der SiL-Kopplung aber keine realen Steuerungsapplikationen verwendet werden, kann oftmals nicht das reale Verhalten der Anlage abgebildet werden und speziell das Finden von Fehlern ist nicht möglich.

Hardware-In-The-Loop-Kopplung

Die HiL-Kopplung wird im Folgenden als Kopplungsverfahren definiert, bei der eine reale Steuerungshardware, auf welcher die echte Steuerungsapplikation der Anlage läuft, mit dem Simulationsmodell der Anlage gekoppelt wird.

Aufgrund dessen, dass die Steuerung in harter Echtzeit¹ arbeitet, muss die Kopplung in der Regel mittels echtzeitfähigem Bus (z.B. Sercos, Profibus, etc.) und Computer mit einem in Echtzeit laufenden Betriebssystem erfolgen. Daraus ergeben sich ein hoher Aufwand für die Infrastruktur und zusätzlich erhebliche Einschränkungen an das Simulationsmodell der Anlage aufgrund der Echtzeitanforderungen der Steuerungshardware.

Vor dem Hintergrund der durchgängigen modellbasierten Entwicklung ist der Ansatz der HiL-Kopplung im Allgemeinen wenig geeignet, da hierfür in der Regel spezielle Anlagenmodelle erstellt werden müssen, um die Echtzeitanforderungen zu erfüllen. Das Verwenden von Modellen aus vorherigen Entwicklungsphasen ist daher kaum

möglich. Für das Finden von Fehlern in der Steuerungsapplikation ist das Verfahren der HiL-Kopplung dagegen gut einsetzbar, da die echte Steuerungsapplikation auf der echten Steuerungshardware direkt in die VIBN einbezogen wird.

3 VIBN mit Rexroth Industriesteuerungen

3.1 Modelica

Die Modellbildung und Simulation technischer Systeme, die im vorherigen Kapitel immer nur als Blackbox behandelt wurde, stellt einen integralen Bestandteil der VIBN dar. Speziell unter dem Gesichtspunkt hoch automatisierter, selbst einstellender und intelligent vernetzter Produktionsanlagen steigt die Nachfrage nach cyber-physikalischen Industrieanlagen und daher der Bedarf nach detaillierten Simulationsmodellen, die über einzelne Domänen hinaus gehen, deutlich. Bei Bosch Rexroth wird für die Entwicklung dieser detaillierten, domänenübergreifenden Systemmodelle Modelica verwendet, vgl. [6], [7]. Modelica [8], als universelle Modellierungssprache, ist der fortschrittlichste Ansatz um derartige domänenübergreifende Modelle zu entwickeln. Im Gegensatz zu üblicher Systemengineering-Software ermöglicht Modelica dem Anwender, Systeme mittels akausaler Gleichungen zu beschreiben, ohne sich Gedanken über die mathematischen Details machen zu müssen. Die Transformation in ein lösbares mathematisches Modell wird automatisch durch einen Modelica Compiler realisiert. Die verschiedenen kommerziellen und freien Modelica-basierten Modellierungs- und Simulationsprogramme sind bereits heute Stand der Technik bei Entwurf und Optimierung von Systemen und Arbeitsabläufen. Da Modelica ein nicht-proprietärer Sprachstandard ist, arbeitet eine große Community bestehend aus akademischen und industriellen Vertretern an der Weiterentwicklung der Kapazitäten von Modelica, z.B. im Bereich von modellprädiktiver Regelung oder cloud-basierter Simulation. Auch der weit verbreitete Modellaustauschstandard FMI wird von der Modelica-Community gepflegt und weiterentwickelt.

3.2 Das OpenCore-Interface - Software-schnittstelle zur Rexroth Industriesteuerung

Das Rexroth OpenCore-Interface [9] ist eine universelle Schnittstelle, mit der ein direkter Funktionszugriff auf den Steuerungs- und Antriebskern von Rexroth Industriesteuerungen ermöglicht wird. Mit diesem können mit modernen Hochsprachen wie C++ oder Java Softwareprogramme geschrieben werden, die es ermöglichen, Antriebs- und Steuerungssysteme mit klassischen IT-Systemen zu kombinieren, vgl. [10]. Für die VIBN von Rexroth Industriesteuergeräten wird das OpenCore-Interface in der Modelica-Bibliothek `mpli4Modelica` implementiert. Diese Bibliothek, die Kernfunktionen des Rexroth OpenCore Software

¹d.h. mit festem Zeittakt, welcher nicht überschritten werden darf

Development Kits aufruft, ermöglicht aus einer Modelica-basierten Simulationsumgebung direkt auf die Steuerung zuzugreifen.

3.3 Erweiterung der HiL-Kopplung mit dem OpenCore-Interface

Die VIBN mit Rexroth Industriesteuerungen basiert auf dem Ansatz der HiL-Kopplung, erweitert diesen aber erheblich, um die beiden zentralen Anforderungen der durchgängigen modellbasierten Entwicklung und dem Auffinden von Fehlern in der Steuerungssoftware zu ermitteln. Erreicht wird dies dadurch, dass die limitierenden Echtzeitanforderungen mit Hilfe des OpenCore-Interfaces aufgehoben werden, indem die Steuerung in einen Simulationsmodus geschaltet wird. In diesem wird der Motionkernel der Steuerung, der für den Aufruf der einzelnen Motion- und Sercostasks zuständig ist, in einen Zustand versetzt, in welchem die Tasks nicht mehr synchron zum internen Takt der Steuerung, sondern durch externe Vorgabe aufgerufen werden, vgl. **Bild 5**.

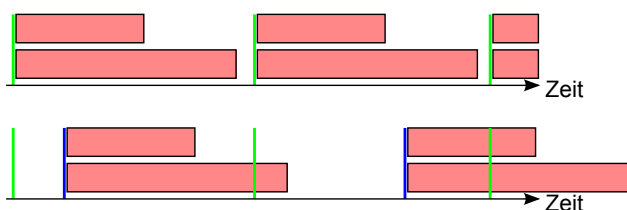


Bild 5 Modi der Steuerung
oben: zyklisch Abarbeitung zweier Motiontasks (rot) mit dem internen Takt (grün)
unten: Abarbeitung der Motiontasks durch externe Benutzervorgabe (blau)

Dennoch wird innerhalb der Firmware der Steuerung sichergestellt, dass die Steuerung konsistent und deterministisch arbeitet. Das manuelle Ansprechen der Motion- und Sercostasks ermöglicht es, beliebig komplexe virtuelle Modelle der Anlage, deren Simulation in der Regel langsamer als der Echtzeittakt der Steuerung abläuft, mit der Steuerung zu kombinieren. Somit ist über diese Technik neben der guten Fehleranalyse der Steuerungsapplikation auch eine vollständig durchgängige modellbasierte Entwicklung gewährleistet. Aufgrund des Entfallens der Echtzeitanforderungen wird die komplizierte Infrastruktur aus echtzeitfähigem Bus und Realtime-PC, welche üblicherweise für die HiL-Kopplung erforderlich ist, durch einen Standardcomputer und eine Ethernet-Verbindung ersetzt.

Die Synchronisierung der Daten und das Anstoßen der Tasks der Steuerungsapplikation während der Simulation wird von der Modelica-Bibliothek `mlpi4Modelica` übernommen, welche die Funktionalität des OpenCore-Interfaces für Modelica-basierte Modellierungs- und Simulationstools zur Verfügung stellt und Teil des Rexroth OpenCore Software Development Kits ist.

Zu Beginn des definierten Austauschzeitpunktes² werden von der Simulation alle relevanten Daten an die Steuerung übertragen. Anschließend sendet die Simulation dem Mo-

tionkernel der Steuerung das Signal, die Motion- und Sercostasks abzuarbeiten. Im letzten Schritt der Kommunikation werden alle relevanten Daten von der Steuerung abgefragt und zur Simulation transferiert. Die Zeitschritte zwischen den Austauschzeitpunkten rechnet nur die Simulation, wobei die Daten der Steuerung konstant gehalten werden, was auch dem realen Verhalten während des Betriebs an der echten Anlage entspricht.

3.4 Visualisierung technischer Systeme für die VIBN

Die Auswertung von Simulationsläufen einer Anlage stellt eine anspruchsvolle Aufgabe dar. Gerade im Bereich Robotik sind Graphen mit dem Bewegungsverhalten der Anlage oft nur schwer interpretierbar. Daher wurde bei Bosch Rexroth parallel zur Entwicklung der VIBN-Schnittstelle für Industriesteuerungen damit begonnen, eine Visualisierung zu entwickeln, die es ermöglicht, das Bewegungsverhalten einer Anlage nachzuvollziehen. Obwohl die meisten Modelica-basierten Modellierungs- und Simulationstools eine Visualisierung der Mechanik aus dem Modell erzeugen können, sind diese in ihrem Umfang eingeschränkt. Diese eigenständige Visualisierung ist in der Lage, ebenfalls unter Verwendung des OpenCore-Interfaces, anlagenrelevante Daten aus der Steuerung zu extrahieren. So kann beispielsweise die komplette Geometrie von Robotern zur Laufzeit direkt aus der Steuerung abgefragt werden und zusätzlich der Arbeitsraum berechnet und dargestellt sowie der Arbeitsbereich und Hindernisse visualisiert werden. Dabei können speziell letzte genutzt werden, um das Eindringen der Anlage in den Bereich eines anderen Systems zu detektieren. In **Bild 6** ist die Visualisierung für einen Delta-Roboter aus Steuerungsdaten gezeigt.

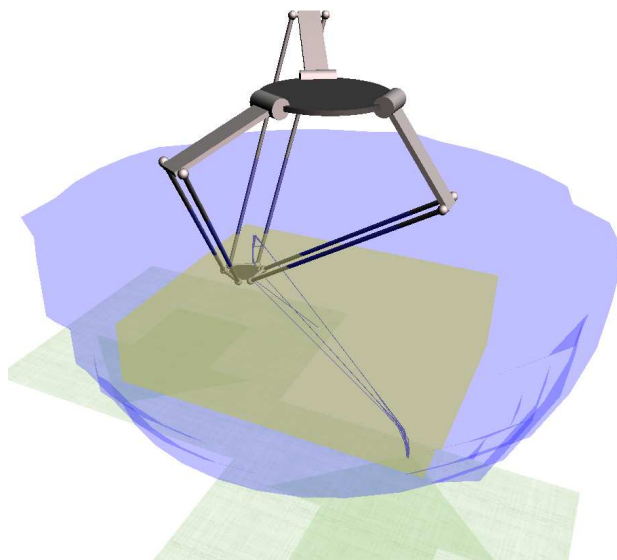


Bild 6 Visualisierung eines Deltaroboters (grau) mit Arbeitsraum (hellblau), Arbeitsbereich (gelb), Trajektorie (dunkelblau) und Bändern (grün)

Die Visualisierung erfolgt mit der OpenSource-Bibliothek

²üblicherweise der Motion- bzw. Sercostakt des Steuerungsapplikation auf der Industriesteuerung

Object Oriented Graphics Rendering Engine. Technische Details können unter [12] eingesehen werden.

Ein Teilprogramm der Visualisierung kann im Live-Betrieb zusätzlich verwendet werden, um Daten an realen Anlagen für die Analyse aufzuzeichnen. Dadurch können die vorgestellten Methoden nicht nur zur Inbetriebnahme, sondern auch zur modellgestützten Diagnose verwendet werden.

4 Durchführung der VIBN eines Delta-Roboters

Delta-Roboter werden in der Industrie heute vielfach eingesetzt. Aufgrund ihrer parallelkinematischen Bauart erreichen sie hohe Geschwindigkeiten und Beschleunigungen und einen hohen Durchsatz bei einem geringen Energiebedarf. Diesen Vorteilen gegenüber stehen allerdings eine geringe Lastaufnahme und ein erhöhter Aufwand für die Steuerung, vgl. [13].

Für die Programmierung von Delta-Robotern an Rexroth Industriesteuerungen sind die Transformationen zwischen den Motoren und dem Endeffektor des Roboters bereits in der Entwicklungsumgebungen für Steuerungsapplikationen hinterlegt. Innerhalb der Steuerungsapplikation werden die Algorithmen für die Generierung der Sollpositionen, beispielsweise aus den Input-Daten einer Kamera, implementiert. Diese Sollwerte werden anschließend an den Antriebsregler übertragen, der daraufhin den Motor entsprechend ansteuert. Da allerdings keine Rückführung der Istwerte des Motors an die Steuerung erfolgt, kann nicht garantiert werden, dass der Roboter die vorgegebenen Sollpositionen erreicht.

Ziel der VIBN war es, für eine vorgegebene Steuerungsapplikation zu prüfen, ob der Delta-Roboter die geforderten Sollpositionen erreicht.

Zur Überprüfung der Steuerungsapplikation wurde im Modelica-Tool Dymola [14] ein Mechanikmodell des Roboters erstellt. Dieses wurde über vorhandene Modelica-Modelle von Antriebsregler und Motor mittels Elementen der Bibliothek `mlpi4Modelica` direkt mit der Steuerungsapplikation gekoppelt, vgl. **Bild 7**.

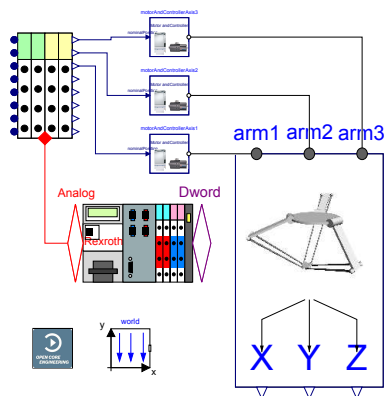


Bild 7 Simulationsmodell des Delta-Roboters mit den Kopplungskomponenten der Modelica-Bibliothek `mlpi4Modelica`

Obwohl vor allem der Mechanikaufbau des Anlagenmodells sehr einfach gestaltet ist, wäre eine HiL-Kopplung zur

Überprüfung der Steuerungsapplikation nicht möglich gewesen, da der mechanische Aufbau des Roboters auf nicht-lineare Gleichungssysteme führt, bei denen nicht sichergestellt werden kann, dass die Lösung innerhalb einer festen Anzahl von Iterationen konvergiert. Die Modelle von Antriebsregler und Motor, die für die Anforderung an die Simulation relevant sind, würden die Echtzeitanforderungen der HiL-Simulation ebenfalls nicht erfüllen.

Mit Hilfe der Rexroth HiL-Kopplung war es dennoch möglich, die Anforderung an die Steuerungsapplikation zu überprüfen. Die Kopplung zwischen Simulations-PC und Steuerungshardware erfolgte, wie bereits zuvor beschrieben, mittels Ethernet-Verbindung.

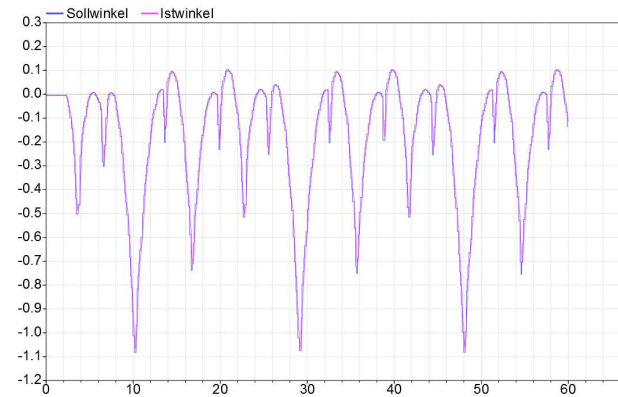


Bild 8 Sollwinkel [Rad] der Steuerung und Istwinkel [Rad] der Simulation eines der Motoren des Delta-Roboters über die Simulationsdauer

Wie in **Bild 8** zu sehen ist, konnte gezeigt werden, dass die Sollpositionen der Steuerungsapplikation vom Roboter erreicht werden können und die Applikation in der realen Anlage verwendet werden kann.

5 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wurde ein Überblick über die virtuelle Inbetriebnahme technischer Systeme gegeben. Speziell auf die Anforderungen, die sich auch im Zuge von Industrie 4.0 herauskristallieren, wurde hierbei eingegangen. Darüber hinaus wurde eine Hardware-In-The-Loop-Methode für Rexroth Industriesteuerungen vorgestellt, welche die übliche HiL-Simulation, im Bezug auf die gefundenen Anforderungen an die virtuelle Inbetriebnahme, erweitert. An einem einfachen Beispiel, der Kopplung einer Rexroth-Industriesteuerung mit einem Delta-Robotermodell, wurde diese neue Technik angewendet.

Die Erstellung der Anlagenmodelle wurde in dieser Arbeit nicht thematisiert, stellt aber einen wesentlichen Aspekt der virtuellen Inbetriebnahme dar. Hierbei spielt speziell die automatisierte Generierung von Modellen aus CAD-Daten einen interessanten Punkt dar. Auch die Erstellung der Steuerungsalgorithmen für die Steuerungsapplikationen ist ein Punkt, in welchem speziell die virtuellen Methoden angreifen können. Angefangen vom Rapid Control Prototyping bis hin zur modellprädiktiven Regelung gibt es hier Möglichkeiten, die Entwicklung und Inbetriebnahme, virtuell wie real, zu verbessern und zu beschleunigen.

6 Literatur

- [1] DIN 19246 1991-6: *Messen, Steuern, Regeln; Abwicklung von Projekten; Begriffe*. Berlin, Beuth Verlag, 1991.
- [2] o. Autor: *VDW: Abteilungsübergreifende Projektierung komplexer Maschinen und Anlagen*. Aachen, WZL, 1997.
- [3] o. Autor: *Umsetzungsempfehlung für das Zukunftsprojekt Industrie 4.0*. o. Ort, BMBF, 2013.
- [4] Dumitrescu, R.: *Systems Engineering für Industrie 4.0*. Mannheim, Vortragsfolien 3DEXPERIENCE Forum, 2014.
- [5] Wünsch, G.: *Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme*. München, Herbert Utz Verlag, 2008.
- [6] Mikelsons, L. et al.: *Symbolic Model Reduction Applied to Realtime Simulation of a Construction Machine*. In: Proceedings of the 7th Modelica Conference, September 20-22, 2009, Como, Italy.
- [7] Hofmann, A. et al.: *Simulating Collisions within the Modelica MultiBody library*. In: Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden.
- [8] Homepage der Modelica Association: www.modelica.org
- [9] Homepage zum OpenCore Interface: <http://www.boschrexroth.com/de/de/produkte/engineering/open-core-engineering/die-features-von-open-core-engineering/open-core-interface/open-core-interface-1> (07. Januar 2015).
- [10] Engels, E., Gabler, T.: *Universelle Programmierschnittstelle für Motion-Logic Systeme*. In: Struktur, Funktionen und Anwendung in Forschung und Lehre, Tagungsband AALE 2012.
- [11] Homepage des OpenCore Engineering Networks <https://www.boschrexroth.com/network> (07. Januar 2015).
- [12] Homepage zu Ogre3d: <http://www.ogre3d.org> (08. Januar 2015).
- [13] Martini, A.: *Delta-Roboter - Aufbau, Arbeitsweise und Anwendung*. o. Ort, o. Jahr, In: Technologien der Fertigungsautomatisierung - Seminarreihe zu ausgewählten Forschungsthemen der industriellen Anwendung. <https://wiki.zimt.unisiegen.de/fertigungsautomatisierung/index.php/Delta-Roboter> (07. Januar 2015).
- [14] Multi-Engineering Modeling and Simulation - Dymola: www.dynasim.se (08. Januar 2015).