

Modellbasierte Entwicklungsmethoden als Enabler von Smart Services im Kontext von Industrie 4.0

Rüdiger Kampfmann M.Sc., Dr.-Ing. Nils Menager

Bosch Rexroth AG, Zum Eisengiesser 1, 97816 Lohr am Main, Deutschland,

{ruediger.kampfmann, nils.menager}@boschrexroth.de

Tel.: 09352/18-6415

Kurzfassung

Ständig steigende Anforderungen an industrielle Anlagen, wie zum Beispiel ein höherer Durchsatz oder mehr Flexibilität, führen zu einer gesteigerten Komplexität dieser Systeme. Zusätzlich verlagert sich immer mehr Funktionalität aus dem Hardware- in den Softwarebereich, so dass dessen Bedeutung stetig zunimmt. Diesem Wandel mit wettbewerbsfähiger Entwicklungszeit zu begegnen, ist eine der wichtigsten Herausforderungen im Automatisierungssektor. Einen Ansatz hierzu stellt die Verwendung modellbasierter Entwicklungsmethoden dar. Während in den frühen Phasen des Entwicklungsprozesses modellbasierte Methoden zunehmend häufiger eingesetzt werden, besteht vor allem in den späteren Entwicklungsphasen sowie in der Phase des Betriebs erheblicher Nachholbedarf. In diesem Beitrag werden zunächst die bereits heute in der Praxis verwendeten Methoden am Beispiel einer komplexen Roboterkinematik vorgestellt. Anschließend wird im Wesentlichen die Phase des Betriebs betrachtet und dargestellt, welche Mehrwerte sich durch die Verwendung so genannter Smart Services auf Basis der bereits vorhandenen physikalischen Simulationsmodelle ergeben.

1 Einleitung

1.1 Motivation

Forderungen seitens der Maschinenbetreiber nach flexibleren und performanteren Verpackungsanlagen führen zu einer Verlagerung der mechanischen Funktionalität hin in den Bereich der Steuerungssoftware und somit zu einer erhöhten Gesamtkomplexität. Weiterhin wird durch die voranschreitende vierte industrielle Revolution und die damit einhergehende Digitalisierung der Softwareanteil an der Gesamtwertschöpfung einer Anlage deutlich gesteigert. Grundidee dieser Revolution sind intelligente, mit Sensorik ausgestattete Produktionsanlagen,

die sich selbst überwachen und autonom auf sich ändernde Bedingungen wie Verschleiß oder einen Produktwechsel reagieren.

Mit dieser Entwicklung muss auch ein Umdenken auf dem Gebiet der Systementwicklung einhergehen, denn für derart komplexe Anlagen versagen bisherige Entwicklungsansätze zu meist, beziehungsweise führen zu nicht wirtschaftlichen Entwicklungskosten. In der Praxis werden häufig die einzelnen physikalischen Domänen, wie Mechanik, Pneumatik, Elektrik und Steuerungstechnik gesondert betrachtet. Dies widerspricht nicht nur der Idee von Industrie 4.0, der Verschmelzung geregelter und gesteuerter Anlagen hin zu cyber-physischen Systemen, sondern führt dazu, dass auftretende Fehler erst bei der Systemintegration erkannt werden. Die Behebung dieser Fehler ist dann mit deutlichen zeitlichen und finanziellen Aufwänden verbunden.

Das modellbasierte Engineering ist ein Ansatz, um mit dieser Herausforderung umzugehen. Der Grundgedanke hierbei ist, den kompletten Lebenszyklus einer Anlage durchgängig mit Modellen zu begleiten. Durchgängigkeit bedeutet in diesem Kontext, dass Modelle von der Planungs-, über die Entwicklungs- bis hin in die Phase des Betriebs kontinuierlich wiederverwendet werden. Dadurch lässt sich einerseits Aufwand durch zeit- und kostenintensive Reimplementierung vermeiden, andererseits können die vorhandenen Informationen durch die Modelle zwischen den unterschiedlichen Phasen auf einfache Weise ausgetauscht werden. Besonders bei Betrachtungen bezüglich der Systemdynamik entsteht hierdurch eine Vielzahl neuer Möglichkeiten, die in diesem Beitrag betrachtet werden. Wichtig bei der Einführung eines modellbasierten Workflows ist, dass dieser möglichst nahtlos in bestehende Prozesse eingreift. Das bedeutet, dass diese nicht ersetzt, sondern an sinnvollen Stellen um die Vorteile der virtuellen Methoden ergänzt werden.

Ein Use-Case des modellbasierten Engineerings, der bereits heute zunehmend in der industriellen Praxis verbreitet ist, ist die Auslegungsunterstützung. Bei diesem Anwendungsfall wird ein Systemmodell der Anlage erstellt, um an diesem mittels Simulation zu überprüfen, ob die Anforderungen an die Dynamik von den ausgewählten Antriebskomponenten erfüllt werden. Gerade im Verpackungsbereich handelt es sich bei den aufgebauten Systemen oft um komplexe Roboterkinematiken, bei denen sich die auftretenden Belastungen nur schwer anhand von Erfahrung vorhersagen lassen. Durch die Verwendung von Simulationen wird diese Problematik umgangen und eine punktgenaue Auslegung reproduzierbar erzielt.

Um mehrachsige Kinematiken möglichst genau abzubilden, ist es allerdings erforderlich, die Steuerungsfunktionalität im Modell genau zu berücksichtigen, da die auftretenden Kräfte stark vom generierten Pfad abhängig sind. Ein Ansatz hierfür ist die Verwendung einer Hardware-

in-the-Loop Simulation, bei der eine reale Steuerung mit einer realen Applikation in die Simulation eingebunden wird. Mit diesem Ansatz lässt sich nicht nur die Auslegung der Antriebskomponenten besser vornehmen, sondern es ergibt sich auch mit der Virtuellen Inbetriebnahme ein weiterer Use-Case des modellbasierten Engineerings.

Der Grundgedanke der virtuellen Inbetriebnahme ist, Teile der Inbetriebnahme an einem virtuellen statt einem realen Prototyp vorzunehmen. Hierdurch kann einerseits bereits mit der Validierung der Steuerungsapplikation begonnen werden, bevor die reale Anlage errichtet wurde, andererseits ist es auf diese Art möglich, gefährliche Maschinenzustände zu testen, ohne Schaden an Mensch und Umwelt zu verursachen. Diese beiden Anwendungsfälle führen zu deutlichen Effizienz- und Qualitätssteigerungen während der Entwicklung einer Anlage.

Modellbasiertes Engineering bietet allerdings während des Betriebs einer Anlage noch deutlich mehr Möglichkeiten, Mehrwert zu generieren. Hierzu werden Modelle, die bereits aus früheren Phasen vorhanden sind, in Verbindung mit vernetzten Anlagen und deren Sensordaten verwendet, um neue Dienste, sogenannte Smart Services, anzubieten.

Um aussagekräftige Ergebnisse zu erzielen, ist es in einem ersten Schritt erforderlich, das vorhandene Modell mittels mathematischer Parameterschätzverfahren an die reale Anlage anzupassen, um so zum Beispiel die echten Reibungsparameter zu ermitteln, die im Rahmen der Auslegungsphase lediglich abgeschätzt wurden. Dieses Modell, welches nun reale Fertigungstoleranzen und Verschleiß berücksichtigt, ist ein Teil des „Digitalen Zwillings“ der Anlage. Dieser ist schließlich zentraler Baustein für die Umsetzung der Smart Services, die beispielsweise durch Diagnose- und Regelungsverfahren neue digitale Geschäftsmodelle ermöglichen.

Im Rahmen dieses Beitrags werden exemplarisch an einer komplexen Roboterkinematik aus dem Verpackungsbereich, einer 4-Achs-Delta-Kinematik, die unterschiedlichen Anwendungsfälle modellbasierter Methoden und der dadurch generierte Mehrwert aufgezeigt. Dabei wird insbesondere Wert darauf gelegt, dass sich diese Methoden möglichst intuitiv in das bereits existierende Umfeld eingliedern, um den Aufwand bei der Einführung modellbasierter Methoden zu reduzieren.

1.2 Aufbau des Beitrags

Dieser Beitrag ist folgendermaßen gegliedert. Zuerst wird in Abschnitt 2 kurz auf die verwendete Modelltechnologie und Toollandschaft eingegangen. Anschließend wird das Systemmodell der Roboterkinematik, bestehend aus Mechanik-, Getriebe- und elektrischen Antriebskomponenten, vorgestellt. Danach wird im nächsten Abschnitt auf Basis dieses Modells die Auswahl der Antriebskomponenten getroffen. Abschnitt 3 zeigt auf, wie dieses Modell verwendet

werden kann, um bei der Applikationserstellung zu unterstützen. Schließlich wird in Abschnitt 4 demonstriert, wie durch die Kombination von modellbasierten Methoden und vernetzten Anlagen Smart Services ermöglicht werden. Abschließend erfolgt in Abschnitt 5 eine Zusammenfassung der Ergebnisse.

2 Modellierung

Zentraler Baustein des modellbasierten Engineerings ist das Modell. Um ein hinreichend genaues Abbild der Verpackungsanlage für die Systemsimulation zu erstellen, müssen im Modell unterschiedliche physikalische Domänen wie Elektrik, Mechanik und Steuerungstechnik abgebildet werden. Daher ist es sinnvoll, eine Modellierungstechnologie zu verwenden, die die domänenübergreifende Simulation unterstützt. Außerdem sollte diese auf einem offenen Standard basieren, um so auch die Verwendung einer Open Source Simulationsumgebung zu ermöglichen. Denn gerade die für die Verpackungsbranche typischen kleinen und mittelständischen Unternehmen scheuen hohe Anfangsinvestitionen bei der Evaluierung neuer Technologien. Open Source Lösungen mindern diese Einstiegsinvestitionen. Mit der Modellierungssprache Modelica existiert ein solcher offener Modellstandard, der die domänenübergreifende Modellierung unterstützt. Gleichzeitig existieren im Modelica-Umfeld mehrere kommerzielle und frei verfügbare Simulationsumgebungen. In diesem Beitrag wird die kommerzielle Modelica-basierte Simulationssoftware Dymola [1] von Dassault Systèmes verwendet, um das Anlagenmodell zu erstellen und zu simulieren.

Bei der in diesem Beitrag betrachteten Anlage handelt es sich um einen 4-Achs-Delta-Roboter. Diese kommen aufgrund ihrer hohen Dynamik sehr häufig im Bereich der Primärverpackung zum Einsatz. Gerade diese hohe Dynamik in Verbindung mit einer komplexen Parallelkinematik erfordert eine detaillierte Betrachtung der Antriebskonfiguration, wie sie nur mithilfe von Simulationsmethoden erreicht werden kann und ist daher ein sehr gutes Beispiel, um die Vorteile des modellbasierten Engineerings zu demonstrieren.

Abbildung 1 zeigt das Blockschaltbild des 4-Achs-Delta-Roboters in Modelica. Wie auf dem Bild zu erkennen ist, werden die 3D-Mechanik, die einzelnen Getriebe und die elektrischen Antriebskomponenten berücksichtigt. Die Roboterachsen werden üblicherweise mit permanent erregten Synchronmotoren angetrieben. Im Systemmodell wird daher sowohl der Antriebsregler als auch der Motor abgebildet. Die Getriebe werden ohne Zahnspiel und Verdrehsteifigkeit mit konstantem Wirkungsgrad und fixer Übersetzung modelliert. Zusätzlich wird eine Reibungskomponente in jedem Antriebsstrang berücksichtigt.

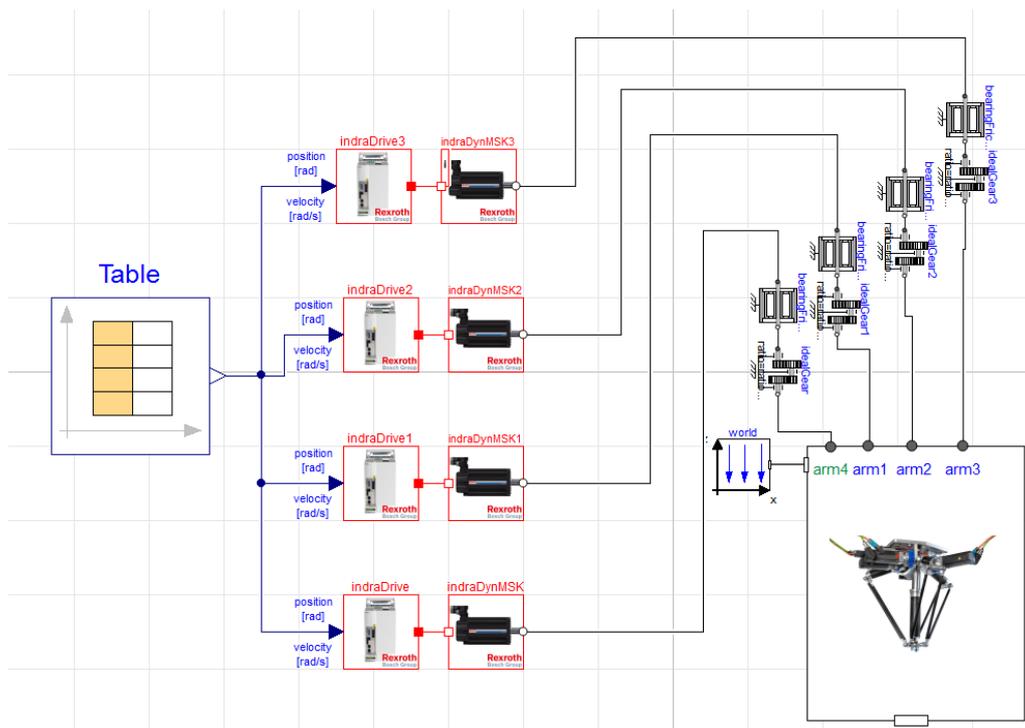


Abbildung 1: Blockschaltbild der 4-Achs-Delta-Mechanik

In Abbildung 2 ist die Animation der Delta-Mechanik zu sehen. Auf das zugehörige Blockschaltbild der Mechanik wird an dieser Stelle verzichtet. Für die Modellierung wurde nur auf Komponenten aus der Modelica Standardbibliothek zurückgegriffen.

In der Standardkonfiguration besitzen die einzelnen Subkomponenten eine Vielzahl an Parametrierungsoptionen. So erfordert zum Beispiel allein eine 3D-Massenkomponente standardmäßig sechs Parameter für den Trägheitstensor. Daher wurde beim Aufbau eine Parametrierungsmaske erstellt, die aus wenigen Basisparametern die restlichen Modellparameter ableitet. Auf diese Art und Weise kann die Parametrierung des Modells auch von weniger erfahrenen Personen vorgenommen werden, da nur die aus der Praxis üblichen Daten erforderlich sind.

Am Blockschaltbild zeigt sich auch der Vorteil der Modelica-Technologie: die objektorientierte Modellierung. Sollte sich beispielsweise herausstellen, dass ein komplexerer Ansatz für die Getriebe-Modellierung erforderlich ist, kann dieser Block sehr einfach ausgetauscht werden, ohne Änderungen an den anderen Komponenten vornehmen zu müssen, wie dies beispielsweise in signalbasierten Simulationsumgebungen wie Simulink der Fall wäre.

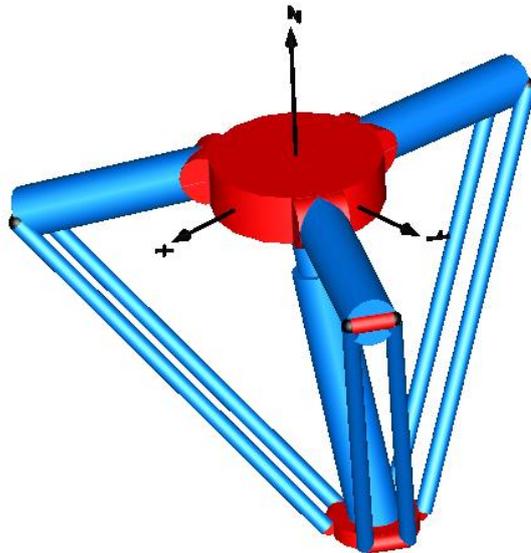


Abbildung 2: 4-Achs-Delta-Mechanik

3 Auslegung der Antriebskomponenten

Üblicherweise bietet jeder Hersteller zur Auslegung der elektrischen Antriebskomponenten ein Software-Tool an, mit dem auf Basis der auftretenden Lastzyklen die Auswahl des Motors und des zugehörigen Antriebsreglers vorgenommen wird. Die Herausforderung bei der Auslegung ist es, von den Anforderungen auf Maschinenebene auf die am Motor auftretenden Belastungen zu schließen, denn üblicherweise ist für den Maschinenbetreiber nur die Gesamtperformance der Anlage und nicht die der einzelnen Komponenten entscheidend. Bei Verpackungsanlagen sind diese Anforderungen üblicherweise eine bestimmte Anzahl an Picks pro Minute. Bei einfachen Kinematiken lassen sich die daraus resultierenden Motorlastprofile eventuell noch per Hand herleiten. Bei komplexeren Kinematiken, wie dem in diesem Beitrag betrachteten Delta-Roboter, ist es weitaus schwieriger, vom Bewegungsprofil am Tool Center Point auf die auftretenden Belastungen der einzelnen Antriebsachsen zu schließen. Bisher wird daher die Auswahl der Antriebskomponenten zumeist allein auf die Erfahrung der entsprechenden Mitarbeiter gestützt. Die Verwendung des Simulationsmodells zur Berechnung der auftretenden Lasten ist daher sinnvoll, um eine punktgenaue Auslegung der Antriebskomponenten

zu ermöglichen. Um mithilfe des Simulationsmodells die auftretenden Belastungen möglichst genau zu berechnen, ist allerdings auch die Kenntnis über das genaue Verfahrensprofil der 4-Achs-Kinematik notwendig, da aus dem generierten Pfad direkt die an der Achse auftretenden Beschleunigungen und somit auch die korrespondierenden Belastungen resultieren. Da sich die generierten Pfade von Steuerungshersteller zu Steuerungshersteller deutlich unterscheiden können, beispielsweise bei Überschleifvorgängen, muss die Pfadgenerierung mit im Simulationsmodell berücksichtigt werden, damit die Auswahl der Antriebskomponenten zur verwendeten Robotersteuerung passt. Hierzu gibt es unterschiedliche Möglichkeiten. So kann beispielsweise die Trajektorie aufgezeichnet und über eine Tabelle im Modell geladen werden oder aber die reale Steuerung wird über Hardware-in-the-Loop Methoden mit dem Simulationsmodell gekoppelt. Problematisch ist, dass in beiden Fällen bereits eine Steuerungsapplikation erstellt werden muss. Diese liegt aber zumeist während der Auslegungsphase noch nicht vor. Eine vorgefertigte Modellkomponente, die die Funktionalität der Pfadgenerierung exakt abbildet und zugleich einfach zu parametrieren ist, ist daher zu favorisieren. Für die Rexroth Steuerung Indraworks MLC wurde eine solche Modellkomponente in der Modellierungssprache Modelica über die Kapselung der originalen Robotik-Firmware in einer Modellkomponente geschaffen. So können mit dem einfach zu parametrierenden Modell aus Abschnitt 2 in Kombination mit der Firmware-Komponente auf einfache Art und Weise Lastprofile berechnet werden. Über einen Export dieser Ergebnisse können diese Profile ins entsprechende Auslegungstool transferiert werden. Um die Auslegung der Antriebskomponenten durchzuführen, werden zuerst aus der CAD-Konstruktion der Anlage die anzufahrenden Endpunkte der Pfade ermittelt. In der Simulationsumgebung werden diese Pfade dann über die Pfadgenerierungskomponente einprogrammiert. Genauso erfolgt die Parametrierung der Längen und Massen der Roboterkinematik. Hier kann einerseits bei einer Fremdkinematik auf das Datenblatt des Herstellers zurückgegriffen werden, oder aber diese werden ebenfalls aus dem CAD-Modell abgeleitet. Weiterhin muss die Parametrierung der Geschwindigkeits- und Beschleunigungswerte vorgenommen werden. Außerdem kann eine Pick-and-Place-Zeit eingegeben werden. Nachdem die Simulation durchgeführt wurde, werden diese Ergebnisse exportiert und im Auslegungstool geladen, sodass die übliche Auslegung vorgenommen werden kann. Durch das Simulationsmodell wird somit die Funktionalität des Auslegungstools erweitert, sodass auch Antriebskomponenten für komplexe Roboterkinematiken punktgenau ausgelegt werden können.

Großer Vorteil bei diesem Ansatz ist zum Beispiel, dass bei bekanntem Verfahrensprofil sehr einfach für unterschiedliche Getriebekonfigurationen Lastprofile abgeleitet werden können, denn gerade die exakte Abstimmung der Getriebeübersetzung mit den Motorkomponenten birgt in

der Praxis häufig Optimierungspotential. So ist es sehr schwierig, die Getriebeübersetzung derart zu wählen, dass das beste Beschleunigungs-Trägheitsverhältnis erreicht wird.

4 Applikationserstellung

Neben der Auslegung der Antriebskomponenten kann das Anlagenmodell auch unterstützend bei der Erstellung der Steuerungsapplikation zum Einsatz kommen. Einerseits können unter Verwendung der Methodik der Codegenerierung Modellbestandteile auf die Steuerung transferiert werden, um diese beispielsweise zur modellbasierten Vorsteuerung zu verwenden. Andererseits kann, wie schon im vorherigen Abschnitt angedeutet, bereits am Anlagenmodell mit dem Test der Steuerungsapplikation begonnen werden. Dieser Anwendungsfall wird als virtuelle Inbetriebnahme bezeichnet.

4.1 Codegenerierung

Die Codegenerierung ist ein wichtiger Aspekt des modellbasierten Engineerings und bedeutet, automatisch Quellcode aus vorhandenen Modellen abzuleiten, um diesen in anderen Umgebungen auszuführen. Hierdurch wird die Verwendung von Modellen über Toolgrenzen hinweg ermöglicht, sodass Modelle sowohl in anderen Softwaretools angebunden als auch auf eingebetteten Systemen ausgeführt werden können.

Um diese Methodik für Industriesteuerungen zu nutzen, ist eine auf die entsprechende Steuerungshardware zugeschnittene Toolkette erforderlich. Diese übersetzt das Modell in Steuerungscode und generiert die dafür erforderlichen Interfaces. Für Rexroth Steuerungen ist auf Basis des Functional Mock-up Interface (FMI) eine Toolkette verfügbar, die diese Funktionalität implementiert. Das Functional Mock-up Interface ist ein offener Standard, um Modelle zwischen unterschiedlichen Tools auszutauschen [2]. Weiterhin wurde bei der Definition des Standards der Code-Export zur Verwendung der Modelle auf eingebetteten Systemen vorgesehen. Der Vorteil bei der Verwendung dieses Standards ist, dass dieser nicht nur von fast allen Modelica-basierten Tools, sondern auch von einer Vielzahl weiterer Simulationstools unterstützt wird.

Abbildung 3 zeigt die Toolkette. Aus dem Simulationstool wird das Modell über das Functional Mock-up Interface exportiert. Danach wird das Modell um bestimmte Routinen ergänzt, die für die Simulation des Modells in Echtzeit auf der Steuerung notwendig sind, und für die Zielhardware kompiliert. Wird dieses Modul auf die Steuerung geladen, kann es über einen IEC61131 Funktionsblock aufgerufen werden. Zur Verwendung dieser Toolkette sind daher keine weiteren Programmierkenntnisse erforderlich. Die Erstellung der Applikation wird weiterhin ausschließlich in dem dafür üblichen Engineeringtool vorgenommen.

Für Hersteller von Verpackungsanlagen leiten sich aus der Codegenerierung einige Anwendungsfälle ab. Zum Beispiel kann, sofern für die betrachtete Anlage ein spezieller Regler erforderlich ist, dieser aus dem Simulationsmodell abgeleitet werden. Dieser Use-Case wird als Rapid Control Prototyping (RCP) bezeichnet. Insbesondere für Anlagen mit hydraulischen Komponenten müssen häufig spezielle Regelungsalgorithmen entwickelt werden. Um ein möglichst genaues virtuelles Abbild der Anlage für die Auslegung zu erhalten, ist es erforderlich, den Regelungsalgorithmus detailliert im Simulationsmodell abzubilden. Wird dieser Regler nun mittels RCP auf die Steuerung transferiert, wird die Reimplementierung des Reglers vermieden. Bei der in diesem Beitrag betrachteten 4-Achs-Kinematik ist dies allerdings nicht erforderlich. Die Regelungsalgorithmen, die für den Betrieb der Servomotoren notwendig sind, sind bereits auf den Antriebsreglern implementiert.

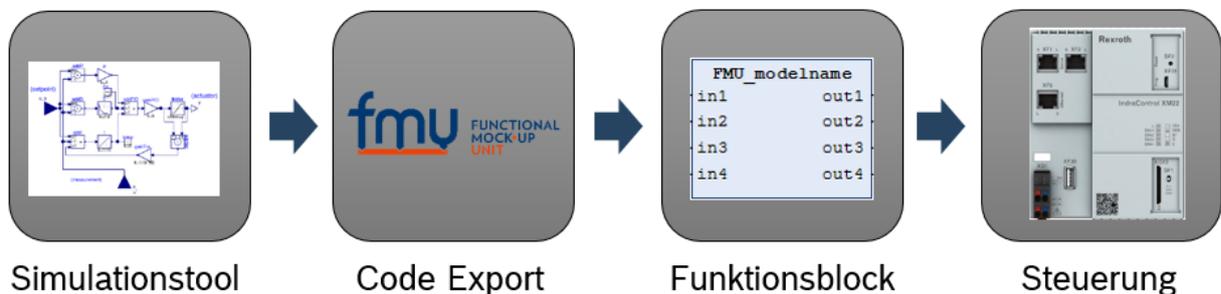


Abbildung 3: Toolkette für Codegenerierung

Allerdings gibt es neben der Verwendung des Reglers auf der Steuerung weitere Anwendungsfälle, die auf der Verwendung des Modells der Regelstrecke auf der Steuerung basieren. Auf diese Verfahren wird in Abschnitt 5 eingegangen.

4.2 Virtuelle Inbetriebnahme

Neben der Ableitung von Applikationsbestandteilen aus dem bereits für die Auslegung erstellten Modell kann mithilfe des Anlagenmodells auch eine Verifikation der Steuerungsapplikation vorgenommen werden. Die Verwendung der virtuellen anstatt der realen Anlage bietet zahl-

reiche Vorteile. Das virtuelle Abbild ist bereits in einer sehr frühen Phase der Entwicklung verfügbar. Wird bereits in dieser Phase mit der Validierung der Steuerungsapplikation begonnen, kann die für die Inbetriebnahme an der realen Anlage notwendige Zeit und somit die Time-to-Market deutlich reduziert werden. Am virtuellen Prototyp können auf einfache Art und Weise gefährliche Maschinenzustände getestet werden ohne Schaden an Mensch, Umwelt und Maschine zu verursachen. Ein sehr gutes Beispiel stellt die betrachtete 4-Achs-Kinematik dar, bei der die Haltekraft des Pickers die mögliche Maximalbeschleunigung limitiert. Mithilfe der virtuellen Inbetriebnahme können die resultierenden Kräfte am Picker sehr gut abgeschätzt und in der Steuerungsapplikation berücksichtigt werden, so dass sich bei der Inbetriebnahme an der realen Anlage kein Produkt versehentlich vom Picker lösen kann.

Für die Durchführung der virtuellen Inbetriebnahme muss das Anlagenmodell mit der Steuerungsapplikation verbunden werden. Hierzu existieren unterschiedliche Kopplungsstrategien. Im Sinne der Durchgängigkeit sollte für den Aufbau dieser Kopplung möglichst wenig Aufwand, beispielsweise durch Anpassungen am Modell, notwendig sein. Ebenfalls sollte nach der Durchführung der virtuellen Inbetriebnahme der Wechsel zur realen Anlage möglichst einfach sein, daher ist die Kopplung via Hardware-in-the-Loop in Kombination mit einem Modellinterface in der Simulationsumgebung ein geeigneter Ansatz. Bei Hardware-in-the-Loop Verfahren wird das Applikationsprogramm auf der realen Steuerung ausgeführt. Der Wechsel zur realen Anlagen verursacht daher keinen großen Aufwand. Im Anlagenmodell muss allerdings über ein Interface auf Steuerungsvariablen zugegriffen werden. Mit dem Open Core Interface [3] hat Bosch Rexroth die Möglichkeit geschaffen, auf Steuerungsfunktionen über unterschiedliche Programmiersprachen zuzugreifen. Durch die Bibliothek mlpi4Modelica besteht die Möglichkeit, diese Funktionalität von der Modellierungssprache Modelica aus zu verwenden. Ein wesentlicher Baustein zur Durchführung der virtuellen Inbetriebnahme ist die Synchronisation der Steuerungszeit mit dem Ablauf der Simulation, denn üblicherweise erfolgt die Abarbeitung der Steuerungsroutinen in Echtzeit. Mithilfe des Open Core Interfaces kann allerdings auch diese Problematik umgangen werden, indem, solange die Simulation durchgeführt wird, der interne Taktgeber der Steuerung von der Simulation aus gesteuert wird. Simulation und Applikation werden folglich synchron und deterministisch ausgeführt. Die Ausführung der Applikation auf der Steuerung wird auf die Simulationsgeschwindigkeit reduziert. Dadurch entfallen eventuell erforderliche Anpassungen am Modell, da dieses nicht in Echtzeit simuliert werden muss.

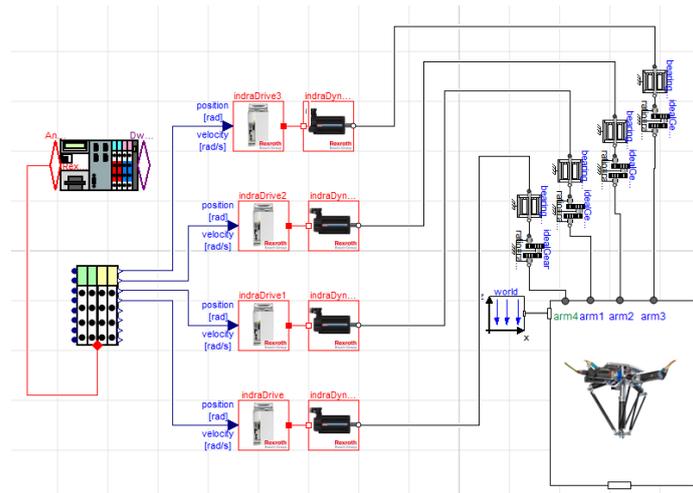


Abbildung 4: 4-Achs-Delta-Mechanik mit Hardware-in-the-Loop Kopplung

Abbildung 4 zeigt das Blockschaftbild des für die Durchführung der virtuellen Inbetriebnahme verwendeten Modells. Deutlich zu erkennen ist, dass die Motorsollwerte im Unterschied zu Abbildung 1 über ein Interface zur Steuerung generiert werden.

5 Smart Services

Neben den in diesem Beitrag bereits vorgestellten, in der Praxis etablierten Anwendungsfällen des modellbasierten Engineerings eröffnen sich mit der voranschreitenden Vernetzung im Kontext von Industrie 4.0 neue Möglichkeiten. Moderne, mit zahlreichen Sensoren ausgestattete intelligente Anlagen zeichnen stetig Daten der Antriebskomponenten auf. Werden diese Daten mit den vorhandenen physikalischen Simulationsmodellen verknüpft, lässt sich ein exaktes Modell der Anlage ableiten. Auf Basis dieses Modells können anschließend neue digitale Dienstleistungen, die auch als Smart Services bezeichnet werden, implementiert werden. Dadurch wird es ermöglicht, durch die Verwendung der Modelle nicht nur bei der Entwicklung der Anlage zu unterstützen, sondern auch kontinuierlich während des Betriebes Mehrwert zu generieren. Beispiele für diese Dienste sind die Detektion von Verschleiß, die Erhöhung der Regelgüte sowie die Diagnose von Kollisionen. Für all diese Verfahren ist das zuvor erwähnte exakte Anlagenmodell notwendig. Die direkte Verwendung des für die Auslegung erstellten Modells ist nicht möglich, da in diesem Fertigungstoleranzen oder Verschleiß nicht berücksichtigt wurden. Allerdings wurden bereits alle für die Dynamik wesentliche Effekte berücksichtigt. Es handelt sich bei den notwendigen Modellanpassungen daher nicht um strukturelle Änderungen, sondern nur um eine leicht geänderte Parametrierung. Für diesen Fall wurden Parameterschätzverfahren entwickelt, die aus Messwerten über eine Variation der Modellparameter die wahrscheinlichsten Parameter berechnen. Diese Methodik wird auch als

Maximum Likelihood Estimation bezeichnet. In [4] wurde ein Softwareframework vorgestellt, das auf diesem Verfahren basiert und einen automatischen Workflow ermöglicht, um für Modelle, die über das Functional Mock-up Interface bereitgestellt werden, den zu den Messwerten korrespondierenden Parametersatz zu berechnen. Wird dieses Framework auf einem Server ausgeführt und mit der vernetzten Produktionsanlage verbunden, kann über die aufgezeichneten Sensordaten in regelmäßigen Zeitabständen der entsprechende Parametersatz berechnet werden. Um mit dieser Methodik möglichst viele Steuerungshersteller zu erreichen, ist eine Standardisierung der verwendeten Protokolle zum Austausch von Messdaten und Parametern sinnvoll. Aus diesem Vorgehen ergeben sich zahlreiche Anwendungsfälle. Wird die zeitliche Änderung der Parameterdaten überwacht, kann über deren Änderung Verschleiß detektiert werden. Weitere Anwendungsfälle basieren auf der in Abschnitt 4.1 vorgestellten Codegenerierung. Werden die exakten Modellparameter zur Steuerung zurückgesendet, kann das Robotermodell auf der Steuerung verwendet werden, um die Belastungen der einzelnen Achsen vorherzusagen. Diese Information wird dann an die einzelnen Antriebsregler weitergeleitet, um mittels modellbasierter Vorsteuerung die Regelgüte zu erhöhen. Andererseits kann das Anlagenmodell verwendet werden, um über einen Vergleich der berechneten und gemessenen Werte ein Fehlverhalten der Anlage zu diagnostizieren.

Parameter	Einheit	Startwert	Geschätzter Wert
Masse Oberarm	[kg]	0.1	0.08
Masse Unterarm	[kg]	1.5	1.74
Masse Basisplatte	[kg]	0.87	1.1
Motor und Getriebe Trägheit	[kgm ²]	0.000144	0.000155
Getriebewirkungsgrad	[1]	1.0	0.907
Coloumb Reibwert	[Nm]	0.12	0.105
Viskoser Reibwert	[Nms/ rad]	0.001	0.00128

Tabelle 1: Parameterwerte der Delta-Kinematik

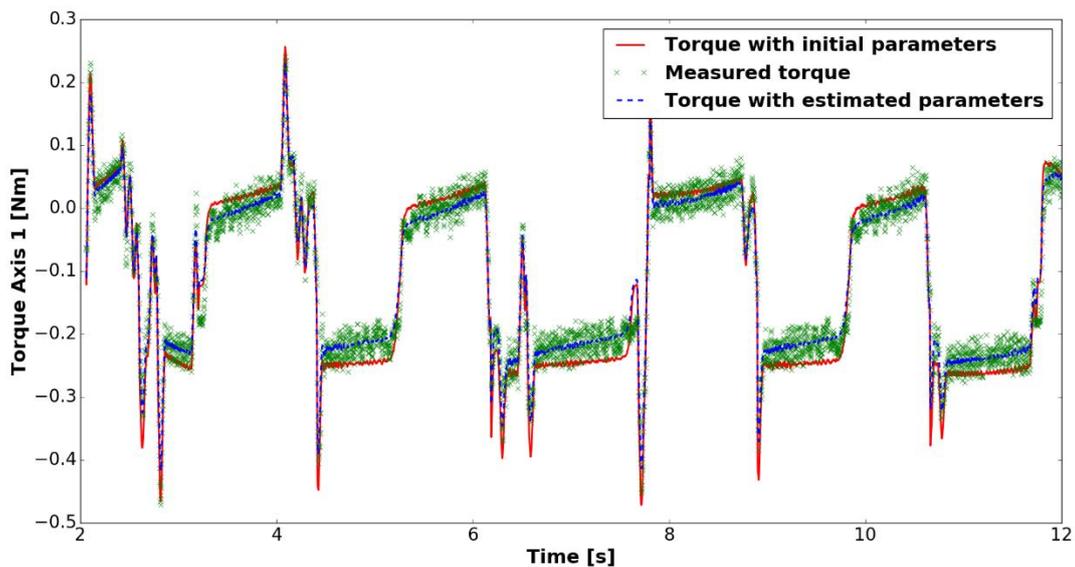


Abbildung 5: Drehmomentwerte für eine Roboterachse

Beispielsweise indizieren bei der Delta-Kinematik erhöhte Drehmomente an den Antriebsachsen eine Kollision. Bisher kann oftmals nur ein bestimmter Drehmomentbereich überwacht werden, eine exakte Vorhersage des aktuellen Sollmoments ist noch nicht möglich. Ein wesentlicher Vorteil dieser Smart Services ist, dass diese direkt aus den vorhandenen physikalischen Modellen abgeleitet werden. So können mit diesem Ansatz beliebige vernetzte Anlagen mit neuen Diensten erweitert werden, um sich so selbst zu überwachen und auf geänderte Bedingungen einzustellen, ganz im Sinne von Industrie 4.0.

An der in diesem Beitrag betrachteten 4-Achs-Delta-Kinematik lässt sich diese Methodik sehr gut demonstrieren. Die reale Anlage besteht in diesem Fall aus einer Autonox24 Kinematik, die von 4 Rexroth MS2N Servomotoren angetrieben wird. Diese werden über eine Rexroth Indraworks MLC Steuerung kommandiert. Über die I4.0-Fähigkeiten der Steuerung können die auftretenden Drehmomente zum Parameterschätzer gesendet werden.

Tabelle 1 zeigt die Änderung des identifizierten Parametersatzes gegenüber der für die Auslegung verwendeten Parameter. Gerade im Bereich der Reibung zeigen sich deutliche Änderungen.

Anhand von Abbildung 5 ist ersichtlich, dass mit den geänderten Parameterwertem das an der Anlage auftretende Drehmoment wesentlich genauer vorhergesagt werden kann. Dieses Modell kann nun auf der Steuerung zur Verbesserung der Regelung verwendet werden.

6 Zusammenfassung

In diesem Beitrag wurden unterschiedliche Anwendungsfälle des modellbasierten Engineerings an einem praxisrelevanten Beispiel aus dem Verpackungsbereich demonstriert und deren Vorteile zur Effizienz- und Qualitätssteigerung in allen Phasen des Lebenszyklus einer Anlage aufgezeigt. Durch diese durchgängige Nutzung von Modellen über den gesamten Entwicklungsprozess hinweg bis in den Betrieb lässt sich das gesamte Potential modellbasierter Entwicklungsmethoden ausschöpfen.

7 Literatur

- [1] Dassault Systèmes, <https://www.3ds.com>, abgerufen am 19.01.2018.
- [2] BLOCHWITZ, Torsten, et al. The functional mockup interface for tool independent exchange of simulation models. In: Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical University; Dresden; Germany. Linköping University Electronic Press, 2011. S. 105-114.
- [3] Elmar Engels et Thomas Gabler, Universelle Programmierschnittstelle für Motion-Logic Systeme : Struktur, Funktionen und Anwendung in der Forschung und Lehre, Tagungsband AALE 2012, 2012.
- [4] Rüdiger Kampfmann, Danny Mösch, Nils Menager, Parameter Estimation based on FMI, Proceedings of the 12th International Modelica Conference, Prague, Tschechische Republik, 15-17 Mai, 2017, Volume , Issue 132, 2017-07-04, Seiten 313-319, ISSN 1650-3740