

9. Fachtagung Baumaschinentechnik 2022

Automatisierung, Antriebssysteme, Bauverfahren

3D-Bahnkurvensteuerung eines Auslegers am Beispiel einer Hubarbeitsbühne

Benedikt Müller, Heiko Baum, Raphael Alt (FLUIDON)
Markus Beckers (meastream)



Dr. Raphael Alt
FLUIDON Gesellschaft für Fluidtechnik mbH
Jülicher Straße 338a
D – 52070 Aachen
info@fluidon.com



meastream GmbH
Gartenstraße 38
D – 52249 Eschweiler

3D-Bahnkurvensteuerung eines Auslegers am Beispiel einer Hubarbeitsbühne

Die automatische 3D-Bahnkurvensteuerung des Auslegers einer Hubarbeitsbühne bedarf der Orchestrierung von Teilsystemen der Mechanik, Hydraulik, Elektronik und Steuerungstechnik. Die modellbasierte Entwicklung bietet hierfür das geeignete Entwicklungswerkzeug, das vollständig durch die Digital-Twin-Methodik unterstützt wird und somit einen Teil eines zukünftigen Produktlebenszyklusmanagements abdeckt. Einer der offensichtlichsten Vorteile der virtuellen Inbetriebnahme ist der Zeitgewinn im Vergleich zum klassischen Prozess der sequenziellen Produktentwicklung, da Steuerungsentwicklung und -test bereits zu einem sehr frühen Projektstadium erfolgen.

1 Einleitung

Haben Sie schon mal versucht, eine Hubarbeitsbühne mit einem mehrteiligen Ausleger entlang eines Objekts zu verfahren, womöglich während eines Arbeitsvorgangs? Gar nicht so einfach, all die Stellachsen mit den Joysticks koordiniert zu bewegen!

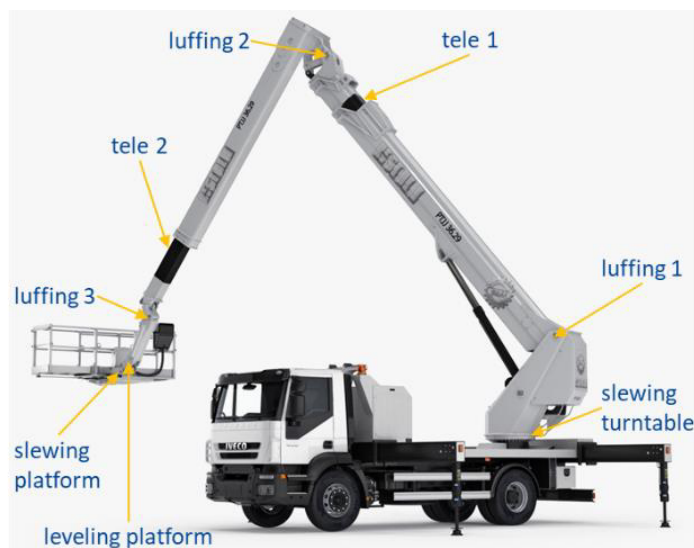


Bild 1: Bewegungsachsen der mobilen Hubarbeitsbühne

Bei der im **Bild 1** dargestellten Arbeitsbühne müssen bis zu 7 Achsen vom Bediener betätigt werden. Lediglich die Nivellierung der Arbeitsplattform erfolgt automatisch. Allein durch die begrenzte Anzahl an Achsen von zwei Joystick können nicht alle Achsen gleichzeitig betätigt werden, was für eine beliebige kontinuierliche Bewegung notwendig wäre.

Das Ziel der hier vorgestellten Entwicklung war es daher, dem Anwender eine deutlich einfacher zu bedienende Arbeitsplattform zur Verfügung zu stellen. Der Bediener muss lediglich noch die gewünschte Bewegungsrichtung im Raum festlegen – die Steuerung kümmert sich um die Umsetzung in Bewegungsvorgaben für die einzelnen Achsen, die dann automatisiert ausgeführt werden.

Die Entwicklung der Bahnkurvensteuerung und der Achsregler erfolgt mit Hilfe von FLUIDON | **Cube**, einer Umgebung für die modellbasierte Entwicklung. Das echtzeitfähige

modulare Simulationsmodell wird hier mit der Steuerungshardware gekoppelt, auf der die Achsregler und die Bahnkurvensteuerung implementiert werden.

2 Zielsetzung

Mit der Bahnkurvensteuerung wird eine deutlich vereinfachte Bedienung der Hubarbeitsbühne erreicht. Dabei können verschiedene Modi umgesetzt werden, die der Anwender je nach Arbeitssituation auswählt: Vorab festgelegte Trajektorien sind dann sinnvoll, wenn die Plattform z.B. entlang von Fassaden bewegt werden soll, wie beispielsweise für Reinigungsarbeiten üblich. Auch für wiederkehrende Arbeiten kann dieser Modus sinnvoll eingesetzt werden. Bei komplexeren Bewegungen und einmalig auszuführenden Arbeiten erfolgt hingegen eine direkte Vorgabe der Bewegungsrichtung der Plattform durch den Anwender, z.B. über einen 3D-Joystick.

Für den Anwender ergibt sich aus der Bahnkurvensteuerung eine vereinfachte Bedienung mit einer Reihe von Vorteilen wie

- Erhöhte Arbeitsgeschwindigkeit und Produktivität
- Deutlich reduzierte Arbeitszeit
- Bediener kann sich auf seine Kernaufgabe konzentrieren
- Reduzierter Schulungsbedarf
- Gleichmäßige Plattformbewegung
- Verbesserte Sicherheit

Die technische Umsetzung der Bahnkurvensteuerung erfordert allerdings ein komplexes Zusammenspiel von Mechanik, Hydraulik und Steuerung:

- Je nach Position und gewünschter Bewegungsrichtung müssen bis zu 8 Achsen synchron bewegt werden, was die direkte Betätigung der Achsen durch den Anwender schwierig gestaltet. Die Steuerung soll daher eine Richtungs- oder Bahnvorgabe des Bediener in eine Kombination aus Bewegungsvorgaben für die 8 Achsen umsetzen.
- Die geforderten Bewegungen müssen von den Achsreglern der 8 Achsen in Stellsignale der Ventile gewandelt werden. Hohe Massen- und Reibkräfte und ein großer Geschwindigkeitsbereich von der langsamen kontinuierlichen Arbeitsbewegung bis zur schnellen Zustellbewegung stellen hohe Anforderungen an diese Regler. Hier wird besonders das komplexe Zusammenwirken von Mechanik (z.B. Kinematik, Reibverhältnisse), Hydraulik und Regelungstechnik deutlich.
- Aufgrund der begrenzten installierten Leistung können nicht alle Bewegungen gleichzeitig mit ihrer maximal möglichen Geschwindigkeit ausgeführt werden. Ein zwischen Achs- und Bahnsteuerung geschalteter Teil der Steuerung muss daher sinnvolle Bewegungsvorgaben zwischen den Teilnehmern vermitteln, wenn die Leistungsgrenze erreicht wird.

Der Prototyp der Bahnkurvensteuerung ist das Ergebnis der Zusammenarbeit von meastream und Fluidon: meastream bringt seine Erfahrung mit der Optimierung mehrachsiger Bewegungen und der Berechnung von Trajektorien auch für unterbestimmte

Systeme ein, während Fluidon für die Modellierung und Simulation des mechatronischen Systems, die Entwicklung der Achsregler und die Steuerung verantwortlich zeichnet.

3 Modellbasierte Entwicklung

Komplexe Produkte, insbesondere die Steuerungs- und Antriebssysteme multifunktionaler Arbeitsmaschinen, können nur effizient entwickelt werden, wenn die Integration der verschiedenen Domänen wie z.B. Mechanik, Hydraulik, Elektronik und Steuerungstechnik früh im Entwicklungsprozess erfolgt und regelmäßig aktualisiert wird.

Bereits der Entwurf einer solchen Maschine wirft viele Fragen auf: Die Kinematik der Maschine bestimmt deren Bewegungsmöglichkeiten. Anlenkpunkte von Zylindern, Überlappung von Teleskopröhren, Reibverhältnisse und Massenkräfte beeinflussen jedoch auch wesentlich die Anforderungen an das Hydrauliksystem oder die Steuerung. Diese Schnittstellen zwischen den Domänen führen häufig zu Problemen im Entwicklungsprozess, wenn Informationen unzureichend synchronisiert werden oder schlichtweg die eine Seite die andere nicht versteht. Ein Mittel, dieser Schnittstellenproblematik zu begegnen, ist die möglichst frühe Systemintegration.

Die modellbasierte Entwicklung stellt hierzu einen Werkzeugkasten zur Verfügung, mit dem die Systemintegration anhand eines virtuellen Abbilds (digitaler Zwilling) der Maschine erfolgt und durch den die komplexen und vernetzten Aufgabenstellungen der Systemintegration besser und effizienter erledigt werden können (**Bild 2**).

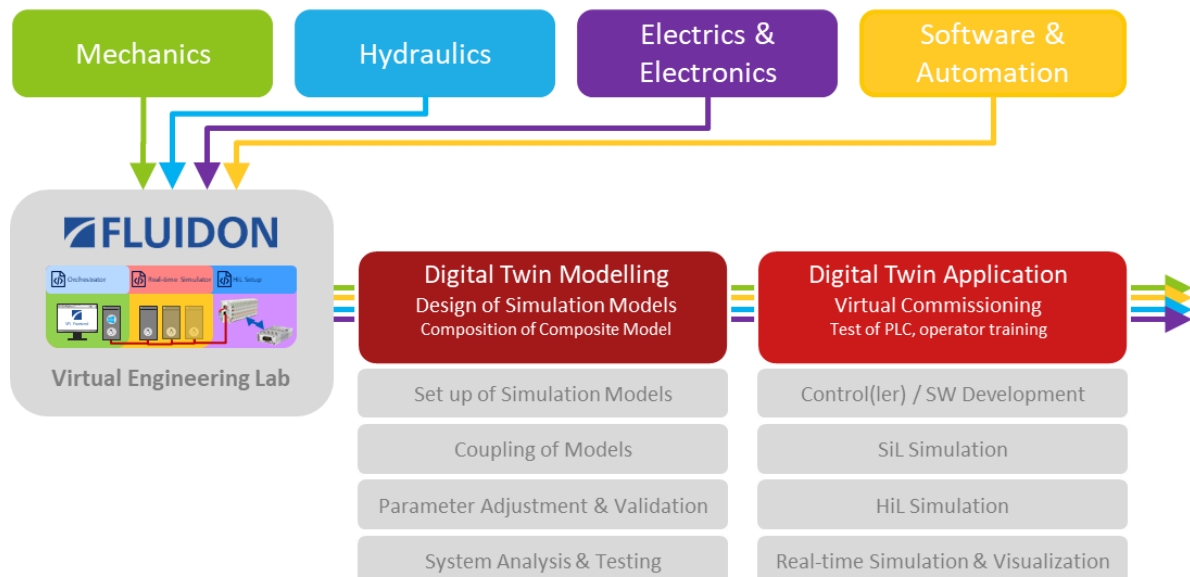


Bild 2: Domänenübergreifende, modellbasierte Entwicklung

3.1 Modularer digitaler Zwilling

„A digital twin is a virtual representation that serves as the real-time digital counterpart of a physical object or process.“ [1]

Zur modellbasierten Entwicklung gilt es zunächst, die für die Zielsetzung entscheidenden Effekte im Modell abzubilden. Dies sollte nach dem Prinzip „so einfach wie möglich, so

genau wie nötig“ erfolgen, da eine genauere Abbildung meist auch mit hohem Rechenaufwand einher geht. Wird ein Modell für die Steuerungsentwicklung benötigt, so ist es meist auch erforderlich oder zumindest wünschenswert, dass dieses in Echtzeit berechnet werden kann.

Diese Echtzeitfähigkeit ist, gerade bei komplexen hydraulisch angetriebenen Maschinen, eine der großen Herausforderungen bei der Entwicklung eines digitalen Zwillings. Warum ist das gerade bei hydraulischen Antrieben so? Die Antwort liegt in der im Vergleich zu anderen Anwendungsfeldern recht kleinen erforderlichen Simulationsschrittweite (**Bild 3**). Typische Schrittweiten für die Hydrauliksimulation liegen in der Größenordnung $1e-6$ bis $1e-5$ s, während z.B. mechanische Mehrkörpersimulationsmodelle oft mit Schrittweiten von $1e-4$ bis $1e-3$ s ausreichend genau und numerisch stabil rechnen. Die meist komplexen und stark vernetzten Hydrauliksysteme sind daher auch mit sehr leistungsfähigen Rechnern nicht ohne Weiteres in Echtzeit zu simulieren.

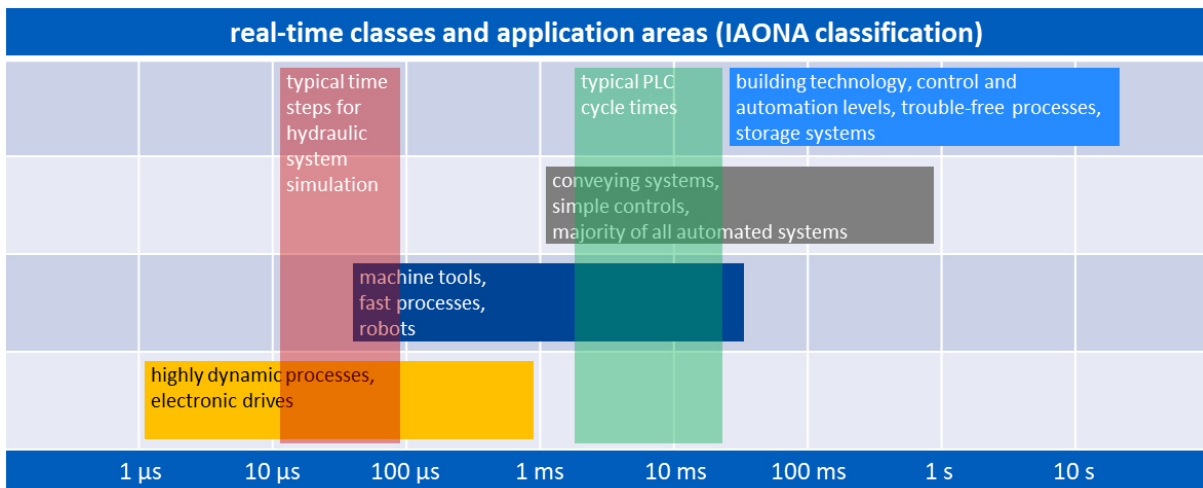


Bild 3: Echtzeitanforderungen in der Hydrauliksimulation vs. IAONA-Klassifizierung

Ein Ansatz, auch komplexe Modelle echtzeitfähig zu machen, liegt in der Parallelisierung von Berechnungsvorgängen. Dabei gilt es jedoch, die richtigen Pakete zu schnüren: Da die Verwaltung der Threads ebenfalls Rechenaufwand und Zeit kostet, ist eine parallele Berechnung nicht umsonst zu haben. Erfolgt die Parallelisierung mit einer zu feingranularen Aufteilung, so kann sehr schnell der Zeitgewinn durch zusätzlichen Overhead wieder aufgeessen werden.

Das im Folgenden als Beispiel gezeigte Modell ist daher nicht auf der Ebene von Bauteilen oder gar einzelnen Rechenvorgängen parallelisiert, sondern auf Modulebene. Dabei stellt ein Modul typischerweise eine funktionale Einheit, z.B. den hydraulischen Antrieb einer Achse dar.

3.1.1 Submodell Mechanik

Die FMU der Mechanik wurde im gezeigten Beispiel in OpenModelica erstellt. Da das mechanische Modell der Hubarbeitsbühne, im Gegensatz zur Hydraulik, bereits als Gesamtmodell echtzeitfähig ist, muss es nicht weiter modularisiert werden, sondern wird als ein Modul im Gesamtmodell abgebildet.

3.1.2 Submodelle Hydraulik

Bild 4 zeigt beispielhaft das DSHplus-Modell einer typischen hydraulischen Antriebsachse, wie sie in der Hubarbeitsbühne eingesetzt wird. Zu erkennen sind das LS-Wegeventil, die Senkbremsventile, der Zylinder und die Verbindungsleitungen. Über IO-Bauteile werden die Werte mit den weiteren Teilmodellen, wie z.B. anderen hydraulischen Achsen, der Druckversorgung oder dem Mechanikmodell ausgetauscht. Eine hydraulische Verbindungsleitung wird an dieser Schnittstelle durch die Austauschgrößen Druck p und Volumenstrom Q dargestellt: Das Modell der Achse erhält den Wert von p als Eingang und gibt den berechneten Volumenstrom Q als Ausgang an die Druckversorgung zurück. Ähnlich wird eine mechanische Verbindung beschrieben durch den Kraftausgang F und die Eingänge Weg x und Geschwindigkeit v , die im Mechanikmodell berechnet werden. [2]

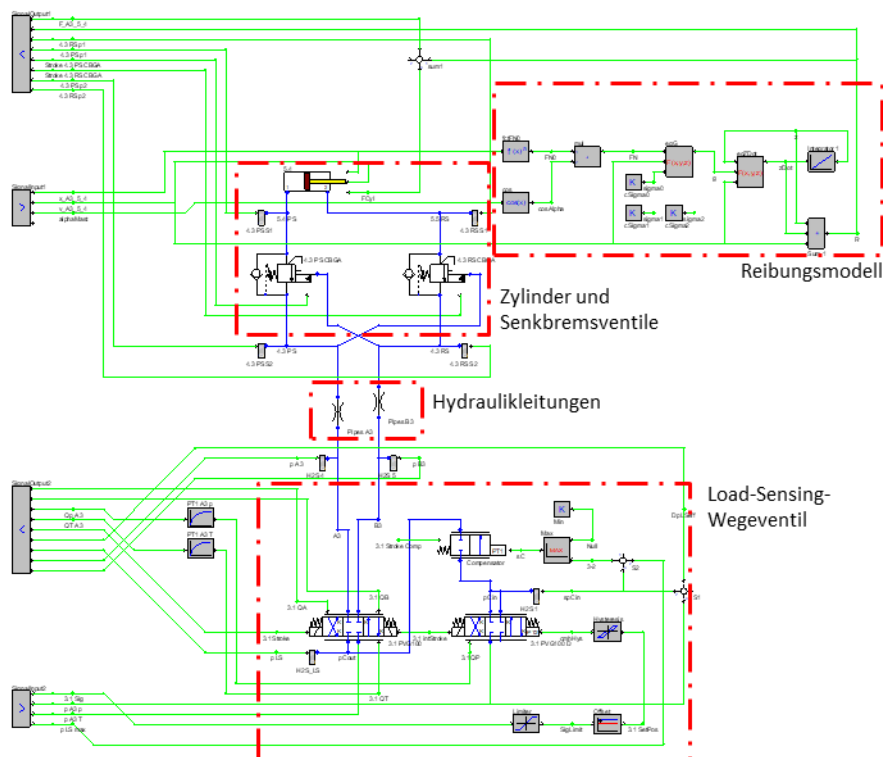


Bild 4: Submodell einer hydraulischen Achse mit Reibungsmodell

Die Luffing-Achsen der Auslegerteile erzeugen Drehbewegungen um die Gelenke der Ausleger, kombiniert mit Drehbewegungen in den Gelenkaugen der Zylinder und ggf. zusätzlichen Hebeln. All diesen Gelenken ist gemein, dass die Reibkräfte auf kleine Hebelarme wirken. Andererseits wirken bei diesen Bewegungen große Massen und Inertialmomente. Daher wird bei diesen Achsen die Reibung vernachlässigt.

Bei den Teleskopierbewegungen hat die Reibung im System hingegen wesentlichen Einfluss auf die Auslegung der Hydraulik und des Achsreglers und muss daher im Modell berücksichtigt werden. Die Reibung wird nicht im Mechanikmodell, sondern in den Submodellen der hydraulischen Achsen abgebildet. Dies hat den Vorteil, dass hier bereits mit kleiner Schrittweite gerechnet wird und das Reibmodell nicht zu numerischen Instabilitäten führt. Die Reibung wird nach dem LuGre-Modell abgebildet [3]. Dieser

Modellierungsansatz bildet die Reibung bei kleinen Bewegungen und Richtungswechseln besser ab als z.B. ein einfaches Stribeck-Modell. Zugleich verhält er sich numerisch stabiler und ist daher in der numerischen Simulation weit verbreitet.

Das Teilmodell einer Achse kann für sich in Betrieb genommen werden. Hierbei zeigen sich bereits erste mögliche Schwachstellen der technischen Realisierung, die damit früh im Entwicklungsprozess und nicht erst am realen Prototyp korrigiert werden können.

3.1.3 Composite-Modell

Die Module werden als sogenannte Functional Mockup Units (FMU) nach dem Standard FMI 2.0 Co-Simulation aus dem jeweiligen Simulationstool exportiert. „The Functional Mock-up Interface (FMI) is a free standard that defines a container and an interface to exchange dynamic models [...]“ [4]. Die FMU ist eine Datei, die das Simulationsmodell enthält, das die standardisierte FMI-Schnittstelle zur Verfügung stellt. Eine für die Co-Simulation erstellte FMU enthält zusätzlich den passenden Solver für das Simulationsmodell.

Der Zusammenbau der Gesamtmaschine aus den so erstellten Modulen erfolgt im sogenannten Composite-Modell, **Fehler! Verweisquelle konnte nicht gefunden werden..**

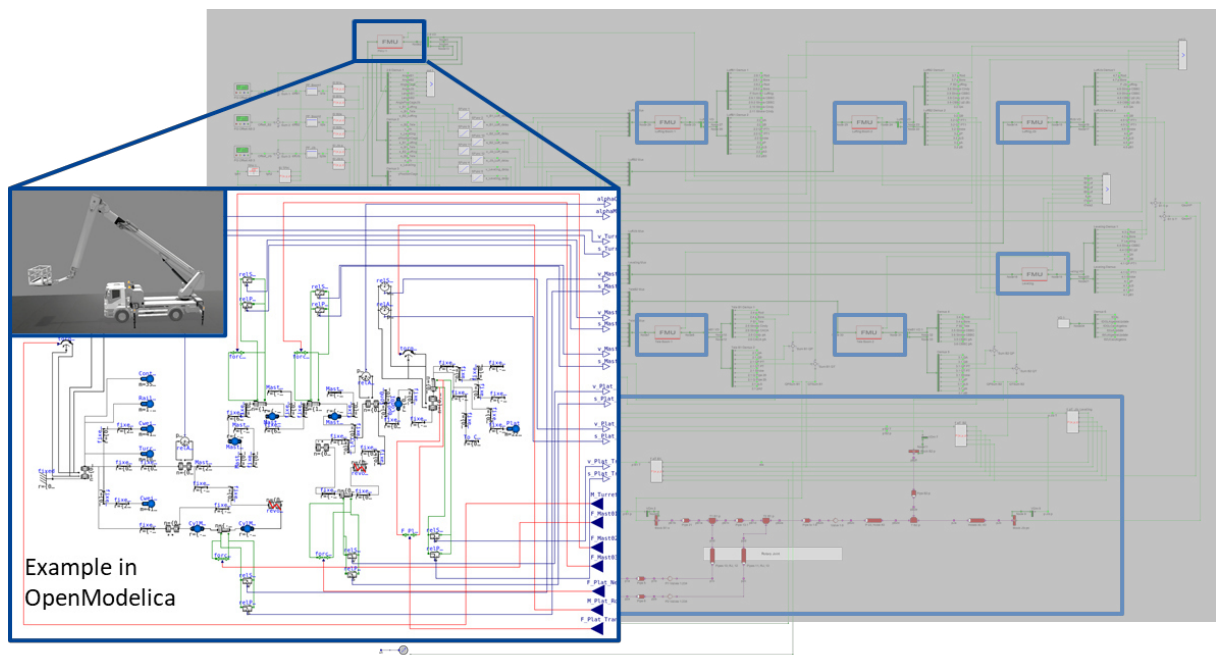


Bild 5: Composite-Modell der Hubarbeitsbühne

In Bild 5 ist die FMU des mechanischen Submodells markiert und das darin gekapselte Mechanikmodell dargestellt. Hinter den weiteren mit FMU bezeichneten Bausteinen, die im Bild erkennbar sind, verbirgt sich jeweils eine hydraulische Achse. Die Druckversorgung ist hier Teil des Gesamtmodells, könnte aber ebenso in eine weitere FMU gekapselt werden. Die Verschaltung der FMUs erfolgt in DSHplus, das in der Lage ist, die Berechnung der FMUs parallel auszuführen. **Bild 6** zeigt, wie durch das modulare, FMU-basierte Modellkonzept und die Parallelisierung erheblich Rechenzeit eingespart wird. Die

Simulation des Gesamtmodells der im Beispiel gezeigten Hubarbeitsbühne ist jetzt in Echtzeit möglich.

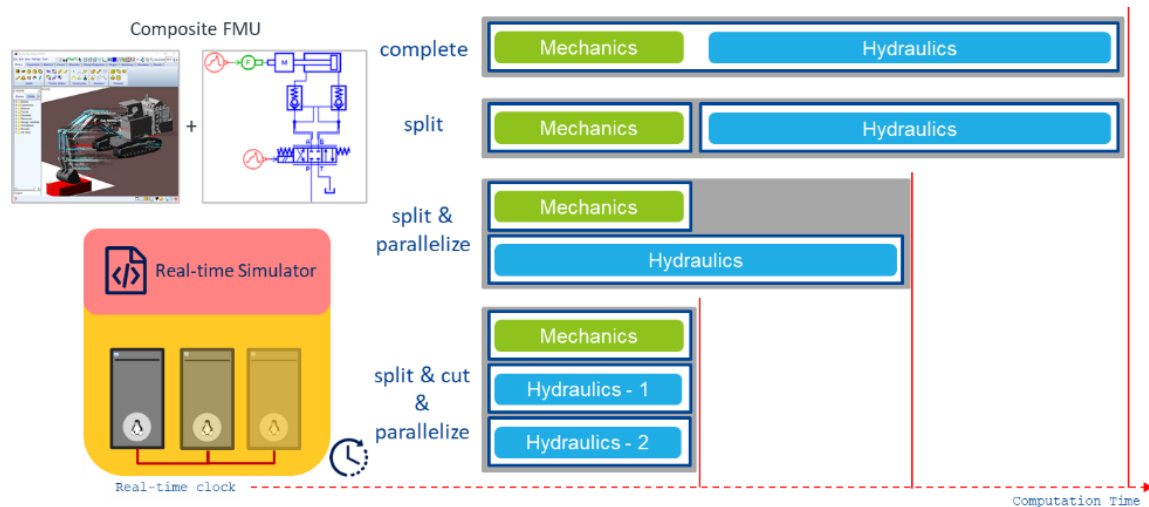


Bild 6: Echtzeitoptimierung durch parallele Berechnung des modularen Modells

Neben der Möglichkeit der parallelen Berechnung bietet der FMU-basierte Modellierungsansatz noch weitere wichtige Vorteile:

- Mit FMI 2.0 wird eine standardisierte Schnittstelle verwendet. Daher können alle FMUs, die diesen Standard erfüllen, in das Gesamtmodell eingebunden werden, unabhängig davon, mit welchem Tool sie erstellt wurden. Ein Beispiel ist das Mechanik-Submodell der Hubarbeitsbühne, das hier in OpenModelica erstellt und im DSHplus Composite Model mit den hydraulischen FMUs verschaltet wird. So kann jede Domäne in ihrem Tool der Wahl arbeiten.
- Nicht nur die Berechnung kann parallelisiert werden: Die modulare Struktur mit einer Aufteilung in funktionale Einheiten erlaubt es, die Submodelle simultan zu entwickeln und in Betrieb zu nehmen.
- Module können in anderen Maschinen oder Varianten wiederverwendet werden. Damit entsteht ein Baukasten, aus dem z.B. eine Neuentwicklung sich bedient: Der Entwicklungsaufwand wird reduziert, „unnötige“ Varianten werden vermieden und vorhandenes Knowhow systematischer genutzt.

3.2 Hardware-in-the-Loop

Das Composite Model wird wiederum als FMU exportiert und auf die Echtzeitumgebung des FLUIDON | **Cube** geladen. Das System stellt die IOs des Modells über EtherCAT zur Verfügung. An den EtherCAT angeschlossene analoge oder digitale Ein- und Ausgangsklemmen oder Feldbusklemmen, z.B. CAN, ermöglichen in einem Hardware-in-the-Loop (HiL) Setup eine sehr schnelle und flexible Kopplung mit der Steuerungshardware (**Bild 7**).



Bild 7: Echtzeit-HiL-Setup des FLUIDON | Cube

Wie in der realen Maschine sind die Joysticks über CAN mit der Steuerungshardware verbunden. Eine Vielzahl von Sensoren liefert weitere Eingangsgrößen für die Steuerung, wie z.B. die relativen Winkel der Ausleger, die teleskopierten Längen oder die Abweichung der Plattform von der horizontalen Lage. Innerhalb der HiL-Umgebung stellt das Simulationsmodell diese Größen zur Verfügung. Im Gegensatz zur realen Maschine werden diese Größen jedoch nicht über CAN ausgegeben und an die Maschinensteuerung übertragen: Um den Aufbau zu vereinfachen, erfolgt die Kopplung zwischen Simulationsumgebung und Steuerungshardware aktuell über OPC-UA. Die Kommunikation über diese Schnittstelle garantiert zwar keine harte Echtzeit, ist aber im lokalen Netzwerk ausreichend performant, um eine Datenübertragung in der Zykluszeit der Maschinensteuerung und schneller zu gewährleisten. Gerade zu einem frühen Zeitpunkt in der Entwicklung kann eine solche Kopplung sinnvoll sein, wenn die Schnittstellen zur Sensorik oder Aktuatorik noch nicht endgültig festgelegt sind. Im nächsten geplanten Entwicklungsschritt wird diese Datenübertragung – wie in der realen Maschine – über CAN erfolgen, sodass die Steuerung 1:1 in die reale Maschine eingebaut werden kann. Alternativ kann ebenfalls ein Feldbussystem basierend auf EtherNet/IP eingesetzt werden.

4 Bahnkurvensteuerung

Nachdem nun die Werkzeuge für die virtuelle Steuerungsentwicklung und -inbetriebnahme vorliegen, erfolgt die Entwicklung der prototypischen Steuerung der Hubarbeitsbühne mit Hilfe der HiL-Simulation.

Bei der heute meist üblichen Steuerung der Hubarbeitsbühne steuert der Benutzer die einzelnen Achsen separat. Dies erfolgt typischerweise über Joysticks. Bei einer Arbeitsbühne mit mehrteiligem Ausleger können allein aufgrund der begrenzten Anzahl an unabhängigen Achsen eines Joysticks nicht alle Achsen gleichzeitig betätigt werden. Koordinierte gleichzeitige Bewegungen der Achsen sind damit nur begrenzt möglich.

Ein kontinuierliches und ruckbegrenztetes Verfahren der Plattform entlang einer bestimmten Trajektorie ist daher für einen menschlichen Bediener nur schwer bis gar nicht möglich. Die aktuelle Entwicklung einer prototypischen neuen Ansteuerung hat daher zum Ziel, dem Anwender eine einfach zu bedienende Arbeitsplattform zur Verfügung zu stellen, die eine Richtungs- oder Bahnvorgabe des Bedieners in eine Kombination aus Bewegungsvorgaben für die Achsen umsetzt. Die Bewegung entlang einer festen Trajektorie kann man

sich zum Beispiel als geradlinige Bewegung der Hebebühne entlang einer Fensterfront vorstellen.

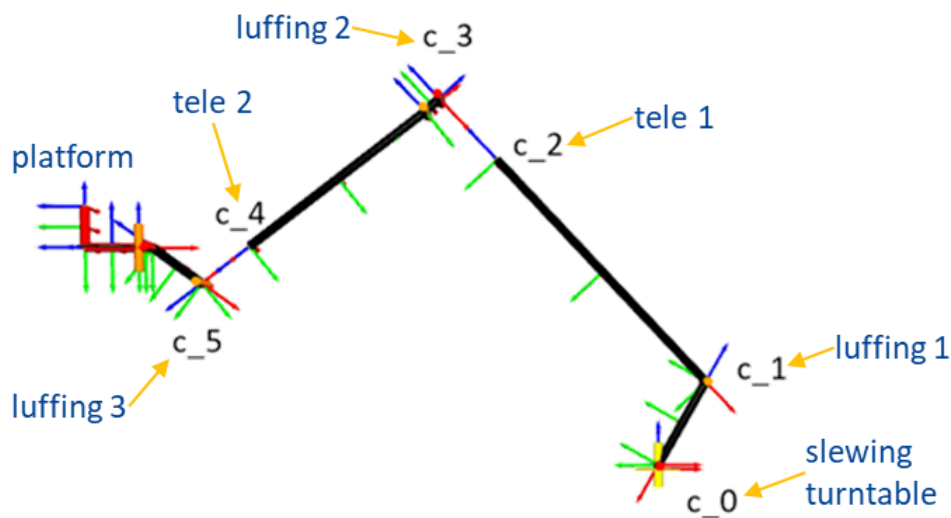


Bild 9: Kinematikmodell zur Bahnkurvenberechnung

Bild 9 zeigt die schematische Darstellung des Kinematikmodells, das der Ausgangspunkt für die Bahnkurvenberechnung ist. Die Parameter c_i stellen die Rotation bzw. Translation der Achsen dar. Ein Parametersatz C enthält diese Parameter c_i und beschreibt somit eine Position der Hubarbeitsbühne. Bei bekannten Parametersätzen C ist es eine einfache Operation, die daraus resultierende Position der Hebebühne zu berechnen (Vorwärts-Rechnung). Für die gewünschte Applikation ist jedoch der umgekehrte Weg nötig: Damit ein Bediener eine Hebebühne im Raum bewegen kann, muss für jede Position der Hebebühne ein Parametersatz C bestimmt werden, der für jede Achse einen Parameter c_i enthält. Hierbei handelt es sich um ein inverses Problem [5].

Zur Lösung dieses Problems wurde für erste Realisierbarkeitstests das Downhill-Simplex-Verfahren von John Melder und Roger Mead eingesetzt. [6]. Dieses lokale Optimierungsverfahren, das keine Ableitungsinformationen benötigt, berechnet iterativ für jeden Punkt $P_i = (x_i, y_i, z_i)$ der Sollgeometrie den für diese Position benötigten aktuellen Parametersatz c_i , wobei die Konfiguration für den vorherigen Punkt $P_{(i-1)}$ als Startwert der Optimierung verwendet wird. Mit Hinblick auf die benötigte Echtzeitberechnung wird man in Zukunft wahrscheinlich auf „einfachere“, in Hardware besser implementierbare Algorithmen wie zum Beispiel eine Newton-Minimierung [7], zurückgreifen.

Im folgenden Beispiel wird gezeigt, wie sich für eine Bahnvorgabe der zeitliche Verlauf von C ergibt, der dann als Sollwert auf die Achsregler gegeben wird.

Bild 10 zeigt eine Visualisierung des Kinematik-Modells für die Hebebühne. Die beiden linken Grafiken zeigen die Ausgangsposition, die beiden rechten Grafiken die Endposition einer Bewegung entlang einer Geraden im Raum.

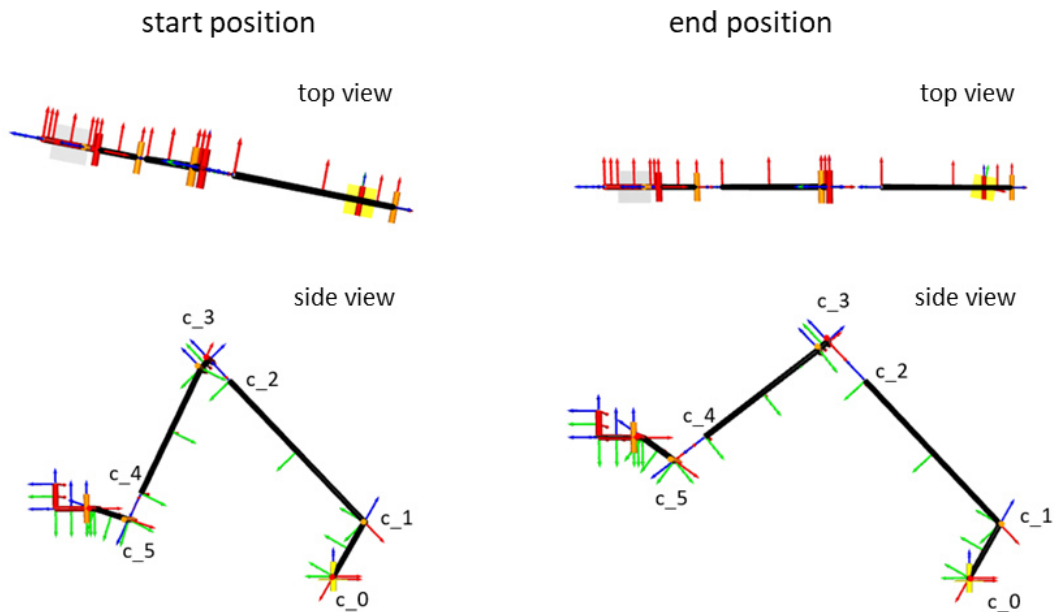


Bild 10: Start- und Endposition der Trajektorie

Es ist zu erkennen, dass alle Parameter c_i gleichzeitig angepasst werden müssen, um die Bewegung der Hebebühne entlang der Geraden zu realisieren. Es wird nun aus diesen Positionen ein Parametersatz C errechnet, sodass eine optimierte Bewegung erfolgen kann. Optimierungen lassen sich durch diverse Kriterien festlegen wie z. B.:

- Wegoptimierung
- Optimierung hydraulisches System
- Ruckbegrenzte Bewegung
- Vermeiden von kritischen Geschwindigkeitsbereichen mit Stick-Slip

Möchte man die Hebebühne vom Anfangs- zum Endpunkt der Trajektorie geradlinig verfahren, liefert eine Optimierung die Parameter c_i wie in **Bild 11** dargestellt, welche diese Bewegung für jede Achse vorgeben.

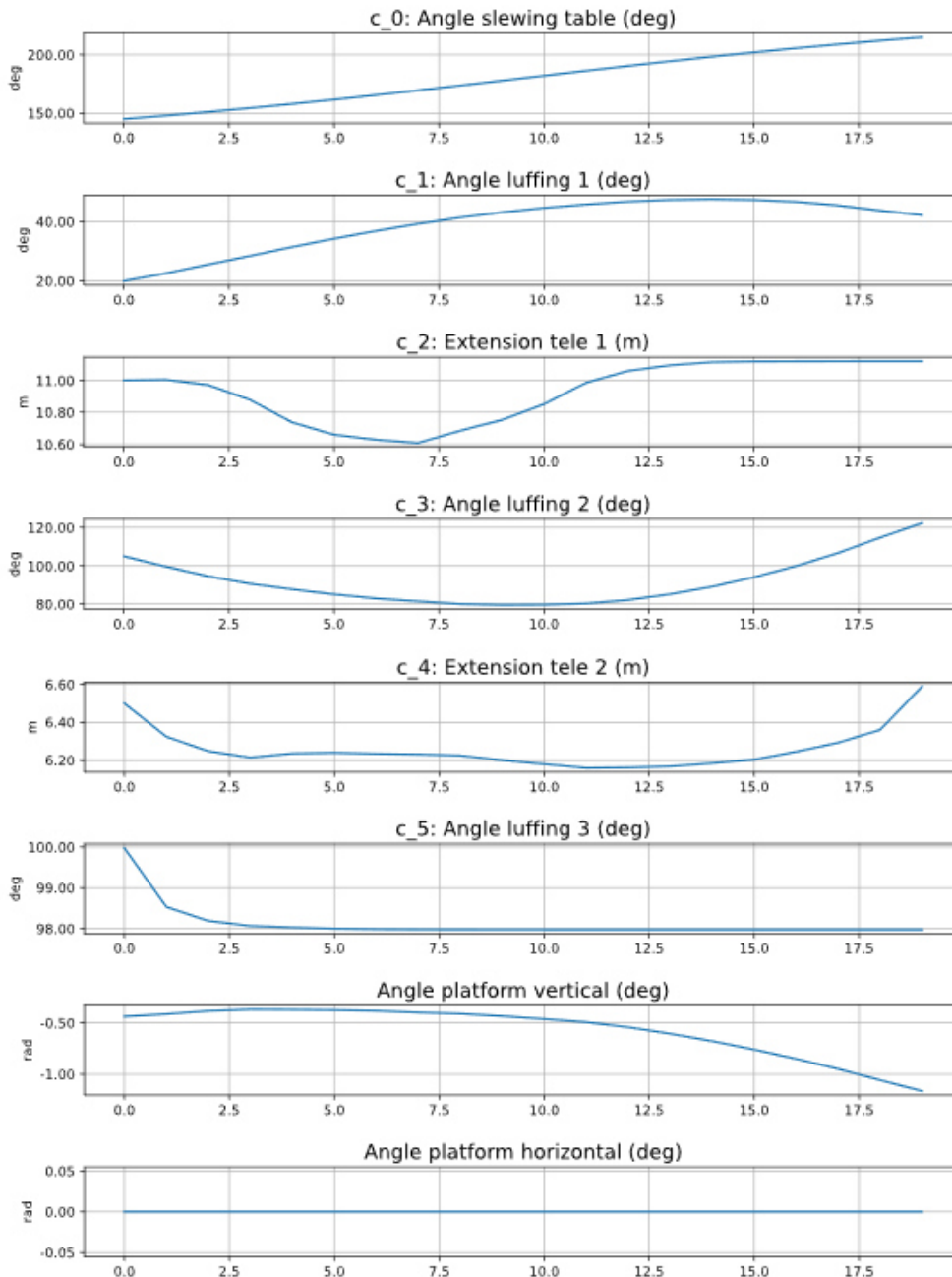


Bild 11: Zeitlicher Verlauf der Parameter c_i für die gewünschte lineare Bahn

Die Positions- und die daraus berechneten Geschwindigkeitsvorgaben sind die Sollwerte für die Achsregler, die aus diesen Vorgaben und den aktuell an der Maschine gemessenen Positionen (bzw. Winkeln) die Stellsignale der Ventile generieren.

Die Bedienung der Hebebühne kann nun stark vereinfacht werden. Verwendet man klassische Joysticks mit zwei Achsen, so kann z. B. der linke Joystick die x- und y-Richtung im Raum vorgeben, während der rechte Joystick die Bewegung in die vertikale z-Richtung und die Drehbewegung der Plattform steuert. Mit einem 3D-Joystick ist auch eine Einhandbedienung möglich.

Die Entwicklung und Inbetriebnahme der Bahnkurvensteuerung kann jetzt mit Hilfe des Cube-RT-Setups zu großen Teilen „am Schreibtisch“ erfolgen – mit im Vergleich zur

Entwicklung am realen Prototyp deutlich reduziertem Aufwand. Neben der Zeit- und Kosteneinsparung ist ein weiterer Vorteil der virtuellen Tests (**Bild 13**), dass reproduzierbare Testabläufe leichter realisiert werden können, sodass z. B. die Regelgüte bei unterschiedlichen Parametern besser verglichen werden kann.

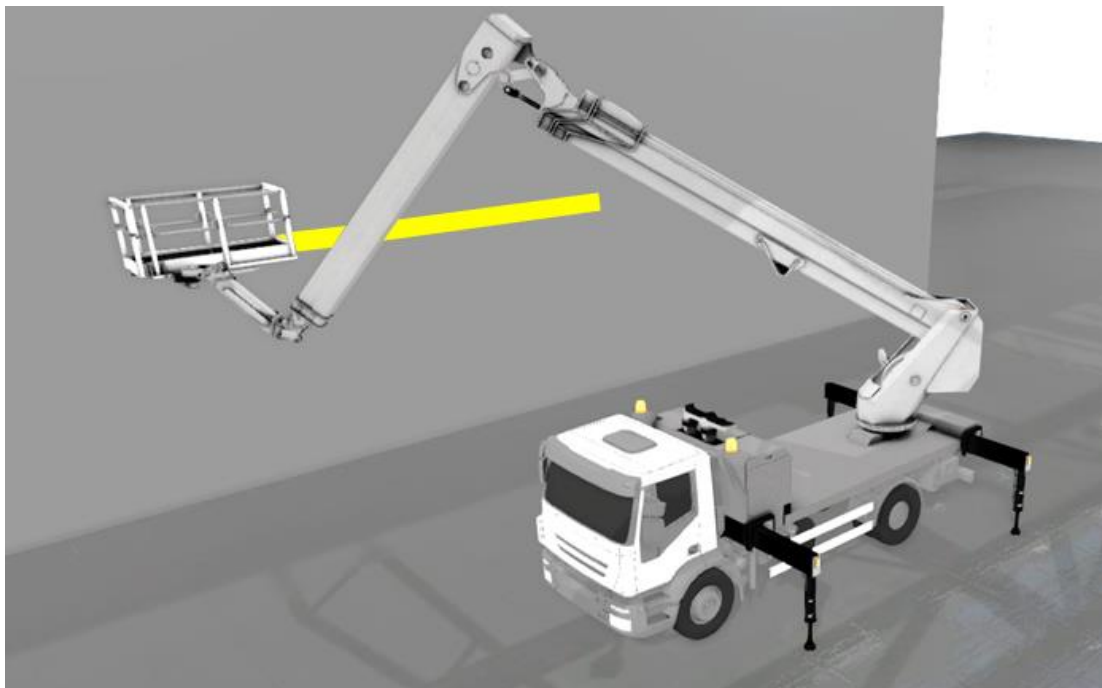


Bild 13: Visualisierung der Trajektorie im virtuellen Test

6 Zusammenfassung

Am Beispiel einer Hubarbeitsbühne wurde gezeigt, welche Vorteile die modellbasierte Entwicklung und die virtuelle Inbetriebnahme insbesondere für die Elektronik- und Softwareingenieure bietet. Während im klassischen Prozess die Steuerungsentwicklung und der Test oft nur am realen Prototyp möglich sind, erfolgen Steuerungsentwicklung und -tests nun im Zusammenspiel mit dem virtuellen Prototyp. Nach dem Aufbau des echtzeitfähigen Simulationsmodells findet die Validierung direkt auf der realen SPS/ECU in der Hardware-in-the-Loop-Umgebung (HiL) statt. Die Mensch-Maschine-Schnittstelle (MMS, engl. Human-Machine-Interface HMI), bestehend aus Komponenten wie Joysticks, Bedienfeldern und Displays, kann bereits Teil dieses Testaufbaus sein und ermöglicht nicht nur einen Test aus Engineering-Sicht, sondern auch Anwender- oder Kundentests und Feedback auf Basis des virtuellen Prototyps. Extremsituationen der Maschine können in dieser Phase ohne Risiko getestet werden. Die frühe Steuerungsintegration deckt Designprobleme viel früher auf als der traditionelle Prozess, ermöglicht somit kostengünstigere Korrekturmaßnahmen als in einem späteren Stadium und führt zu einer besseren Qualität des Produktdesigns. Die Zeit bis zur Markteinführung eines neuen Systems wird hierdurch deutlich verkürzt.

Quellenverzeichnis:

- [1] N. N., Digital twin – Wikipedia, https://en.wikipedia.org/wiki/Digital_twin, visited on March 15, 2022.
- [2] Völker, L., Untersuchung des Kommunikationsintervalls bei der gekoppelten Simulation, PhD Thesis Karlsruher Institut für Technologie (KIT), 2011.
- [3] de Wit, C. C., Olsson, H., Aström, K.J., Lischinski, P., A New Model for Control of Systems with Friction, In: IEEE Transactions on Automation Control, Vol. 40, 1995
- [4] N. N., Functional Mockup Interface, <https://fmi-standard.org/>, visited March 15, 2022.
- [5] Tarantola. A., Inverse Problem Theory and Methods for Model Parameter Estimation, Society for Industrial and Applied Mathematics, 2004.
- [6] Nelder, J. A., Mead, R., A Simplex Method for Function Minimization, The Computer Journal, Vol. 7, 308-313, 1965
- [7] Polyak, B. T., Newton's method and its use in optimization, European Journal of Operations Research, 181(3), 1086-1096, 2007