

# 119

Karlsruher Schriftenreihe  
Fahrzeugsystemtechnik

## 13. Kolloquium Mobilhydraulik

8./9. Oktober 2024, Karlsruhe

# MHK.13

Kolloquium Mobilhydraulik  
Karlsruhe

# Simulation im Lebenszyklus der mobilen Arbeitsmaschine

*Benedikt Müller<sup>1</sup>, Heiko Baum<sup>2</sup>*

<sup>1</sup> FLUIDON Gesellschaft für Fluidtechnik mbH, Aachen, Deutschland  
E-Mail: benedikt.mueller@fluidon.com; Tel.: +49 (0)241 980 935 63

<sup>2</sup> FLUIDON Gesellschaft für Fluidtechnik mbH, Aachen, Deutschland  
E-Mail: heiko.baum@fluidon.com; Tel.: +49 (0)241 980 935 61

## Kurzfassung

Der Beitrag stellt eine Simulationslösung vor, die den gesamten Lebenszyklus von mobilen Arbeitsmaschinen umfasst, von der Konzeption und Entwicklung bis hin zu Inbetriebnahme, Anwenderschulung, Condition Monitoring und Predictive Maintenance. Die Anwendung wird anhand von zwei Baumaschinen demonstriert: Mit Hilfe der virtuellen Entwicklungsumgebung Cube hat Fluidon gemeinsam mit Meastream den Prototypen einer Bahnkurvensteuerung für komplexe mehrachsige Maschinenbewegungen entwickelt, der hier für eine Hubarbeitsbühne eingesetzt wird. An einem Minibagger zeigen wir, wie die echtzeitfähige Maschinensimulation während der Einsatzphase Anwendung findet: Das Modell liefert Daten für virtuelle Sensoren, die Condition Monitoring und Predictive Maintenance unterstützen.

**Schlagworte:** Echtzeitsimulation, Hardware-in-the-Loop, Functional Mockup Interface, Bahnkurvensteuerung, Softsensor, digitaler Zwilling

# 1 Einleitung

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.

Schon beim Entwurf einer solchen Maschine gibt es viele Fragen: Die Bewegung der Maschine hängt von ihrer Kinematik ab. Aber auch die Anlenkpunkte der Zylinder, die Überlappung der Teleskoprohre, die Reibung und die Massenkräfte haben einen großen Einfluss auf die Anforderungen an das Hydrauliksystem und die Steuerung. Diese Schnittstellen zwischen den Domänen Mechanik und Hydraulik 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 hierfür 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 (**Abbildung 1.1**).

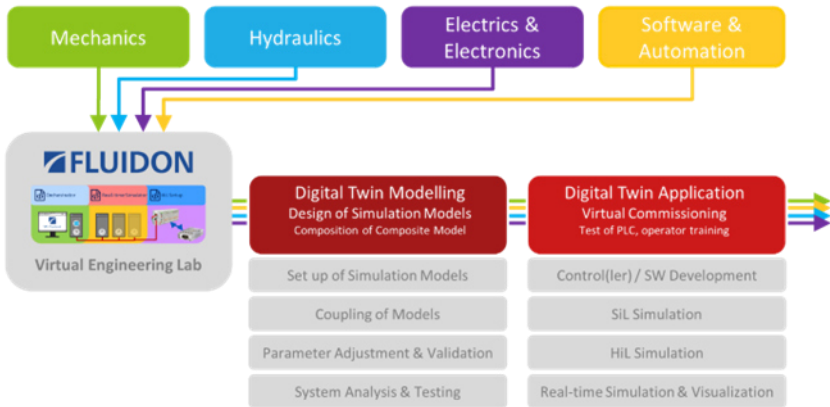


Abbildung 1.1: Domänenübergreifende, modellbasierte Entwicklung

Am Beispiel einer Hubarbeitsbühne wird im Folgenden gezeigt, wie das domänenübergreifende Modell modular und basierend auf standardisierten Schnittstellen aufgebaut ist. Echtzeitfähige Modelle erleichtern die Steuerungsentwicklung mittels Software- oder Hardware-in-the-Loop und sind Voraussetzung für eine realitätsnahe 3D-Visualisierung.

Doch auch über die Entwicklung hinaus begleiten die Modelle den Lebenszyklus der realen Maschine (**Abbildung 1.2**): Virtuelle Sensoren, die anhand eines Baggers erläutert werden, liefern Daten für Condition Monitoring, vorbeugende Wartung oder – wie hier näher gezeigt – erzeugen Informationen für die Baudatenmodellierung.

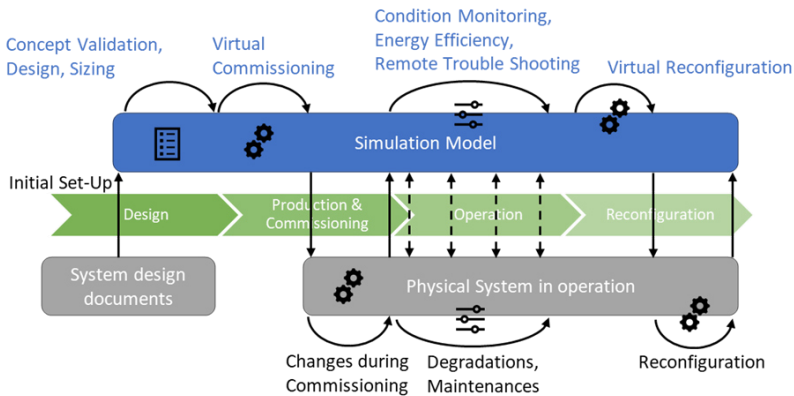


Abbildung 1.2: Simulation als Teil des digitalen Zwillings begleitet den Lebenszyklus des Produkts

## 2 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 (**Abbildung 2.1**). 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.

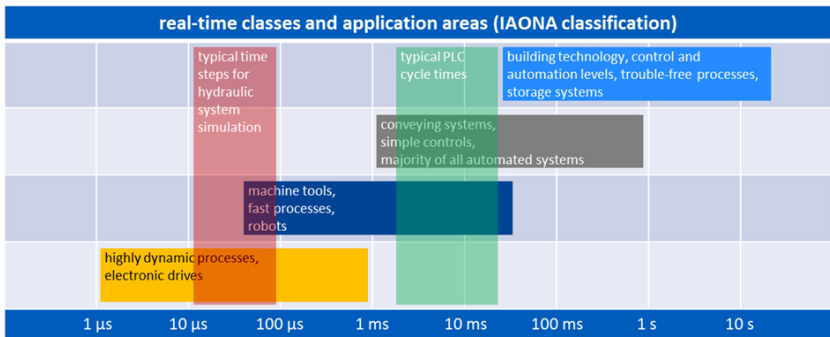


Abbildung 2.1: Echtzeitanforderungen in der Hydrauliksimulation vs. IAONA-Klassifizierung

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 aufgefressen 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.

## 2.1 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.

## 2.2 Submodelle Hydraulik

**Abbildung 2.2** 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]

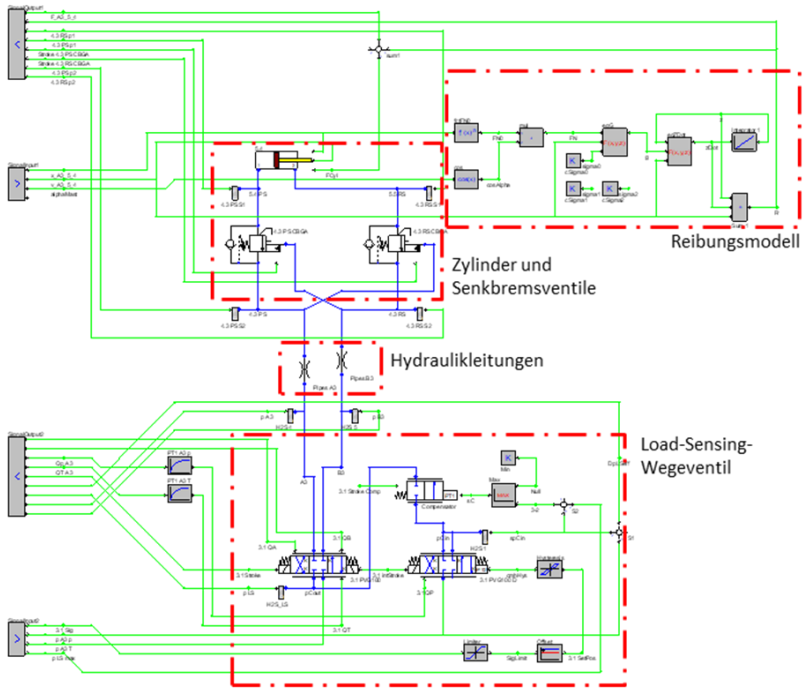


Abbildung 2.2: Submodell einer hydraulischen Achse mit Reibungsmodell

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 deshalb 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.

## 2.3 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, **Abbildung 2.3**.

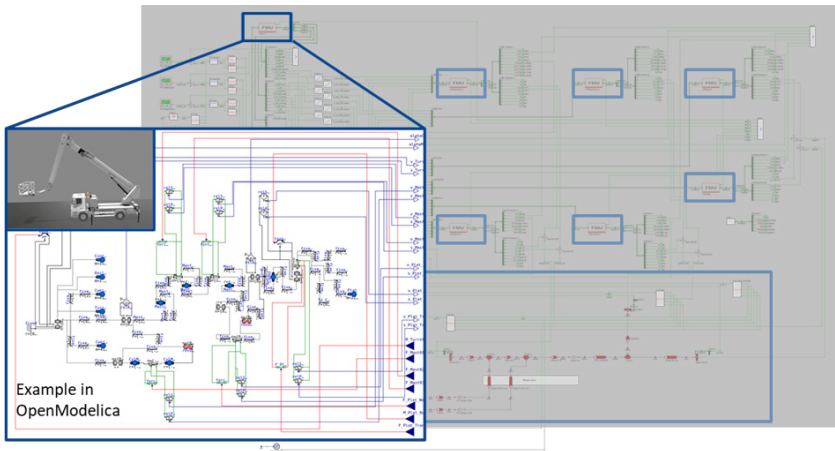


Abbildung 2.3: Composite-Modell der Hubarbeitsbühne

In Abbildung 2.3 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. **Abbildung 2.4** 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.

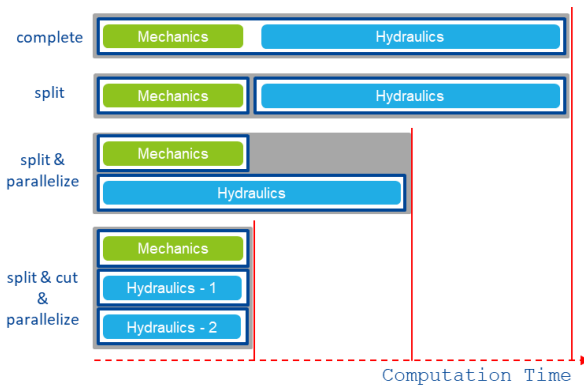


Abbildung 2.4: Echtzeitoptimierung durch parallele Berechnung des modularen Modells

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.
- Durch den modularen Aufbau lassen sich leichter unterschiedliche Konzepte untersuchen oder Varianten vergleichen: Das Submodell einer Antriebsachse kann z. B. einen hydraulischen oder elektrischen Antrieb darstellen, die sich, bei gleicher mechanischer Schnittstelle, leicht austauschen und simulativ miteinander vergleichen lassen.

- 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.

Zukünftig wird der Verwaltung von Daten, die den Digitalen Zwilling ausmachen, eine Schlüsselrolle zukommen. Dies betrifft u. A. auch die Simulations(teil)modelle. Im Forschungsprojekt Fluid 4.0 erarbeitet ein Konsortium dazu branchenübergreifende Lösungen. Fluidon konzipieren darin die automatische Übernahme von simulationsrelevanten Komponentenparameter aus den Komponentenverwaltungsschalen, um das Gesamtsystemsimulationsmodell schneller und vor allem fehlerfrei zu bedaten (**Abbildung 2.5**). [5]

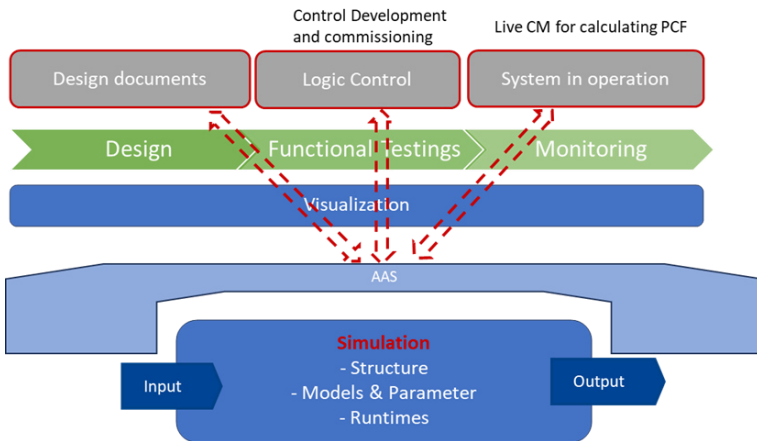


Abbildung 2.5: Modellverwaltung mit der Verwaltungsschale (Asset Administration Shell, kurz AAS)

### 3 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 (**Abbildung 3.1**).



Abbildung 3.1: Fluidon Cube Echtzeit-HiL-Setup

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 ruckbegrenztes Verfahren der Plattform entlang einer bestimmten Trajektorie ist daher für einen menschlichen Bediener nur schwer bis gar nicht möglich. Die neue prototypische Ansteuerung, gemeinsam von Fluidon und Meastream entwickelt, 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.

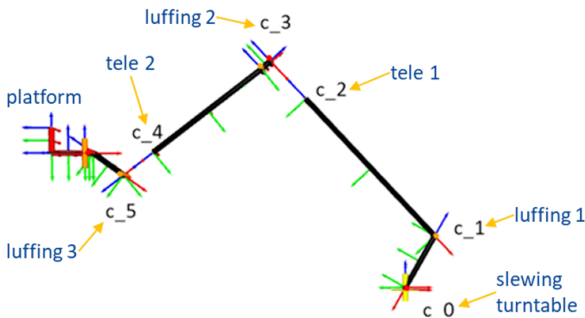
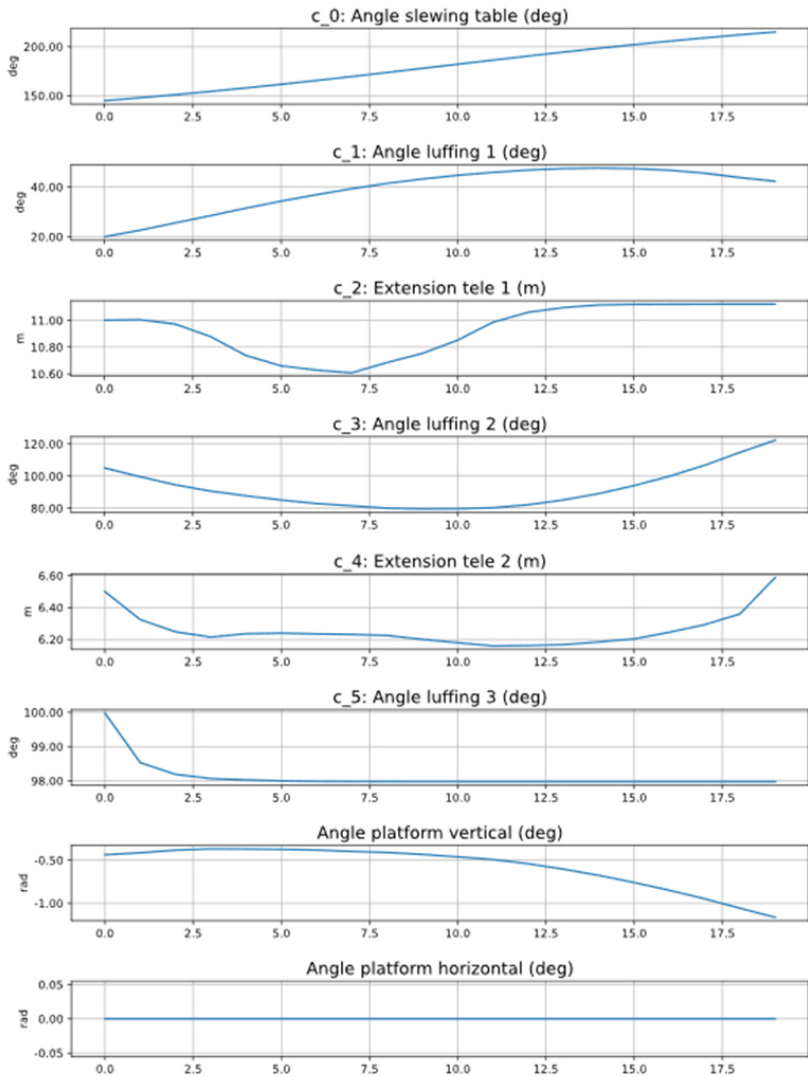


Abbildung 4.1: Kinematikmodell zur Bahnkurvenberechnung

**Abbildung 4.1** 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. Da das kinematische System unbestimmt ist, ergeben sich für eine Position der Arbeitsbühne unendlich viele Lösungen. Ein mathematisches Optimierungsverfahren bestimmt für jede Position der Hebebühne einen Parametersatz  $C$ , der für jede Achse einen Parameter  $c_i$  enthält, nach Optimierungskriterien 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 **Abbildung 4.2** dargestellt, welche diese Bewegung für jede Achse vorgeben.

Abbildung 4.2: 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.

Entwicklung und Inbetriebnahme der Bahnkurvensteuerung können 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 (**Abbildung 4.3**), dass reproduzierbare Testabläufe leichter realisiert werden können, sodass z. B. die Regelgüte bei unterschiedlichen Parametern besser verglichen werden kann.

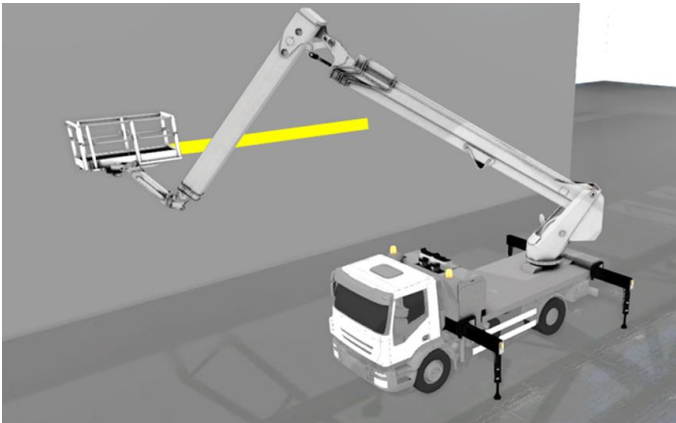


Abbildung 4.3: 3D-Visualisierung der Trajektorie im virtuellen Test

## 5 Virtuelle Sensoren

Das vom Bundesministerium für Digitales und Verkehr geförderte Forschungsprojekt Off-Highway-Twins 2 hat zum Ziel, einen digitalen Zwilling der Baustelle und ihrer Umgebung zur Verfügung zu stellen: “Fusion von Maschinen- und Geodaten in Digitalen Zwillingen von Infrastrukturobjekten und deren Umgebung für die Erfassung, Planung, Bau, Betrieb und Instandhaltung” [6].

Die von den vernetzten Baumaschinen gesammelten Daten werden in der Cloud mit Geodaten zusammengeführt und im Building Information Modeling (BIM) abgebildet (**Abbildung 5.1**).

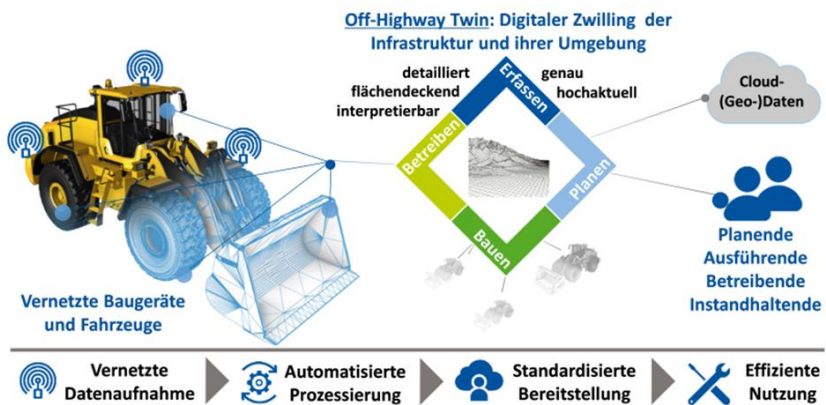


Abbildung 5.1: Off-Highway Twins: 1-zu-1-Abbilder der realen Welt in Echtzeit

Da nicht alle erforderlichen Maschinendaten durch Sensoren gemessen werden können, oder entsprechende Sensorik teuer oder zu empfindlich für den Baustelleneinsatz wäre, sieht das Konzept den Einsatz sogenannter virtueller Sensoren vor, auch als Softsensoren bezeichnet. Fluidon kommt im

Rahmen des Projekts die Aufgabe zu, diese Sensoren und die dafür notwendige Infrastruktur zu entwickeln.

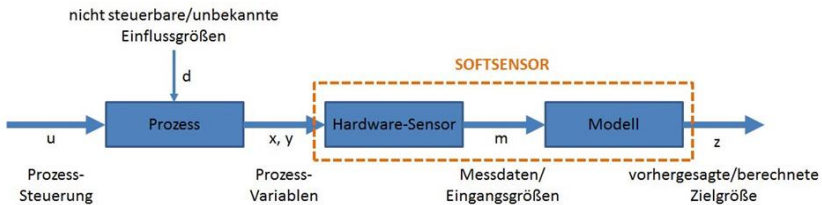


Abbildung 5.2: Prinzip des Softsensors [7]

**Abbildung 5.2** zeigt das Prinzip des Softsensors: “Der Softsensor ermittelt mithilfe von Hardware-Sensoren korrelierende Prozessgrößen ( $x$ ,  $y$ ) und übermittelt diese als Eingangsgroessen ( $m$ ) in das hinterlegte Modell, welches in Form eines Auswertalgorithmus die Zielgroesse berechnet. Der zu überwachende Prozess wird dabei durch steuerbare ( $u$ ) und nicht steuerbare ( $d$ ) Einflussgrößen beeinflusst.” [7] Ein Softsensor ist also ein Modell, das mit Hilfe von Messgrößen realer Sensoren als Eingangsgroessen neue “virtuelle Messgrößen” ermittelt. An dieser Stelle kommen wiederum die oben beschriebenen Simulationsmodelle der Maschine zum Einsatz. Die erforderliche Simulationsumgebung läuft je nach Anforderung auf der Maschine selbst und / oder als Cloud-Applikation.

Zunächst jedoch zur Kommunikations-Infrastruktur: Gemeinsam mit den Projektpartnern (vgl. [6]) wurde die Infrastruktur zum Datenaustausch geplant und umgesetzt. In **Abbildung 5.3** wird der prinzipielle Aufbau gezeigt. Die Datenübertragung zwischen den Teilnehmern erfolgt per MQTT. Die Daten der realen Sensoren auf der Maschine, aber auch weitere Steuerungsdaten, die z. B. auf dem Maschinen-CAN verfügbar sind, können mit einer Datenrate von bis zu 1 kHz zur Verfügung gestellt werden. Um Softsensoren direkt auf der Maschine umzusetzen, hat Fluidon den Cube Edge Simulator entwickelt. Dieses Echtzeit-Simulationssystem läuft auf

einem robusten Industrie-PC auf der Maschine. Über seine MQTT-Schnittstelle empfängt die Simulation die realen Sensordaten oder Inputs aus der Maschinensteuerung und stellt hier andererseits seine Outputs, die "Messgrößen" der Softsensoren, zur weiteren Verarbeitung zur Verfügung.

Über eine Mobilfunkverbindung werden die MQTT-Daten mit den Cloud-basierten Applikationen ausgetauscht. Fluidon entwickelt z. Zt. den Cube Cloud Simulator, der eine direkte Anbindung an den MQTT-Datenaustausch ermöglicht. Auf diesem Cloud-Simulator können Softsensoren oder Auswertungen umgesetzt werden, die größere Rechenleistung erfordern. Rechenaufwändige Algorithmen, z. B. zur Parameteroptimierung von Simulationsmodellen, lassen sich hier durch massiv parallelisierte Berechnungen umsetzen.

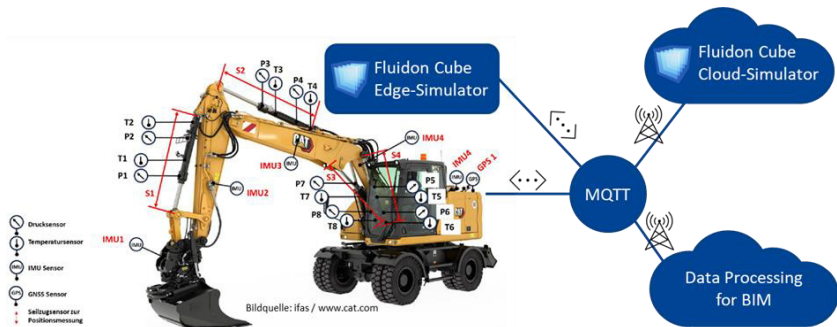


Abbildung 5.3: Infrastruktur Off-Highway-Twins 2

Die Grundanforderungen an das Modell der Maschine, das auf dem Edge-Simulator eingesetzt wird, sind ähnlich denen an ein Modell für die HiL-Simulation: Das Modell muss die Maschinendynamik ausreichend genau abbilden und in Echtzeit laufen. Darüber hinaus wird es hier notwendig, den Modellzustand ständig mit dem Zustand der realen Maschine abzugleichen. Dies ist erforderlich, da zum einen das Modell durch Vereinfachungen und kleinen Parameterabweichungen nie zu 100% das reale Verhalten abbilden

kann und u. U. auch nicht erfasste bzw. modellierte Einflussgrößen auf das Modell wirken und ohne zusätzliche Maßnahmen zu einem Auseinanderdriften der Zustände führen. Hinzu kommen Abweichungen durch Messfehler oder stochastische Einflüsse. Zum anderen ist es erforderlich, das Modell in einen Startzustand zu bringen, der dem der realen Maschine entspricht.

Fluidon setzt zu diesem Zweck einen Synchronisationsregler ein in Anlehnung an [8], dessen Prinzip in der folgenden **Abbildung 5.4** dargestellt ist.

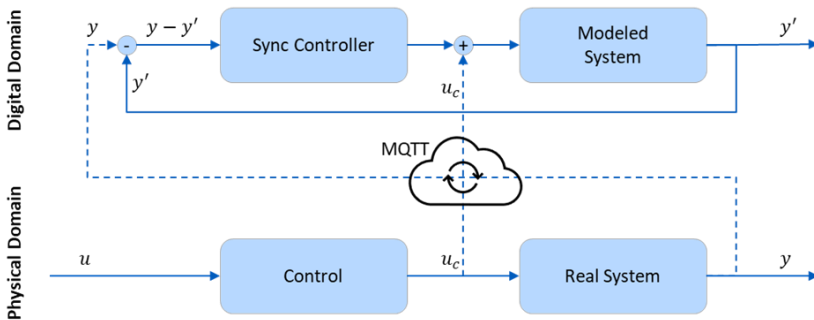


Abbildung 5.4: Prinzip der Modellsynchronisation

Der Synchronisationsregler hat zur Aufgabe, das Modell so nachzuregeln, dass seine Ausgabegröße  $y'$  der Ausgabegröße des realen Systems  $y$  möglichst nahe kommt. Er erhält daher diese Größe  $y$  als Sollwertvorgabe, die mit dem Istwert  $y'$  verglichen wird. Zugleich wird das Steuersignal  $u_c$  der realen Steuerung nach einem Feed-Forward-Prinzip dem Systemmodell aufgeschaltet, so dass der Regler nur noch die Abweichungen des Systemverhaltens ausgleichen muss. Am Beispiel des Bagger-Auslegers wird im Folgenden gezeigt, wie dies für eine hydraulische Achse umgesetzt wurde (**Abbildung 5.5**).

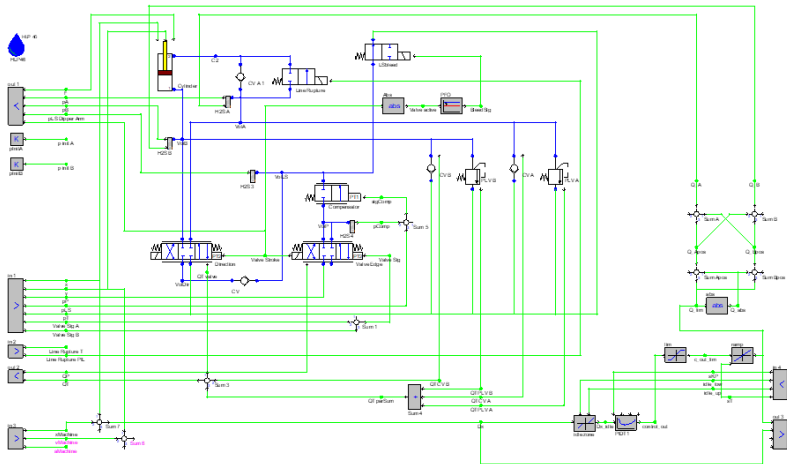


Abbildung 5.5: Modell der Achse mit Synchronisationsregler

Der Regler hat die Aufgabe, die Zylinderposition des Modells der realen Zylinderposition nachzuführen. Seine Stellgrößen sind ein A- und B-seitiger Volumenstrom, die unmittelbar dem Zylinder aufgebracht werden. Ein klassischer Positionsregler hätte typischerweise ein Ventilstellsignal als Ausgang. Das ist für den Synchronisationsregler jedoch kein sinnvoller Ansatz: Wenn z. B. das Ventil bereits bei 100% Stellweg ist, die reale Zylinderdrehgeschwindigkeit aber trotzdem größer, bliebe keine Marge mehr zum Nachregeln. Diese Situation kann z. B. dann eintreten, wenn der im Modell parametrisierte maximale Ventilquerschnitt etwas zu klein ist.

Der Regler wurde im gezeigten Beispiel für erste Tests zunächst recht einfach gehalten: Kern ist ein PID1-Regler, für dessen Eingang optional eine Totzone eingestellt werden kann. Das Ausgangssignal wird in den weiteren Blöcken in der Höhe und dem Gradienten begrenzt. Die anschließende Volumenstromberechnung berücksichtigt neben den Vorzeichen das Flächenverhältnis des Zylinders und generiert die Stellwerte  $Q_A$  und  $Q_B$ .

Auch hier wird zukünftig in der Modellierung ein modulares Prinzip verfolgt: Der Synchronisationsregler wird in eine eigene FMU "verpackt" und kann so standardisiert und – mit ggf. angepassten Parametern – auch für weitere Achsen eingesetzt werden. Das Modell der Achse bleibt nahezu unverändert: Lediglich die neuen Eingänge für die Stellgrößen müssen ergänzt werden. Werden diese nicht verwendet, so verhält sich das Modell wie bisher und kann z. B. unverändert für HiL-Simulationen eingesetzt werden, so dass nur ein Modellstand erforderlich ist. In **Abbildung 5.6** sind erste Testergebnisse aus der Inbetriebnahme des Reglers dargestellt.

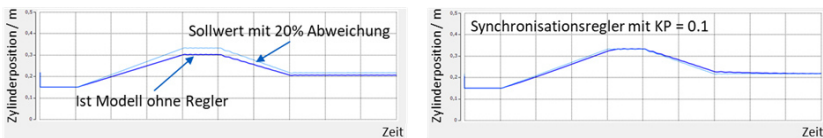


Abbildung 5.6: Testergebnisse Modellnachführung

Im Testszenario führt der Ausleger eine Auf- und Abwärtsbewegung bei jeweils konstantem Ventilsignal durch. Dabei ist der Volumenstrom im Modell geringer, so dass ohne Regler eine 20%ige Abweichung der Position entsteht. Bereits ein einfacher P-Regler ist in der Lage, das System nachzuführen und diese Abweichung auszugleichen, wie rechts im Bild gezeigt.

Im nächsten Testszenario, **Abbildung 5.7**, wird überprüft, wie gut der Synchronisationsregler das Modell in die Startposition entsprechend der realen Maschinenposition bringen kann, wiederum am Beispiel des Bagger-Auslegers. Der Zylinder des Modells startet links bei  $x = 455$  mm, die Startposition der realen Maschine ist bei  $x = 10$  mm. Nach 10 s erreicht der einfache P-Regler die Sollposition mit einer Abweichung von 2 mm. In die entgegengesetzte Richtung wird nach 8 s eine Abweichung von 1 mm erzielt.

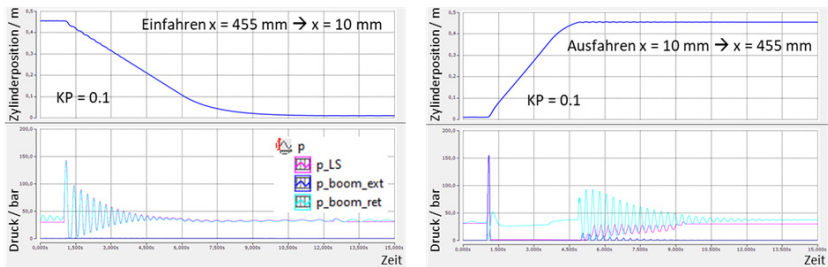


Abbildung 5.7: Testergebnisse Startwertsynchronisation

Damit ist es nun möglich, das Systemmodell in Echtzeit mit der realen Maschine zu synchronisieren. Für die nachgeschaltete Prozesskette steht hierdurch eine Vielzahl an zusätzlichen modellinternen Größen zur Verfügung, die auf der realen Maschine nicht oder nur schwer zugänglich sind, und z. B. als Input für Condition Monitoring oder vorausschauende Wartung dienen können.

In den aktuellen Entwicklungsschritten im Projekt Off-Highway-Twins 2 werden die Daten den Input für zwei weitere Prozesse bilden:

Zum einen werden die Modellparameter in einem automatisierten Ablauf optimiert (vgl. [9]). Als Qualitätskriterium bzw. Maß für die Abweichung zwischen Modell und Realität bietet sich hier der Ausgang der Synchronisationsregler nach der Startphase an.

Um Informationen zum Baufortschritt zu liefern, wird untersucht, wie KI das Bewegungsprofil und die Differenzen der gemessenen und im lastfreien Modell simulierten Zylinderdrücke nutzt. So kann auf die aktuelle Tätigkeit des Baggers und dessen Ergebnis geschlossen werden.



## 6 Zusammenfassung

Die Modellierung und Nutzung eines digitalen Zwillings über den Lebenszyklus einer Arbeitsmaschine wurden an mehreren Beispielen demonstriert.

Der modellbasierte Entwicklungsansatz erzwingt - im positiven Sinne - eine frühe Systemintegration aller beteiligten technischen Domänen. So werden Mechanik, Hydraulik, Elektronik und Maschinensteuerung bereits in der Konzeptphase im virtuellen Gesamtsystem getestet und mögliche Probleme frühzeitig im Entwicklungsprozess erkannt.

Der entstehende digitale Zwilling begleitet auch den weiteren Produktlebenszyklus. Am Beispiel der Hubarbeitsbühne wurde mit Hilfe von Fluidon Cube ein Prototyp einer 3D-Bahnsteuerung entwickelt, der die Bedienung der Hubarbeitsbühne erheblich vereinfacht. In der Betriebsphase unterstützt das Modell die Analyse komplexer Probleme im Feld oder hilft bei der Vorbereitung einer Nachrüstung von modernisierten Steuerungen für Maschinen oder Anlagen.

Der virtuelle Vertreter der Maschine ist (fast) immer und überall verfügbar. Dies ist insbesondere bei standortübergreifender Entwicklung oder eingeschränkter Verfügbarkeit von realen Prototypen von Vorteil und reduziert im Servicefall die Ausfallzeiten beim Kunden vor Ort.

Am Beispiel eines Baggers wurde gezeigt, wie das echtzeitfähige Modell beim realen Betrieb der Maschine mit dieser synchronisiert wird und dann in der Lage ist, eine Vielzahl an zusätzlichen Informationen zu liefern, die auf der realen Maschine nicht oder nur mit hohen Kosten verfügbar sind. Diese virtuellen Sensordaten dienen z. B. als Input für Condition Monitoring oder liefern Informationen für nachgeschaltete Prozesse, wie im Beispiel des Projekts "Off-Highway-Twins 2" zur Erstellung und Aktualisierung des Building Information Model.

## Literatur

- [1] Wikipedia: Digital twin, [https://en.wikipedia.org/wiki/Digital\\_twin](https://en.wikipedia.org/wiki/Digital_twin), 15.03.2022
- [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, IEEE Transactions on Automation Control, Vol. 40, 1995
- [4] Functional Mockup Interface, <https://fmi-standard.org/>, 15.03.2022
- [5] <https://fluidon.com/dshplus/anwendungen/forschungsprojekte/fluid-40>, 01.07.2024
- [6] <https://bmdv.bund.de/SharedDocs/DE/Artikel/DG/mfund-projekte/off-highway-twins-2.html>, 01.07.2024
- [7] Wikipedia: Softsensor, <https://de.wikipedia.org/wiki/Softsensor>, 25.06.2024
- [8] Akbarian, F., Fitzgerald, E., Kihl, M.: Synchronization in Digital Twins for Industrial Control Systems, 2020
- [9] Sender, B.: Automatisierte Parametrierung von Systemmodellen für mobile Arbeitsmaschinen, 10. Fachtagung Baumaschinentechnik Dresden, 26.– 27.09.2024

## Danksagung

Für das Projekt Fluid 4.0:



**Finanziert von der  
Europäischen Union**  
NextGenerationEU

Gefördert durch:



Bundesministerium  
für Wirtschaft  
und Klimaschutz

aufgrund eines Beschlusses  
des Deutschen Bundestages

Das Projekt Off-Highway-Twins 2 wird gefördert vom Bundesministerium für Digitales und Verkehr aufgrund eines Beschlusses des Deutschen Bundestages.