

## Herausforderungen und Lösungsansätze im Engineering von Produktionssystemen

# Durchgängige virtuelle Inbetriebnahme

S. Nowinski, C. Eble, L. Klingel, D. Littfinski, A. Verl

**ZUSAMMENFASSUNG** Im Engineeringprozess von Produktionssystemen werden zunehmend sich schnell ändernde Anforderungen an Steuerung und zugehörige Simulationen gestellt. Die X-in-the-Loop (XiL)-Konfigurationen, die durch Softwarewerkzeuge etabliert sind, unterstützen simulativ die Steuerungsprogrammierung und -absicherung in diesem Prozess. Dieser Beitrag stellt die Herausforderungen im Zusammenhang der XiL-Konfigurationen anhand von Erfahrungen aus der Praxis vor. Zudem wird das durchgängige Ansteuern der XiL-Konfigurationen beispielhaft mit einer Anwendung an einer Roboterzelle veranschaulicht.

### STICHWÖRTER

Engineering, Simulation, Industrieroboter

## Continuous virtual commissioning – Challenges and solution approaches for the continuous engineering of production systems

**ABSTRACT** In the engineering process of production machines, increasingly fast-changing requirements are set on simulation and related control systems. The X-in-the-Loop (XiL) configurations, which have been established through software tools, provide simulative support for control programming and validation in this process. This article shows the challenges in context with the XiL configurations based on practical experience. Furthermore, the continuous management of XiL configurations is represented using an application on a robot cell as an example.

## 1 Einleitung

Durch den ständigen Wandel der Fertigungsindustrie werden starre Produktionssysteme aufgelöst und durch flexible Strukturen ersetzt. Damit wird die Fertigung kundenindividueller Produkte bis hin zur Losgröße 1 ermöglicht [1]. Die Sicherstellung der Erschwinglichkeit von personalisierten Produkten ist eine erhebliche Herausforderung, da sie von mehreren Faktoren und ständigen Veränderungen abhängig ist [2]. Geeignete Automatisierungs- und Informationstechnologien spielen eine immer größere Rolle. Intelligente Anlagenkomponenten werden mit neuen Steuerungskonzepten und angepassten Prozessabläufen vernetzt. Damit verbunden sind steigende Anforderungen an Steuerungskonzepte und -programme für die Koordination der komplexen Prozesse und multidisziplinären Wechselwirkungen. Fehler im Steuerungsprogramm, die erst in einem späten Stadium entdeckt werden, können nur mit hohem Zeit-, Personal- und Kostenaufwand behoben werden und verzögern die ohnehin schon teure Inbetriebnahme. Angesichts dieser Umstände gewinnen geeignete Test- und Analyseverfahren für Steuerungssysteme zunehmend an Bedeutung [1].

Um diese Testverfahren im Bereich der Simulationen anzuwenden zu können, werden zunächst Modelle benötigt. Die Erstellung von Modellen erfordert eine aktive Analyse und Synthese. Eine Modellanalyse ist von Nöten, da Modelle nur eine Abstraktion der Realität darstellen und auf Eingabedaten basieren. Diese Analyse beinhaltet die Bewertung und Interpretation von Systemelementen, Teilsystemen und dem Gesamtsystem.

Während der gesamten Systementwicklung werden Produktmodelle und Daten mit Autorenwerkzeugen und Product-Lifecycle-Management-Systemen erstellt, gespeichert und organisiert [3].

Ein methodischer Ansatz, der eine strukturierte Vorgehensweise im Entwicklungsprozess bietet, ist das V-Modell. Es ist in der Automobilindustrie weit verbreitet und wird von den Sicherheitsstandards der Branche, wie ISO 26262, anerkannt. Für jede Phase des Modells werden eine Prüfliste und Qualitätsdokumente erstellt, wobei die Eintritts- und Austrittskriterien klar definiert sind. Das V-Modell erweitert das Wasserfallmodell und wird auch als Verifizierungs- und Validierungsmodell bezeichnet. Allgemein beschreibt die linke Seite des V-Modells die Entwicklungsaktivitäten, während die rechte Seite das Testen umfasst. Es ist wichtig, eine Phase abzuschließen, bevor zur nächsten übergegangen wird. Die Entwicklungs- und Testaktivitäten sollten parallel verlaufen, um die Produktentwicklungszeit zu verkürzen [4].

Das folgende Kapitel erklärt grundlegend den Zusammenhang von virtueller und realer Inbetriebnahme. Diese sind spezifisch abgeleitet durch die Phasen der Systemsimulation und Systemintegration im Rahmen des V-Modells.

Der Beitrag zielt darauf ab, bestehende Potenziale im Kontext mit den Herausforderungen der virtuellen Inbetriebnahme (VIBN) aufzuzeigen und durch eine beispielhafte Roboteranwendung praktisch zu beleuchten. Dafür wird ein durchgängiger Ansatz herausgearbeitet. Dieses Vorgehen ermöglicht ein Verknüpfen konzeptioneller Modelle mit den Handlungsempfehlungen aus den Herausforderungen, um ein Übergang zwischen theoretischer Analyse und praktischer Umsetzung zu schaffen.

## 2 Virtuelle und reale Inbetriebnahme

Im Anlagen- und Sondermaschinenbau werden die Produkte in der Regel zuerst beim Hersteller aufgebaut und in Betrieb genommen. Nach erfolgreichem „Factory Acceptance Test“ (FAT), also einer Konformitätsprüfung, wird das Produkt vor Ort beim Kunden aufgebaut und ein „Site Acceptance Test“ (SAT) der installierten Anlage durchgeführt. Der SAT dient zur finalen Überprüfung aller Abnahmekriterien sowie Planung möglicher Nachbesserungen. Ein vollständig integrierter Test von optimierten Steuerungs- und Prozessabläufen kann vor allem für Neuentwicklungen bereits zuvor im Anlagenentwicklungsprozess durchgeführt werden.

Problematisch sind unter anderem große Anlagen, die wegen ihrer Größe vorab nicht real in einer Halle aufgebaut werden können oder deren einzelne Anlagenteile an verschiedenen Orten entwickelt oder in Betrieb genommen werden. Aufgrund kurzer Projektdurchlaufzeiten ist es nicht unüblich, mehrere Maschinen oder Anlagen gleichzeitig in Betrieb zu nehmen. Dies erfordert kurzfristig einen erhöhten Platzbedarf sowie das entsprechende Personal.

Anlagenbetreiber sehen sich einem zunehmenden Wettbewerbsdruck ausgesetzt und reagieren mit kürzeren Produktlebenszyklen und einem steigenden Produktportfolio. Dadurch verkleinert sich das Zeitfenster für die Inbetriebnahme (IBN). Für diese Phase muss oftmals die Abnahmefähigkeit einer Anlage vor Ort beim Kunden verifiziert werden. Fehler im Steuerungsprogramm haben häufig eine Verzögerung der IBN zur Folge, da fehlerhafte Prozesse erst dann auftreten, wenn einzelne Komponenten funktional in einem Gesamtsystem integriert sind [1].

Ergänzend zu der IBN bietet die VIBN den Vorteil, dass Auslegungs- und Planungsfehler bereits vor dem Bau von Produktionssystemen erkannt und behoben werden können. Überlagerte Schnittstellen für die Informationstechnik werden dabei ebenfalls berücksichtigt. Die VDI 4499-2 nennt [5]:

- Semantik- oder Ablauffehler,
- Testen und Validieren von dynamischem Verhalten von Anlagen,
- Berücksichtigung von Sonderfällen,
- Prüfen der geplanten Anlagenparameter,
- gezielte Simulation von Störungssituationen.

Zudem kann nach dem FAT beziehungsweise SAT die VIBN zur Optimierung von Leit- und Steuerungssystemen verwendet werden. Insbesondere in der Zeit unmittelbar nach dem SAT, in der teilweise noch eine Produktionsbegleitung stattfindet, können Fehlerbehebungen oder Optimierungen am digitalen Modell stattfinden, ohne dass die Anlage beim Kunden blockiert wird.

In der VIBN entspricht das XiL-System dem virtuellen Testaufbau. Das Steuerungssystem wird dabei nicht im Simulationsmodell integriert, sondern daran gekoppelt. Die XiL-Konfigurationen werden unterschieden durch die Art des Steuerungssystems, den Kommunikationsweg, das Simulationsmodell und Simulationswerkzeug sowie den Modellierungsumfang. In Abhängigkeit der gewünschten Nähe zur realen Komponente, Maschine oder Anlage wird das entsprechende XiL-System eingesetzt. Mit zunehmender Realitätsnähe steigt der Erstellungsaufwand. Die Komplexität des Simulationsmodells ist davon unabhängig.

Entsprechend dem Entwicklungsfortschritt werden im Idealfall Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) und

Hardware-in-the-Loop (HiL) nacheinander eingesetzt [6]. Diese Phasen sind nach VDI/VDE 3693 wie folgt definiert [7]:

- MIL: Aktoren und Sensoren der Anlage (Anlagenmodell) mit abstrahierter Steuerung (zum Beispiel Automatenmodell) als gemeinsames Simulationsmodell auf einem Rechner vernetzt
  - SiL: Simulationsrechner mit Anlagenmodell über virtuelle Kommunikation (zum Beispiel Shared Memory, OPC, TCP/IP) mit emulierter Steuerung (zum Beispiel ICE-61131-Code) verknüpft
  - HiL: Simulationsrechner mit Anlagenmodell über reale Kommunikation (zum Beispiel Profibus, EtherCAT, Profinet) mit realer Steuerung (zum Beispiel ICE-61131-Code) verbunden
- Zum einen dient die VIBN als virtuelle Absicherung, zum anderen kann sie zur Beschleunigung der Entwicklung eingesetzt werden.

Simulationswerkzeuge, die eine integrierte Umgebung für Simulation und Steuerung bieten, können die Werkzeugkette im Engineering-Prozess vereinfachen. Durch die integrierte Umgebung sind vor allem in frühen Entwicklungsphasen weniger Schnittstellen zur Kommunikation und Austauschformate zwischen Werkzeugen notwendig.

Ein Experiment zur Entwicklung eines Steuerungsprogramms für ein Materialtransportsystem zeigte, dass schon in kleinen Projekten durch die VIBN die Zeit für die Fehlersuche reduziert und eine Qualitätssteigerung beobachtet werden kann [8]. So kann die Projektdauer verkürzt und die Softwarequalität gesteigert werden [1]. Eine ganzheitliche und durchgängige Softwareentwicklung ermöglicht es, dass Anwendungsprogramme durch die Automatisierung der Software-Lieferkette schneller in die Produktion überführt werden können. Dies führt zu sicheren Programmen von besserer Qualität, erlaubt schnelleres Feedback und führt zu einer schnelleren Markteinführung des Produkts [9].

Grundlegend kann durch eine Simulationsumgebung ein Simulationsexperiment gestartet werden [10]. Dabei ist es möglich, einen echtzeitfähigen Solver zu verwenden, der das Simulationsmodell der virtuellen Produktionsanlage zur Laufzeit berechnet [11]. Ein virtuelles Modell wird mit 3D-CAD-Dateien erstellt. Bewegungen können dann mit einem virtuellen Modell mit einem Werkzeug geplant werden. Dies erfolgt durch die Anpassung von Punkten an geometrische Elemente. Reaktionen auf Ein- und Ausgaben oder Programmpositionen können durchgeführt werden. Layouts können optimiert, Kollisionen geprüft oder die Erreichbarkeit von Bewegungspunkten verbessert werden. Das Bewegungsprogramm wird in die herstellersistemische Sprache für Roboter generiert, gespeichert und übertragen. Aufgrund von Modellabweichungen zur Realität können nachfolgend Anpassungen erforderlich sein, die sich jedoch durch den Einsatz von Sensoren minimieren lassen [12].

Moderne Robotersysteme werden für einen spezifischen Einsatzzweck explizit mit vordefinierten Flexibilitätsreserven geplant und installiert. Besonders bei der Planung, IBN und Integration in bestehende Fertigungssysteme entstehen signifikante Zeitaufwände. Von zukünftigen Robotersystemen wird erwartet, dass sie schnell und nach Möglichkeit ohne manuellen Integrationsaufwand für einen Einsatzzweck konfiguriert werden können [13].

Befragte Unternehmen aus dem VDMA-Arbeitskreis „Simulation und Visualisierung im Maschinenbau“ sehen den größten Bedarf und das höchste Entwicklungspotenzial von Simulationstechnologien im Bereich VIBN [14].

Tabelle. Verteilung und Abhängigkeiten der klassifizierten Herausforderungen in den XiL-Phasen.

| Klassifizierung \ Phasen | Generell | MiL | MiL/SiL | SiL | SiL/HiL | HiL   |
|--------------------------|----------|-----|---------|-----|---------|-------|
| Personal                 | 1        |     |         |     |         |       |
| Projektleitung           | 2 3      |     |         |     |         |       |
| Engineering              | 3 4      |     |         |     |         |       |
| Elektrokonstruktion      | 5        |     |         |     |         |       |
| Safety                   | 6        |     |         | 12  |         |       |
| Simulationskonfiguration | 7        |     | 10      | 11  |         | 18 19 |
| Modellbildung            | 8        | 9   | 10      | 11  | 13 14   | 20 21 |
| Steuerung                | 6 7 8    | 9   |         | 12  |         | 21    |
| Feldbuskonfiguration     |          |     |         |     | 13      | 17 18 |
| Steuerungskonfiguration  |          |     |         |     | 14      |       |
| Hardware                 |          |     |         |     |         | 15    |
| Safetykonfiguration      |          |     |         |     |         | 17    |
| Hardwarekonfiguration    |          |     |         |     |         | 16    |

Die derzeitigen Ansätze für die VIBN sind aufgrund unterschiedlicher Implementierungen und Rahmenbedingungen zeitaufwendig und erfordern, neben dem organisatorischen Vorbereiten und Integrieren, die technischen Aufwände wie ein manuelles Umschalten zwischen den drei verschiedenen XiL-Konfigurationen. Zudem müssen bisherige Bewegungssteuerungen, vor allem bei Robotersteuerungen, von jedem Teilsystem einer Anlage mit seiner anlagenspezifisch proprietären Programmiersprache, je nach Hersteller, einzeln erstellt werden.

Mechanische Komponenten, Antriebstechnik, Steuerungstechnik, Software sowie produktionstechnische und -organisatorische Aspekte sind eng miteinander verknüpft und erfordern einen systematischen Ansatz, Erfahrungen und Kompetenzen, die in ihrer Gesamtheit für viele Unternehmen neu sind. Dies trifft vor allem für Anwendungen in der Industrierobotik zu, da diese Automatisierungslösungen weitgehend anwendungsspezifisch sind [15].

Aus diesem Grund ist es von entscheidender Bedeutung, Methoden für eine durchgängige VIBN zu entwickeln, um diese engen Verknüpfungen zu lösen.

### 3 Herausforderungen in der Industrie

Um zunächst einen Einblick zu geben, welche Herausforderungen im VIBN-Prozess aktuell sind, und in welchen Bereichen der Bedarf an Durchgängigkeit vorhanden ist, werden aktuelle Punkte aus dem Feld des Sondermaschinenbaus aufgeführt. Die Punkte basieren auf Erfahrungen, die in Gesprächen und Interviews mit Workshop-Charakter durch Beteiligte und Initiatoren des AiF-Forschungsprojekts (unter anderem Anlagenherstellern, Dienstleistern, Entwicklungsingenieuren und Integratoren) entlang der VIBN-Prozessketten gesammelt wurden.

Zu Beginn des Kapitels werden die Forschungsmethoden aufgeführt, gefolgt von den generellen Herausforderungen die sich über den vollständigen VIBN-Prozess erstrecken. Anschließend werden ausgewählte Lösungsmöglichkeiten diskutiert.

#### 3.1 Forschungsdesign und Forschungsmethoden

Neben den aufgeführten XiL-Konfigurationen der VIBN ist auch die jeweilige Einordnung der funktionsorientierten Steuerungsebenen von Relevanz. Nach *Pritschow* lassen sich diese Ebenen in einem hierarchischen Modell darstellen – von der Fabriksteuerung bis hin zur Prozessebene, wobei mit zunehmender Prozessnähe die Ansprüche an das Zeitverhalten steigen und der Datenaustauschumfang im Allgemeinen abnimmt [16].

In der VIBN beteiligte Disziplinen wie Planung, Engineering, Simulation, Steuerung und Safety lassen sich nicht nur bestimmten Prozessphasen, sondern auch spezifischen Steuerungsebenen zuweisen. So sind beispielsweise Tätigkeiten wie Prozessplanung der Planungsebene zuzuordnen, während modellbasierte Simulationen auf der Zellebene, Maschinenebene oder darunter angesiedelt sind. Die reale Ansteuerung erfolgt schließlich auf Einzelsteuerungs- und Prozessebene, typischerweise im Rahmen der Integration oder finalen Testphasen.

Die Klassifikationen in der nachfolgenden **Tabelle** basieren auf der Norm ISO/IEC/IEEE 15288:2023 [17], welche, neben dem genannten V-Modell, Prozesse entlang des Lebenszyklus technischer Systeme definiert. Die Zuweisung spezifischer Tätigkeiten zu den Phasen der VIBN ermöglicht eine Einordnung der funktionalen Abläufe und Zuständigkeiten innerhalb interdisziplinärer Entwicklungsprozesse.

In den folgenden Abschnitten werden die verschiedenen XiL-Konfigurationen mit ihren Übergangsphasen generell und im Einzelnen betrachtet. Dabei wird auf die in der Tabelle aufgeführten Punkte verwiesen.

#### 3.2 Generell in allen Konfigurationen der VIBN

Generell ist für eine VIBN Expertenwissen in mehreren Fachgebieten nötig. Fehlt dieses Wissen, ist eine VIBN nicht oder nur eingeschränkt möglich (Punkt 1).

Kapazitätsengpässe verzögern die Modellbildung und VIBN durch kurzfristige Änderungen von Projektlaufzeiten und hohe

Projektdichte (Punkt ②). Kurzfristige Änderungen von Anforderungen durch den Kunden können weitreichende Auswirkungen haben und einen Bruch in der VIBN erzeugen (Punkt ③).

Änderungen durch den Konstrukteur oder Inbetriebnehmer haben zur Folge, dass ein Modellabgleich / -anpassung mit dem Erhalt der Funktionalitäten getestet werden müssen (Punkt ④).

Wenn die Sensor-/Aktor (SA)-Liste und der Schaltplan nicht rechtzeitig fertiggestellt sind, verzögern sich Modellierungsstart und Modellfertigstellung beziehungsweise die VIBN (Punkt ⑤).

Safety-Projekte werden im Projektverlauf für die SPS (speicherprogrammierbare Steuerung) entwickelt. Für die Lauffähigkeit des Simulationsmodells muss die Safety-Logik (Signalverlauf) modelliert und Signale passend manipuliert werden. Ohne die Berücksichtigung von Safety ist ein Modell nicht mehr lauffähig. Alternative Lösungen, wie steuerungsseitiges Exkludieren der Safety, würde zu unterschiedlichen Projektständen führen (Punkt ⑥).

Eine VIBN beziehungsweise Ablaufsimulation mit Robotern ist nicht uneingeschränkt möglich, da in der Regel eine Co-Simulationsschnittstelle zwischen VIBN- und Roboter-Software benötigt wird (Punkt ⑦).

Roboterprogramme werden grundsätzlich erst während der IBN fertiggestellt. Abweichungen zwischen realer Anlage und idealem VIBN-Modell sind oft nicht vermeidbar. Die VIBN der Gesamtanlage ist unter anderem vom Fertigstellungszeitpunkt der Roboterprogramme abhängig. Anlagenteile können sonst nur eingeschränkt getestet werden (Punkt ⑧).

### 3.3 MiL

In der MiL-Phase werden für Machbarkeitsuntersuchungen, Materialflusssimulationen oder Taktzeitanalysen in der Regel zeit-/ereignisdiskrete Simulationsmodelle verwendet, die anschließend nicht zur VIBN verwendet werden können. Roboterabläufe können unter Umständen separat in VIBN-Software eingelernt werden. Durchgängige Lösungen existieren nur eingeschränkt. Die Modellbildung und die Generierung von primitiven Steuerungssequenzen erfordern viel Zeit und Expertenwissen. Die Inputqualität bestimmt die Ergebnislänge (Punkt ⑨).

Bei modularen, vernetzten Anlagen wird bereits für die MiL-Konfiguration eine zentrale Primitivsteuerung (Leitsystem) benötigt. Dezentrale Primitivsteuerungen auf der Maschinen- oder Zellenebene reichen nicht aus. Die Generierung einer zentralen Steuerungseinheit ist nicht trivial und erfordert Expertenwissen in mehreren Disziplinen. Ohne eine übergreifende Primitivsteuerung sind etwa Machbarkeitsuntersuchungen nicht möglich (Punkt ⑩).

### 3.4 Übergang MiL zu SiL

Für den Übergang zwischen MiL zu SiL müssen unter Umständen Verhaltensmodelle detaillierter abgebildet werden. Bestehende Modelle sind zu erweitern oder auszutauschen. Können Modelle nicht automatisiert ausgetauscht oder angepasst werden, entsteht beim Phasenübergang ein enormer Modellierungsaufwand (Punkt ⑪).

Unabhängig von bereits bestehenden Verhaltensmodellen werden für die emulierte Steuerung beziehungsweise für das Steuerungsprojekt meist zusätzliche Signale benötigt, zum Beispiel für Sicherungen, die Sensor-/Aktor-Spannungen, und weitere Abhän-

gigkeiten. Werden diese neuen zusätzlichen Signale nicht entsprechend simuliert, ist eine Ablaufsimulation nicht weiter möglich, sodass der VIBN-Ablauf somit zeitweise unterbrochen wird (Punkt ⑫).

### 3.5 SiL

Um Verhaltensmodelle in der nachfolgenden SiL-Phase mit Ein- und Ausgangssignalen der SPS versorgen zu können, wird zum Signalaustausch zwischen emulierter Steuerung und VIBN-Software ein direkter Datenkanal benötigt. Unterstützt eine VIBN-Software keinen direkten Austausch mit den emulierten Steuerungen über entsprechende Protokolle oder haben emulierte Steuerungen nur eingeschränkte Funktionalität, ist SiL in dieser Kombination unpassend (Punkt ⑬).

Prinzipiell ist ein Safety-Projekt Bestandteil jeder Anlage. Das Safety-Projekt beziehungsweise die Safety-Logik läuft auf einer Safety-Logik-Klemme oder wird durch eine virtuelle Safety meist direkt auf der jeweiligen Steuerung eingebunden. Kann die Safety bei SiL nicht integriert werden, müssen Änderungen im Steuerungsprojekt vorgenommen werden oder generelle Umschaltmöglichkeiten für den Betrieb mit der realen oder virtuellen Anlage vorgesehen werden (Punkt ⑭).

### 3.6 Übergang von SiL zu HiL

Wenn die Feldbustopologie bei dem Wechsel zwischen SiL zu HiL noch nicht final verfügbar ist, verzögert sich die modellseitige Steuerungsanbindung an den Feldbus oder muss iterativ aktualisiert werden (Punkt ⑮).

Für die iterative Erweiterung und Aktualisierung der HiL-Steuerungsanbindung und der bereits implementierten Verhaltensmodelle ist in der Regel ein höherer Detaillierungsgrad bei Feldbus- und Antriebskomponenten nötig. Daraus resultiert ein zusätzlicher Modellierungsaufwand zum Umschalten von Steuerungssignalen und Verhaltensmodellen (Punkt ⑯).

### 3.7 HiL

Eine VIBN umfangreicher Anlagen oder parallel mehrerer Anlagen erfordert enorme Investitionen in die Hardwareinfrastruktur. Ohne die passende Hardware ist die HiL-Phase, und damit eine vollständige VIBN nicht möglich (Punkt ⑰).

Die Hardware muss physisch vorliegen und konfiguriert werden. Ohne diese ist nur eine eingeschränkte Flexibilität möglich. Abhängig von der Steuerungsarchitektur, Anlagentyp und -größe ist eine HiLS unter Umständen technisch nur sehr aufwendig oder nicht praktikabel umsetzbar beziehungsweise im Vergleich zu SiL-Steuerungen kaufmännisch nicht vertretbar (Punkt ⑱).

Die Safety-Logik-Klemme muss in der Regel physisch mithilfe eines Buskopplers in den Feldbus integriert werden. Dafür ist Hardware- und Expertenwissen notwendig (Punkt ⑲).

Die Feldbussimulation kann durch unterschiedliche Störeinflüsse beeinträchtigt werden. Aufgrund von Konfigurations- oder Stabilitätsproblemen kann ein fehlerhafter Signalaustausch erfolgen und das Debuggen der eigentlichen Programmierarbeit erschweren. Das mögliche Fehlerspektrum ist breit und verhindert eine reibungslose VIBN. Die Akzeptanz aller Beteiligten sinkt dadurch (Punkt ⑳).

„Harte Echtzeitanforderungen“ erfordern unter Umständen Anpassungen im Steuerungsprojekt, zum Beispiel Zykluszeiten von NC (Numerical Control)-Achsen. Je nach Modellgröße können zu geringe Rechenschrittweiten nicht eingehalten werden. Aufgrund der für eine stabile Simulation nötigen Änderungen liegen für das Steuerungsprojekt folglich unterschiedliche Stände vor (Punkt 19).

Virtuelle Modelle von Feldbusgeräten sind oft nicht verfügbar und müssen je nach Detaillierungsgrad aufwendig modelliert werden. Daraus resultiert eine Verzögerung der Modellbildung und VIBN an sich. Je nach Projektzeitpunkt kann eine fehlende Verfügbarkeit virtueller Komponenten folgeschwer sein (Punkt 20).

Je nach Anlagenkonzept und -größe besteht die Möglichkeit, dass einzelne Anlagenteile an unterschiedlichen Orten entwickelt, hergestellt und in Betrieb genommen werden können. Eine ganzheitliche VIBN mit einer HiL-Konfiguration ist nicht praktikabel umsetzbar (Punkt 21).

### 3.8 Grenzen für Unternehmen

Abhängig von der Art des Unternehmens sind unterschiedliche Grenzen zur Optimierung gesetzt. Für kleine und mittelständische Unternehmen können ein begrenztes Budget und Personal komplexe Automatisierungs- und Modularisierungslösungen erschweren. Dabei ist zu hinterfragen, in welchen Dimensionen sich das jeweilige Unternehmen bewegt, damit keine ineffizienten Skalierungen erfolgen beziehungsweise abnehmende Skalenerträge resultieren. Bei Konzernen können langsame Entscheidungsprozesse und bürokratische Hürden dazu führen, dass bereichsübergreifende Innovationen und neue Technologien erst spät eingeführt werden können. Allgemein können zusätzlich Widerstände in Arbeitsprozessen gegenüber Veränderungen auftreten.

Neben den aufgezeigten Grenzen werden im Nachfolgenden die Ergebnisse eingeordnet.

### 3.9 Diskussion und Einordnung der Ergebnisse

Die zuvor identifizierten Punkte verdeutlichen die Vielschichtigkeit der Herausforderungen über die gesamte VIBN hinweg. Durch Berücksichtigung des Erfahrungswissens aus den Herausforderungen lassen sich nachfolgend allgemeine Best Practices der durchgängigen VIBN ableiten.

Ein zentraler Aspekt der Prozess- und Effizienzoptimierung in den XiL-Konfigurationen ist die Vermeidung veralteter Deployment-Prinzipien und der Auf- oder Ausbau von automatisierten Pipelines, um Tests und Deployments frühzeitig über die komplette Engineering-Kette zu integrieren und synchronisieren. Ergänzend dazu sorgen kurze, agile Entwicklungszyklen (im Vergleich zu einem Komplett-Release eines Projekts) mit kontinuierlichen Feedback-Schleifen für die transparente und schnelle Identifizierung von Engpässen.

Zudem führt eine Infrastrukturautomatisierung zu schnell rekonfigurierbaren, konsistenten, und vernetzten Umgebungen, vor allem durch vordefinierte Konfigurationen und der skriptbasierten Generierung. Dadurch können teilautomatisierte, manuelle und damit fehleranfällige Eingriffe vermieden werden. Die Unterteilung monolithischer Systeme in lose gekoppelte, unabhängige Komponenten sorgt dabei für eine bessere Skalierung und Verlagerung auf frühe Entwicklungsphasen.

Die Flexibilität, das Management und Steuerung der XiL-Konfigurationen kann durch die Verwendung von Monitoring-Tools (Audits) und automatisierten Prüf- und Freigabemechanismen verbessert werden. Die Überprüfung der Vollständigkeit machen dabei „Security-by-Design“-Architekturen möglich. Weiterführend sollten offene statt isolierte und proprietäre Schnittstellen und Standards durch die Architektur vorgegeben werden, um nahtlose Integrationen und Erweiterungen zu erlauben. Die Entwicklung von und mit modularen und anpassungsfähigen Bausteinen sorgt für die nachhaltige Wiederverwendung, im Vergleich zu einem geschlossenen Black-Box-System. Die Reduzierung von redundanten oder obsoleten Entwicklungen kann durch die Verwendung von Single-Source-of-Truth (SSOT)-Systemen erreicht werden, um den Mehraufwand zu eliminieren.

Eine Schlüsselfunktion übernimmt die Kommunikation und Organisation durch eine ereignisbasierte Architektur für asynchrone Kommunikation, Dokumentation und schnellere Anpassungen sowie verbesserte Zusammenarbeit und Interoperabilität zwischen Projektbeteiligten.

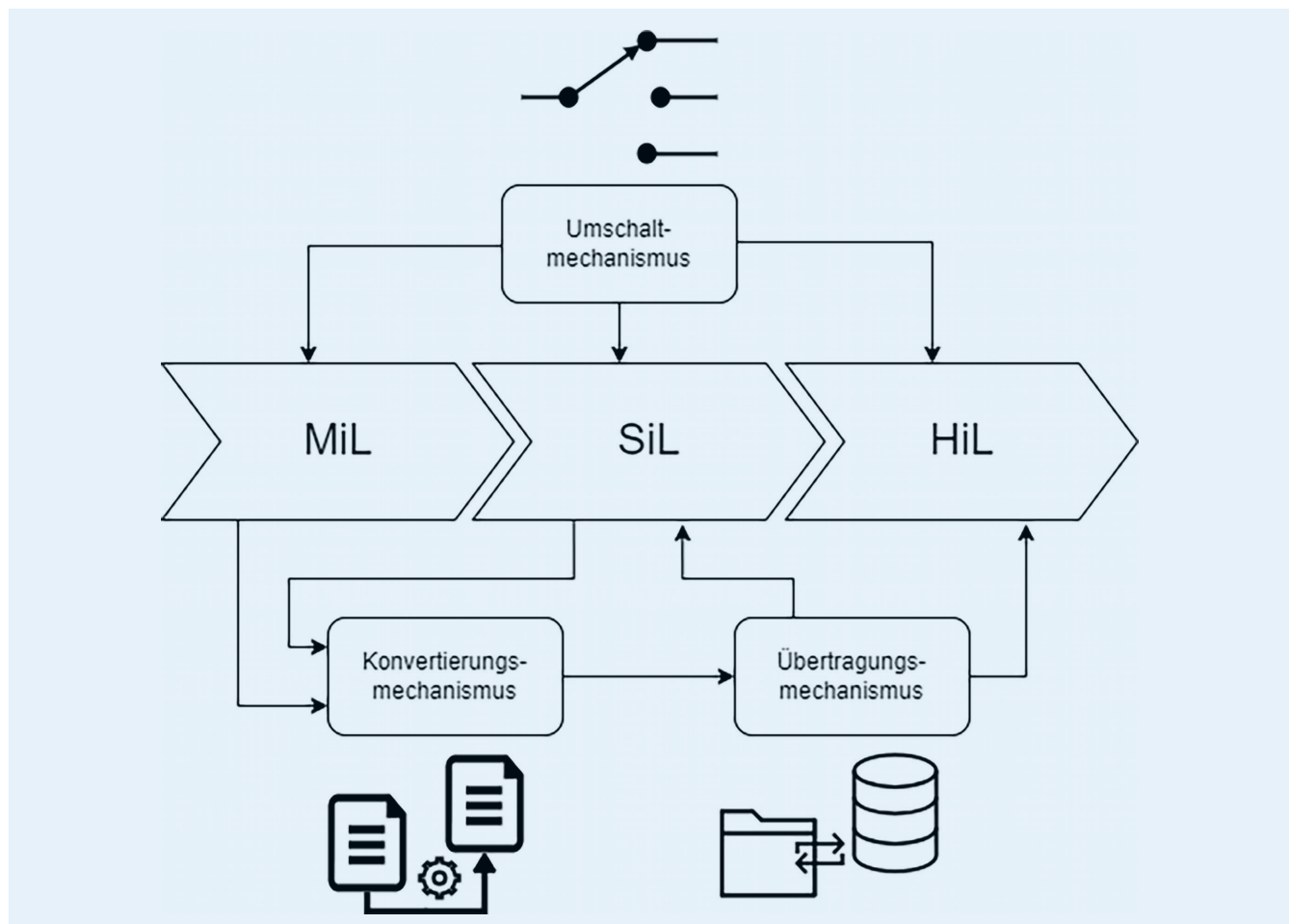
Zur Fokussierung der Konzeptentwicklung kann eine erste Auswahl getroffen werden, die sich an den Best Practices sowie an der ursprünglichen Problemstellung orientiert. Die Auswahl umfasst vor allem die Punkte 2, 3, 4, und 7 der Herausforderungen, welche im nachfolgenden Konzept in einem ersten Ansatz aufgegriffen werden.

## 4 Konzept der durchgängigen VIBN

Ziel ist es, eine Architektur zu entwickeln, welche die genannten industriellen Herausforderungen adressiert. Im ersten Ansatz wird eine Architektur vorgestellt, die grundlegende dynamische, erweiterbare und entkoppelte Methoden für eine durchgängige VIBN anhand eines Anwendungsfalls bereitstellt. Dadurch soll ein dynamisches Umschalten und das Erstellen und Übertragen von Programmen von numerischen Bewegungssteuerungen in ein systemneutrales Format zwischen den XiL-Konfigurationen ermöglicht werden.

Die genannten Best Practices lassen sich nachfolgend auf die Architektur des Anwendungsfalls übertragen. Dabei werden integrierte Schnittstellen verwendet, die unabhängig von proprietären Datenformaten eingesetzt werden können. Für eine Kommunikation zwischen Simulations- und Steuerungskomponente wird eine Anwenderschnittstelle mit Benutzerinteraktionsmöglichkeiten als zentrale Einheit definiert. Die Kommunikation zwischen den Komponenten umfasst dabei Signale, Daten, Interaktionen, und Befehle. Abhängig von der Konfiguration der Anwenderschnittstelle können die unterschiedlichen XiL-Konfigurationen angesprochen werden. In jeder XiL-Konfiguration wird eine Instanz des Simulationstools und dessen Anbindung benötigt. Die Konfigurationsphasen SiL und HiL sind jeweils an die Steuerung gebunden. Für MiL ist zwar keine direkte Ansteuerung durch das Steuerungstool erforderlich, jedoch muss die im Simulationstool integrierte MiL-Steuerung mit den Folgephasen abgeglichen werden.

Im Folgenden werden Mechanismen für die Interoperabilität zwischen Simulation und Steuerung eingeführt. Die Mechanismen werden nach der Erstellung und Initialisierung der XiL-Phasen iterativ angewendet: Zuerst ein Umschaltmechanismus für den Wechsel von Schnittstellen und Konfigurationen, dann ein Konvertierungsmechanismus für die Anpassung von Steuerungs-



**Bild 1.** Anwendung der Mechanismen auf die XiL (X-in-the-Loop)-Konfigurationen. Grafik: ISW Uni Stuttgart

programmen und zuletzt ein Übertragungsmechanismus für die Sicherstellung des Datenaustauschs. Der Zusammenhang der Mechanismen mit den XiL-Konfigurationen ist in **Bild 1** dargestellt.

#### 4.1 Umschaltmechanismus

Das Konzept des Umschaltmechanismus, lehnt sich an *Nibert et al.* an, um eine Modularität zwischen den XiL-Konfigurationen zu gewährleisten. Ein einziges funktionales Modell wird dabei in verschiedenen Entwicklungsstufen verwendet. Auf diese Weise können die funktionalen Komponenten von den Schnittstellenkomponenten entkoppelt werden [18]. Das funktionale Modell folgt somit dem SSOT-Konzept.

Neben dem Umschalten zwischen den I/O-Schnittstellen, kann der Mechanismus verwendet werden, um Blöcke im Blockdiagramm der Simulation dynamisch zu manipulieren. Dies erlaubt die flexible Anpassung an verschiedene Testanforderungen und Simulationsphasen. Die Manipulation umfasst dabei das Ein- und Ausschließen bestimmter Simulationsblöcke sowie die dynamische Änderung von Parametern. Dabei wird besonders auf den Synergieeffekt eingegangen, der durch die Verwendung des Echtzeitbetriebssystems und das Ausnutzen der Parallelisierung der Modellberechnung beim Umschalten entsteht [11].

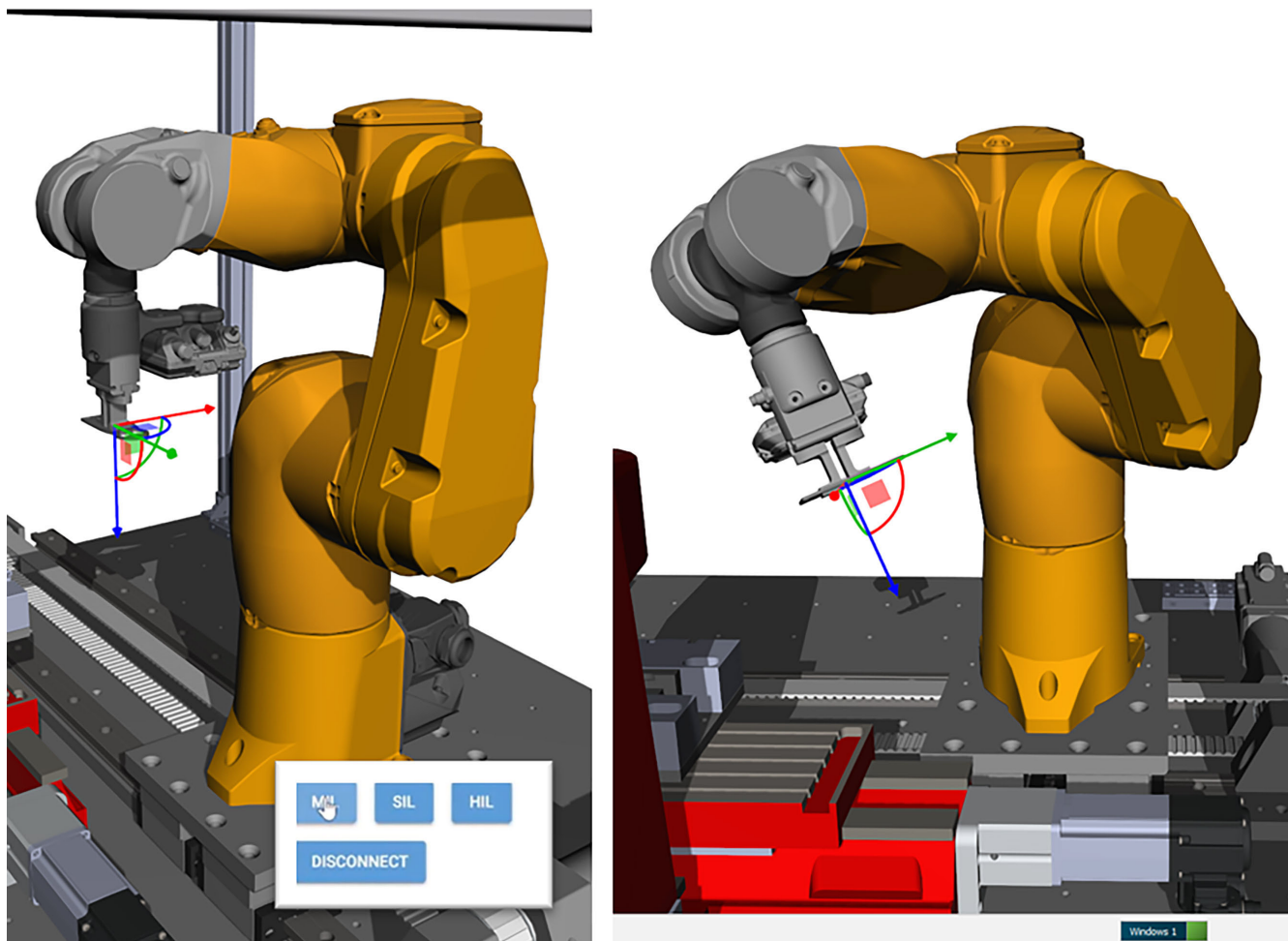
#### 4.2 Konvertierungsmechanismus

Der Konvertierungsmechanismus wird vor allem für den Übergang zwischen MiL und HiL (beziehungsweise MiL und SiL) eingeführt und ist folglich auch nach der HiL-Phase für Rekonfigurationen der jeweiligen Anlage gedacht. Dabei soll eine Konvertierung zwischen der erzeugten Maschinen- oder Steuerungssprache aus dem Ablaufprogramm einer integrierten MiL-Steuerung des Simulationswerkzeugs, in ein geeignetes SPS-, NC-, oder Roboterprogramm für die reale Steuerung stattfinden.

Beim Übergang gleicher, standardisierter Sprachen unterscheidet sich die Syntax zwischen Simulation und Steuerung nicht signifikant voneinander, da G-Code grundsätzlich auf DIN 66025-1 [19] und DIN 66025-2 [20] basiert und SPS-Programmiersprachen auf IEC 61131-3 [21] oder durch eine dezentrale, vernetzte Automatisierungssprache wie IEC 61499 [22] festgelegt sind. Dennoch werden dabei Parameter und Strukturelemente für den Übergang angepasst, um das Soll-Verhalten zwischen den XiL-Phasen korrekt darzustellen.

#### 4.3 Übertragungsmechanismus

Der Übertragungsmechanismus ermöglicht die Datenkonsistenz und das Deployment von Steuerungsprogrammen auf die reale Steuerung. Die Kommunikation wird nach dem Server-Client-Prinzip aufgebaut. Dabei wird zusätzlich ein Zwischen-



**Bild 2.** Links: Umschalten in die MiL (Model-in-the-Loop)-Konfiguration durch die GUI (Graphical User Interface); rechts: Durchführen des MiL-Teachings mit der Erstellung eines Ablaufprogramms [Software: ISG-virtuos]. Grafik: ISW Uni Stuttgart

speicher verwendet, um Verbindungsprobleme, oder zeitversetzte Zugriffe des Clients zu kompensieren. Die entscheidende Funktion des Übertragungsmechanismus ist, dass sowohl Simulation als auch Steuerung jederzeit auf dieselben Datenzustände zurückgreifen können, selbst wenn der Client (Steuerung) einen asynchronen Zugriff auf den Server durchführt, um die Daten zu synchronisieren.

## 5 Realisierung

Um die Logik für die durchgängige VIBN weiterführend visualisieren und steuern zu können, wird eine entwickelte web-basierte Anwenderschnittstelle eingesetzt. Die aufgeführten Mechanismen werden in diese integriert. Durch die Verwendung einer PC-basierten TwinCAT-Steuerung ist in diesem Anwendungsfall eine SiL-Konfiguration nicht notwendig.

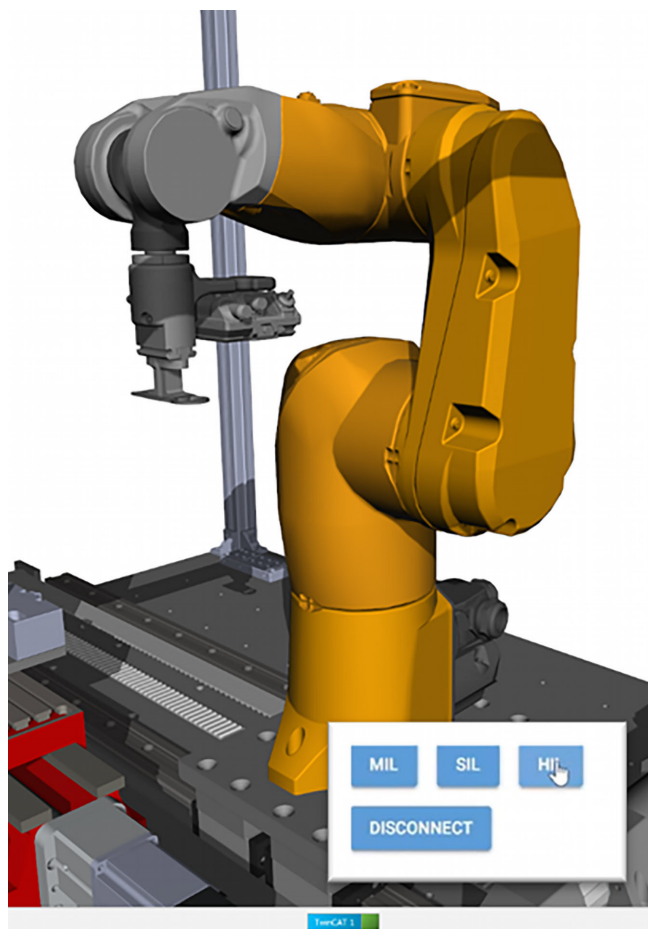
Durch das Ausführen eines Skripts wird automatisch das mit XiL-Adaptern modifizierte Simulationsprojekt initiiert, gefolgt von einer GUI (Graphical User Interface). Die GUI ermöglicht die Auswahl der XiL-Konfigurationen über drei Buttons. Durch die Auswahl einer XiL-Konfiguration werden die nötigen Parametrierungen und Verknüpfungen in der Simulation mit allen benötigten Solvern, Blöcken sowie Ein-/und Ausgängen automatisch erstellt. Anschließend wird automatisch eine Initialisierung der

Simulation durchgeführt, und diese in den Start-Modus versetzt. So kann etwa durch das Auswählen der MIL-Phase direkt das Einlernen von Punkten in dem Simulationsmodell erfolgen. Mit diesen Punkten kann dann ein Ablaufprogramm erstellt werden. Das wird in **Bild 2** verdeutlicht.

Durch den Wechsel der Konfiguration beispielsweise in die HIL-Phase wird automatisch ein Herunterfahren und Stoppen der Simulation der vorherigen Phase durchgeführt, gefolgt von einer automatischen Rekonfiguration der Simulation mit einer erneuten Initialisierung und dem Start der neuen Phase, wie **Bild 3** zeigt.

Anschließend kann das Konvertieren der NC-Datei über die GUI erfolgen. Da die Simulation (Slave) durch das Umschalten schon betriebsbereit ist, kann direkt über die GUI das Steuerungsprojekt (Master) in den TwinCAT-„Run-Mode“ versetzt werden. Der Ablauf wird in **Bild 4** deutlich.

Nach erfolgreicher Konvertierung wird die neue Datei im Zwischenspeicher abgelegt. Sobald der Client, also der Steuerungs-PC, die Dateien synchronisiert hat, kann der Anwender die einzelnen NC-Dateien im jeweiligen Kanal der HMI (Human-Machine-Interface)-Steuerung öffnen und nach einer Leistungsfreigabe der Achsen einen automatischen Programmstart durchführen. Dieser Ablauf ist an dem virtuellen und realen Modell eines „Stäubli TX2-40“-Roboters in **Bild 5** ersichtlich.



**Bild 3.** Umschalten in die HiL (Hardware-in-the-Loop)-Phase [Software: ISG-virtuos]. Grafik: ISW Uni Stuttgart

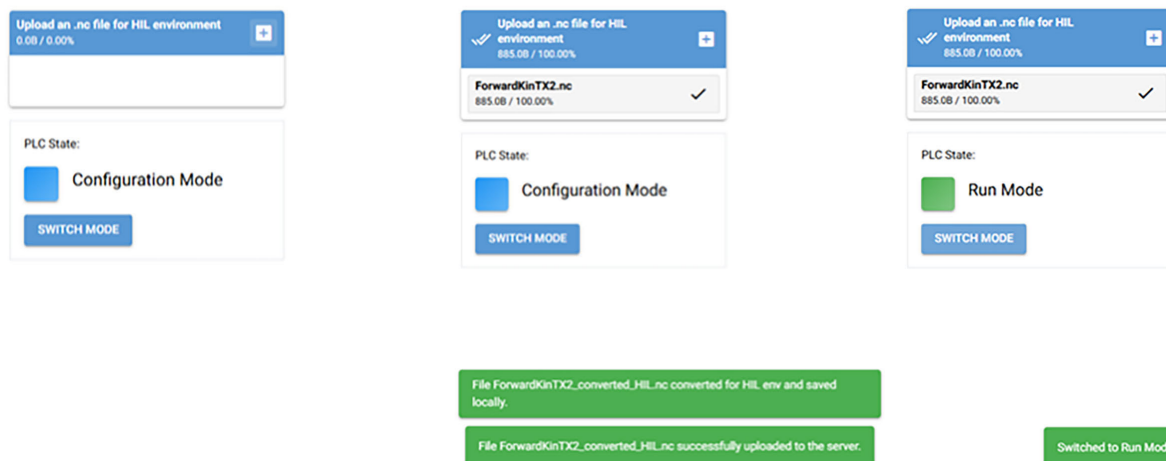
Der für die HiL-Phase verwendete Industrie-PC (IPC) kann direkt für die reale IBN eingesetzt werden. Somit können die zuvor erstellten Simulations- und Steuerungsfortschritte wiederverwendet werden. Nach dem Durchführen der Tests zwischen dem IPC und dem Simulations-PC in der HiL-Phase kann die EtherCAT-Verbindung nun zwischen dem IPC und dem realen Buskoppler hergestellt werden. In einem späteren Entwicklungsverlauf ist es möglich, die Tests bei Änderungen mit dem vorgestellten Ansatz wiederholt durchzuführen.

## 6 Zusammenfassung und Ausblick

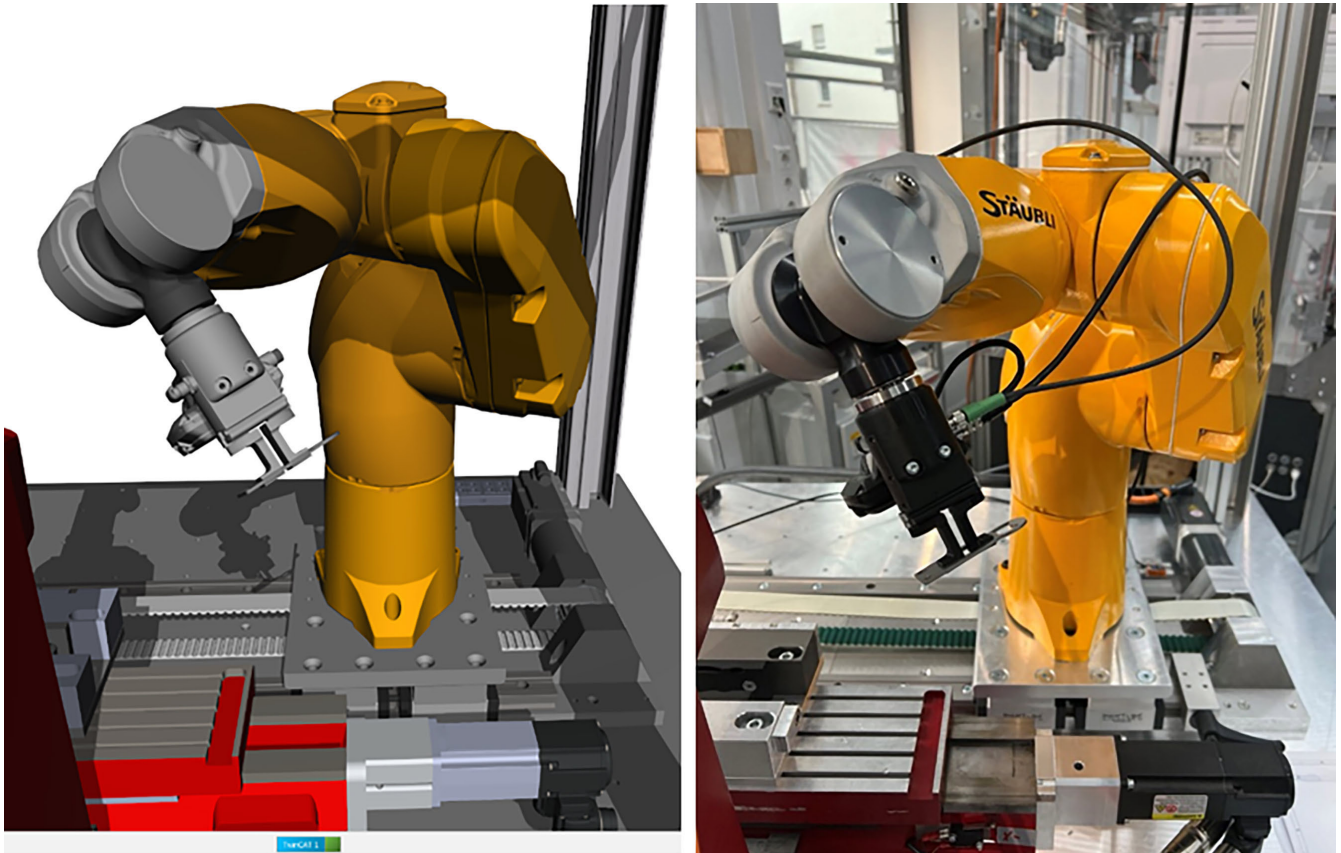
Dieser Beitrag beschäftigte sich mit Methoden für die durchgängige VIBN. Dafür wurden Herausforderungen in der VIBN im Sondermaschinenbau aufgeführt. Anhand dieser wurden dann die Best Practices eingeführt. Sie stellten den Grundstein für ein Konzept, dass die Durchgängigkeit für gesteuerte Maschinen ermöglicht. Dafür wurden drei Mechanismen eingeführt. Diese Mechanismen können nach dem iterativen Aufbau der XiL-Phasen integriert werden. Ein Umschaltmechanismus ermöglicht den Wechsel von Schnittstellen und Konfigurationen innerhalb der VIBN-Phasen. Darauf folgt ein Konvertierungsmechanismus, der die Steuerungsprogramme anpasst und zwischen Folgephasen konvertiert. Schließlich wird ein Übertragungsmechanismus eingesetzt, der den Austausch der NC-Daten zwischen den Phasen sicherstellt.

Die Verwendung der Mechanismen durch eine Webanwendung an der realen Anlage zeigt, dass ein dynamisches Umschalten, Konvertieren und Übertragen zwischen Simulation und Steuerung umgesetzt werden kann.

Die verschiedenen Einschränkungen im derzeitigen VIBN-Prozess zeigen, dass Kontinuität und Flexibilität beeinträchtigt sind. Zu den Herausforderungen gehören die dynamischen Anforderungen an das Simulationsmodell, die manuelle Anpassung und Synchronisation von Simulationsmodellen und realen Anlagen sowie die fehlende Standardisierung von Prozessen und Schnittstellen. Diese Einschränkungen erfordern aufwendige Anpassungen und führen zu zeitintensiven Projekten, insbesondere wenn Änderungen von Zulieferern oder neue Testanforderungen in den Prozess eingebunden werden müssen.



**Bild 4.** Auswahl und Konvertieren der NC (Numerical Control)-Datei aus der Mil-Steuerung mit Auswahlsatz an Konfigurationsparametern (links); lokales Abspeichern und Austausch mit der Steuerung (Mitte); Umschalten in den Run-Modus des Steuerungsprojekts auf dem Sim-PC (rechts) [UI framework: NiceGUI]. Grafik: ISW Uni Stuttgart



**Bild 5.** Ausführung eines MiL-Bewegungsablaufs auf der Steuerung in einer HiL-Konfiguration (links) und im realen Betrieb (rechts) [Software: ISG-virtuos].  
Grafik: ISW Uni Stuttgart

Die vorgestellten Mechanismen bieten erste Lösungsansätze, um eine durchgängige VIBN zu ermöglichen. Sie weisen aber bei den Herausforderungen noch erhebliches Optimierungspotenzial auf. Zukünftig können Standardisierungsansätze und vernetzte Maschinen mit mehreren Automatisierungsplattformen und Hierarchieebenen in fortführenden Forschungsprojekten untersucht werden, um sich den Herausforderungen vollständig zu stellen.

## FÖRDERHINWEIS

AiF Projekt GmbH – KK5311205RL4.

## LITERATUR

- [1] Striffler, N.; Voigt, T.: Concepts and trends of virtual commissioning – A comprehensive review. *Journal of Manufacturing Systems* 71 (2023), pp. 664–680
- [2] Ahleroff, S.; Zhong, R. Y.; Xu, X.: A Digital Twin Reference for Mass Personalization in Industry 4.0. *Procedia CIRP* 93 (2020), pp. 228–233
- [3] VDI/VDE-Gesellschaft: VDI/VDE 2206: Entwicklung mechatronischer und cyber-physischer Systeme. Berlin: Beuth Verlag 2021
- [4] Vinoth Kannan, K.: Model-Based Automotive Software Development. In: Kathiresan, M.; Neelaveni, R. (Hrsg.): *Automotive Embedded Systems*. Cham: Springer International Publishing 2021, pp. 71–87
- [5] Verein Deutscher Ingenieure: VDI 4499 Blatt 2: Digitale Fabrik – Digitaler Fabrikbetrieb. Berlin: Beuth Verlag 2011
- [6] VDMA, ISW Uni Stuttgart, RIF Institut für Forschung und Transfer e.V. (Hrsg.): *Leitfaden Virtuelle Inbetriebnahme. Handlungsempfehlungen zum wirtschaftlichen Einstieg*. Stand: 2020. Internet: [www.vdma.org/c/document\\_library/get\\_file?uuid=66118d8d-68fc-55a7-a25b-a90ffe89f1f2&groupId=34570](http://www.vdma.org/c/document_library/get_file?uuid=66118d8d-68fc-55a7-a25b-a90ffe89f1f2&groupId=34570). Zugriff am 28.05.2025
- [7] Verein Deutscher Ingenieure: VDI/VDE 3693 Blatt 1: Virtuelle Inbetriebnahme – Modellarten und Begriffe. Berlin: Beuth Verlag 2023
- [8] Schamp, M.; Hoedt, S.; Claeys, A. et al.: Impact of a virtual twin on commissioning time and quality. *IFAC-PapersOnLine* 51 (2018) 11, pp. 1047–1052
- [9] van Merode, H.: CI/CD Concepts. In: van Merode, H. (ed.): *Continuous Integration (CI) and Continuous Delivery (CD). A Practical Guide to Designing and Developing Pipelines*. Berkeley, CA: Apress; Imprint Apress 2023, pp. 11–27
- [10] Verein Deutscher Ingenieure: VDI 3633 Blatt 8:2023–08: Simulation von Logistik-, Materialfluss- und Produktionssystemen – Maschinennahe Simulation – 3-D-Bewegungs- und Prozesssimulation. Berlin: Beuth Verlag 2023
- [11] Scheifele, C.: Plattform zur Echtzeit-Co-Simulation für die virtuelle Inbetriebnahme. Dissertation, Universität Stuttgart, 2019
- [12] Pott, A.; Dietz, T.: Programmierung und Zellsimulation. In: Pott, A.; Dietz, T. (Hrsg.): *Industrielle Robotersysteme. Entscheidungswissen für die Planung und Umsetzung wirtschaftlicher Roboterlösungen*. Wiesbaden: Springer Vieweg 2019, S. 65–77
- [13] Fechter, M.: Wandlungsfähige Roboter für die Automobilproduktion. In: Bauernhansl, T.; Fechter, M.; Dietz, T. (Hrsg.): *Entwicklung, Aufbau und Demonstration einer wandlungsfähigen (Fahrzeug-) Forschungsproduktion*. Heidelberg: Springer Vieweg 2020, S. 55–68
- [14] Kübler, K.; Jaensch, F.; Daniel, C. et al.: Nutzen von Digitalen Zwillingen und Virtueller Inbetriebnahme für den Maschinen- und Anlagenbau. In: Verl, A.; Röck, S.; Scheifele, C. (Hrsg.): *Echtzeitsimulation in der Produktionsautomatisierung*. Heidelberg: Springer 2024, S. 141–154
- [15] Pott, A.; Dietz, T. (Hrsg.): *Industrielle Robotersysteme. Entscheidungswissen für die Planung und Umsetzung wirtschaftlicher Roboterlösungen*. Wiesbaden: Springer Vieweg 2019

- [16] Pritschow, G.: Einführung in die Steuerungstechnik. München: Hanser 2006
- [17] ISO/IEC/IEEE 15288:2023-05: System- und Software-Engineering – System-Lebenszyklus-Prozesse. Berlin: Beuth Verlag 2023
- [18] Nibert, J.; Herniter, M.; Chambers, Z.: Model-Based System Design for MIL, SIL, and HIL. World Electric Vehicle Journal 5 (2012) 4, pp. 1121–1130
- [19] DIN 66025-1:1983-01: Programmaufbau für numerisch gesteuerte Arbeitsmaschinen – Allgemeines. Deutsche Fassung, Ausgabe 1983
- [20] DIN 66025-2:1988-09: Programmaufbau für numerisch gesteuerte Arbeitsmaschinen – Wegbedingungen und Zusatzfunktionen. Deutsche Fassung, Ausgabe 1988 [zurückgezogen]
- [21] DIN EN IEC 61131-3:2023-09 – Entwurf: Speicherprogrammierbare Steuerungen – Teil 3: Programmiersprachen (IEC 65B/1229/CDV:2023). Deutsche Fassung, Ausgabe 2023
- [22] DIN EN 61499-1:2014-09: Funktionsbausteine für industrielle Leitsysteme – Teil 1: Architektur (IEC 61499-1:2012). Deutsche Fassung, Ausgabe 2014

**Simon Nowinski, M.Sc.** 

*simon.nowinski@isw.uni-stuttgart.de*

*Tel. +49 711 / 685-82436*

**Lars Klingel, M.Sc.** 

**Daniel Littfinski, M.Sc.**

**Prof. Dr.-Ing. Alexander Verl** 

Universität Stuttgart

Institut für Steuerungstechnik der Werkzeugmaschinen

und Fertigungseinrichtungen (ISW)

Seidenstr. 36, 70174 Stuttgart

[www.isw.uni-stuttgart.de](http://www.isw.uni-stuttgart.de)

**Christian Eble, M.Sc.** 

Zimmer Systems GmbH

Am Glockenloch 2, 77866 Rheinau

[www.zimmer-group.com/de/](http://www.zimmer-group.com/de/)

## LIZENZ



Dieser Fachaufsatz steht unter der Lizenz Creative Commons  
Namensnennung 4.0 International (CC BY 4.0)