



Marc Schnierle, Jana Hönig und Sascha Röck

Zusammenfassung

Der Einsatz von X-in-the-Loop Simulationsmethoden (XiLS) ermöglicht im mechatronischen Entwicklungsprozess die virtuelle Erprobung und Inbetriebnahme von Automatisierungssystemen im Maschinen- und Anlagenbau. Die daraus resultierenden Anwendungsfelder sind bislang hinsichtlich der Interaktion zwischen Mensch, realer Umgebung und Simulation stark limitiert. Diese Einschränkungen sollen durch die Erweiterung der XiLS-Methodenreihe um die sogenannte Mixed-Reality-in-the-Loop Simulation (MRiLS) überwunden werden. Die MRiLS erweitert die Kopplung zwischen Steuerungsausprägung und Digitalem Zwilling unter Berücksichtigung der realen Umgebung und des Menschen durch moderne Visualisierungs- und Interaktionsmethoden der Mixed Reality. Diese Erweiterung eröffnet neue Anwendungspotenziale des Digitalen Zwillings über den gesamten Lebenszyklus einer Anlage hinweg.

16.1 Motivation

Der steigende Automatisierungsgrad gepaart mit immer kürzer werdenden Entwicklungszeiten erhöht die Anforderungen an den mechatronischen Entwicklungsprozess im Maschinen- und Anlagenbau. Die X-in-the-Loop Simulationsmethoden (XiLS) bilden einen Baustein, um diesen gewachsenen Anforderungen entsprechen zu können. Insbesondere die XiLS-Methoden zur virtuellen Erprobung und Inbetriebnahme von hochautomatisierten Maschinen und Anlagen ermöglichen durch die Parallelisierung von

M. Schnierle · J. Hönig · S. Röck (✉)

Virtual Automation Lab (VAL), Hochschule Esslingen, Esslingen, Deutschland

E-Mail: Sascha.Roeck@hs-esslingen.de

© Der/die Autor(en) 2024

A. Verl, S. Röck und C. Scheifele (Hrsg.),

Echtzeitsimulation in der Produktionsautomatisierung,

https://doi.org/10.1007/978-3-662-66217-5_16

Entwicklungsschritten und die frühzeitige Simulation von Störsituationen eine Verkürzung der Entwicklungsdauer und erfahren daher einen wachsenden Einsatz [1]. Dabei können die Testkonfigurationen Model-in-the-Loop Simulation (MiLS), Software-in-the-Loop Simulation (SiLS) und Hardware-in-the-Loop Simulation (HiLS) unterschieden werden [2], die jeweils durch die Kopplung einer Steuerungsausprägung mit einem Digitalen Zwilling der Anlage oder einer Anlagenkomponente charakterisiert sind. Bei der MiLS wird ein Modell der Steuerung, indem die Steuerungsfunktionen hinterlegt sind, gegen den Digitalen Zwilling der Anlage getestet. Als Erweiterung wird in einer SiLS die auf dem Modell der Steuerung aufbauende Steuerungssoftware eingebunden. Die HiLS bezieht darüber hinaus zusätzlich die Zielhardware der Steuerung ein, die in der Regel über einen industriellen Feldbus mit einem echtzeitfähigen Digitalen Zwilling gekoppelt ist. Während sich die Testkonfigurationen durch die Ausprägungsstufe von Steuerung und Digitalem Zwilling unterscheiden, ist die konzeptionelle Architektur kongruent.

In der bisherigen Anwendung der oben beschriebenen Testkonfigurationen lassen sich jedoch folgende Defizite identifizieren:

- **Exozentrische Visualisierung des Digitalen Zwillings**

Die Visualisierung des Digitalen Zwillings erfolgt überwiegend in einer zweidimensionalen und vom Betrachter unabhängigen (exozentrischen) Projektion dreidimensionaler Objekte auf einem Computerbildschirm. Durch diese Projektion gehen Tiefeninformationen für das stereoskopische Sehen des Menschen verloren und die räumliche Wahrnehmung wird stark beeinträchtigt. Damit verhindert die exozentrische Visualisierung die vollständige Integration des Menschen in den Simulationskreislauf sowie eine stufenlose visuelle Verschmelzung von Realität (z.B. Umgebung und Mensch) und Virtualität (Digitaler Zwilling).

- **Beschränkte intuitive Interaktion des Menschen**

Neben den Limitierungen der Visualisierung kann der Mensch und dessen Verhalten bislang nur sehr reduziert in die Simulation einbezogen werden, zum Beispiel durch Interaktionen mit Maus und Tastatur. Eine intuitive Interaktion und Exploration in der Simulationsszene sind bislang nicht möglich.

- **Fehlende Multiuser-Kollaboration**

XiLS-Anwendungen werden überwiegend in isolierten Einzelarbeitsplatzsystemen ausgeführt, die eine Zusammenarbeit mehrerer Benutzer in einer Simulationsszene ausschließen. Dies umfasst neben der Multiuser-Kollaboration in einem lokalen System auch die Zusammenarbeit über Standorte hinweg.

Zur Lösung der beschriebenen Defizite stellt dieser Beitrag die Methode der Mixed-Reality-in-the-Loop Simulation (MRiLS) als Erweiterung der XiLS-Methodenreihe vor (siehe Abb. 16.1). Die MRiLS vereint moderne Visualisierungsmethoden z. B. der

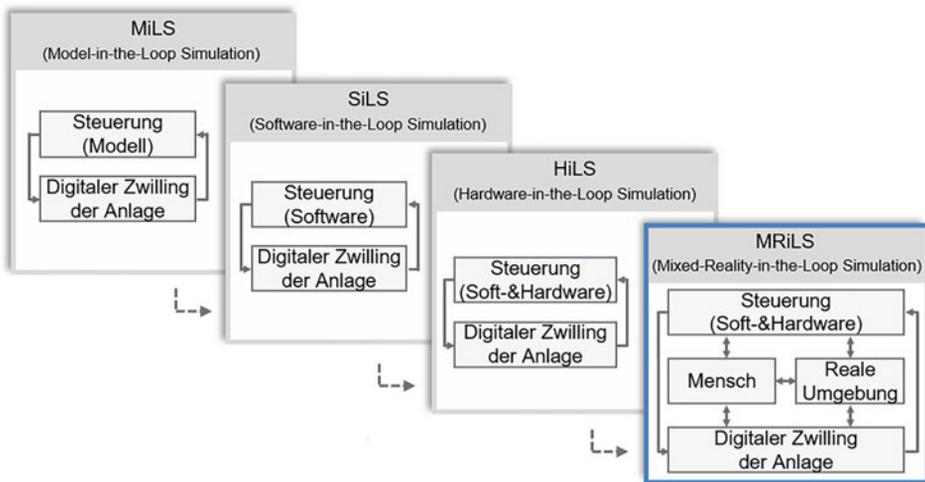


Abb. 16.1 Erweiterung der XiLS-Methodenreihe um die Mixed-Reality-in-the-Loop Simulation (MRiLS)

Augmented Reality (AR) oder Virtual Reality (VR) mit Interaktionsmethoden wie Gestensteuerung oder haptischem Feedback, um ein möglichst intensives Präsenzgefühl beim Nutzer in der virtuellen Szene zu erzielen. Damit wird eine stufenlose Kombination von Realität und Virtualität sowie die vollständige Integration des Menschen in den Simulationskreislauf ermöglicht.

Einführendes Beispiel

Nachfolgend ist das Konzept der MRiLS am Beispiel der Roboterprogrammierung für eine Sondermaschine dargestellt (siehe Abb. 16.2).

Im Beispiel sind die reale Steuerung und große Teile der realen Anlage um den Digitalen Zwilling eines Handhabungsroboters (zur deutlichen Erkennbarkeit in Abb. 16.2 grün eingefärbt) ergänzt. Der Digitale Zwilling wird durch die reale Steuerung betrieben und den Nutzern über Mixed Reality-Brillen als Erweiterung in der realen Umgebung vom Betrachter abhängig (egozentrisch) visualisiert. Die Nutzer nehmen den Digitalen Zwilling in der realen Umgebung verankert wahr, können Abstände räumlich präzise abschätzen, Kollisionen mit realen Anlagenkomponenten erkennen und mittels Gestensteuerung manipulieren. Durch die egozentrische Visualisierung und die intuitive Interaktion werden die Nutzer vollständig in den Simulationskreislauf der MRiLS integriert. Diese Eigenschaften ermöglichen Inbetriebnahme- oder Planungsprozesse im Kontext der realen Umgebung. Gegenüber der hier im Beispiel exemplarisch dargestellten Konfiguration der MRiLS mit einem hohen Anteil an realen Komponenten sind auch



Abb. 16.2 Konzept der Mixed-Reality-in-the-Loop Simulation am Beispiel einer Sondermaschine

Ausprägungsstufen mit einem höheren Anteil an virtuellen Komponenten bis hin zur vollständig virtuellen Szene möglich, die besonders für Anwendungen im Engineering, der Mitarbeiterschulung oder dem Vertrieb geeignet sind.

Immersion und Präsenzgefühl

Eine wichtige Basis für die skizzierte Integration des Menschen in den XiLS-Kreislauf ist die Immersion und das resultierende Präsenzgefühl. Die Immersion ist eine objektiv bewertbare Fähigkeit eines Systems, dem Nutzer eine umfassende, umgebende und lebendige Illusion der Realität zu vermitteln [3, 4]. Ein hoher Grad an Immersion wird erreicht, wenn dem Nutzer die Umgebung basierend auf seinem Standort und seiner Orientierung realitätsnah präsentiert wird und synthetische Stimuli für alle relevanten Sinne als Reaktion auf seine Position und Aktionen bereitgestellt werden [4]. Die Präsenz oder das Präsenzgefühl ist die menschliche Reaktion auf diese Immersion und beschreibt den subjektiven Bewusstseinszustand und das psychologische Gefühl, sich tatsächlich in der virtuellen Umgebung zu befinden [3, 5]. Ein System mit definiertem Immersionsgrad ruft bei Nutzern unterschiedliche subjektive Intensitäten des Präsenzgefühls hervor [6]. Das Präsenzgefühl umfasst neben der Ortsillusion (Gefühl an einem real erscheinenden Ort zu sein) und der Plausibilitätsillusion (Umgebung reagiert so, wie der Nutzer es erwartet) auch die Repräsentation des eigenen Körpers als Überschneidung dieser beiden Illusionen [7]. Die Repräsentation des eigenen Körpers hat dabei direkten Einfluss auf die Ortsillusion (Nutzer wird an dem Ort dargestellt, an dem er sich auch wahrnimmt) und die Plausibilitätsillusion (virtuelle Repräsentation bewegt sich synchron zu eigenen Gliedmaßen).

Leitziele für die MRiLS

Zur Erreichung eines hohen Präsenzgefühls des Menschen im Simulationskreislauf strebt die MRiLS folgende Leitziele an (siehe Abb. 16.3):

- **Egozentrische realzeitsynchrone Visualisierung**

Zur Steigerung des Immersionsgrads soll die geometrische Repräsentation des Digitalen Zwillings egozentrisch und auf die Umgebung zeitsynchronisiert visualisiert werden. Diese nutzerabhängige Darstellung des Digitalen Zwillings erhöht das räumliche Präsenzgefühl, da die visualisierte Perspektive im Bezug zur Blickrichtung des Nutzers kontinuierlich angepasst wird. Dies beinhaltet eine aus Nutzersicht stabile Verankerung der virtuellen Objekte in Bezug auf den Ort und die Zeit sowohl an statischen als auch an dynamischen Objekten der realen Umgebung und eine perspektivisch korrekte Darstellung der virtuellen Objekte in der realen Umgebung.

- **Intuitive multimodale Mensch-Modell-Interaktion**

Für ein hohes Präsenzgefühl des Nutzers in einer MRiLS muss eine intuitive und multimodale Interaktion mit dem Digitalen Zwilling erreicht werden. Dafür müssen alle für das Szenario relevanten menschlichen Sinne durch Interaktionseingaben (z.B. Controller, oder Gesten- & Sprachsteuerung) sowie Interaktionsausgaben (z.B. auditives oder haptisches Feedback) angesprochen werden. Der entstehende Interaktionskreislauf erfordert zudem intuitive Interaktionsmechanismen im Digitalen Zwilling, um die plausible Reaktion des Digitalen Zwillings auf Nutzerinteraktionen sicherzustellen.

- **Multiuser-Kollaboration**

Die informationstechnische Vernetzung mehrerer Nutzer soll die Zusammenarbeit in einer gemeinsamen Simulationsszene erlauben. Der Digitale Zwilling muss für alle Nutzer in einem gemeinsamen Koordinatensystem positioniert und über verschiedene Endgeräte angezeigt werden. Bei standortübergreifenden Multiuser-Kollaborationen müssen Nutzer zudem wechselseitig durch Avatare visualisiert werden.



Abb. 16.3 Leitziele der Mixed-Reality-in-the-Loop Simulation

In den nachfolgenden Abschnitten werden diese drei Leitziele sowie Aspekte zu deren Erreichung detaillierter beleuchtet.

16.2 Egozentrische realzeitsynchrone Visualisierung

In konventionellen Anwendungen der XiLS werden wie eingangs dargestellt die dreidimensionalen Geometrieobjekte der Digitalen Zwillinge exozentrisch auf Computerbildschirme projiziert. Im Gegensatz hierzu ermöglichen moderne Visualisierungsmethoden der Mixed Reality, wie Augmented Reality oder Virtual Reality, eine dreidimensionale egozentrische Visualisierung der Digitalen Zwillinge in der realen Umgebung (siehe Abb. 16.4).

Die stufenlose Kombination der Realität und Virtualität zu einem Realitäts-Virtualitäts-Kontinuum mittels moderner Visualisierungstechnologien kann entsprechend Abb. 16.5 in vier Hauptkategorien der Mixed Reality (MR) gegliedert werden.

Die reale Umgebung (RE – Real Environment) enthält ausschließlich physisch existierende Elemente, während die virtuelle Realität (VR – Virtual Reality) vollständig aus computergenerierten Elementen aufgebaut ist. Zwischen diesen beiden Extremen liegen zwei Mischformen, die entsprechend dem Anteil von Realität und Virtualität unterschieden werden. Bei der erweiterten Realität (AR – Augmented Reality) überwiegen die realen Komponenten (z. B. virtueller Roboter in realer Fertigungszelle), während bei der erweiterten Virtualität (AV – Augmented Virtuality) die virtuellen Komponenten dominieren (z. B. reale Hände im vollständig virtuellen Raum).

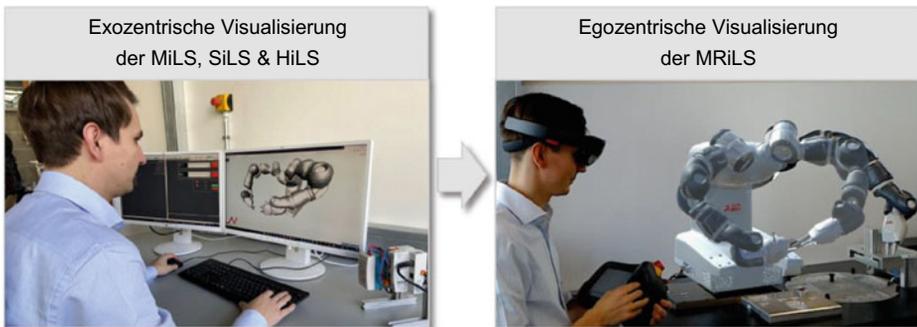


Abb. 16.4 Erweiterung der exozentrischen zweidimensional projizierten Visualisierung zu einer egozentrischen dreidimensionalen Perspektive bei der MRiLS

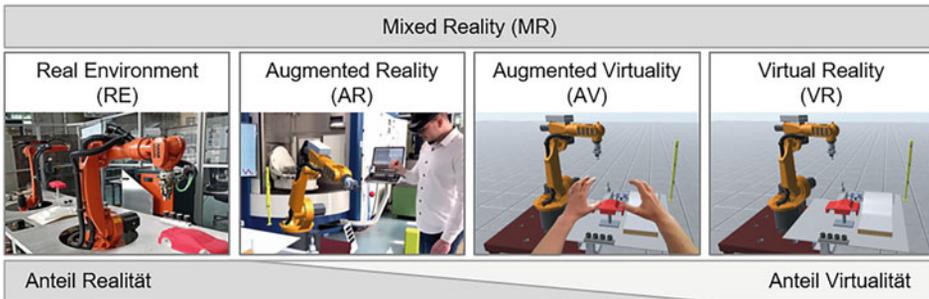


Abb. 16.5 Realitäts-Virtualitäts-Kontinuum in Anlehnung an Milgram et al. [8]

Ausgabegeräte zur egozentrischen Visualisierung

Im Vergleich zur konventionellen Visualisierung auf Computerbildschirmen erfolgt die visuelle Ausgabe in der MRiLS in unterschiedlichen Konfigurationen des Realitäts-Virtualitäts-Kontinuums. Hinsichtlich der dafür einsetzbaren Endgeräte kann auf Entwicklungen der Gaming-Industrie in den Bereichen der Head-Mounted-Displays (HMD) und der Handheld-Geräte zurückgegriffen werden. HMD-Endgeräte wie beispielsweise AR- und VR-Brillen werden unmittelbar vor dem Auge getragen (siehe Abb. 16.6).

Die Grundlage der immersiven Visualisierung bildet das stereoskopische Sehvermögen des Menschen. Ein betrachtetes Objekt wird von jedem Auge mit einem geringfügig unterschiedlichen Winkel wahrgenommen und das menschliche Gehirn verarbeitet diese beiden perspektivischen Seheindrücke zu einem dreidimensionalen Bild. Aus der Kombination dieser perspektivischen Seheindrücke resultiert das räumliche Sehen sowie die Tiefenwahrnehmung des Menschen [9]. Dieses Wirkprinzip wird in HMD-Endgeräten mit stereoskopischer Visualisierung durch separate Bildanzeigen pro Auge ausgenutzt. Neben den AR- und VR-Brillen können auch Handheld-Geräte wie Tablets oder Smartphones zur egozentrischen Visualisierung zum Einsatz kommen, bei denen jedoch durch die monoskopische Visualisierung keine Tiefeninformationen dargestellt werden können.

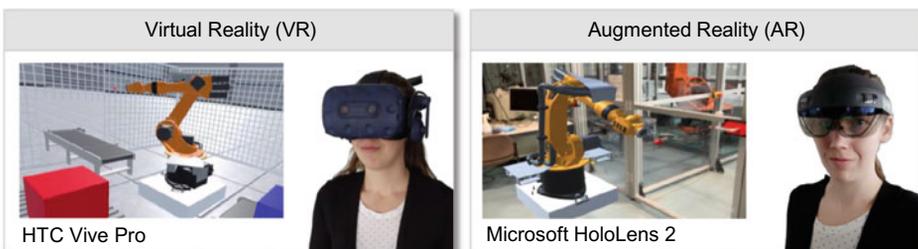


Abb. 16.6 Ausgewählte Head-Mounted-Displays für Virtual Reality und Augmented Reality

Neben der Unterscheidung zwischen stereoskopischer und monoskopischer Visualisierung ist bei Augmented Reality-Endgeräten zusätzlich zwischen dem Optical-See-Through Prinzip und dem Video-See-Through Prinzip zu differenzieren. Bei Optical-See-Through-Endgeräten (z. B. Microsoft HoloLens) betrachtet der Nutzer die reale Umgebung durch eine transparente Projektionsfläche. Die virtuellen Objekte werden auf die Projektionsfläche und somit in das reale Sichtfeld des Nutzers projiziert. Bei Video-See-Through-Endgeräten (z. B. Tablet) wird die Außenwelt über Kameras aufgenommen, mit virtuellen Objekten ergänzt und dem Nutzer über die Endgerätedisplays angezeigt. Die Wahrnehmung der realen Welt erfolgt in diesem Fall nicht mit der menschlichen Sehauflösung wie bei Optical-See-Through-Endgeräten, sondern mit der Kamera- bzw. Displayauflösung.

Egozentrische Positionierung der virtuellen Komponenten

Die egozentrische Visualisierung in einer MRiLS muss sich im Gegensatz zur exozentrischen Visualisierung auf eine Änderung der Blickrichtung des Nutzers adaptieren, um die korrekte Position und Perspektive des Digitalen Zwillings aus Sicht des Nutzers anzeigen zu können (siehe Abb. 16.7).

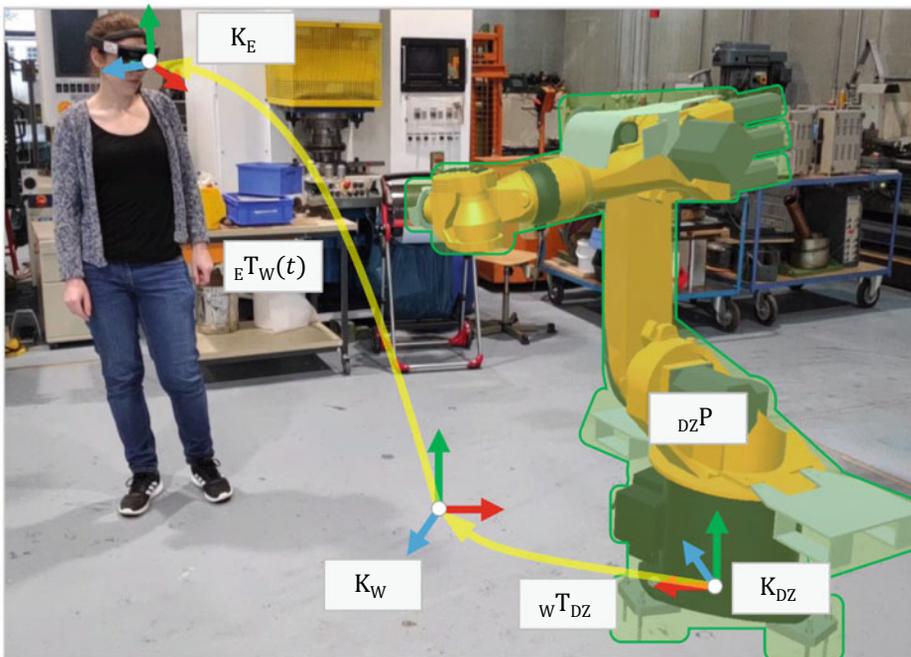


Abb. 16.7 Egozentrische Visualisierung virtueller Anlagenkomponenten in einer MRiLS

Die dafür zu visualisierenden Geometrieobjekte (engl. Meshes) des Digitalen Zwillings können als Matrix ${}_{DZ}P$ mit Geometrieknoten (engl. Vertices) in homogenen Koordinaten im Koordinatensystem K_{DZ} des Digitalen Zwillings beschrieben werden:

$${}_{DZ}P = [p_1, \dots, p_n] \text{ mit } p_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}. \quad (16.1)$$

Um die Meshes ${}_{DZ}P$ des Digitalen Zwillings im sich dynamisch bewegenden Sichtfeld des Nutzers perspektivisch korrekt darzustellen, muss ${}_{DZ}P$ in jedem Visualisierungstakt aus dem Koordinatensystem K_{DZ} des Digitalen Zwillings in das Koordinatensystem K_E des bewegten Endgeräts transformiert werden.

Für die räumliche Transformation der Meshes ${}_{DZ}P$ von K_{DZ} nach K_E ist die homogene Transformationsmatrix ${}_{E}T_{DZ} \in \mathbb{R}^{4 \times 4}$ erforderlich. Diese setzt sich aus der Transformation ${}_{W}T_{DZ}$ vom Objektkoordinatensystem K_{DZ} in das Weltkoordinatensystem K_W und der zeitvarianten Transformation ${}_{E}T_W(t)$ von K_W in das Endgerät-Koordinatensystem K_E zusammen (vgl. Abb. 16.7):

$${}_{E}T_{DZ}(t) = {}_{E}T_W(t) \cdot {}_{W}T_{DZ} = \begin{bmatrix} {}_{E}R_{DZ} & {}_{E}\vec{p}_{DZ} \\ [0,0,0] & [1] \end{bmatrix}. \quad (16.2)$$

Die Rotationsmatrix ${}_{E}R_{DZ} \in \mathbb{R}^{3 \times 3}$ beschreibt die Rotation und der Ortsvektor ${}_{E}\vec{p}_{DZ} \in \mathbb{R}^{3 \times 1}$ die Position von K_{DZ} in K_E . Die Transformationsmatrizen ${}_{E}T_W(t)$ und damit ${}_{E}T_{DZ}(t)$ sind zeitabhängig, da der Nutzer seine Position und Orientierung im Raum und somit die Position und Orientierung zum ortsfesten Weltkoordinatensystem K_W fortlaufend ändert.

Mithilfe von ${}_{E}T_{DZ}(t)$ können die Meshes ${}_{DZ}P$ des Digitalen Zwillings in das Koordinatensystem K_E des Endgeräts transformiert werden:

$${}_{E}P(t) = {}_{E}T_{DZ}(t) \cdot {}_{DZ}P = {}_{E}T_W(t) \cdot {}_{W}T_{DZ} \cdot {}_{DZ}P. \quad (16.3)$$

Für die kontinuierliche Bestimmung der zeitabhängigen Transformationsmatrix ${}_{E}T_W(t)$ wird die Position und Orientierung des Endgeräts über Trackingsysteme ermittelt, die sich hinsichtlich der Trackingmethoden in Outside-In und Inside-Out Systeme klassifizieren lassen.

Bei Outside-In Trackingsystemen befindet sich die Tracking-Logik nicht im Endgerät, sondern in einem externen System (z. B. im Raum installiertes Kamerasystem). Das getrackte passive Objekt hat dabei keine unmittelbare Kenntnis über seine Position [9]. Diese Art des Trackings erfordert zwingend eine gerätetechnische Ausstattung im Raum und eignet sich daher insbesondere für stationäre Aufbauten.



Abb. 16.8 Markerbasiertes und markerloses Tracking bei Inside-Out-Verfahren

Im Gegensatz hierzu stehen Methoden des Inside-Out Trackings, bei denen das Endgerät seine Positionsdaten über Referenzen in der Umgebung selbst ermittelt [9, 10]. Dabei ist zwischen markerbasierten und markerlosen Trackingverfahren zu differenzieren. Bei markerbasierten Verfahren können sowohl aktive als auch passive Marker eingesetzt werden. Ein Beispiel für aktive Marker sind Infrarot-basierte Systeme (siehe Abb. 16.8 links), bei denen Basisstationen als Infrarot-Marker fungieren und paarweise Laserstrahlen aussenden, die auf die Sensorik des Headsets treffen (z. B. HTC Lighthouse Technologie). Aus der zeitlichen Differenz der auftreffenden Laserstrahlen ermittelt das Endgerät seine Position im Raum.

Im Gegensatz hierzu werden passive Marker wie Muster- oder QR-Codes bildverarbeitend erkannt und als Referenzpunkt genutzt (siehe Abb. 16.8 Mitte). Während markerbasierte Verfahren aktive oder passive Marker im Raum benötigen, werden beim markerlosen Tracking (siehe Abb. 16.8 rechts) Merkmale der Umgebung im Endgerät zur Positionsbestimmung genutzt (z. B. Microsoft HoloLens).

Datengetriebene Bewegung virtueller Komponenten

Die egozentrische Visualisierung und die dafür notwendige kontinuierliche Transformation der statischen Meshes $_{DZ}P$ in das Endgeräte-Koordinatensystem K_E bilden die Grundlage für die Visualisierung einer MRiLS. Über diese Basis hinaus ist die Integration von industriellen Steuerungsdaten in jedem Visualisierungstakt der MRiLS von zentraler Bedeutung. Die MRiLS soll Positionsdaten von industriellen Steuerungssystemen mit den kinematischen Freiheitsgraden des Digitalen Zwillings verknüpfen und damit Anwendungen, wie die eingangs dargestellte Roboterprogrammierung, ermöglichen. Die datengetriebene Transformation der Geometrieobjekte innerhalb des Koordinatensystems K_{DZ} führt dabei zu zeitvarianten Meshes $_{DZ}P(t)$ und es gilt:

$$\begin{aligned} {}_{DZ}P(t) &= [{}_{DZ}T_{O1}(t) \cdot {}_{O1}P, {}_{DZ}T_{O2}(t) \cdot {}_{O2}P, \dots, {}_{DZ}T_{On}(t) \cdot {}_{On}P] \\ {}_E P(t) &= {}_E T_W(t) \cdot {}_W T_{DZ} \cdot {}_{DZ}P(t) \end{aligned} \quad (16.4)$$

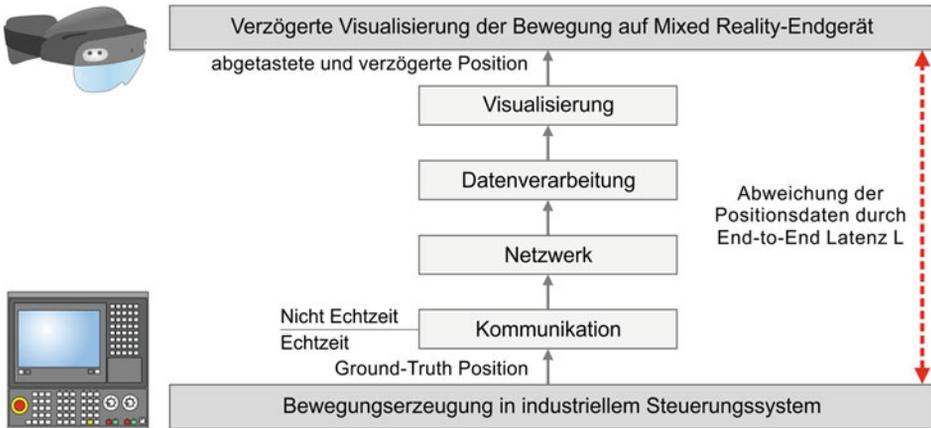


Abb. 16.9 Übertragungsweg von dem industriellen Steuerungssystem zum Mixed Reality-Endgerät in einer MRiLS

${}_{01}P, {}_{02}P, \dots, {}_{0n}P$ stellen die einzelnen Geometrieknoten der bewegten Meshes innerhalb des Digitalen Zwillings dar, die jeweils aus dem körperfesten Objektkoordinatensystem K_{O_i} in das Koordinatensystem des Digitalen Zwillings K_{DZ} transformiert werden. Der kinematische Freiheitsgrad des jeweiligen Objekts führt zu zeitvarianten Transformationsmatrizen ${}_{DZ}T_{O_i}(t)$ und damit zu zeitvarianten Meshes ${}_{DZ}P(t)$.

Positionierungsfehler durch Latenz- und Abtastvorgänge

Die in ${}_{DZ}P(t)$ einfließenden Steuerungsdaten unterliegen auf dem Übertragungsweg vom industriellen Steuerungssystem zum Mixed Reality-Endgerät verschiedenen Latenz- und Abtastvorgängen, die zu einer Abweichung zwischen visualisierten und auf dem Steuerungssystem erzeugten Positionsdaten führen (siehe Abb. 16.9).

Die Abtastraten und Netzwerklatenzen der Kommunikation, Datenverarbeitungszeiten sowie die Visualisierungslatenz des Endgeräts resultieren in einer sogenannten End-to-End Latenz L zwischen Steuerungssystem und Mixed Reality-Endgerät. Die End-to-End Latenz eines Mixed Reality-Systems bezeichnet die Zeitspanne von der Entstehung einer Information im Steuerungssystem bis zu deren Visualisierung auf dem Endgerät [11]. Durch schwankende Netzwerkauslastungen, dynamische Verarbeitungszeiten und Schwebungseffekte durch diskrete Mehrfachabtastungen ist in einer MRiLS von zeitvarianten End-to-End Latenzen auszugehen.

Abb. 16.10 zeigt die durch die End-to-End Latenz auftretende Positionsabweichung zwischen den visualisierten Positionsdaten auf dem Endgerät und der tatsächlichen Position eines Roboters (Ist-Positionen vom Steuerungssystem) am Anwendungsbeispiel einer MRiLS in Augmented Reality-Konfiguration. Das Szenario erweitert eine reale Roboterkinematik um einen virtuellen Greifer (zur besseren Erkennbarkeit grün eingefärbt und

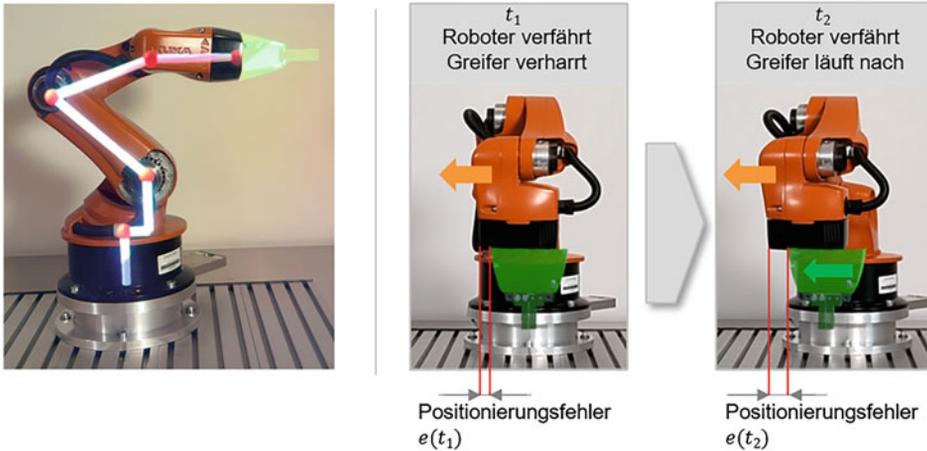


Abb. 16.10 Positionierungsfehler virtueller Anlagenkomponenten durch Latenz- und Abtasteffekte

durch das virtuelle Kinematikmodell des Roboters ergänzt), der mithilfe eines markerbasierten passiven Trackingsystems und einer kinematischen Vorwärtstransformation am Endeffektor des realen Roboters positioniert wird. Bei einer Achsbewegung der Roboterkinematik entstehen Positionierungsfehler $e(t)$ zwischen realer und virtueller Kinematik (siehe Abb. 16.10, Zeitpunkte t_1 und t_2), da der virtuelle Greifer durch die End-to-End Latenz der Positionsänderung der realen Roboterkinematik nur verzögert folgen kann.

Diese Positionsabweichungen schränken zum einen das Präsenzgefühl des Nutzers ein und limitieren zum anderen die Anwendbarkeit der MRiLS für industrielle Anwendungsszenarien in der Produktionsautomatisierung. Aus diesem Grund ist eine Kompensationsmethode zur Reduktion des auftretenden Positionierungsfehlers $e(t)$ eine notwendige Bedingung für den praxisorientierten Einsatz der MRiLS. Eine ausführliche Beschreibung dieser nachfolgend in Kurzform dargestellten Methode kann in [12] nachgelesen werden.

Kompensationsmethode für die Positionierungsfehler

Das Prinzip zur Reduktion der auftretenden Positionierungsfehler ist an einem exemplarischen Positionsverlauf einer Roboterachse in Abb. 16.11 dargestellt. Der zum Zeitpunkt t_k auf der Steuerung erzeugte Positionswert $s(t_k)$ wird um die Latenz L_k verzögert auf dem Endgerät zum Zeitpunkt $t_{k,L} = t_k + L_k$ visualisiert. Der reale Positionswert auf der Steuerung ist jedoch zu diesem Zeitpunkt bereits $s(t_{k,L})$ und dementsprechend ergibt sich der Positionierungsfehler zu:

$$e(t_{k,L}) = s(t_{k,L}) - s(t_k). \quad (16.5)$$

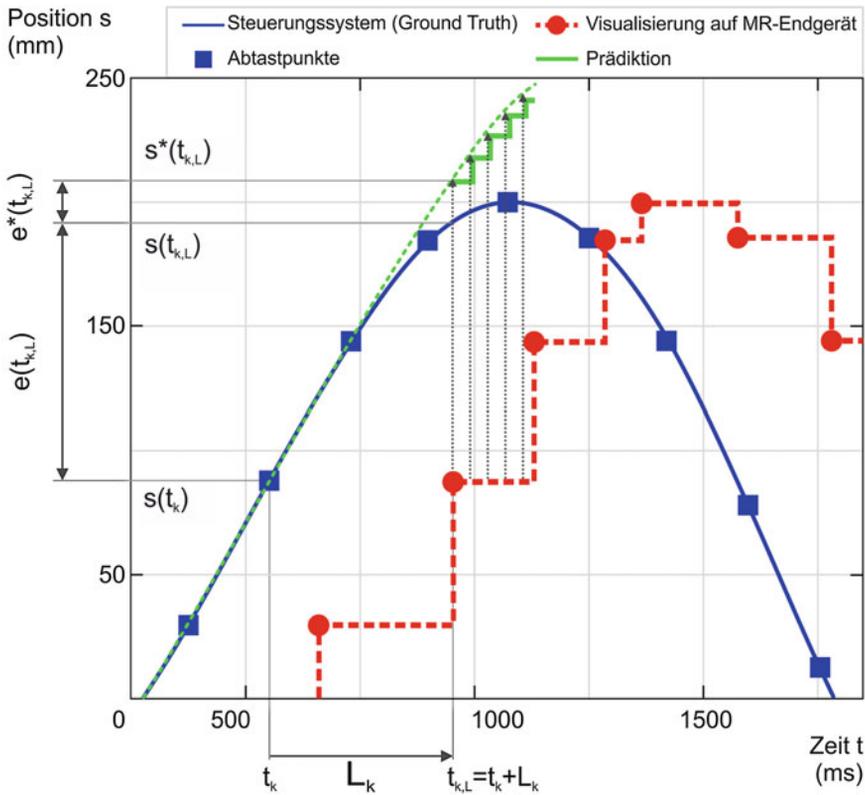


Abb. 16.11 Prädiktionsbasierte Kompensation von Positionierungsfehlern in einer MRiLS

Unter Einsatz der aus der Steuerung abgetasteten Positionsdaten $[s(t_0), \dots, s(t_k)]$ und den zugehörigen End-to-End Latenzen $[L_0, \dots, L_k]$ kann mittels einer Prädiktionsfunktion ein Schätzwert $s^*(t_{k,L})$ der Position $s(t_{k,L})$ berechnet werden. Der bleibende Positionierungsfehler $e^*(t_{k,L})$ soll dabei betragsmäßig kleiner als der ursprüngliche Positionierungsfehler $e(t_{k,L})$ sein:

$$|e^*(t_{k,L})| = |s(t_{k,L}) - s^*(t_{k,L})| < |s(t_{k,L}) - s(t_k)| = |e(t_{k,L})|. \quad (16.6)$$

Für die Berechnung von $s^*(t_{k,L})$ und die damit zu erzielende Reduzierung des Positionierungsfehlers ist sowohl eine Schätzung der End-to-End Latenz L_k^* als auch eine Prädiktionsfunktion notwendig.

Schätzung der End-to-End Latenz

Die Schätzung der End-to-End Latenz muss für jeden Visualisierungstakt des MR-Endgeräts zur Laufzeit der MRiLS erfolgen und im Kontext der industriellen Einsatzszenarien ohne zusätzliche Messhardware durchführbar sein.

Die End-to-End Latenz L_k^* wird dazu in zwei Latenzanteile aufgeteilt:

$$L_k^* = L_{app,k}^* + L_{visu,k}^* \quad (16.7)$$

- Die Applikationslatenz $L_{app,k}^*$ beschreibt die Zeitspanne von der Informationsentstehung im Steuerungssystem bis zur Verarbeitung auf dem MR-Endgerät (vor dem Start des Renderingauftrags in der Grafikpipeline).
- Die Visualisierungslatenz $L_{visu,k}^*$ beschreibt die Zeitspanne vom Start des Renderingauftrags in der Grafikpipeline des MR-Endgeräts bis zur für den Nutzer sichtbaren Visualisierung auf dem Display des MR-Endgeräts.

Schätzung der Applikationslatenz $L_{app,k}^*$ durch Uhrensynchronisation

Die dem Abtastzeitpunkt t_k zugehörige Applikationslatenz $L_{app,k}^*$ kann durch Subtraktion des Zeitstempels $C_1(t_k)$ der Steuerung von dem verzögerten Zeitstempel $C_2(t_k + L_{app,k})$ des MR-Endgeräts geschätzt werden:

$$L_{app,k}^* = C_2(t_k + L_{app,k}) - C_1(t_k). \quad (16.8)$$

Dazu wird $C_1(t_k)$ anhand der Systemuhr C_1 auf der Steuerung erstellt und mit dem Datenpaket der Positionsdaten an das MR-Endgerät übertragen. $C_2(t_k + L_{app,k})$ hingegen wird anhand der Systemuhr C_2 nach Verarbeitung der Positionsdaten auf dem MR-Endgerät ermittelt. Die Applikationslatenz wird anschließend auf dem MR-Endgerät nach Gl. (16.8) berechnet. Dies jedoch setzt eine gemeinsame Zeitbasis für die beiden Zeitstempel voraus, was die Synchronisation der Systemuhren erforderlich macht.

Für die Synchronisation werden die Systemuhren C_1 und C_2 jeweils mit einem linearen Uhrenmodell $C_i = \gamma_i \cdot t + \Theta_i$ über eine gemeinsame Referenzzeit abgebildet (siehe Abb. 16.12).

Die sich zwischen Steuerungssystem und MR-Endgerät unterscheidenden Uhrenparameter γ_i und Θ_i führen zu einem unbekanntem Uhrenoffset $\Phi(t) = C_2(t) - C_1(t)$. Da sich das MR-Endgerät auf die stabile Zeitbasis in der Echtzeitebene der Steuerung synchronisieren soll, kann $\gamma_1 = 1$ und $\Theta_1 = 0$ gesetzt werden [13]. Der Uhrenoffset lässt sich unter dieser Vorgabe mit den unbekanntem Uhrenparametern γ_2 und Θ_2 ausdrücken:

$$\Phi(t) = C_2(t) - C_1(t) = (\gamma_2 - \gamma_1) \cdot t + \Theta_2 - \Theta_1 = (\gamma_2 - 1) \cdot t + \Theta_2. \quad (16.9)$$

Für die Bestimmung von γ_2 und Θ_2 können Ansätze aus dem Bereich der Synchronisation von physikalischen Uhren in verteilten Systemen adaptiert werden [14], die auf

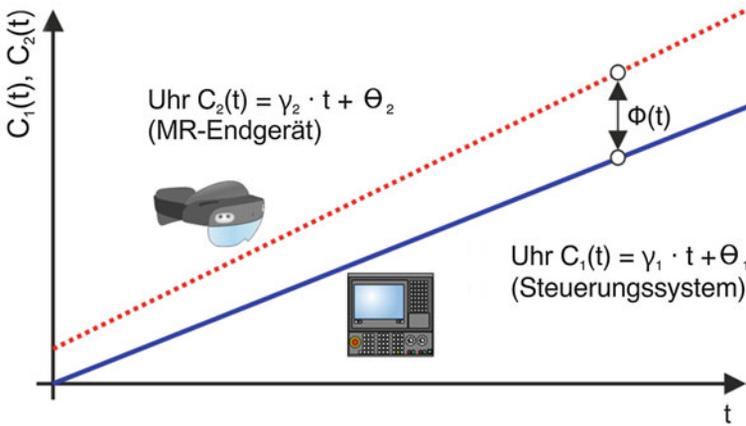


Abb. 16.12 Lineares Uhrenmodell für MR-Endgerät und Steuerungssystem mit Uhrenoffset Φ

dem Algorithmus nach Christian [15] aufbauen und zum Beispiel beim Network Time Protocol (NTP, RFC 958) oder Precision Time Protocol (PTP, IEEE 1588) zum Einsatz kommen. Zur Synchronisierung auf der Anwendungsebene des MR-Endgeräts werden diese Grundprinzipien übernommen und in eine Client-Server-Kommunikationsabfolge zwischen Steuerungssystem und MR-Endgerät überführt (siehe Abb. 16.13). Das MR-Endgerät sendet dazu an einen Synchronisationsmaster (z. B. Feldbusteilnehmer auf Echtzeitebene der Steuerung) zum Zeitstempel $C_2^r(t_{i,1}^{sync})$ eine Synchronisierungsanfrage, die beim Master zum Zeitstempel $C_1^r(t_{i,2}^{sync})$ eintrifft und nach einer Verarbeitungszeit mit dem Zeitstempel $C_1^r(t_{i,3}^{sync})$ beantwortet wird. Die Antwort trifft zum Zeitstempel $C_2^r(t_{i,4}^{sync})$ schließlich beim MR-Endgerät ein.

Unter Annahme symmetrischer Paketlaufzeiten $\Delta t_{i,12}^{sync} = \Delta t_{i,34}^{sync}$ kann der Uhrenoffset Φ_i^* in jeder Synchronisationssequenz mit folgendem Zusammenhang bestimmt werden (Details zur Herleitung siehe [16]):

$$\Phi_i^* = \Phi^*(t_{i,4}^{sync}) = \frac{C_2^r(t_{i,1}^{sync}) - C_1^r(t_{i,2}^{sync})}{2} + \frac{C_2^r(t_{i,4}^{sync}) - C_1^r(t_{i,3}^{sync})}{2}. \quad (16.10)$$

Zur Identifikation der Uhren-Parameter γ_2 und Θ_2 muss die Bedingung $\Phi_i^* - \Phi \stackrel{!}{=} 0$ in jeder Synchronisationssequenz erfüllt sein. Das nach mehreren Synchronisationssequenzen daraus entstehende überbestimmte Gleichungssystem wird mittels eines Least Squares Ansatzes in eine lineare Normalgleichung überführt, deren Lösung eine gute Näherung der Uhren-Parameter liefert. Nach erfolgter Parameteridentifikation kann die Systemuhr C_2 anhand des Uhrenmodells auf dem MR-Endgerät berechnet werden. Da C_2

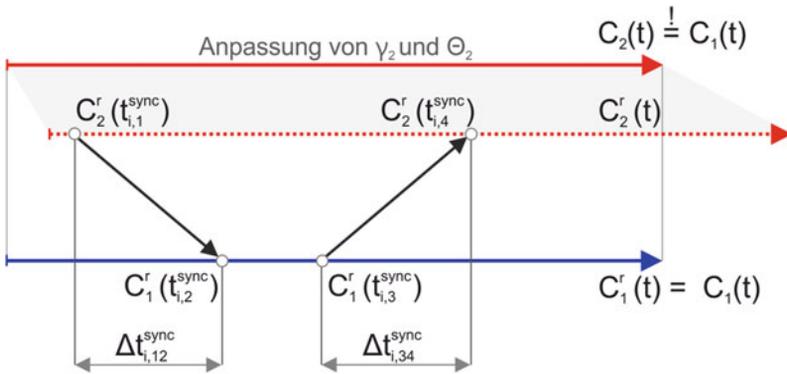


Abb. 16.13 Synchronisationssequenz zur Bestimmung des Uhrenoffsets Φ^*

nun näherungsweise mit der Systemuhr C_1 des Steuerungssystems übereinstimmt, können deren Zeitstempel für die Schätzung von $L_{app,k}^*$ nach Gl. (16.8) herangezogen werden.

Schätzung der Visualisierungslatenz $L_{visu,k}^*$ durch Endgeräte-Kalibrierung mit Visible Light Communication

Zur Bestimmung der Visualisierungslatenz wird eine Messmethode nach dem Prinzip der Visible Light Communication (VLC) verwendet (siehe Abb. 16.14).

Die Messmethode zeigt für jeden Steuerungstakt den Zeitstempel des Steuerungssystems $C_1(t_{VLC})$ codiert in einer LED-Matrix an, während das nach der oben beschriebenen Methode über einen Feldbusteilnehmer auf die Steuerung synchronisierte MR-Endgerät

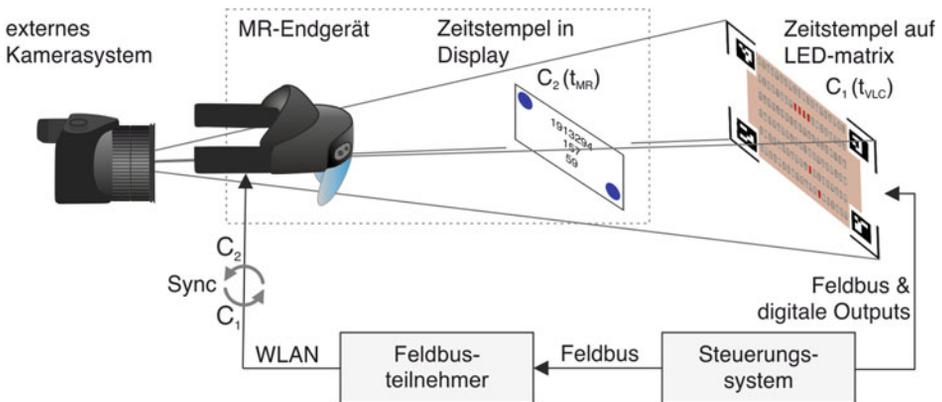


Abb. 16.14 Messmethode zur Bestimmung der Visualisierungslatenz eines Optical See-Through MR-Endgeräts nach dem Prinzip der Visible Light Communication (VLC)

jeweils seinen beim Start des Renderingauftrags gesetzten Zeitstempel $C_2(t_{MR})$ als virtuellen Inhalt im eigenen Display darstellt. Über ein externes Kamerasystem können beide Zeitstempel anschließend bildverarbeitend ausgewertet werden. Der Zeitstempel $C_1(t_{VLC})$ wird vom Kamerasystem (bei einer Bildfrequenz von 240 Hz und einer Belichtungszeit von ca. 4.2 ms) nahezu unmittelbar erfasst, während der Zeitstempel $C_2(t_{MR})$ erst nach dessen Verarbeitung in der Rendering-Pipeline im Kamerabild erscheint. Die Differenz beider Zeitstempel bildet näherungsweise die Visualisierungslatenz ab, wobei die Belichtungszeit des Kamerasystems dabei deutlich kleiner sein muss als die zu erwartende Visualisierungslatenz, die auf gängigen MR-Geräten wie bspw. der Microsoft HoloLens 2 Industrial in der Größenordnung um 50 ms liegt. Für einen Abtastzeitpunkt t_k (vgl. Abb. 16.11) wird der vom MR-Endgerät visualisierte Zeitstempel zum Zeitpunkt $t_{MR} = t_k + L_{app,k}$ gesetzt, während das Steuerungssystem einen Zeitstempel auf der LED-Matrix darstellt, der dem Zeitpunkt $t_{VLC} = t_k + L_k^*$ zugehört. Die Visualisierungslatenz $L_{visu,k}^*$ kann anschließend entsprechend der auswertbaren Zeitstempel berechnet werden:

$$L_{visu,k}^* = C_1(t_{VLC}) - C_2(t_{MR}) = C_1(t_k + L_k^*) - C_2(t_k + L_{app,k}). \quad (16.11)$$

Messergebnisse auf Basis dieser Methode zeigen eine starke Korrelation von $L_{visu,k}^*$ zu der Renderingzeit Δt_{render} , die in der Anwendungsebene des MR-Endgeräts über die Grafikbibliothek softwaretechnisch ermittelbar ist. Zentraler Einflussfaktor auf Δt_{render} ist dabei die Belastung der Renderingpipeline, die von der Anzahl der zu visualisierenden Geometrieknoten der Meshes DZP sowie von der Komplexität des Szenengraphs abhängig ist. Anhand zahlreicher Messungen konnte herausgefunden werden, dass bis zur Belastungsgrenze der Renderingpipeline die Renderingzeit und die Visualisierungslatenz bei gleichbleibender Anzahl an Geometrieknoten nahezu konstant bleiben [12]. $L_{visu,k}^*$ kann damit als zeitinvariant und von t_k unabhängig angenommen werden und wird zu L_{visu}^* . Eine mitlaufende Auswertung von Δt_{render} ist auf dem MR-Endgerät ohne zusätzliche Hardware durchführbar, durch die aufgrund der starken Korrelation auf die Visualisierungslatenz geschlossen werden kann. Die Korrelation zwischen Renderingzeit und Visualisierungslatenz wird für das MR-Endgerät einmalig mittels Visible Light Communication gemessen und auf dem MR-Endgerät in Form einer Kennlinie (L_{visu}^* über Δt_{render}) hinterlegt und steht somit für die Berechnung der End-to-End Latenz auf dem MR-Endgerät zur Verfügung.

Mit den Schätzungen der Applikationslatenz $L_{app,k}^*$ und der konstanten Visualisierungslatenz L_{visu}^* kann die End-to-End Latenz L_k^* zwischen dem industriellen Steuerungssystem und der Visualisierung auf dem MR-Endgerät nach Gl. 16.7 berechnet und der im Folgenden beschriebenen Prädiktionsfunktion zugeführt werden.

Prädiktionsfunktion zur Positionsschätzung

Mithilfe der geschätzten End-to-End Latenz L_k^* können die am Endgerät eingetroffenen Positionsdaten des Steuerungssystems in einer Prädiktionsfunktion für $s^*(t)$ genutzt

werden. Die Prädiktionsfunktion kann zum Beispiel unter Annahme konstanter Beschleunigung zwischen zwei Abtastungen mit einem abschnittsweise bestimmten Polynom zweiten Grades formuliert werden. Dazu werden in jedem Abtastschritt die Parameter des Polynoms für das nächste Prädiktionszeitintervall anhand vorangegangener Positionswerte bestimmt. Anschließend wird die Polynomfunktion auf dem Endgerät im Prädiktionszeitintervall zwischen zwei Abtastzeitpunkten interpoliert (vgl. Abb. 16.11). Die Prädiktionsfunktion kann damit für ein Prädiktionszeitintervall wie folgt aufgestellt werden:

$$s^*(\tau) = a_k \cdot \tau^2 + b_k \cdot \tau + c_k \text{ mit } \{\tau \in \mathbb{R} | (t_k + L_k) \leq \tau \leq (t_{k+1} + L_{k+1})\}. \quad (16.12)$$

Anwendungsbeispiel der Kompensationsmethode

Für das in der Einführung skizzierte MRiLS-Anwendungsszenario eines virtuellen Greifers an einer realen Roboterkinematik zeigt Abb. 16.15 den Ist-Positionsverlauf (Ground Truth), den aufgrund der Latenz verzögerten Positionsverlauf (unkompensiert) und den prognostizierte Positionsverlauf (kompensiert) des End-Effektors. Die End-to-End Latenz kann in diesem Anwendungsbeispiel bei einer mittleren Latenz von $L_k^* \approx 110$ ms bis auf ca. 4 ms genau geschätzt und der Positionierungsfehler mit der vorgestellten Kompensationsmethode um 80 % reduziert werden.

Das Diagramm in Abb. 16.15 hebt zusätzlich die drei Zeitpunkte A, B & C hervor, deren Visualisierungen auf dem MR-Endgerät mit und ohne Kompensation in Abb. 16.16 dargestellt sind. Es ist gut zu erkennen, dass die Positionierungsfehler durch Einsatz der Kompensationsmethode deutlich reduziert werden können. Zur weiteren Bewertung der Methode werden die gemessenen Positionierungsfehler (mit und ohne Kompensation) in einem Fehlerfeld-Diagramm dargestellt (siehe Abb. 16.17). Darin werden die Fehler, die durch die zeitliche Verzögerung aufgrund der Latenz L bei konstanter Geschwindigkeit v

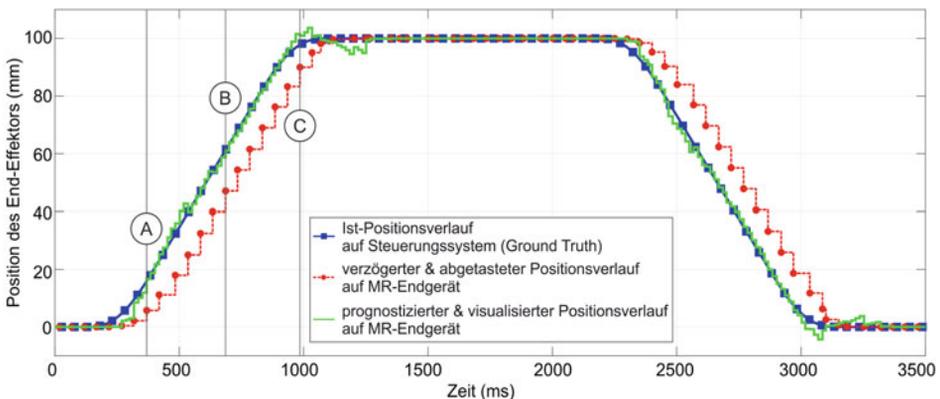


Abb. 16.15 Soll-Positionsverlauf (Ground Truth) des End-Effektors verglichen mit unkompensiertem und kompensiertem Positionsverlauf auf dem MR-Endgerät

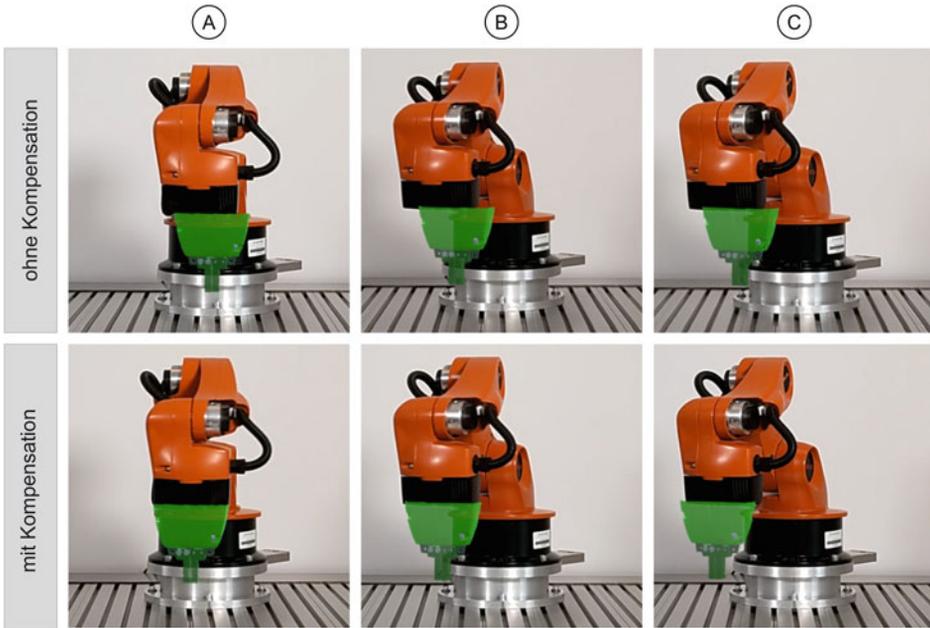
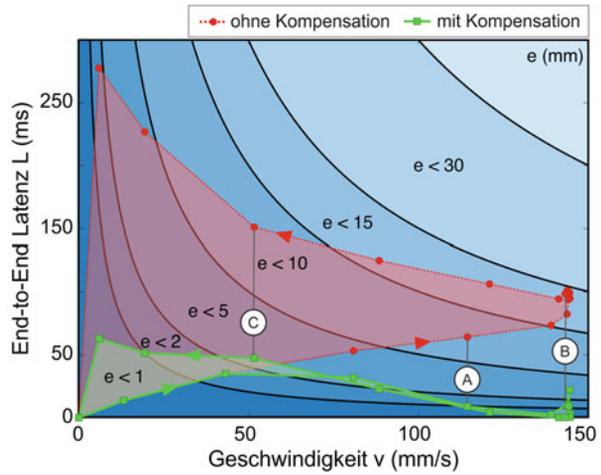


Abb. 16.16 MRiLS-Szenen auf dem MR-Endgerät „Microsoft HoloLens 2 Industrial“ mit und ohne Kompensation

Abb. 16.17
Fehlerfeld-Diagramm mit unkompenzierten und kompensierten Fehlern bei der Positionierung eines virtuellen Greifers nach Abb. 16.15 und 16.16 für $t < 1500$ ms



auf einer geradlinigen Bahn entstehen, entsprechend dem Zusammenhang $e = v \cdot L$ abgebildet. Jedes Fehlerfeld drückt einen Fehlerbereich mit $e < e_{max}$ aus. Das Diagramm zeigt, dass die MRiLS mit Kompensationsmethode in Fehlerfeldern für deutlich kleinere Fehler ausgeführt werden kann, ohne Infrastrukturanpassungen zur Latenzreduktion vornehmen oder die Geschwindigkeit des Automatisierungssystems verringern zu müssen.

Zusammenfassend reduziert die Kompensationsmethode den Positionierungsfehler bewegter virtueller Anlagenkomponenten in einer MRiLS, der durch Abtast- und Latenzeffekte zwischen einem Steuerungssystem und einem MR-Endgerät verursacht wird. Dies verbessert die Anwendbarkeit der MRiLS insbesondere in Szenarien, die durch eine Kombination von bewegten virtuellen und realen Komponenten eines Automatisierungssystems charakterisiert sind.

16.3 Intuitive multimodale Mensch-Modell-Interaktion

Die vollständige Integration des Menschen in ein Simulationsszenario bedarf neben der egozentrischen Visualisierung zusätzlich auch intuitive, immersive und multimodale Interaktionsmöglichkeiten zwischen Mensch und der virtuellen Umgebung. Die visuelle Ausgabe stellt zwar die Basis der Immersion dar, ist aber allein nicht ausreichend [4]. Zur Erreichung eines erhöhten Immersionsgrads sollten weitere menschliche Sinne angesprochen werden (Multimodalität), wie beispielsweise die auditive sowie die haptische Wahrnehmung. Zur Steigerung des subjektiven Präsenzgefühls sind zudem intuitive Interaktionsmechanismen zu integrieren, sodass die Interaktionen für den Menschen plausibel erscheinen.

Interaktionseingabe

Die Interaktionseingabe ermöglicht die Manipulation der virtuellen Umgebung durch Auslösen (z. B. Betätigen eines virtuellen Maschinenschalters) von hinterlegten Methoden in der MRiLS (z. B. Maschine starten). Die Interaktionsmöglichkeiten sind abhängig vom eingesetzten Endgerät und variieren je nach Hersteller und Gerätetyp stark. So können zum Beispiel Controller, Gesten- oder Sprachsteuerungen zum Einsatz kommen (siehe Abb. 16.18). Nachfolgend werden die verschiedenen Eingabemöglichkeiten vorgestellt.



Abb. 16.18 Controller, Gesten- und Sprachsteuerungen zur Interaktion mit einem Digitalen Zwilling

- **Controller:** Controller sind physisch existierende Eingabegeräte, mit denen der Nutzer handgeführte Eingaben tätigen kann. Über die getrackte Position (Ort und Rotation) des Controllers wird die Handposition des Nutzers und dessen Bewegungen in die virtuelle Welt übertragen. Für weitere Eingabeoptionen verfügen Controller über Taster, worüber implementierte Aktionen ausgelöst werden können (z. B. Öffnen eines Menüs). Die Kombination aus Position und Tasterbetätigung des Controllers ermöglicht eine klare Zuordnung bei der Interaktionsausführung. Damit kann beispielsweise ein virtuelles Objekt über die Positionierung des Controllers ausgewählt und über das Betätigen eines Tasters gegriffen werden.
- **Gestensteuerung:** Eine Gestensteuerung ermöglicht die Manipulation der virtuellen Komponenten über vordefinierte Gesten, wie zum Beispiel Bewegungen der Finger, des Körpers, der Augen oder dem Gesicht (Mimikerkennung). Die relevanten Merkmale werden über bildverarbeitende Algorithmen aus Kamerabildern oder über am Körper angebrachte Sensoren erfasst. Die Erkennung der Geste erfolgt durch die Analyse von Bewegungsabfolgen. Durch Finger-Tracking kann das Manipulieren von virtuellen Komponenten durch intuitive Gesten erzielt werden (z. B. Greifen durch Hand schließen und Loslassen durch Hand öffnen). Erweiternd können zusätzliche Sensoren am Körper angebracht werden, um weitere Positionsinformationen des Nutzers, wie beispielsweise die Arm- oder Beinposition in die MRiLS zu integrieren. Mittels Eye-Tracking Methoden kann zudem die Blickrichtung des Menschen erfasst und zur Interaktion genutzt werden.
- **Sprachsteuerung:** Neben den zuvor vorgestellten bewegungsbasierten Eingabemöglichkeiten kann auch Sprache zur Interaktionseingabe eingesetzt werden. Über Sprachbefehle kann der Mensch den Digitalen Zwilling steuern sowie die virtuellen Komponenten manipulieren oder Aktionen hervorrufen. Dabei wird die Spracheingabe mit einem Mikrophon aufgenommen und über eine Merkmalsextraktion mittels eines Akustik- sowie Sprachmodells in einen Text (Zeichenkette/String) umgewandelt (Speech-to-Text). Die Zeichenkette kann anschließend im Modell weiterverarbeitet und zum Aufruf von Funktionen eingesetzt werden, um eine Aktion im Digitalen Zwilling auszulösen.

Interaktionsausgabe

Die Interaktionsausgabe beschreibt die Wirkung der virtuellen Umgebung auf den Menschen. Neben der visuellen Ausgabe ist für die MRiLS insbesondere das auditive und haptische Feedback relevant.

- **Auditives Feedback:** Mittels auditiver Ausgabe werden dem Nutzer Geräusche und Töne (z. B. Maschinengeräusche) in Form von Audiosignalen wiedergegeben, sodass die virtuelle Umgebung in realistischer und plausibler Art wahrgenommen wird. Zudem können auditive Ausgaben wie Tonsignale und Sprachsequenzen bei der

Auswahl von Objekten, zur Hilfestellung oder für zusätzliche Informationen (z. B. Maschinenstatus) verwendet werden. Die Sprachausgabe von Textelementen erfolgt über die automatisierte Umwandlung von Text (Zeichenkette/String) in Sprachsequenzen (Text-to-Speech). Die auditive Ausgabe wird über die Lautsprecher der Endgeräte oder über externe Lautsprecher umgesetzt und kann neben der Informationsbereitstellung auch zur Orientierung im Raum dienen. Durch eine Laufzeitdifferenz zwischen der Ausgabe am rechten und am linken Ohr kann der Mensch die Quelle des Audiosignals orten und sich hierdurch im virtuellen Raum besser orientieren.

- **Haptisches Feedback:** Das Ziel der haptischen Ausgabe ist es, dem Menschen die virtuellen Komponenten durch synthetische Stimuli haptisch erfahrbar zu machen. Dabei ist zwischen mechanischem, neuralem und kontaktlosem haptischem Feedback zu unterscheiden. Das mechanisch haptische Feedback kann sowohl über eine haptisch taktile wie auch eine elektro-mechanische Rückkopplung erzeugt werden. Die haptisch taktile Rückkopplung spricht über Druckpunkte oder Vibration den Tastsinn des Menschen an ausgewählten Körperstellen an. Häufig wird das haptisch taktile Feedback in Datenhandschuhe an der Handfläche oder an einzelnen Fingern positioniert, sodass ein Feedback beim Greifen von virtuellen Objekten generiert werden kann. Alternativ kann das Vibrations-Feedback auch in Controllern integriert sein, um eine Rückmeldung bei der Interaktion mit virtuellen Objekten zu geben. Die elektro-mechanische Rückkopplung ist im Gegensatz zur haptisch taktilen Rückkopplung eine direkte Kraft-rückkopplung und erfolgt über externe aktive Skeletstrukturen, die mechanisch an den interagierenden Körperteilen angebracht werden. Dadurch können physikalische Eigenschaften über die Ansteuerung der externen Skeletstrukturen für eine Interaktion nachgebildet werden. Bei dem neuralen haptischen Feedback werden durch am Körper befestigte Elektroden Stromimpulse erzeugt, die die entsprechenden Muskelpartien in unterschiedlichen Situationen stimulieren. Im Gegensatz zu diesen Methoden der Reizerzeugung am Körper des Menschen, werden bei dem kontaktlosen haptischen Feedback die Reize ohne direkten Kontakt mit der Reizquelle übertragen. Beispiele hierfür sind pneumatische Luftimpulse oder Ultraschallwellen, die über Lautsprecher gezielt auf die Hautoberfläche ausgesendet werden, um dort eine leichte Vibration zu erzeugen.

Abstraktion des Interaktionskreislaufes

Um die Vielzahl an multimodalen Interaktionseingaben und -ausgaben für einen Digitalen Zwilling und dessen Verhaltens- und Interaktionsmodell ohne Adaptionaufwand endgeräteunabhängig verwenden zu können, ist die Abstraktion der Interaktion zwischen Mensch und Digitalem Zwilling erforderlich. In Abb. 16.19 ist das Konzept der Interaktionsabstraktion dargestellt, die gerätespezifische Ein- und Ausgaben in geräteunabhängige Funktionsaufrufe im Digitalen Zwilling abbildet.

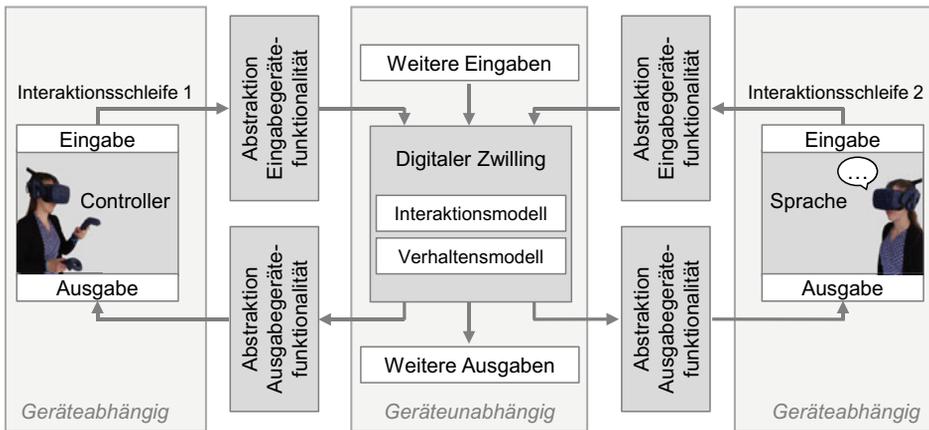


Abb. 16.19 Konzept der Interaktionsabstraktion in Anlehnung an Schnierle et al. [17]

Die Abstraktionsschicht kann als Softwaremodul mit Treiber-Funktion aufgefasst werden, die das Interaktions- und Verhaltensmodell von den spezifischen Gerätefunktionen entkoppelt. Die Eingabe des Nutzers (z. B. Betätigen eines virtuellen Tasters über Controller, Touch-Bedienung, Gestensteuerung oder durch Sprachbefehl) wird über die Abstraktionsschicht mittels definierter Events in gebildeten Interaktionskategorien an den Digitalen Zwilling übermittelt und dort entsprechend des geräteunabhängigen Verhaltens- und Interaktionsmodells ausgewertet. Die darauffolgenden Reaktionen des Verhaltens- und Interaktionsmodells werden gleichermaßen in Events überführt, die entsprechend der Fähigkeiten des angebotenen Ausgabegerätes in eine geräteabhängige Ausgabegeräte-funktionalität aufgelöst werden. Die abstrahierte Ausgabegerätefunktionalität umfasst wie die Eingabegerätefunktionalität verschiedene Kategorien von Interaktionen. Die Ausgabe kann dabei sowohl verschiedene Wahrnehmungen des Menschen, wie die visuelle, haptische oder auditive Wahrnehmung ansprechen (z. B. visuelle Hervorhebung von Objekten, Vibration, auditive Rückmeldung) als auch verschiedene Ausprägungsstufen der Gerätetechnik für dieselbe Modalität umfassen (z. B. Vibration des Controllers oder des Smartphones). Die Ein- und Ausgaben können entweder unidirektional (z. B. Audioausgabe) oder in eine bidirektionale Interaktionsschleife integriert sein, die eine Ein- und Ausgabe mit derselben Gerätetechnik umfasst, (z. B. Positionseingabe und haptische Rückmeldung über denselben Controller).

Die Interaktionsabstraktion ermöglicht neben dem flexiblen Einsatz verschiedener Eingabe- und Ausgabemöglichkeiten auf Basis eines gemeinsamen geräteunabhängigen Digitalen Zwillings zudem die Bereitstellung einer möglichst umfassenden multimodalen Illusion der virtuellen Komponenten entsprechend der individuell eingesetzten Gerätetechnik des jeweiligen Nutzers.



Abb. 16.20 Mensch-Modell-Interaktion am Beispiel eines Robotermodells

Interaktionsmodell für die intuitive Manipulation einer Roboterkinematik

Intuitive und natürliche Interaktionen erfordern neben einer multimodalen Ein- und Ausgabe auch ein entsprechendes Verhaltens- und Interaktionsmodell des Digitalen Zwillinges. Während das Verhaltensmodell das allgemeine Systemverhalten des Digitalen Zwillinges beschreibt (Kinematik, Dynamik, Kollision, ...), ergänzt das Interaktionsmodell den Digitalen Zwilling um das Reaktionsverhalten bei einer Interaktion mit dem Nutzer. Dabei löst eine Interaktion einen im Interaktionsmodell definierten Funktionsaufruf aus, welcher gegebenenfalls das Systemverhalten beeinflusst. Die Wirkung der Interaktion im Verhaltensmodell kann vom einfachen Umschalten eines virtuellen Schalters bis hin zur Manipulation ganzer Maschinenkinematiken reichen. Insbesondere für Schulungs- und Planungszwecke ist die intuitive Manipulation der Maschinenkinematik eine wichtige Anwendung. Abb. 16.20 zeigt beispielhaft eine intuitive Manipulation des Digitalen Zwillinges eines Industrieroboters. Der Nutzer greift per Handgeste ein virtuelles Interaktionsobjekt (rote Kugel), welches kinematisch mit dem Roboter gekoppelt ist und somit die intuitive Manipulation der Roboterpose ermöglicht.

Für die intuitive Manipulation aller Achsen eines 6-achsigen Industrieroboters sind mehrere Interaktionsobjekte möglich. Die Kinematik kann in zwei Teilprobleme zerlegt werden, sodass die kinematischen Ketten der Hauptachsen $q_{Haupt} = [q_1, q_2, q_3]$ und der Handachsen $q_{Hand} = [q_4, q_5, q_6]$ getrennt voneinander betrachtet werden können (siehe Abb. 16.21 links und Mitte). Die Zerlegung der räumlichen Kinematik in zwei zu beeinflussende Ebenen ermöglicht eine entkoppelte und damit intuitivere Manipulation der Roboterkinematik über zwei virtuelle Interaktionsobjekte (siehe Abb. 16.21 rechts). Während über die blaue Kugel die Hauptachsen q_{Haupt} und damit die Position der Handwurzel manipuliert werden, können über die Handachsen q_{Hand} unabhängig davon die Orientierung des Endeffektors über die rote Kugel verändert werden.

Für die Interaktionseingabe ist ein Interaktionsmodell für die numerische Berechnung der inversen Kinematiken (Rückwärtstransformationen) in der Hauptachsen- und in der Handachsen-Ebene anhand der Position des jeweiligen Interaktionsobjekts hinterlegt. Als Interaktionsausgabe können neben grafischen Zusatzinformationen über Gelenkwinkel

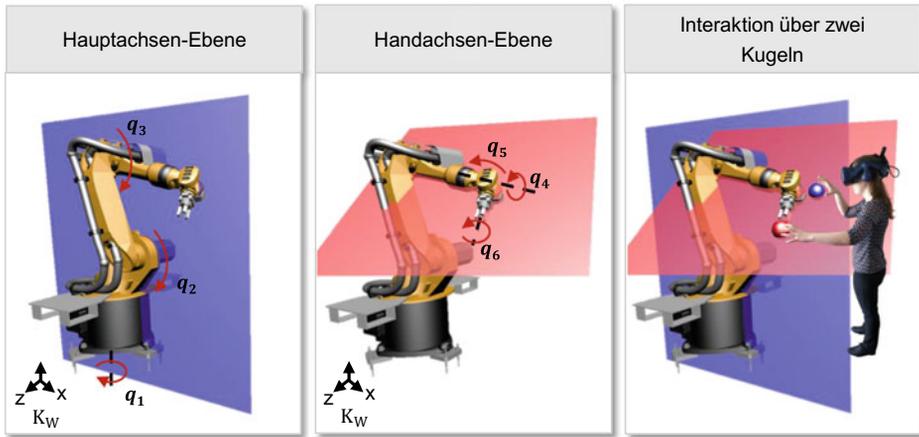


Abb. 16.21 Intuitive Manipulation eines 6-achsigen Industrieroboters mit zwei virtuellen Interaktionsobjekten

und Endeffektorposition auch akustische oder haptische Ausgaben beispielsweise beim Erreichen der Achsgrenzen oder bei ungünstigen Positionen realisiert werden, sofern die entsprechenden Ausgabegeräte zur Verfügung stehen.

16.4 Multiuser-Kollaboration

Die Kollaboration mehrerer Nutzer in einer gemeinsamen Mixed Reality-Umgebung (auch Shared Experience genannt) erfordert sowohl in standortübergreifenden Szenarien als auch in Szenarien an einem gemeinsamen Standort die Einbindung aller beteiligten Mixed Reality-Endgeräte in einen MRiLS-Kreislauf (siehe Abb. 16.22).

Für die Multiuser-Kollaboration ist eine Synchronisierung des Systemzustands zwischen allen Nutzern sicherzustellen. Unabhängig von den eingesetzten Endgeräten müssen sowohl der Digitale Zwilling, die Steuerungsdaten als auch die Nutzer-Positionsdaten für jeden Teilnehmer der MRiLS möglichst latenzarm zur Verfügung gestellt werden. Um dies zu ermöglichen, kommt ein servicebasierter Softwareansatz in Form einer Middleware entsprechend des ‚Digital Twin as a Service‘ – Paradigmas zum Einsatz. Die Middleware bindet die Steuerungsdaten und weitere externe Datenquellen (Assets) über eine standardisierte Asset-Schnittstelle (bspw. über OPC UA oder MQTT) an und stellt diese den MR-Endgeräten vereinheitlicht über die Client-Schnittstelle bereit (vgl. Abb. 16.22). Durch die zentrale Instanz des Digitalen Zwillings existiert zu jedem Zeitpunkt ein konsistenter Systemzustand, der an die angebotenen Endgeräte übertragen werden kann.

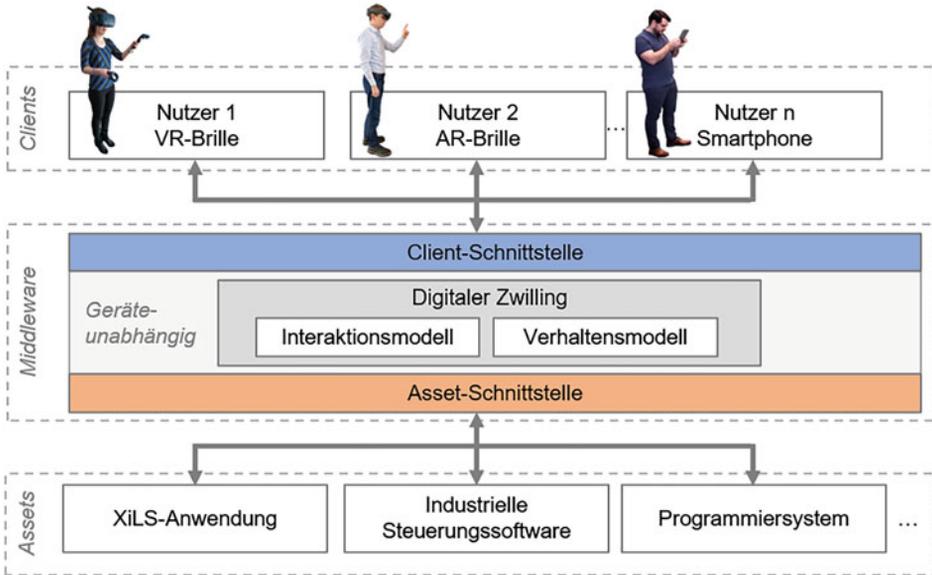


Abb. 16.22 Multiuser-Kollaboration in einer Mixed-Reality-in-the-Loop Simulation

Die Middleware verwaltet und aggregiert zudem die ein- und ausgehenden Datenströme und reduziert damit die auf die Assets wirkende Kommunikationslast im Vergleich zu direkten Punkt-zu-Punkt Verbindungen zwischen Clients und Assets. Zum Austausch der Positionsdaten aller angebotenen Nutzer können Konzepte aus dem Multiplayer-Gaming adaptiert werden. Um den angebotenen heterogenen Endgeräten (z. B. AR/VR-Brillen, Smartphones oder Tablets) einen konsistenten Zustand der Daten bereitzustellen, wird das Netzwerkmodell ‚Snapshot Interpolation‘ eingesetzt. Bei der Snapshot Interpolation erfolgt die Berechnung der Simulation auf einem Gerät (z. B. dedizierter Server), das die Ergebnisse (Zustandsdaten der virtuellen Szene) in Form einer Momentaufnahme (sog. Snapshot) zyklisch an die angebotenen Clients sendet [18]. Die angebotenen Endgeräte können bidirektional mit dem Server kommunizieren und sind eigenständig für die Rekonstruktion der Simulation und der Visualisierung auf dem jeweiligen Endgerät zuständig. Die angebotenen heterogenen Endgeräte können über die zuvor vorgestellte Interaktionsabstraktion mit der Szene interagieren.

Neben der informationstechnischen Vernetzung bedarf es bei der kollaborativen MRiLS der konsistenten Positionierung des Digitalen Zwilling in den MRiLS-Szenen der einzelnen Nutzer. Dafür muss für alle nutzerabhängigen (egozentrischen) Visualisierungen einer Szene ein gemeinsames Koordinatensystem sichergestellt werden, um den Digitalen Zwilling für alle Nutzer an derselben Stelle in der Szene zu visualisieren und jeden Nutzer eindeutig in der Szene zu lokalisieren. Hierbei ist zwischen zwei Szenarien zu



Abb. 16.23 Kollaboration an unterschiedlichen und an einem gemeinsamen Standort

unterscheiden, der Kollaboration an unterschiedlichen Standorten und der Kollaboration an einem gemeinsamen Standort (siehe Abb. 16.23).

- **Kollaboration an unterschiedlichen Standorten:**

Mehrere Nutzer interagieren an unterschiedlichen Standorten mit dem Digitalen Zwilling, wobei die Nutzer keine Wechselwirkungen mit der realen Umgebung haben (VR). Die Lokalisierung der einzelnen Nutzer erfolgt relativ zu dem Koordinatensystem des Digitalen Zwillings auf Basis des nutzerindividuellen Trackings. Neben der Positionierung des Digitalen Zwillings bedarf es in diesem Szenario der virtuellen Repräsentation der einzelnen Nutzer (Avatare), sodass jeder Nutzer zu jedem Zeitpunkt die Position der jeweils anderen Nutzer kennt.

- **Kollaboration an einem gemeinsamen Standort:**

Interagieren mehrere Nutzer an einem gemeinsamen Ort unter Einbeziehung der realen Umgebung (AR), ist neben dem nutzerindividuellen Tracking und der daraus resultierenden relativen Positionierung der Nutzer zum Koordinatensystem des Digitalen Zwillings auch eine zwischen den Nutzern konsistente Verankerung des Digitalen Zwillings im Raum notwendig, beispielsweise über visuelle Marker. Der Marker dient den einzelnen Endgeräten als globales Ursprungs-Koordinatensystem für den Digitalen Zwilling, sodass die Position des Digitalen Zwillings für alle Nutzer im Raum konsistent ist.

16.5 Realisierung und Anwendung

In diesem Abschnitt werden die Realisierung und verschiedene Einsatzszenarien einer MRiLS vorgestellt, die im Rahmen des BMBF-Projekts "Hybrides Interaktionskonzept für Schulungen mittels Mixed-Reality-in-the-Loop Simulation" (Förderkennzeichen 16SV8348) erstellt wurden.

Technologische Architektur

Die Architektur der Realisierung ist in Abb. 16.24 dargestellt. Das zentrale Modul bildet die am Virtual Automation Lab (VAL) an der Hochschule Esslingen entwickelte Digital Twin as a Service Plattform (VAL DTaaSP, [19]), die eine angebundene Hardware-in-the-Loop-Simulation (HiLS) mit Mixed Reality-Endgeräten in Form einer Middleware verbindet. Die HiLS wird mit der Echtzeit-Simulationsumgebung ISG-virtuos und dem Steuerungssystem Beckhoff TwinCAT umgesetzt und über OPC UA mit der Middleware gekoppelt [20].

Die VAL DTaaSP bietet neben den Basisdiensten zur Benutzer- und Assetverwaltung auch ein webbrowsersbasiertes Engineering-Werkzeug, das VAL 3D-Webstudio, zur Modellierung und Visualisierung der MRiLS-Szene im Webbrowser. Durch die browserbasierte Bereitstellung kann von jedem im Netzwerk angemeldeten Endgerät auf die VAL DTaaSP zugegriffen werden, ohne dass eine Installation notwendig ist. Die

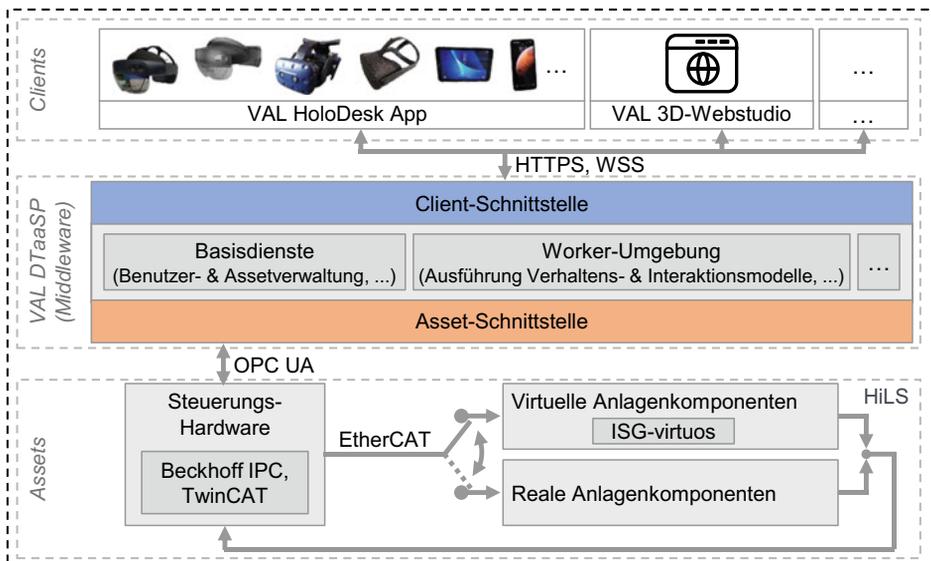


Abb. 16.24 Architektur der Realisierung einer MRiLS

Verhaltens- und Interaktionsmodelle des Digitalen Zwillings können in diesem Entwicklungswerkzeug erstellt, mit den Daten der Assets (hier die HiLS) verknüpft und in der Worker-Umgebung der Middleware ausgeführt werden. Die Bereitstellung und Ausführung der Mixed Reality-Szene auf unterschiedlichen Mixed Reality-Endgeräten erfolgt über die Schnittstellenanwendung VAL HoloDesk, die über das HTTP- und WebSocket-Protokoll an die VAL DTaaSP angebunden ist. Die virtuellen Komponenten werden von der Middleware auf Grundlage einer gemeinsamen Datenbasis an die angebotenen Endgeräte bereitgestellt, ohne dass endgeräte- und applikationsspezifischer

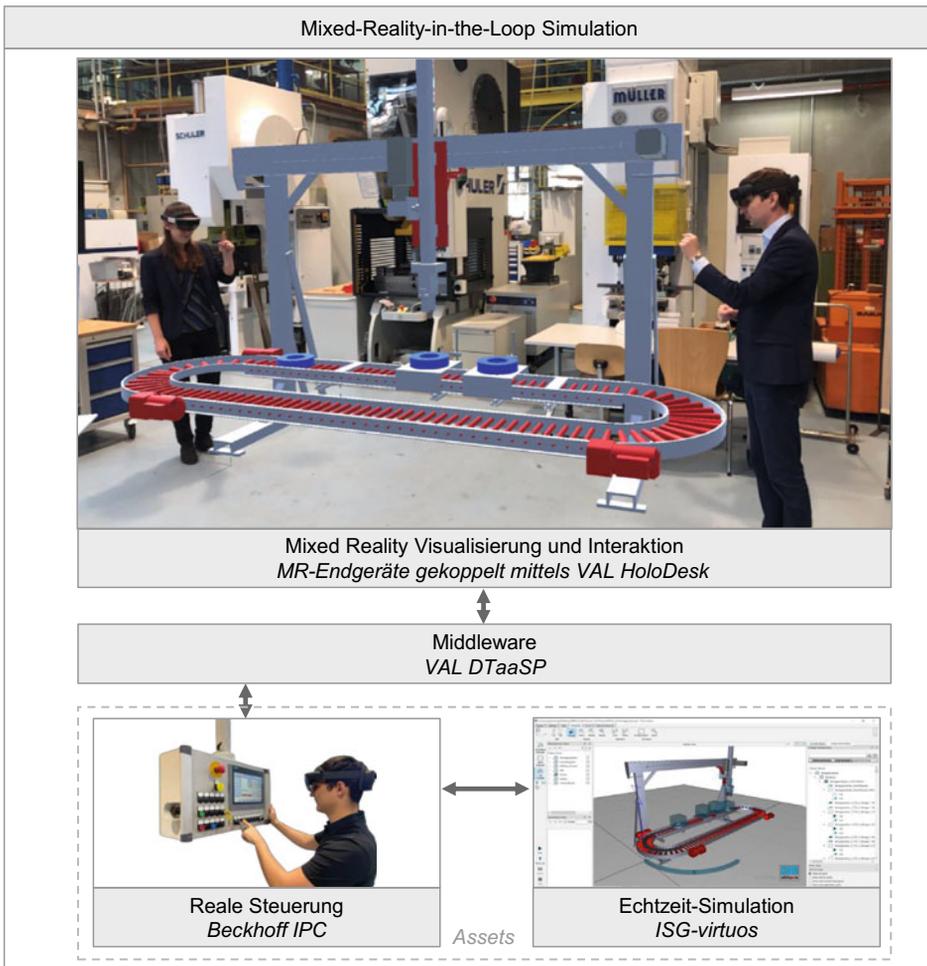


Abb. 16.25 MRiLS eines automatisierten Handhabungssystems in einer AR-Szene

Adaptionen notwendig sind. Die Integration mehrerer Nutzer in einer Szene erfolgt über eine gemeinsame Instanz des Digitalen Zwillings auf der VAL DTaaSP.

Einsatz der MRiLS

Auf Basis der zuvor beschriebenen Architektur wurde die MRiLS eines automatisierten Handhabungssystems der Roth Steuerungstechnik GmbH für die Inbetriebnahme und Schulung technischer Fachkräfte umgesetzt. Abb. 16.25 zeigt beispielhaft den Einsatz der MRiLS in einer Augmented Reality-Szene.

Die beiden MR-User interagieren in dieser Szene mit einem Digitalen Zwilling der Anlage und können damit das Maschinenverhalten intuitiv erlernen. Für die Inbetriebnahme können zudem reale und virtuelle Anlagenkomponenten in einer gemeinsamen Szene „vermischt“ werden, um bereits real existierende Anlagenkomponenten virtuell zu erweitern. Diese Anwendung ermöglicht eine teilvirtuelle Inbetriebnahme unter Einbeziehung des Menschen und der realen Umgebung. Damit können ohne Gefährdung

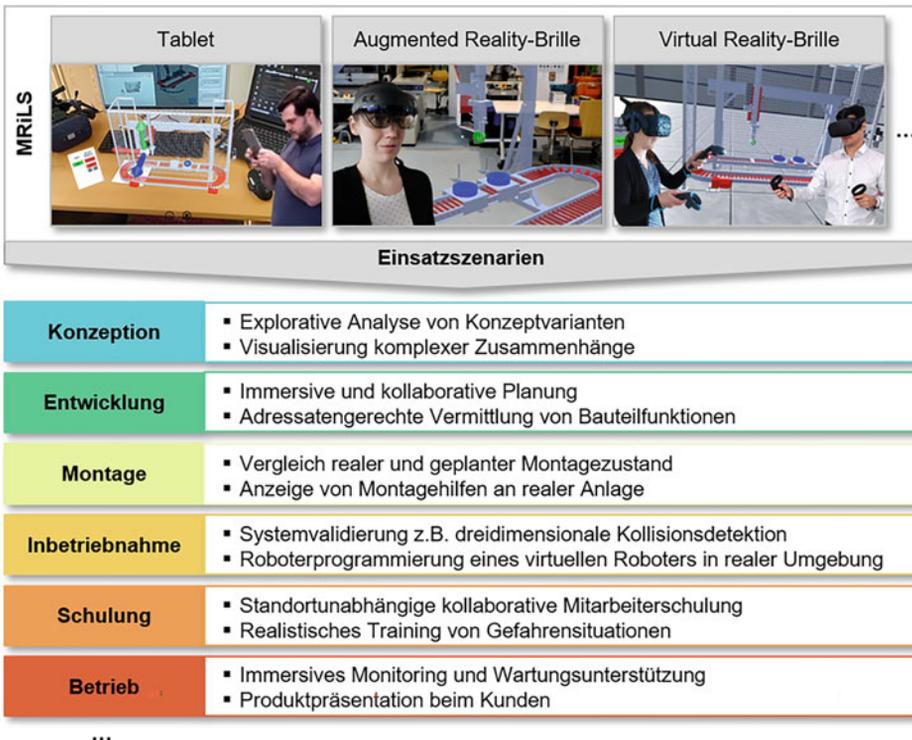


Abb. 16.26 Einsatz der MRiLS während ausgewählten Phasen des Lebenszyklus einer Anlage

von Mensch und Maschine ganze Prozessabläufe getestet, optimiert und geschult werden. Weitere Einsatzmöglichkeiten der MRiLS in verschiedenen Phasen des Lebenszyklus einer Maschine oder Anlage von der Konzeption bis in den Betrieb sind in Abb. 16.26 zusammengefasst.

16.6 Zusammenfassung

Die X-in-the-Loop Simulationsmethoden (XiLS) bilden einen methodischen Baustein, um die wachsenden Anforderungen im mechatronischen Entwicklungsprozess des Maschinen- und Anlagenbaus zu erfüllen. Die bislang eingesetzten XiLS Methoden sind jedoch hinsichtlich der Interaktion zwischen Mensch und Simulation stark limitiert und werden über die Virtuelle Inbetriebnahme von Steuerungssystemen hinaus kaum eingesetzt. Als Querschnittsmethode für neue Anwendungspotentiale von Digitalen Zwillingen wird in diesem Beitrag die Mixed-Reality-in-the-Loop Simulation (MRiLS) vorgestellt, die ein Realitäts-Virtualitäts-Kontinuum aus realen und virtuellen Anlagenkomponenten aufbaut und den Menschen mittels moderner Visualisierungstechnologien vollständig in den Simulationskreislauf integriert. Zur Erreichung eines hohen Präsenzgefühls des Menschen im Simulationskreislauf strebt die MRiLS eine egozentrische realzeitsynchrone Visualisierung, eine intuitive multimodale Mensch-Modell-Interaktion sowie eine Multiuser-Kollaboration an. Aus diesen Zielen resultieren vielfältige Anforderungen an die MRiLS, die in diesem Beitrag ausführlich erörtert und zu denen Lösungsansätze vorgeschlagen werden.

Literatur

1. Verband Deutscher Maschinen- und Anlagenbau e. V. (2017) IT-Report „Simulation im Maschinenbau“. VDMA-Verlag
2. VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (2016) Virtuelle Inbetriebnahme – Modellarten und Glossar, VDI/VDE 3693 Blatt 1
3. Slater M, Wilbur S (1997) A Framework for Immersive Virtual Environments (FIVE): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and Virtual Environments* 6:603–616, MIT Press, Cambridge
4. Sherman WR, Craig AB (2003) Understanding virtual reality – Interface, Application, and Design, in *Presence: Teleoperators and Virtual Environments* 12(4):441–442, MIT Press, Cambridge
5. Oh CS, Bailenson JN, Welch GF (2018) A systematic review of social presence: Definition, antecedents, and implications. *Frontiers in Robotics and AI*, Bd. 5. doi: 10.3389/frobt.2018.00114
6. Slater M (2003) A note on presence terminology. *Presence-Connect* 3(3):1-5
7. Slater M (2009) Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*. 364(1535):3549–3557. doi: 10.1098/rstb.2009.0138

8. Milgram P, Takemura H, Utsumi A, Kishino F (1995) Augmented reality: A class of displays on the reality-virtuality continuum. *Proceedings of SPIE 2351, Telem manipulator and Telepresence Technologies*. doi: 10.1117/12.197321
9. Dörner R, Broll W, Grimm P, Jung B (2019) *Virtual und Augmented Reality (VR/AR) – Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Springer Vieweg, Heidelberg. doi: 10.1007/978-3-662-58861-1
10. Tönnis M (2010) *Augmented Reality – Einblicke in die Erweiterte Realität*. Springer, Heidelberg. doi: 10.1007/978-3-642-14179-9
11. Schlegel M (2011) *Zeitkalibrierung in Augmented Reality Anwendungen*. Dissertation. TU München
12. Schnierle M, Röck S (2022) Latency and sampling compensation in mixed-reality-in-the-loop simulations of production systems. *Production Engineering* 17:341–353. Springer. doi: 10.1007/s11740-022-01175-2
13. Puttnies H, Schweissguth E, Timmermann D, Schacht J (2019) Clock synchronization using linear programming, multicasts, and temperature compensation. *IEEE Global Communications Conference (GLOBECOM)* 1–6. doi: 10.1109/GLOBECOM38437.2019.9013260
14. Gore RN, Lisova E, Åkerberg J, Björkman M (2021) CoSiWiNeT: A clock synchronization algorithm for wide area IIoT network. *Applied Science* 11. MDPI, Basel. doi: 10.3390/app112411985
15. Cristian F (1989) Probabilistic clock synchronization. *Distributed Computing* 3(3):146–158. doi: 10.1007/BF01784024
16. Gore RN (2021) *Investigating software-based clock synchronization for industrial networks*. Licentiate Theses. Mälardalen University Sweden - School of Innovation, Design und Engineering
17. Schnierle M, Polak C, Röck S (2019) Mensch-Roboter-Interaktion mit Mixed Reality auf Basis einer „Digital Twin as a Service“-Plattform, *atp magazin* 5/2019, Robotik und Digital Twin in der Smart Factory. Vulkan-Verlag. doi: 10.17560/atp.v61i5.2415
18. Welper G (2015) Transformed snapshot interpolation. *arXiv: Numerical Analysis*. doi: 10.48550/arXiv.1505.01227
19. Schnierle M, Röck S (2018) Plattform für die Mixed-Reality-in-the-Loop-Simulation. *wt-online*, Springer-VDI. doi: 10.37544/1436-4980-2018-09-59
20. Hönig J, Schnierle M, Scheifele C, Spielmann T, Münster C, Roth A, Röck S, Verl A (2021) Mixed-reality-in-the-loop simulation von Produktionssystemen zur Aus- und Weiterbildung. *atp magazin*, *atp* 63:6–7, Vulkan-Verlag. doi: 10.17560/atp.v63i6-7.2538

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

