DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg
Campus Friedrichshafen

Digitaler Zwilling: Verbesserung und Weiterentwicklung der Steuerung, Visualisierung und Simulation

Studienarbeit T3200

des Studienganges Elektrotechnik – Automation
an der dualen Hochschule Baden-Württemberg Ravensburg
Campus Friedrichshafen

von

Lukas Kehrer

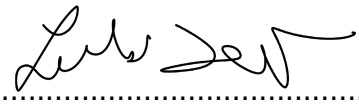
Abgabedatum:	05. Juli 2024
Bearbeitungszeitraum:	08.04.2024 - 05.07.2024
Matrikelnummer, Kurs:	5121766, TEA21
Gutachter der Dualen Hochschule:	Prof. Dr. Thorsten Kever

I. Sperrvermerk

Gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsverordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 07.10.2022:

„Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung vom Dualen Partner vorliegt."

Friedrichshafen, den 05.07.2024



.....

Lukas Kehrer

II. Eigenständigkeitserklärung

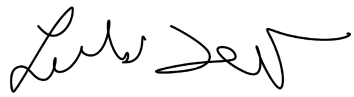
Eigenständigkeitserklärung gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 25.07.2018.

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema:

„Digitaler Zwilling: Verbesserung und Weiterentwicklung der Steuerung, Visualisierung und Simulation“

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Friedrichshafen, den 05.07.2024



.....

Lukas Kehrer

III. Kurzfassung

Diese Studienarbeit befasst sich mit der Erstellung eines digitalen Zwillings eines realen Produktionsnetzwerkes, genannt *Zukunftsfabrik Bodensee*. Als Ausgangslage steht bereits ein grundlegendes digitales Modell aus einer vorangegangenen Studienarbeit zur Verfügung. Dieses wurde jedoch aufgrund von drastischen Mängeln verworfen und daher ein gesamter Neuaufbau des Modells beschlossen. Der Schwerpunkt dieser Studienarbeit liegt dabei auf der Simulation. Aufgrund der Komplexität dieser Aufgabe wurde diese in zwei inhaltliche und somit auch zwei Teile der Studienarbeit getrennt. Der bereits erstellte, erste Teil, inkludiert die theoretischen Grundlagen sowie die Festlegung der gesamten Methodik. Außerdem wurde im Rahmen der ersten Studienarbeit das nötige Fachwissen erlernt, um das statische, digitale Modell dynamisieren zu können. Das Fachwissen wurde am alten Modell angewendet, um ein vollständiges, dynamisches Modell zu erhalten, was später auch auf das neue Modell übertragen wurde.

In diesem zweiten Teil der Studienarbeit wird nun auf die Umsetzung einer Simulation eingegangen. Hierfür wird der Montage-Prozess der Prozesskette virtuell abgebildet. Zudem soll anhand einer möglichen Schnittstelle zur virtuellen, speicherprogrammierbaren Steuerung (SPS) aufgezeigt werden, wie eine Kommunikation zwischen der SPS und dem digitalen Zwilling umgesetzt werden kann.

IV. Abstract

This thesis deals with the creation of a digital twin of a real production network, called *Zukunftsfabrik Bodensee*. A basic digital model from a previous thesis is already available as a starting point. However, this was discarded due to drastic deficiencies, and therefore a complete rebuild of the model was decided. The focus of this thesis is on simulation. Due to the complexity of this task, it was divided into two content-related and thus also two parts of the thesis. The first part, which has already been completed, includes the theoretical foundations and the definition of the entire methodology. In addition, the necessary specialist knowledge was acquired as part of the first thesis to be able to dynamize the static, digital model. The expertise was applied to the old model to obtain a complete, dynamic model, which was later transferred to the new model.

In this second part of the thesis, the realization of a simulation is now discussed. For this purpose, the assembly process of the process chain is modelled virtually. In addition, a possible interface to the virtual programmable logic controller (PLC) will be used to show how communication between the PLC and the digital twin can be implemented.

V. Abbildungs- und Tabellenverzeichnis

Abbildungen

Abbildung 6.1: Funktion "Inverse Kinematik" am Beispiel des UR5e-Roboters 3
Abbildung 6.2: Ablaufschema des UR5e-Roboters im Sequenz-Editor 5
Abbildung 6.3: Ablaufschema des UR5e-Greifers im Sequenz-Editor 6
Abbildung 6.4: Auflegen des Werkstückträgers und Werkstücks an Station 1 7
Abbildung 6.5: Aufnehmen und Platzieren des Bolzens an Station 2 8
Abbildung 6.6: Verknüpfung der Operationen des Umlegers im Sequenz-Editor 8
Abbildung 6.7: Übersicht Umlegeprozess 9
Abbildung 6.8: Aufnehmen und Ablegen des fertig montierten Werkstücks 10
Abbildung 6.9: Übersicht PLC-Variablen in TIA Portal 13
Abbildung 6.10: Symboltabelle der Signale in NX MCD 14
Abbildung 6.11: Definition des Signaladapters in NX MCD 15
Abbildung 6.12: Signalzuordnung in NX MCD zwischen PLCSIM und MCD 17

VI. Abkürzungs- und Fremdbegriffsverzeichnis

Abkürzung	**Erläuterung**
CAD	Rechnerunterstütztes Konstruieren
MCD	Mechatronics Concept Designer
SPS	Speicherprogrammierbare Steuerung

VII. Inhaltsverzeichnis

I. Sperrvermerk i
II. Eigenständigkeitserklärung ii
III. Kurzfassung iii
IV. Abstract iv
V. Abbildungs- und Tabellenverzeichnis v
VI. Abkürzungs- und Fremdbegriffsverzeichnis vi
VII. Inhaltsverzeichnis vii
1 Vorwort 1
6 Umsetzung 2
6.1 Erstellung des dynamischen 3D-Modells 2
6.2 Erstellung einer Simulation in NX MCD 4
6.3 Schwierigkeiten bei der Umsetzung 11
6.4 Mögliche Schnittstelle zur virtuellen SPS 12
7 Fazit und Ausblick 18
Literaturverzeichnis 19

1 Vorwort

In der sich ständig weiterentwickelnden Welt der industriellen Fertigung, ist die Optimierung von Produktionsabläufen elementar für Effizienz und Erfolg. Der stetige Druck, flexibel und vor allem schnell auf wechselnde Marktbedingungen und steigende Kundenerwartungen reagieren zu können, ist dauerhaft präsent. Stellen Sie sich vor Sie besäßen eine Zeitmaschine, mithilfe derer Sie in die Zukunft ihrer Produktionslinie blicken könnten. Es wäre Ihnen dadurch erlaubt potenzielle Herausforderungen vorherzusehen und für diese bereits vor realer Umsetzung der Produktionslinie optimale Lösungen zu finden. Genau das ermöglicht die Technologie des digitalen Zwillings. Durch ein bereits vor der realen Umsetzung bestehendes digitales Modell, lässt sich dessen Verhalten voraussagen und testen.

Im Rahmen dieser Arbeit gilt es eine digitale Repräsentation einer physischen Produktionslinie zu erstellen. Dieser sogenannte Digitale Zwilling soll diesem Produktionssystem maßgetreu entsprechen, sowie dessen reale Daten in Echtzeit empfangen und verarbeiten können. Aufgrund des enormen Umfangs dieser Aufgabe, wird diese in zwei Teile gegliedert. Hierbei enthält dieser zweite Teil der Studienarbeit nun die Umsetzung einer Simulation. Hierfür wird der Montage-Prozess der Prozesskette virtuell abgebildet. Zudem soll anhand einer möglichen Schnittstelle zur virtuellen, speicherprogrammierbaren Steuerung (SPS) aufgezeigt werden, wie eine Kommunikation zwischen der SPS und dem digitalen Zwilling umgesetzt werden kann.

Es wird in dieser Arbeit direkt an die erste Studienarbeit angeknüpft, folglich startet diese Dokumentation direkt mit dem Kapitel 6.

6 Umsetzung

6.1 Erstellung des dynamischen 3D-Modells

Zunächst werden noch, ergänzend zu den aufgezeigten Funktionen in der ersten Studienarbeit, einige weitere Funktionen benötigt, um eine Simulation umsetzen zu können, welche nun an dieser Stelle kurz beschrieben werden.

Inverse Kinematik:

Damit sich die Roboter im digitalen Zwilling bewegen können, gibt es in NX MCD die Funktion „Inverse Kinematik“, womit sich die Umsetzung der Bewegung des Roboters wesentlich vereinfachen lässt. Generell versteht man unter dem Begriff „Inverse Kinematik“ einen speziellen Bereich der Robotik und Automatisierung, der sich mit der Bestimmung der erforderlichen Gelenkwinkel oder Antriebsbewegungen befasst, um einen Roboterarm oder ein anderes mechanisches System in eine gewünschte Position zu bringen. Im Gegensatz zur Vorwärtskinematik, bei der die Position und Orientierung des Endeffektors (zum Beispiel ein Greifer) aus gegebenen Gelenkwinkeln berechnet wird, arbeitet die Inverse Kinematik in umgekehrter Weise: Sie berechnet die erforderlichen Gelenkwinkel, um eine bestimmte Position und Orientierung des Endeffektors zu erreichen.

In NX MCD wird die Inverse Kinematik für die Roboter wie folgt umgesetzt:

Im ersten Schritt muss zunächst ein kinematisches Modell des Roboters erstellt werden. Dies beinhaltet die Festlegung von Gliedern (Starrkörper), Drehgelenken und deren Bewegungsgrenzen. Dies wurde bereits im Rahmen der ersten Studienarbeit näher beschrieben und umgesetzt.

Als nächstes definiert man einen Starrkörper als Endeffektor (hier den Greifer) und gibt zudem noch einen Startpunkt sowie eine Startorientierung an (Startposition). Danach definiert man mehrere Zielpositionen, welche der Roboter nach und nach abfahren

soll. In Abbildung 6.1 wird dies anhand des UR5e-Roboters der Station 1 veranschaulicht. Die Positionen werden hierbei als Posen bezeichnet.

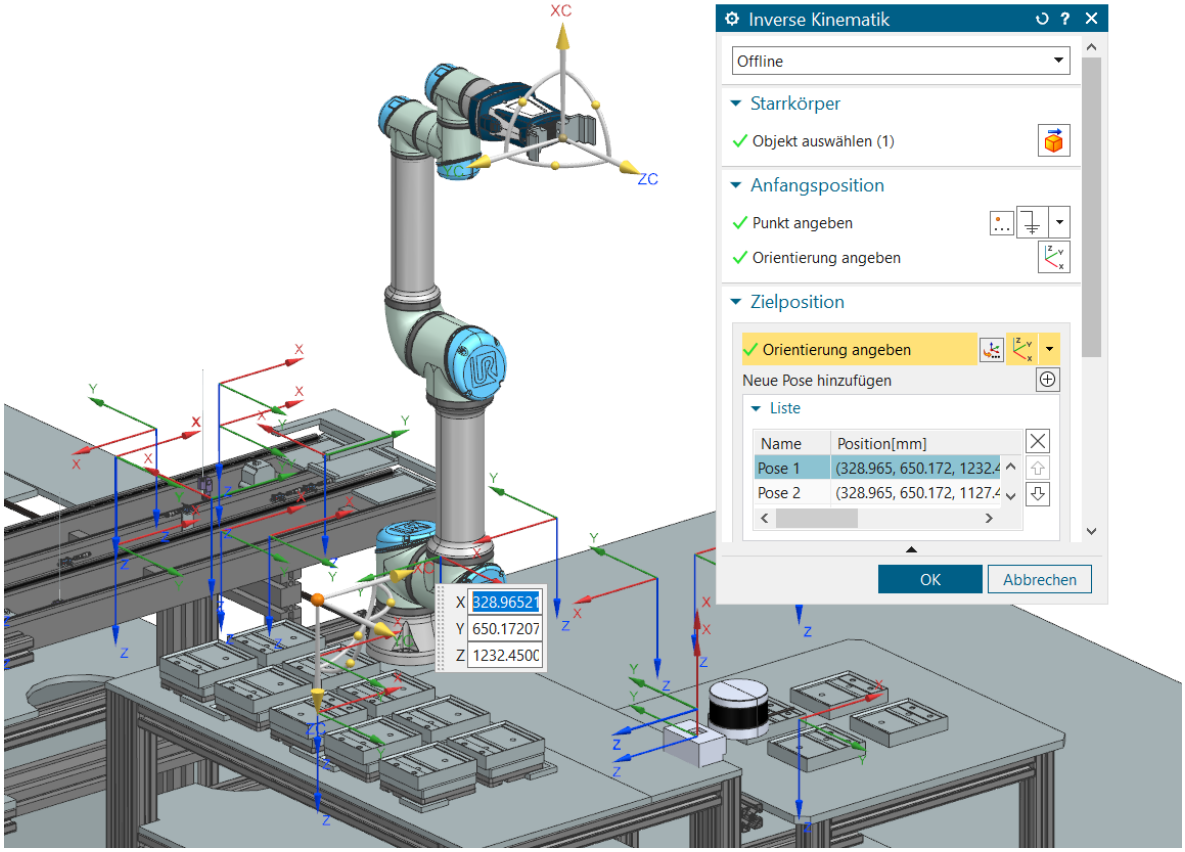


Abbildung 6.1: Funktion "Inverse Kinematik" am Beispiel des UR5e-Roboters

Als erste Pose wurde hier zunächst das Anfahren des Werkstückträgers definiert, um diesen später greifen zu können. Anschließend soll der Roboter sich über das Förderband ausrichten, damit dort später der Werkstückträger abgelegt werden kann. Danach soll er das Werkstück anfahren, um auch dieses später auf den Werkstückträger zu legen. Zu sehen ist in der Abbildung auch, dass mehrere Posen definiert wurden, wobei jede Pose als Koordinate (x, y, z) dargestellt wird. Dies dient zum einen dazu, dass der Roboter einen fest definierten Bewegungsablauf hat, um unrealistische Bewegungen zu vermeiden und zum anderen, dass sich dadurch auch gezielt lineare Bewegungen umsetzen lassen. Hat man alle nötigen Posen definiert, werden im Anschluss automatisch, über die Software, die Positionsregler für jedes Gelenk erstellt und die Operationen derer im Sequenz-Editor hinterlegt.

<u>Operationen und Sequenz Editor:</u>

Operationen werden in NX MCD dazu verwendet, um auf die Parameter der Objekte wie Gelenke oder Transportflächen zuzugreifen. Beispielsweise können Operationen

dazu verwendet werden, um die Startzeit, Position und Geschwindigkeit eines Aktuators zu steuern. Für die Simulation in NX MCD werden Sequenzen von Operationen im sogenannten „Sequenz Editor“ erstellt, um Bewegungsabläufe im digitalen Zwilling koordinieren zu können. Beim Sequenz Editor handelt es sich um ein Gantt-Diagramm-ähnlichen Navigator, der die Operationen und ihre Reihenfolge anzeigt. Dieser Editor dient dazu, beliebige Operationen hinzuzufügen, logisch zu verknüpfen und Parameter der Operationen zu ändern. Die Operationen können hierbei zeitbasiert oder durch ein Ereignis ausgelöst werden. Es ist auch möglich Operationen miteinander zu verknüpfen, sodass beispielsweise eine Operation erst nach Ablauf der vorherigen ausgeführt wird. [1] [2]

6.2 Erstellung einer Simulation in NX MCD

Der digitale Zwilling ist nun mit allen nötigen dynamischen Eigenschaften versehen worden, um eine erste Simulation zu erstellen. Diese wurde zunächst direkt in NX MCD umgesetzt, da zu diesem Zeitpunkt noch keine Schnittstelle zusammen mit dem Team der SPS definiert werden konnte, um die Signale zwischen der SPS und dem digitalen Zwilling auszutauschen. Es wird später noch auf eine mögliche Schnittstelle eingegangen, welche unabhängig vom SPS-Team erstellt wurde und eine zukünftige Möglichkeit zur Umsetzung aufzeigt.

Simulation des Montage-Prozesses:

An dieser Stelle soll die Umsetzung der Simulation des Montage-Prozesses näher betrachtet werden. Der Montage-Prozess beginnt zunächst an der Station 1 mit dem Auflegen des Werkstückträgers und Werkstücks. Durch die Funktion „Inverse Kinematik“ wurden bereits, wie vorher beschrieben, bei der Definition der unterschiedlichen Posen, die Operationen zum Verfahren des Roboters automatisch im Sequenz-Editor erstellt. Hierfür wurde für jedes Gelenk und jede Pose eine Operation erstellt, welche dem Positionsregler die Geschwindigkeit in Grad/s und die Position in Grad übermittelt. Zum Aufnehmen und Ablegen des Werkstückträgers und des Werkstücks auf das Förderband sowie für das Aufnehmen und Ablegen des fertig montierten Werkstücks auf den mobilen Huskey Roboter, wurden hierbei insgesamt 28 Posen definiert, um ein sauberes Verfahren des Roboters zu gewährleisten. Da der Roboter 6 Gelenke hat, entspricht dies 6 Gelenke x 28 Posen = 168 Operationen.

Würde man nun die Simulation starten, so würde der Roboter einfach nur nach und nach alle Posen direkt abfahren. Damit der Roboter auch in bestimmten Posen erstmal verharrt, um beispielsweise das Greifen des Werkstückträgers zu ermöglichen, müssen die Operationen noch verknüpft werden. Abbildung 6.2 zeigt das Ablaufschema (Inverse Kinematik) des UR5e-Roboters im Sequenz-Editor. Die blauen und grünen Kästchen stehen hierbei für die unterschiedlichen Operationen, die für die Bewegung des Roboters benötigt werden, wobei jedes Kästchen eine Pose entspricht.

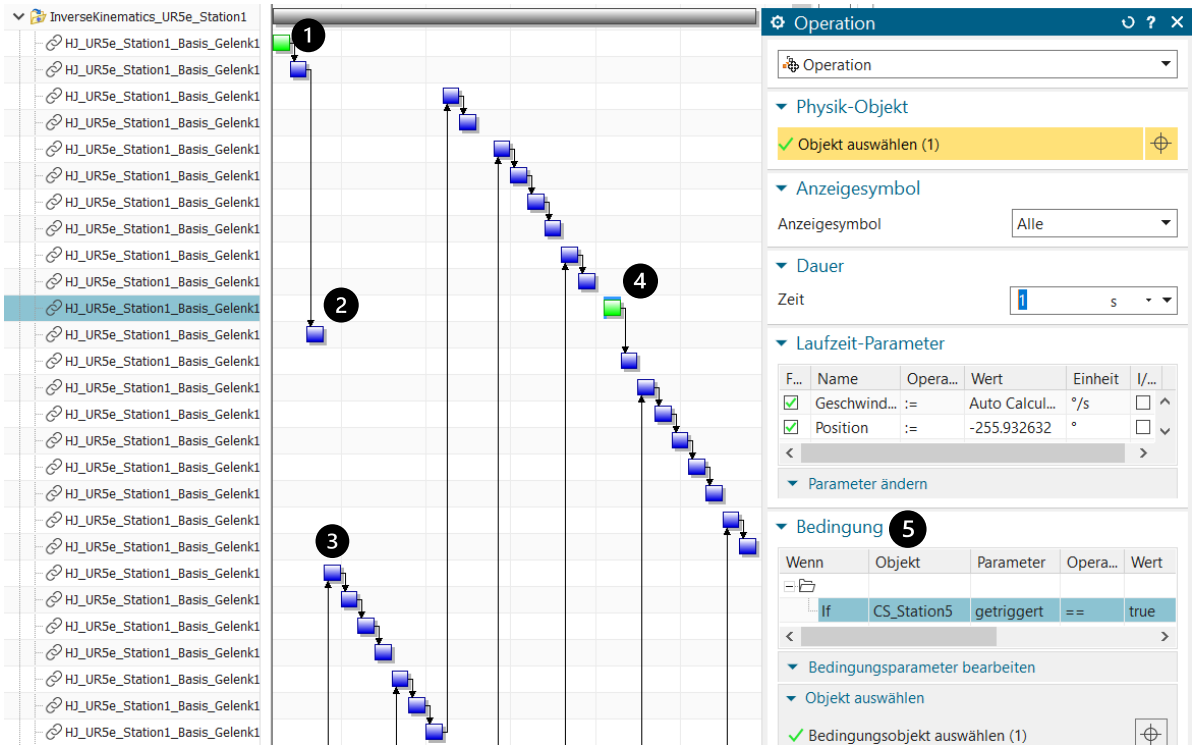


Abbildung 6.2: Ablaufschema des UR5e-Roboters im Sequenz-Editor

Zu sehen ist in diesem Ausschnitt dabei nur das Ablaufschema eines Gelenks des Roboters. Das Ablaufschema der anderen 5 Gelenke ist jedoch analog dazu aufgebaut. Als erste Pose (1) ist das Anfahren des Werkstückträgers definiert worden. Die grüne Farbe des Kästchens bedeutet zudem, dass zusätzlich noch eine Bedingung hinterlegt wurde, in diesem Fall ist es ein Startbefehl der Simulation. Die Pfeile zwischen den Operationen bedeuten, dass diese miteinander verknüpft sind, sprich, jede weitere Operation/Pose wird erst ausgeführt, wenn die vorherige abgeschlossen ist. Nach der dritten Operation (2) befindet sich der Roboter-Greifer direkt über dem Werkstückträger und bleibt erst mal in dieser Position stehen. Es folgt nun das Greifen des Werkstückträgers. Abbildung 6.3 zeigt dabei die Umsetzung des Greif-Vorgangs. Sobald die Position über dem Werkstückträger erreicht wurde, löst dies die Operation

„Greifen des Werkstückträgers“ aus (Zeile 169), welche ein „true“ beim Parameter „Ziehpunkt“ und ein „false“ beim Parameter „Freigabe“ des Greifers schreibt. Dies bewirkt, dass der Greifer beginnt, nach einem Kollisionskörper zu greifen und diesen auch festhält, wenn sich einer im Bereich der Greiferbacken befindet. Anschließend wird die Operation „Greifer ist geschlossen“ (Zeile 170) ausgeführt, welche mit einem Ereignis hinterlegt wurde. Dabei wird das Ereignis ausgelöst, wenn der Greifer den Parameter „ist geschlossen“ mit „true“ zurückgibt. Wenn der Greifer geschlossen ist, so wird wieder in das Ablaufschema des Roboters eingesprungen (Abb. 6.2) und die nächste Pose wird angefahren (3).

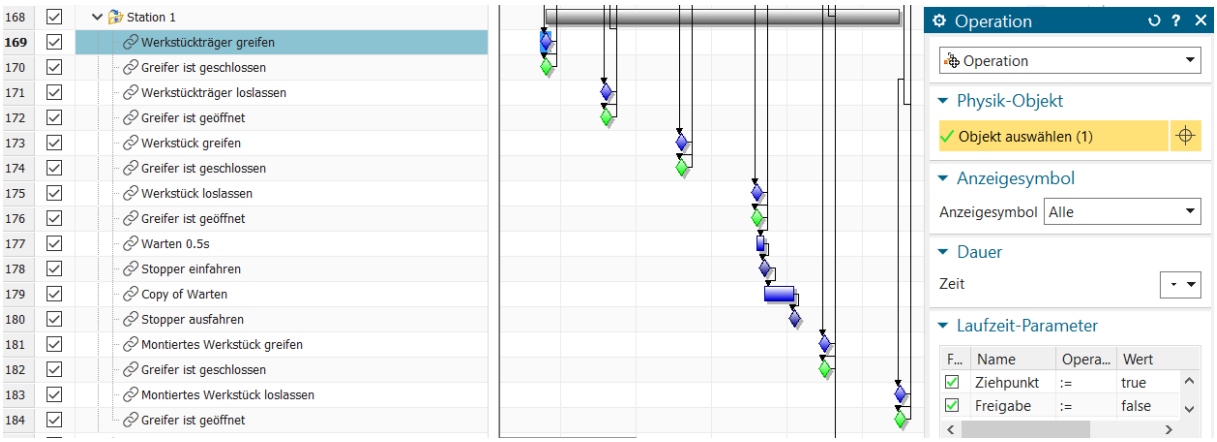


Abbildung 6.3: Ablaufschema des UR5e-Greifers im Sequenz-Editor

Der Roboter verfährt dann über das Förderband, es wird wieder in das Ablaufschema des Greifers gewechselt, dieser öffnet sich daraufhin (Zeile 171) und sobald der Greifer geöffnet ist, wird daraufhin der Roboter zur nächsten Pose verfahren. Sobald der Werkstückträger auf das Band aufsetzt, wird dieser noch von einem Stopper in Position gehalten, da nun noch das Werkstück auf den Werkstückträger platziert werden muss. Dies ist analog zum Aufsetzen des Werkstückträgers umgesetzt. Abbildung 6.4 veranschaulicht die Montage-Prozess-Schritte der Station 1 nochmal bildlich, von oben links nach unten rechts. Der letzte Bewegungsablauf des Roboters (ab (4)) in Abbildung 6.2, ist das Aufnehmen des fertig montierten Werkstücks und Ablegen dessen auf den mobilen Huskey Roboter. Diese Operation ist aber wieder mit einer Bedingung verknüpft, deren Definition sichtbar ist (5). Die Operation wird nämlich erst ausgeführt, wenn der Sensor am Band an Station 5 ausgelöst wird.

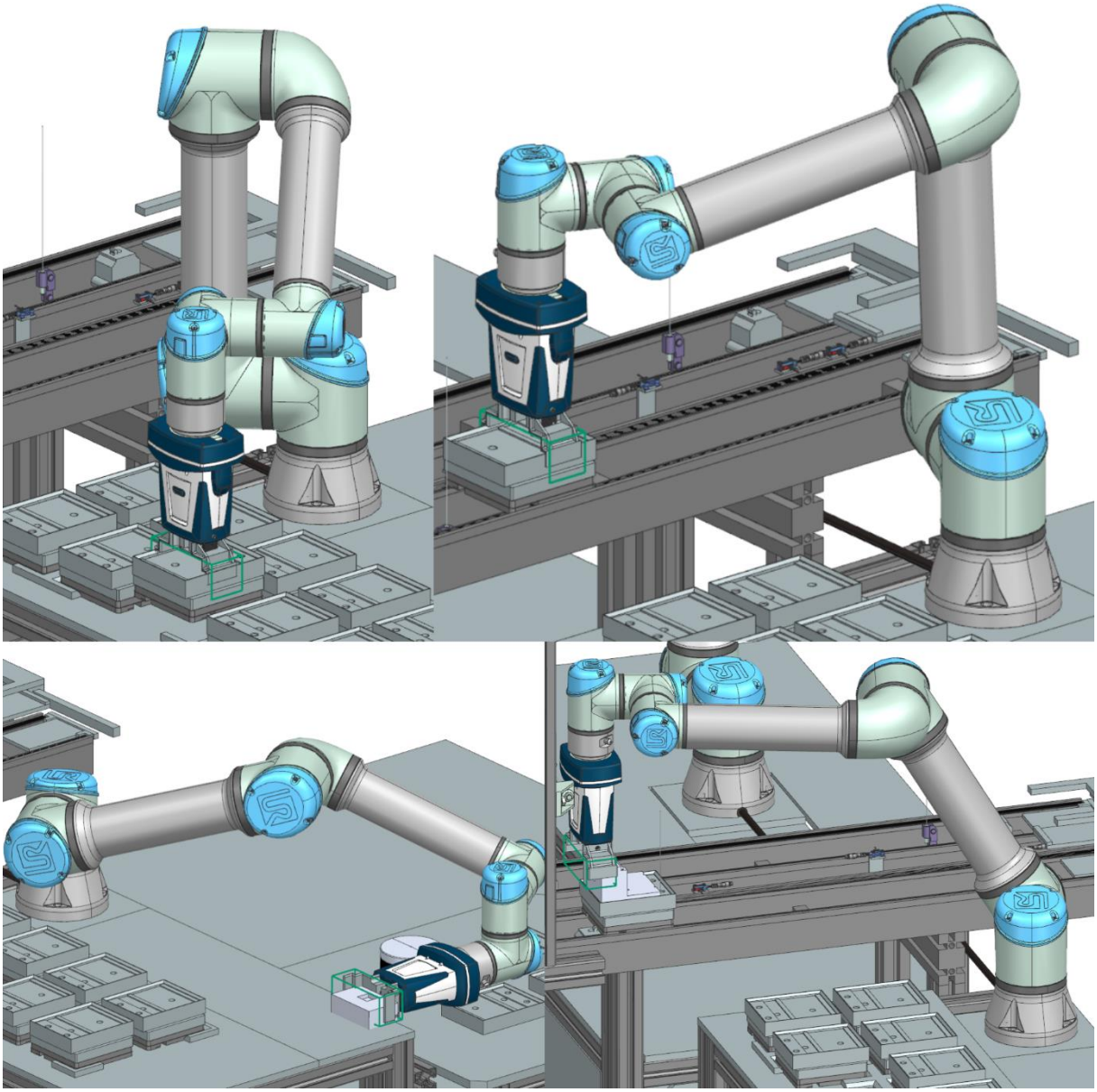


Abbildung 6.4: Auflegen des Werkstückträgers und Werkstücks an Station 1

Nachdem nun das Werkstück auf den Werkstückträger platziert wurde, wird der Stopper eingefahren und das Werkstück fährt weiter bis zur Station 2, wo es wieder von einem Stopper in Position gehalten wird. Sobald der Sensor an der Station 2 eine Kollision mit dem Werkstück erkennt, wird der UR3e-Roboter dieser Station aktiviert und es beginnt das Ablaufschema dieses Roboters. Da die Umsetzung derer an Station 1 ähnelt, wird diese hier nicht näher aufgezeigt, jedoch wird der Montage-Prozess kurz erläutert. An der Station 3 werden vom UR3e-Roboter 2-mal jeweils nacheinander 1 Bolzen aus einem Behälter entnommen, welche anschließend auf dem Werkstückträger platziert werden. Abbildung 6.5 zeigt das Aufnehmen und Platzieren einer der Bolzen von links nach rechts.

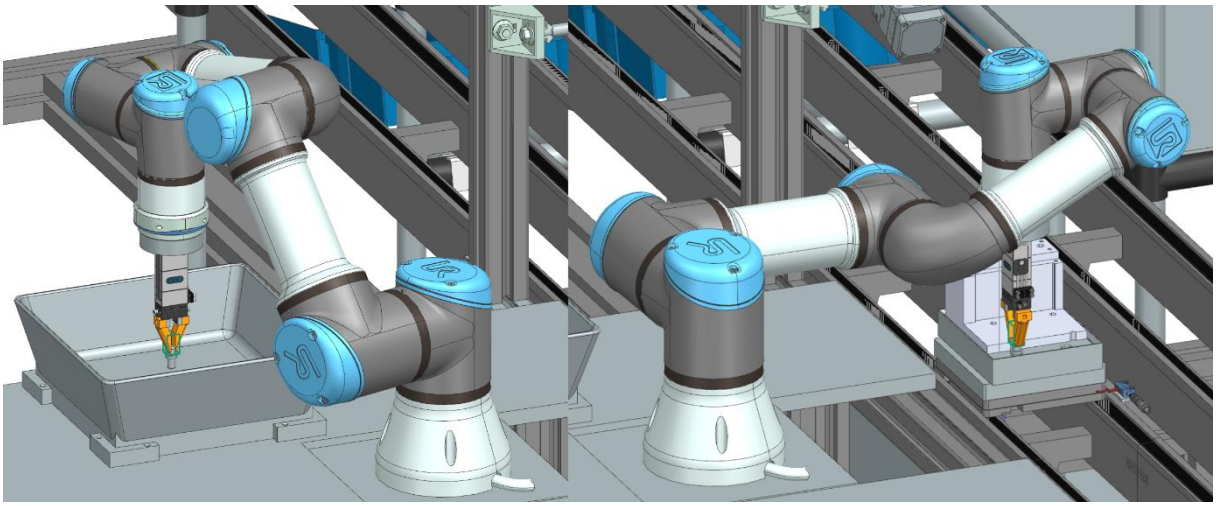


Abbildung 6.5: Aufnehmen und Platzieren des Bolzens an Station 2

Sobald beide Bolzen platziert sind, fährt der Stopper ein und das Werkstück fährt weiter zur nächsten Station. Es folgt davor jedoch noch das Umlegen des Werkstückträgers samt Werkstück auf das andere Band. Die Umsetzung des Umlege-Prozesses ist in Abbildung 6.6 dargestellt, was das Ablaufschema im Sequenz-Editor aufzeigt.

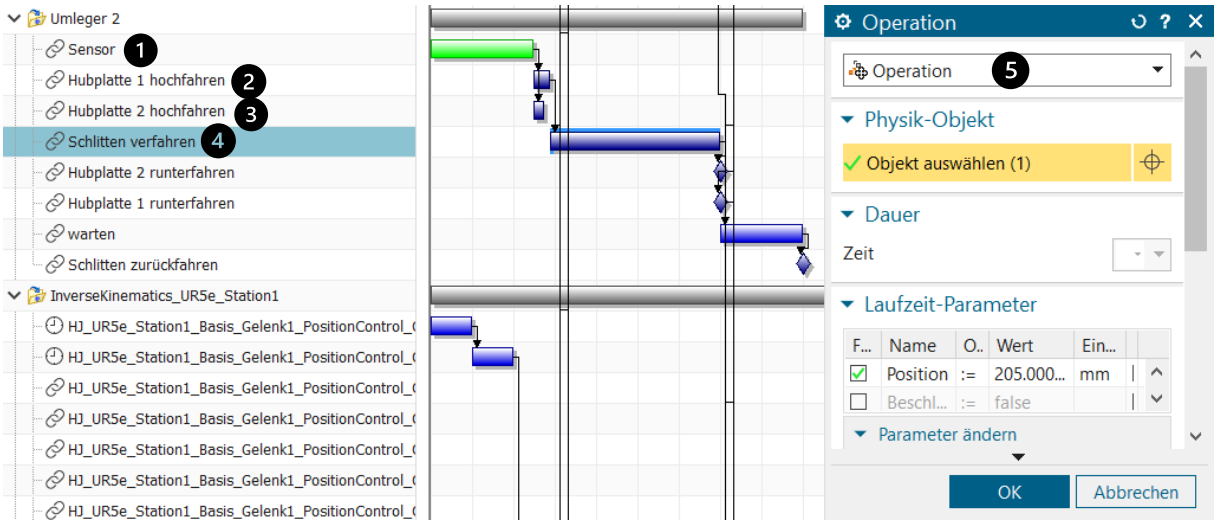


Abbildung 6.6: Verknüpfung der Operationen des Umlegers im Sequenz-Editor

Der Übersicht halber ist in Abbildung 6.7 der Umleger zu sehen, um die Operationen direkt zuordnen zu können. Die erste Operation wird hierbei wieder durch ein Ereignis ausgelöst, was daran erkennbar ist, dass diese im Sequenz-Editor grün dargestellt wird. Dabei wird es ausgelöst, sobald der Sensor (1) vor dem Umleger auslöst. Dies geschieht, wenn das als Kollisionssensor definierte Objekt eine Kollision mit einem Kollisionskörper erkennt, was in diesem Fall der Werkstückträger wäre. Sobald der Sensor auslöst, wird eine gewisse Zeit gewartet, damit der Werkstückträger zunächst

in den Umleger fahren kann. Diese Zeit wurde ebenfalls in der Operation des Sensors hinterlegt. Die nächsten Operationen werden immer erst mit dem Ende der vorherigen Operation ausgeführt.

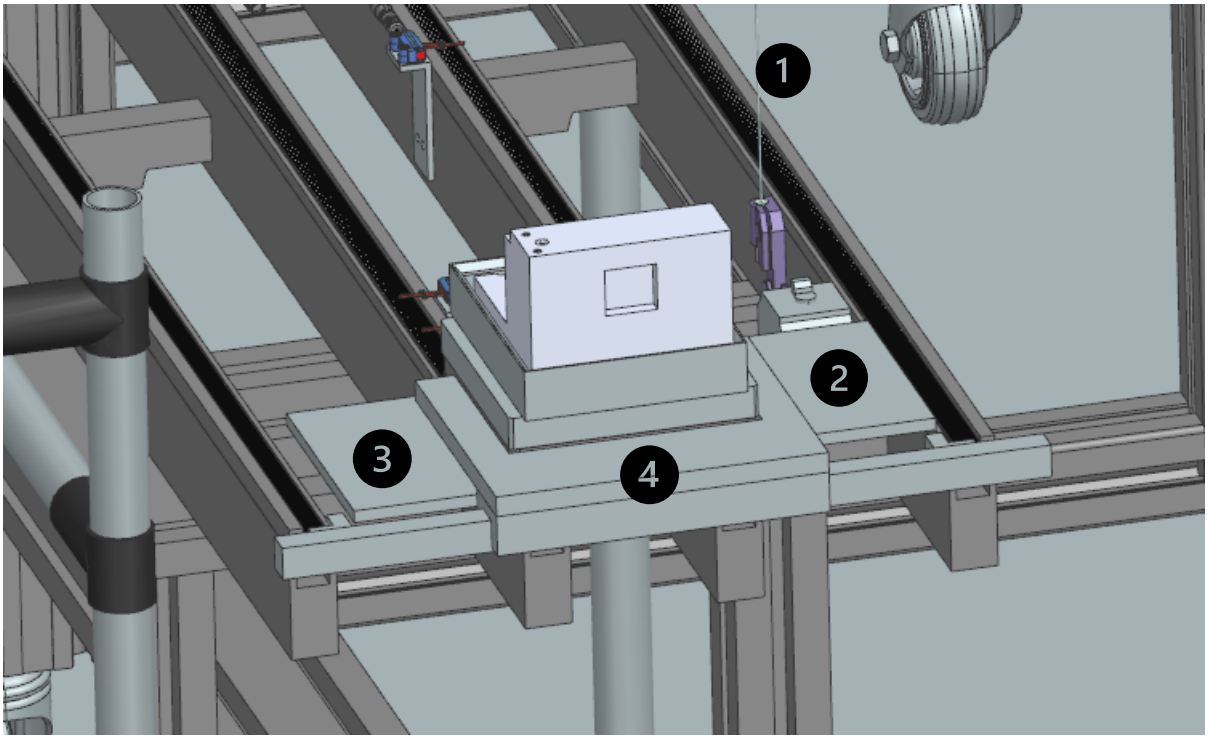


Abbildung 6.7: Übersicht Umlegeprozess

Dies erkennt man wieder an dem Pfeil, welcher vom Ende der vorangegangenen Operation auf den Anfang der nächsten zeigt. Sobald die hinterlegte Zeit abgelaufen ist, werden die beiden Hubplatten (2, 3) hochgefahren, um den Werkstückträger vom Band anzuheben. Anschließend wird der Schlitten (4) und somit auch der Werkstückträger zum anderen Förderband verfahren. In Abbildung 6.6 ist dabei noch zu sehen, wie das Verfahren des Schlittens als Operation definiert wurde (5). Hierbei greift die Operation auf den Positions-Parameter des Positionsreglers des Schlittens zu und ändert den Wert auf 205 mm, was ein Verfahren zum anderen Förderband entspricht. Analog dazu wurden so auch die Operationen der Hubplatten definiert. Nachdem der Schlitten nun zum anderen Förderband verfahren ist, werden die Hubplatten wieder gesenkt und nach einer Wartezeit der Schlitten wieder zurückgefahren.

Das Werkstück fährt anschließend bis zum Stopper der Station 3, dem Handarbeitsplatz. Hier wird das Werkstück an der realen Anlage von einem Mitarbeiter entnommen und mit weiteren Baugruppen händisch zusammengesetzt. An dieser

Stelle wurde in der Simulation lediglich eine Senke und eine Quelle definiert. Sobald das Werkstück den Stopper an der Station 3 erreicht, verschwindet das Werkstück (Senke) und es wird das bereits fertig montierte Werkstück auf dem Werkstückträger platziert (Quelle). Sobald das montierte Werkstück auf dem Werkstückträger erzeugt wurde, wird der Stopper eingefahren und das Werkstück fährt bis zum Stopper der Station 5. Ein Sensor am Band wird beim Passieren des Werkstücks ausgelöst, was wiederum den letzten Bewegungsablauf des Roboters an Station 1, wie bereits anfangs erwähnt, aktiviert. Der Roboter greift dann das Werkstück und setzt dieses auf den mobilen Huskey Roboter ab. In Abbildung 6.8 ist auch dieser Prozess nochmal bildlich dargestellt.

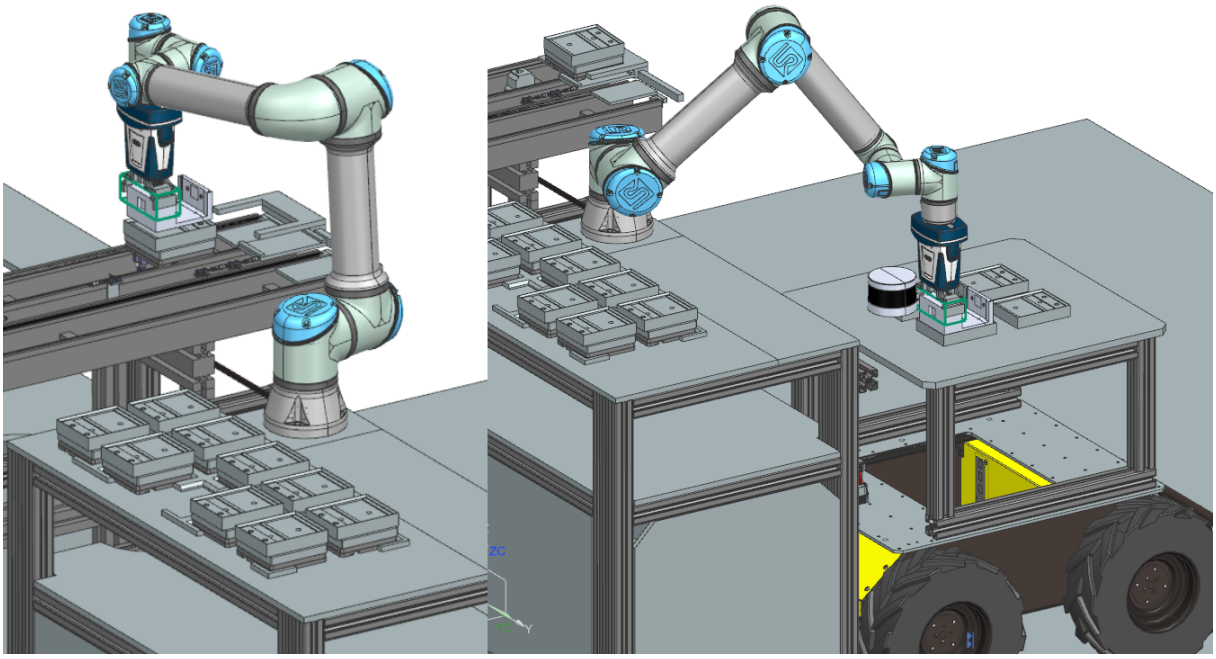


Abbildung 6.8: Aufnehmen und Ablegen des fertig montierten Werkstücks

Geleichzeitig wird der Stopper gelöst und der leere Werkstückträger fährt in den zweiten Umleger, wo er wieder auf das erste Band umgesetzt wird und anschließend wieder bis zum Stopper der Station 1 fährt.

6.3 Schwierigkeiten bei der Umsetzung

An dieser Stelle soll kurz auf Schwierigkeiten bei der Umsetzung eingegangen werden, um eventuell Verbesserungspotenzial für weitere Studienarbeiten am digitalen Zwilling aufzuzeigen.

Roboter:

Die Umsetzung der Roboter hat mitunter am meisten Zeit in Anspruch genommen. Es ist aufgefallen, dass der Roboter nicht immer zu den gewünschten Posen verfährt und zum Teil sehr unrealistische Bewegungen ausführt. Selbst wenn zwischen 2 Posen nur eine Drehung von 90 Grad oder eine Lineare Bewegung von 20 cm erfolgen sollte, konnte es sein, dass er sich einmal um sich selbst verdrehte oder sonstige komische Bewegungen ausgeführt hat. Der Ursprung dieses Verhaltens konnte im Rahmen der Studienarbeit nicht herausgefunden werden. Ein weiterer Mehraufwand war damit verbunden, dass die Verlinkungen der Operationen des Roboters verschwanden, wenn man neue Posen hinzufügte oder auch nur angepasst hat. Dabei mussten die Verlinkungen jedes Mal neu erstellt werden. Eventuell gibt es hier eine Abhilfe, welche jedoch trotz Recherche nicht gefunden wurde. Außerdem kann man für einen Roboter nicht mehrere inverse Kinematik Funktionen erstellen, beispielsweise um mehrere Unterfunktionen eines Roboters zu erstellen. So muss man alle Posen, welche der Roboter anfahren muss, in einer Funktion hinterlegen, was bei einer größeren Simulation nicht mehr händelbar wäre, da jede Operation dadurch immer wieder neu verlinkt werden müsste, wenn neue Posen hinzukämen.

Werkstückträger:

Es kam immer mal wieder vor, dass sich der Werkstückträger im Förderband verkantet hat. Abhilfe konnte zunächst dadurch geschaffen werden, indem die Reibwerte der Förderband-Führungen und der Außenkanten des Werkstückträgers angepasst wurden. Hierfür wurde ein neues Kollisionsmaterial definiert, welches keine Reibung aufweist. Dieses wurde dann den Kollisionskörpern der Führungen sowie der Außenkanten des Werkstückträgers zugewiesen.

6.4 Mögliche Schnittstelle zur virtuellen SPS

Damit der digitale Zwilling zukünftig über eine virtuelle SPS gesteuert werden kann, wird nun eine mögliche Schnittstelle aufgezeigt, über welche man Signale zwischen dem digitalen Zwilling und einer virtuellen SPS austauschen kann. Hierfür wurde eine S7-Kommunikation über PLCSIM Advanced konfiguriert. Die Kommunikation mit externen Programmen erfolgt in MCD über die Definition und Zuordnung von Signalen. Im Folgenden werden einige Befehle erläutert, welche für den Signalaustausch von Bedeutung sind. Anschließend wird die Umsetzung der Schnittstelle am Beispiel eines der Umleger aufgezeigt.

Signal:

Mit dem Befehl „Signal“ kann man im Modell ein Signal erzeugen, um die physikalischen Eigenschaften eines Objekts über einen Laufzeitausdruck zu steuern. Ein Laufzeitausdruck ist ein nicht statischer Wert, der sich während der Laufzeit einer Simulation ändern kann. Dieser Ausdruck ist entweder intern mit MCD verknüpft oder wird durch eine Verknüpfung mit einem Signal aus einer externen Quelle, wie z. B. PLCSIM Advanced, bestimmt.

Symboltabelle:

Über den Befehl „Symboltabelle“ wird eine Liste von Symbolen erstellt, die zur eindeutigen Benennung von Signalen verwendet werden. Man kann eine Symboltabelle auch aus einer externen Quelle, wie z.B. TIA Portal, importieren.

Signaladapter:

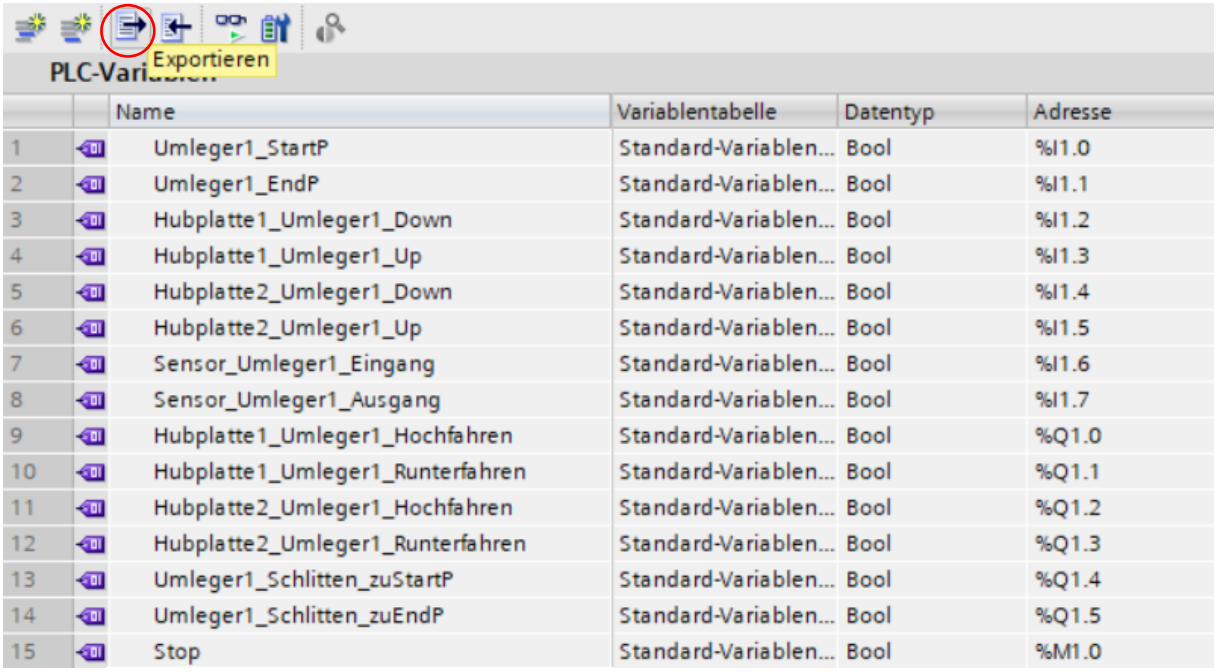
Mit Hilfe des Signaladapters können Signale und Laufzeitausdrücke miteinander verknüpft werden, wobei mehrere Signale und Laufzeitausdrücke pro Signaladapter möglich sind. Darüber hinaus lassen sich über diesen Befehl sowohl Signale als auch Laufzeitausdrücke erstellen.

Signalzuordnung:

Als letztes werden über den Befehl „Signalzuordnung“ die Signale aus MCD mit den Signalen aus TIA bzw. PLCSIM Advanced verknüpft. [1] [3]

Umsetzung am Beispiel des Umlegers:

Damit die Schnittstelle im Verlauf der Umsetzung auch getestet und validiert werden kann, wurde zunächst in TIA Portal ein Programmcode erstellt, welcher den Umleger später steuern soll. Es wurde zudem bereits in TIA eine PLC-Variablentabelle erstellt, worin sämtliche Ein -und Ausgangssignale definiert wurden, welche für das Steuern des Umlegers benötigt werden. Um den Vorgang der Signalerstellung in MCD zu erleichtern, wurden sämtliche Signale aus TIA Portal als .xlsx File exportiert. Abbildung 6.9 zeigt alle definierten Variablen/Signale sowie rot eingekreist, wo man klicken muss, um die Signale zu exportieren.



		Name	Variablentabelle	Datentyp	Adresse
1		Umleger1_StartP	Standard-Variablen...	Bool	%I1.0
2		Umleger1_EndP	Standard-Variablen...	Bool	%I1.1
3		Hubplatte1_Umleger1_Down	Standard-Variablen...	Bool	%I1.2
4		Hubplatte1_Umleger1_Up	Standard-Variablen...	Bool	%I1.3
5		Hubplatte2_Umleger1_Down	Standard-Variablen...	Bool	%I1.4
6		Hubplatte2_Umleger1_Up	Standard-Variablen...	Bool	%I1.5
7		Sensor_Umleger1_Eingang	Standard-Variablen...	Bool	%I1.6
8		Sensor_Umleger1_Ausgang	Standard-Variablen...	Bool	%I1.7
9		Hubplatte1_Umleger1_Hochfahren	Standard-Variablen...	Bool	%Q1.0
10		Hubplatte1_Umleger1_Runterfahren	Standard-Variablen...	Bool	%Q1.1
11		Hubplatte2_Umleger1_Hochfahren	Standard-Variablen...	Bool	%Q1.2
12		Hubplatte2_Umleger1_Runterfahren	Standard-Variablen...	Bool	%Q1.3
13		Umleger1_Schlitten_zuStartP	Standard-Variablen...	Bool	%Q1.4
14		Umleger1_Schlitten_zuEndP	Standard-Variablen...	Bool	%Q1.5
15		Stop	Standard-Variablen...	Bool	%M1.0

Abbildung 6.9: Übersicht PLC-Variablen in TIA Portal

Auf die Logik des SPS-Steuerungsprogramms des Umlegers wird hier nicht näher eingegangen, da diese nur zum Test der Schnittstelle erstellt wurde. Nachdem die Signale exportiert wurden, können diese nun über den Befehl „Symboltabelle“ in MCD importiert werden (siehe Abbildung 6.10). Es macht Sinn, dass die Signale (Symbolname) in MCD genau die gleiche Benennung wie die externen Signale (TIA bzw. PLCSIM) haben, denn dadurch kann die Signalzuweisung später automatisch mit nur einem Klick erfolgen.

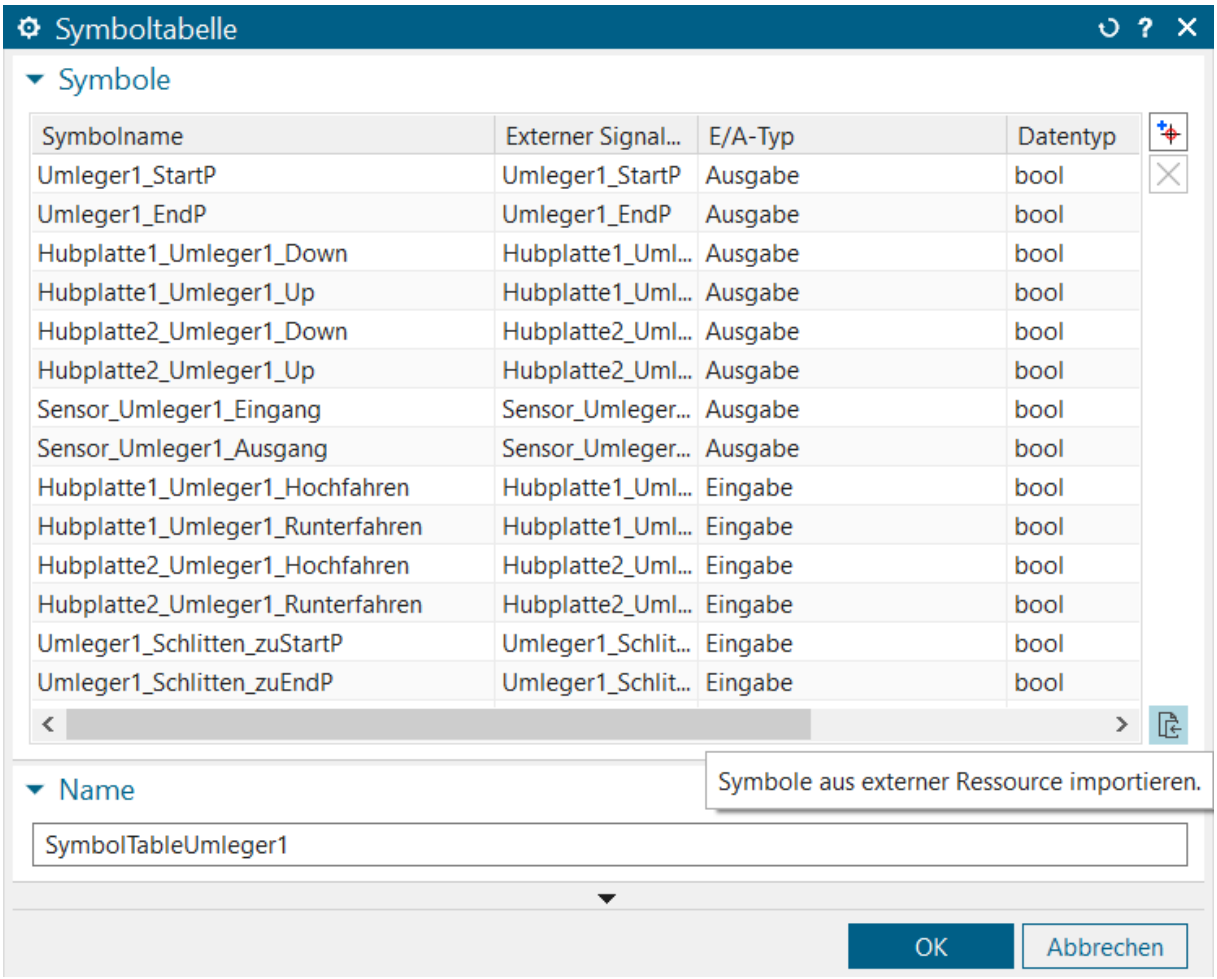


Abbildung 6.10: Symboltabelle der Signale in NX MCD

Ist die Symboltabelle erstellt, definiert man danach einen Signaladapter. Klickt man auf diesen Befehl, so öffnet sich das in Abbildung 6.11 dargestellte Fenster. Der Signaladapter für den Umleger ist hier schon fertig konfiguriert. Es soll dennoch anhand des Signals zum Hochfahren der Hubplatte1 (blau markiert) erläutert werden, wie der Signaladapter zu konfigurieren ist. Unter dem Register „Parameter“ (rot eingekreist) wählt man einen Parameter einer dynamischen Eigenschaft aus, welche mit einem Signal verbunden werden soll. Für das Hochfahren der Hubplatte1 wird als Parameter die Eigenschaft „Aktiv“ des Positionsreglers „PC_Hubplatte1_Umleger1_Hochfahren_SPS“ der Hubplatte1 ausgewählt. Mit dieser Eigenschaft lässt sich der Positionsregler später aktivieren und deaktivieren, wobei die im Positionsregler definierten Eigenschaften aktiv geschaltet werden. Im Falle der Hubplatte1 wäre dies ein Verfahren der Hubplatte1 um 20 mm nach oben. Anschließend wird noch ein eindeutiger Parametername vergeben. Um die Parameter von den Signalen später unterscheiden zu können, beginnen diese mit dem Kürzel „pa“, was für „Parameter“ stehen soll. Bei dem nun erstellten Parameter

„pa_Hubplatte1_Umleger1_Hochfahren“ lässt sich auch ablesen, dass es sich um einen boolschen Datentyp handelt, welcher nur geschrieben werden kann (erkennbar an dem „W“ unter „Lesen/Schreiben“).

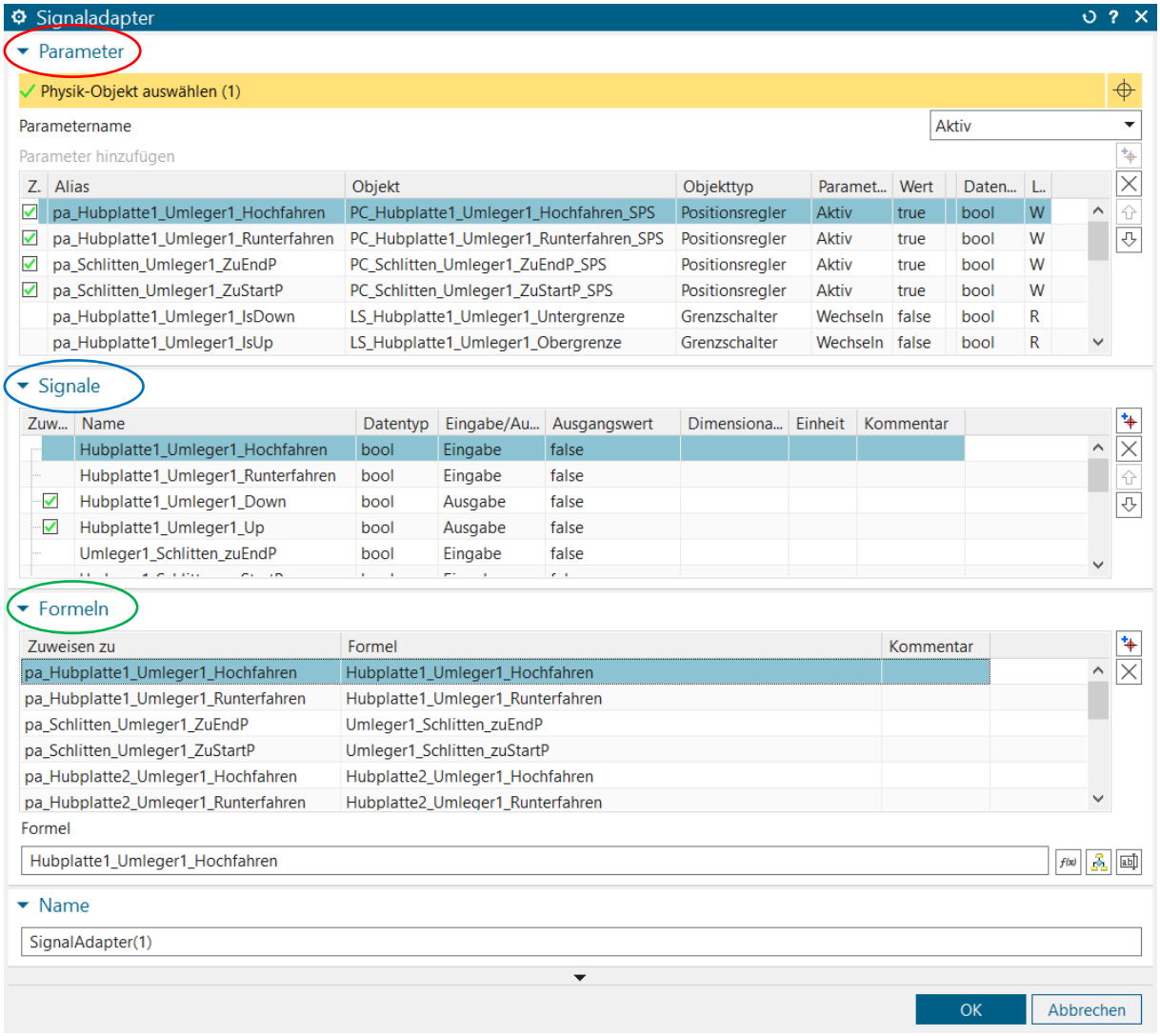


Abbildung 6.11: Definition des Signaladapters in NX MCD

Unter dem Reiter „Signale“ (blau eingekreist) kann dann ein zugehöriges Signal hinzugefügt werden, mit dem der Parameter verbunden werden soll. Klickt man hierbei auf das Symbol zum Hinzufügen neuer Signale, so zeigt es bereits eine Liste von Signalen an, welche zuvor durch die Symboltabelle importiert wurden. Für das Hochfahren der Hubplatte1 wird hier das Signal „Hubplatte1_Umleger1_Hochfahren“ ausgewählt. Dabei ist zu beachten, dass der Datentyp mit dem des zu verknüpfenden Parameters übereinstimmt und dass der „Eingabe/Ausgabe“ Wert ebenfalls zum „Lesen/Schreiben“ Wert des Parameters passt. Für einen Parameter, der geschrieben werden muss, muss das Signal ein Eingang von einer externen

Quelle sein. In diesem Fall haben beide einen booleschen Datentyp, und da der Parameter nur geschrieben werden kann, ist die Definition des Signals als Eingang angemessen. Der Parameter und das Signal müssen nun noch logisch miteinander verknüpft werden, was unter dem Reiter „Formeln“ (grün eingekreist) definiert wird. Hier könnte man komplizierte Verknüpfungen definieren oder beispielsweise einen Parameter von mehreren Signalen abhängig machen. Da aber die Logik Teil des Automatisierungsprogramms sein soll und nicht des digitalen Zwillings, wurde an dieser Stelle eine einfache Zuweisung umgesetzt. Die Formel für den Parameter zum Hochfahren der Hubplatte1 ist also einfach nur das Signal „Hubplatte1_Umleger1_Hochfahren“, was einer Zuweisung entspricht. Damit wäre nun ein Signal zum Steuern des digitalen Zwillings erstellt. Diese Schritte müssen dann noch für alle weiteren benötigten Signale gemacht werden. Sind alle Signale definiert worden, kann der Signaladapter mit einem Klick auf „OK“ erstellt werden.

Damit im letzten Schritt eine Signalzuordnung zwischen der virtuellen SPS und MCD erfolgen kann, muss bereits eine Instanz der SPS in PLCSIM Advanced in Betrieb sein. Sobald eine Instanz läuft und auch das Automatisierungsprogramm in die Instanz geladen wurde, kann man den Befehl „Signalzuordnung“ anklicken, womit sich das in Abbildung 6.12 dargestellte Fenster öffnet. Oben rechts wählt man zunächst als externen Signaltyp „PLCSIM-Assistent“ aus. Anschließend wählt man darunter die PLCSIM Advanced Instanz aus, auf welcher das entsprechende Automatisierungsprogramm läuft. Sollte dies das erste Mal sein, dass man eine Signalzuordnung erstellt, so wird zunächst keine Instanz angezeigt. Durch einen Klick auf das grün eingekreiste Symbol öffnet sich ein Fenster, in welchem sich die gewünschte Instanz auswählen lässt und die dazugehörigen Variablen/Signale für die Signalzuordnung freigeben lassen. Nachdem nun die richtige Instant ausgewählt wurde, werden in der Signalzuordnung alle MCD-Signale links und alle Externen Signale (PLCSIM) rechts aufgelistet. Nun muss eine Zuordnung zwischen den internen und externen Signalen erfolgen. Nun spielt sich der Vorteil aus, dass die internen Signale gleich benannt wurden, wie die externen Signale. Durch einen Klick auf „Automatische Zuordnung durchführen“ (rot eingekreist) werden die internen Signale automatisch mit den externen Signalen verknüpft. Die Verknüpfung wird dann im unteren Teil unter „Zugeordnete Signale“ aufgelistet, wobei man diese

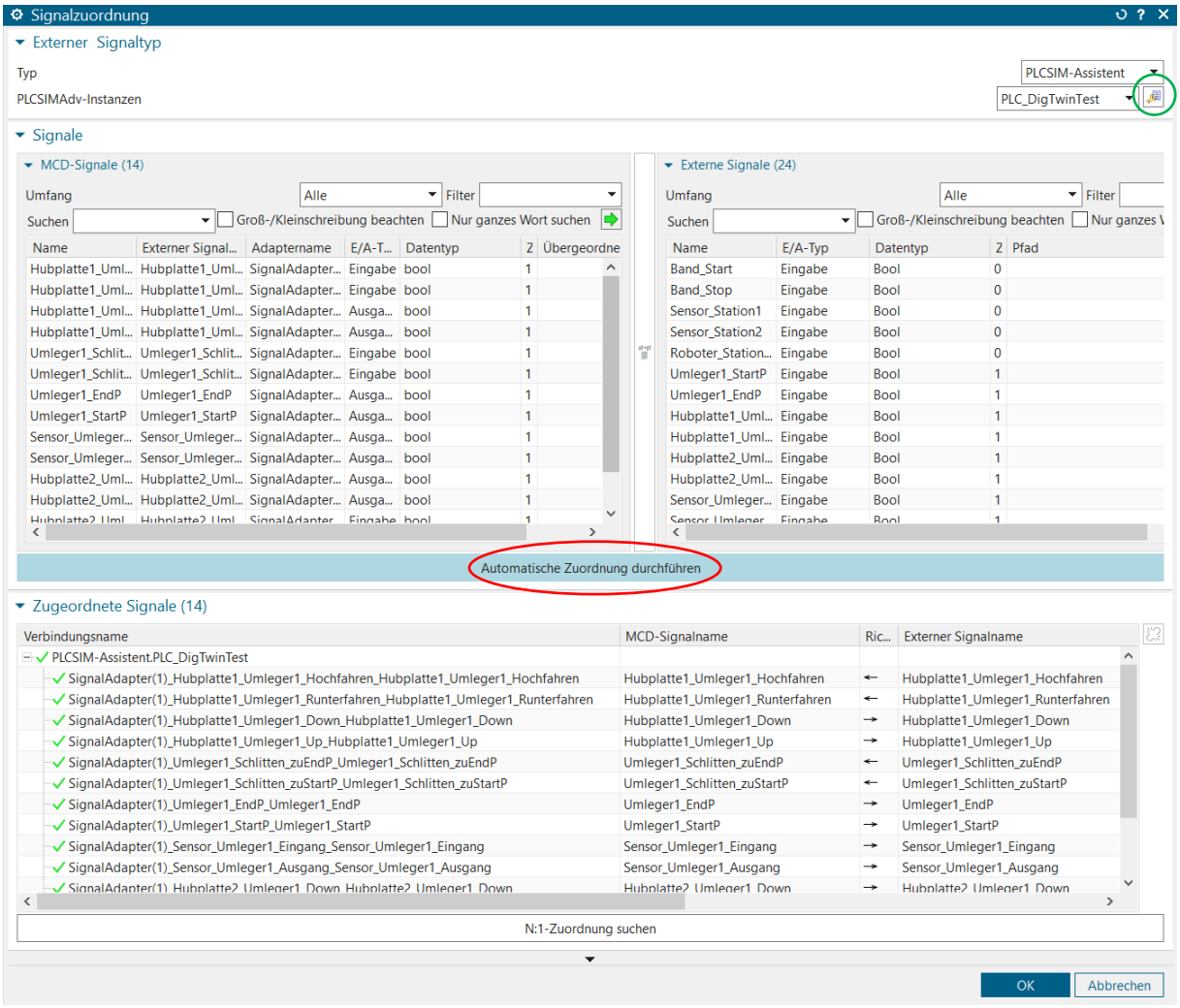


Abbildung 6.12: Signalzuordnung in NX MCD zwischen PLCSIM und MCD

nochmal kontrollieren kann. Mit einem Klick auf „OK“ ist die Signalzuordnung abgeschlossen und die Simulation kann gestartet und getestet werden.

7 Fazit und Ausblick

Im Rahmen dieser Studienarbeit wurde nun das statische 3D-Modell des digitalen Zwillings mit allen nötigen, dynamischen Eigenschaften versehen, um die Abläufe und Bewegungen des realen Produktionsnetzwerks abbilden zu können.

Das dynamische 3D-Modell konnte durch die Umsetzung einer Simulation in MCD getestet werden. Hierbei konnte bereits der Ablauf des Montage Prozesses simuliert werden, wobei Schwierigkeiten aufgezeigt und zum Teil beseitigt werden konnten.

Für die Kommunikation zwischen dem digitalen Zwilling und der SPS wurde eine Schnittstelle definiert und getestet, womit sich Signale austauschen lassen und somit der digitale Zwilling über eine virtuelle SPS steuern lassen kann. Das Steuern des Umlegers über die umgesetzte Schnittstelle wurde ausgiebig simuliert und konnte validiert werden. Zukünftig kann diese Schnittstelle genutzt werden, um den digitalen Zwilling mit dem SPS-Programm der realen Anlage zu steuern.

Mögliche nächste Schritte wären die Umsetzung des Demontageprozesses sowie die Nutzung des digitalen Zwillings, um Anpassungen und Erweiterungen vorab zu testen, bevor sie an der realen Anlage umgesetzt werden. Darüber hinaus wäre es für die nächsten Studienarbeiten denkbar, an einer Schnittstelle zwischen dem UR5e-Roboter und MCD zu forschen, damit die Roboter nicht mehr in MCD simuliert werden müssen, was sehr aufwändig und fehleranfällig ist.

Literaturverzeichnis

[1] S. AG, „Siemens.com," Siemens, 2021. [Online]. Available: https://docs.sw.siemens.com/de-DE/doc/209349590/PL20200522120320484.mechatronics/id1101745. [Zugriff am 4 Juli 2024].

[2] H. S. Darmstadt, „Erstellung eines dynamischen 3D-Modells mithilfe MCD," 2020. [Online]. Available: https://support.industry.siemens.com/cs/attachments/109816686/lp-sce-digital-twin-course-mcd-lm-5-de.zip.

[3] H. S. Darmstadt, „Signalerstellung für ein 3D-Modell im MCD," 2020. [Online]. Available: https://support.industry.siemens.com/cs/attachments/109816686/lp-sce-digital-twin-course-mcd-lm-6-de.zip.