

Aufbau virtueller Versuchsanlagen und deren Nutzung in Lehrveranstaltungen zur Automatisierungstechnik

M. Othman¹, C. Stöcker²

Zusammenfassung

In der Ingenieurausbildung ist die Arbeit an Anlagen und Maschinen ein wesentlicher Baustein, um die Anwendbarkeit, Möglichkeiten und Grenzen der in der Theorie vermittelten Methoden an praktischen Problemstellungen zu erproben und einschätzen zu lernen. Zu diesem Zweck werden an vielen Hochschulen und Universitäten unterschiedliche Versuchsanlagen eingesetzt. Die Errichtung und der Betrieb solcher Anlagen erfordern jedoch Platz und sind typischerweise mit hohem Einsatz personeller und finanzieller Ressourcen verbunden. Dieser Beitrag betrachtet den Einsatz virtueller Versuchsanlagen als eine mögliche Ergänzung der bestehenden Laborinfrastruktur. Virtuelle Anlagen sind solche, die nicht physisch existieren, sondern deren dynamisches Verhalten simuliert und in einer 3D-Visualisierung auf einem Monitor oder in einer VR-Brille dargestellt werden kann. Virtuelle Anlagen haben gegenüber realen Versuchsanlagen viele Vorteile in Bezug auf die Dauer und Kosten für deren Realisierung, sowie deren Erweiterung und Anpassung an neue Lehrinhalte und die Skalierung und Vervielfältigung für den Einsatz in Praktika. Auf der anderen Seite müssen Laboringenieure für die Entwicklung, Pflege und Instandhaltung virtueller Anlagen über andere Kompetenzen (insb. im Bereich der IT und Programmierung) verfügen, als bei realen Anlagen. Zudem wird für den Betrieb solcher Anlagen zusätzliche Soft- und Hardware benötigt. Am Beispiel eines virtuellen Aufzugs beschreibt dieser Beitrag im Einzelnen diese Vor- und Nachteile und gibt Hinweise für den Einsatz einer virtuellen Anlage in Praktika und Übungen.

Stichwörter

Virtuelles Labor, Virtuelle Versuchsanlage, Simulation, Virtual Reality, 3D-Visualisierung, Unity, game4automation, TwinCAT3, Steuerungstechnik, Regelungstechnik, Aufzugsanlage

1 Einleitung

1.1 Motivation

Labor- und Versuchsanlagen sind in der Ingenieurausbildung ein wichtiges Mittel, um theoretisch erlerntes Wissen in die Praxis umzusetzen und die Anwendbarkeit und Grenzen von Methoden an praktischen Problemstellungen zu erfahren. Der finanzielle und personelle Aufwand für die Errichtung und den Betrieb dieser Anlagen ist jedoch häufig groß. Änderungen an bestehenden Anlagen sind entweder nicht umsetzbar oder sie erfordern umfangreiche Planungsvorleistungen sowie erneuten Einsatz finanzieller und personeller Ressourcen. Wenn also Veränderungen in den Lehrinhalten oder den Lernzielen, Anpassungen von Laboranlagen erforderlich machen, sind diese Umbauten selten kurzfristig möglich. Eine Ergänzung zur bestehenden Laborinfrastruktur können dann virtuelle Versuchs-

¹ Moaid Othman, M. Sc., FH Bielefeld, moaid_ismail.othman@fh-bielefeld.de

² Prof. Dr.-Ing. Christian Stöcker, FH Bielefeld, christian.stoecker@fh-bielefeld.de

anlagen sein. Dies sind Simulationsmodelle technischer Anlagen (die nicht zwingend real existieren müssen), deren dynamisches Verhalten realitätsnah ist. Genauso wie bei realen Anlagen, kann das Verhalten dieser Modelle durch Eingangssignale beeinflusst und durch Ausgangssignale erfasst werden. Über geeignete Schnittstellen ist es also möglich, virtuelle Anlagen mit realen Steuerungen zu verbinden und die entwickelte Automatisierungssoftware daran zu testen und zu erproben. Bei Einsatz einer VR-Brille kann darüber hinaus das Verhalten der automatisierten Anlage in der virtuellen Realität erfahren werden.

1.2 Zielstellung und Inhalte des Beitrags

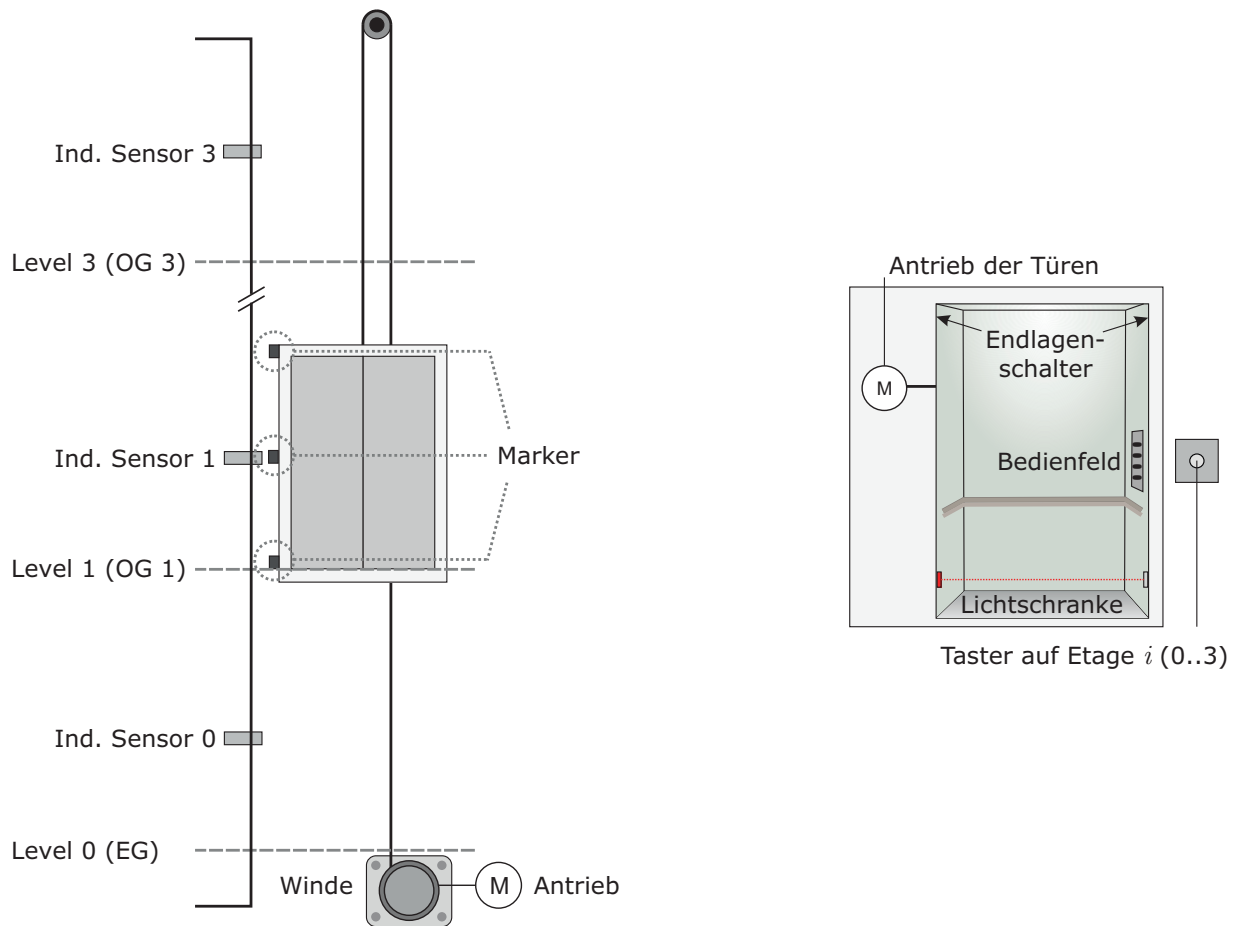
Das Ziel dieses Beitrags ist, die Erfahrungen, die bei der Konzeptionierung, Realisierung und Anwendung einer virtuellen Anlage gesammelt wurden, zu beschreiben. Es sollen Informationen gegeben werden, die eine mögliche Entscheidungsfindung bzgl. der Erstellung virtueller Versuchsanlagen in Institutionen zur Ausbildung von Automatisierungsingenieuren unterstützen. Das Beispiel eines Prototyps einer virtuellen Anlage, anhand dessen die Autoren ihre Erfahrungen beschreiben, ist ein virtueller Aufzug, der in Kapitel ?? spezifiziert wird. Hier werden zudem die für den Aufbau der Anlage verwendete Soft- und Hardware, die für die Entwicklung benötigten Kompetenzen sowie eine Abschätzung der entstandenen Kosten und der Entwicklungsdauer angegeben. In Kapitel ?? wird dann eine mögliche Praktikumsaufgabe an dem virtuellen Aufzug dargestellt und es werden unterschiedliche Varianten eines möglichen Praktikumsaufbaus erklärt, welche die Verfügbarkeit spezieller Hardware (Industrie-PC mit Soft-SPS und VR-Brille) berücksichtigen. In Kapitel ?? werden schließlich die Argumente für und wider den Einsatz virtueller Anlagen in der Ingenieurausbildung zusammengefasst. Es sei hier erwähnt, dass die Idee, neu entwickelten Programmcode der Automatisierungstechnik an Simulationsmodellen zu erproben, nicht neu ist. Das Testen von Steuerungscode an Simulationen ist unter dem Begriff *virtuelle Inbetriebnahme* bekannt und wird seit Anfang der 2000er Jahre Wissenschaft diskutiert [?, ?] und seit einigen Jahren auch industriell eingesetzt [?]. Zudem werden virtuelle Anlagen auch für Schulungen von Anlagenbedienern eingesetzt [?].

2 Aufbau einer virtuellen Anlage

2.1 Beispiel: Aufzugsanlage

Das Beispiel, anhand dessen im Folgenden das Vorgehen für die Realisierung einer virtuellen Anlage erläutert wird, ist ein Aufzug, der in Bild ?? schematisch dargestellt ist. Das Teilbild (a) veranschaulicht die Umsetzung der vertikalen Bewegung und der Positionierung des Fahrkorbs. Der Aufzug soll über vier Etagen (Erdgeschoss EG, 1. Obergeschoss OG1, 2. Obergeschoss OG2 und 3. Obergeschoss OG3) fahren, von denen in Bild ??(a) nur EG, OG1 und OG3 gezeigt werden. In jedem Geschoss ist ein induktiver Sensor eingebaut. Diese Sensoren dienen zur Positionierung des Aufzugs und zur Steuerung der Fahrgeschwindigkeit beim Beschleunigen und Abbremsen. An dem Aufzug sind oben, in der Mitte und unten jeweils ein metallischer Marker montiert, die von den induktiven Sensoren erfasst werden, wenn sich der Aufzug daran vorbei bewegt. Die Bewegung des Fahrkorbs wird über eine motorisierte Winde realisiert. Der Antrieb, der die Winde bewegt, wird über zwei Signale angesteuert: Ein binäres Enable-Signal aktiviert oder deaktiviert den Antrieb und über ein weiteres analoges Signal kann eine Geschwindigkeit eingestellt werden.

Das Teilbild (b) zeigt eine Detailansicht des Fahrkorbs. Die Türen lassen sich über einen Antrieb öffnen und schließen, wobei deren Positionen über Endlagenschalter erfasst werden. Der hier eingesetzte Antrieb wird über zwei binäre Signale angesteuert. Über ein Enable-Signal kann der Antrieb aktiviert oder deaktiviert werden und über ein weiteres Signal kann die Bewegungsrichtung vorgegeben werden. Der Antrieb bewegt sich nur mit fest eingestellter Geschwindigkeit; eine Geschwindigkeitsregelung ist hierbei nicht vorgesehen. Eine Lichtschranke im Zugang zum Fahrkorb soll Hindernisse im Bereich der Türen erkennen. Im Fahrkorb befindet sich ein Bedienfeld, über das einerseits der Fahrwunsch in die einzelnen Ebenen eingegeben werden kann und andererseits eine manuelle Steuerung



(a) Positionierung des Fahrkorbs

(b) Detailansicht des Fahrkorbs

Bild 1: Schematische Darstellung der Aufzugsanlage

der Türen vorgenommen werden kann. Auf jeder Ebene befindet sich zudem ein Taster, über den ein Fahrwunsch angezeigt und entsprechend der Fahrkorb gerufen werden kann.

Zusätzlich zu den Türen des Fahrkorbs gibt es noch auf jede Ebene eine Tür, die den Aufzugsschacht von der jeweiligen Ebene trennt, die aber in Bild ?? nicht eingezeichnet sind. Diese Türen werden nur geöffnet, wenn der Fahrkorb in der entsprechenden Ebene hält. Andernfalls müssen diese Türen immer geschlossen sein. Der in diesen Türen eingesetzte Antrieb ist der gleiche Typ, wie der Antrieb der Türen im Fahrkorb.

Die für eine Automatisierung der Aufzugsanlage verwendbaren Ein- und Ausgangssignale werden aus Sicht einer Steuerung in Tabelle ?? zusammengefasst. In der Tabelle werden grob drei Typen von Signalen unterschieden: digitale Eingänge (DI), digitale Ausgänge (DO) und analoge Ausgänge (AO). Insgesamt verfügt die Anlage über einen analogen und 11 digitale Ausgänge sowie über 26 digitale Eingänge.

2.2 Software-Tools für die Entwicklung der virtuellen Anlage

Für die Realisierung des zuvor beschriebenen Aufzugsystems als digitale Anlage wird die Spiele-Engine Unity verwendet. Unity ist ursprünglich als Entwicklungs- und Laufzeitumgebung für Spiele entwickelt worden, wird jedoch zunehmend auch in anderen Bereichen eingesetzt. In [?] wird erklärt, wie VR-Applikationen mit Unity erstellt und für die Ausbildung von Studierenden in der Mechatronik verwendet werden können. Zudem werden in Unity erstellte Simulationsmodelle auch im industriellen Umfeld eingesetzt, wie z. B. [?] beschreibt.

Ein besonderer Nutzen von Unity für den Aufbau virtueller Anlagen liegt in der *Physics Engine*. Dies ist

Tabelle 1: IO-Signale der Aufzugsanlage

| Beschreibung | Anzahl | Typ |
|---|--------|-----|
| Antrieb der Winde: Enable | 1 | DO |
| Antrieb der Winde: Velocity | 1 | AO |
| Positionserfassung des Fahrkorbs mit induktivem Näherungssensor | 4 | DI |
| Antrieb der Türen des Fahrkorbs: Enable | 1 | DO |
| Antrieb der Türen des Fahrkorbs: Open/Close | 1 | DO |
| Antrieb der Türen in den Ebenen: Enable | 4 | DO |
| Antrieb der Türen in den Ebenen: Open/Close | 4 | DO |
| Endlagenschalter der Türen des Fahrkorbs: isOpen/isClosed | 2 | DI |
| Endlagenschalter der Türen auf den Ebenen: isOpen/isClosed | 8 | DI |
| Lichtschranke | 1 | DI |
| Taster auf dem Bedienfeld im Fahrkorb | 7 | DI |
| Taster zur Angabe des Fahrwunsches auf jeder Ebene | 4 | DI |

ein Softwarepaket, das Objekten in der virtuellen Welt ein an realen physikalischen Gesetzmäßigkeiten angepasstes Verhalten gibt. So werden in dieser Engine u.a. die Schwerkraft und andere Kräfte (z. B. Reibung) sowie auch Kollisionen simuliert. Es ist also nicht erforderlich, das dynamische Verhalten der virtuellen Anlage mathematisch zu modellieren (z. B. in einem Zustandsraummodell), da also die Dynamik durch die Physics Engine berechnet wird. Dies ist ein wesentlicher Vorteil gegenüber anderen Simulationswerkzeugen, in denen das zu simulierende Verhalten mathematisch beschrieben werden muss, mit der Konsequenz, dass für die Simulation typischerweise nur ein stark vereinfachtes Modell verwendet wird und daher nur ein kleiner Teil aller Effekte, die in einer realen Anlage auftreten können, erfahrbar gemacht werden.

In Unity kann folglich die virtuelle Anlage zusammen mit ihrer Umgebung aufgebaut werden, wobei das dynamische Verhalten frei beweglicher Komponenten in dem gesamten Modell grundsätzlichen physikalischen Gesetzmäßigkeiten folgt. In dem betrachteten Beispiel wird i. W. ein Hochhaus mit vier Ebenen, einem Aufzugsschacht und dem Fahrkorb realisiert (vgl. Bild ??). Das dynamische Verhalten, das durch die Aktorik in dem Modell erzwungen werden soll, wird durch die o. g. Physics Engine jedoch nicht abgedeckt. Die vertikale Bewegung des Fahrkorbs innerhalb des Aufzugsschachts, wie auch die horizontale Bewegung der Türen (zum Öffnen und Schließen) müssen durch sog. Verhaltensmodelle programmiert werden. Verhaltensmodelle sind Funktionen, in denen die Effekte von Aktoren auf Objekte in dem Modell beschrieben werden. Gleiches gilt auch für die in der virtuellen Anlage verwendete Sensorik. Die Umwandlung simulierter physikalischer Größen in einen Messwert muss ebenfalls durch Verhaltensmodelle programmiert werden. In der Beispielanlage werden solche Verhaltensmodelle für die Sensorik insb. für Näherungssensoren und Taster benötigt. Die Implementierung der Verhaltensmodelle erfolgt in der Programmiersprache C#.

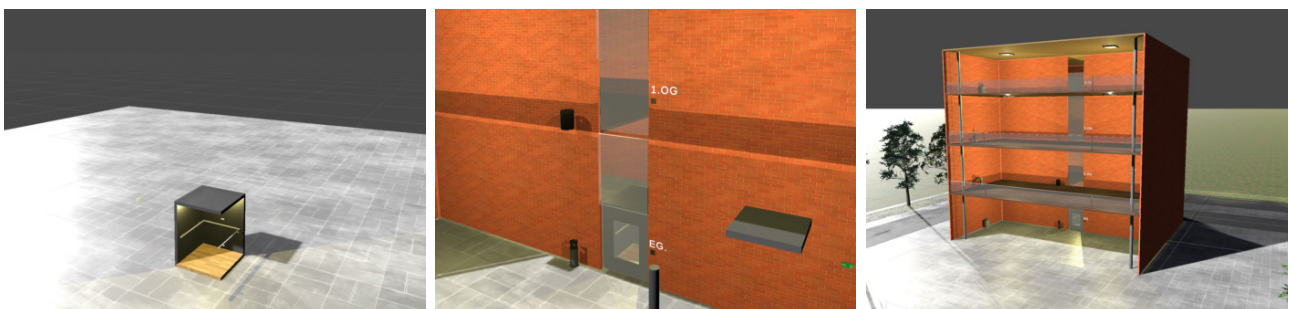


Bild 2: Beginn der Modellierung, ein Zwischenstand und das fertige 3D-Modell

Für die umfassende und realitätsnahe Simulation einer virtuellen Anlage ist natürlich das Verhalten automatisierungstechnisch relevanter Komponenten der Sensorik und Aktorik unerlässlich und damit die Implementierung der Verhaltensmodelle eine Notwendigkeit. Die Tatsache, dass aber diese Verhaltensmodelle für typische Sensor- und Aktortypen in unterschiedlichen Simulationsmodellen wiederkehrend eingesetzt werden können, hat ein Geschäftsmodell eröffnet, das auf der Lizenzierung von Verhaltensmodellen für solche automatisierungstechnischen Standardkomponenten basiert. Alternativ zu der Möglichkeit, diese Funktionen selber zu schreiben, kann man diese auch kaufen. Ein Unternehmen, das sich auf die Programmierung dieser Verhaltensmodelle spezialisiert hat, ist die Firma inSight GmbH, die mit dem Produkt game4automation eine Reihe von Verhaltensmodellen für Sensorik und Aktorik anbietet und das auch in dem hier beschriebenen Projekt zum Einsatz kommt.

2.3 Schnittstelle zwischen virtueller Anlage und SPS

Damit die virtuelle Anlage später auch durch eine reale speicherprogrammierbare Steuerung (SPS) automatisiert werden kann, muss eine Schnittstelle implementiert werden, über die ein Datenaustausch zwischen der Simulation und der SPS erfolgt. Neben den Verhaltensmodellen enthält game4automation solche Schnittstellen für die Kopplung zwischen der virtuellen Anlage in Unity und Steuerungen unterschiedlicher Hersteller. In dem hier erklärten Beispiel wird eine PC-basierte Steuerung der Firma Beckhoff eingesetzt. Die Entwicklungs- und Laufzeitumgebung der Steuerungen dieser Firma ist TwinCAT3. Eine Kopplung zwischen Unity und TwinCAT ist sowohl über OPC/UA, wie auch über das proprietäre Kommunikationsprotokoll ADS möglich.

Der Informationsaustausch zwischen diesen beiden Programme erfolgt über Variablen, was in Bild ?? am Beispiel von zwei Sensorsignalen (bLimitSenClosedInside und bLimitSenOpenInside) und zwei Aktorsignalen (bMotorLiftEnable und nMotorLiftVel) veranschaulicht ist. Auf der linken Seite der Abbildung ist ein Codeausschnitt aus einem SPS-Projekt in TwinCAT dargestellt. Die hierin gelb hervorgehobenen Variablen sind in dem SPS-Projekt in einer globalen Variablenliste deklariert. Auf der rechten Seite von Bild ?? ist ein Ausschnitt der Liste aller in der Simulation verwendeten Signale dargestellt. Die hier dargestellten Signale haben dieselben Bezeichner, wie die Variablen in TwinCAT, was eine notwendige Voraussetzung für die Informationskopplung ist.

Die Signale bLimitSenClosedInside und bLimitSenOpenInside sind in Unity Objekten zugeordnet, die das Verhalten von Endlagenschaltern simulieren. Wenn diese Objekte mit anderen Objekten in Kontakt kommen – der Endlagenschalter also aktiviert ist – wird das zugehörige Signal auf TRUE gesetzt, andernfalls ist dessen Wert FALSE. Durch die Schnittstelle werden diese Signale aus Unity ausgelesen und der jeweilige Wert wird dann in TwinCAT in die gleichnamige Variable geschrieben, wo diese dann, wie im Codeausschnitt in den Zeilen 60 und 61 dargestellt, ausgelesen werden können.

Bei der Ansteuerung von Aktoren funktioniert dieser Informationsaustausch in die umgekehrte Richtung. Die Signale bMotorLiftEnable und nMotorLiftVel sind in Unity einem Objekt zugeordnet, welches das Verhalten eines einfachen Antriebs hat. Wenn bMotorLiftEnable den Wert TRUE hat, wird der Antrieb aktiviert und über nMotorLiftVel kann dann eine Geschwindigkeit vorgegeben

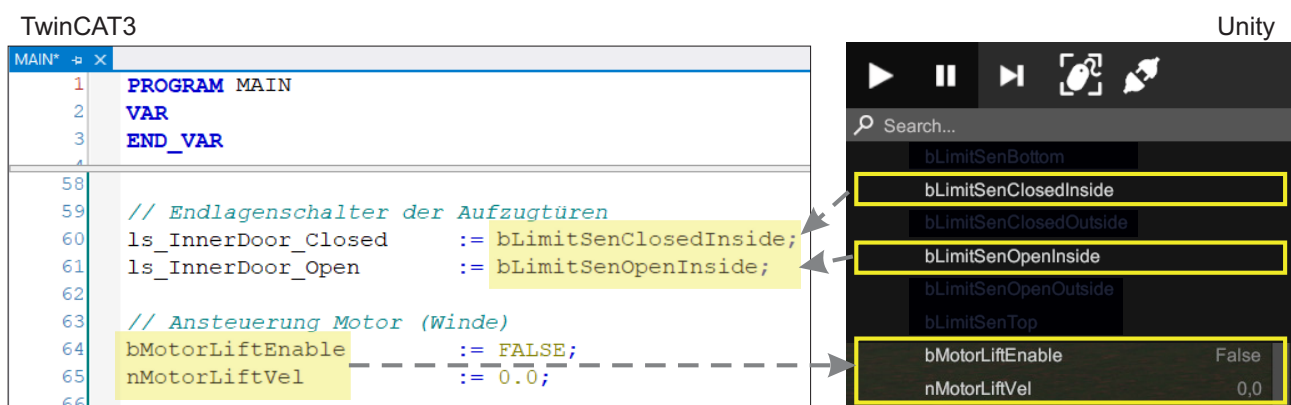


Bild 3: Informationsaustausch zwischen TwinCAT und Unity

werden. Im TwinCAT-Projekt existieren Variablen mit denselben Bezeichnern. Die Schnittstelle liest die Werte dieser Variablen aus und schreibt diese dann in die Signale in Unity, wodurch die Aktoransteuerung realisiert wird.

2.4 3D-Simulation der virtuellen Anlage

Das mit Unity entwickelte Modell der virtuellen Anlage kann in der Laufzeitumgebung von Unity direkt ausgeführt werden und in einer 3D-Visualisierung entweder auf einem normalen Bildschirm oder auf einer VR-Brille angezeigt werden. Interaktive Elemente wie z. B. Taster, können am Bildschirm mit der Maus bedient werden. Bei Einsatz einer VR-Brille können hierfür Touch-Controller verwendet werden. Die vollständig entwickelte virtuelle Anlage, inkl. der Verhaltensmodelle und Schnittstellen zu externen Steuerungen kann aber auch als ausführbare Datei erzeugt werden. Damit ist es möglich, die Simulation auf Rechnern zu starten, auf denen Unity nicht installiert ist und die Simulationen dennoch in vollem Umfang nutzen zu können. Dies ist insb. für die Verwendung der Modelle in online-Praktika wichtig, bei denen Studierende ggf. eigene Hardware verwenden. Diese ausführbaren Dateien sind keinen Lizenzbeschränkungen unterworfen und können daher beliebig oft kopiert werden.

Die Systemanforderungen für die Ausführung des fertigen Projekts orientieren sich an den Anforderungen von Unity. Für die Ausführung der Applikation wird ein Computer mit Windows 10 und einer Grafikkarte mit DirectX 10 oder besser benötigt. Als Entwickler-PC wird ein leistungsstarker Computer empfohlen, insb. wenn eine VR-Version entwickelt werden soll. Zur Orientierung folgt eine technische Beschreibung der Computer, die für die Entwicklung dieses Projekts verwendet wurden. 3D-Entwicklung: Intel Xeon E-2186M, NVIDIA Quadro P3200 6GB Grafikspeiche, 16GB DDR4 Arbeitsspeicher und VR-Entwicklung: Intel Core™ i9 9900K, NVIDIA GeForce RTX 2080 OC mit 8GB GDDR6 Grafikspeicher, 16GB DDR4 Arbeitsspeicher.

2.5 Benötigte Kompetenzen

Für die Entwicklung virtueller Systeme werden unterschiedliche Kompetenzen benötigt, die den Umgang mit Soft- und Hardware betrifft, die typischerweise nicht im klassischen Laborumfeld eingesetzt werden. Die benötigten Kompetenzen sind in der Tabelle ?? zusammengefasst wobei auch das aus Sicht der Autoren notwendige Niveau der Kompetenzerfüllung angegeben ist.

Tabelle 2: Für die Entwicklung einer virtuellen Anlage benötigte Kompetenzen

| Soft- und Hardware | Anwendung | Kompetenz |
|----------------------------|---|----------------------------|
| Unity | Aufbau eines 3D-Modells der virtuellen Anlage durch pick&place von Objekten im virtuellen Raum | Erweiterte Grundkenntnisse |
| game4automation | Anwendung von Verhaltensmodellen und Implementierung einer Schnittstelle zu einer realen Steuerung | Erweiterte Grundkenntnisse |
| IDE für C# | u. a. Entwicklung eigener Verhaltensmodelle | Fundierte Kenntnisse |
| SPS oder Industrie-PC | Laufzeitumgebung für die entwickelte Automatisierungssoftware | Grundkenntnisse |
| IDE für SPS-Programmierung | Erstellen eines Templates für ein SPS-Projekt, das mit den Signalen der virtuellen Anlage verbunden werden kann | Grundkenntnisse |
| VR-Brille | Darstellung der Simulation in der virtuellen Realität | Erweiterte Grundkenntnisse |

2.6 Kosten und Dauer der Entwicklung

Die Dauer der Entwicklung ist stark abhängig von der Komplexität des Projekts und den Vorkenntnissen des Entwicklers. Für das hier beschriebene Beispielprojekt des virtuellen Aufzugs wurden insg. ca. 100 Arbeitsstunden investiert. Eine Einarbeitungszeit in die benötigte Soft- und Hardware ist hierin nicht enthalten. Die Summe der Investitionskosten beläuft sich auf ca. 4400 EUR. Diese teilen sich wie folgt auf: 700 Euro Lizenzgebühren für game4automation, 2.500 EUR für den Entwicklungsrechner, 1.200 EUR für VR-Brille. Wiederkehrende Kosten fallen nicht an.

2.7 Erweiterbarkeit der Anlage

Ein großer Vorteil von virtuellen Anlagen gegenüber realen Laboranlagen ist, dass diese schnell und einfach erweitert werden können. Um dies zu verdeutlichen, sind bei dem Beispiel der virtuellen Aufzugsanlage mehrere Erweiterungen vorgenommen worden. In einem ersten Schritt wurde die Anzahl der Stockwerke, zwischen denen der Aufzug verfährt, verändert. Dazu wurde zunächst ein Stockwerk als ein Modul konfiguriert, das dann in der virtuellen Umgebung per copy&paste vervielfältigt werden kann. Der zeitliche Aufwand für die Zusammenfassung aller Elemente eines Stockwerks zu einem Modul beträgt dabei ca. 30 Minuten und das Kopieren der Stockwerke kostet dann jeweils nur wenige Minuten.

In einem zweiten Schritt wurde in der virtuellen Aufzugsanlage ein weiterer Aufzugsschacht mit einem Fahrkorb durch Kopieren des ersten Schachts hinzugefügt. Der zeitliche Aufwand für diese Erweiterung beträgt ca. 30 Minuten. Der Zweck einer solchen Erweiterung kann sein, die Komplexität der Anlage zu erhöhen und damit die Herausforderungen an einen Steuerungsentwurf erhöhen, da in diesem System eine Koordination zwischen den Aufzügen erforderlich ist.

Diese beiden Erweiterungen deuten an, dass eine Anpassung einer virtuellen Anlage an neue Lehrinhalte oder geänderte Lernziele mit geringem Aufwand und Kosten möglich ist. Ähnliche Änderungen an einer realen Versuchsanlage würden offensichtlich einen deutlich längeren Planungsaufwand erfordern und wären auch in der Realisierung um ein Vielfaches zeit- und kostenintensiver.

3 Anwendung der virtuellen Anlage im Praktikum

3.1 Beispiel für eine Praktikumsaufgabe am virtuellen Aufzug

Im Folgenden wird ein Beispiel gegeben für eine mögliche Praktikumsaufgabe, bei der die hier erklärte virtuelle Aufzugsanlage eingesetzt wird. In der Aufgabe soll eine Ablaufsteuerung mit einer in der Lehrveranstaltung behandelten Entwurfsmethode für die Positionierung des Fahrkorbs eines Aufzugs entworfen und in der Programmiersprache ST auf einer speicherprogrammierbaren Steuerung implementiert werden. Die theoretischen Grundlagen zu dieser Praktikumsaufgabe werden in dem folgenden Video erläutert: <https://youtu.be/wDNiReGEZGs>.

Mit der Aufgabe werden die folgenden Lernziele verfolgt: Die Praktikums Teilnehmer können

- die Spezifikation einer Ablaufsteuerung in Form eines UML-Zustandsdiagramms beschreiben,
- die Steuerung gem. dem entwickelten UML-Zustandsdiagramm in ST implementieren,
- Aktoren aus der Steuerung über Einzelsteuerfunktionen ansteuern,
- die Steuerung durch geeignete Testfälle an dem simulierten System experimentell erproben und
- die Entwurfsmethodik hinsichtlich des Aufwands und Nutzen gegenüber rein heuristischen Verfahren einschätzen und bewerten.

Das Ziel dieser Praktikumsaufgabe ist die Entwicklung eines Funktionsbausteins zur automatischen Positionierung des Fahrkorbs. Nachfolgend ist das Verhalten des gesteuerten Fahrkorbs beschrieben, das als Spezifikation der zu entwickelnden Steuerung zu verstehen ist.

Wenn der Aufzug wie in Abb. ??(a) dargestellt in OG1 halten soll, muss dieser so positioniert werden,

dass sich der mittlere Marker genau vor dem induktiven Sensor 1 befindet. Gleiches gilt für die Positionierung des Aufzugs in allen anderen Geschossen. Um den Komfort bei der Fahrt zu erhöhen, sollen Anfahrt und Abbremsen des Aufzugs mit verringerter Geschwindigkeit erfolgen. Dafür werden der obere und der untere Marker am Aufzug verwendet, was an folgenden zwei Beispielen erläutert wird: Im ersten Beispiel soll der Aufzug ausgehend aus der in Abb. ??(a) dargestellten Situation von OG1 nach EG fahren. Dabei werden die folgenden Schritte durchlaufen:

- Beim *Anfahren* fährt der Aufzug mit geringer Geschwindigkeit herab, bis der obere Marker am induktiven Sensor 1 erfasst wird. Danach beschleunigt der Aufzug auf normale (höhere) Geschwindigkeit.
- Sobald der untere Marker am induktiven Sensor 0 erfasst wird, beginnt das *Abbremsen*. Dafür wird die Geschwindigkeit des Aufzugs zunächst wieder verringert, bis der induktive Sensor 0 den mittleren Marker detektiert. Dann soll der Aufzug stehen bleiben.

Im zweiten Beispiel soll der Aufzug ausgehend aus der in Abb. ?? dargestellten Situation von OG1 nach OG3 fahren. Dabei werden die folgenden Schritte durchlaufen:

- Beim *Anfahren* fährt der Aufzug mit geringer Geschwindigkeit herauf, bis der untere Marker am induktiven Sensor 1 erfasst wird. Danach beschleunigt der Aufzug auf normale (höhere) Geschwindigkeit.
- Sobald der obere Marker am induktiven Sensor 3 erfasst wird, beginnt das *Abbremsen*. Dafür wird die Geschwindigkeit des Aufzugs zunächst wieder verringert, bis der induktive Sensor 3 den mittleren Marker detektiert. Dann soll der Aufzug stehen bleiben.

3.2 Varianten des Praktikumsaufbaus

Je nach verfügbarer Hardware, sind bei einem Praktikum unter Verwendung einer virtuellen Anlage, unterschiedliche Varianten des Aufbaus umsetzbar. Im Folgenden werden vier aufeinander aufbauende Varianten beschrieben. In jedem dieser vorgestellten Aufbauten ist eine Durchführung und Bearbeitung einer Praktikumsaufgabe, wie sie im vorherigen Abschnitt beispielhaft beschrieben wurde, möglich.

Die nachfolgend vorgestellten Varianten des Praktikumsaufbaus werden in Bild ?? in kompakter Form gezeigt. Die in dieser Abbildung dargestellten gestrichelten Linien sind Abgrenzungen, innerhalb derer die Hard- und Software angegeben ist, die für die Umsetzung der jeweiligen Variante benötigt wird. Die Variante B ist also eine Erweiterung der Variante A, Variante C erweitert die Variante B und in der Variante D wird sämtliche in der Abbildung gezeigte Hardware verwendet.

Für die Umsetzung der **Variante A** ist lediglich ein PC oder Laptop erforderlich, auf dem die Entwicklungs- und Laufzeitumgebung TwinCAT installiert ist und der die in Abschnitt ?? beschriebenen Anforderungen hinsichtlich der Darstellung der 3D-Visualisierung auf einem Monitor erfüllt.

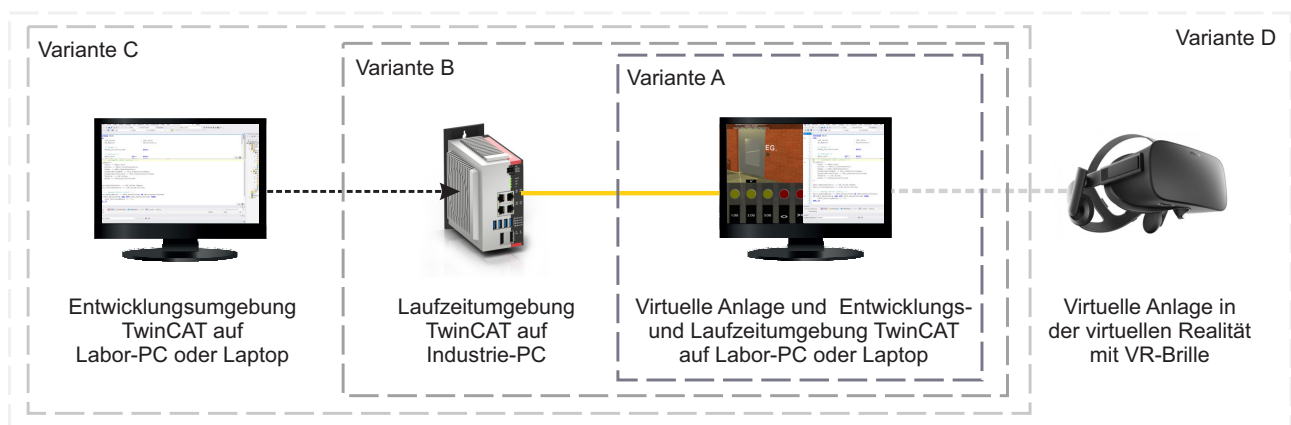


Bild 4: Varianten eines möglichen Praktikumsaufbaus

In der Entwicklungsphase kann die Steuerung also auf diesem Rechner entworfen und implementiert werden. In der Erprobungsphase werden dann die virtuelle Anlage als ausführbare Datei und die Steuerung in der Laufzeitumgebung TwinCAT gestartet. In der Simulation der virtuellen Anlage werden dann die Aktionen ausgeführt, die durch die Steuerung vorgegeben werden, wodurch ein Test der entwickelten Steuerung möglich ist. Diese sehr einfache Konfiguration des Versuchsaufbaus eignet sich z. B. für vorbereitende Aufgaben, die die Studierenden vor dem eigentlichen Praktikum lösen sollen oder für die Durchführung des Praktikums im vollständigen Distanzunterricht.

Wenn im Praktikum auch industriell eingesetzte Hardware der Steuerungstechnik verwendet werden soll, kann der Praktikumsaufbau – wie in **Variante B** gezeigt – um eine SPS oder einen Industrie-PC erweitert werden. Hierbei wird der Industrie-PC mit dem Entwicklungsrechner über ein Ethernet-Kabel (in Bild ?? gelb dargestellt) verbunden. Der wesentliche Unterschied zur Variante A betrifft die Erprobungsphase. Die auf dem PC oder Laptop entwickelte Steuerung muss nämlich zuvor auf den Industrie-PC geladen und dort gestartet werden. In der Simulation der virtuellen Anlage sind darüber hinaus keine Veränderungen der Einstellungen im Vergleich zu Variante A vorzunehmen.

Um den Workflow der Steuerungsentwicklung noch realitätsnäher zu gestalten, kann der Entwicklungsrechner (häufig auch als Engineering-Station bezeichnet) ein anderes Gerät sein, als der PC, auf dem die Simulation läuft, wie dies in **Variante C** dargestellt ist. Nach der Entwicklung und Implementierung der Steuerung muss der Code wieder auf den Industrie-PC geladen und dieser dann gestartet werden. In der Erprobungsphase kommt der Entwicklungsrechner dann nicht mehr zum Einsatz. Auf dem PC, auf dem die Simulation der virtuellen Anlage ausgeführt wird, muss in dieser Konfiguration TwinCAT nicht installiert sein. Der Vorteil der Trennung der Geräte für die Steuerungsentwicklung und die Ausführung der Simulation ist, dass der Entwicklungsrechner nicht die hohen Anforderungen an die Grafikleistung erfüllen muss. Daher können als Entwicklungsrechner ggf. auch private Geräte von Studierenden verwendet werden, was eine Vorbereitung der Implementierung der Steuerung (z. B. als Teil der vorbereitenden Aufgaben) möglich macht.

Die 3D-Visualisierung auf einem Monitor ist eine gute Möglichkeit, das Verhalten des gesteuerten Systems zu veranschaulichen. Darüber hinaus kann – wie in **Variante D** – die virtuelle Anlage aber auch unter Verwendung einer VR-Brille in der virtuellen Realität als echte 3D-Simulation erfahrbar gemacht werden. Dabei können sich die Studierenden im virtuellen Raum bewegen, das Verhalten der gesteuerten Anlage beobachten und empfinden den virtuellen Raum als „fast“ reale Praktikums Umgebung. Im Beispiel der Aufzugsanlage können die Studierenden u. a. die Bewegung des Aufzugs von Außen beobachten oder sogar im Aufzug mitfahren und hier die Taster bedienen, deren Funktionalität sie selber programmiert haben.

Die hier dargestellten Varianten des Praktikumsaufbaus stellen keine abschließende Aufzählung dar. So ist z. B. auch möglich die Variante A nur um eine VR-Brille zu erweitern, ohne dass ein Industrie-PC eingesetzt wird. Weiterhin kann natürlich auch eine Trennung der Geräte für die Entwicklung und Ausführung der Steuerung einerseits und die Ausführung der Simulation andererseits erfolgen, ohne Verwendung eines Industrie-PCs. Je nach Zielstellung des Praktikums kann also die Konfiguration des Aufbaus an die Aufgabenstellung oder sogar an die Phase der Durchführung (z. B. Entwicklungs- und Erprobungsphase) angepasst werden.

4 Zusammenfassung und Ausblick

4.1 Bewertung der Vor- und Nachteile virtueller Anlagen

In dem Beitrag wurden einige Vor- und Nachteile virtueller Anlagen gegenüber realen Laboranlagen genannt und erläutert. Hier sollen diese Aspekte, die für oder gegen deren Verwendung sprechen nochmal zusammengefasst werden.

Virtuelle Versuchsanlagen haben aus folgenden Gründen einen besonderen Nutzen in der praktischen Ingenieursausbildung:

- Kurze Entwicklungszeiten mit geringen Investitionskosten
- Vernachlässigbarer Platzbedarf (nennenswert ist hier lediglich ein abgesperrter, freier Bereich, in

dem man sich mit einer VR-Brille bewegen kann)

- Einfach planbare, schnell umsetzbare und kostengünstige Erweiterungen, die eine Anpassung der virtuellen Anlagen an neue Aufgabenstellungen, überarbeitete Lehrinhalte und geänderte Lernziele möglich macht
- Mehrere Praktikumsgruppen können an Kopien derselben virtuellen Anlage zur gleichen Zeit arbeiten
- Vervielfältigungen und Verteilung Simulationsmodelle virtueller Anlagen sind möglich, da diese Dateien keinen Lizenzbeschränkungen unterliegen
- Einsatz in unterschiedlichen Varianten von Praktikumsaufbauten je nach verfügbarer Hardware möglich

Die Entwicklung und Anwendung virtueller Anlagen als Ergänzung bestehender Laboraufbauten birgt folgende Herausforderungen und Probleme:

- Laboringenieure benötigen neue Kompetenzen im Bereich der IT, die für den Betrieb „klassischer“ Labore nicht notwendig sind
- Simulationsmodelle sind Idealisierungen der Realität, in denen unerwartete Störungen oder technische Fehler nicht vorkommen
- Visualisierungen des Verhaltens der automatisierten virtuellen Anlage an einem Monitor sind bei Weitem nicht so wertvoll, wie die Erfahrungen an realen Anlagen
- 3D-Simulationen in der virtuellen Realität bedingen Kompetenzen in der Inbetriebnahme und Anwendung von VR-Brillen

4.2 Vision zu der Zukunft der Laborpraktika

Aus der vorangegangenen zusammenfassenden Bewertung zur Verwendung virtueller Anlagen im Rahmen einer Ingenieursausbildung ist zu erkennen, dass es keine Argumentation exklusiv für oder wider Praktika mit virtuellen Laboranlagen gibt. Vielmehr können diese Art der Anlagen eine sinnvolle Ergänzung der bestehenden Laborinfrastruktur sein. Diese Einschätzung wird sich auch in Zukunft mit einer zunehmenden Digitalisierung der Lehre nicht verändern, da die praktischen Erfahrungen mit realen Anlagen unersetzbar bleiben.

Dennoch ist es vorstellbar, dass virtuelle Anlagen zukünftig häufiger in Laborpraktika oder auch in Übungen im Rahmen von Lehrveranstaltungen mit seminaristischem Unterricht (z. B. in der im Abschnitt ?? beschriebenen Variante A) eingesetzt werden. Besonders interessant ist dabei, dass die als ausführbaren Dateien exportierten Simulationsmodelle der virtuellen Anlagen lizenzfrei nutzbar sind und damit auch zwischen Lehrenden ausgetauscht werden können. Sollten also unterschiedliche Entwickler virtueller Anlagen (z. B. an verschiedenen Hochschulen) die Bereitschaft haben, diese Entwicklungen freizugeben, könnte zukünftig eine Plattform zum Austausch virtueller Anlagen entstehen.

5 Literatur

- [1] E. Kucera, O. Haffner und R. Leskovský, „Interactive and virtual/mixed reality applications for mechatronics education developed in unity engine“, 2018 Cybernetics Informatics (K&I), 2018, S. 1–5.
- [2] C. G. Lee und S. C. Park, „Survey on the virtual commissioning of manufacturing systems“, Journal of Computational Design and Engineering, 2014, S. 213–222.
- [3] M. E. Macías García, A. A. Cortés Pérez und A. R. Izaguirre Alegría, „Cyber-Physical Labs to enhance engineering training and education“, Int. Journal on Interactive Design and Manufacturing, 2020, S. 1253–1269.
- [4] G. Reinhart und G. Wunsch, „Economic application of virtual commissioning to mechatronic production systems“, Production Engineering, 2007, S. 371–379.
- [5] R. Stark, T. Damerau und K. Lindow, „Industrie 4.0 – Digitale Neugestaltung der Produktentstehung und Produktion am Standort Berlin“, Industrie 4.0 grenzenlos, 2016, S. 169–184.
- [6] J. Wang et al., „Simulation and visualization of industrial processes in unity“, Conference on Summer Computer Simulation (SummerSim '15), 2015, S. 1–7.