Lehrstuhl für Computergestützte Modellierung und Simulation
Ingenieurfakultät Bau Geo Umwelt
Technische Universität München

Fabrication Information Modelling – BIM-basierte Modellierung von Fertigungsinformationen für Additive Manufacturing

Wissenschaftliche Arbeit zur Erlangung des Grades

Master of Science (M.Sc.)

an der Ingenieurfakultät Bau Geo Umwelt der
Technischen Universität München.

Betreut von Prof. Dr.-Ing. André Borrmann
M. Sc. Simon Vilgertshofer
Lehrstuhl für Computergestützte Modellierung und Simulation

Eingereicht von Martin Slepicka [redacted]
[redacted]
[redacted]
e-Mail: martin.slepicka@tum.de

Eingereicht am TT. Monat JJJJ

Abstract

Additive manufacturing (AM) has been playing an increasingly important role in industry for several years. AM convinces as a fast and resource-saving manufacturing method, especially for the creation of models, samples and prototypes (rapid prototyping). After the use of this method was initially limited to the mechanical engineering sector, the possible areas of application are now expanding to many other sectors of the economy, as a steadily increasing number of research projects are opening up more and more materials for 3D printing and removing more and more restrictions. Interest in 3D printing is also growing in the construction industry; while this method was initially only used to create architectural models, the development of efficient concrete and steel printing processes means that it can now be used in many other tasks in construction projects. As a digital fabrication method, AM has the potential to automate many processes traditionally performed by hand in the construction industry, and could thus give a boost to the sector's productivity, which has been stagnant for years. Building Information Modeling (BIM) is a particularly well-suited interface for the use of AM in the construction industry. Since AM is fundamentally based on high-quality digital models, it is only natural to use high-quality BIM models for this purpose, because this way a closed digital chain from design to production can be realized with corresponding advantages. The main objective of this work is to investigate whether such a mutual integration of these two digital methods is possible and to highlight how BIM needs to be extended to represent manufacturing models for AM. For this purpose, material, process and machine parameters and their mutual interaction are investigated, an automated derivation of manufacturing data from a BIM is prototypically implemented and possibilities are investigated to make the data generated in this way usable for further use in the BIM context.

Zusammenfassung

Additive Fertigung (*englisch*: Additive Manufacturing, AM) nimmt in der Industrie seit einigen Jahren einen immer größeren Stellenwert ein. AM überzeugt als schnelles und ressourcensparendes Fertigungsverfahren, vor allem bei der Erstellung von Modellen, Mustern und Prototypen (Rapid Prototyping). Nachdem der Einsatz dieser Methode zunächst auf den Maschinenbau beschränkt war, erweitern sich die möglichen Anwendungsgebiete mittlerweile auf viele weitere Wirtschaftszweige, da durch eine stetig steigende Anzahl von Forschungsprojekten immer mehr Materialien für den 3D-Druck erschlossen und immer mehr Einschränkungen aufgehoben werden. Auch in der Bauindustrie wächst das Interesse am 3D-Druck: Wurde das Verfahren zunächst nur für die Erstellung von Architekturmodellen genutzt, kann es durch die Entwicklung effizienter Druckverfahren für Beton und Stahl inzwischen auch für viele weitere Aufgaben bei Bauprojekten angewendet werden. Als digitales Fertigungsverfahren hat AM das Potential, viele Prozesse zu automatisieren, die in der Bauindustrie traditionell von Hand ausgeführt werden, und könnte so die seit Jahren stagnierende Produktivität der Branche ankurbeln. Eine besonders gut geeignete Schnittstelle für den Einsatz von AM in der Bauindustrie stellt hierbei das Building Information Modeling (BIM) dar. Da AM fundamental auf hochwertigen digitalen Modellen basiert, liegt es nahe, hierfür hochwertige BIM-Modelle zu verwenden, da auf diese Weise eine geschlossene digitale Kette vom Entwurf bis zur Produktion mit entsprechenden Vorteilen realisiert werden kann. Das Hauptziel dieser Arbeit ist es, zu untersuchen, ob eine solche gegenseitige Integration dieser beiden digitalen Methoden möglich ist und aufzuzeigen, wie BIM erweitert werden muss, um Fertigungsmodelle für AM darzustellen. Dazu werden Material-, Prozess- und Maschinenparameter und deren gegenseitige Wechselwirkung untersucht, eine automatisierte Ableitung von Fertigungsdaten aus einem BIM prototypisch implementiert und Möglichkeiten untersucht, die so generierten Daten für die weitere Verwendung im BIM-Kontext nutzbar zu machen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

2D	Zweidimensional
3D	Dreidimensional
AM	Additive Fertigung (Additive Manufacturing)
API	Application Programming Interface
BIM	Building Information Modeling
CAD	Computer Aided Design
CAM	Rechnergestütztes Fertigen (Computer Aided Manufacturing)
CIM	Rechnergestützte Fertigung (Computer Integrated Management)
CNC	Rechnergestützte Numerische Steuerung (Computerized Numerical Control)
DDSS	Design Decision Support System
DEM	Diskrete Elemente Methode
FDM	Fused deposition modeling
FIM	Fabrication Information Modelling
FLM	Extrusionsverfahren (Fused Layer Modeling)
GIS	Geoinformationssystem
IAI	International Alliance for Interoperability
IFC	Industry Foundation Classes
LoC	Level of Coordination
LOD	Level of Development
LoG	Level of Geometry
LoI	Level of Information
LPBF	selektives Laserschmelzen (Laser Powder-Bed Fusion)
PLM	Product Lifecycle Management
RC	Robotersteuerung (Robot Control)
ROS	Robot Operating System
SC3DP	Shotcrete 3D Printing
SCA	selektive Zement Aktivierung (selective cement activation)
SPF	STEP Physical Format
SPI	selektive Zementleim Intrusion (selective paste intrusion)
TCP	Werkzeugmittelpunkt (Tool Center Point)
WAAM	Lichtbogen-Additive Manufacturing (Wire and Arc Additive Manufacturing)

Kapitel 1

Einleitung

Die Methodik des Building Information Modeling (BIM) wird im Bauwesen immer mehr zum Einsatz gebracht und stetig weiterentwickelt. Dabei ist BIM mehr als nur eine einfache Weiterentwicklung des immer noch weitestgehend angewendeten 2D-CAD-Zeichnens, da in einem BIM-Modell neben den Geometrieinformationen auch semantische Daten des Bauwerks erfasst werden. Mit diesem holistischen Ansatz, alle nötigen Informationen in einem Datenmodell zu erfassen, kann BIM über den gesamten Lebenszyklus eines Gebäudes in allen verschiedenen Planungsphasen sowie bei der Instandhaltung unterstützend eingesetzt werden. Eine aktuelle Neuentwicklung ist das Konzept, Fertigungsdaten für automatisierte Herstellungsprozesse, wie zum Beispiel der Additive Fertigung (*englisch*: Additive Manufacturing, AM), aus einem BIM-Modell abzuleiten.

Additive Fertigung ist schon seit geraumer Zeit in der Industrie etabliert und kommt hauptsächlich bei der Fertigung von Modellen, Mustern, Prototypen und Werkzeugen zum Einsatz. Ein wesentlicher Vorteil von AM ist, dass der Einsatz der Fertigungsmethode besonders ressourcensparend ist und zudem eine größere Formenvielfalt bei den Bauteilen erzielt werden kann. Im Bauwesen wird diese Technik jedoch trotz der aktuell sehr intensiven Forschung noch kaum eingesetzt, da die momentan möglichen Fertigungstechniken für die Baustelle oder in der Bauteilproduktion noch nicht ausgereift genug sind. Zudem ist auch die Erstellung der hierfür notwendigen Fertigungsinformationen bisher noch mit viel Aufwand verbunden. Es muss hierzu aus dem geometrischen Modell – im Bauwesen meist B-Rep-Modelle – eine vollständige räumliche Darstellung, also inklusive der inneren Struktur des Volumenkörpers, erstellt und daraus ein Druckpfad für den 3D-Druckvorgang abgeleitet werden. Dieser Prozess wird derzeit noch nicht ausreichend durch entsprechende Software unterstützt und ist zudem aufgrund notwendiger Dateitypkonvertierungen in den einzelnen Zwischenschritten sehr fehleranfällig.

Das Ziel dieser Masterarbeit ist es, Konzepte zu entwickeln, wie die Modellierung der Fertigungsinformationen unter Verwendung des zugrunde liegenden BIM-Modells weitestgehend automatisiert werden kann. Zunächst ist es hierfür erforderlich, alle nötigen Parameter zu identifizieren, die vorhanden sein müssen, damit diese Automatisierung möglich ist. Zu diesen teilweise nach AM-Fertigungsmethode unterschiedlichen Parametern zählen im Falle von Extrusionsdruckverfahren z.B. Materialeigenschaften, wie etwa die Aushärtezeit oder die Fließeigenschaften des Betons, sowie Prozesseigenschaften, wie Pfadgestaltung und Düsengeschwindigkeit. Sind die Parameter, die mit der Geometrie verknüpft sind, also vom Designer vorgegeben wurden, nicht im Normbereich der Fertigungsmethode, lässt sich schon direkt ausschließen, dass AM für das Bauteil eingesetzt werden kann. Die eben genannten Parameter haben also einen direkten Einfluss darauf, welche Geometrien überhaupt mit den entsprechenden AM-Methoden realisiert werden

können, bzw. umgekehrt setzen bestimmte Geometrien voraus, dass entsprechende Parameter gesetzt werden. Weiterhin ist zu untersuchen, ob anhand der Geometrie diese Informationen auch automatisch angereichert werden können, sodass z.B. dem Designer bereits beim Erstellen der Geometrie unter anderem Vorschläge zur Materialwahl gegeben werden kann. Nach Möglichkeit sollen hier auch verschiedene Bauvarianten vorgeschlagen werden, die in Folgeschritten auf Effizienz verglichen werden können.

Mit all diese Informationen, ob ein Bauteil überhaupt realisiert werden kann und wenn ja, mit welchen Materialien und in welcher Variation am sinnvollsten, soll der Designer später beim Planungsprozess versorgt werden, um somit zu jeder Zeit in seinen Designentscheidungen unterstützt zu werden. Um ein solches Design Decision Support System (DDSS) zu realisieren, müssen daher all diese Informationen in dem Modell verankert werden. Hierfür ist ein multi-Level of Development (LOD) Konzept mit einer abgewandelten LOD400-Stufe, in der alle erweiterten Geometrie- und Prozessinformationen enthalten sind und aus der direkt die Robotersteuerung (Druckpfad) abgeleitet werden kann, denkbar (vgl. Abb. 1.1). Der Graphabschnitt in der Mitte des Bildes symbolisiert hierbei die Vernetzung der einzelnen Detailstufen, die links auch noch in geometrischer Repräsentation dargestellt sind. Anhand dieser Vernetzung soll dargestellt werden, dass regelbasiert aus groben Detailstufen feinere erstellt werden können. In Abb. 1.1 wurde in einem ersten Implementierungsvorschlag die LOD400-Detailstufe vollständig durch ein als „Fabrication Information Model (FIM)“ bezeichnetes Konstrukt ersetzt, für das ein erweiterter Regelsatz erstellt werden soll. Wie ein solches Regelset für die Modellierung von Fertigungsinformationen letztlich aussehen soll, was genau der Begriff FIM bedeutet und wie er sich in den BIM-Kontext eingliedert, wird im Folgenden ausgearbeitet. Idealerweise soll in einem FIM zusätzlich die Möglichkeit bestehen, dass vorab über Simulationen Eigenschaften, wie z.B. Tragfähigkeit oder Wärmeisolation, des zu fertigenden Bauteils abgeschätzt werden können. Als Grundlage hierfür könnte z.B. die Datenstruktur der Industry Foundation Classes (IFC) dienen. In dieser Arbeit soll daher konzeptionell untersucht werden, ob sich

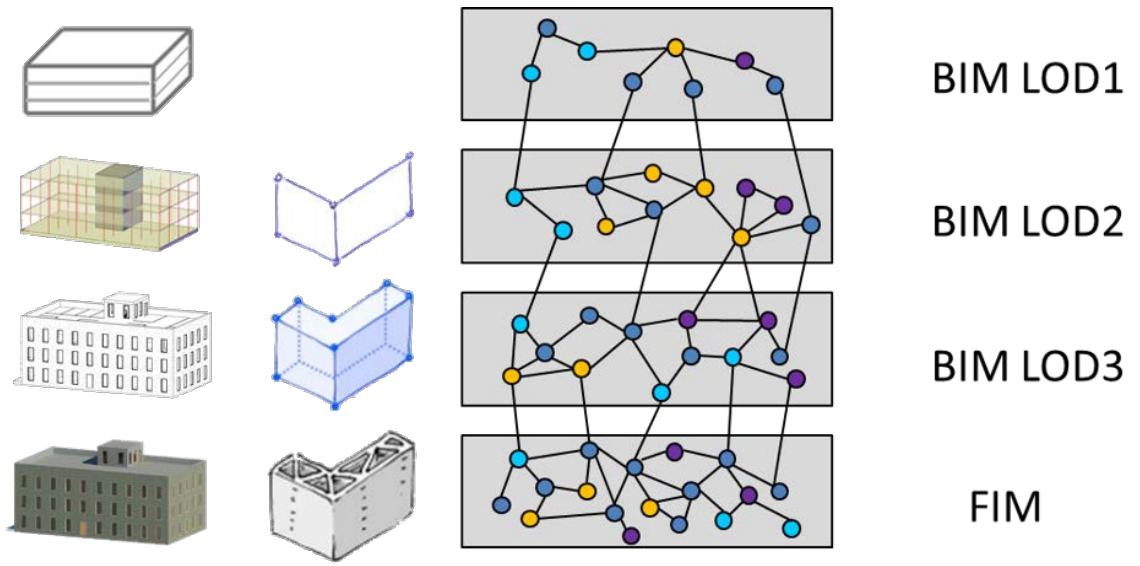


Abbildung 1.1: FIM Detailstufe eines erweiterten BIM-Modells. (KLOFT et al., 2020)

IFC hierzu als Datenformat eignet und wenn ja, welche Erweiterungen hierfür notwendig sind.

Als praktischen Teil dieser Arbeit soll an einem kleinen repräsentativen Beispiel ein Druckvorgang mit vorhergehender BIM-Planung in zwei verschiedenen Varianten durchgeführt werden. Über alle Teilbereiche dieser Arbeit wird hierzu nur ein Druckverfahren untersucht, das Extrusionsdruckverfahren (WEGER et al., 2018), da die verschiedenen AM-Methoden mitunter sehr unterschiedliche Voraussetzungen an das Modell sowie die verwendeten Materialien haben und eine gesamtheitliche Betrachtung der Druckverfahren den Rahmen dieser Arbeit sprengen würde.

Zusammengefasst dargestellt wurden in dieser Masterarbeit folgende Fragestellungen bearbeitet:

- Welche zusätzlichen geometrischen und semantischen Informationen müssen in einem BIM Modell mindestens vorhanden sein, damit daraus (teil)automatisch ein FIM für AM abgeleitet werden kann?
- Sind diese Informationen auch automatisch bestimmbar, bzw. können direkt aus der Geometrie Mindestanforderungen abgeleitet werden?
- Welcher Detaillierungsgrad des Modells ist für ein AM geeignetes FIM notwendig und lässt sich das ansatzweise mit dem bestehenden Datenformat IFC umsetzen?

Kapitel 2

Theoretische Grundlagen

Wie bereits in den einleitenden Worten angedeutet, werden im Rahmen dieser Masterarbeit Konzepte entwickelt, wie aus bestehenden BIM-Modellen (teil-)automatisiert Fertigungsinformationen für AM abgeleitet werden können und auf welche Art und Weise diese Informationen im BIM-Kontext repräsentiert werden können. Hierfür werden im Folgenden einige grundlegende Begriffe und Methoden näher erläutert.

2.1 Building Information Modeling

Unter einem Building Information Model (BIM) versteht man im einfachsten Sinne ein Bauwerksmodell, welches nicht nur geometrische Informationen enthält, sondern auch mit semantischen Informationen angereichert ist. Damit lässt sich ein solches Modell über den gesamten Lebenszyklus des Bauwerks verwenden. Beim Erstellen eines BIM wird demnach nicht nur das geometrische Modell des Bauwerks, sondern auch ein zugehöriges Informationsmodell erzeugt. Beide Teile werden in mehreren Schritten immer weiter verfeinert, sodass sämtliche Aufgaben, die während der Lebenszeit des Bauwerks anfallen, mit Hilfe des BIM-Modells bearbeitet werden können. Dieser Prozess bis hin zum fertigen Modell wird entsprechend Building Information Modeling (BIM) genannt. Abbildung 2.1 zeigt in vereinfachter Weise die einzelnen Anwendungsgebiete über den Lebenszyklus eines Bauwerks. (BORRMANN et al., 2015b)

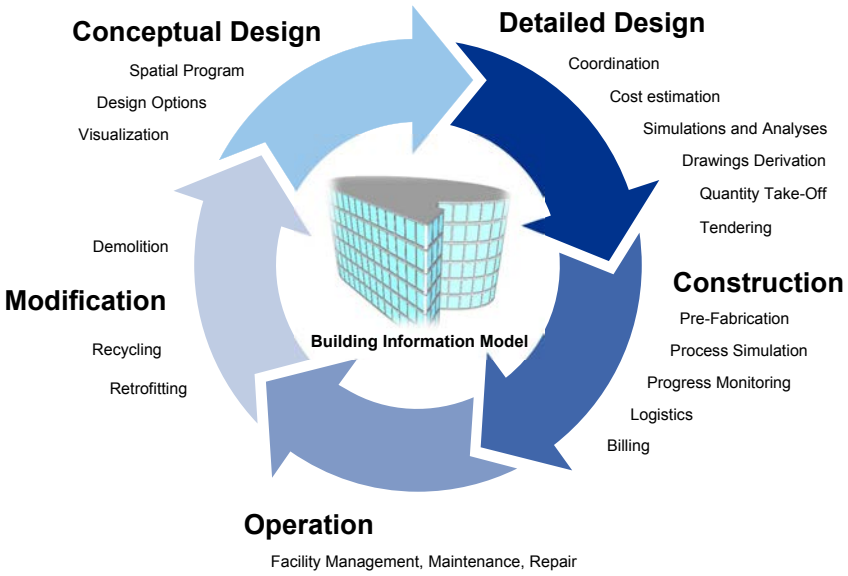


Abbildung 2.1: Verwendung eines BIM über den gesamten Lebenszyklus eines Bauwerks. (BORRMANN et al., 2015b)

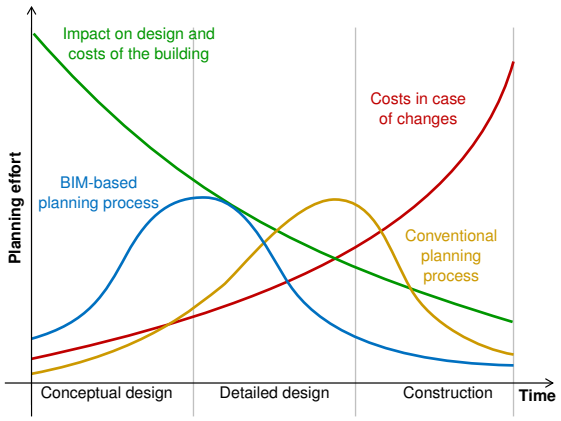


Abbildung 2.2: Verschobener Planungsaufwand durch BIM. (MACLEAMY, 2004)

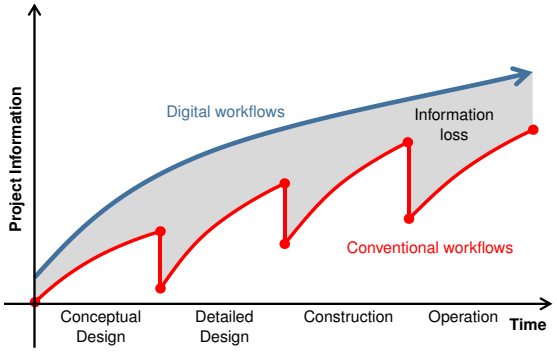


Abbildung 2.3: Projekt Informationsinhalt im Vergleich. (nach EASTMAN et al., 2011)

Ähnlich wie bei der konventionellen Planung passiert die schrittweise Verfeinerung des Modells in vielen Bereichen in einem iterativen Prozess und involviert über die gesamte Planungsphase in der Regel eine große Anzahl an Spezialisten mit entsprechend umfangreichen Datenaustausch (CHIU, 2002). Beim BIM-Modellieren ist jedoch der größte Planungsaufwand im Vergleich zum konventionellen Planen zeitlich vorgeschoben, da schon viel früher Informationen eingearbeitet werden, die für andere Disziplinen nötig sind (BORRMANN et al., 2015b). Abbildung 2.2 veranschaulicht dies und deutet auf die daraus resultierenden Vorteile hin. Wie dargestellt, wird der Hauptteil des Planungsaufwandes in einem sehr frühen Zeitraum ausgeführt, so können vorzeitig Auswirkungen von Designentscheidungen abgeschätzt und rechtzeitig Änderungen vorgenommen werden, bevor unnötige Kosten entstehen (BORRMANN et al., 2015b).

Ein weiterer Vorteil der BIM-Modellierung wird in Abb. 2.3 dargestellt. Da bei der BIM-Modellierung durchgehend mit digitalen Informationen in hoher Datenqualität gearbeitet wird, ist der Planungsprozess theoretisch ohne Informationsverlust und damit sehr effizient bewältigbar. Bei konventioneller Planung entstehen oft Probleme (Informationsverluste) beim Datenaustausch mit anderen Disziplinen, da noch hauptsächlich mit 2D-Plänen – teils noch in ausgedruckter Form – und proprietären Datenformaten gearbeitet wird.

BIM muss bei einem Planungsprozess aber auch nicht mit all seinen Möglichkeiten eingesetzt werden. Die Methodik kann auch nur für einzelne ausgewählte Disziplinen angewendet werden und muss daher nicht zwangsläufig über den gesamten Lebenszyklus angewendet werden. In diesem Fall spricht man von **„little bim“** im Gegensatz zum vollumfassenden Einsatz über alle Teildisziplinen (**„BIG BIM“**). Demnach kann BIM in verschiedene „Anwendungsdimensionen“ unterteilt werden, ausgehend von der Geometrie (3D) erhöht jede Ergänzung hierzu diese Dimension um eins (vgl. Abb. 2.4). Es können zu der Geometrie Bauablaufpläne erstellt (4D), Kosten abgeschätzt (5D), Verbrauchsanalysen durchgeführt (6D) und Bestandserhaltungsmaßnahmen unterstützt werden (7D) (FU et al., 2006). Mittlerweile wird sogar mit 8D Modellen gearbeitet, die auch sicherheitsrelevante Informationen enthalten (KAMARDEEN, 2010).

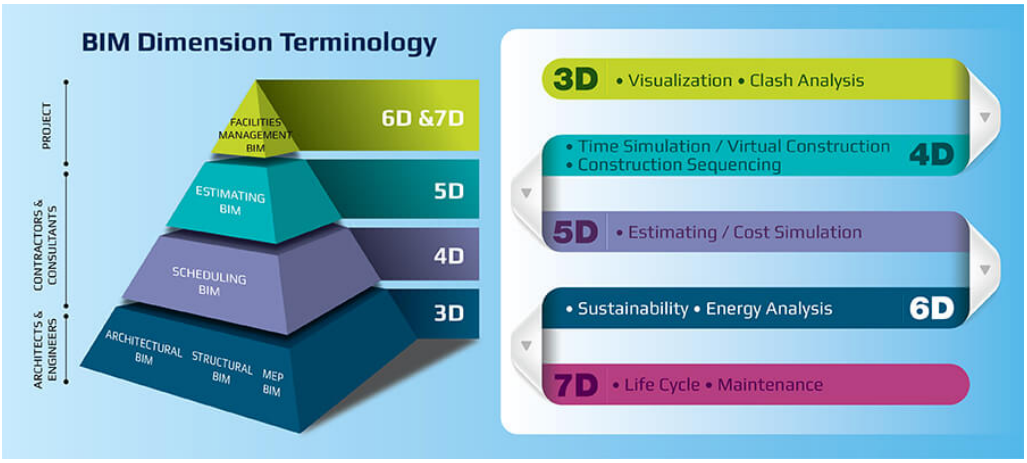


Abbildung 2.4: BIM in den verschiedenen Erweiterungsstufen. (SINGH & PATEL, 2020)

2.1.1 Level of Development

Bei der Erstellung eines BIM-Modells wird häufig sehr früh die 3D-Geometrie in hoher Genauigkeit erzeugt, zunächst auch ohne der zugrunde liegenden Entscheidungsstütze prüfender Ingenieure. Mit einer ausgereiften Geometrie entsteht dabei der Eindruck, dass auch weitere Annahmen getroffen werden können. Reicht der Informationsgehalt oder die Qualität der Informationen jedoch nicht aus, können diese Annahmen allerdings völlig falsch getroffen werden. (ABUALDENIEN et al., 2020)

Um den Ausarbeitungsgrad des Modells, also den Modellstatus, zu bewerten, wurden sogenannte Levels of Development (LODs) eingeführt. Diese werden vom LOD 100 (Vorentwurf) bis zum LOD 500 (As Built) üblicherweise in 100er-Schritten aufgelistet, in

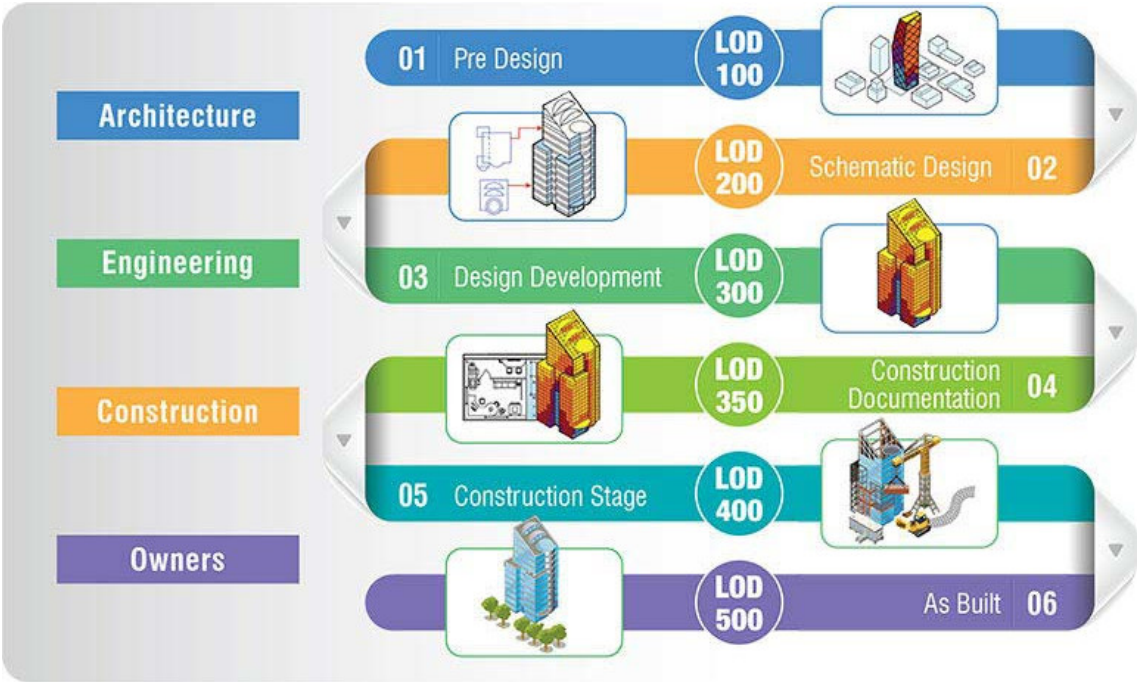


Abbildung 2.5: Vereinfachte Darstellung der verschiedenen LOD-Stufen. Nach »Level of Development« (2020).

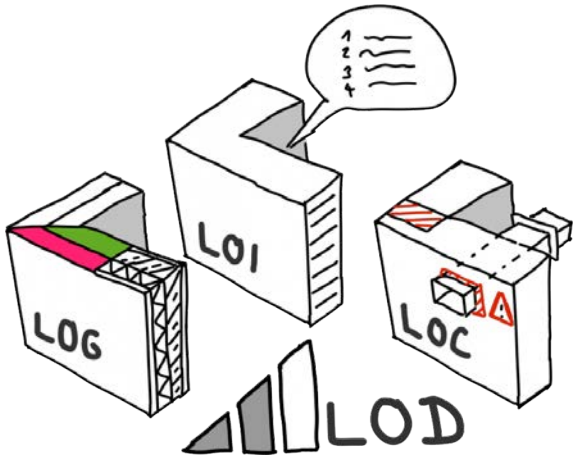


Abbildung 2.6: Einzelne Bestandteile des LOD. (»BIMpedia«, 2020)

gewissen Fällen werden aber auch Zwischenlevels, wie z.B. LOD 350, definiert (siehe Abb. 2.5). Mit dem LOD-Parameter lässt sich z.B. abschätzen, ob das BIM-Modell ausreichend detailliert ist um gewisse Informationen aus dem Modell abzuleiten. (BIM FORUM, 2019; HOOPER, 2015)

Genauer gesagt ist der LOD-Parameter eine Zusammensetzung aus drei anderen Level-Parametern, nämlich dem Level of Geometry (LoG), dem Level of Information (LoI) und dem Level of Coordination (LoC). Dabei ist entsprechend der LoG der Detaillierungsgrad der Geometrie, also die Realitätsgenauigkeit der Gestaltung, der LoI das Maß an Informationsgehalt und der LoC ein Maß für den Koordinationsgrad, also ob das Modell qualitativ hochwertig und entsprechend geprüft ist. Der LOD ist also der Gesamt-Reifegrad des BIM-Modells (vgl. Abb. 2.6). (»BIMpedia«, 2020)

2.1.2 Multi-LOD BIM

Wie zuvor beschrieben sind an der Verfeinerung eines BIM-Modells in der Regel diverse Spezialisten vieler verschiedener Disziplinen beteiligt, die bei diesem Prozess über das semantisch reichhaltige 3D-Modell kommunizieren (BORRMANN et al., 2015b; CHIU, 2002). Hierbei werden die einzelnen Elemente des groben Vorentwurfs stückweise unter Berücksichtigung der Auflagen, die für das entsprechende Projekt gelten, genauer definiert. In jeder Detaillierungsebene werden oft verschiedene Varianten unter anderem auf die bestmögliche Leistung oder Ästhetik untersucht und nach Evaluationsschritten wird sich schließlich für eine entschieden (ZAHEDI & PETZOLD, 2018). Beispielhaft ist dieser Prozess in Abb. 2.7 dargestellt.

Wie in Abb. 2.7 verdeutlicht wird, werden nicht immer alle Varianten in den einzelnen LODs ausgearbeitet; Begrifflich wird das auch adaptive Detaillierung genannt oder Detaillierung auf Anfrage (ZAHEDI & PETZOLD, 2018). Die Entscheidung mit welcher Variante weitergearbeitet wird, ist nicht zwingend endgültig und kann jederzeit – z.B., wenn nach weiteren Simulationen in einem höheren LOD eine Unzulänglichkeit nachgewiesen wird – abgeändert werden. Zudem werden entsprechend der Resultate eventueller Simulationen

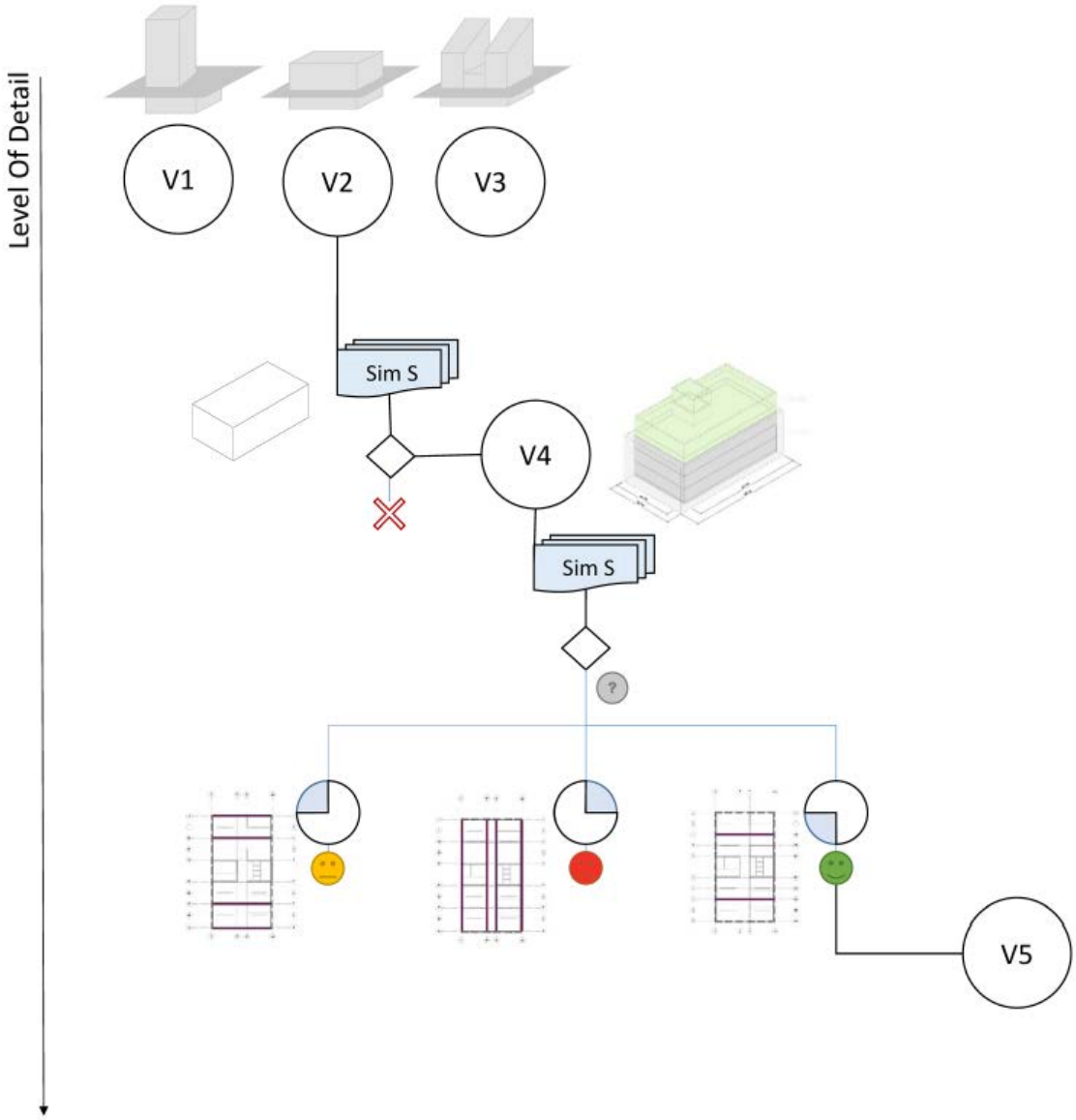


Abbildung 2.7: Exemplarischer Detailierungsprozess. (ZAHEDI & PETZOLD, 2018)

oft Optionen für die weitere Verfeinerung geliefert. In dieser Hinsicht ist es also sinnvoll, in einem BIM-Modell verschiedene Designvarianten und Optionen über die verschiedenen LODs zu erhalten. In einem aktuellen Forschungsprojekt wurde daher ein **multi-LOD meta-Modell** entwickelt, mit dem unter anderem verschiedene LOD inklusive einzelner Designvarianten konsistent abgebildet und verwaltet werden können (ABUALDENIEN et al., 2020).

In Infrastrukturprojekten, welche sich über viele Kilometer hinweg erstrecken können, sind multi-LOD Modelle von ganz besonderer Bedeutung und werden hier auch etwas anders gehandhabt. Bei der Planung von derartigen Infrastruktureinrichtungen, wie etwa dem Tunnel der 2. S-Bahn-Stammstrecke in München, wird in verschiedenen Skalen gearbeitet, die vom Kilometerbereich (Trassenplanung) bis hin zum Zentimeterbereich (Detailplanung) reichen können. Dabei muss nicht nur das Gebäude für sich geplant werden, sondern auch dessen Einbindung in die Umgebung, also den Anschluss an bestehende Infrastruktur und die genaue geografische Position. Hierzu müssen also zum einen zusätzlich zu den BIM-

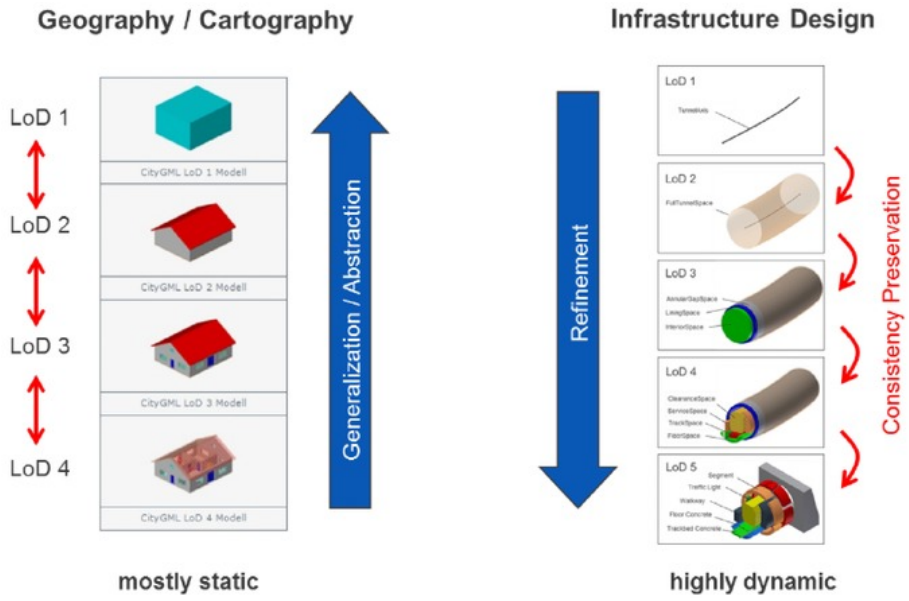


Abbildung 2.8: Unterschiedliches Verwendung von LOD in GIS- und BIM-Anwendungen. (BORRMANN et al., 2015a)

Werkzeugen auch Geoinformationssysteme (GIS) eingesetzt werden und zum anderen verschiedenskalige Repräsentationen konsistent modelliert werden (BORRMANN et al., 2015a; RAMOS JUBIERRE, 2017).

Sowohl in GIS als auch in BIM ist der Begriff LOD definiert, jedoch ist in beiden Systemen die Herangehensweise etwas anders (vgl. Abb. 2.8). Bei einem GIS wird in der Regel zunächst ein hochaufgelöstes Modell erstellt (bottom-up) und mittels Generalisierung gröbere Detaillevel erstellt, wogegen im Infrastrukturplanungsverlauf (z.B. mit BIM) mit groben Modellen begonnen wird (top-down), welche dann sukzessive detailliert werden (BORRMANN et al., 2015a). Während beim GIS die einzelnen LOD-Stufen nicht zwingend miteinander gekoppelt sein müssen, ist es für die Infrastrukturplanung unbedingt notwendig, dass über verschiedene Detailstufen Konsistenz gewahrt wird. Es kann beim Planungsverlauf vorkommen, dass in höheren Detailstufen ein Problem erkannt wird, wodurch in einer niederen Detailstufe ausgebessert werden muss. Sind die einzelnen Detailstufen hierbei unabhängig voneinander, so muss bei einer nachträglichen Änderung einer niedrigeren Stufe wiederholt jede höhere Stufe neu abgeleitet werden. BORRMANN et al. (2015a) haben hier ein Konzept entwickelt, bei dem nicht explizite geometrische Modelle für jede Detailstufe abgespeichert werden, sondern der Konstruktionsverlauf, also die Reihenfolge der angewendeten geometrischen Operationen in sogenannten „prozeduralen Modellen“. Auf diese Weise kann bei einer Änderung in einer niedrigeren Detailstufe automatisiert jede bearbeitete höhere Detailstufe darauf angepasst werden.

In welchem Bezug steht aber nun das multi-LOD Konzept mit AM, bzw. mit dem Modellieren der entsprechenden Fertigungsinformationen? Die Antwort liegt in folgender Frage: Ab welcher Detailstufe können Fertigungsinformationen für AM modelliert werden? Eine genaue Antwort wird es hier nicht geben können, da dies immer abhängig von der Komplexität des entsprechenden Bauprojekts sein wird. Jedoch können zwei extreme Her-

angehensweisen untersucht und beschrieben werden, zwischen denen sich die typische Herangehensweise befinden wird.

Zum einen wäre es möglich, dass der BIM-Designer ohne den Einsatz von AM-Methoden zu bedenken die Planung bis zu einem hohen Detailgrad (LOD400) durchführt und erst dann die Fertigungsinformationen generiert werden. Auf diese Weise ist es jedoch unter Umständen notwendig, dass diverse Anpassungen am Modell vorgenommen werden, wenn die additive Fertigung unter den gegebenen Voraussetzungen nicht anwendbar ist. Zum anderen könnten schon von niedrigen Detailstufen an (LOD100) die erweiterten Möglichkeiten sowie die Limitationen für den Bau durch AM berücksichtigt werden. Jedoch muss hierfür der Designer ausreichend Erfahrung mit den AM-Methoden besitzen, um schon im frühen Designstadium fundierte Entscheidungen treffen zu können. Eine weitere Beschreibung dieser Herangehensweisen und eine Antwort auf die zuvor gestellte Frage wird in Kapitel 4 gegeben.

2.2 Industry Foundation Classes (IFC)

Die an einem Bauprojekt beteiligten Teams arbeiten nicht zwangsweise mit denselben Softwarelösungen oder müssen gar für bestimmte Aufgaben unterschiedliche Software verwenden. Es gibt aufgrund der Vielfältigkeit der Aufgabenbereiche im Bauwesen keine allumfassende Softwarelösung, die alle Disziplinen in einem Tool vereint. Daher ist es unabdingbar, dass bei der Erstellung und Verfeinerung eines BIM-Modells ein Datenaustausch und unter Umständen auch Datenkonvertierungen stattfinden müssen, um die einzelnen Aufgaben bearbeiten zu können.

Kommerzielle Software nutzt in der Regel eigene proprietäre Datenformate, deren Austausch problematisch sein kann, z.B. wenn einer der am Projekt beteiligten Parteien eine nicht kompatible Software benutzt. Im besten Fall besteht die Möglichkeit, die Daten für einen Austausch zu konvertieren, jedoch kann es hierbei immer vorkommen, dass Daten verloren gehen. Häufig sind gewisse Konvertierungen aber nicht vorgesehen, sodass ein Austausch mit der entsprechenden Partei nicht möglich ist (vgl. Abschnitt 2.1). Ein BIM-Projekt kann natürlich auch ausschließlich mit kommerzieller Software durchgeführt werden, man spricht dann vom sogenannten „closedBIM“. In einem solchen Szenario kann allerdings durch Datenverlust ein erheblicher Mehraufwand entstehen; Laut einer 2004 durchgeführten Studie vom US-amerikanischen Institut für Standards und Technologie (NIST) betrugen die Gesamtkosten dieses Aufwands etwa 15,8 Mrd. US-Dollar für das Jahr 2002 und ist bis heute ein guter Vergleichswert (O'Connor et al., 2004).

Um eine herstellerneutrale Lösung zum Austausch von Bauwerksdaten zu ermöglichen, wurde hierzu das Datenmodell der Industry Foundation Classes (IFC) als Austauschdatenformat entwickelt. Angestoßen wurde dies unter anderem durch die Firma *Autodesk*, indem 1995 die International Alliance for Interoperability (IAI) – seit 2005 umgetauft in *buildingSMART* – gegründet wurde. In Version 1.0 wurde IFC im Jahr 1997 veröffentlicht, mit den ersten Implementierungen in Softwarepaketen für Bauzwecke ab Version 1.5 im

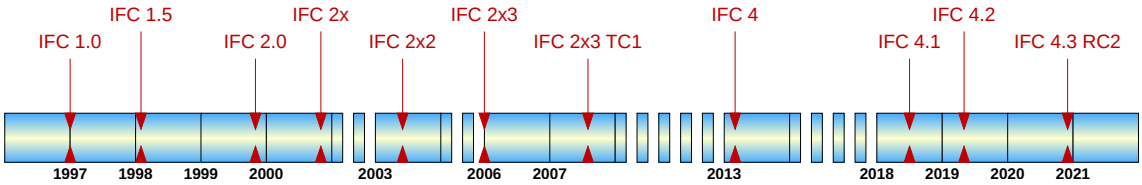


Abbildung 2.9: Versionsgeschichte des IFC Standards (nach BORRMANN et al., 2015b).

Jahr 1998. Seither wird der Standard stetig weiterentwickelt und Neuerungen hinzugefügt (vgl. Abb. 2.9) und liegt aktuell in Version 4.3 vor.

Bis einschließlich Version 4.0 wurde beim IFC-Format der Schwerpunkt auf die Darstellung von und Verwaltung von Gebäuden gelegt (BORRMANN et al., 2015b), jedoch wird diese Limitierung in den jüngsten Erweiterungen aufgebrochen. Mit dem Update auf Version 4.1 wurde eine Basis für Infrastrukturprojekte geschaffen, die mit Version 4.2 für Brückenbauwerke erweitert wurde. Schließlich – seit Version 4.3 – ist nun eine detaillierte Beschreibung von Infrastruktur Konstruktionen möglich. (BORRMANN et al., 2015b; »IFC Release Notes«, 2020)

Das IFC-Format kann lizenzgebührenfrei verwendet werden und hat dadurch und durch eine Normierung im Jahr 2013 als ISO 16739 (seit 2017 DIN EN ISO 16739) sehr schnell an Bedeutung gewonnen. Die Einführung von IFC hat wesentlich zur Realisierung von „openBIM“ beigetragen, also der Nutzung von offenen Austauschformaten, um Datenverluste bei Nutzung von Software verschiedener Hersteller zu vermeiden. Zudem sichert der offene Standard eine Datenlesbarkeit über eine sehr lange Zeit – ein nicht unwichtiger Vorteil im Anbetracht der üblichen Lebenszeit von Bauwerken (mehrere Jahrzehnte). (BORRMANN et al., 2015b)

2.2.1 Datenmodell und Dateiformat

Das IFC-Format wurde zunächst in Anlehnung an das im Maschinenwesen weit verbreitete Austauschformat STEP (**ST**andard for the **E**xchange of **P**roduct model data) entwickelt, welches auf der Datenmodellierungssprache EXPRESS basiert. Mit dieser deklarativen Sprache können objektorientierte Datenmodelle definiert werden und genauso wurde damit für IFC ein Entity-Relationship Modell geschaffen. Dieses besteht aus mehreren Hundert Entitäten und Relationen, die über eine Vererbungshierarchie geordnet sind, wodurch Eigenschaften an untere Ebenen weitergegeben werden können (vgl. Abb. 2.10). (BORRMANN et al., 2015b; LIEBICH et al., 2013)

Die Relationen im IFC-Schema sind in Form von objektifizierten Beziehungen definiert, d.h. sie werden nicht direkt ausgeführt, sondern über ein zwischengeschaltetes Objekt. Dieses Zwischenobjekt, welches die Relation beschreibt, gehört zu einer Vererbungshierarchie von Beziehungsklassen und lässt sich über seine Attribute noch weiter detaillieren (vgl. Abb. 2.11). Zudem lassen sich mit der EXPRESS Sprache inverse Beziehungen definieren, es werden hierzu nur Attribute angelegt, die in die umgekehrte Richtung weisen.

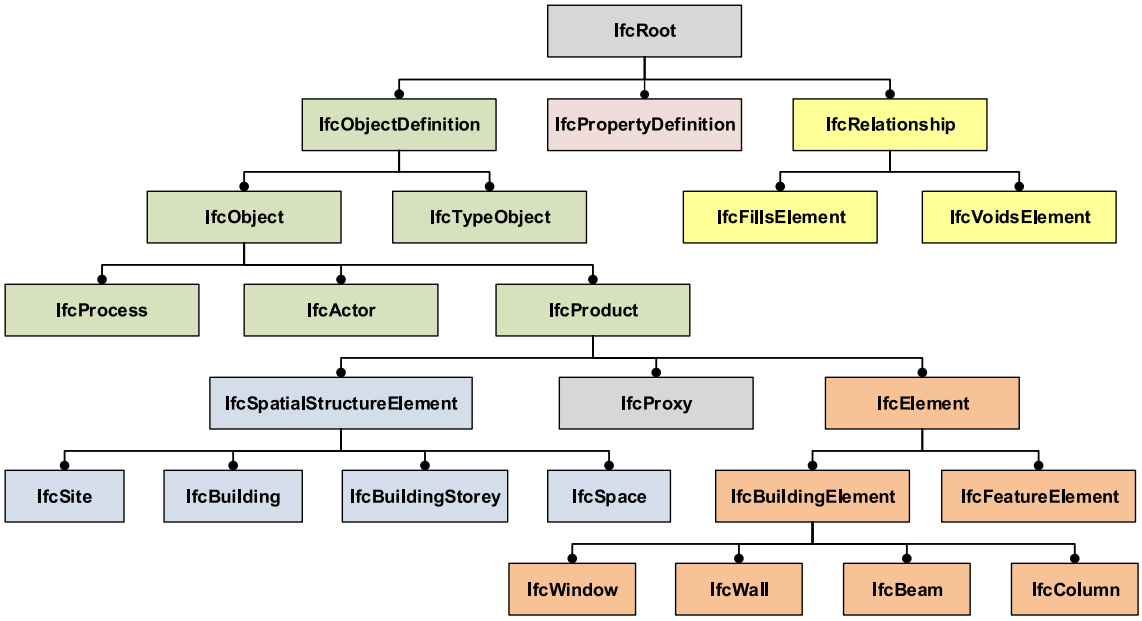


Abbildung 2.10: Wichtige Klassen aus dem IFC-Schema. (BORRMANN et al., 2015b)

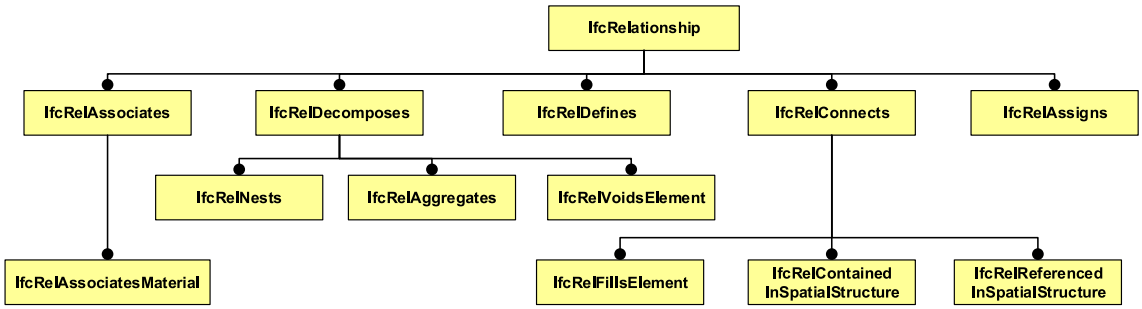


Abbildung 2.11: Hierarchie der Relationen im IFC-Schema. (BORRMANN et al., 2015b)

Mit den definierten Klassen können so reale Objekte sowie Prozesse, Regeln, Personen und andere für das Bauprojekt wichtige Daten abgebildet werden und über die Beziehungsklassen miteinander in Relation gebracht werden. Damit werden Bauteile nicht nur als einfache Objektsammlung dargestellt, sondern ein besonderer Wert auf die jeweiligen Funktionen und die Interaktion der Bestandteile gelegt.

Das IFC-Schema kann in unterschiedlichen Dateiformaten codiert werden, die meisten davon sind im `ASCII`-Format, also direkt über einen Texteditor lesbar. Am gängigsten um mit der besten Kompatibilität ist hierbei das STEP Physical Format (SPF), das über die Dateiendung „.ifc“ erkennbar ist. Andere `ASCII`-Typ Formate sind *ifcXML*, *ifcOWL* und *ifcJSON*. Auch wenn die Lesbarkeit im ASCII Format einen Vorteil darstellt, so gibt es einige Nachteile, die damit einhergehen. Unter anderem ist die Dateigröße hierdurch sehr groß – *ifcZIP*, also eine komprimierte Version von den IFC-Standardformaten, hat lediglich einen Speicherbedarf von 17% der SPF Version. Außerdem haben textbasierte Formate den entscheidenden Nachteil, dass Informationen nicht einfach aus der Dateimitte entnommen werden kann, also immer die gesamte Datei geparst werden muss. Eine binär-Version von IFC stellen die sich noch in der Entwicklung befindenden Formate *ifcHDF5*

und *ifcSQLite* dar, mit denen eben besagter Nachteil entfällt, also ein hoch effizienter Dateizugriff möglich sein wird. (BORRMANN et al., 2015b; BUILDINGSMART, 2020a)

2.2.2 Schichtstruktur des IFC-Standards

Jede der an einem Bauprojekt beteiligten Parteien bearbeitet einen eigenen Satz an Aufgaben, sodass jede Partei einen ganz eigenen Anspruch an das Austauschformat hat. Um der Komplexität Herr zu werden, die das IFC-Format daher haben muss, wurde das Datenmodell in mehrere Schichten eingeteilt. Eine solche Modularität erleichtert gleichzeitig die Wartung und eventuelle Erweiterungen. Die folgende Beschreibung wurde nach den Schilderungen von BORRMANN et al. (2015b) und LIEBICH et al. (2013) zusammengefasst:

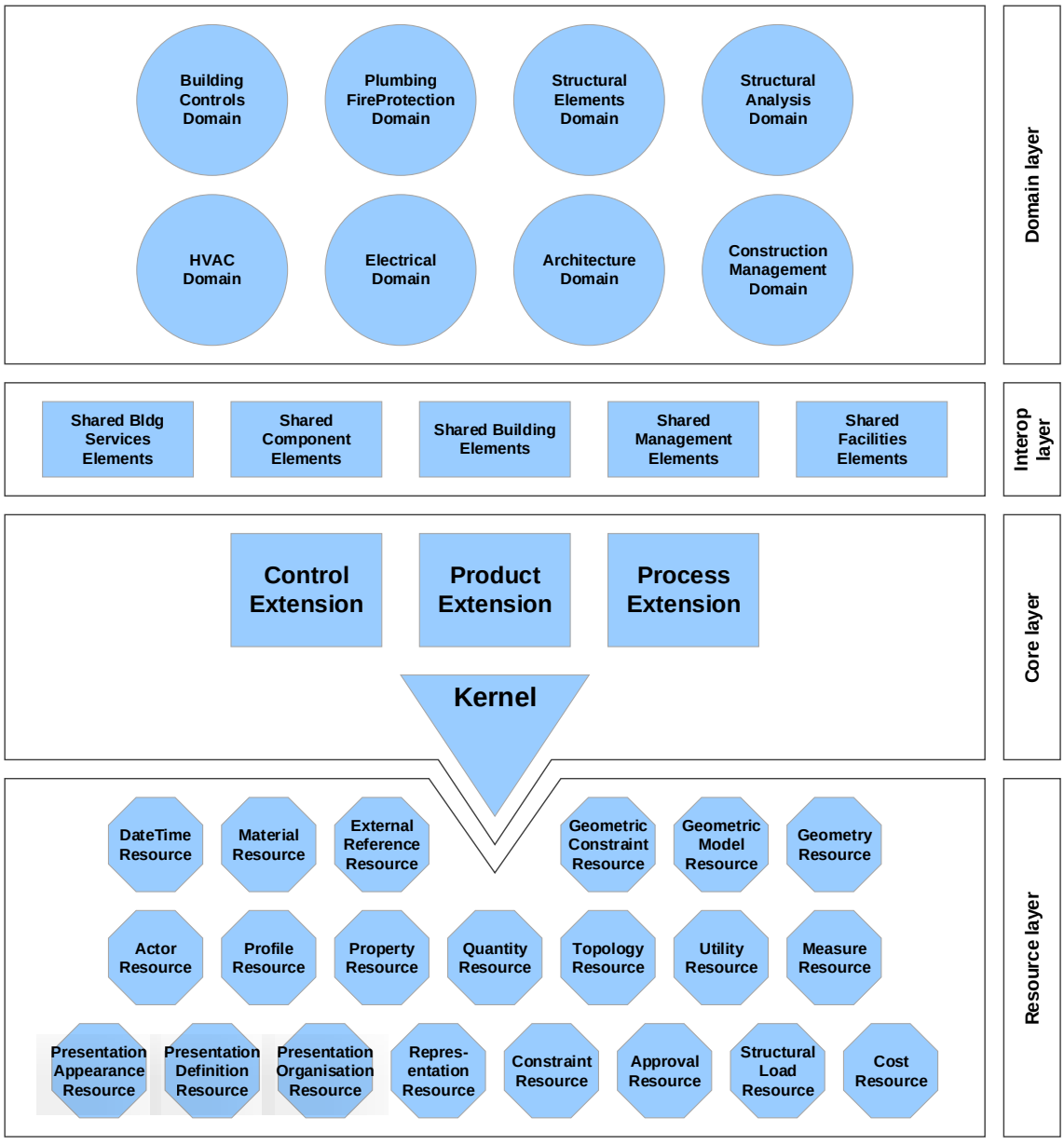


Abbildung 2.12: Schichtstruktur der IFC-Implementierung. Nach LIEBICH et al. (2013)

Wie in Abb. 2.12 dargestellt, ist das IFC-Schema in vier Schichten eingeteilt, die Domain-, Shared-, Core- und Resource Schicht. Diese Schichten können in derselben Reihenfolge auf die jeweils nächste Schicht verweisen, jedoch nicht in umgekehrter Folge.

In der obersten Schicht (Domain) werden dabei nur derart spezialisierte Klassen gesammelt, die nur jeweils einer Domäne zugeordnet werden können, also keinen Nutzen für andere Domänen haben. Auf der nächsten Schicht (Shared) werden Klassen gesammelt, die von mehreren Disziplinen in gleicher Weise benutzt werden können, z.B. wichtige Bauteilklassen.

Die dritte Schicht (Core) liefert alle grundlegenden Klassen für das Datenmodell. Hier werden die abstrakten Basisklassen definiert, von denen die Klassen höherer Schichten abgeleitet werden, darunter auch der Ausgangspunkt des Vererbungsbaumes *IfcRoot*. Mit dieser abstrakten Klasse wird einem Objekt eine eindeutige Identität (GUID) zugewiesen und stellt Attribute für Name und Beschreibung bereit.

Alle nicht von *IfcRoot* abgeleitete Klassen, also keine eigene Identität besitzen können, werden im Ressource Layer, der untersten Schicht, definiert. Dies sind grundlegende Datenstrukturen, wie etwa geometrische Basiselemente (z.B. ein Punkt oder Vektor) oder Elemente zur Beschreibung von Materialien, die über das gesamte Schema verwendet werden. Da diese keine eindeutige Identität aufweisen können, müssen sie immer von einer Instanz von einer Unterklasse von *IfcRoot* referenziert werden.

Mit diesem Datenmodell lassen sich mit besonderem Fokus für Bauprojekte sehr detaillierte digitale Modelle erstellen. Ein Alleinstellungsmerkmal von IFC sind seine Beziehungsklassen, die eine umfassende semantische Beschreibung von Bauwerken erlauben. Zudem ist es mit diesem Standard direkt möglich, ein multi-LOD-Modell (vgl. Abschnitt 2.1.2) abzubilden, da z.B. zu einem Bauteil gleich mehrere Geometrien zugewiesen sein können, gleiches gilt für andere semantische Informationen. Z.B. ist es bei Straßenbauprojekten durchaus sinnvoll, verschiedene Detailstufen gleichzeitig bereitzustellen, da einige Aufgaben womöglich ein hochaufgelöstes und fein detailliertes Modell der Straße voraussetzen, dagegen andere nur die Information des Alignments der Straße benötigen (VILGERTSHOFER & BORRMANN, 2017).

Im Bezug auf AM könnte das Datenformat IFC geeignet sein, um sowohl geometrische und semantische als auch Fertigungsinformationen abzubilden. Für das verwandte Datenformat STEP wurde im Jahr 2007 eine Erweiterung für Maschinensteuerungsinformationen (STEP-NC) als Standard veröffentlicht. Eine analoge Umsetzung für das IFC-Datenformat ist daher durchaus denkbar und auch ohne diese Erweiterungen könnten für die Darstellung von Fertigungsinformationen bereits einige der vorhandenen Klassen verwendet werden. Ein genauerer Überblick über die STEP-NC Erweiterung wird hierzu in Abschnitt 2.4.1 gegeben sowie die Verwendung von den bereits vorhandenen IFC-Klassen für AM konzeptionell in Kapitel 7 vorgestellt.

2.3 Additive Fertigung

Additive Fertigung (*englisch*: Additive Manufacturing, AM) ist ein Überbegriff für alle Methoden, bei denen Bauteile durch das computergesteuerte, in der Regel schichtweise Hinzufügen von Material erstellt werden. Es wird also für den Fertigungsprozess ein Endeffektor, oder einfach gesagt der Druckkopf, in die richtige Position gebracht und damit Material an dieser Stelle aufgebracht. Insgesamt kann damit eine sehr große Geometriefreiheit gewährleistet werden. (WEGER et al., 2018)

Diese, auch 3D-Druckverfahren genannte Methoden, werden seit 1987 zunächst in Form der **Stereolithografie** – entwickelt von der Firma *3D Systems* – kommerziell eingesetzt. Wenig später, im Jahr 1991, wurden weitere AM-Technologien auf den Markt gebracht, unter anderem die **Fused deposition modeling (FDM)** Methode. Die ersten AM-Methoden, neben Stereolithografie und FDM, waren hierbei zunächst beschränkt auf den Einsatz von Kunststoffen, jedoch wurden seither neue Verarbeitungsmethoden entwickelt und immer mehr Werkstoffe für AM erschlossen. Unter anderem können nun auch die wichtigen Baustoffe Beton und Stahl mit entsprechenden AM-Methoden verarbeitet werden.

Das Gegenstück zur Additiven Fertigung stellt die Subtraktive Fertigung dar. Bei dieser Methode wird, wie der Name schon verrät, Material von einem grob vorgefertigten Bauteil abgetragen, um die gewünschte Endgeometrie zu erhalten. Die ersten numerisch gesteuerten Maschinen dieser Art wurden bereits 1952 entwickelt (PEASE, 1952) und sind in der Industrie nunmehr schon seit Langem fest verankert.

Vergleicht man Additive und Subtraktive Fertigung miteinander, so kann man schnell die Vorteile der AM-Methoden verdeutlichen. Bei der Subtraktiven Herstellung ist das

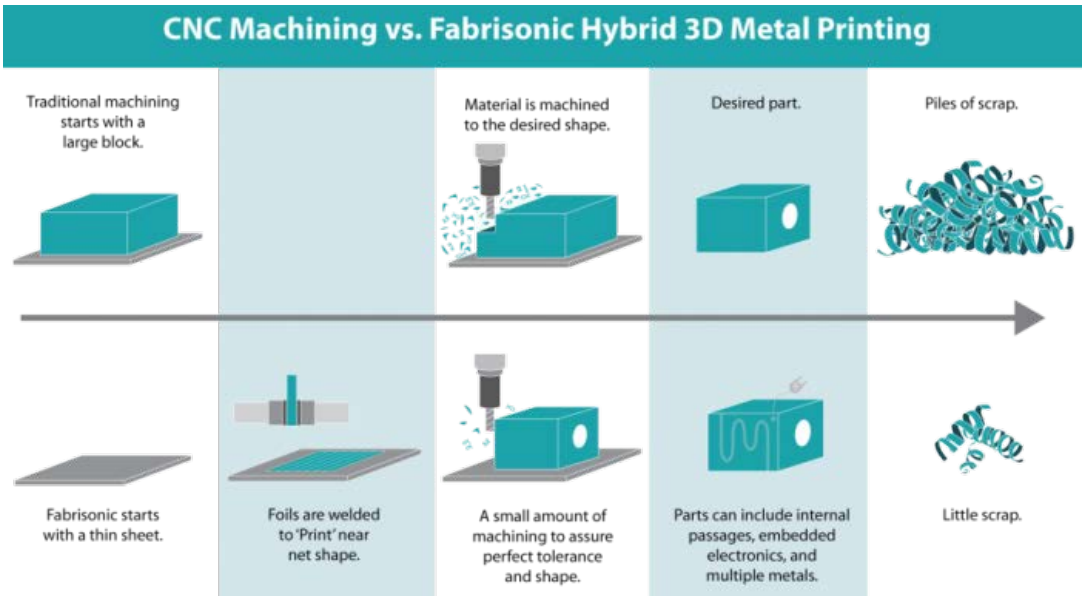


Abbildung 2.13: Vergleich zwischen Subtraktiver und Hybrider Fertigung, also Additiver Fertigung mit Subtraktiver Nachbearbeitung. (»Additive Manufacturing Processes Cheat Sheet [Draft]«, 2020)

Anfangsobjekt ein Materialkörper, der die gesamte Soll-Geometrie umschließt, von dem im Laufe des Prozesses immer weiter Material abgetragen wird. Ist die Zielgeometrie sehr komplex, so kann es sein, dass sehr viel Material abgenommen werden muss. Im Gegenzug baut die Additive Fertigung die Geometrie von Grund an auf und bis auf eventuelle Nachbearbeitungsschritte muss hierbei kein Material abgetragen werden. Zusätzlich ist es auch noch möglich, innere Strukturen im Bauteil zu verwirklichen. In Abb. 2.13 wird genau dieser Vergleich verdeutlicht.

Die mittlerweile vielen verschiedenen AM-Methoden lassen sich unter anderen nach ihrer Materialverarbeitung Kategorisieren. Für die Baubrache von besonderem Interesse sind hierbei die beiden Kategorien der **Partikelbett-Verfahren**, zu denen z.B. die Stereolithografie gehört, und der **Extrusionsverfahren**, wie etwa FDM. Diese beiden Kategorien werden im Folgenden etwas genauer erklärt und jeweils einzelne Verfahren vorgestellt.

2.3.1 Partikelbett-Verfahren

Bei Partikelbett-Verfahren wird immer zunächst eine ebene Schicht nicht eigenständig erhärtendes Material auf den Untergrund aufgetragen und anschließend nur an gewünschten Stellen zum erhärten gebracht, entweder mit einem Bindemittel, einer chemischen- oder physikalischen Aktivierung (vgl. Abb. 2.14). Einfach gesagt wird zunächst ein Partikelbett (Flüssigkeit oder Pulver) aufgebracht und nur an bestimmten Stellen eine Bindung erzeugt. Am Beispiel der Stereolithografie ist das Partikelbett eine lichtsensitive Flüssigkeit, die in dem zweiten Schritt mit einem Laserstrahl gezielt photochemisch aktiviert wird und zu einem Kunststoff polymerisiert wird (HENKE et al., 2016; WOHLERS & GORNET, 2014). Weitere für die Baubranche wichtigere Beispiele für Partikelbett-Verfahren sind folgende:

- **Selektive Zement Aktivierung (*englisch*: selective cement activation, SCA):**
  Hier wird eine Partikelschicht aus Zement gemischt mit der Gesteinskörnung aufgebracht und an den gewünschten Stellen durch Aufbringen von Wasser verfestigt. Je gröber die Gesteinskörnung, umso weniger geometrische Auflösung kann erzielt werden, jedoch kann so die Druckgeschwindigkeit erhöht werden. Allerdings muss beachtet werden, dass die Schichtdicke nicht größer als die maximale Eindringtiefe des Wassers sein darf.
- **Selektive Zementleim Intrusion (*englisch*: selective paste intrusion, SPI):**
  Hierbei wird als Partikelschicht allein die Gesteinskörnung (in der Regel Sand) aufgebracht, die durch Intrusion von Zementleim an den gewünschten Stellen ausgehärtet wird. Größere Gesteinskörnungen haben denselben Einfluss wie bei der SCA.
- **Selektives Laserschmelzen (*englisch*: Laser Powder-Bed Fusion, LPBF):**
  Hier wird als Partikelschicht Metallpulver aufgetragen, welches mit einem Laserstrahl lokal umgeschmolzen wird. Als Werkstoffe eignen sich hier u.a. Edelstahl, Titan, Edelmetalllegierungen, etc. (vgl. Abb. 2.21).

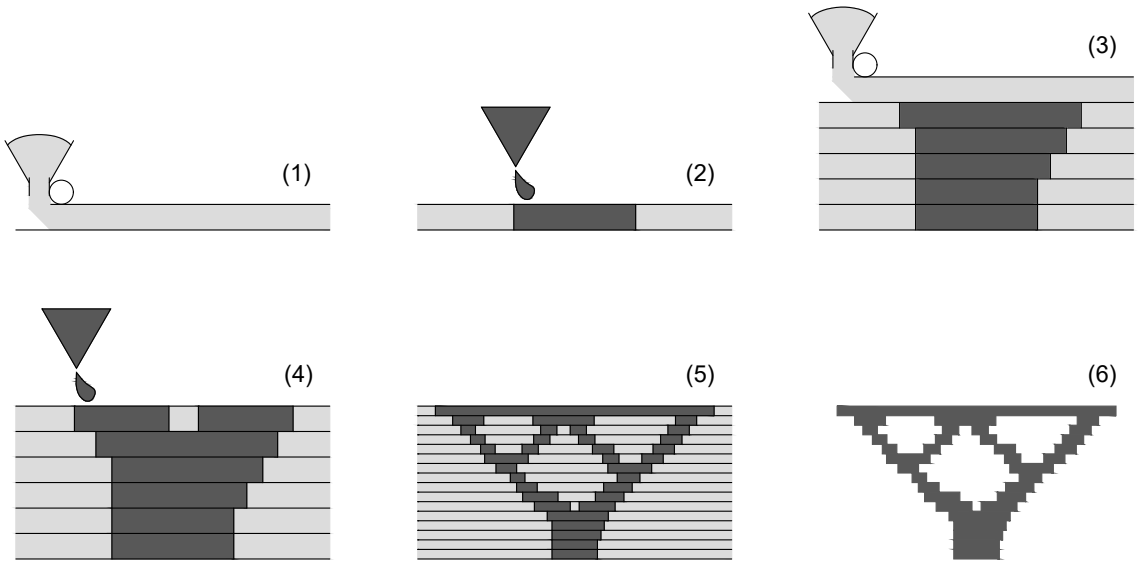


Abbildung 2.14: Funktionsprinzip der Partikelbett-Verfahren. (nach WEGER et al., 2018)

Ein Vorteil dieser Technik ist, dass durch das Partikelbett bereits eine Stützstruktur gewährleistet ist und damit eine sehr große Geometriefreiheit geboten wird. Weiterhin kann mit dieser Technik abhängig von der verwendeten Körnung des Partikelbetts eine sehr hohe Oberflächenauflösung erzielt werden (jedoch mit teils deutlich erhöhtem Zeitaufwand). In manchen Fällen ist es zudem problematisch, Partikelbettreste aus dem fertig gedruckten Bauteil zu entfernen, z.B. wenn geschlossene Strukturen erstellt werden sollen.

2.3.2 Extrusionsverfahren

Unter den Extrusionsverfahren versteht man Methoden, bei denen Material schichtweise der gewünschten Geometrie entsprechend computergesteuert abgelegt wird (WEGER et al., 2018). Der Ablegevorgang kann hierbei jedoch in ganz unterschiedlicher Form passieren, bei FDM wird hierbei ein thermoplastischer Kunststoff geschmolzen und schichtweise aufgetragen (vgl. Abb. 2.15, WOHLERS und GORNET (2014)). Als Werkstoffe kommen mittlerweile neben Kunststoffen unter anderem auch selbsthärtende Pasten (z.B. Beton) oder geschmolzener Stahl infrage. Beispiele für ablegende Verfahren sind:

- **Lichtbogen-Additive Manufacturing (*englisch*: Wire and Arc Additive Manufacturing, WAAM):**
  Bei diesem Stahl-Druckverfahren wird die Geometrie erzeugt, indem durch kontinuierliches Schmelzen eines Stahldrahtes durch einen Lichtbogen schichtweise eine Schweißnaht aufgebracht wird (RODRIGUES et al., 2019).
- **Extrusionsverfahren (*englisch*: Fused Layer Modeling, FLM):**
  Extrusion hat in der Verfahrenstechnik folgende Bedeutung: „Beim Extrudieren (Strangpressen) werden feste bis dickflüssige und härtbare Massen kontinuierlich aus einer Druckkammer durch eine profilierte Öffnung (Düse, Matrize, Mundstück) gepresst, die Form und Querschnitt bestimmt.“ (»Glossar«, 2020)

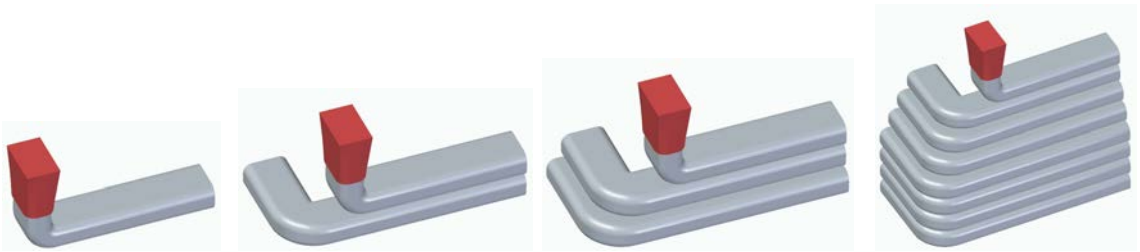


Abbildung 2.15: Vereinfachtes Funktionsprinzip der ablegenden Verfahren. (nach WEGER et al., 2018)

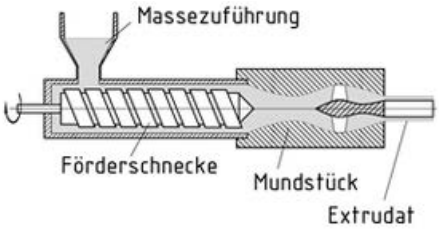


Abbildung 2.16: Schneckenextruder. (»Glossar«, 2020)

Genau auf diese Weise wird z.B. das pastöse Material Beton (in diesem Fall) über einen Schneckenextruder durch eine Düse gedrückt (vgl. Abb. 2.16) und strangförmig in Schichten aufeinander abgelegt (WEGER et al., 2018). Die Form und Größe der Düse kann hierbei variieren, je nach Anwendungsfall und Betonmischung (vgl. HENKE et al., 2020, S. 5). In einem aktuellen Forschungsprojekt wird außerdem daran gearbeitet einen Extruder zu entwickeln, in dem die Betonmischung erst kurz vor der Düse stattfindet. Auf diese Weise kann recht präzise, während dem Druckvorgang die Betonzusammensetzung verändert werden und somit eine „betoninterne“ Struktur erzeugt werden.

- **Shotcrete 3D Printing (SC3DP):**
  Shotcrete wurde bereits Anfang des 20. Jahrhunderts von Carl Ethan Akeley entwickelt (TEICHERT, 2002) und bezeichnet das pneumatische Auftragen von Beton über eine Spritzdüse. Ursprünglich wurde das Verfahren zur Reparatur einer Betonfassade entwickelt und kommt heute unter anderem im Tunnelbau zum Einsatz. Genau dieses Verfahren wird beim SC3DP durch das Anbringen einer solchen Düse an einen Roboterarm automatisiert.

2.3.3 Additive Fertigung im Bauwesen

Betrachtet man allgemein einen üblichen Bauvorgang, so ist der gesamte Prozess im eigentlichen Sinne eine Additive Fertigung, da dem Gebäude Stück für Stück zusätzliche Elemente – konventionell hauptsächlich in Handarbeit oder maximal teilautomatisiert – hinzugefügt werden. Über eine grundsätzliche Einsetzbarkeit der additiven Fertigung im Bauwesen muss also nicht diskutiert werden. Auch weil Gebäude in der Regel Einzelstücke darstellen, da sie speziell für den entsprechenden Baugrund und den zugewiesenen

Zweck konzipiert werden, ist es sinnvoll, auf AM-Methoden zurückzugreifen. Zudem steht dem Architekten unter Anwendung von AM eine deutlich breitere Geometriefreiheit zur Verfügung und nachdem neben dem 3D-Drucker keine besonderen Werkzeuge benötigt werden, können auch sehr kostengünstig viele unterschiedliche Einzelteile erstellt werden (vgl. Abschnitte 2.3, 2.3.1 und 2.3.2). Mit konventionellen Methoden müssten hier z.B. spezielle Schalungen erstellt werden, die im Nachgang keinen weiteren Nutzen mehr haben und somit auch noch aufwendig entsorgt werden müssen. Des Weiteren besteht mit AM die Möglichkeit, Bauteile in der Form zu optimieren, also z.B. die Tragfähigkeit zu steigern oder die Materialkosten zu verringern, sowie mit zusätzlichen Funktionen zu versehen, wie z.B. eine innere Struktur zur Verringerung des Wärmedurchflusses. (HENKE, 2016)

Anhand der vielen Vorteile beim Einsatz von AM stellt sich die Frage: „Warum wird AM nicht standardmäßig beim Bau eingesetzt?“ Es gibt hier einige entscheidende Hindernisse, die einen Einsatz dieser Technologie bisher verhindert haben. Zwei Probleme zeichnen sich dabei besonders ab, zum einen eignen sich nicht alle Werkstoffe für diese Form der Verarbeitung und zum anderen gibt es bei den Druckprozessen Einschränkungen bezüglich der Größe der zu erstellenden Bauteile. Seit den Anfängen des 3D-Drucks steigt jedoch die Anzahl an Forschungsprojekten diesbezüglich kontinuierlich an, daher ist es abzusehen, dass diese Hindernisse bis hin zur Praxistauglichkeit der AM-Methoden überwunden werden. (HENKE, 2016)

Eines der ersten AM-Systeme, die im Bauwesen eingesetzt werden, ist das Contour Crafting. Diese Technik wurde in den 1990er Jahren an der USC Universitiy in Süd-Kalifornien entwickelt und wird seither von der Firma *Contour Crafting Corporation* weiterentwickelt (CC CORP, 2017; KHOSHNEVIS, 1999). Mit dieser Technik werden die Konturen vom Druckobjekt 3D-gedruckt (Extrusionsdruckverfahren, vgl. Abschnitt 2.3.2) und entstehende Hohlräume mit Frischbeton oder isolationsfähigem Material verfüllt. Mit Hilfe am Druckkopf angebrachter Traufeln wird zudem die Oberfläche des Druckkörpers direkt während dem Druckvorgang zu einer weitestgehend glatten Fläche nachbearbeitet. Weitere Aufgaben, wie z.B. das Einbringen von Bewehrungen und die Integration von Haustechnik, kann hiermit auch gleichzeitig über einen separaten Greifarm realisiert werden (CC CORP, 2017). Eine erste technische Umsetzung zum Einsatz der selektiven Aktivierung von Beton im Bauwesen ist der 3D-Drucker *D-Shape* vom Bauingenieur Enrico Dini. Mit diesem Drucker können auf einer Fläche von nun bis zu 12 m x 12 m Gesteinkörnungen selektiv gebunden werden (DINI, n. d.).

Nachfolgend zu den oben genannten Techniken wurden in den letzten Jahren immer mehr Werkzeuge für Additive Manufacturing entwickelt (vgl. Abb. 2.17) und eine Vielzahl an Beispielprojekten durchgeführt, bei denen AM zum Einsatz kam. Hier ist z.B. das chinesische Unternehmen WinSun zu nennen, die bereits komplette Bauwerke additiv gefertigt haben, mitunter ein fünfstöckiges Haus. Aktuell setzt WinSun seine Technologie ein, um schnell Quarantänestationen zur Bekämpfung des Coronaviruses 3D zu drucken (WINSUN, 2020). Auch in Deutschland wird aktuell ein AM-Bauprojekt unternommen, bei dem mit einem Portaldrucker in Contour Crafting Bauweise die Wände eines Einfami-

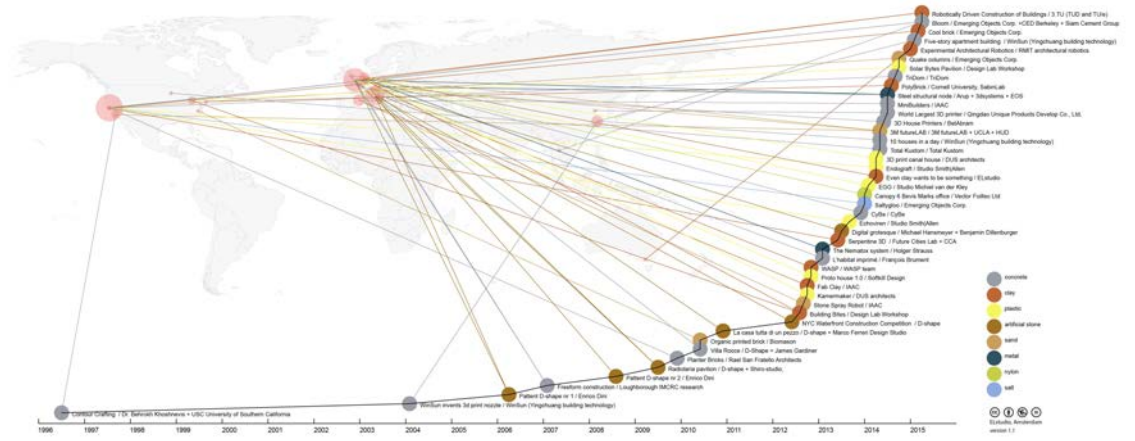


Abbildung 2.17: Übersicht über die Entwicklungen der letzten 20 Jahre im Bereich AM. Auf der vertikalen Achse (hier nicht dargestellt) wird die Anzahl der bisher gestarteten Forschungsprojekte angezeigt. (LANGENBERG, 2015)

Abbildung 2.18: 3D-Gedruckte Wand mit integrierter Treppe. (PA, 2019a)

lienhauses gedruckt werden (BAULINKS, 2020). Die Wände werden hierbei dreischalig gedruckt und später mit Dämmmaterial verfüllt. Alle Anschlüsse und Rohrleitungen werden direkt beim Druckprozess mit verlegt, ein nachträgliches Einbauen in die ausgehärtete Betonwand ist somit nicht nötig.

Die meisten der bisher 3D-gedruckten Bauwerke sehen bis auf die Oberflächenbeschaffenheit nicht viel anders aus als konventionell gebaute Objekte. In Zukunft ist es aber denkbar, dass mithilfe der 3D-Druck-Technologie ganz neuartige Konzeptideen auf dem Bau Einzug finden. Hierzu gibt es bereits eine Vielzahl an Beispielprojekten, anhand derer die Gestaltungsfreiheiten verdeutlicht werden. Die Abbildungen 2.18 und 2.19 zeigen hierzu recht eindrucksvoll die erweiterten Möglichkeiten, die der Einsatz von 3D-Betondruck liefert.

Abbildung 2.19: Freiformwand designed von Architekt Luai Kurdi. (PA, 2019b)

2.3.4 Aufbau eines AM-Systems

Prinzipiell wird der Druckvorgang bei allen AM-Verfahren in ganz ähnlicher Weise durchgeführt, der Endeffektor, bzw. einfach gesagt der Druckkopf, wird jeweils in die richtige Position gebracht und dort Material aufgebracht (vgl. Abschnitte 2.3.1 und 2.3.2). Es muss hierzu also der Druckkopf mittels einer Maschine versetzt und diesem dann das Material mittels geeigneter Beförderungsmechanismen zugeführt werden können.

Beförderungsmechanismus

Je nach Material können für die Beförderung verschiedene Systeme genutzt werden, sowohl mechanische, z.B. Förderschnecken, als auch pneumatische, z.B. Überdruck oder Unterdruck Pumpsysteme, kommen dabei zum Einsatz. Wird mit flüssigem oder pastösem Material gearbeitet, so werden in der Regel Pumpsysteme über Rohrleitungen und Schläuche eingesetzt (NÄTHER et al., 2017). Im Fall von Betonextrusion wurden für verschiedene Betonmischungen rheologische Untersuchungen durchgeführt, inwiefern diese **pumpfähig**, sowie anschließend **extrudierbar** und **verbaubar** sind (MECHTCHERINE & NERELLA, 2019; OGURA et al., 2018). Feststoffe können in Form eines Drahtes oder Filaments mechanisch über eine Spindel zugeführt werden (vgl. RODRIGUES et al., 2019), oder als Pulver mechanisch mittels Förderschnecken oder pneumatisch im Gasstrom (vgl. Abb. 2.20). Bei additiver Fertigung von Metallen kann über die eben genannte Gasstromzufuhr gleichzeitig Inertgas für den Schmelzprozess eingeleitet werden (THOMPSON et al., 2015).

Im Fall von selektiv bindenden Verfahren kommt unter Verwendung von Feststoffen als Ausgangsstoff zur Materialzufuhr auch noch ein **Verteilungsmechanismus** hinzu, mit dem ein

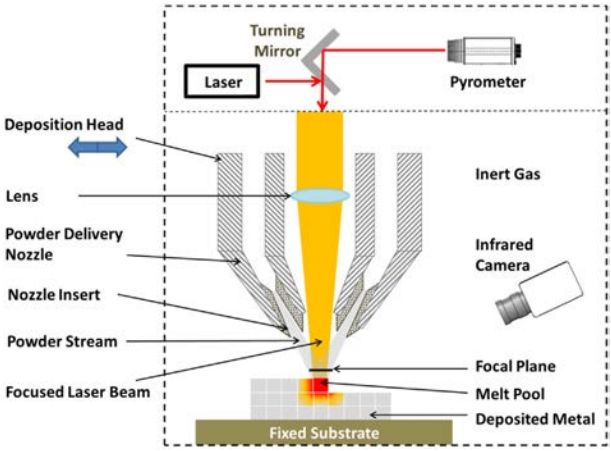


Abbildung 2.20: Direktes Laserschmelzen mit Zufuhr des Metallpulvers über Gasstromdüse. (THOMPSON et al., 2015)

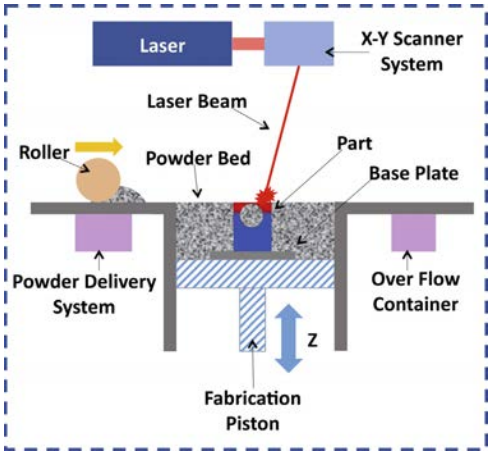
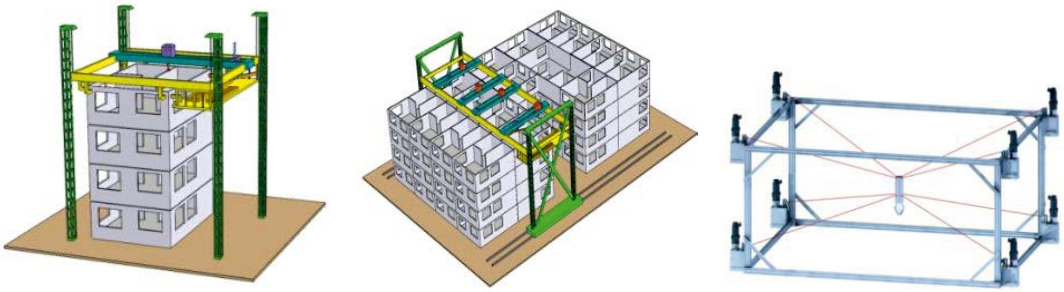


Abbildung 2.21: Mechanismus für das selektive Laserschmelzen. (THOMPSON et al., 2015)

ebenes Partikelbett erzeugt werden kann. Hierzu wird häufig eine Walze oder ein Rechen verwendet, womit ein am Rand aufgebrachter Partikelhaufen über den gesamten Bauraum verteilt wird (vgl. Abb. 2.21). Die Materialvorgänge bei diesem Verteilungsprozess sind hierbei Thema aktueller Forschungsprojekte, dazu wurde unter anderem der Verteilungsprozess mit einem Rechen mittels Diskreter Elemente Methode (DEM) Simulationen und Experimenten untersucht, um den Einfluss der Partikelgröße und der Schichthöhe auf die Verteilbarkeit zu bestimmen (CHEN et al., 2019). In ähnlicher Weise untersuchten ZHANG et al. (2017) mittels DEM Simulationen, Machine Learning Vorhersagen und Experimenten die Verteilung von *Ti-6Al-4V*-Pulver (eine hochfeste Titanlegierung) durch eine Walze bei verschiedenen Bewegungs- und Rotationsgeschwindigkeiten, um Vorhersagen über die Partikelverteilung und damit einhergehend über die Oberflächenrauheit treffen zu können. Als Letztes sei hier noch die Arbeit von WEI et al. (2018) zu nennen, hier wurde eine Modifikation der LPBF Methode entwickelt, mit der mehrere Metalle gleichzeitig gedruckt werden können.

Endeffektor Positionierung

Die Auswahl der Maschine für die Endeffektor Positionierung ist allein davon abhängig, wie komplex der abzufahrende Druckpfad ist, um das gewünschte Objekt mit der entsprechenden Druckmethode zu erstellen. Es bieten sich verschiedene Maschinensysteme an, je nachdem über wie viele Freiheitsgrade der Endeffektor verfügen soll. NÄTHER et al. (2017) beschreiben hierzu in ihrem Abschlussbericht eine Auswahl von verschiedenen Maschinenkonzepten, darunter werden **Portalroboter**, einachsig fahrbarer **Portalkran**, **Seilroboter** und **Autobetonpumpen** als Manipulatoren genannt, jeweils mit den entsprechenden Vor- und Nachteilen. Auch **Industrieroboter** können für den 3D-Druck eingesetzt werden (vgl. HENKE, 2016) sowie Kombinationen aus den genannten Systemen. Abbildung 2.22 zeigt die oben genannten Maschinensysteme.



(a) Portalroboter (b) Portalkran (c) Seilroboter

(d) Autobetonpumpe (e) Industrieroboter

Abbildung 2.22: Auswahl an möglichen Manipulatoren für den Einsatz in der additiven Fertigung. (HENKE, 2016; NÄTHER et al., 2017)

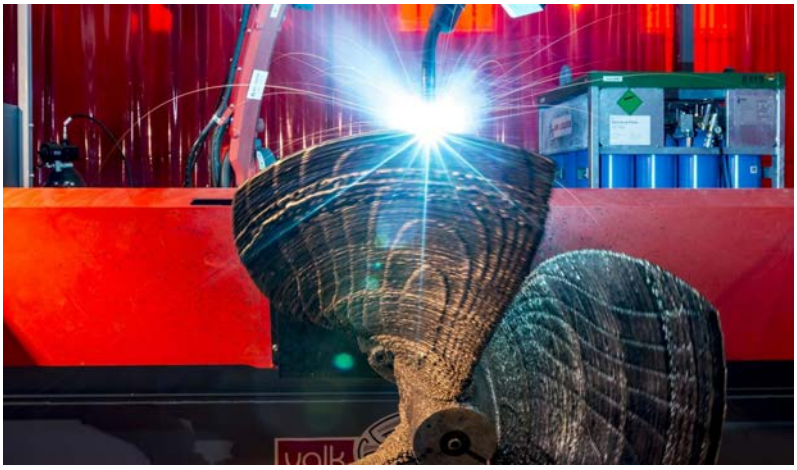
An dieser Stelle lässt sich ein Unterschied zwischen den beiden in Abschnitte 2.3.1 und 2.3.2 vorgestellten Grundmethoden zeigen. Bei den **selektiv bindenden Verfahren** wird aufgrund des notwendigen Partikelbetts (vgl. Abschnitt 2.3.1) immer in horizontalen Schichten gedruckt und damit muss der Endeffektor auch nur in x-, y- und z-Richtung bewegt werden. Zudem beschränkt sich der Bauraum auf das Partikelbett, also einen fest eingegrenzten Raum. Wohl aus diesen Gründen wird bei selektiv bindenden Systemen in der Regel ein Portalroboter zur Positionierung eingesetzt. Ein recht anschauliches Beispiel

Abbildung 2.23: D-shape Druckersystem. (DINI, n. d.)

hierfür ist das seit 2004 entwickelte Maschinensystem D-Shape (vgl. Abb. 2.23). Hier wird ein Portalroboter für die Positionierung des Endeffektors benutzt, der mittels eines Multidüsensystems über die gesamte Breite des Bauraums drucken kann (DINI, n. d.).

Bei den **ablegenden Verfahren** kann der Werkzeugpfad im Vergleich zu den selektiv bindenden Verfahren deutlich komplexer ausgeführt werden, da das Material beim Extrusionsvorgang direkt aufgetragen wird und somit – natürlich eingeschränkt durch Materialeigenschaften und Einwirkung der Schwerkraft – in alle Raumrichtungen gefahren werden kann (vgl. Abb. 2.24). In Abb. 2.24a und 2.24b wird hierbei mit besonders schnell aushärtenden Materialien gearbeitet, wodurch besonders viel Freiheit in der Pfadgestaltung gegeben ist. Dagegen wird in Abb. 2.24c mit Beton gearbeitet, der in pastöser Form

(a) Freie Kunstharz Extrusion. (MATAERIAL, 2016)



(b) 3D-gedruckte Schiffsschraube mittels WAAM. (RAMLAB, 2017)

(c) 3D gedruckter Überhang. (STEVENSON, 2020)

Abbildung 2.24: Beispiele für Druckkörper, die mit ablegenden Verfahren in nicht ebener Ausführung erzeugt wurden.

schichtweise aufgetragen wird. Im gezeigten Beispiel würde beim Druck von ausschließlich horizontalen Schichten der Druckkörper in sich zusammenfallen, da im oberen Bereich des Bogens aufeinanderfolgende Schichten zueinander einen sehr großen Überhang aufweisen würden. Damit die Bogenstruktur dennoch erfolgreich gedruckt werden kann, muss, wie im Bild zu sehen ist, segmentweise auch parallel zu einer schrägen Bezugsebene gedruckt werden (mehr dazu in Kapitel 5). Abbildung 2.24b zeigt außerdem, dass bei der Anwendung von ablegenden Verfahren auch auf unebenen Oberflächen gedruckt werden kann, hier werden auf einen Zylinder in ellipsoiden Bahnen die Rotorblätter aufgedruckt.

In den gezeigten Anwendungsfällen muss der Endeffektor neben der x, y, z-Positionierung zusätzlich noch in alle Richtungen geneigt werden können, hierzu ist von den gezeigten Manipulatoren (Abb. 2.22) ohne Einschränkungen nur der Industrieroboter (2.22e) fähig und mit Limitationen auch der Seilroboter (2.22c) und die Autobetonpumpe (2.22d). Ein Nachteil des Industrieroboters ist jedoch seine recht geringe Reichweite, ein mittelgroßer Roboter der Marke *KUKA* hat laut Herstellerinformationen eine maximale Reichweite von etwa 3 m (KUKA, 2017), somit muss das zu erzeugende Bauteil innerhalb einer Halbkugel mit dem Radius 3 m fertigbar sein. Benötigt man einen größeren Bauraum, so lässt sich ein solcher Roboter jedoch gut mit anderen Systemen kombinieren, wie z.B. eine Schienenbefestigung (zusätzliche Bewegungsmöglichkeit entlang einer Achse), eine Befestigung an einem Portalsystem (Bewegungsmöglichkeit entlang bis zu drei zusätzlichen Achsen) oder gar eine Montage auf einer frei fahrbaren Plattform.

Anwendungsort

Je nach eingesetztem Druckverfahren und Manipulator lässt sich additive Fertigung in unterschiedlichen Szenarien einsetzen. Zum einen können mit den beschriebenen Methoden fernab der Baustelle Bauteile in einer entsprechend Konzipierten Fabrik vorgefertigt werden (**off-site Vorfertigung**). Auf diese Weise kann eine einheitliche Bauumgebung geschaffen (z.B. konstante Temperatur und Luftfeuchtigkeit) und damit eine gewisse Qualität gewährleistet werden. Allerdings müssen nach dem Druckprozess die meist unhandlich schweren Bauteile zum Einsatzort transportiert werden. Dies ist besser handhabbar in Bastellennähe (**on-site Vorfertigung**), etwa in einem eigens dafür eingerichteten Zelt. Hier müssten nur die Rohstoffe über entsprechend weite Transportwege geliefert werden, nicht das fertige Bauteil, allerdings muss hierfür genügend viel Platz vorhanden sein und es kann nur eingeschränkt für konstante Bauverhältnisse gesorgt werden. Letztlich kann auch das Gebäude direkt auf der Baustelle gedruckt werden (**in-situ**). Auf diese Weise muss allerdings gewährleistet werden, dass das gesamte Gebäude sich innerhalb des Operationsbereichs des eingesetzten Manipulators befindet.

2.4 Maschinensteuerung

Um eine additive Fertigungsmethode durchzuführen, muss, wie zum Teil schon in Abschnitt 2.3 beschrieben, ein mit einem entsprechenden Druckkopf ausgerüsteter Bewe-

gungsapparat oder Roboterarm (Manipulator) sich auf bestimmte Art und Weise bewegen und an fest definierten Positionen Materie auftragen bzw. aktivieren. Die zum 3D-Druck notwendigen Bewegungsmuster und Aktionen werden hierzu der Maschine einprogrammiert, sodass alle Aktoren (Motoren für die Bewegung und das AM-Werkzeug) in entsprechender Weise gesteuert bzw. geregelt werden. Bei mehrachsigen Systemen – also bei allen Maschinen, die sich in mehr als eine Richtung bewegen können – müssen hierzu alle Achsen aufeinander abgestimmt werden, um eine ruckfreie und möglichst präzise Bewegungsabfolge zu erzielen, die Gesamtbewegung wird also auf die einzelnen Achsen und Gelenke umgerechnet. Hierzu kann z.B. ein Steuercomputer eingesetzt werden, der zunächst die einzelnen Achszustände über den zeitlichen Verlauf berechnet, diese Informationen dann an eine Steuereinheit übermittelt und damit die Achsen über Regler entsprechen bewegt (z.B. computergesteuerte Autobetonpumpe: NÄTHER et al., 2017, S. 28–31).

Hochpräzise Werkzeugmaschinen für industrielle Fertigungsbetriebe sind hierfür mit einer CNC-Steuereinheit ausgestattet (vgl. Abschnitt 2.4.1), durch die beliebige Bahnabfolgen mit einfachen Programmierverfahren realisiert werden können. In sehr ähnlicher Weise wird dies für Industrieroboter gehandhabt, über eine RC-Steuereinheit (vgl. Abschnitt 2.4.2) werden alle Achsbewegungen, die in diesem Fall auf komplizierte Art und Weise zusammenhängen, geregelt (BARTENSCHLAGER et al., 2013). Auf den genauen Unterschied zwischen RC und CNC wird später noch im Detail eingegangen. Es ist jedoch auch möglich, CNC-Befehle in RC-Anweisungen zu übersetzen, jedoch sind hier einige Einschränkungen zu beachten. Für alle im Abschnitt 2.3.4 beschriebenen Manipulatoren kann also prinzipiell ganz einheitlich eine CNC-Steuerung verwendet werden, im Folgenden wird daher diese Art der Steuerung näher beschrieben. Im Anschluss daran wird zudem die Robotersteuerung genauer beschrieben, da in dieser Arbeit vorrangig der Einsatz von einem Roboterarm für den Beton-3D-Druck behandelt wird. Im Detail wird besonders der Roboter **UR10e** von *Universal Robots* betrachtet, da dieses Modell für den praktischen Teil dieser Arbeit eingesetzt wurde und mit diesem Modell einfach verschiedene Methoden zur Steuerung getestet werden können aufgrund guter Dokumentation.

2.4.1 Rechnergestützte numerische Steuerung (CNC)

Der Begriff Rechnergestützte Numerische Steuerung (*englisch*: Computerized Numerical Control, CNC) bezeichnet eine Ablaufsteuerung von Werkzeugmaschinen, bei der digitale numerische Informationen in reale Achsbewegungen umgewandelt werden. Es ist eine computergestützte Steuerungstechnik, die sich seit den 1970er Jahren etabliert hat und eine Weiterentwicklung des analogen Vorgängers, der Numerische Steuerung (NC), darstellt. Die Bahnsteuerung der elektronisch gesteuerten Werkzeugmaschinen wurde zuvor noch durch das Einlesen von Lochstreifen realisiert (NC) und dann schließlich mit der Entwicklung von ausreichend leistungsfähigen Mikrochips durch computergestützte Regelung (CNC) ersetzt. Heute werden die beiden Begriffe synonym verwendet, d.h. sowohl CNC als auch NC beschreiben die computergestützte Steuerung von Werkzeugmaschinen, da mittlerweile alle derartigen Maschinensysteme nach demselben Prinzip aufgebaut sind. Die ersten derartigen Maschinen wurden in der Luftfahrtindustrie zur Fertigung komplexer

Bauteile eingesetzt und finden heute in fast allen Fertigungsbereichen eine Anwendung. (CARLSSON, 1984; HEHENBERGER, 2020)

Der CNC-Steuereinheit, welche als eine Art Interpreter zu verstehen ist, werden Eingangsdaten übermittelt, die dann in Signale zur Steuerung der Achsantriebe übersetzt werden. Für die Eingabe der Daten gibt es je nach CNC-Maschine verschiedene Möglichkeiten, sofern vorhanden direkt per Bedienpult oder in Form eines NC-Programms. In jedem Fall werden für den Fertigungsvorgang alle erforderlichen Positionswerte direkt als Koordinaten im Bezugskoordinatensystem der Steuerung eingegeben und eventuelle Bahnführungen zwischen diesen Positionen definiert. (AHRBERG et al., 2011; HEHENBERGER, 2020)

Bei den Bahnführungen unterscheidet man in drei verschiedene Steuerungsarten, der **Punkt-**, **Strecken-** und **Bahnsteuerung**. Unter Anwendung der Punktsteuerung fährt das Werkzeug von Startpunkt zu Endpunkt im Eilgang, also mit maximaler Achsgeschwindigkeit, auf **unbestimmtem Fahrweg** (der genaue Weg ist nicht vorgegeben) zu. Mit der Streckensteuerung lassen sich die Achsgeschwindigkeiten programmieren, werden aber sequenziell angesteuert, d.h. es sind hier nur achsparallele Bewegungen möglich. Die Bahnsteuerung stellt hier die wichtigste Variante dar, hier können beliebige Bewegungsmuster realisiert werden mit zwei oder mehr gleichzeitig geregelten Achsen. Ein Interpolator berechnet hierfür alle nötigen Zwischenpositionen (vgl. Abb. 2.25) und regelt die an der Bewegung beteiligten Achsen in der Art, dass alle gleichzeitig die gewünschte Zielposition erreichen.

Bei einem AM-Prozess wird also der zuvor festgelegte Druckpfad mittels Bahnsteuerung abgefahren. Hierzu wird das zu erstellende Bauteil zuvor am Computer 3D-modelliert und aus dem resultierenden Modell der Werkzeugpfad abgeleitet. Häufig wird zum Ableiten des Druckpfads das sogenannte **Slicing** eingesetzt, dabei wird die 3D-Geometrie des Bauteils in bestimmten Abständen (passend zum gewählten AM-Verfahren) in Scheiben, bzw. 2D-Flächen, geschnitten. Die 2D-Flächen werden dann über einen Algorithmus durch (optimierte) raumfüllende Kurven ersetzt, denen beim Druckprozess das Druckwerkzeug folgen soll. Der so erstellte Druckpfad wird dann in einzelne Wegpunkte diskretisiert und diese an die Maschine übermittelt, mit der Direktion, dass alle Wegpunkte über die Bahnsteuerung abgefahren werden sollen. Wie zwischen den Wegpunkten gefahren werden soll, wird von der Maschine wie beschrieben interpoliert (vgl. Abb. 2.25). Die Übermittlung

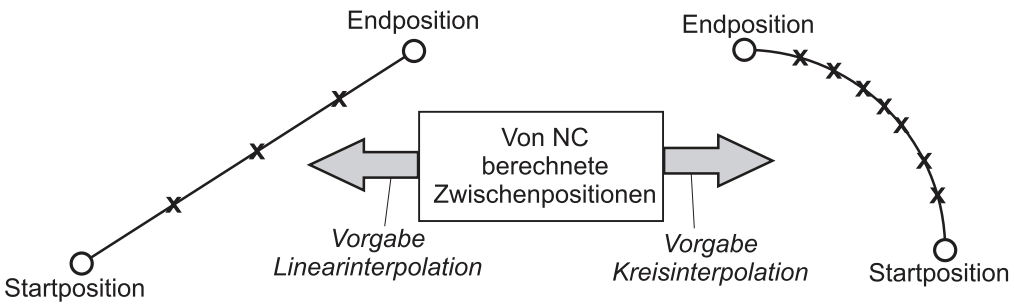


Abbildung 2.25: Interpolation von Zwischenpositionen bei Bahnführung. (HEHENBERGER, 2020)

der Wegpunkte und der Direktion zur Werkzeugmaschine erfolgt in der Regel über eine Art Ablaufbeschreibung, dem sogenannten **G-Code**, in Form einer ASCII codierten Textdatei. Im Fall von Roboterarmen kann dies mittlerweile zwar genauso durchgeführt werden, wird aber normalerweise etwas anders gehandhabt, wie in Abschnitt 2.4.2 erläutert.

NC-Programm nach DIN 66025/ISO 6983 (G-Code)

Ein NC-Programm stellt im Wesentlichen nichts anderes dar als eine Art Ablaufbeschreibung in Form einer Wegbeschreibung (Zielpositionen bzw. Wegpunkte und Bahnführung), eventuell angereichert mit Zusatzinformationen, wie z.B. welches Werkzeug wann und wie eingesetzt werden soll. Als Grundlage hierfür hat sich der Standard **DIN 66025/ISO 6983**, der sogenannte G-Code, durchgesetzt. Mit dieser Programmiersprache wird der Code in **Sätze** eingeteilt, die jeweils für sich eine geschlossene Maschinenoperation definieren und hierzu in einzelne **Wörter** aufgeteilt sind, die Positionen und Bahnführung sowie maschinenspezifische Aktionen beschreiben.

Die Wörter werden in Adressschreibweise mit einem Adressbuchstaben und Zahlenwert geschrieben, dabei gibt der Buchstabe an, welche Funktionsgruppe mit dem Wort ausgeführt werden soll. Z.B. wird mit dem Adressbuchstaben G die **Wegbedingung** definiert, und stellen zusammen mit den Wörtern für die Koordinaten (X, Y, Z, etc.) den geometrischen Teil der NC-Programmierung dar. Andere Funktionsgruppen sind unter anderem für Werkzeuginformationen (T), Vorschubgeschwindigkeit (F) und maschinenspezifische Hilfsfunktionen (M) zuständig.

Von den G-Wörtern sind unter anderem die Worte `G00`, `G01` und `G02/03` von besonderem Interesse. Hiermit werden die zuvor beschriebenen Steuerungsarten Punktsteuerung (G00) und Bahnsteuerung (G01–G03) angewendet, die Streckensteuerung findet in der Regel keine Anwendung mehr, da dasselbe Verhalten auch mit wenig mehr Aufwand durch die Bahnsteuerung realisiert werden kann. Die Worte, die eine Bewegung einleiten, werden in einem NC-Programm immer gefolgt von Positionsworten (z.B. `X100 Y100`), die den anzusteuernden Endpunkt der Maschinenoperation darstellen.

Die Programmiersprache nach dieser Norm ist jedoch sehr limitiert, da die Sprache auf die damals begrenzte Rechnerleistung angepasst wurde (erste Implementierungen in den 1950er Jahren) und heutige Möglichkeiten nur begrenzt ausgenutzt werden können. Nach DIN 66025/ISO 6983 müssen z.B. alle komplexen Bahnen (z.B. Splines) durch Geraden, Kreisbögen oder Parabeln approximiert werden. Dies führt je nach geforderter Komplexität und Genauigkeit zu einer sehr hohen Anzahl von Bahnsegmenten und damit zu sehr vielen Sätzen, welche kaum fehlerfrei per Hand programmiert werden können. Auch der Befehlsumfang ist nicht mehr zeitgemäß, da mit zunehmender Automatisierung auch Prozessüberwachung, Messzyklen und höhere mathematische Operationen Teil der Steuerung sein müssen und dies nicht mit dem G-Code zu bewerkstelligen ist. Ein Ausweg hierfür wäre das Einbinden höherer Programmiersprachen in ein konventionelles NC-Programm, also eine Erweiterung der Funktionalität und Vereinfachung der Struktur,

jedoch kann auch mit Erweiterungen kein objektorientiertes Format daraus erstellt werden, was für deutlich komplexere NC-Programme nötig wäre. (WECK, 2006)

NC-Programmiermethoden

Für einfache Werkzeugpfade kann ein solches Programm zwar recht schnell manuell erstellt werden, jedoch kann dies bei komplexen Geometrien wie schon erwähnt sehr aufwendig und fehleranfällig werden, da z.B. unübersichtlich viele Einzelpositionen per Hand berechnet und textuell eingegeben werden müssen. Aus diesem Grund wurde schon sehr früh eine ganze Reihe weiterer rechnergestützter NC-Programmierverfahren entwickelt. Neben der Entwicklung höherer Programmiersprachen und grafisch-interaktiver Unterstützung von NC-Systemen wurde hierzu auch das direkte Ableiten von NC-Programmen aus Computer Aided Design (CAD)-Dateien, als Teildisziplin des Rechnergestützten Fertigens (*englisch*: Computer Aided Manufacturings, CAM), entwickelt.

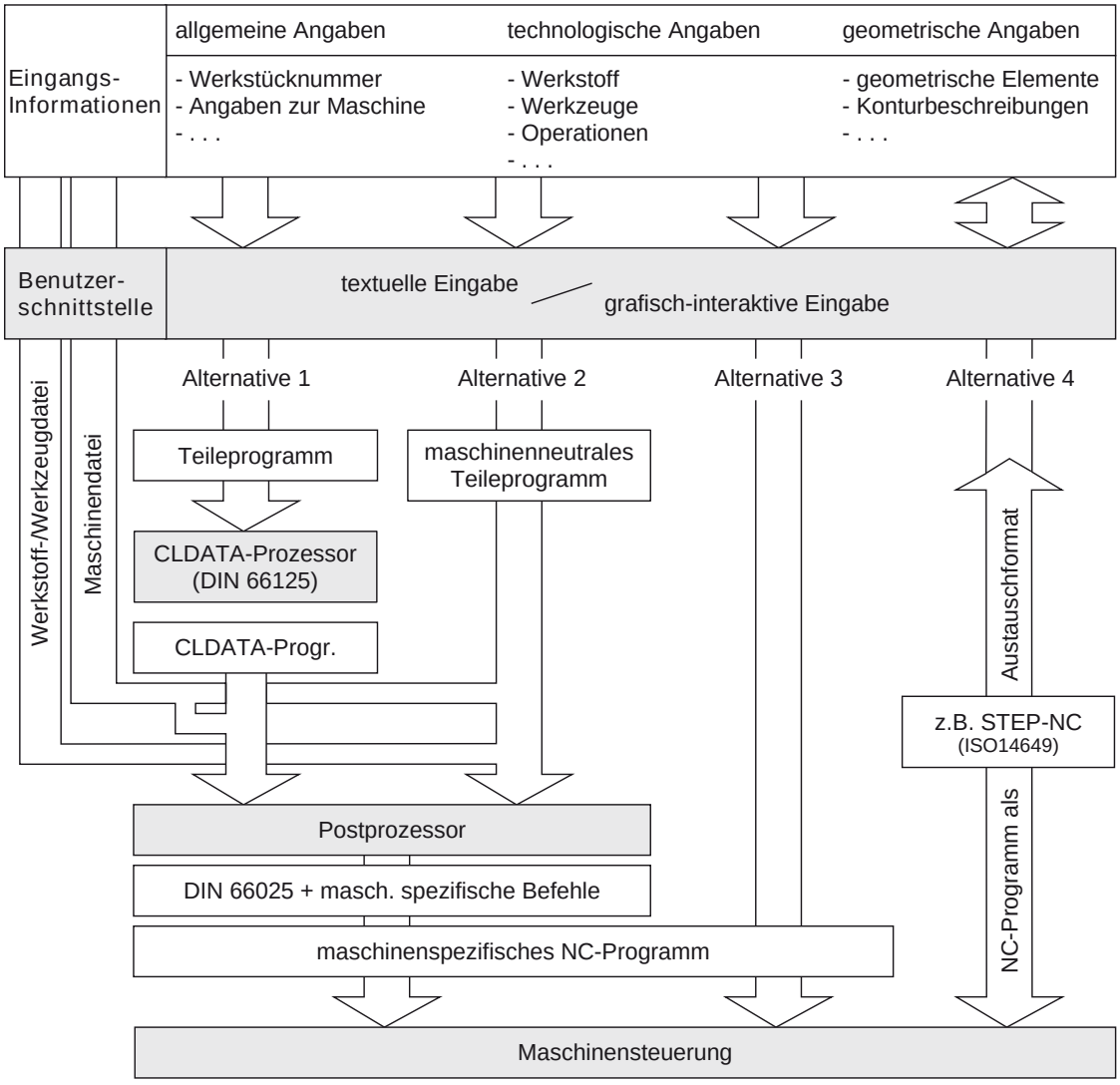


Abbildung 2.26: Alternativen der maschinellen NC-Programmierung. (WECK, 2006)

Abbildung 2.26 zeigt ganz allgemein die Funktionsweise rechnergestützter NC-Programmiermethoden, hier wird in der Regel (Alternative 1 und 2) zunächst auf Basis der Geometriedaten ein maschinenneutrales Teileprogramm erstellt, die dann durch zusätzliche Informationen (Werkstoff-/Werkzeugdatei und Maschinendatei) im Postprozessor zu einem maschinenspezifischen NC-Programm umgewandelt werden. In speziellen Fällen, also bei eigenen Softwarelösungen von den entsprechenden Maschinenherstellern, werden diese Zwischenschritte auch weggelassen und direkt maschinenspezifische Programme erstellt (Alternative 3) (WECK, 2006).

Alternative 4 stellt hierzu eine Modernisierung dar, denn hier wird im Gegensatz zu den anderen Möglichkeiten kein NC-Programm nach DIN 66025/ISO 6983 als Ausgabeformat erzeugt, sondern es werden alle Steuerungsinformationen direkt im eigens hierfür erweiterten Datenformat **STEP-NC** modelliert. Es müssen demnach keine Datentransformationen stattfinden, somit entfällt der Postprozessor-Schritt und die entsprechende Datei kann direkt als Austauschformat für alle beteiligten Disziplinen verwendet werden. Außerdem ist es hiermit direkt möglich, Änderungen auf Fertigungsebene, wie z.B. Optimierungen an der CNC-Maschine oder Geometrieanpassungen, direkt im Modell zu aktualisieren und für die anderen Bereiche verfügbar zu machen.

STEP-NC

STEP-NC (ISO 14649) stellt ein objektorientiertes Datenformat dar, welches durchgängig über alle Bereiche in der Fertigung eingesetzt werden kann (vgl. STEP TOOLS, 2007). Im Wesentlichen ist STEP-NC eine Erweiterung des STEP-Datenformats, wodurch zusätzlich zu semantischen und geometrischen Informationen auch die NC-Steuerung einer Fertigungsmaschine mit diesem Format abgebildet werden kann. Mit diesem Datenformat sind demnach STEP-Geometriedaten direkt in der NC-Steuerung nutzbar, ganz ohne Datenkonvertierung und den damit einhergehenden Fehlerquellen. Im Gegensatz zum NC-Programm nach DIN 66025/ISO 6983 sind die Anweisungen im STEP-NC Datenformat nicht in einzelne Sätze verpackt, sondern an Objekte geknüpft, die über Relationen mit dem entsprechenden Bauteilmerkmal oder sogar dem gesamten Bauteil in Verbindung stehen. Auf diese Weise können höhere Funktionalitäten wie Kollisionsprüfungen, Soll-Ist-Abgleiche sowie adaptive Pfadplanung und vieles mehr realisiert werden und eine durchgängige Datenverfügbarkeit (Feedback) über sämtliche an der Fertigung beteiligten Disziplinen ermöglicht werden (HEHENBERGER, 2020; RODRIGUEZ & ALVARES, 2019).

Abbildung 2.27 zeigt stark vereinfacht die Datenstruktur einer STEP-NC Datei, der Arbeitsschritt (`machining_workingstep`) stellt dabei alle Informationen für die NC-Steuerung bereit, die zur Bearbeitung eines Features (`machining_feature`) notwendig sind. Die dafür möglichen Attribute können dabei zum Teil alternativ eingesetzt werden, z.B. kann entweder direkt ein Werkzeugpfad (`toolpath`) angegeben werden, der genau eingehalten werden soll, oder nur eine generelle Strategie (`strategy`), also eine Art Bewegungsmuster, um eine komplette Fläche anzusteuern (WECK, 2006).

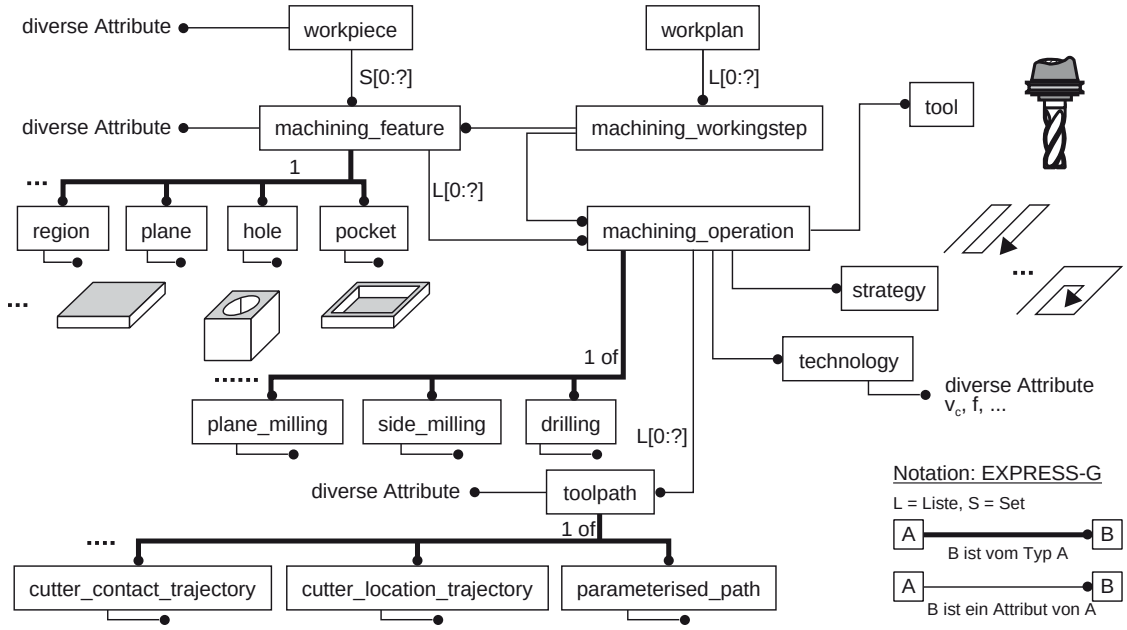


Abbildung 2.27: Auszug aus dem STEP-NC Datenmodell. (WECK, 2006)

Aktuell ist das STEP-NC Datenformat hauptsächlich auf Zerspanungsprozesse ausgelegt, allerdings gibt es Bemühungen, auch additive Fertigungsprozesse im Datenmodell zu definieren, um einen derartigen Fertigungsablauf darstellen zu können. RODRIGUEZ und ALVARES (2019) haben hierzu fünf Hauptaktivitäten identifiziert, die zur Implementierung von STEP-NC in additiver Fertigung notwendig sind und damit erfolgreich ein STEP-NC Programm erstellt, mit dem beispielhaft ein Druckvorgang simuliert und auch durchgeführt werden konnte.

Trotz der vielen Vorteile, die das Datenformat STEP-NC mit sich bringt, hat sich der seit 2003 eingeführte Standard in der Industrie noch nicht durchsetzen können, da die CNC-Steuereinheiten aktuell verwendeter Maschinen immer noch grundlegend auf dem Standard DIN 66025/ISO 6983 basieren und erst eine entsprechende Nachfrage bei den Anwendern aufkommen muss, damit Steuerungshersteller entsprechend umrüsten (WECK, 2006). Aber besonders in Hinblick auf die momentan rasante Entwicklung im Bereich der additiven Fertigung und dem immer größer werdenden Grad an Automatisierung wird dieser Standard sicherlich rasch an Bedeutung gewinnen.

2.4.2 Robotersteuerung (RC)

Genau wie bei CNC-Maschinen gibt es ganz verschiedene Bauarten von Robotern, wobei mit etwa 62% hauptsächlich Knickarmroboter verwendet werden, so auch im Zuge dieser Masterarbeit. Knickarmroboter gehören neben SCARA- und Portalrobotern zu den sogenannten seriellen Industrierobotern, bei denen die Antriebe in einer kinematischen Kette miteinander gekoppelt sind, also seriell zueinander geschaltet sind. Daneben gibt es noch die sogenannten parallelen Roboter, darunter Delta- und Seilroboter, die sich dadurch auszeichnen, dass alle Antriebe zueinander parallel geschaltet sind. Die zuletzt

genannten Seilroboter können durch ihren fachwerkartigen Aufbau sehr leicht konzipiert werden und dennoch sehr große Dimensionen und Traglasten erzielen. Die Eignung für den Einsatz solcher Robotersysteme für den 3D-Druck von Gebäuden wurde hierzu in einer eigenen Forschungsarbeit untersucht (vgl. BOSSCHER et al., 2007).

Wie schon zuvor beschrieben ist die Steuerung von CNC-Maschinen und Roboterarme auf den ersten Blick sehr ähnlich, dennoch gibt es hier einige Unterschiede. Die nach menschlichem Vorbild konzipierten Knickarmroboter (vgl. Abb. 2.22e) z.B. werden für viele unterschiedliche Tätigkeitsbereiche eingesetzt, die komplexe, aber automatisierbare Bewegungsabläufe beinhalten. Im Gegensatz zu CNC-Maschinen, die in der Regel für spezielle Aufgaben konzipiert werden, können Roboterarme deutlich flexibler eingesetzt werden, indem einfach das Werkzeug ausgetauscht wird und mittels kleinerer Anpassungen an dem Steuerungsprogramm der Bewegungsablauf geändert wird. Diese Flexibilität hat aber auch den Nachteil, dass im Vergleich zu CNC-Maschinen ihre Positionsgenauigkeit sowie der Widerstand gegen dynamische Störeffekte deutlich geringer ist. Aus diesem Grund können Robotersysteme nicht in allen Aufgabenbereichen der industriellen Fertigung eingesetzt werden, also sobald Positionen hoch präzise angefahren werden müssen. (POTT & DIETZ, 2019)

Bei der Steuerung von Robotersystemen bezieht man sich auf einen speziellen Punkt, den Werkzeugmittelpunkt (*englisch*: Tool Center Point, TCP), also auf den Ort, an dem Werkzeugarbeiten ausgeführt werden sollen. Für Roboter mit mindestens 4 Freiheitsgraden, z.B. Knickarmroboter, reicht allerdings ein Punkt zur Bewegungsbeschreibung nicht aus, da das Werkzeug auch rotiert werden kann. Um die räumliche Lage des Werkzeugs zu beschreiben, wird daher die Position und Orientierung **eines** Koordinatensystems im Werkzeugmittelpunkt verwendet. Daher spricht man beim Begriff TCP im Prinzip weniger von einem Punkt, sondern eher von einem definierten Koordinatensystem an diesem Punkt. Die Kombination aus Position und Orientierung dieses Koordinatensystems wird hierbei auch **Pose** des Roboters genannt.

Bei einem Knickarmroboter mit 6 Freiheitsgraden z.B. wird die Pose entweder über die kartesischen Koordinaten und die Rotationswinkel entlang der drei Hauptachsen dargestellt oder über die 6 Gelenkstellungen. Von diesen beiden Darstellungen ist die Gelenkstellung für die Robotersteuerung die wichtigere, da hiermit tatsächlich die Gelenkeinstellungen beschrieben werden (vgl. Abb. 2.28, hier Drehung um z_i-Achsen möglich), für den Benutzer ist sie jedoch nicht besonders aussagekräftig. Diese Darstellungen können jedoch in die jeweils andere umgerechnet werden, da alle Gelenke kinematisch miteinander verbunden sind. Wird aus Gelenkstellungen die kartesische Lage berechnet, spricht man von **Vorwärtskinematik**, umgekehrt, also von kartesischen Koordinaten in die Gelenkstellungen, spricht man von **Rückwärtskinematik**. Eine Eigenheit von Knickarmrobotern ist es, dass in der Regel eine Pose durch mehrere Armstellungen erzielt werden kann. Diese Tatsache kann unter anderem bei der Rückwärtskinematik zu Problemen führen, da die entsprechenden Gleichungen in den meisten Fällen über keine eindeutige Lösung verfügen und eine Lösung gewählt werden muss (vgl. Abb. 2.30).

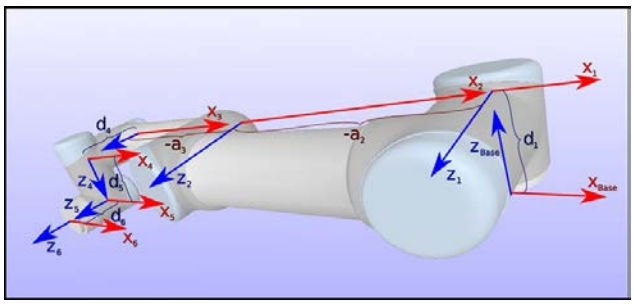


Abbildung 2.28: Gelenkkoordinatensysteme des UR10e Roboters. (UR, 2020c)

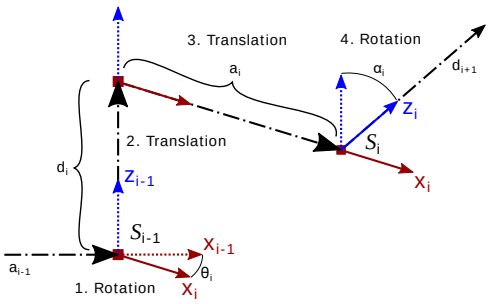


Abbildung 2.29: Koordinatensystem Transformation nach Denavit und Hartenberg. (nach SIEGERT & BOCIONEK, 2013)

Vorwärtskinematik

Ein serieller Roboter ist wie zuvor beschrieben dadurch charakterisiert, dass mehrere Achsen (Arme) in einer kinematischen Kette über Gelenke starr miteinander verbunden sind. Ordnet man diesen Armen jeweils in ihrem Gelenk eindeutig ein Koordinatensystem zu, so kann man die Gelenkbewegungen durch die Verschiebung und/oder Rotation des entsprechenden Koordinatensystems in das darauffolgende beschreiben. Eine solche Verschiebung von einem Koordinatensystem in das andere wird mit **homogenen Transformationen**, insgesamt also über eine 4x4 Transformationsmatrix realisiert. Am anschaulichsten (einfache Transformationsmatrizen) ist dies zu zeigen, wenn die Koordinatensysteme nach DENAVIT und HARTENBERG (1955) festgelegt werden (vgl. SIEGERT & BOCIONEK, 2013, S. 55f.). Für den UR10e Roboter ist diese Koordinatensystemanordnung in Abb. 2.28 dargestellt.

Unter dieser Voraussetzung lassen sich zwei aufeinander folgende Koordinatensysteme ($\mathcal{S}_{i-1}$ und $\mathcal{S}_i$) über einfache Translationen und elementare Rotationen durchführen. Wie in Abb. 2.29 dargestellt kann das Koordinatensystem $\mathcal{S}_{i-1}$ über folgende Schritte in $\mathcal{S}_i$ überführt werden (SIEGERT & BOCIONEK, 2013):

1. Rotation um die z-Achse: $\boldsymbol{Rot}(z, \theta_i)$
2. Translation in z-Richtung: $\boldsymbol{Trans}(0, 0, d_i)$
3. Translation in x-Richtung: $\boldsymbol{Trans}(a_i, 0, 0)$
4. Rotation um die x-Achse: $\boldsymbol{Rot}(x, \alpha_i)$

Durch aufmultiplizieren aller vier Matrizen wird dann eine Gesamttransformationsmatrix ${}^{i-1}\boldsymbol{T}_i$ für das i-te Gelenk erzeugt (*Sinus* und *Cosinus* sind hier mit s bzw. c abgekürzt):

$$
{}^{i-1}\boldsymbol{T}_i = \underbrace{\begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{Rot}(z,\theta_i)} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{Trans}(0,0,d_i)} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{Trans}(a_i,0,0)} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{Rot}(x,\alpha_i)}
$$

Ausmultipliziert ergibt sich damit:

$$
{}^{i-1}\boldsymbol{T}_i = \begin{bmatrix} c\theta_i & -c\alpha_i \cdot s\theta_i & s\alpha_i \cdot s\theta_i & a_i \cdot c\theta_i \\ s\theta_i & c\alpha_i \cdot c\theta_i & -s\alpha_i \cdot c\theta_i & a_i \cdot s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}
$$

Zur Bestimmung der Lage des TCP müssen ausgehend vom Basisgelenk bis zum Werkzeug alle Transformationen der Reihe nach ausgeführt werden. Das bedeutet, dass die Transformationsmatrizen aller Gelenke aufmultipliziert und die entsprechenden Winkel- und Längenmaße eingesetzt werden müssen. Für einen Knickarmroboter mit 6 Gelenken ergibt sich damit folgender Ausdruck:

$$
{}^{0}\boldsymbol{T}_6 = {}^{0}\boldsymbol{T}_1 \cdot {}^{1}\boldsymbol{T}_2 \cdot {}^{2}\boldsymbol{T}_3 \cdot {}^{3}\boldsymbol{T}_4 \cdot {}^{4}\boldsymbol{T}_5 \cdot {}^{5}\boldsymbol{T}_6
$$

Da hier 6 Terme der Form von Gleichung (2.1) zusammen multipliziert werden ergeben sich für die einzelnen Elemente sehr unhandliche Gleichungen, die an dieser Stelle nur anhand des ersten Elements mit schon eingerechneten α-Werten für einen UR Roboter näher betrachtet werden. Auf jeden Fall ergibt sich durch die Multiplikation der Transformationsmatrizen wieder eine 4x4 Transformationsmatrix folgender Form:

$$
{}^{0}\boldsymbol{T}_6 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}
$$

$$
\begin{aligned}
n_x = & \cos(\theta_6) \cdot (\sin(\theta_1) \cdot \sin(\theta_5) + \\
& \frac{[\cos(\theta_1) \cdot \cos(\theta_2 + \theta_3 + \theta_4) - \sin(\theta_1) \cdot \sin(\theta_2 + \theta_3 + \theta_4)] \cdot \cos(\theta_5)}{2,0} + \\
& \frac{[\cos(\theta_1) \cdot \cos(\theta_2 + \theta_3 + \theta_4) + \sin(\theta_1) \cdot \sin(\theta_2 + \theta_3 + \theta_4)] \cdot \cos(\theta_5)}{2,0} - \\
& \frac{[\sin(\theta_1) \cdot \cos(\theta_2 + \theta_3 + \theta_4) + \cos(\theta_1) \cdot \sin(\theta_2 + \theta_3 + \theta_4)] \cdot \sin(\theta_6)}{2,0} - \\
& \frac{[\sin(\theta_1) \cdot \cos(\theta_2 + \theta_3 + \theta_4) - \cos(\theta_1) \cdot \sin(\theta_2 + \theta_3 + \theta_4)] \cdot \sin(\theta_6)}{2,0}
\end{aligned} \tag{2.3}
$$

Aus dieser Matrix lassen sich dann alle Positions- und Orientierungswerte direkt ablesen. Die Vektoren $\mathbf{n}$, $\mathbf{s}$ und $\mathbf{a}$ Spannen hierzu ein Koordinatensystem mit dem TCP an der Position $\mathbf{p}$ als Ursprung auf.

Setzt man in die Gleichungen für die einzelnen Matrixelemente die Parameter a, d, α und θ ein, so kann man also für alle Armstellungen die TCP-Lage errechnen. Von diesen 4 Parametern sind die ersten drei (a, d, α) roboterspezifisch, wovon a und d die axialen Abstände zwischen zwei Gelenken und α die Verdrehung zwischen zwei Gelenken darstellen. Im Prinzip beschreiben a, d und α die Bauform des Roboters, also die Armlängen und

Tabelle 2.1: Werte für den UR10e Roboter

Gelenk	θ [rad] Bereich	a [m]	d [m]	α [rad]
Base	$\pm 2\pi$	0	0,1807	$\pi/2$
Shoulder	$\pm 2\pi$	$-0,6127$	0	0
Elbow	$\pm 2\pi$	$-0,57155$	0	0
Wrist 1	$\pm 2\pi$	0	0,17415	$\pi/2$
Wrist 2	$\pm 2\pi$	0	0,11985	$-\pi/2$
Wrist 3	$\pm 2\pi$	0	0,11655	0

die Anordnung der Gelenke. Der Parameter θ_i beschreibt hier die Gelenkeinstellung des entsprechenden Gelenks und kann je nach Roboter aus einem unterschiedlichen Bereich gewählt werden. Für den in dieser Arbeit zum Einsatz kommende UR10e Roboter sind die Werte in Tabelle 2.1 gegeben. Die θ-Wertebereiche stellen hier jedoch nur die maximalen Gelenkeinstellungen des Roboters dar und können durch äußere Barrieren, wie z.B. Wände, feststehende Objekte oder einprogrammierte Grenzen, weiter eingeschränkt werden. Für das Verständnis der oben aufgeführten Rechnung wird von Universal Robots auf ihrer Support-Webseite[1] ein sehr anschauliches Video verlinkt und eine Excel-Tabelle bereitgestellt, mit der die Vorwärtskinematik für alle UR-Robotermodelle berechnet werden kann.

Rückwärtskinematik

Um Roboterbewegungen einzuprogrammieren, gibt man in der Regel eine diskretisierte Raumkurve vor, die der Roboter mit dem TCP in einer gewissen Orientierung abfahren soll. Jedoch wird der Roboter über seine Gelenkstellungen gesteuert, bei Knickarmrobotern also über die Einstellung der zuvor genannten Winkel θ_i (jeweils ein Winkel pro Gelenk des Roboters). Dies macht also eine Rückrechnung (Rückwärtskinematik) von kartesischen Koordinaten in Gelenkstellungen, also eine Umkehrung der Vorwärtskinematik, unabdingbar. Bei translatorischen Systemen, wie z.B. Portal- oder Gantry-Roboter, oder parallelen Systemen ist dies recht einfach zu bewerkstelligen, hier sind Umkehrfunktionen einfach zu bilden und das Gleichungssystem in der Regel eindeutig lösbar. Bei Rotationssystemen, wie dem Knickarmroboter, ist diese Berechnung allerdings sehr aufwendig und nicht notwendigerweise (eindeutig) lösbar, da hierfür die Lösung eines nichtlinearen trigonometrischen Gleichungssystems erfordert wird, also die Umkehrung der Gleichung (2.2).

In jedem Fall ist es möglich, auf numerischem Weg die inverse Kinematik zu lösen, dies ist aber mit hohem Rechenaufwand verbunden und kostet Zeit (mehrere Millisekunden). Vorteilhafter ist hier eine analytische Lösung, die aber nicht für alle Robotersysteme möglich ist. Es gilt hierbei, je komplexer ein Robotersystem aufgebaut ist, umso schwieriger ist die Formulierung der analytischen Lösung. Eines der einfachsten Robotersysteme sind

[1] https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/

Ortho-parallele Manipulatoren mit sphärischer Wrist, hier ist das erste Gelenk senkrecht zu den zwei nächsten, Gelenk 2 und 3 parallel zueinander und die letzten drei Achsen schneiden sich in einem Punkt. Diese Bauart ist besonders beliebt und die meisten in der Industrie eingesetzten Roboter sind von diesem Typ (BRANDSTÖTTER et al., 2014). Hier ist das Aufstellen der analytischen Lösung recht leicht möglich, da das Problem in zwei Teilprobleme zerlegt werden kann, die Positionierung und die Orientierung des Endeffektors (BRANDSTÖTTER et al., 2014). Der UR10 Roboter ist allerdings etwas komplizierter, da bei diesem Roboter keine sphärische Wrist verbaut wurde (die letzten drei Achsen schneiden sich nicht in einem Punkt), wodurch das Aufstellen der analytischen Lösung etwas schwieriger ist (vgl. HAWKINS, 2013; KEBRIA et al., 2016).

Aber egal auf welche Weise das Gleichungssystem gelöst wird, bei einem Knickarmroboter kann es, sofern das System überhaupt eine Lösung hat, häufig mehr als eine Lösung geben. Dies ist einfach damit zu begründen, dass eine Pose des TCP durch verschiedene Armstellungen des Roboters realisiert werden kann. Im Gleichungssystem der Vorwärtskinematik (vgl. Teilgleichung 2.3) wird diese Tatsache durch die trigonometrischen Funktionen sichtbar, deren Umkehrfunktionen (Rückwärtskinematik) keine eindeutige Lösung geben werden. Zwar bedingen sich die Lösungen für die 6 Gelenkwinkel zum Teil gegenseitig, aber dennoch gibt es für bestimmte Konfigurationen bis zu 8 verschiedene Möglichkeiten (vgl. Abb. 2.30).

Ein weiteres Problem bei der Rückwärtskinematik ist, dass in bestimmten TCP-Posen zwei Achsen des Roboters kollinear liegen und somit bestimmte Bewegungen nicht eindeutig einer Achse zugeordnet werden können. Es gibt in diesen Situationen unendlich viele Möglichkeiten, die Bewegung durchzuführen, wobei manche davon gegenseitige Achsbewegungen mit unendlicher Geschwindigkeit verlangen würden. Bei der Programmierung eines solchen Roboters kann es also unter anderem aus diesem Grund zu unerwartetem Verhalten kommen. In Abb. 2.31 werden diese kritischen Armstellungen visualisiert. Links im Bild ist die Wrist-Singularität dargestellt, bei der durch Parallellage der dargestellten Achsen eine gegenseitige Verdrehung dieser Achsen möglich ist, ohne die Position des TCP zu ändern. In der Mitte ist die Elbow-Singularität dargestellt, bei der in der gezeigten Stellung ein Wechsel zwischen nach unten gebeugten (Elbow-Down)

Pattern	Shoulder	Elbow	Wrist	Figure1 (Tool Down)	Figure1 (Tool UP)	Pattern	Shoulder	Elbow	Wrist	Figure1 (Tool Down)	Figure1 (Tool UP)
1	Left Side	Down	Tool Down: Outer Tool UP: Inner			5	Right Side	UP	Tool Down: Inner Tool UP: Outer		
2	Left Side	Down	Tool Down: Inner Tool UP: Outer			6	Right Side	UP	Tool Down: Outer Tool UP: Inner		
3	Left Side	UP	Tool Down: Outer Tool UP: Inner			7	Right Side	Down	Tool Down: Inner Tool UP: Outer		
4	Left Side	UP	Tool Down: Inner Tool UP: Outer			8	Right Side	Down	Tool Down: Outer Tool UP: Inner		

Abbildung 2.30: Verschiedene Gelenkstellungen für dieselbe TCP-Lage. (UR, 2020a)

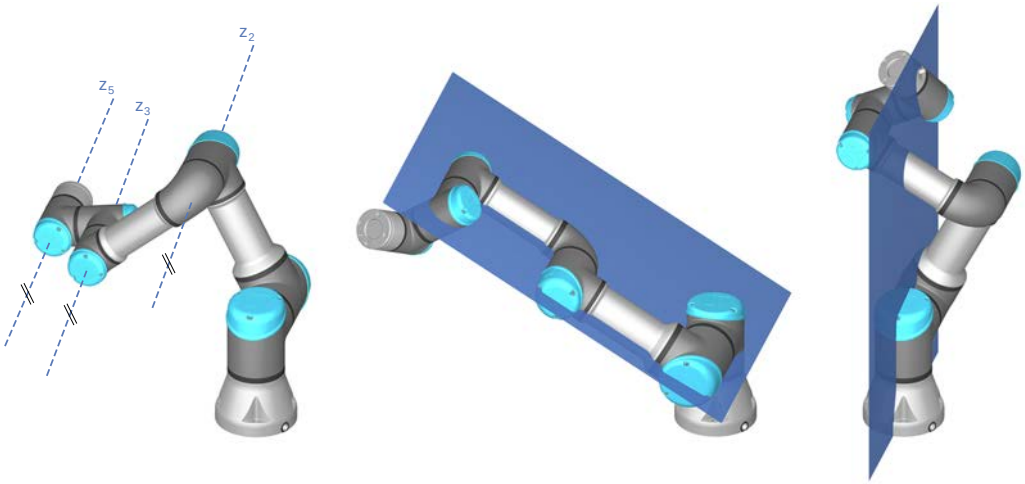


Abbildung 2.31: Verschiedene Singularitätstypen bei einem UR-Roboter: Wrist- (links), Elbow- (Mitte) und Shoulder-Singularität (rechts). (ILIAN, BONEV, 2019)

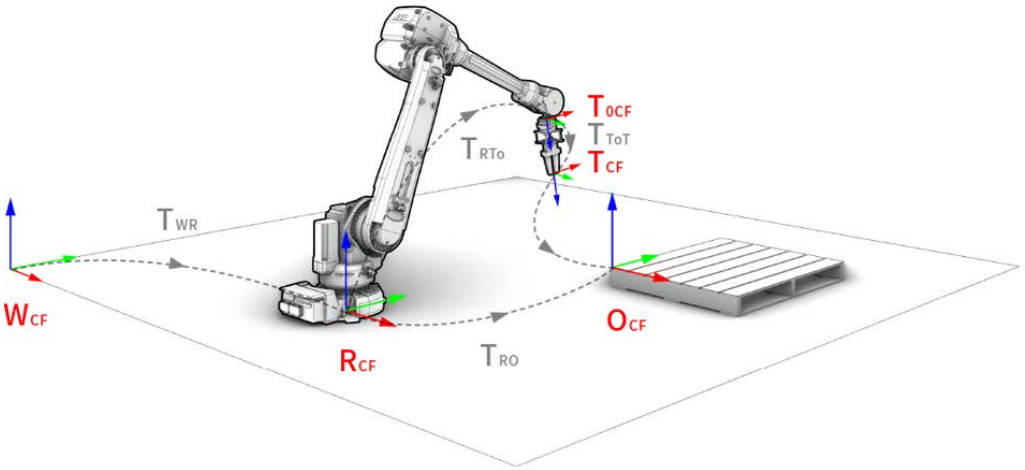


Abbildung 2.32: Koordinatensystemkonvention eines Robotersetups nach GRAMAZIO KOHLER (2020)

und nach oben gebeugten (Elbow-Up) Arm durchlaufen wird. Genau bei der gezeigten Lage ist theoretisch eine unendlich hohe Geschwindigkeit erforderlich, um eine ruckfreie Bewegung zischen den Down- und Up-Stellungen zu ermöglichen. Rechts ist schließlich die Shoulder-Singularität dargestellt, bei der für die Basisgelenkstellung 2 Winkel möglich sind (wird um 180° gedreht, liegt der TCP wieder in der gleichen Pose).

Abschließend zur Kinematik ist noch zu erwähnen, dass bisher nur roboterinterne Koordinatensysteme behandelt wurden. Jedoch soll ein Roboter Einfluss auf seine Umgebung nehmen und muss daher auf sinnvolle Weise mit dieser in Relation gebracht werden. Um eine bessere Kooperation verschiedener Programmierer und Anwender zu gewährleisten, sollte hier eine einheitliche Vorschrift gelten. Die Entwickler von *Compas Fab*[2] definieren hierzu verschiedene Koordinatensysteme, das Weltkoordinatensystem ($\boldsymbol{W}_{CF}$) in dem

[2]Ermöglicht die Planung und Ausführung von Fertigungsprozessen mit Robotern direkt über parametrische CAD-Werkzeuge durch Bereitstellung von Schnittstellen zu Softwarepaketen, wie z.B. dem Robot Operating System (ROS) Framework.

sowohl der Roboter mit seinem eigenen Basiskoordinatensystem ($\boldsymbol{R}_{CF}$), als auch das zu bearbeitende Objekt mit einem Objektkoordinatensystem ($\boldsymbol{O}_{CF}$) verortet wird (GRAMAZIO KOHLER, 2020). Zudem wird dem Werkzeug ein eigenes Koordinatensystem zugeordnet, da dieses aber in der Position abhängig von der Größe des Werkzeugs ist, wird dem Flansch des Roboters ein System zugeordnet ($\boldsymbol{T}_{0CF}$) relativ dazu der wirkliche TCP über das Werkzeugkoordinatensystem ($\boldsymbol{T}_{CF}$) definiert. Diese Koordinatensysteme und die zugehörigen Transformationen, die zum Wechsel der Referenz gebraucht werden, sind in Abb. 2.32 dargestellt.

Roboterprogrammierung

Die komplette Kinematik eines Roboters übernimmt dessen Steuereinheit und sind für den Programmierer nicht direkt von Bedeutung. Wie schon erwähnt ist es jedoch sinnvoll, über bestimmte Eigenheiten der im Hintergrund ablaufenden Berechnungen Bescheid zu wissen, damit Singularitäten und sonstiges unerwartetes Verhalten nicht auftritt. Für die Programmierung von Robotern haben sich deshalb verschiedene Verfahren entwickelt, die man in Online und Offline Verfahren einteilt.

Als physisches Werkzeug für die Online Programmierung wird das **Handgerät** des Roboters verwendet. Dies ist ein kleiner, tragbarer und direkt am Roboter angeschlossener Steuercomputer, über dessen Software der Roboter direkt gesteuert werden kann. Für Notfälle ist an diesem Gerät auch ein Notausschalter angebracht, um schnell manuell den Roboter zu stoppen, z.B. wenn fehlerhaft programmiert wurde und sich der Roboter unvorhergesehen bewegt. Zur Eingabe von Befehlen waren am Handgerät früher Tasten und ein kleines Display angebracht, bei modernen Geräten ist hierfür nunmehr ein Touchdisplay vorhanden. Für die Offline Programmierung wird einfach ein normaler PC verwendet. Das damit erstellte Programm kann entweder mit einem USB-Laufwerk oder per Netzwerk an den Roboter übertragen werden.

Bei den **Online Verfahren** wird direkt am Roboter graphisch-interaktiv das Programm erstellt, das später eigenständig abgefahren werden soll. Hierzu zählt zum einen das **Teach-In** Verfahren, bei dem der Roboter mittels eines Steuerhebels am Handgerät in die gewünschten Posen gebracht wird, diese nacheinander abgespeichert werden und mit zusätzlichen Bearbeitungsanweisungen für das Werkzeug zu einem vollständigen Programm verarbeitet werden. Ein weiteres Verfahren ist das **Playback** Verfahren, bei dem der Programmierer den Roboter durch die gewünschte Bewegung führt, die der Roboter später eigenständig fahren soll. Eine manuelle Eingabe des Roboterprogramms direkt in die Handgerät-Steuerung ist auch möglich, wird aber aufgrund der Fehleranfälligkeit in der Regel nicht mehr so angewendet.

Mit den **Offline Verfahren** wird das Programm abseits vom Roboter erstellt und in der Regel über Simulationswerkzeuge vor dem Einsatz auf Fehler geprüft. Auf diese Weise kann der Roboter während der Programmierung auch noch für andere Zwecke genutzt werden und das Programm kann anhand von Simulationsergebnissen weiter optimiert werden. Offline wird entweder rein textuell mit Programmiersprachen programmiert, die auf

Roboterproblematik zugeschnitten sind, oder wie auch bei der CNC-Steuerung unterstützt durch CAD/CAM-Software.

Genauso vielfältig wie die Bauart von Robotern sind hier jedoch die von den verschiedenen Roboterherstellern eigens entwickelten Programmiersprachen. Zwar sind die Befehlssätze der einzelnen Sprachen durchaus vergleichbar, aber dennoch untereinander nicht kompatibel und ganz im Gegensatz zu Werkzeugmaschinen (DIN 66025/ISO 6983) nicht standardisiert. Beispielsweise hat die Firma Kuka die *Kuka Robot Language* (KRL) entwickelt, Stäubli die Sprache *VAL3* und Universal Robots, deren Roboter UR10e im Zuge dieser Masterarbeit verwendet wird, *URscript*. Das Programmieren mit diesen Roboter-Sprachen ist vergleichbar mit dem höherer Programmiersprachen, es können im Gegensatz zu dem CNC-Steuerungscode Schleifen und Verzweigungen verwendet und Variablen definiert werden.

Eine weitere Ähnlichkeit zu der CNC-Steuerung sind die möglichen Bewegungsarten bzw. die Bahnführungen. Auch hier gibt es zum einen die Punktsteuerung und die Bahnsteuerung. Bei Ersterem wird nur jeweils die nächste anzusteuernde Pose angegeben und der Roboter wählt selbstständig einen Weg dorthin. Auf dem kinematisch schnellstmöglichen Weg, der in der Regel gewählt wird, werden alle Achsen schnellstmöglicher Zeit auf die Zielposition bewegt, was nicht zwangsläufig geradlinig geschehen muss. Sind die anzusteuernden Punkte weit voneinander entfernt, kann es daher sinnvoll sein, mit der Bahnsteuerung auch den Weg dazwischen festzulegen, um z.B. Kollisionen durch eine unkontrollierte Bahnführung zu vermeiden.

Bei der Bahnsteuerung kann eine beliebige Bahn vorgegeben werden. Diese wird in kurzen Abständen in sogenannte Interpolationspunkte eingeteilt, für die jeweils eine Rückwärtstransformation ausgeführt wird. Zwischen diesen Interpolationspunkten wird dann nochmal feininterpoliert, sodass sich der TCP ziemlich genau auf dieser Bahn bewegt. Hauptsächlich werden hier aber entweder lineare Bewegungen oder Kreisbewegungen verwendet (vgl. Abb. 2.25).

Robot Operating System (ROS)

Eine weitere Möglichkeit um Robotersteuerungsprogramme zu realisieren, sind sogenannte „robot frameworks“, wie z.B. *Player*, *YARP*, *Orocos*, *CARMEN*, *Orca*, *MOOS*, das *Microsoft Robotics Studio* und Robot Operating System (ROS). Alle diese Frameworks liefern eine Sammlung von Software-Werkzeugen, -Bibliotheken und Konventionen, mit denen bestimmte Roboter gesteuert werden können. ROS ist hierbei besonders hervorzuheben, da zum einen der gesamte Umfang der Tools und Bibliotheken frei verfügbar ist (open source) und zum anderen besonders auf die Wiederverwertbarkeit von Code wert gelegt wurde. Das Framework ist so einfach und modular gestaltet, dass es auch in Kombination mit anderen Frameworks zu Einsatz kommen kann und eigene Pakete und „Stacks“, die auf den persönlichen Nutzen zugeschnitten sind, erstellt werden können.

Mit ROS, sowie den anderen Frameworks, ist es möglich, Programme zu entwickeln, die zuverlässig in verschiedenen Szenarien eingesetzt werden können, unabhängig vom verwendeten Robotersystem. Es werden beim „Abspielen“ des Steuerungsprogramms lediglich die Roboterinformationen über einen sogenannten Treiber benötigt, damit der Code mit dem entsprechenden Roboter ausgeführt werden kann. Auf diese Weise ist es möglich, mit ROS verschiedene Programmiersprachen einzusetzen. Die entsprechenden Werkzeuge und Bibliotheken wurden hierzu unter anderem für Sprachen, wie z.B. C++ und Python, implementiert. Unter der Verwendung von höheren Programmiersprachen ist es damit auch sehr leicht möglich, gleichzeitig zur Robotersteuerung im selben Steuerungsprogramm andere Maschinen, wie z.B. den AM-Druckkopf oder Scannersysteme, zu steuern.

2.5 Rechnergestützte Produktion

Durch immer komplexer werdende Produkte und Bauprojekte steigt auch der entsprechende Planungs- und Produktionsaufwand immer weiter, sodass Maßnahmen getroffen werden müssen, um diesen Mehraufwand zu kompensieren (HEHENBERGER, 2020). Im Maschinenwesen wird deshalb im Zuge des „Industry 4.0“-Konzepts immer mehr die gegenseitige Integration von Fertigung und Informations- und Kommunikationstechniken vorangetrieben (LEE et al., 2015). Eine sehr gute Darstellung, die das Zusammenspiel

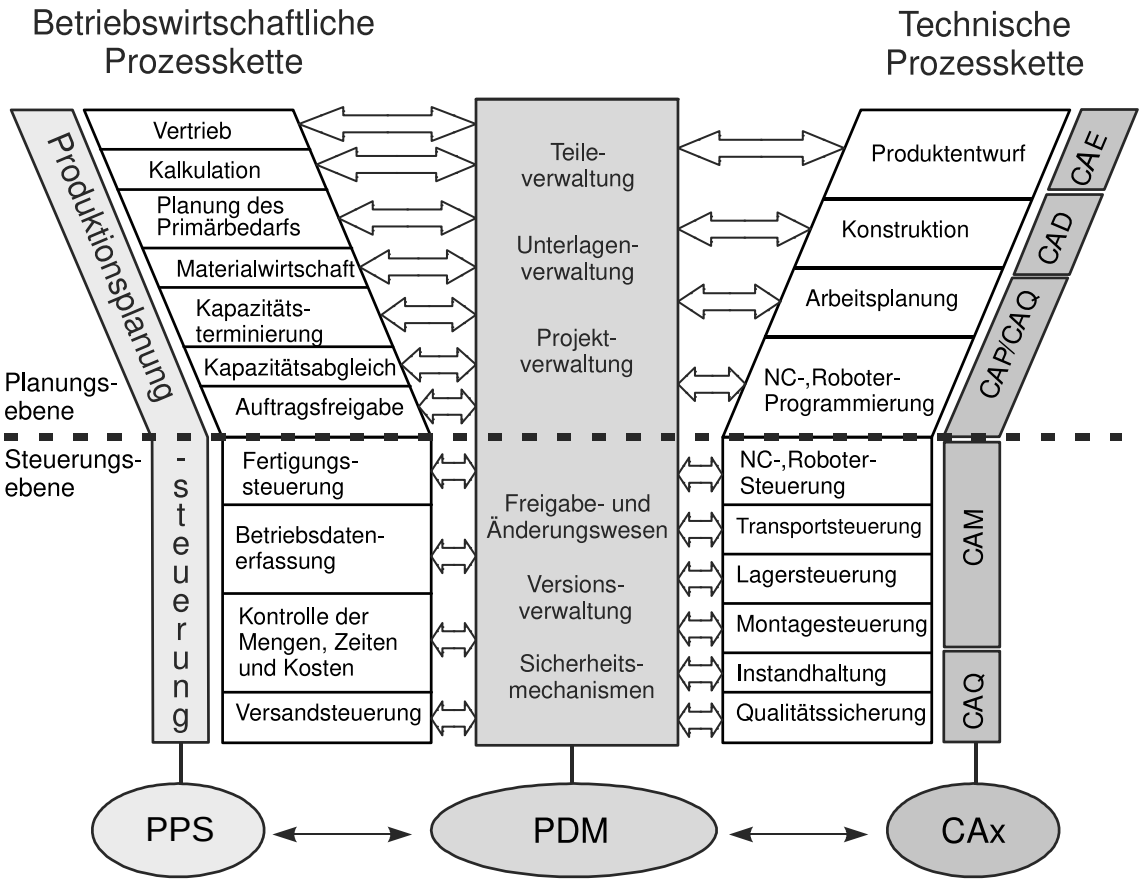


Abbildung 2.33: Y-CIM-Modell (nach HEHENBERGER, 2020; SCHEER, 2013)

aller beteiligten Bereiche und die nötige Kommunikation visualisiert, ist hierbei das Y-CIM-Modell von August-Wilhelm Scheer dar (SCHEER, 2013). Der Begriff Rechnergestützte Fertigung (*englisch*: Computer Integrated Management, CIM) stellt hierbei einen Sammelbegriff für die verschiedenen Tätigkeitsbereiche in einem produzierenden Unternehmen dar, die durch computergestützte Methoden unterstützt werden sollen (vgl. Abb. 2.33).

Auf eine ähnliche Weise werden durch BIM vergleichbare Tätigkeitsbereiche vernetzt, mit dem Ziel, deren Effizienz zu steigern und für eine möglichst verschwendungsfreie Planung, Ausführung und Instandhaltung zu sorgen (vgl. Abschnitt 2.1). Jedoch ist die automatisierte Fertigung für diesen Wirtschaftssektor noch wenig entwickelt, ganz im Gegensatz zum Maschinenwesen. Aus diesem Grund und mangels gewisser Voraussetzungen, die BIM derzeit noch nicht erfüllt, gibt es immer noch wenige Ansätze, um computergestützte Fertigung mit in den BIM-Workflow zu integrieren. Auch wenn das hier gezeigte Y-CIM-Modell (Abb. 2.33) nicht unbedingt allgemein auf das Bauwesen angewendet werden kann, lassen sich damit dennoch ein paar Bereiche visualisieren, die in BIM noch nicht integriert sind, besonders auf der Seite der Steuerungsebene.

Manuelle Arbeitsabläufe lassen sich in einem 4D BIM Modell bereits sehr gut modellieren, hierbei ist es möglich, nach der 3D Planung eine Termin- und Ressourcenplanung aufzustellen und diese bei der Bauausführung mit einer Ist-Datenerfassung (Baufortschritt) zu kontrollieren. Z.B. lässt sich für den Bau von Stahlbetonwänden ein genauer Arbeitsplan erstellen, mit zeitlicher Reihenfolge, wann die Bewehrung verlegt, die Schalung verbaut und der Beton gegossen werden soll. Für digitale Fertigungsmethoden, wie AM, wurden in BIM allerdings noch keine derartigen Mechanismen implementiert, obwohl hier eine weitaus genauere bzw. lückenlose Prozessbeschreibung notwendig ist, da für den automatischen Fertigungsprozess jeder einzelne Teilschritt beschrieben sein muss.

Während der Modellierung der Prozessbeschreibung muss zudem eine gute Abstimmung aller beteiligten Material-, Prozess- und Maschinenparameter erfolgen, um eine erfolgreiche Fertigung und gute Qualität, sowie Leistungsfähigkeit des Bauteils gewährleistet werden kann. Jede AM-Methode kann dabei einen ganz unterschiedlichen Satz an Parametern erfordern und diese können auf ganz unterschiedliche Art und Weise miteinander interagieren. Daher ist es in der Regel notwendig, die Parameteranpassungen iterativ durchzuführen und entsprechende Zwischenergebnisse mithilfe von Simulationen zu evaluieren. Im Rahmen der Modellierung der Fertigungsinformationen sind also neben den erwähnten Druckpfaden (Liniengeometrien und Prozessinformationen) zudem auch Informationen über die Materialverteilung (Volumenmodell und Materialkennwerte) im Raum notwendig (PAOLINI et al., 2019). Sind alle Prozesse beschrieben und die Parameter abgestimmt, muss abschließend noch eine Übersetzung in maschinenverständlichen Code stattfinden, wie in Abschnitt 2.4 beschrieben.

In dieser Arbeit wird hierzu der Begriff des **Fabrication Information Modellings (FIMs)** eingeführt. Als eine Herangehensweise zur Modellierung von Fertigungsinformationen für AM und der entsprechenden Erzeugung von Fabrication Information Models (FIM), stellt dies eine Schnittstelle zwischen **digitalem Design** und **digitaler Fertigung** dar. Auch

wenn in dieser Arbeit ausschließlich additive Verfahren betrachtet werden, wurde an dieser Stelle bewusst der Begriff *digitale Fertigung* gewählt, da hiermit die Möglichkeit offengelassen wird, dass andere computergesteuerte Fertigungsmethoden (z.B. subtraktive Verfahren) mit in das Konzept einbezogen werden können. Eine genaue Ausarbeitung des FIM-Konzepts wird in Kapitel 4 vorgestellt, anschließend wird in Kapitel 5 das Konzept prototypisch implementiert und mit einem Laborversuch getestet und schließlich wird in Kapitel 7 ein mögliches Datenformat für FIM untersucht.

2.6 Bearbeitungsfokus dieser Arbeit

Wie in Abschnitt 2.3 beschrieben, haben die verschiedenen Druckmethoden ganz unterschiedliche Voraussetzungen und entsprechend eigene Vor- und Nachteile beim Einsatz für bestimmte Aufgaben. Sollen z.B. große Objekte mit wenig geometrischen Feinheiten erstellt werden, so ist der Einsatz von ablegenden Verfahren tendenziell zu bevorzugen. Im Umkehrschluss könnte man gleichzeitig annehmen, dass bei Objekten mit vielen geometrischen Feinheiten Partikelbett-Verfahren besser geeignet sind. Wird aber AM nur indirekt eingesetzt, z.B. indem damit Schalungselemente gedruckt werden, dann sind sehr wohl auch die Partikelbett-Verfahren für den Bau großer Objekte geeignet. Eine ganz einfache Einteilung der Aufgabenbereiche, für die jeweils die einzelnen Methoden eingesetzt werden können, gibt es also in diesem Sinn nicht. In diesem Zusammenhang bleibt zudem abzuwarten, welche Anwendungsverfahren für die Grundprinzipien noch entwickelt werden.

Aufgrund des zeitlichen Rahmens einer Masterarbeit ist es jedoch unmöglich, alle Methoden der additiven Fertigung zu berücksichtigen. Im Rahmen dieser Arbeit wird deshalb nur die Methode der **Betonextrusion** (vgl. Abschnitt 2.3.2) näher betrachtet. Dies hat den Hintergrund, dass bei dieser Methode die Planung der Werkzeugbahn im Vergleich zu anderen Methoden, wie z.B. Shotcrete, leicht nachvollziehbar ist und ein Versuchsaufbau einfach realisiert werden kann. Ein weiterer Grund ist ein aktuelles Forschungsprojekt, in dem ein Verfahren entwickelt wird, wie der Beton düsennahe während des Druckprozesses gemischt werden kann. Diese Methode ermöglicht eine Veränderung der Betonzusammensetzung über den Druckprozess hinweg und steigert damit die Anzahl der in der Planung zu berücksichtigen Parameter und somit auch die Anforderungen an ein zugrunde liegendes Fertigungsmodell.

Eine weitere Einschränkung muss bei der zu betrachtenden Bauteilart getroffen werden. Angesichts der sehr hohen Geometriefreiheit, die mit additiver Fertigung geboten wird, verschwimmen die Grenzen hier zwar und es muss eigentlich eher nach Funktion oder nach geometrischer Komplexität abgegrenzt werden, wie z.B. eine in die Wand integrierte Treppe zeigt (vgl. Abb. 2.18), aber gerade auch deswegen ist eine Vereinfachung zunächst wichtig. Öffnungen für Fenster oder Türen müssen z.B. gesondert gehandhabt werden, da gegebenenfalls Stützstrukturen beim Druck eingesetzt werden müssen. Bei großen

Überhängen, wie z.B. in Abb. 2.24c gezeigt, muss der Winkel der Bezugsebene während des Druckvorgangs mehrmals geändert werden.

Aus Zeitgründen werden hier der Einfachheit halber vorerst nur Wände betrachtet, die zur Wärmeisolierung funktionsaktiviert sind, keine Öffnungen oder große Überhänge besitzen und insgesamt in horizontalen Schichten gedruckt werden können. In diesem Zusammenhang wird jedoch untersucht, wie parallel verschiedene Varianten von Fertigungsmodellen für ein Bauteil erstellt werden können.

Kapitel 3

Verwendete Software

Für die in dieser Arbeit bearbeiteten Aufgaben lassen sich diverse Softwarelösungen verwenden. Da hauptsächlich geometrische Bearbeitungen durchgeführt werden, könnten prinzipiell alle CAD-fähigen Programme verwendet werden. Dennoch ist es vorteilhaft, eine BIM-fähige Software zu verwenden, um grundlegend die Möglichkeit zu haben, semantische Informationen aus dem Gebäudemodell zu nutzen bzw. um Prozessparameter und Druckerdaten direkt mit an das Modell anheften zu können.

Im Folgenden werden drei Softwarelösungen vorgestellt, mit denen eine Bearbeitung der Aufgabenstellung durchgeführt werden kann. Dabei wird zunächst das native BIM-System Revit, dann das über ein Plug-in BIM-fähige Tool Rhinoceros3D und schließlich das nicht BIM-fähige Tool Siemens NX näher beleuchtet. Im Anschluss daran wird kurz die visuelle Programmierung erläutert und zuletzt das direkte Steuerungsprogramm des verwendeten Roboters beschrieben.

3.1 Autodesk Revit

Das Produkt Revit von der Firma Autodesk ist ein BIM-fähiges parametrisches 3D-Modellierungstool mit dem Fokus auf Gebäudemodellierung. Seit Version 2021 werden jedoch auch erweiterte Funktionen im Bereich Infrastruktur angeboten, bisher hauptsächlich für den Brückenbau, aber es werden laut der öffentlichen Ankündigungen im Rahmen des 20-jährigen Jubiläums der Software im Jahre 2020 weitere Neuerungen in diese Richtung folgen (CROTTY, 2020).

Ursprünglich wurde Revit von der Firma Charles River Software (später Revit Technology Corporation), gegründet im Jahr 1997, entwickelt und in Version 1.0 im Jahr 2000 auf den Markt gebracht. Innerhalb von 2 Jahren entwickelte das Softwareteam die Software sehr zügig weiter und brachte Anfang 2002 die Softwareversion 4.1 heraus. Im selben Jahr noch wurde die Entwicklerfirma von Autodesk aufgekauft und Revit wurde nach einer Reihe von weiteren Updates ab 2005 zunächst getrennt für Architektur und Tragwerksplanung und später auch noch für Gebäudetechnik weiterentwickelt. Die getrennten Softwarepakete, Revit Architecture, Revit Structure und Revit MEP, wurden dann schließlich 2013 zu einem *„all in one bundle“* zusammengeführt. Als nun BIM-fähiges Tool wird Revit im jährlichen Turnus weiterentwickelt. (»Revit History«, 2019)

Die gesamte Funktionalität des Revit-Benutzerumfangs lässt sich über das bereitgestellte Application Programming Interface (API) einfach ansteuern, somit ist es möglich, auch automatisierte Prozesse einfach in Revit-Projekten umzusetzen. Noch intuitiver lässt sich

das API über das **Dynamo**-Plugin steuern, einer visuellen Programmierschnittstelle (vgl. Abschnitt 3.4). Dynamo bietet hierbei einen umfangreichen Satz an nützlichen Werkzeugen zur einfachen Durchführung komplexer Arbeitsabläufe in Revit, wie z.B. die Erstellung komplexer parametrischer Geometrien oder adaptiver Elemente. Es basiert auf einem open-source Projekt und ist daher kostenfrei nutzbar und wird zudem ständig von Nutzern um weitere Funktionalität erweitert. (»BIM Blog«, 2015)

3.2 Rhinoceros3D

Die Software Rhinoceros3D, in der Regel als **Rhino** abgekürzt, ist ein Tool zur 3D-Gestaltung, welches auf Basis von NURBS-Modellierung arbeitet. Damit können beliebige Freiformen äußerst detailgenau dargestellt und über das Rhino-Dateiformat `.3dm` mit anderen Designern ausgetauscht werden. Die Software wird seit 1993, basierend auf einem Studienprojekt namens Sculptura, von dem Unternehmen *McNeel & Associates* entwickelt und kam 1998 in Version 1.0 für Windows auf den Markt. (»McNeel Wiki«, 2020)

Rhino ist seit 2018 in der Version 6 (bzw. aktuell als Rhino 6.30.20266[1]) erhältlich und wird regelmäßig mit kleineren Updates verbessert. Für das Scripting und das Softwaredevelopment bietet Rhino gleich mehrere Schnittstellen, davon seien z.B. die plattformübergreifenden *RhinoCommon* (.NET) und *Rhino.Python* Schnittstellen genannt. Seit der 6. Version ist auch das von McNeel & Associates entwickelte visuelle Programmiertool **Grasshopper** fester Bestandteil der Software. Vergleichbar zu Dynamo (siehe Abschnitt 3.1) können hiermit komplexe oder repetitive Arbeitsabläufe einfach ausgeführt werden (vgl. Abschnitt 3.4).

Alleinstehend ist Rhino ein reines CAD-Tool, kann aber durch Plugins (VisualARQ oder auch GeometryGym) BIM-fähig gemacht werden. Mit beiden genannten Plugins werden jeweils eigene neue Werkzeuge zum Bearbeiten von BIM-Modellen geliefert und es können IFC-Dateien in Rhino und Grasshopper geladen und auch wieder exportiert werden (GEOMETRYGYM, 2020; »VisualARQ Features«, 2020).

3.3 Siemens NX

NX, aktuell entwickelt von der Firma Siemens Digital Industries Software, ist eine umfassende Softwarelösung, welche für die Bereiche des digitalen Designs, Fertigung und Simulation (CAD/CAM/CAE) eingesetzt werden kann. Im eigentlichen Sinn ist NX eine Sammlung verschiedener einzelner Bestandteile, die sowohl jeweils für sich alleine verwendet werden können als auch als Komplettlösung für den gesamten Entstehungsprozess eines Bauteils. Die Software stellt auf Basis ihres **Parasolid-Modellierkern**s leistungsstarke parametrische 3D-Modellierungswerkzeuge bereit, wodurch Werkstücke sehr genau geometrisch beschrieben werden können. (»NX. Answers for Industry.« 2013)

[1] Stand 23.09.2020

Schon seit Anfang der 1970er wird das CAD-System zunächst von der Firma United Computing entwickelt, erst unter dem Namen **Unigraphics** und dann nach der Kombination mit der Software (I-DEAS) als NX. Übernommen wurde NX von Siemens im Jahr 2007 mit dem Kauf der Firma United Computing – bzw. zu diesem Zeitpunkt UGS Corporation – und seither als Product Lifecycle Management (PLM) Software weiterentwickelt (»Siemens Closes Acquisition of UGS«, 2007). Seit Januar 2019 wird NX als sogenannter „continuous Release“ veröffentlicht, also nicht mehr als fixe Version verkauft, sondern als ein laufender Service, der stetig, etwa alle 6 Monate geupdated wird (CHANATRY, 2018).

Mit dem Softwarepaket NX wird zudem eine Sammlung von APIs für verschiedene Programmiersprachen geliefert, unter anderem für C++, .NET und Python. Es können hiermit entweder eigenständige Programme, die auf eine laufende NX Instanz zugreifen können, oder direkt in das User Interface eingebettete Plugins erstellt werden. Dabei ist der komplette Funktionsumfang von allen NX Tools verfügbar. (SIEMENS, 2016)

Ein Grund, warum die Software trotz mangelnder BIM-Unterstützung hier in Betracht gezogen wird, ist die Tatsache, dass eines der Softwarepakete in der NX Sammlung auf additive Fertigung zugeschnitten ist. Dieses AM-CAM Paket erlaubt eine schnelle Übersetzung von 3D-Geometrien in maschinenverarbeitbaren Code und bietet die Möglichkeit, einen Druckvorgang zunächst zu simulieren und visuell darzustellen.

3.4 Visuelle Programmierung

Der Begriff *Visuelle Programmierung* beschreibt eine Programmiertechnik, bei der nicht direkt mit einer textbasierten Programmiersprache gearbeitet wird, sondern mit grafischen Elementen und deren Verknüpfungen. In einer Programmierumgebung, einer Art Leinwand, werden dabei sogenannte Blöcke angeordnet und über sogenannte Wires miteinander insgesamt zu einem Programm verbunden. Die folgende kurze Erläuterung der Funktionsweise kann in ähnlicher Weise den entsprechenden Tutorials von Dynamo oder Grasshopper entnommen werden (DYNAMOBIM, 2016; GRASSHOPPER3D, 2020).

Ein Block ist in diesem Zusammenhang ein vordefiniertes Element, das für eine Code-Sequenz steht. Hat ein Code-Block ein oder mehrere Eingänge (Verbindungsfelder auf der linken Seite des Blocks), können Variablen an den Block übergeben werden und damit innerhalb des Blocks verarbeitet werden. Genauso verhält es sich mit Ausgängen (Verbindungsfelder auf der rechten Seite des Blocks), werden innerhalb des Blocks Daten generiert, können sie auf der rechten Seite ausgegeben werden.

Ein und Ausgänge von Blöcken können mit den erwähnten Wires verbunden werden. Wires verbildlichen also eine Referenzierung von Daten. Ausgänge von Blöcken können auf diese Weise mit mehreren verschiedenen Eingängen verbunden werden umgekehrt ist das aber nicht bei jeder visuellen Programmiersprache der Fall. Im Fall von Dynamo müssen mehrere Variablen zuerst zu einer Liste kombiniert werden, um mit einem einzelnen

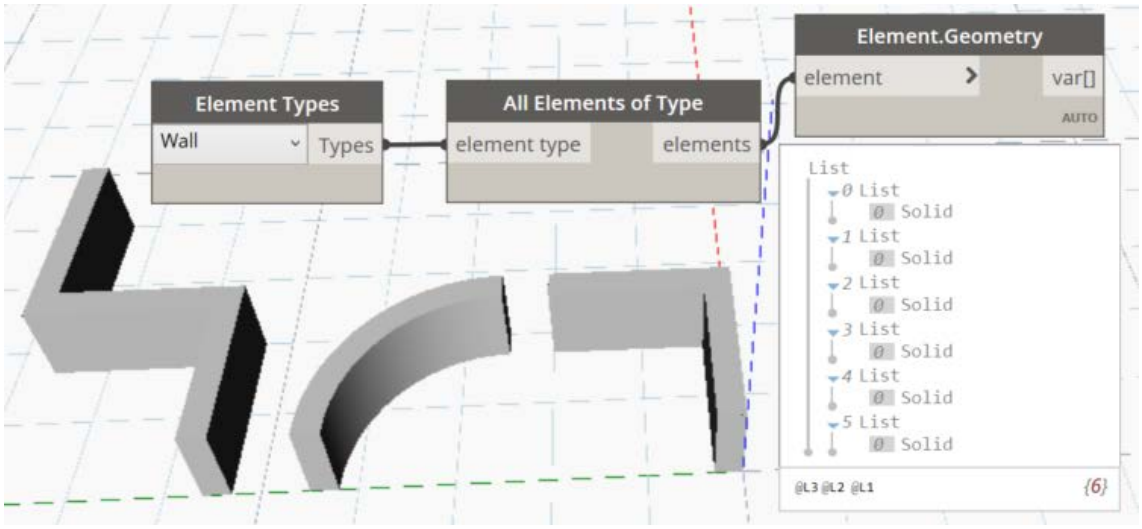


Abbildung 3.1: Dynamo Beispiel: Drei vordefinierte Blöcke, die jeweils über ein Wire miteinander verbunden sind und zusammen Geometrien aus einer laufenden Revit Instanz extrahieren.

Eingang verbunden werden zu können, dagegen können bei Grasshopper mehrere Wires auf einen Eingang verknüpft werden.

Ein einfaches Beispiel ist in Abb. 3.1 dargestellt. Die hier dargestellten Blöcke importieren alle Wandelemente aus der laufenden Revit Instanz und extrahieren die Geometrie. Hier wird auch deutlich gezeigt, dass Parameter im Block festgelegt werden und dass Daten auch aus externen Quellen (hier eine Revit Instanz) stammen können.

3.5 UR10e

Der Roboter UR10e von der Firma Universal Robots wird in dieser Arbeit zur Validierung des im folgenden Kapitel beschriebenen FIM-Erzeugungsalgorithmuses verwendet. Hierbei dient der Roboter als Manipulator für den Extrusionsdruck des im FIM dargestellten Bauteils. Der Roboter kann für diese Aufgabe über verschiedene Methoden gesteuert werden, über das Handgerät nach Online- oder Offline-Programmierung oder per Netzwerkschnittstelle. In diesem Abschnitt werden die damit zusammenhängenden Softwarepakete kurz erläutert.

3.5.1 Polyscope

Polyscope ist eine Software, die von der Firma Universal Robots zur direkten Steuerung ihrer eigenen Roboter entwickelt wurde. Diese Software ist auf dem Handgerät des in dieser Arbeit verwendeten Roboters (UR10e) installiert und liegt aktuell in der Version 5.9.4[2] vor. Neben der Handgerätversion gibt es dieses Programm auch in einer Simulationsversion sowohl zur Installation auf einem Linux-System als auch als Virtual Machine

[2]Stand 02.11.2020

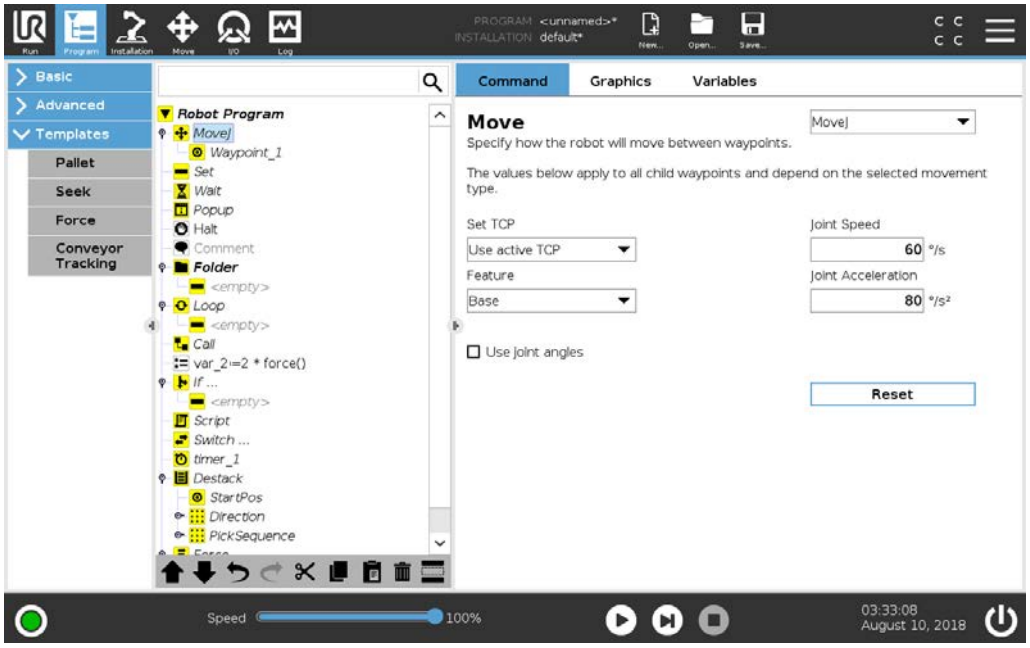


Abbildung 3.2: Polyscope Programmierung. (UR, 2020d)

Image, zur Verwendung auf anderen Systemen. Polyscope liefert hierbei eine grafische Benutzeroberfläche, über die unter anderem Einstellungen am Roboter vorgenommen (in der Simulationsversion nur für einen virtuellen Roboter), Steuerungsprogramme geschrieben und simuliert werden können.

3.5.2 Online Programmierung

Die Online-Programmierung der Steuerungsprogramme in Polyscope kann über das Touchdisplay durch Auswahl und Modifikation von **Programmknoten** durchgeführt werden. Diese werden in einer listenartigen Programmstruktur aneinandergereiht und können über den *Command* Tab weiter konfiguriert werden (vgl. Abb. 3.2). Zusätzliche Module, die an den Roboter angebracht werden (z.B. ein Kameramodul), können auf dieselbe Weise angesteuert werden, jedoch muss für jedes extra Modul eine eigene Erweiterung installiert werden, ein sogenanntes URCap. Sobald das Programm erstellt ist, kann dieses im selben Fenster gestartet (sowohl live als auch simuliert) und zu Testzwecken über einen Schieberegler die Geschwindigkeit angepasst werden. In der Regel erfordert der Start eines Programms eine sogenannte **Satzkompensationsfahrt**, also eine Bewegung zu einem definierten Startpunkt für die der Benutzer während der Fahrt durchgehend die Freigabe erteilen muss. Über den *Graphics* Tab kann sowohl während des regulären Ablaufs als auch der Simulation die Roboterstellung virtuell verfolgt werden.

3.5.3 Offline Programmierung

An einem PC kann im einfachsten Fall mit jedem beliebigen Editor eine Skript-Datei geschrieben werden (Dateiendung sollte *.script lauten), die am Roboter mit dem *Script* Programmknoten in ein Programm eingebunden werden kann. In einer solchen Datei wer-

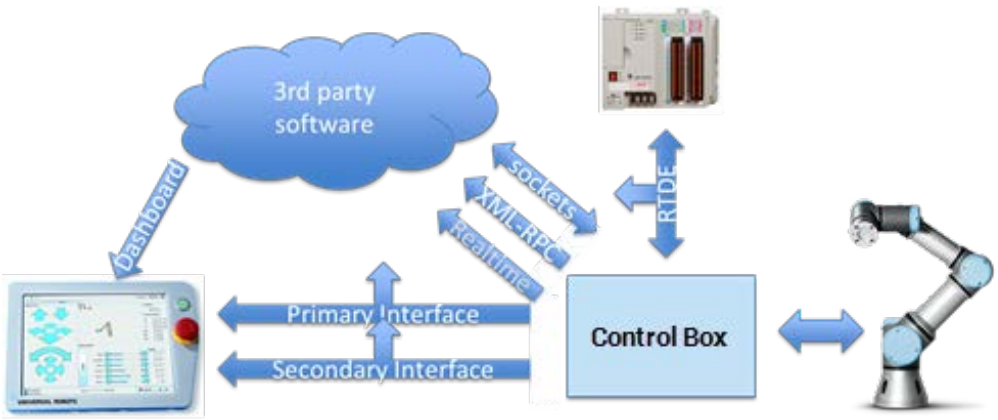


Abbildung 3.3: Kommunikationsschnittstellen von UR-Robotern. (UR, 2020b)

den alle Anweisungen für den Roboter in der herstellerspezifischen Programmiersprache URScript geschrieben. Diese Programmiersprache stellt eine einfache Skriptsprache dar, die über Variablen, Typen, Kontrollstrukturen, Funktionen etc. verfügt, und damit im Vergleich zum G-Code, der für CNC-Steuerung verwendet wird, deutlich vielseitiger. Komplett Offline kann mit Softwarepaketen wie z.B. RoboDK[3] gearbeitet werden. Mit genau dieser Software kann sowohl auf Skriptebene wie auch für die grafische Benutzeroberfläche programmiert werden.

Mit anderen Systemen, wie z.B. dem ROS, kann in einer ganz allgemeinen Programmiersprache programmiert werden. Das so erstellte Programm wird dann mit diesem System über einen Treiber an den Roboter übersendet und ausgeführt. In diesem Kontext ist der Treiber eine Art Übersetzer für die Interpretation des herstellerunabhängigen Programms.

3.5.4 Remote Control

Wie zuvor schon erwähnt, ist es bei den meisten Offline Verfahren zudem möglich, über das TCP/IP Protokoll Daten zu übertragen. Nicht erwähnt wurde jedoch, dass die komplette Steuerung auf diese Weise stattfinden kann. Z.B. kann mit ROS ein Steuerungsserver eingerichtet werden, über den Befehle an den Roboter übermittelt werden können, also komplett standortunabhängig via Internet. Abbildung 3.3 zeigt hierbei verschiedene Möglichkeiten, über die Daten empfangen und gesendet werden können.

Da diese Form der Steuerung in dieser Arbeit eine sehr untergeordnete Rolle spielt, wird hier nicht genau auf die verschiedenen Kommunikationsmöglichkeiten mit dem Roboter eingegangen. Eines ist aber hervorzuheben, mit dieser Art der Steuerung lassen sich Steuerungsprogramme adaptiv gestalten, also so, dass sie während dem Bewegungsablauf des Roboters verändert werden können. Im Fall von 3D-Druck, z.B., könnte der Druckvorgang laufend kontrolliert und wenn nötig der Druckpfad angepasst werden.

[3] https://robodk.com/doc/de/Robots-Universal-Robots.html

Kapitel 4

Fabrication Information Modelling (FIM)

In Abschnitt 2.5 wurde bereits von der Notwendigkeit berichtet, warum ein Mechanismus entwickelt werden muss, mit dem aus einem BIM-Modell Fertigungsinformationen für digitale Fertigungsverfahren abgeleitet werden können. Die immer wichtiger werdenden digitalen Fertigungsmethoden sind derzeit im Baugewerbe noch deutlich unterrepräsentiert und digitale Modelle hierfür entsprechend wenig entwickelt. Wie in Abschnitt 2.3 gezeigt, ist das Thema AM überaus komplex und wird zudem ständig um neue Möglichkeiten und Werkstoffe erweitert. Auch die vielen Kombinationsmöglichkeiten verschiedener Maschinensysteme (Abschnitt 2.3.4) und die Komplexität der entsprechenden Steuerungstechnik (Abschnitt 2.4) haben dafür gesorgt, dass bisher keine einheitliche Herangehensweise zur Integration von AM in das digitale Design entwickelt wurde.

Im Folgenden wird hierzu die Methodik des Fabrication Information Modellings (FIMs) vorgestellt. Hierzu wird zunächst der nötige Informationsgehalt des zugrunde liegenden FIM-Modells definiert und dann die zeitliche Einordnung sowie Vorgehensweise bei dessen Modellierung beschrieben.

4.1 Informationsumfang eines FIMs

Die für eine sinnvolle Integration von AM in BIM notwendigen Informationen können in drei Kategorien eingeteilt werden. Zuerst sind hier die **Prozessinformationen** zu nennen, die für die Steuerung der automatischen Fertigung notwendig sind. Darauf aufbauend können **Volumenmodelle** für die Repräsentation von Materialverteilungen generiert werden, mit deren Hilfe Simulationen durchgeführt und das entsprechende Bauteil *„as-planned“* dargestellt werden kann. Und schließlich sollte das FIM auch ein Konzept zur Handhabung von Validierungsdaten und „as-built“-Informationen beinhalten, z.B. in Form eines **digitalen Zwillings**[1]. Essenziell sind von den drei genannten Kategorien jedoch nur die Prozessinformationen, da diese effektiv den Fertigungsprozess beschreiben. Die anderen beiden Kategorien müssen nicht zwingend modelliert werden, um das entsprechende Bauteil zu fertigen. Ohne diese Daten lässt sich der Fertigungsprozess jedoch nicht optimieren und es können auch keine Vorhersagen über die Leistungsfähigkeit des Bauteils getroffen werden. Auch ist es dann unmöglich, die Fertigungsinformationen für eine weitere Nutzung im BIM-Kontext, wie z.B. der Gebäudeinstandhaltung, zugänglich zu machen. Wie die einzelnen Informationsbausteine zusammenhängen, wird in Abb. 4.1 gezeigt.

[1] Digitale Repräsentation eines Objekts aus der realen Welt, als „ digitales Abbild“ in der digitalen Welt (Tao et al., 2019)

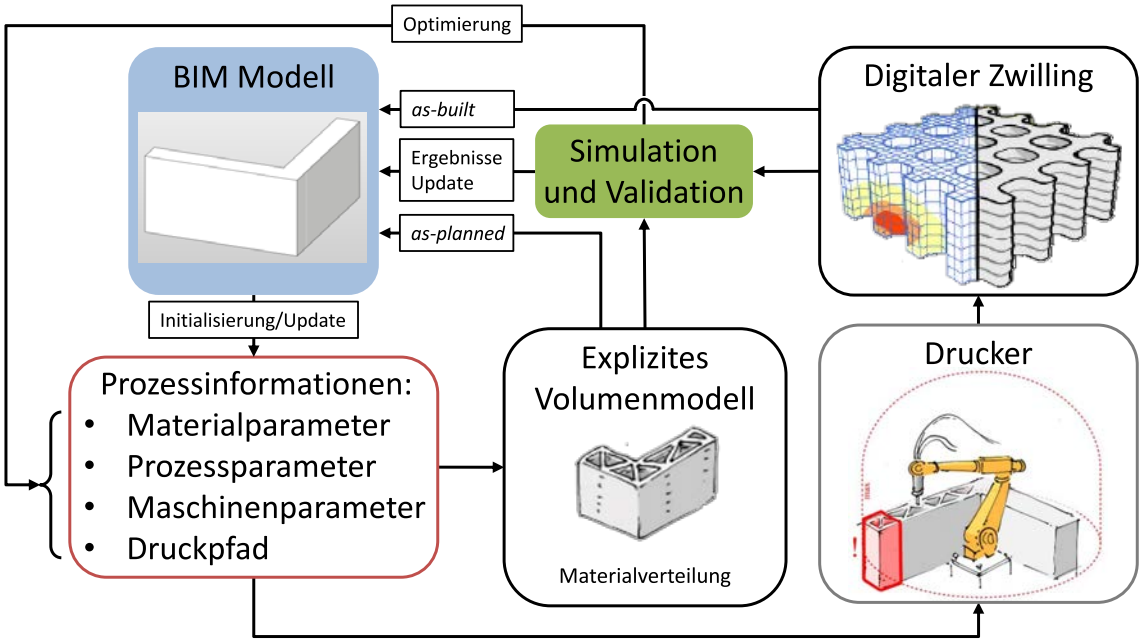


Abbildung 4.1: Zusammenhang der einzelnen FIM-Bausteine. Die farbig hinterlegten Bausteine sind hierbei nicht im FIM integriert. Auch sichtbar sind mehrere Schnittstellen, die zwischen BIM und FIM möglich sind.

4.1.1 Prozessinformationen

Eine absolute Festlegung des genauen Umfangs der Prozessdaten ist eine komplexe Aufgabe und muss aus den folgenden Gründen flexibel gehalten werden. Wie bereits in Abschnitt 2.3 dargestellt, gibt es viele verschiedene AM-Verfahren, die jeweils mit verschiedenen Maschinensystemen kombiniert werden können. Daher ist es möglich, dass jedes Verfahren jeweils einen eigenen Satz an Material-, Prozess- und Maschinenparametern haben kann. Für die AM-Methode **Betonextrusion** wurden hierzu die wichtigsten Parameter identifiziert und in Tabelle 4.1 aufgelistet, die für die Modellierung der Steuerungsinformationen zwingend notwendig sind. Zu erwähnen ist hierbei, dass dieser Parametersatz nur auf eine einfache Variante der Betonextrusion mit stationärem Robotersystem zutrifft und daher auch hierfür nur als minimaler Informationsgehalt angesehen werden kann. Wird z.B. während des Druckvorgangs die Betonzusammensetzung variiert, so müssen die einzelnen Materialströme separat voneinander modelliert werden. Denkbar ist z.B. auch, dass eine veränderbare Düse eingesetzt werden kann, womit auch die Formänderung der Düse entsprechend modelliert werden muss. Um demnach eine genaue Aussage treffen zu können, welche Parameter alle in das FIM aufgenommen werden müssen, ist eine genauere Untersuchung aller relevanten AM-Verfahren und aller Maschinensysteme notwendig.

Eine weitere wichtige Prozessinformation, die modelliert werden muss, ist der Druckpfad. Dieser wird aber nicht einfach nur aus den Geometriedaten des BIM-Modells abgeleitet, sondern abhängig von den zuvor genannten Parametern entworfen. Alle in Tabelle 4.1 gezeigten Parameter haben also einen Einfluss darauf, wie letztendlich der Druckpfad aussehen und auf welche Art und Weise er ausgeführt wird. Hierbei gelten z.B. die

Tabelle 4.1: Wichtige Druckparameter

Materialparameter	**Prozessparameter**	**Maschinenparameter**
Betonzusammensetzung	Düsengeschwindigkeit	Achsmaße
Konsistenz	Extrusionsrate	max. Achsauslenkung
Aushärtezeit	Schichthöhe	max. Achsgeschwindigkeit
Festigkeit	Düsenorientierung	max. Achsbeschleunigung
	Düsenform	maximale Materialförderrate
	Bezugsebene	

Maschinenparameter als limitierender Faktor, was die druckbare Bauteilgröße angeht. Dementsprechend können Druckpfade nur in einem bestimmten Bauraum (abgeleiteter Parameter) und in einer bestimmten Länge pro Schicht geplant werden. Zudem geben die Maschinenparameter einen „Arbeitsbereich“ vor, in dem Prozessparameter gewählt werden können. Z.B. kann die Extrusionsrate nicht die maschinenspezifische maximale Förderrate und die Düsengeschwindigkeit (bzw. Druckgeschwindigkeit) nicht die maximale Achsgeschwindigkeit überschreiten. Wie genau vorgegangen werden muss, um Druckpfad und Parameter aufeinander abzustimmen, wird in Abschnitt 4.3 beschrieben. Eine genauere Beschreibung der Parameter und deren Einfluss wird in Kapitel 5 im Rahmen der prototypischen Implementierung eines FIMs gegeben.

Dargestellt wird der Druckpfad am besten über einfache Liniengeometrien, oder genauer gesagt, durch eine Aneinanderkettung von Linien, Kreissegmenten und/oder Splines zu einer **Polykurve**. Diese Liniengeometrie beschreibt den exakten Weg, den der Druckkopf nachfahren muss, um das gewünschte Bauteil erstellen zu können. Kurven dieser Art können auf einfache Weise parametrisiert werden und damit können alle 3D-Druck-Parameter, die einen Einfluss auf das Druckgeschehen haben, linear zum Druckpfad referenziert werden. Z.B. könnte eine Änderung der Druckgeschwindigkeit in einem bestimmten Parameterbereich der Polykurve (beispielsweise an Ecken) funktional dargestellt werden. Das entsprechende Vorgehen wird in Abschnitt 4.3 konzeptuell beschreiben und anhand der prototypischen Implementierung in Kapitel 5 beispielhaft gezeigt.

Mit den Parametern und dem Druckpfad sind die wichtigsten Bestandteile der Prozessinformationen vollständig. Damit aber mit diesen Daten auch ein Druckprozess durchgeführt werden kann, müssen Parameter und Druckpfad in ein maschinenverständliches Programm übersetzt werden. Prinzipiell kann das FIM derart gestaltet werden, dass damit direkt die Maschinensteuerung durchgeführt werden kann – ganz nach dem Vorbild von STEP-NC (vgl. Abschnitt 2.4.1). Aber um alle denkbaren Maschinensysteme zu unterstützen, ist es sinnvoller, ein Übersetzungstool in das FIM zu integrieren, mit welchem Maschinensteuerungsprogramme aus den modellierten Daten abgeleitet werden können. Allerdings müssen hier verschiedene Datenformate für die Steuerungsprogramme unterstützt werden, jeweils für verschiedene Robotersysteme und Werkzeugmaschinen.

Eine Problematik gibt es jedoch, wenn die Maschinensteuerung derart externalisiert wird: Es kann sowohl die Extrudersteuerung als auch eine eventuelle Aufnahme eines digitalen Zwillings (vgl. Abschnitt 4.1.3) nicht direkt mit dem Druckpfad referenziert werden, da

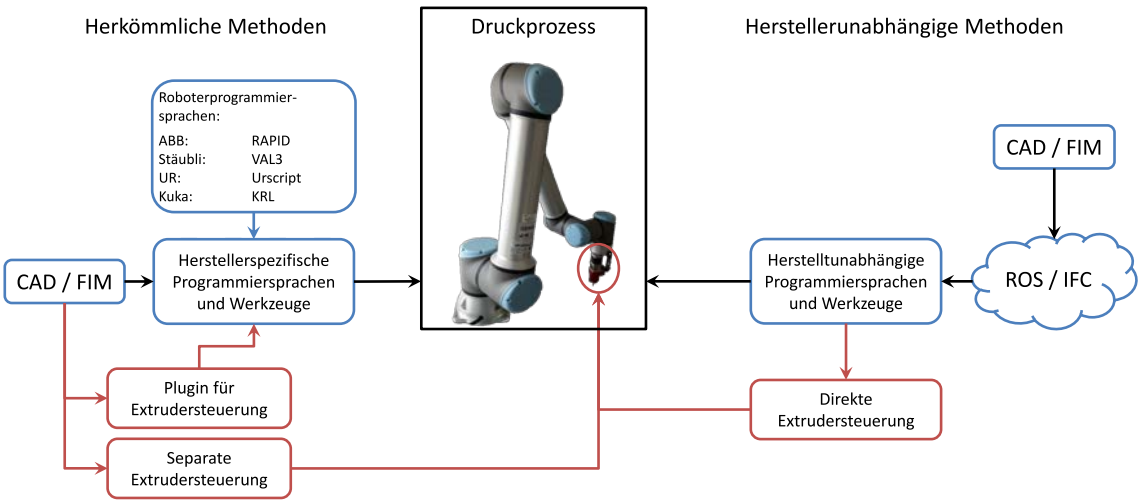


Abbildung 4.2: Gleichzeitige Steuerung von einem Roboter und weiteren Peripheriegeräten, wie hier der Extruder. Die Verwendung von einem „robot framework“, wie z.B. ROS, erleichtern deutlich den Aufwand der Implementierung.

entsprechende Steuerungsprogrammiersprachen nicht für diese Zwecke ausgestattet sind. Natürlich kann die Position des TCPs jederzeit von der Maschine ausgelesen werden und der entsprechend nächstmögliche Wegpunkt auf dem Druckpfad ermittelt werden, jedoch kann dies bei nahe zusammenliegenden Wegpunkten und gröberen Abweichungen durch den Roboter fehlerhaft sein. Für manche Steuerungssysteme ist es möglich, dass eigene Plug-Ins implementiert werden, die eine solche Funktionalität realisieren können, dies müsste aber dann für jedes entsprechende Robotersystem eigens durchgeführt und gepflegt werden und stellt damit einen erheblichen Aufwand dar. Eine bessere Lösung ist es an dieser Stelle anstatt der Herstellerspezifischen ein „robot framework“, wie z.B. ROS, zu verwenden, mit dem gleichzeitig verschiedene Systeme gesteuert werden können. Damit kann exakt an jedem Wegpunkt auch der Extruder entsprechend angesteuert werden sowie eventuell eine Aufnahme der gedruckten Geometrie erfolgen. Abbildung 4.2 stellt hier die verschiedenen Möglichkeiten gegeneinander auf. Alternativ dazu kann auch eine Implementierung direkter Steuerungsmöglichkeiten durch das FIM hierfür eine Lösung sein, allerdings muss hier erst untersucht werden, ob ein solches Vorhaben überhaupt mit einem entsprechenden Datenmodell für das FIM vereinbar ist (vgl. Kapitel 7).

4.1.2 Volumenmodell für Materialverteilung

Für eine Beschreibung des Druckprozesses ist eine 3D-Repräsentation des Druckkörpers, also eine Art *„pre-print Modell“*, nicht zwingend erforderlich. Dennoch ist eine Integration einer 3D-Repräsentation trotz der Optionalität durchaus sinnvoll, wie im Folgenden ausgeführt wird. Für eine Untersuchung, ob der 3D-Drucker den geplanten Druckpfad mit den eingestellten Parametern rein vom Bewegungsablauf her überhaupt ausführen kann, ist ein 3D-Modell nicht nötig, dies kann allein mit den entsprechenden Parametern und dem Druckpfad durchgeführt werden. Hierzu kann z.B. mit einer entsprechenden Software der Druckpfad und ein digitales Modell eines Robotersystems geladen und

eine Bewegungssimulation durchgeführt werden. Jedoch ist es für die Modellierung der Fertigungsinformationen, also des FIMs, sinnvoll, wenn unterstützend Simulationen durchgeführt werden, die über die Performanz des Bauteils und die Strukturstabilität während des Druckvorgangs vorhersagen treffen können. Nur mit 3D-Daten kann das Materialverhalten während dem Druckprozess simuliert und ein eventuell planungsbedingtes Materialversagen vorzeitig erkannt werden.

Es kann hiermit aber nicht nur die Druckbarkeit geprüft, sondern auch eine Optimierung durchgeführt werden. Z.B. kann über Vorhersagen des Materialversagens auch eine optimale Druckgeschwindigkeit ermitteln werden, also die Geschwindigkeit, bei welcher der Druckkörper stabil bleibt (kein plastisches Versagen aufgrund Eigengewicht) aber gleichzeitig ein hoher Schichtenverbund gewährleistet wird (kurze Aushärtezeit vor Auflage der nächsten Schicht). Auch die spätere Leistungsfähigkeit kann mit einem entsprechenden Volumenmodell vorab per Simulation abgeschätzt werden. Wenn z.B. der Wärmedurchgang durch die Innenstruktur eingedämmt werden soll, ist es sinnvoll, die Geometrie der Innenstruktur auf maximale Effektivität zu optimieren. Auch könnten diese Daten hilfreich sein für weitere Bauplanungen, indem durch Simulationen gesammelte Erfahrungen für ein Bauteil in einer Art Wissensdatenbank gesammelt und bei der Planung von anderen Bauteilen angewendet werden können. Auf ein solches System, in der Einleitung (Kapitel 1) als DDSS bezeichnet, wird in dieser Arbeit jedoch nicht im Detail eingegangen.

Für eine derartige Modellierung der Materialverteilung gibt es verschiedene Möglichkeiten. Es kann z.B. das Volumen über **Boundary representation** (B-Rep) oder per **Volume representation** (V-Rep) dargestellt werden. Zur ersten Kategorie gehören dabei unter anderem polygonale Mesh-Darstellungen oder bivariate NURBS Flächen und zur zweiten polyedrische Mesh-Darstellung oder Voxels (PAOLINI et al., 2019). Die jeweiligen Darstellungsarten haben dabei verschiedene Vor- und Nachteile. In Kapitel 5 wird hierzu untersucht, wie sich B-Rep Modelle aus dem Druckpfad über eine Sweep-Operation erstellen lassen und wie geeignet derartige Modelle letztendlich sind. Und in Kapitel 7 wird nochmals auf ein mögliches Datenmodell für die Darstellung dieser 3D-Geometrie eingegangen.

4.1.3 Digitaler Zwilling

Für einige Aufgaben im BIM-Kontext ist es hilfreich, für Bauteile ein „ as-built“-Modell zu erstellen, wenn es z.B. um Gebäudeinstandhaltung und den Abriss geht. Wie zuvor schon erwähnt ist die Erstellung eines solchen Modells rein optional, bietet aber besonders für digitale Fertigungsprozesse, wie im Folgenden ausgeführt, einige Vorteile und wird darum als Bestandteil eines FIMs definiert.

Bei der Anwendung von AM-Methoden kann die Erstellung eines solchen Modells direkt mit dem Druckprozess gekoppelt werden, indem Kameras oder Laserscanner am Druckkopf angebracht und damit beim Druckprozess mitgeführt werden. Auf diese Weise wird der Druckkörper direkt während dem Druckprozess Schicht für Schicht gescannt, wobei unter anderem gleichzeitig der Bereich vor und hinter der Druckdüse aufgezeichnet werden

kann. Es kann also mit zeitlichem Versatz die Innenstruktur eines Bauteils aufgezeichnet werden und so eventuelle Formänderungen infolge chemischer oder mechanischer Prozesse wahrgenommen, also ein kontinuierlicher Überwachungsprozess durchgeführt werden. Schließlich kann nach dem Druckprozess und entsprechender Aushärtezeit ein weiterer Scann durchgeführt werden und das damit gewonnene *„ post-print Modell“* zur Validierung der Planungsdaten herangezogen und als Basis für eventuelle weitere Nachbearbeitungsschritte, wie z.B. eine Oberflächenbehandlung, verwendet werden. Auch für die Planung anderer Projekte können solche Daten als Erfahrungswerte sehr vorteilhaft sein, z.B. können etwaige Formänderungen bei der Prozessplanung direkt berücksichtigt werden. Es wurde bewusst die neue Bezeichnung „post-print“ anstatt „as-built“ gewählt, da der Druckvorgang nicht unbedingt der letzte Bearbeitungsschritt sein muss, folgen aber keine Nachbearbeitungsschritte, dann sind beide Begriffe als zueinander synonym aufzufassen.

Genau wie bei den Volumenmodellen gibt es für einen digitalen Zwilling verschiedene Möglichkeiten der Darstellung. Für einen möglichst einfachen Soll-Ist-Abgleich, ist es aber sinnvoll, dass sowohl das Volumenmodell als auch der digitale Zwilling im gleichen Datentyp vorliegen. In Kapitel 7 wird hierauf konzeptionell eingegangen, eine tief greifende Ausarbeitung und prototypische Implementierung eines digitalen Zwillings wird in dieser Arbeit jedoch nicht durchgeführt.

4.2 Zeitliche Einordnung von FIM

Wie bereits in Abschnitt 2.1.2 beschrieben, sind mehrere Herangehensweisen vorstellbar, wie FIM in den BIM-Workflow eingegliedert werden kann. Im Extremfall kann entweder bereits von Anfang an bedacht werden, dass AM eingesetzt wird und durchgehend unter dieser Voraussetzung geplant werden, oder bis zu einer hohen Detailstufe naiv geplant werden und erst dann eine AM-Planung angefangen werden. Wird schon in der frühen Planung festgelegt, welche AM-Methode verwendet werden soll, muss bei der Planung eventuell eine eingeschränkte Freiheit in Kauf genommen werden, dafür können aber Probleme bei der Modellierung der Fertigungsinformationen leicht vermieden werden. Umgekehrt, wenn frei designet wird, kann es sein, dass die Erzeugung der Fertigungsdaten kompliziert wird, da eventuell einige Details für eine Verarbeitbarkeit mit AM angepasst werden müssen. Um alle Herangehensweisen abdecken zu können und aufgrund der im Folgenden beschriebenen Komplexität bei der Modellierung der Fertigungsdaten, wird in dieser Arbeit ein Mechanismus vorgeschlagen, der entkoppelt aber parallel zu BIM ablaufen (vgl. Abb. 4.3) und ein separates Fertigungsinformationsmodell (FIM) erzeugen soll. Viele der innerhalb des FIM-Prozesses erzeugten Daten und Zwischenergebnisse werden hierbei unter Umständen mehrfach überarbeitet (vgl. Abb. 4.1).

Wie aus Abb. 4.3 zudem ersichtlich ist, kann der eigentliche FIM-Prozess erst dann beginnen, sobald genügend Informationen hierzu vorhanden sind. Eine Berücksichtigung von AM-Methoden in der frühen Planungsphase bedeutet nicht, dass hier schon Ferti-

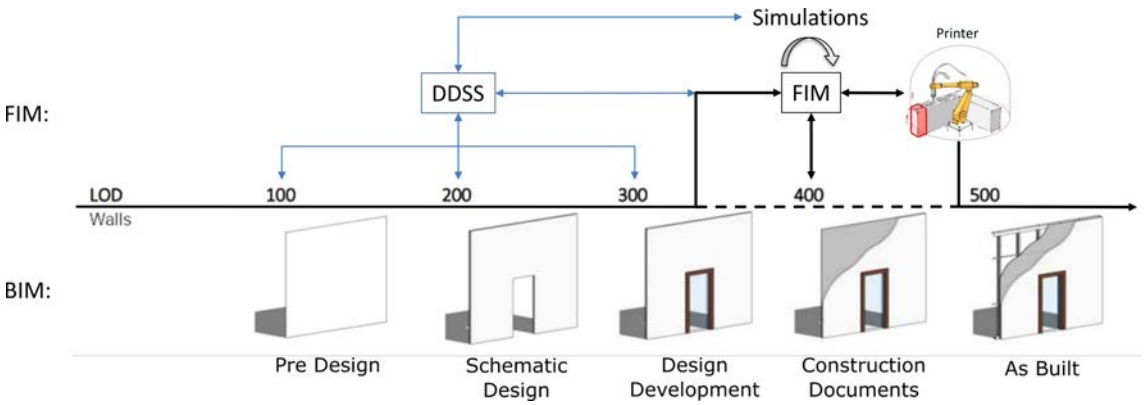


Abbildung 4.3: FIM als paralleler Prozess zu BIM für die Erstellung von Fertigungsdaten mit DDSS als Schnittstelle zu frühen Planungsphasen.

gungsdaten modelliert werden. Es werden in der frühen Phase lediglich Entscheidungen getroffen, welche die spätere Modellierung der Fertigungsdaten erleichtern. Diese teils speziellen Entscheidungen, z.B. Form und Ausstattung einer Wand, müssen hierbei jedoch mit fundierter Erfahrung mit der entsprechenden AM-Methode getroffen werden. Da nicht jeder Designer über eine entsprechende Erfahrung verfügt, wird hierfür in einem aktuellen Forschungsprojekt ein System entwickelt, das in der Einleitung (Kapitel 1) erwähnte DDSS. Dieses System soll als Schnittstelle zwischen frühem BIM-Design und FIM die Entscheidungsfindung erleichtern. Bis zu einem gewissen Detaillevel des BIM-Modells kann also indirekt auf die Modellierung Einfluss genommen werden, um erst dann vereinfacht mit dem FIM zu beginnen.

Um mit der Gestaltung der Fertigungsinformationen für ein Bauteil beginnen zu können, ist es zunächst notwendig, dass die Geometrie des zu druckenden Objekts definiert ist. Für die konventionelle Bauweise von z.B. Wandelementen reicht es hierbei aus, die Geometrie über die Außenflächen (B-Rep) zu beschreiben. Da mit AM aber zusätzlich die Möglichkeit besteht, auch das Innere eines Bauteils zu definieren, müssen hierzu vorab weitere Informationen modelliert werden. Es kann dabei entweder bereits eine Geometrie für die Innenstruktur vorgegeben werden oder indirekt eine bestimmte Eigenschaft bestimmt werden, die das Bauteil mit einer entsprechenden Innenstruktur erfüllen soll, wie z.B. eine erhöhte Wärmedurchgangsisolation. Wird nur eine entsprechende Eigenschaft vorgegeben, die das Bauteil haben soll, kann während der Erstellung der Fertigungsinformationen auch eine Variantenanalyse durchgeführt werden, also z.B. verschiedene Innenstrukturen modelliert und per Simulation miteinander verglichen werden. Sind diese Daten vorhanden, können Fertigungsinformationen wie im Folgenden beschrieben erzeugt werden.

4.3 Vorgehensweise von FIM

Wenn nicht schon in der frühen Planungsphase geschehen (mit Hilfe des DDSS), muss zunächst abhängig von der Geometrie des Bauteils ein geeignetes AM-Verfahren mit entsprechenden Manipulatorsystem (vgl. Abschnitt 2.3) gewählt werden. Bei der Auswahl des Verfahrens kommen dabei verschiedene Faktoren, wie z.B. maximal möglicher

Überhang, Zellstrukturen, interne Materialgraduierung und Ähnliches zum Tragen. Das Maschinensystem wird anhand des benötigten Bauraums und des eingesetzten Verfahrens ausgewählt.

Prozessinformationen (essentiell)

Entsprechend dem Verfahren und der Maschine wird dann ein passender Parametersatz angelegt und dessen Werte optimiert. Wie aber bereits in Abschnitt 4.1 angedeutet, kann diese Optimierung ein durchaus komplizierter Prozess sein, da die im Parametersatz enthaltenen Parameter (vgl. Tabelle 4.1) sich gegenseitig beeinflussen oder einschränken können. Ein paar der Parameter sind fest an bestimmte Entscheidungen gebunden, wie z.B. die Maschinenparameter, die anderen können aus einem gewissen Wertebereich gewählt werden, der unter anderem durch die fixen Parameter gegeben ist. Es bestehen selbst in einem einfachen Beispiel mehrere Abhängigkeiten, die es erforderlich machen, dass iterativ und simulationsgestützt eine optimale Lösung für die Parameterbelegung gefunden wird. Genauere Hinweise auf die verschiedenen Zusammenhänge werden hierzu in Kapitel 5 gegeben.

Um die erste Iteration zu starten, bietet es sich an, die Parameter, welche die Düsenform beschreiben und die Schichthöhe festzulegen, eine geeignete Innenstruktur (oder mehrere für einen Variantenvergleich, Kapitel 5) zu wählen und in einem ersten Schritt damit einen Druckpfad (pro Variante) zu erstellen. Ein Beispiel, wie solch ein Pfad erstellt werden kann, wird in Kapitel 5 gezeigt und detailliert erklärt, es wird hier auch detaillierter auf mögliche Innenstrukturen eingegangen. War diese Operation erfolgreich, muss mit geeigneten Softwarelösungen geprüft werden, ob der generierte Pfad überhaupt gültig ist und gegebenenfalls nachbearbeitet werden. Mit gültig ist hier zu verstehen, dass der Pfad von dem eingesetzten Maschinensystem überhaupt störungsfrei angesteuert werden kann.

Dieser ansteuerbare Pfad muss aber noch lange nicht ausführbar sein, denn die anderen Parameter (Geschwindigkeiten und Materialparameter) können hier noch Problem bereiten. Daher muss nun anhand der Länge des Pfades pro Schicht die Geschwindigkeit und der Volumenstrom des Extruders eingestellt werden, sodass diese die maximalen Werte der Maschine nicht übersteigen und mit für eine bestimmte Materialzusammensetzung verträglich sind. Bei dieser Einstellung spielen die Eigenschaften des Baustoffs eine große Rolle. Im Fall von Beton kann die Geschwindigkeit einen starken Einfluss auf die Qualität des gedruckten Bauteils haben und im schlimmsten Fall ist mit dem gegebenen Druckpfad eine sinnvolle Einstellung gar nicht erst möglich. Falls keine Einstellung gefunden werden kann, muss an dieser Stelle von Neuem begonnen werden.

Welcher Parameter auf welche Weise neu geplant werden muss, kann nur schwer vollautomatisch gestaltet werden, da häufig Simulationen notwendig sind, um fundierte Entscheidungen treffen zu können und damit eine individuelle Auswertung der entsprechenden Ergebnisse einhergeht. Es kann hier möglich sein, dass das Druckverfahren geändert, die Materialzusammensetzung neu eingestellt oder der Druckpfad bzw. schlimmer noch,

die Geometrie des Bauteils neu angepasst werden muss. Geometrische Besonderheiten, wie spezielle Öffnungen oder Einbauten, können hier auch ganz individuelle Probleme verursachen. Eine vollständig automatische Generierung von Fertigungsdaten für alle Bauteilarten und Fertigungsmethoden wird mit dem FIM daher noch nicht in Aussicht gestellt. Für spezielle Anwendungen, bei denen eine Reihe von Einschränkungen in der Designfreiheit gegeben sind, ist es durchaus denkbar, dass automatische Systeme implementiert werden können, für eine definitive Aussage hierzu muss jedoch noch deutlich mehr Erfahrung im Umgang mit AM im Bauwesen erlangt werden.

Volumenmodell (optional)

Sobald ein potenzieller Druckpfad erzeugt wurde, kann mit dessen Hilfe ein Volumenmodell abgeleitet werden, mit dem das Druckergebnis abgeschätzt werden kann. Wie zuvor erwähnt, ist ein solches Modell notwendig für diverse Simulationszwecke, deren Ergebnis wiederum einen Einfluss auf die iterative Pfadplanung haben können oder mit denen ein Vergleich verschiedener Varianten durchgeführt werden kann. Wird z.B. mit einer solchen Simulation festgestellt, dass bestimmte Eigenschaften des fertigen Bauteils unter den geforderten Sollwerten liegt, muss zwingend eine Anpassung an der Geometrie oder dem Druckpfad vorgenommen werden oder eine Variante von der Auswahl ausgeschlossen werden.

Für die Erstellung eines derartigen Volumenmodells sind verschiedene Herangehensweisen möglich, je nachdem welche Repräsentation gewählt wird (vgl. Abschnitt 4.1.2). In jedem Fall kann der Druckpfad als Referenz verwendet werden, z.B. in Form eines Sweeps (vgl. Kapitel 5). Solche geometrischen Operationen können allerdings recht zeitintensiv sein und damit bei vielen kleinen Änderungen am Druckpfad viel Rechenleistung erfordern, wenn jeweils ein Update des Volumenmodells erstellt werden soll.

Maschinensteuerung (essentiell)

Sobald der Druckpfad und sämtliche Parameter optimiert wurden und damit ein zufriedenstellendes Ergebnis zu erwarten ist, kann eine Übersetzung in maschinenverständlichen Code durchgeführt werden, wenn nicht direkt schon im FIM-Datenmodell nach dem STEP-NC Vorbild (vgl. Abschnitt 2.4.1) die Steuerung realisiert werden kann oder soll. Hierzu kann, wie in Abschnitt 2.4.1 beschrieben, der Druckpfad (Polykurve) in einzelne Wegpunkte diskretisiert werden, die der Manipulator (vgl. Abschnitt 2.3.4) ansteuern soll. Mit einem Bahnsteuerungsbefehl, also der Wahl, wie zwischen den Wegpunkten interpoliert werden soll, und der wegpunktweisen Einstellung der Geschwindigkeit ist es dann möglich, ein vollständiges Maschinensteuerungsprogramm zu erstellen. Das FIM ist also nicht zwangsläufig als Ersatz für gängige Steuerungsprogramme, wie etwa den G-Code, gedacht. Es soll aber mindestens eine einfache Möglichkeit bieten, ein solches Steuerungsprogramm aus den modellierten Daten ableiten zu können.

Wird der Druckpfad, wie zuvor vorgeschlagen, als Polykurve abgespeichert, also zusammengebaut aus mathematisch beschreibbaren Kurven, wie Linien, Kreisbögen und Splines, ist es möglich, die Anzahl der Wegpunkte, also den Diskretisierungsgrad der Kurve beliebig einzustellen. Somit kann für den Maschinensteuerungscode eine beliebige Auflösung realisiert werden. Dies ist besonders dann von Vorteil, wenn gleichzeitig zur Maschinensteuerung auch weitere Geräte angesteuert werden müssen, wie etwa der Extruder, und hier z.B. die Extrusionsrate funktional entlang des Druckpfades geändert werden soll. Hier kann an Bereichen des Druckpfades, bei denen große Änderungen stattfinden, die Anzahl der Wegpunkte dichter gewählt werden und an weniger wichtigen Bereichen weniger dicht. Zudem erlaubt diese Darstellung als Kurve mit Referenzierten Parametern eine Übersetzung in jede beliebige Maschinensteuerungssprache, sei es G-Code oder eine der vielen Programmiersprachen für Robotersteuerung.

Digitaler Zwilling (optional)

Eine genaue Ausarbeitung dieses Konzepts ist in dieser Arbeit nicht angedacht und eine mögliche Herangehensweise wurde hierzu schon in Abschnitt 4.1.3 gegeben.

Kapitel 5

Automatisierte Erstellung eines Fertigungsmodells

Im Folgenden wird untersucht, wie die in Kapitel 4 konzeptionell ausgearbeitete Methodik des Fabrication Information Modellings (FIMs) prototypisch umgesetzt werden kann. Zur Bearbeitung dieser Aufgabe wird zunächst untersucht, welche Software für diesen Zweck bisher genutzt wurde und welche Probleme damit einhergingen. Anschließend werden anhand eines Flussdiagramms die Implementierung vorgestellt und die einzelnen Bestandteile kurz erläutert. Darauf folgend werden die in Kapitel 4 beschriebenen FIM-Bestandteile für den Fall der Betonextrusion untersucht und detailliert beschrieben sowie Probleme identifiziert. Dann wird eine prototypische Implementierung einer Pfadgenerierung durchgeführt und diese anschließend genutzt, um eine 3D-Soll-Geometrie des Druckkörpers für Simulationen zu erzeugen. Die erstellten Pfade werden zudem mittels eines Versuchsaufbaus getestet und bei der Ausführung aufgetretene Probleme diskutiert.

5.1 Softwareauswahl

Grundsätzlich kann als Vorlage für den Aufgabenteil der Pfadplanung jegliche 3D-CAD-Darstellung genutzt werden. Die meisten Slicing-Tools, die es auf dem Markt gibt, können unterschiedliche Dateiformate einlesen und davon automatisch Druckpfade ableiten, wie z.B. die kostenlosen Open-Source Softwarelösungen Cura, OctoPrint oder Slic3r. Aber auch wenn diese Softwarepakete in der Regel einen recht großen Bedienumfang liefern, so ist es schwierig, diese Software direkt in einen durchgängigen BIM-Workflow zu integrieren. Die unterstützten Dateiformate sind hierbei fast nur rein geometrische Formate, wie STL, OBJ oder 3MF. In eines dieser Formate müsste also erst exportiert werden, um damit die Druckpfade zu generieren, nur um diese dann zurück in das BIM-Modell zu importieren. Dieses Funktionsprinzip wird in Abb. 5.1 genauer dargestellt.

Wie in Abb. 5.1 erkennbar und nochmals in Abb. 5.2 verbildlicht ist, sind die Konvertierungsprozesse zwischen den verschiedenen Softwaretools unidirektional, sodass eine Rückführung der einzelnen Teilergebnisse nur erschwert, bis teils gar nicht möglich sind. Obwohl in diesem Beispiel im ersten Schritt, der 3D-Gestaltung, mit einem BIM-Modell gearbeitet wird, gehen gleich mit der ersten Konvertierung die semantischen Daten verloren. Ein möglicher Rückkopplungsmechanismus wird in Abb. 5.1 im Punkt 4.2, dem STL zu CAD Konverter, gezeigt, jedoch wird hier nur die erzeugte Geometrie zurückgeführt. Die Umwandlung von einem 3D-Körper in tesselierter Oberflächenbeschreibung zu einer mit Semantik angereicherten B-Rep Darstellung führt zweifelsohne zu diversen

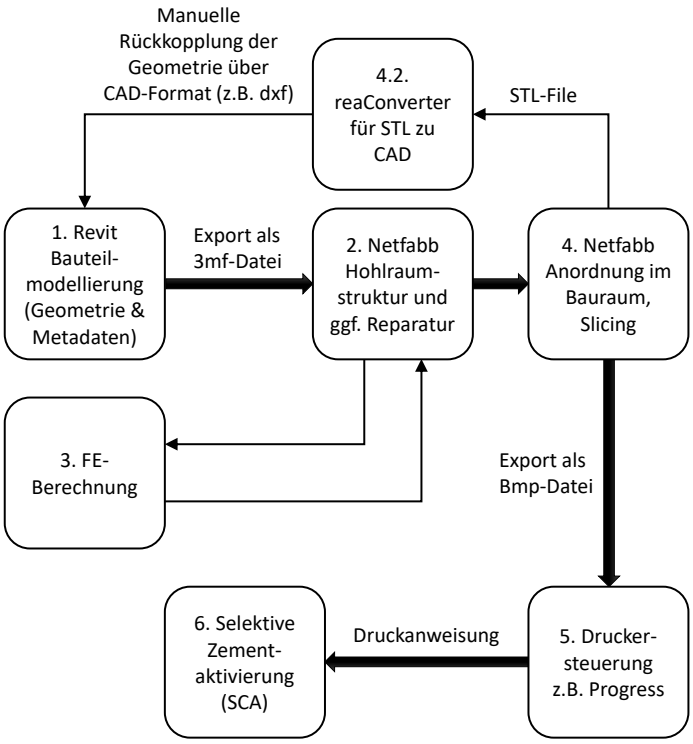


Abbildung 5.1: Datenaustausch Beispiel: Getrennte Softwarelösungen für CAD und CAM. (nach KRUSE, 2019)

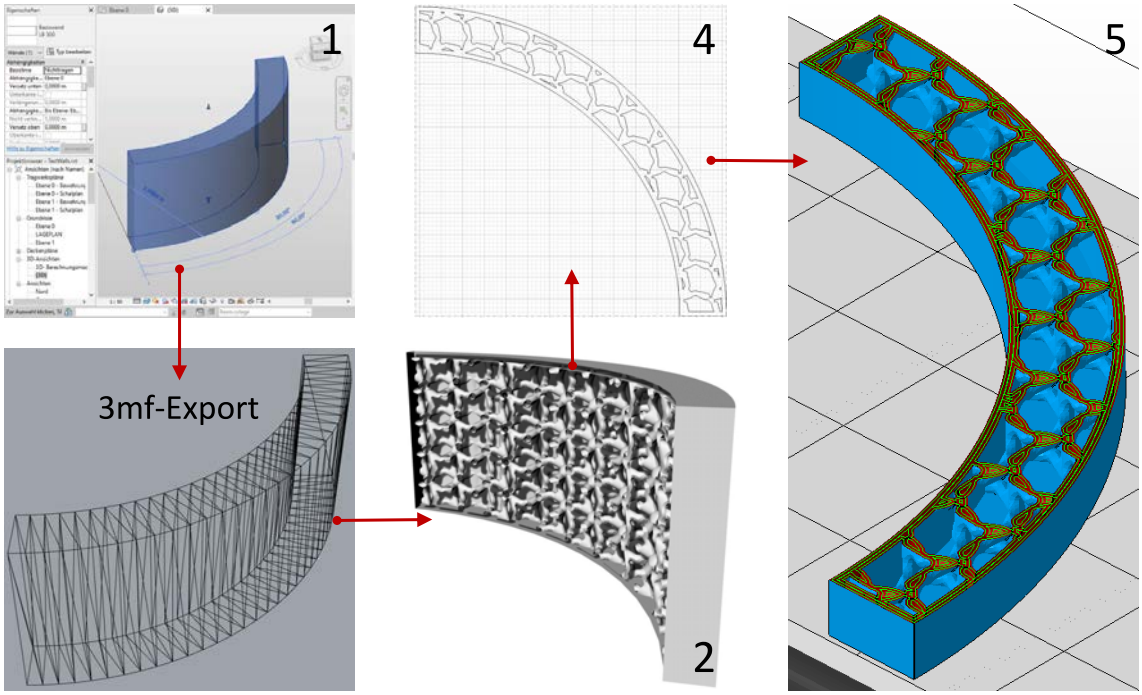


Abbildung 5.2: Datenaustauschszenario von Abb. 5.1 anhand eines Testprojekts. Ausgehend von einem BIM-Projekt (1) wird eine 3MF-Datei exportiert, in Netfabb mit einer Hohlraumstruktur (2) versehen, geslicet (4) und in Druckpfade umgewandelt (5). Die Nummerierung ist hierbei dieselbe wie in Abb. 5.1 und die FE-Berechnung (3) ist in dieser Abbildung nicht dargestellt.

Ungenauigkeiten, und möglicherweise funktioniert eine korrekte Zuweisung bestimmter Informationen auf diese Weise nicht vollständig. Während der Pfadplanung und auch im Nachgang kann es zudem sehr sinnvoll sein, direkt auf semantische Informationen zugreifen zu können. Ohne diese Informationen ist z.B. eine spezielle Pfadplanung nicht unbedingt voll automatisch aus dem Modell ableitbar. Soll z.B. eine aktivierte Wand konzipiert werden, die einseitig verstärkt und auf der anderen Seite mit einer wärmeisolierenden Innenstruktur gedruckt werden soll, muss bei der Pfadplanung bekannt sein, welche Seite die Innen- und welche die Außenwand darstellt. Auch für Simulationen, die im Nachgang zur Pfadplanung durchgeführt werden, ist es in diesem Kontext notwendig zu wissen, welche Seite der Wand außen liegt. Natürlich können semantische Informationen separat als Parameter nach der Dateikonvertierung übergeben werden, dies müsste aber bei jeder Veränderung des Bauteils passieren und stellt somit eine zusätzliche Fehlerquelle dar. Auch eine Rückkopplung von Druckerdaten ist in dieser Ansicht nicht vorgesehen und auch nicht möglich. Zwar ist zumindest diese Konvertierung zum Maschinenprogramm derzeit noch nicht vermeidbar, wie bereits in Abschnitt 2.4.1 und Kapitel 4 erwähnt, jedoch kann hier eine deutlich bessere Rückkopplung mit „robot frameworks“, wie z.B. ROS, erziehlt werden.

Eine bessere Lösung im Vergleich zu dem in Abb. 5.1 gezeigten Workflow ist, die Pfadgenerierung direkt in einer BIM- oder einer PLM-Software durchzuführen, wie z.B. die in Kapitel 3 vorgestellten Softwarepakete Revit, Rhino und Siemens NX. Im Rahmen dieser Arbeit wird der Fokus hierzu hauptsächlich auf die Software Revit gelegt, bei der direkt über das Plug-In Dynamo ein Skript zur Pfadgenerierung erstellt werden kann. Zum Vergleich wurde ein entsprechendes Flussdiagramm der FIM-Herangehensweise erstellt (vgl. Abb. 5.3). Der Datenaustausch in diesem Szenario erfolgt hauptsächlich über das FIM-Planungsskript in Dynamo, welches über die Revit-API immer mit den BIM-Daten verbunden ist. Die im Diagramm gezeigte Druckersteuerung ist bisher nur testweise erfolgt und kann damit nicht vollständig validiert werden, aber erste Versuche hierzu waren sehr vielversprechend. Auch wenn der in Abb. 5.3 gezeigte Workflow das in Kapitel 4 vorgestell-

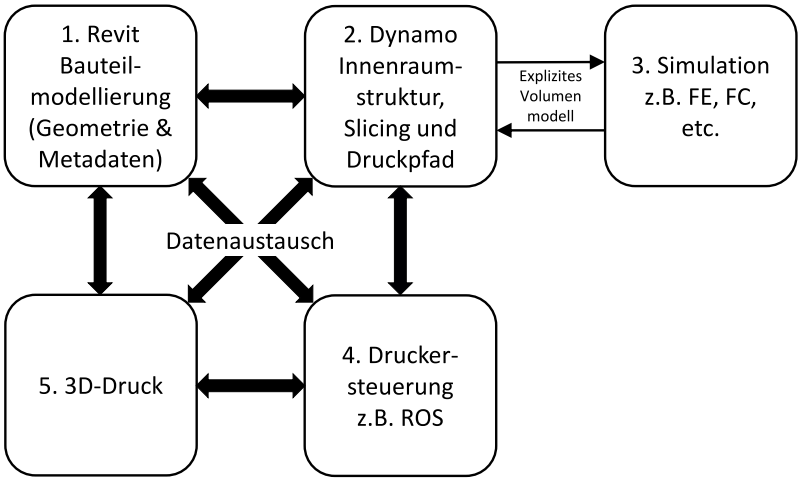


Abbildung 5.3: Datenaustauschszenario unter Anwendung der in dieser Arbeit vorgeschlagenen Methodik, FIM

te Konzept repräsentieren soll, so ist die prototypische Implementierung in diesem Kapitel jedoch noch komplett abhängig von der CAD-Software Revit und entsprechende Daten werden in dessen nativem Datenformat abgespeichert. Eine Speicherung der generierten in einem eigenen FIM-Datenformat wird in Kapitel 7 thematisiert.

Es gibt, wie in Kapitel 3 beschrieben, weitere Alternativen, die anstelle von Revit und Dynamo eingesetzt werden können. Die Software Rhino bietet mit dem Plug-In Grasshopper eine fast identische Umgebung wie Dynamo. Es wurden daher parallel zu der Bearbeitung mit Revit auch bestimmte Arbeitsschritte mit Rhino getestet, insbesondere da es für Grasshopper diverse Plug-Ins gibt, mit denen AM-Daten leicht erstellt und bearbeitet werden können. Die andere genannte Alternative, Siemens NX, ist zwar aufgrund seiner integrierten Simulationsmöglichkeiten allgemein eine gute Softwarewahl, jedoch ist das Programm weniger komfortabel beim Erstellen von automatisierten Pfadgenerierungsskripten. Mit der API von NX lässt sich zwar auch die Pfadplanung durchführen, jedoch werden unter Umständen alle Einzelschritte in der grafischen Benutzeroberfläche visualisiert. Bei größeren Bauteilen kann es sein, dass mehrere tausend Liniensegmente erzeugt werden müssen und daher wäre eine Visualisierung der Schritte nicht sinnvoll. Es bleibt aber noch zu untersuchen, ob es mit der API *NXOpen* nicht doch eine bessere Lösung gibt, allerdings wird das nicht Teil dieser Arbeit sein.

5.2 Beschreibung des implementierten FIM-Arbeitsablaufs

Der Planungsprozess eines ganzen Gebäudes ist ein iterativer Prozess, bei dem immer wieder neue Detaillierungsideen vorgeschlagen, geprüft und dann umgesetzt werden (vgl. Abschnitt 2.1.1). Die Modellierung der Fertigungsdaten für das AM-Verfahren wird wie zuvor beschrieben ganz ähnlich ablaufen. Abbildung 5.4 zeigt den implementierten FIM-Workflow mit dem innerhalb eines Dynamo-Skripts Fertigungsdaten für die Betonextrusionsmethode teilautomatisiert generiert werden können. Dieser Workflow stellt prinzipiell eine in Dynamo realisierte Version von Abb. 5.3 dar.

Der Ablauf des dargestellten Arbeitsablaufs stellt im Wesentlichen eine Dynamo Ausarbeitung des in Abb. 4.1 dargestellten Zusammenhangs der einzelnen FIM-Bausteine dar, jedoch abzüglich des digitalen Zwillings, der wie zuvor erwähnt nicht in dieser Arbeit behandelt wird. Ausgehend von dem BIM-Modell des zu druckenden Bauteils (blau hinterlegt im Bild) werden über ein Dynamo Skript (rot hinterlegt) **Prozessinformationen** in ein oder mehreren Varianten modelliert und daran anschließend je ein **explizites Volumenmodell** daraus abgeleitet, ebenfalls über ein Dynamo Skript. Die Prozessinformationen beinhalten hier zudem ein **Übersetzungsskript**, das entsprechend der generierten Daten Maschinencode ausgibt. Im Skript werden verschiedene Varianten teilautomatisiert über jeweils einen eigenen Codeblock pro Variante als Innenstruktur des Bauteils realisiert. Teilautomatisiert bedeutet in diesem Zusammenhang, dass die verschiedenen Innenstrukturen durch gewählte Parameter weiter angepasst werden können und auch sonst Eingriffe durch den

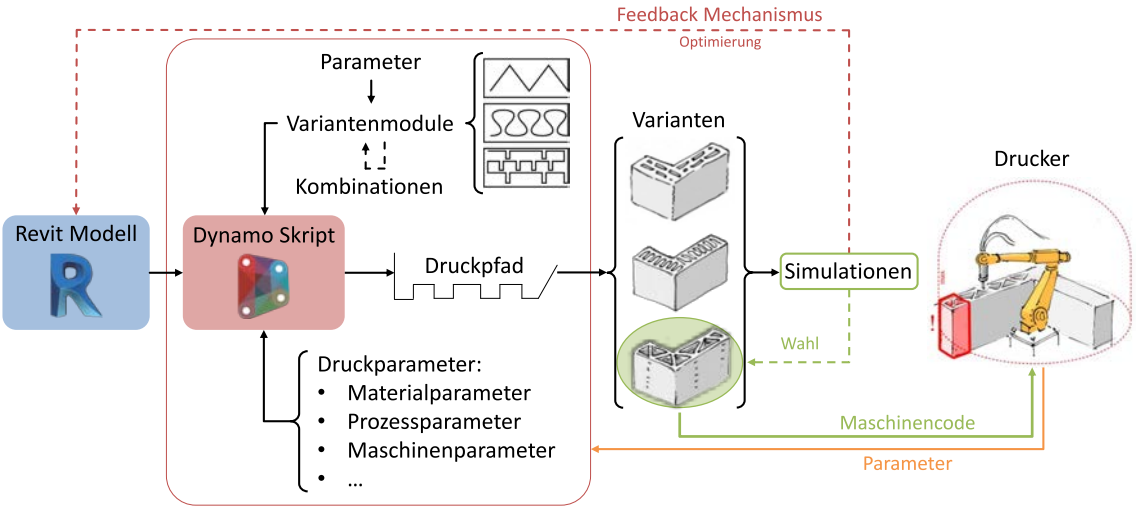


Abbildung 5.4: Arbeitsablauf für die Erstellung eines FIM.

Designer nötig sein können. Was die verschiedenen Varianten bedeuten, wird bei der Parameteranalyse im Folgenden nochmals genauer definiert.

Der im Flussdiagramm dargestellte Feedback Mechanismus und das simulationsgestützte Auswählen einer Variante sind gestrichelt dargestellt, da hier noch keine Validierung der Funktionsweise stattgefunden hat. Bisher wurden nur einfache Beispiele bearbeitet, bei denen lediglich ein Iterationsschritt notwendig war für die Erstellung eines funktionierenden Fertigungsdatensatzes und demnach keine Simulationsdaten nötig waren. Komplexere Beispiele werden aber im Anschluss an diese Arbeit untersucht, um den Feedback Mechanismus zu validieren. Im Kapitel 7 wird darauf aufbauend konzeptionell ausgeführt, wie diese gesammelten Daten in einem passenden Datenformat dargestellt werden können.

5.3 Planung der Prozessinformationen

Im Folgenden wird beschrieben, welche Informationen für den Anwendungsfall Betonextrusion einer Wand notwendig sind, um Fertigungsinformationen zu erstellen und wie diese Erstellung durchgeführt wird. Dazu wird zunächst eine Literaturrecherche durchgeführt, um die Fähigkeiten der Betonextrusion genau zu untersuchen und zu entscheiden, wie viele Informationen zur Verarbeitung mit dieser Methode benötigt werden, die ausgearbeitete Liste wurde bereits in Tabelle 4.1 gegeben. Dann wird ein Konzept aufgestellt, wie ein Druckpfad (teil-)automatisch generiert und in den BIM-Workflow eingebaut werden kann. Und schließlich wird die Implementierung der Pfadgenerierung durchgeführt und mittels Tonextrusion – ausgeführt durch einen UR10e Roboterarm – validiert.

5.3.1 Druckparameter

Bei jeglichem Druckverfahren können die beteiligten Parameter in drei Kategorien eingeteilt werden, die Materialparameter, Prozessparameter und Maschinenparameter. Alle Parameter zusammen definieren, wie und mit welchen Einschränkungen das Objekt

gestaltet werden kann. Im Folgenden werden die wichtigsten Parameter der einzelnen Bereiche vorgestellt, die einen Einfluss auf den Druckkörper haben (vgl. Tabelle 4.1).

Maschinenparameter

Die Parameter, die den Roboter beschreiben, also den Bewegungsapparat, haben einen eher indirekten Einfluss auf das Druckobjekt. Sie beschreiben Limits, z.B. wie groß das Objekt sein kann (Bauraum) oder wie schnell es gedruckt werden kann. Mit den *Achsmaßen* und der *maximalen Auslenkungen* des Roboters wird hierbei der maximal mögliche Bauraum definiert und mit der maximalen Achsgeschwindigkeit und -beschleunigung die maximale Druckkopfgeschwindigkeit.

Die Druckkopfparameter haben dagegen einen sehr aktiven Einfluss auf die realisierbare Form des Druckobjektes. Die *Düsenform* ist unter anderem für die Filamentbreite des Extrusionskörpers und die maximale Schichthöhe verantwortlich, aus Schichthöhe, Düsenform und Materialeigenschaften setzt sich in einem komplexen Zusammenhang der Querschnitt des Filaments zusammen. Über den Querschnitt und die Schichthöhe wird außerdem die Oberflächenbeschaffenheit beschrieben, also das Aussehen der für Extrusionsmethoden charakteristischen Rillenstruktur. Die *maximale Förderrate* beschränkt hierbei nur die Düsengeschwindigkeit, wird diese Geschwindigkeit nicht auf die Förderrate angepasst, kann dies einen sehr starken Einfluss auf den Extrusionsquerschnitt haben sowie auf den Schichtenverbund.

Materialparameter

Die Materialparameter beschreiben ohne Frage die physikalischen Eigenschaften des fertigen Druckobjekts, aber zudem setzen sie Grenzen auch für den Druckprozess selbst. Die rheologischen Eigenschaften des Betons in frischem Zustand sind hierbei besonders wichtig. Damit der Beton pumpfähig ist, muss das Material über eine niedrige Schergrenze und eine plastische Viskosität in der Gleitschicht verfügen. Um ein gleichmäßiges Auftragen des Betons zu ermöglichen, ist dagegen eine niedrige Viskosität und eine hohe Schergrenze erforderlich. Für eine frühe Belastbarkeit ist eine sehr hohe statische Schergrenze und ausgeprägte Thixotropie notwendig, damit Betonschichten aufeinandergestapelt werden können, ohne dass dabei starke Verformungen auftreten. Von diesen Materialparametern sind also drei wichtige Eigenschaften abgeleitet, die **Pumpbarkeit**, die **Extrudierbarkeit** und die **Verbaubarkeit**. (MECHTCHERINE & NERELLA, 2019)

Eine besonders wichtige Eigenschaft des Betons ist die *Aushärtezeit*, denn sie bestimmt, wie schnell das Material verarbeitet werden muss. Zusammen mit der Festigkeit gibt die Aushärtezeit außerdem vor, wie hoch gedruckt werden kann, ohne dass starke Verformungen auftreten. Alle diese Eigenschaften werden jedoch nicht eingehend in dieser Arbeit untersucht, genauere Betrachtungen hierzu werden jedoch von verschiedenen Forschungsprojekten bearbeitet.

Prozessparameter

Die Prozessparameter werden bei der Modellierung des Fertigungsmodells definiert, der hierzu mögliche Wertebereich wird durch Material- und Maschinenparameter vorgegeben. Weitere Einschränkungen oder Anforderungen an diese Parameter können aber auch durch die Geometrie des zu erzeugenden Objekts hinzukommen. Das optimale Einstellen dieser Parameter ist also eine der Hauptaufgaben bei der Generierung eines Fertigungsmodells. Das schwierige an dieser Aufgabe ist jedoch, dass die Parameter sich auf sehr komplizierte Weise gegenseitig beeinflussen und manche Konfigurationen eine genaue Machbarkeitsuntersuchung erfordern. Ein allgemeines „Rezept“ zum Abstimmen dieser Werte, das für alle eventuellen Feinheiten anwendbar ist, lässt sich derzeit daher nur schwer aufstellen.

Innenstrukturen

Die Ausgestaltung der inneren Struktur eines Bauteils stellt im eigentlichen Sinn weniger einen Parameter dar, sondern eine Strategie, wie der Raum zu füllen ist. Im Rahmen dieser Arbeit wird wie in Abschnitt 2.6 beschrieben der Fokus auf Wandsegmente gelegt, die zur Wärmeisolation funktionsaktiviert sind. Eine Möglichkeit zur Steigerung der Isolationsfähigkeit einer Wand ist es, Hohlräume im Inneren der Wand zu erzeugen. Da aus strukturellen Gründen und aufgrund einer nötigen Gleichmäßigkeit der Isolation die Hohlräume nicht einfach chaotisch im Inneren der Wand angeordnet werden dürfen, ist hier für die entsprechende Planung eine bestimmte **Strategie** notwendig. Für die Pfadplanung des Bauteildrucks ist es in diesem Zusammenhang notwendig, eine Füllstrategie für die Innenfläche jeder Druckschicht zu definieren.

Es ist hier unter Umständen möglich, mittels Simulation eine optimale Innenstruktur zu finden und im Anschluss dazu einen Druckpfad abzuleiten. Jedoch birgt dies die Schwierigkeit, dass die gefundene Innenstruktur gegebenenfalls nicht mit der entsprechenden AM-Methode druckbar oder das Ableiten des entsprechenden Druckpfads nur schwer zu realisieren ist. Diese Ingenieuraufgabe (die Erzeugung einer speziellen Innenstruktur) direkt zu automatisieren ist zwar generell möglich, aber erscheint derzeit noch nicht durchführbar zu sein, da hierzu noch viele Erkenntnisse über die genauen Fähigkeiten der einzelnen AM-Methoden fehlen. Die umgekehrte Herangehensweise, also der Vorgabe von verschiedenen Druckpfaden (Vorschlag verschiedener Varianten) und deren anschließender Optimierung, erscheint hier aktuell der praktikablere Lösungsweg zu sein. Eine automatische Gestaltung vorgegebener „Patterns“ ist im Vergleich zur insgesamten automatischen Planung aber durchaus denkbar, wenn auch immer noch nicht ganz einfach. Nicht weil eine entsprechende Planung des Druckpfads aufwendig ist, sondern weil die geforderte Effektivität der Innenstruktur nur schwer vorausgeplant werden kann. Genau hier ist eine separate Simulation notwendig, um verschiedene Strategien miteinander zu vergleichen oder die Effektivität einer Strategie durch Optimierung zu steigern. Wenn im Folgenden von Innenstrukturen gesprochen wird, dann sind eben beschriebene **Füllpatterns** gemeint, die automatisch generiert werden können.

Ganz analog zu der Füllstrategie zur Steigerung der Wärmeisolation sind auch andere Strategien für andere Zwecke denkbar. Z.B. können Aussparungen für Geräteinstallationen oder Kabelleitungen direkt in das Bauteil integriert werden. Zudem können wie schon beschrieben verschiedene Strategien für denselben Zweck eingesetzt werden, je nachdem, welche Strategie sich besonders für die entsprechende Situation eignet. Und schließlich ist es denkbar, dass verschiedene Strategien für eine mehrfache Funktionsaktivierung oder einfach zur Steigerung der Effizienz miteinander kombiniert werden. Wie dies im Detail aussehen kann, wird bei der Beschreibung der Implementierung (Abschnitt 5.5) gezeigt. Eine Bewertung der Leistungsfähigkeit der gezeigten Innenstrukturen wird in dieser Arbeit jedoch nicht durchgeführt, könnte aber möglicherweise in nachfolgenden Arbeiten – zusammen mit der Validierung des Feedback Mechanismuses (vgl. Abschnitt 5.2) – thematisiert werden.

5.3.2 Pfadplanung und Parameterdefinition

Einige der beschriebenen Parameter werden durch die Wahl des Drucksystems, der Bewegungsmaschine, des Materials und besonders wichtig der zu druckenden Geometrie durch gewisse Designentscheidungen vorgegeben. In einem aktuellen Forschungsprojekt wird hierzu untersucht, ob ein **Design Decision Support System (DDSS)** entwickelt werden kann, dass bei dieser Aufgabe Unterstützung bieten kann. Auch wichtig ist an dieser Stelle schon, wie die Innenstruktur des Bauteils aussehen soll. Hier soll ebenfalls je nach Problemstellung das auf einer Wissensdatenbank beruhende DDSS unterstützend eine Vorauswahl der zu planenden Innenstrukturen treffen (Variantenanalyse folgt).

Je nach Bauteilart und dessen Design können hier ganz unterschiedliche Fertigungssysteme sinnvoll sein. Wie aber in Abschnitt 2.6 bereits erwähnt, liegt der Fokus hier auf der Extrusionsdruckmethode für den Bau einer Wand mit integrierter Wärmeisolation. Für diese Aufgabe werden in dieser Arbeit Materialparameter angenommen, die für diese Methode anwendbar sind – genaue Werte sind für diese Arbeit jedoch nicht notwendig – und Maschinenparameter entsprechend eines Armroboters mit Extrusionsaufsatz. Demnach sind nur noch die Prozessparameter einzustellen.

Eine mögliche Herangehensweise zur Definition der übrigen Parameter und zur Erstellung eines Fertigungsmodells ist wie folgt:

1. Es werden drei der offenen Parameter (Düsenform, -orientierung und Schichthöhe) gewählt, verschiedene Innenstrukturen vorgegeben und mit damit der Druckpfad für den Bau der vorgegebenen Geometrie abgeleitet.
2. Die anderen noch offenen Parameter werden dann an diese geometrische Voraussetzung angepasst und mit den anderen schon definierten Parametern möglichst gut abgestimmt.

3. Falls es zu Konflikten mit den vorgegebenen Parametern kommen sollte, die in dieser Gestalt nicht lösbar sind, muss eine neue Iteration (zurück zu Schritt 1) mit einer neuen (besseren) Wahl der offenen Parameter gestartet werden.

4. Es ist in der Regel empfehlenswert, über geeignete Simulationen die Durchführbarkeit des Druckvorgangs zu prüfen. Für eine reine Abschätzung, ob der Druckpfad sich vollständig innerhalb des Bauraums befindet und keine kritische Position (Singularität, vgl. Abschnitt 2.4.2) angesteuert wird, ist hierbei allein der Druckpfad als Input ausreichend. Sollen aber zusätzlich Kollisionsanalysen mit dem Druckobjekt durchgeführt werden, dann ist unter Umständen schon ein explizites Volumenmodell notwendig. Mit diesem 3D-Modell ist es aber zusätzlich möglich, dass die Eigenschaften des fertigen Objekts vorab abgeschätzt werden. Sollte es in einer dieser Analysen Probleme geben, muss die Iteration wiederholt werden, bis ein zufriedenstellendes Ergebnis erzielt wird.

Mit den Parametern Düsenform, -orientierung und Schichthöhe wird in einem ersten Schritt der **Querschnitt des Extrusionsfilaments** abgeschätzt. Eine genaue Berechnung wäre im Rahmen dieser Arbeit nicht sinnvoll, da die dazu nötigen physikalischen Gesetze zu komplex sind und der Nutzen diesen Aufwand nicht rechtfertigt. Eine spätere Erweiterung um ein System, dass genau vorhersagen kann, in welcher Form das Filament gedruckt wird, ist aber denkbar. Bei Verwendung einer runden Düse mit einer Ausrichtung senkrecht zur Bewegungsrichtung kann hier unter bestimmten Voraussetzungen – die hier zunächst als gegeben erachtet werden – ein Rechteck mit abgerundeten Ecken angenommen werden. Die Breite dieses Rechtecks ist vom Düsendurchmesser und dessen Höhe von der Schichthöhe abzuleiten. Über den so definierten Filamentquerschnitt lässt sich der Mindestabstand von nahe liegenden Pfadsegmenten festlegen, in horizontaler (Filamentbreite) und in vertikaler (Filamenthöhe) Richtung. Sind beide Parameter und die Geometrie gegeben, muss also nur jeweils ein Pfad für das Bauteil geplant werden, entsprechend der verschiedenen Innenstrukturen. Im Folgenden wird hier der Einfachheit halber nur von einem einzigen Pfad ausgegangen.

Sobald der Druckpfad verfügbar ist, lassen sich weitere Informationen ableiten bzw. einstellen. Zuerst ist hier die **Schichtgeschwindigkeit** (vgl. Gleichung (5.1)) zu nennen. Mit diesem abgeleiteten Parameter kann eingestellt werden, in welcher Zeit Schichten überdruckt werden und ist damit von besonderer Bedeutung. Für einen erfolgreichen Druck und ein möglichst gutes Ergebnis ist es wichtig, das Überlagern mehrerer Schichten in einem bestimmten Zeitfenster zu halten. Je nach Aushärtezeit und **Frühfestigkeit** des Betons muss dies geschickt eingestellt werden. Geschieht der Druckprozess **zu schnell**, so kann es sein, dass der Beton nicht ausreichend tragfähig ist, um überlagerte Schichten tragen zu können. In einem solchen Fall tritt **Materialversagen** in den unteren Schichten ein und das Gebilde fällt in sich zusammen (MECHTCHERINE et al., 2020; SUIKER et al., 2020). Wird dagegen **zu langsam** gedruckt, so härtet der Beton zu stark aus, bevor er mit einer weiteren Schicht überdruckt wird und es kann kein optimaler **Schichtenverbund** erreicht werden (WOLFS et al., 2019). Wird die Schichtgeschwindigkeit genau richtig eingestellt, ist beim Überlagern zweier Schichten der Beton auf beiden Seiten genügend

viskos, dass an den Grenzflächen etwas Durchdringung stattfindet, wodurch eine bessere Haftung erreicht wird. (JEONG et al., 2019)

$$v_L = \frac{\bar{v}_{nozzle}}{l_L}$$

mit

v_L : Schichtgeschwindigkeit [Schichten/Sekunde]
l_L : Pfadlänge pro Schicht [m]
$\bar{v}_{nozzle}$: mittlere Düsengeschwindikeit entlang des Druckpfades [m/s]

(5.1)

Gleichung (5.1) gibt an, aus welchen Grundparametern die Schichtgeschwindigkeit abgeleitet wird. Um diese Geschwindigkeit zu verringern, damit z.B. kein Materialversagen auftritt, kann entweder die Pfadlänge vergrößert werden oder die Düsengeschwindigkeit verringert werden, genauso natürlich umgekehrt, wenn die Geschwindigkeit vergrößert werden soll. Da die Düsengeschwindigkeit durch die maximalen Maschinenwerte begrenzt ist, ist damit auch die Größe des druckbaren Objekts weiter begrenzt. Ist z.B. die Düsengröße klein, dann müssen sehr lange Wege pro Schicht gefahren werden, sodass möglicherweise trotz Maximalgeschwindigkeit des Roboters die Mindestschichtgeschwindigkeit nicht erreicht wird, um den gesamten Bauraum der Maschine auszunutzen.

Wird dieselbe Betonzusammensetzung über den gesamten Druckprozess verwendet, ist es in der Regel sinnvoll, die Schichtgeschwindigkeit konstant zu halten, um gleichmäßige Verhältnisse über die gesamte Bauteilhöhe sicherzustellen. Jedoch kann es aber sein, dass bei bestimmten Geometrien ein anderes Vorgehen erforderlich ist, wie z.B. beim 3D-Druck einer Kuppel oder ähnlichen geschlossenen Strukturen. Eine Untersuchung von CARNEAU et al. (2019) hat gezeigt, dass gerade ein unterschiedlich schneller Schichtendruck bestimmte Geometrien druckbar macht (vgl. Abb. 5.5). Je mehr die Oberfläche der Kuppel in diesem Beispiel geneigt ist, desto mehr Überhang hat die entsprechende Schicht zur vorherigen und desto mehr Spannung wirkt auf das Material in diesem Bereich. In den ringförmigen Schichten dieser Kuppel kann aber das thixotrope Verhalten von Beton ausgenutzt werden. Da ein geschlossener Ring sich selbst stabilisiert, muss hier möglichst schnell der instabile offene Ring zur geschlossenen Form gedruckt werden, und dies umso schneller, je größer der Überhang ist. Bei gleichbleibender Düsengeschwindigkeit ist

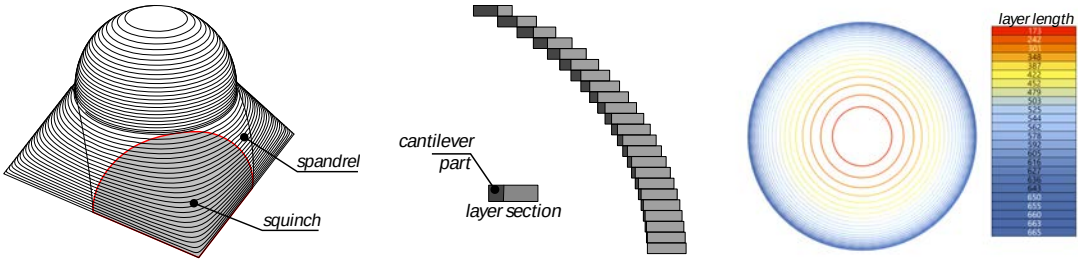


Abbildung 5.5: *links:* Druckpfad für den Betonextrusionsdruck einer Kuppel. *mitte:* Überhang der einzelnen Druckstränge je nach Position in der Kuppel. *rechts:* Weglänge der einzelnen Schichten. (CARNEAU et al., 2019)

das in diesem Beispiel genau der Fall, desto kürzer der Druckpfad pro Schicht wird umso schneller ist die Schichtgeschwindigkeit (vgl. Abb. 5.5 rechts), jedoch kann es auch nötig sein, die Düsengeschwindigkeit innerhalb einer Schicht mehrfach anzupassen. Diese Art von Ingenieurwissen ist aber nur schwer in der automatisierten Pfaderstellung integrierbar, bzw. gerade in solchen Fällen muss immer mittels Simulation und gegebenenfalls mit Modelldruck auf Druckbarkeit geprüft werden. Im Rahmen dieser Arbeit wird aber von solchen Spezialfällen abgesehen.

Wie zuvor erwähnt, gibt es für den beschriebenen Filamentquerschnitt gewisse Voraussetzungen. Zum einen ist eine sinnvoll gewählte Schichthöhe und zum anderen eine gute Abstimmung der beiden Parameter Düsengeschwindigkeit und Extrusionsrate nötig, um keine unerwünschten Formen zu erhalten. Zieht man den zuvor angenommenen Querschnitt entlang des Druckpfades mit der Düsengeschwindigkeit, so wird in einer gewissen Zeit ein bestimmtes Volumen eingeschlossen. Genau dieses Volumen muss in der entsprechenden Zeit mit Beton aufgefüllt werden, somit ergibt die entsprechende Extrusionsrate. Außerdem ist darauf zu achten, dass die **Schichthöhe** nicht größer als die Düsenbreite gewählt wird, da sonst keine guten Ergebnisse beim Ablegen des Filaments erzielt werden. TAKAHASHI und MIYASHITA (2017) haben hierzu eine Untersuchung für die 3D-Druckmethode FDM durchgeführt und dabei verschiedene Messungen mit unterschiedlichen Düsenpositionen bei verschiedenen Extrusionsraten unternommen. In Abb. 5.6 wird das zur Visualisierung verwendete Diagramm erklärt und in Abb. 5.7 werden die Messergebnisse präsentiert.

Die dargestellten Diagramme wurden hierbei in Versuchen mit Kunststoffen erzeugt und treffen nur bedingt für den Werkstoff Beton zu. Das mittlere Diagramm zeichnet aber im Einklang mit Erfahrungsberichten eines anderen Forschungsprojekts ein recht aussagekräftiges Bild. Angestrebt wird die *„stable road“*, im oberen stabilen Bereich sorgt zudem die erhöhte Extrusionsrate für einen stärkeren Druck auf die unterliegende Schicht,

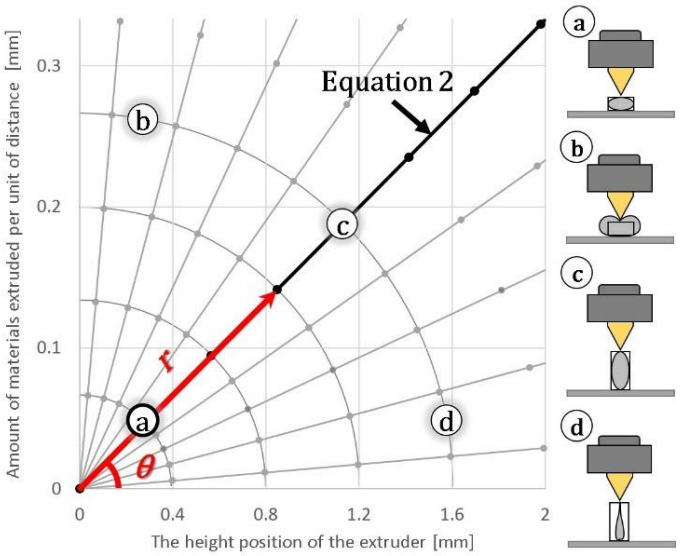


Abbildung 5.6: Düsenhöhe zu Extrusionsrate Diagramm. Erklärung nach (TAKAHASHI & MIYASHITA, 2017).

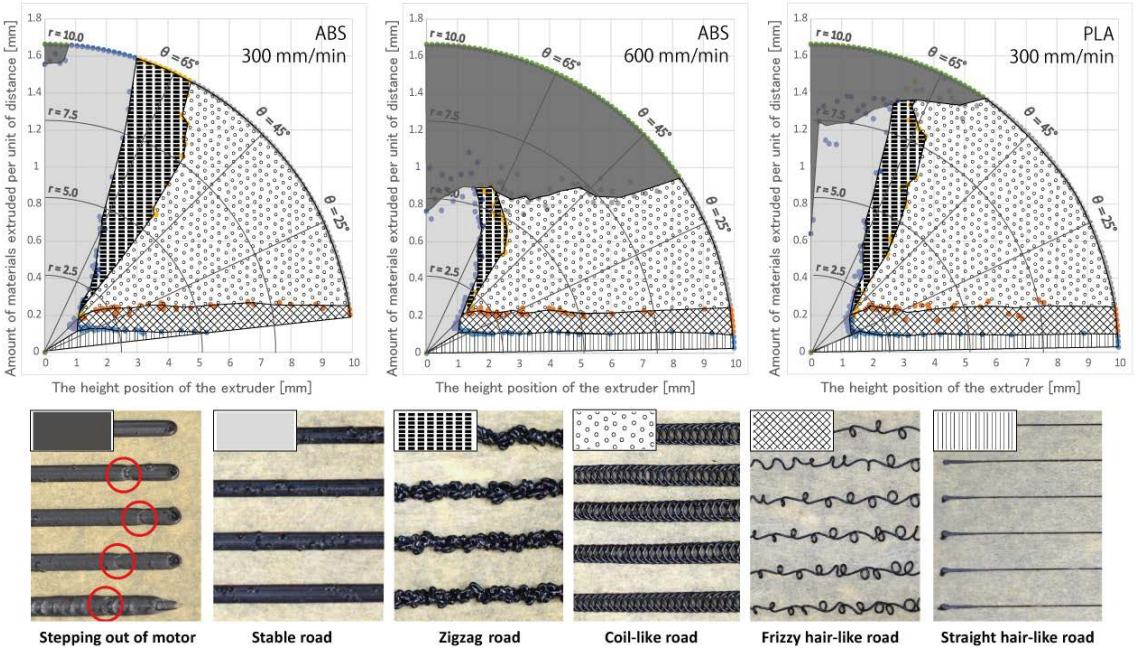


Abbildung 5.7: Messergebnisse von Druckversuchen mit verschiedenen Parameterkombinationen. (TAKAHASHI & MIYASHITA, 2017)

wodurch der Flächenverbund begünstigt wird. Zu viel Material kann allerdings zu unerwünschten Verformungen auch in tieferen Schichten führen. Die anderen Druckspuren sind – abgesehen von eventuell künstlerischem Interesse – nicht wünschenswert, da vermutlich die strukturelle Stabilität stark beeinträchtigt sein wird.

Als letzter noch offener Parameter aus Tabelle 4.1 muss noch die Bezugsebene genannt werden. In einem regulären Druckprozess ist üblicherweise eine horizontale Bezugsebene eingestellt, der Drucktisch. Es ist allerdings auch möglich, dass der Extrusionsdruck auf eine nicht horizontale Bezugsebene ausgeführt wird (vgl. Abb. 2.24). Besonders beim Druck von Überhängen kann dies ein nützliches Werkzeug sein, da auf diese Weise stärkere Neigungen realisiert werden können (CARNEAU et al., 2019).

5.3.3 Pfad- und Prozessoptimierung

Selbst mit gut abgestimmten Druckparametern ist es möglich, dass ein geplanter Pfad nicht möglich ist oder zu einem schlechten Ergebnis führt. Wenn z.B. bei sehr schmalem Filamentquerschnitt (kleine Düse) einreihig und geradlinig in die Höhe gedruckt wird, kann es zum elastischen Knickverhalten kommen (BUSWELL et al., 2018; MECHTCHERINE et al., 2020). Bei der Planung solcher Elemente ist eventuell abzuwägen, ob solche Geometrien nicht verstärkt werden müssen. BUSWELL et al. (2018) zeigen, dass ein paralleler Druck mehrerer Filamente diese Instabilität verringert (vgl. Abb. 5.8b). Zudem ist in Abb. 5.8a zu sehen, dass dieses Problem an den Rändern kaum auftritt. Krumme Pfade sind demnach stabiler gegenüber elastischem Knickversagen, da hier bei diesem Versagensfall auch die Form in horizontaler Richtung verformt werden muss, um ein Umknicken zu bewirken.

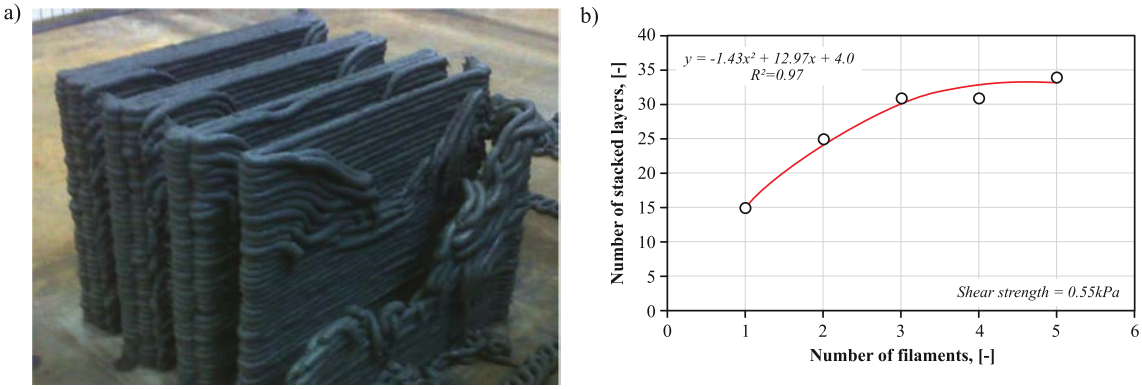


Abbildung 5.8: Baubarkeitsanalyse geradlinig übereinander gedruckter Filamente in Abhängigkeit der Anzahl parallel gedruckter Filamente. (BUSWELL et al., 2018)

Ein weiteres Problem können starke oder zu viele Richtungsänderungen darstellen. Bei Extrusionsverfahren wird aus technischen Gründen beim Drucken einer Ecke die Außenseite abgerundet – genau umgekehrt im Vergleich zu subtraktiven Verfahren, bei denen die Innenseite abgerundet ist. Werden in einem Druckpfad so vier Ecken zueinander ausgerichtet, so entsteht hier ein Hohlraum in der Mitte (vgl. Abb. 5.9 und 5.10). Dieser Effekt wird noch zusätzlich verstärkt, wenn das Druckmaterial nicht viskos genug ist, um starke Richtungsänderungen zu realisieren und dadurch an Ecken „aufreißt“.

Besonders eindrucksvoll ist dieses Problem anhand einer mit dem SC3DP-Verfahren gedruckten Wandecke (vgl. Abb. 5.11). In diesem Fall sind besonders an Positionen, an denen das Werkzeug eine komplette Kehrtwende oder eine harte Richtungsänderung einschlägt, Fehlstellen erkennbar. Die Enden (Kehrtwende) sind aufgrund der geringeren Verweilzeit des Druckkopfes abgesenkt und die Ecke aufgrund von Überschneidung der Druckbahn innen erhöht.

Bei der Pfadplanung sollte daher auf dieses Problem acht gegeben werden und wenn möglich auf häufige Richtungswechsel und spitze Winkel an Knicken verzichtet werden. Eine weitere Möglichkeit dieses Problem abzufedern ist, die Extrusions- und/oder Druckkopfgeschwindigkeit an diese geometrische Voraussetzung anzupassen. Jedoch ist eine solche Anpassung noch nicht genau untersucht worden und möglicherweise ist es nicht möglich, dies für alle möglichen Problemfälle reproduzierbar einzustellen, da hier zu einem Teil auch Materialparameter mitwirken.

In einigen Fällen kann sicherlich mit gewissen Grenzwerten der Erfolg oder Misserfolg eines geplanten Druckprozesses vorhergesagt werden. Beispielsweise geben Materialparameter einen gewissen Aufschluss darauf, welche mechanischen Eigenschaften das Material nach dem Ablegen haben wird. So kann bereits vorhergesagt werden, dass ein Überhang eventuell zu steil designet worden ist und daher mit der geplanten Betonmischung nicht durchführbar ist. Jedoch geht dies nicht immer so einfach und kann zudem bei großen Objekten und hunderten Schichten leicht übersehen werden. In der Regel empfiehlt es sich vor dem Druckvorgang in einer Computersimulation zu verifizieren, dass es zu keinen Problemen kommen wird. Ein Toolkit hierfür liefert z.B. ROS oder in einfacherer Form das **Robots** Plug-In von *MSD Robotics Lab* für *Rhino/Grasshopper*. In

Abbildung 5.9: Druckkörper mit mehreren Richtungswechsel und daraus entstehenden Fehlstellen. (BUSWELL et al., 2018)

Abbildung 5.10: Beispiel-Druckkörper in guter (rechts) und schlechter (links) Qualität. (BUSWELL et al., 2018)

Abbildung 5.11: Shotcrete-Bauteil mit Fehlstellen an den Enden und der Ecke.

beiden Simulationstools kann mit Kollisionsobjekten über den gesamten Druckprozess geprüft werden, ob der Roboter sich kollisionsfrei bewegt, und zudem der gesamte Prozess als Animation durchgespielt werden. Es werden hierfür nur der Druckpfad mit den entsprechenden Parametern und ein 3D-Modell der Umgebung des Roboters benötigt, Modelle von verschiedenen Robotern können online im URDF-Format – im Wesentlichen eine XML und eine 3D-CAD-Datei, die Form und Eigenschaften des Roboters beschreiben – frei heruntergeladen werden.

Zum Nachweis bzw. Vorhersage, ob der Druckpfad zufriedenstellende Ergebnisse liefern wird, kann mithilfe des Druckpfads, wie in Abschnitt 4.3 konzeptionell beschrieben, ein Volumenmodell erzeugt werden. Wie das in dem hier präsentierten Dynamo-Skript realisiert werden kann, wird im Folgenden beschrieben. Da dies allerdings optional ist und nicht zwingend für einen erfolgreichen Druck erfordert wird, kann dieser Punkt auch übersprungen werden und direkt Maschinensteuerungscode erzeugt werden. Wie der Maschinensteuerungcode durchgeführt werden kann, wurde bereits in Abschnitt 2.4.1 beschrieben und nochmals in Kapitel 4 behandelt. Eine detaillierte Beschreibung, wie dieser

Übersetzungsprozess mittels dem Dynamo-Skript für die Roboterprogrammiersprache *URScript* funktioniert, wird in Abschnitt 5.7 beschrieben.

5.4 Erstellung eines expliziten Volumenmodells

Für Simulationen, die Informationen über die genaue Materialverteilung erfordern, ist zudem die 3D-Geometrie des Druckobjektes notwendig, also die Soll-Geometrie. Solche Simulationen sind aber nur unter Umständen für die Planung der Fertigungsinformationen erforderlich (siehe Beschreibung in Abschnitt 4.1.2). Ein Fall, in dem solche Modelle eingesetzt werden müssen, ist unter anderem, wenn eine Variantenanalyse durchgeführt werden soll – z.B. bei einer zur Wärmeisolation aktivierten Wand.

Diese Geometrie kann hierzu aus den festgelegten Parametern und dem Druckpfad abgeleitet werden und in einem sinnvollen Format exportiert oder besser noch direkt im FIM abgespeichert werden. Eine Möglichkeit, die in dieser Arbeit untersucht wird, ist die Sweep-Operation, mit der anhand eines Querschnitts und eines Pfads ein Volumenkörper erstellt werden kann. Sowohl Querschnitt als auch Pfad liegen nach Abfolge der in Abschnitt 5.3.2 beschriebenen Schritte vor und können hierzu direkt verwendet werden.

5.5 Implementierung der Pfadgenerierung

Um die bisherigen Beschreibungen der in dieser Arbeit vorgeschlagenen Methodik – das Fabrication Information Modelling (FIM) – zu validieren, wurde die prototypische Implementierung für das BIM-Tool durchgeführt. In diesem Abschnitt werden alle hierzu vorgenommenen Schritte erklärt. Ausgangspunkt und Testobjekte für die Implementierung sind mehrere Wandsegmente unterschiedlicher Geometrie. Diese werden im BIM-Tool Revit vordefiniert und später jeweils an das Pfadgenerierungstool übergeben. Abbildung 5.12 zeigt 6 Wandobjekte, die zum Test der Pfadgenerierung gezeichnet wurden, dies sind eine reguläre Ecke, eine doppelte Ecke, eine kreisförmige Wand, eine geneigte Freiformwand, eine geneigte Ecke und eine senkrechte Freiformwand.

Für die weitere Bearbeitung der Implementierung wurde der Einfachheit halber hauptsächlich in Revit über das Plug-In Dynamo weitergearbeitet. Aufgrund der sehr großen Ähnlichkeit zwischen den beiden visuellen Programmierschnittstellen Dynamo und Grasshopper (vgl. Abschnitte 3.1 und 3.2) wird im Folgenden nur die Implementierung in Dynamo durchgeführt und erläutert. Lediglich die einleitenden Schritte, wie ein Projekt in Rhino aufgesetzt wird, werden kurz dargestellt. Die Software Siemens NX (vgl. Abschnitt 3.3) wurde aus Zeitgründen nur wenig untersucht, aber auch hier werden die generellen Schritte zum Erstellen eines Projekts kurz beschrieben.

Der erste Schritt bei der Verwendung von **NX** ist zunächst die Konvertierung der BIM-Daten von Revit in ein Siemens NX kompatibles Format, hierzu eignet sich z.B. das ACIS-Format `.sat`. Damit sind die wichtigsten Informationen in NX verfügbar. Um fortzufahren, muss

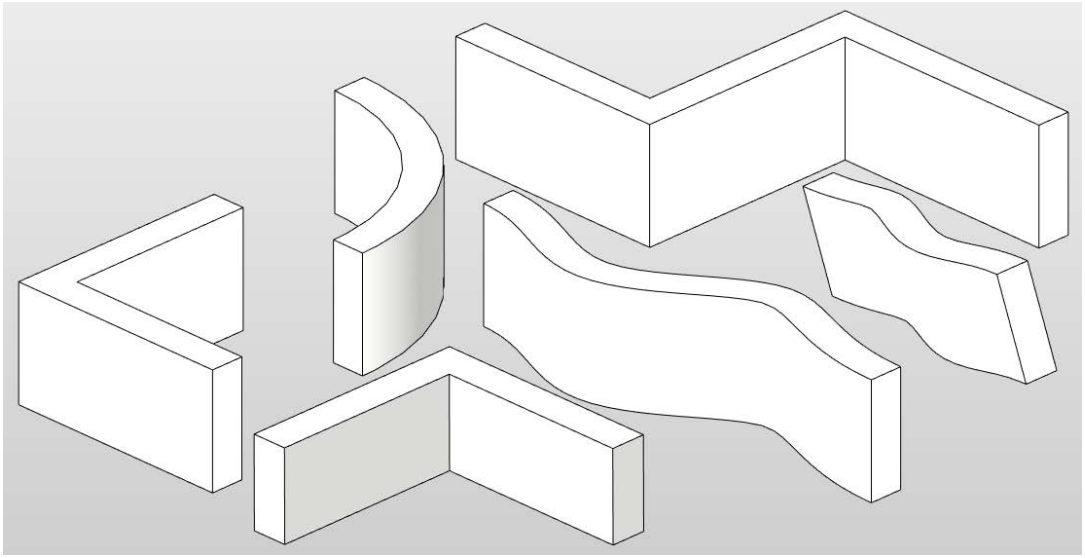


Abbildung 5.12: In Revit erstellte Beispielobjekte zur Pfadgenerierung.

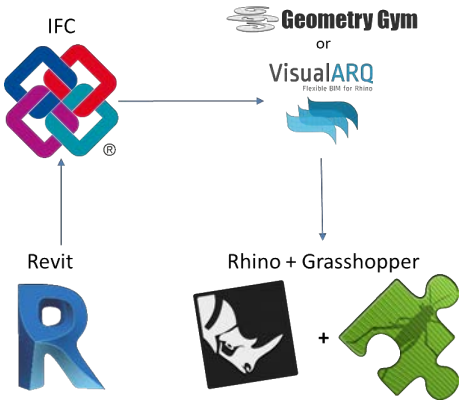


Abbildung 5.13: Übertragungsweg eines BIM-Models von Revit zu Rhino.

ein *Visual Studio* Projekt, je nach Vorliebe in der Programmiersprache C#, VB oder C++, angelegt werden und die entsprechenden *NXOpen* Bibliotheken verlinkt werden. Über diese API-Schnittstelle lässt sich so der komplette Funktionsumfang von NX steuern.

In gleicher Weise lässt sich dieser Prozess für die Softwarelösung **Rhinoceros3D** durchspielen. Hierzu muss das BIM-Modell ausgehend von Revit in das Austauschformat IFC konvertiert werden, die IFC-Datei wird dann mithilfe des Plug-Ins *VisualARQ* oder alternativ mit GeometryGym in Rhino importiert. Das Pfadgenerierungstool kann nun in Rhino mit der visuellen Programmierschnittstelle *Grasshopper* erstellt werden (vgl. Abb. 5.13).

Vorbereitung

Für die Verwendung der Beispielwände in **Dynamo** muss nichts weiter getan werden, als das Plug-In zu starten, während die Beispieldatei in Revit geöffnet ist. In Dynamo kann auf alle Elemente in der im Hintergrund geöffneten Beispieldatei zugegriffen werden, die Wandelemente sind hierbei alle in Ebene 0 eingezeichnet. Zunächst wird also über den

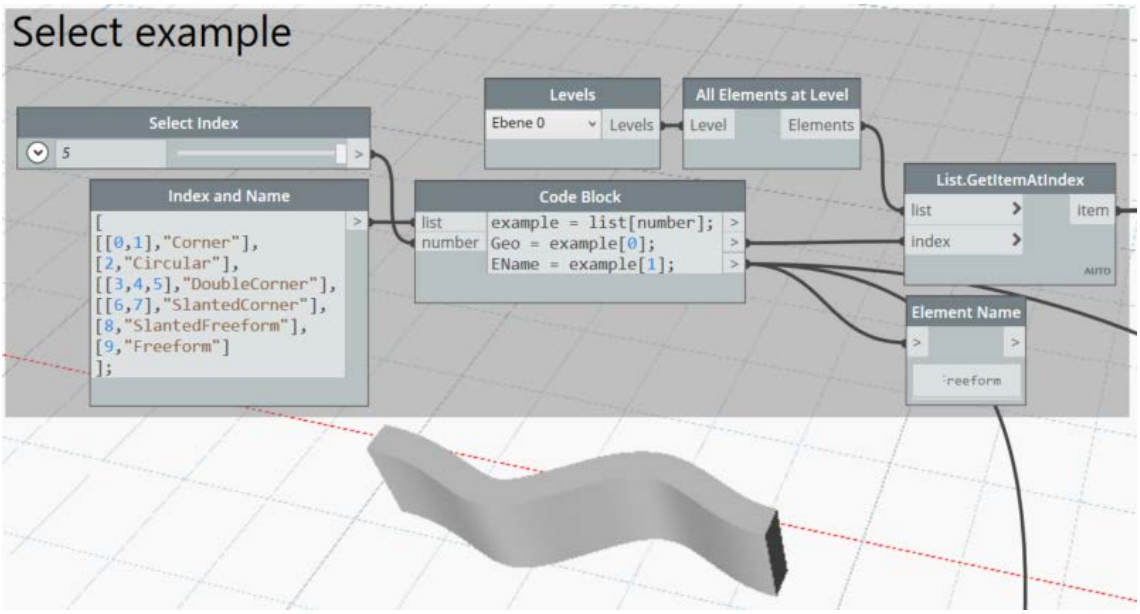


Abbildung 5.14: Übertragung der Beispiel Bauteile in den Dynamo-Workspace.

Programmknoten `All Elements at Level` auf alle Elemente in Ebene 0 zugegriffen und über einen Number Slider eines dieser Elemente ausgewählt. Der Slider ermöglicht hierbei ein einfaches Wechseln zwischen den einzelnen Wandelementen in der Beispieldatei, in Abb. 5.14 wird hierzu z.B. die senkrechte Freiformwand ausgewählt. Zum Einstellen der wählbaren Parameter, hier nur Düsenbreite und Schichthöhe, werden Number Slider eingerichtet, es ist aber auch möglich, diese Parameter direkt im BIM-Modell als globale Parameter zu definieren. Die Düsenorientierung wird im Folgenden nicht wählbar sein, sondern immer als vertikal angenommen.

Slicing

Ab hier wird zunächst gleich zu den handelsüblichen *Slicing*-Tools vorgegangen und die Geometrie in Scheiben geschnitten. Um dies in Dynamo zu realisieren, wird erst aus dem Wandelement der zugewiesene Volumenkörper extrahiert und dieser dann über den Algorithmus 5.1 geschnitten. In Zeile 1 wird hierzu eine Menge von z-Koordinaten erzeugt, angefangen mit der **Schichthöhe** bis zur Höhe des Bauteils mit der Schrittweite entsprechend der **Schichthöhe**. Mit Punkten entlang der z-Achse mit den zuvor definierten z-Koordinaten und mithilfe der z-Achse als Normalenvektor werden somit für jede Schicht jeweils eine Ebene erzeugt (vgl. Zeile 3) – soll eine andere Ebene als die *XY*-Ebene Bezugsebene sein, muss hier ein anderer Normalenvektor gewählt werden. Schließlich wird der Volumenkörper der Wand (hier mit `wallSolid` bezeichnet) über die Funktion `Geometry.Intersect()` in Scheiben geschnitten und in der nächsten Zeile mit der Funktion `Surface.PerimeterCurves()` die Umfangslinien extrahiert. Die *Perimeter*-Funktion erstellt dabei aus einem Flächenobjekt (`Surface`) Kurvenobjekte (`Curve`), und zwar in der Art, dass die jeweiligen Start- und Endpunkte der Kurven zueinander im mathematisch positiven Drehsinn angeordnet sind (vgl. Abb. 5.15). In einem nachfolgenden Schritt werden diese Kurven neu sortiert, sodass die Wandinnenseite an **Indexposition 0** verschoben wird.

Algorithmus 5.1: Slicing der Geometrie

```
z=layerH..height..layerH;
o=Point.ByCoordinates(0,0,z);
planes=Plane.ByOriginNormal(o,Vector.ZAxis());

horizontalCut = List.Flatten(Geometry.Intersect(wallSolid,planes),-1);
curves = Surface.PerimeterCurves(horizontalCut);
```

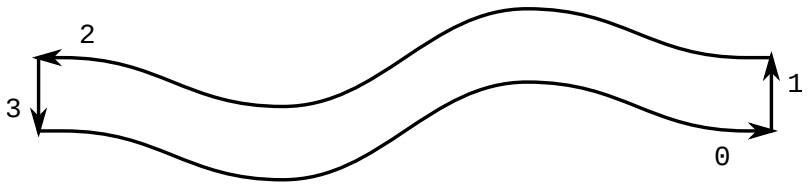


Abbildung 5.15: Umfangslinien eines Schnitts.

Gibt es mehrere Kurvensegmente, die zur Wandinnenseite assoziiert werden können, dann wird diejenige, die am Kurvenende zum Rand anschließt, auf diese Position gesetzt.

Konturpfad

In den folgenden Schritten wird nun der Druckpfad generiert. Um einen gleichmäßigen Oberflächenabschluss über das gesamte Bauteil zu erzeugen, macht es aber zunächst Sinn, den Pfad als erstes um das Bauteil herum zu führen. Zu diesem Zweck wird ein *Benutzerdefinierter Block* definiert, also ein eigenes Modul (`MultiPerimeterPath`), in dem alle hierzu notwendigen Code-Blöcke gesammelt werden. Der in Abb. 5.15 gezeigte Pfad muss zunächst noch etwas (genau um die halbe Düsenbreite) nach innen versetzt werden, da der Werkzeugmittelpunkt (*englisch*: Tool Center Point, TCP) auf die Mitte des Filaments ausgerichtet ist. Hierzu dient ganz einfach die Funktion `Offset`, wobei aber vorauszusetzen ist, dass die zu bearbeitende Kurve eindeutig auf einer horizontalen Ebene verortbar ist (nicht der Fall bei Linien). Ansonsten kann es sein, dass der Offset in eine falsche Richtung ausgeführt wird, bei einer Linie sind theoretisch unendlich viele Richtungen möglich. Der Umfang (vgl. Abb. 5.15) wird zudem als geschlossene Polylinie mit der Offset-Funktion bearbeitet, damit die Ecken des Linienobjekts richtig zugeschnitten werden. Für mehr Modularität lässt sich dieser Block über zwei Parameter modifizieren, zum einen wird die Wahl gelassen, wie viele Umfangslinien ineinander geschachtelt werden (Wandstärke), und zum anderen kann die Startposition verschoben werden (vgl. Abb. 5.16). Die Anzahl der ineinander geschachtelten Umfangslinien hat wie in Abschnitt 5.3.3 beschrieben einen stabilisierenden Effekt, daher ist hiermit ein Stabilitätsfaktor einstellbar. Im Bild auch erkennbar ist, dass die Ausgabeparameter zur besseren Handhabung in einem Dictionary gesammelt und als einzelne Variable weitergegeben werden.

Abbildung 5.17 zeigt den Inhalt des Ausgabe-Dictionaries, die dargestellten Variablen werden hierzu im Folgenden erläutert:

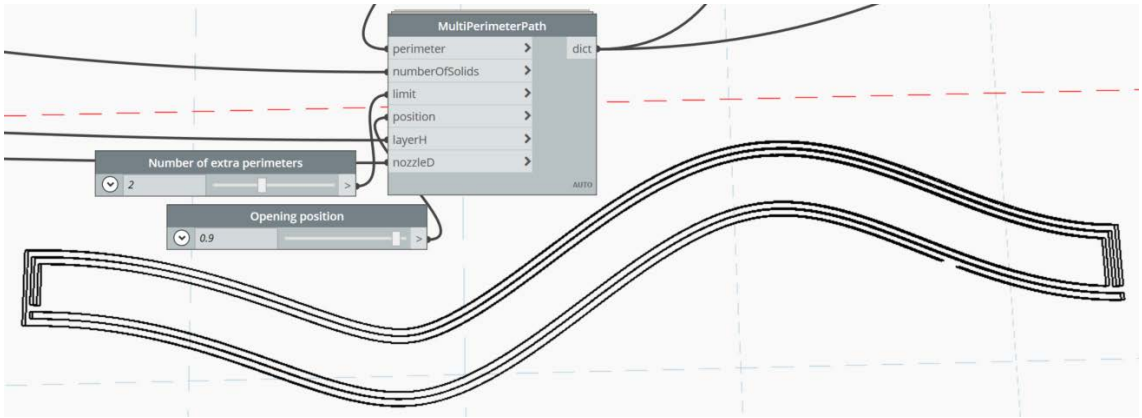


Abbildung 5.16: Erste Pfadgestaltung, mehrfache Umfangslinien zu einem durchgängigen Pfad vereinigt.

Abbildung 5.17: Übergabeparameter der Pfadgenerierungsmodule.

`perimeter`: Stellt den vom Modul abgegrenzten Bereich dar, also den aktuellen Offset.

`path`: Gibt den bisher erzeugten Pfad als `PolyCurve` wieder.

`accIdx`: Stellt den Index der Verbindungsstelle im bisher erstellten Pfad dar, also die Nummer der entsprechenden Kurve in der `PolyCurve`.

`reversed`: Gibt in Form eines booleschen Werts an, ob der entsprechende Offset in der Orientierung umgekehrt wurde.

`numberOfSolids`: Ein Integerwert, der angibt aus wie vielen zusammengesetzten Volumenkörpern das zu druckende Bauteil besteht.

`counter`: Zählvariable für die wiederholte Anwendung des Umrandungspfad-Blocks.

`nozzleD`: Die Düsengröße, die für den Druck gewählt wurde.

Jede weitere Verschachtlung vom Umfang wird durch einen Offset nach innen (negative Distanz) realisiert (`perimeter`-Variable). Bei jedem dieser Offsets wird zudem die

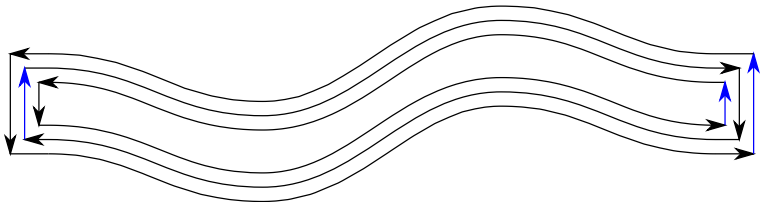


Abbildung 5.18: Orientierung der verschachtelten Offsets des Umfangs. In blau markiert ist jeweils die Indexposition 1, hier wird eine Verbindung nach innen gesetzt.

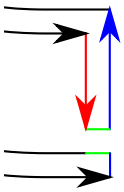


Abbildung 5.19: Detaillierte Ansicht eines Anschlusses

Orientierung der Kurvensegmente umgekehrt und mit der Variable `reversed` vermerkt (vgl. Abb. 5.18). Damit die Richtungskonvention der Linien zueinander stimmt, muss auch die Reihenfolge der Kurvensegmente umsortiert werden, indem die Liste der Kurven invertiert wird und gegebenenfalls um die Zahl der Volumenkörper (`numberOfSolids`) des Bauteils verschoben wird. Der innere Umfang wird dann an einer passenden Stelle mit dem äußeren Umfang verbunden. Hier wurde als geeignete Stelle jeweils die Indexposition 1 der einzelnen Offsets gewählt, also abwechselnd rechter und linker Rand (siehe blaue Segmente in Abb. 5.18). Im bisher zusammengesetzten Pfad (`path`) wird diese Stelle über den Index `accIdx` beschrieben, welcher durch die Anzahl der vorherigen Pfadsegmente plus `numberOfSolids` bestimmt wird.

Die Generierung des Pfades verläuft also stückweise, indem immer wieder neue Innenstrukturen erstellt werden und an der entsprechenden Position mit der äußeren Struktur in Verbindung gebracht werden. Diese Verbindung entsteht durch einfaches Ersetzen des Teilsegments an der Indexposition `accIdx` mit den hierfür vorbereiteten Innensegmenten. Der genaue Vorgang kann anhand von Abb. 5.19 erklärt werden. Das blaue Segment (Pfadsegment an Indexposition `accIdx`) wird zunächst zerteilt, sodass ein `nozzleD` breites Stück entnommen werden kann, entsprechend dem Abstand zweier direkt aneinander angrenzenden Pfadlinien. Das letzte Segment der inneren Struktur (hier rot dargestellt) wird um dieselbe Länge gekürzt. Schließlich werden die beiden Verbindungssegmente (im Bild grün) erstellt und alles miteinander zu einer geschlossenen `PolyCurve` verbunden.

Innenstrukturen

Der in Abb. 5.16 dargestellte Druckpfad ist damit für jeden Schnitt von Anfang bis Ende durchgängig. Schließt man nun alle Pfade in allen Schnittebenen der Reihe nach miteinander zusammen, so könnte dieser Pfad ohne weitere Bearbeitung auch schon z.B. im Contour Crafting eingesetzt werden (vgl. Abschnitt 2.3.3). Durch Anfügen weiterer Module kann in jeder Schnittebene der Pfad um weitere Innenstrukturen erweitert werden. Beispiele für eine weitere Detaillierung sind hierbei z.B. ein Zickzack-Muster, in Abb. 5.20 dargestellt, oder ein Lamellen-Muster, in Abb. 5.21. Die Generierung dieser und ähnlicher Pfade funktioniert dabei auf ähnliche Weise wie bei der einfachen Umrandung. Erst wird die innere Struktur erzeugt und dann mittels demselben Algorithmus wie zuvor an

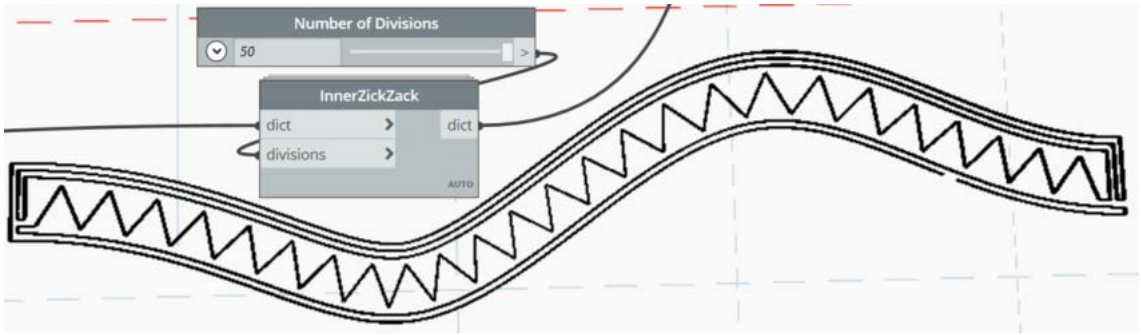


Abbildung 5.20: Zickzack-Modul und Druckpfad mit entsprechender Struktur.

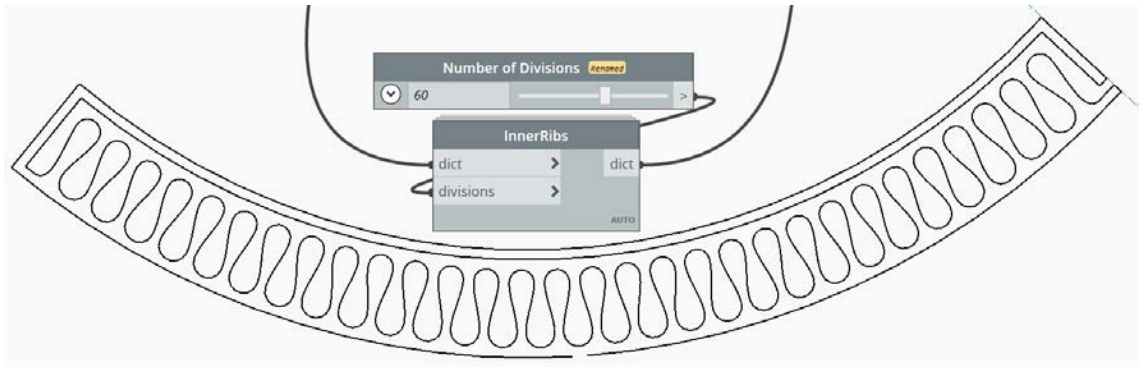


Abbildung 5.21: Lamellen-Modul und Druckpfad mit entsprechender Struktur.

den äußeren Pfad angeschlossen. Es wird im Gegensatz zu den Konturpfaden bei den Innenstrukturen der komplette Innenraum mit dem entsprechenden Muster ausgefüllt.

Die Erzeugung der inneren Struktur beruht in diesen beiden Fällen auf zwei Offsets, da hier ein Hin- und Rückweg geplant werden muss. Auf dem Hinweg wird der Zickzack- bzw. Lamellenweg geplant, indem zunächst die Segmente der Wandinnenseite in eine Anzahl gleichlanger Segmente aufgeteilt werden und die jeweiligen Endpunkte mit der Funktion `ClosestPointTo` auf die Wandaußenseite des inneren Offsets projiziert werden. Wie viele Unterteilungen dabei vorgenommen werden sollen, kann durch einen Parameter (`divisions`) von außen eingestellt werden. Ab hier unterscheiden sich die beiden Module in einigen Details. Die beiden Muster, die wie folgt erstellt werden, können über dieselbe Verbindungsfunktion wie zuvor mit dem äußeren Pfad verbunden werden.

Zickzack: Auf beiden Seiten (Hin- und Rückweg Offset) wird versetzt zueinander jeder zweite Endpunkt ignoriert, die restlichen Punkte werden dann zickzackförmig über Linienobjekte (`Line`) miteinander verbunden. Sobald der Rand erreicht ist, wird mit der Wandaußenseite des äußeren Offsets, hier also dem Rückweg, verbunden.

Lamellen: Auf beiden Seiten (Hin- und Rückweg Offset) wird versetzt zueinander jeder zweite Endpunkt ignoriert, an den restlichen Punkten werden jeweils Halbkreise (`Arc`) mit dem Radius entsprechend dem Abstand zum nächsten ignorierten Punkt abzüglich der Düsenbreite platziert. Die Halbkreise werden dann abwechselnd mit der Funktion (`Curve.ByBlendBetweenCurves`) mit jeweils tangentialen Kurvensegmenten verbunden.

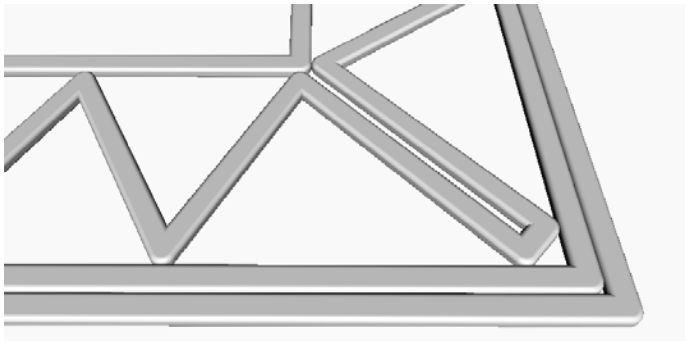


Abbildung 5.22: Eckübergang bei einer Zickzack-Innenstrucktur.

Sobald der Rand erreicht ist, wird mit der Wandaußenseite des äußeren Offsets, hier also dem Rückweg, verbunden.

Die Einstellbarkeit beider Innenstrukturen wurde hierbei jedoch eher einfach gehalten, um eine Gleichmäßigkeit zu garantieren. Die Anzahl der Zacken bzw. Lamellen wird wie zuvor beschrieben direkt von der einstellbaren Anzahl der Offset-Unterteilungen gesteuert. Dabei wurde aber darauf geachtet, dass nur eine geradzahlige Anzahl an Unterteilungen vorgenommen werden kann. Auf diese Weise ist sichergestellt, dass nur ganze Zacken bzw. Lamellen gezeichnet werden und Anfang und Ende des Musters auf derselben Offset-Seite liegen. Die Offset-Seite mit dem Zacken- bzw. Lamellenmuster stellt dabei den Hinweg dar und die andere Offset-Seite den Rückweg zur Schnittstelle mit dem Konturpfad.

Existiert eine Ecke in der Geometrie, wird wie folgt vorgegangen: Es werden jeweils an den Abschnitten vor und nach der Ecke ganze Zacken bzw. Lamellen bis zum Ende der kürzeren Offset-Seite gezeichnet. Anschließend wird eine Eckverbindung wie in Abb. 5.22 dargestellt zwischen den Abschnitten eingefügt. Auf diese Weise entsteht allerdings ein recht großer Luftraum an der Ecke, wenn der Eckwinkel sehr spitz ist, ansonsten konnten keine großen Probleme mit dieser Methode festgestellt werden.

Der in Abb. 5.20 oder Abb. 5.21 gezeigte Pfad kann nun ohne weiteres für einen Druckversuch verwendet werden. Jedoch ist es, wie in Abschnitt 5.3.3 bereits erwähnt, in den meisten Fällen sinnvoll, vor einem Druck mindestens die Kollisionssimulation durchzuführen. Für weitere Simulationen ist aber eine möglichst genaue 3D-Geometrie des gedruckten Bauteils notwendig. Mehr dazu im nächsten Abschnitt.

Strukturkombinationen

Auf ähnliche Weise können diverse andere Innenstrukturen erzeugt werden, die auch, sofern sinnvoll, in unterschiedlichster Art und Weise miteinander kombiniert werden können. Diese Kombinationsmöglichkeit ist bereits in Abb. 5.4 dargestellt worden, dort auch sichtbar ist eine weitere mögliche, hier aber nicht implementierte Innenstruktur, das Honeycombmuster. Eine mögliche Kombination wäre z.B. eine Verstärkung der Wandinnenseite

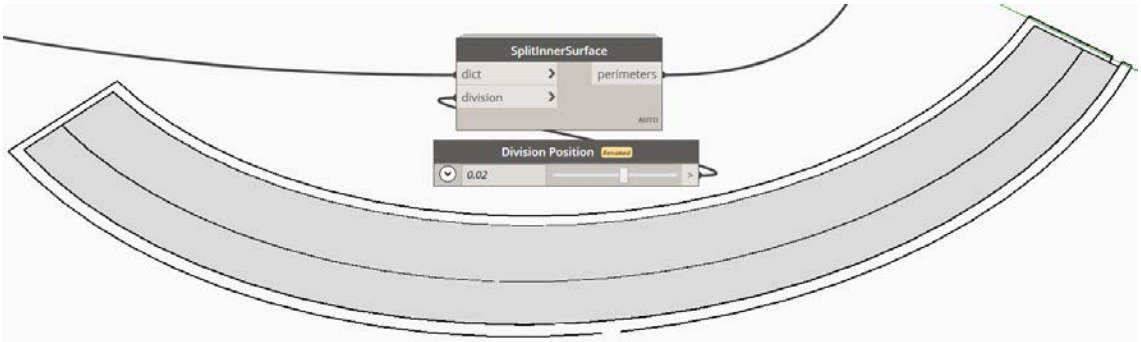


Abbildung 5.23: Aufteilung der Pfadinnenfläche für die Erstellung einer Musterkombination.

(strukturelle Stabilität) und eine Zickzack-, Lamellen- oder Honeycombstruktur an der Wandaußenseite (Wärmeisolation).

Um eine Kombination im Revit-Skript zu realisieren, kann die Innenfläche des Konturpfads (vgl. Abb. 5.16) mittels einer eigenen Funktion (`SplitInnerSurface`) geteilt und die entstandenen jeweils wie oben beschrieben mit einem eigenen Muster befüllt werden (vgl. Abb. 5.23). Sind die jeweiligen Muster erstellt, kann wie zuvor auch der innere Pfad mit dem Äußeren verbunden werden. Diese Kombinationsmöglichkeiten wurden jedoch aus Zeitgründen noch nicht vollständig implementiert.

5.6 Ableitung eines Volumenmodells

Im Folgenden wird erläutert, wie ausgehend von dem Druckpfad das Volumenmodell zur Beschreibung der Materialverteilung für das zu druckende Bauteil erzeugt werden kann (vgl. Abschnitt 4.1.2). Hierzu muss aus dem Pfad, also einem Linienobjekt, ein Volumenkörper erstellt werden. Eine mögliche Methode, die sich hierfür anbietet, ist die Volumenkörper-Operation **Sweep**. Mit dieser Operation kann eine Querschnittsfläche entlang einem Pfad zu einem Volumenkörper gezogen werden. Sowohl Querschnitt als auch Pfad liegen vor, wenn das Skript aus Abschnitt 5.5 angewendet wird. Somit scheint diese Methode einfach durchführbar zu sein.

Zunächst muss hier aber nochmals erwähnt werden, dass der aus dem Dynamo-Skript erhaltene Querschnitt nur eine Abschätzung darstellt und nur auf Erfahrungswerten basiert. Der genaue materialspezifische **Querschnitt** ist hierbei jedoch nicht bekannt und müsste genau genommen in einem unverhältnismäßigen Aufwand in materialwissenschaftlichen Studien ermittelt werden. Zudem können andere Faktoren einen Einfluss auf die Formgebung haben, wie z.B. Inhomogenitäten im Material (Lufteinschlüsse, Klümpchen) oder Richtungswechsel. Der Querschnitt ist also entlang des Druckpfades zudem nicht zwingend konstant und kann sich auch durch Materialschwund bei Aushärteprozessen sowie durch mechanische Belastung nach dem Drucken noch verändern. Mangels guter Alternativen zur Berechnung des wirklichen Querschnitts muss hier ein gewisser Fehler in Kauf genommen werden. Wie groß die Abweichung zum tatsächlichen Querschnitt ist aber hier vorerst nicht von großer Bedeutung.

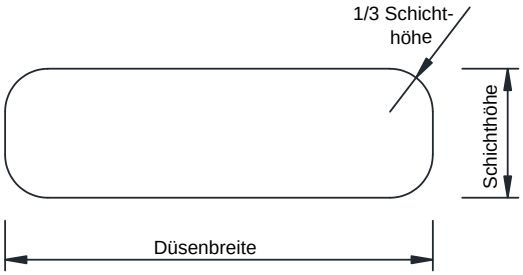


Abbildung 5.24: Extrusionsquerschnitt.

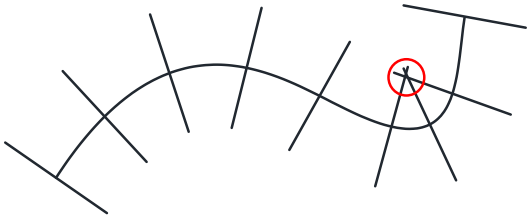
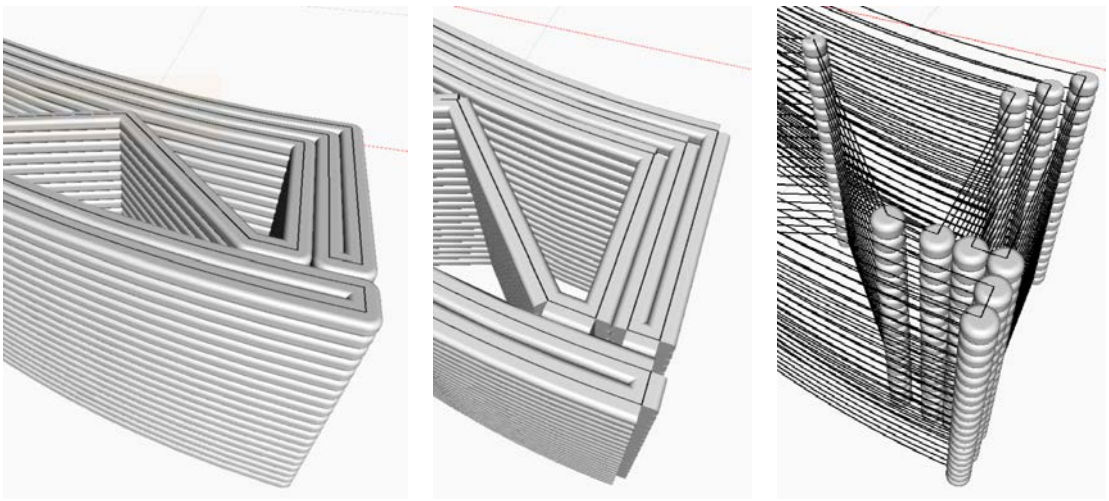


Abbildung 5.25: Skizzierte Darstellung eines Sweeps mit Intersectionproblem an der markierten Stelle.

In Abb. 5.24 wird der für die folgenden Aufgaben gewählte Querschnitt dargestellt. Die Breite des Querschnitts entspricht hierbei direkt der Düsenbreite, die Höhe genau der Schichthöhe und die abgerundeten Ecken haben einen Radius von einem Drittel der Höhe. Auf jeden Fall wird ein hiermit erzeugter Sweep-Volumenkörper eine Idealisierung darstellen.

An dieser Stelle ist es noch notwendig, die Operation Sweep genauer zu beschreiben. Wie schon zuvor erwähnt, ist die Operation eine Volumenkörper-Operation, die aus einer geschlossenen Linie (dem Querschnitt) und einer Führungslinie oder Pfad einen 3D-Körper generiert. Dabei ist das „sweepen“ ungefähr vergleichbar mit dem Ziehen einer lang gezogenen Seifenblase, wenn diese nach dem Aufziehen starr bleiben würde. Der Querschnitt dreht sich bei gekrümmten Pfaden immer genau mit dem Pfad mit und bleibt über den gesamten Pfad formgleich. Vereinfacht ist dies in Abb. 5.25 veranschaulicht. Dies funktioniert allerdings nicht mit jeder Art von Pfaden. Sobald ein Knick, eine zu starke Krümmung oder Überschneidungen auftreten, schlägt die Operation aufgrund von Selbsüberschneidung fehl (siehe rote Markierung in Abb. 5.25). Bei 3D-Druckpfaden kommt dieses Problem aber leider sehr häufig vor. Je nach vorgegebener Geometrie sind Knicke (vgl. Abb. 5.20) und starke Krümmungen nicht selten und denkbar ist es auch dass sich der Druckpfad überkreuzt (HENKE et al., 2020). Die Erzeugung der 3D-Geometrie durch die Sweep Operation ist allein aus diesem Grund schon eine ziemliche Herausforderung.

Eine Möglichkeit, dennoch einen durchgängigen Volumenkörper zu bewerkstelligen (vgl. Abb. 5.26a) ist ein vereinter stückweiser Sweep. Die Sweeps funktionieren so zwar fehlerlos, aber auch hier sind die Ecken ein Problem, durch die fehlende Drehung am Ende des Liniensegments wird die Ecke nicht modelliert (vgl. Abb. 5.26b). Es müssen also nicht nur alle einzelnen Sweeps über boolesche Vereinigungs-Operationen zeitaufwendig zusammengefügt werden, sondern auch noch die fehlenden Stücke an den Ecken durch Rotationskörper ergänzt werden (vgl. Abb. 5.26c). Sobald alle Einzelteile über die boolesche Vereinigung zu einem zusammenhängenden Volumenkörper verschmolzen sind, kann die Geometrie für Simulationsprogramme in verschiedene Datenformate exportiert werden.



(a) Gewünschtes Ergebnis (b) Einzelne Sweeps (c) Rotationskörper

Abbildung 5.26: Über Sweep und Rotationskörper erstellter Volumenkörper

5.7 Maschinensteuerung

Die ein den vorhergehenden Abschnitten erstellten Linien- und Volumengeometrien beschreiben zwar den zu druckenden Körper, allerdings kann mit diesen Informationen nicht direkt der Drucker gesteuert werden. Es muss daher zuvor noch eine Übersetzung in maschinenverständlichen Code stattfinden. Bevor aber der Druckpfad übersetzt werden kann, muss dieser umgewandelt werden in eine Liste von Wegpunkten. Hierzu wird in Dynamo der komplette Druckpfad über die Funktion `Curve.SplitByParameter` in einzelne gleichlange Pfadsegmente geteilt und mit `Curve.EndPoint` nur jeweils der Endpunkt ausgewählt.

Der Funktion `Curve.SplitByParameter` muss hierzu ein Satz an Parametern übergeben werden, welche die Schnittposition entlang des Pfades beschreiben. Hierzu wird mittels eines Schiebereglers und einer kleinen Umrechnung dem Designer die Möglichkeit geboten, die Länge der Segmente anzupassen. Auf diese Weise wird die Geometrie, die durch den Druckpfad beschrieben wird, diskretisiert und ist damit nur eine Annäherung. Bei genügend kleinem Schnittabstand bleibt diese Diskretisierung jedoch völlig unbemerkt.

Abschließend müssen die so erzeugten Wegpunkte in ein für den Roboter verständliches Format übertragen werden. Hierfür wurde ein Python-Skript implementiert, dass die Koordinaten der Wegpunkte ausliest und diese für die einzelnen Komponenten (kartesische x-, y- und z-Koordinaten) in eine URscript-Datei jeweils in ein Array schreibt. Zusätzlich werden in dieser Datei die Anzahl der Punkte und Steuerungsinformationen wie die Beschleunigung, Geschwindigkeit und der Ausrundungsradius zwischen den einzelnen Liniensegmenten mit abgespeichert.

Der Ausrundungsradius sorgt bei Linienführung des Roboters (vgl. Abschnitt 2.4) für eine kontinuierliche Bewegung, da ohne eine Abrundung an Ecken starke Beschleunigungen auftreten würden. Der Wert des Ausrundungsradiuses sollte deutlich kleiner als die Länge

Algorithmus 5.2: Definition der Wegpunktkoordinaten für die Pfadabfolge

```
global X = [0.000,0.201,0.111,0.021, ...
global Y = [0.000,1.964,1.971,1.975, ...
global Z = [0.000,0.000,0.000,0.000, ...
global NrPoints = 15907
global a = 1.2
global v = 0.015
global r = 0.001
global Startpunkt = StartP
```

Algorithmus 5.3: Definition der Pfadabfolge in URscript

```
1 def printing():
2     Loop01 = 0
3     while Loop01 < NrPoints:
4         WP=p[X[Loop01]/10,Y[Loop01]/10,Z[Loop01]/10,0,0,0]
5         WP=pose_add(WP, StartP)
6         movep(WP, a, v, r)
7         Loop01 = Loop01 + 1
8     end
9 end
```

der Liniensegmente sein, die über den Boden verbunden werden. Das Ergebnis der Konvertierung von der Pfadgeometrie in eine URscript-Datei wird in Algorithmus 5.2 dargestellt.

Dieses Koordinaten-Skript muss in einem Roboterprogramm, das mit dem Handgerät des Roboters erstellt wird, als *„Before Start“*-Knoten geladen werden, damit die Koordinaten während dem Druckvorgang abrufbar sind. Anschließend muss nur noch das Steuerungsprogramm erstellt werden. Auch das kann in Form einer URscript-Datei Offline programmiert werden und als *„Before Start“*-Knoten geladen werden. Algorithmus 5.3 zeigt die nötige Codeabfolge, um den Roboter entlang der Wegpunkte zu führen. Die Koordinaten werden hier in Zeile 4 in eine Roboter Pose umgewandelt und in Zeile 5 zu einem vordefinierten Bezugspunkt – selbst gewähltes Objektkoordinatensystem ($\boldsymbol{O}_{CF}$) – referenziert (vgl. Abschnitt 2.4.2). In Zeile 6 wird schließlich die Bewegung eingeleitet. Um das Programm ausführbar zu machen, muss nun nur noch ein Script-Programmknoten in der regulären Programmstruktur erzeugt werden, in dem die Funktion `printing()` aufgerufen wird.

5.8 Export der erzeugten Daten

Der Dynamo-Prototyp bildet zwar den FIM-Arbeitsablauf ab, aber es stellt noch kein Datenmodell zur Speicherung der entsprechend erzeugten Daten dar. Um dennoch diese Daten für andere Programme (CAD-Tools oder Simulationssoftware) nutzbar zu machen, wurden verschiedene Exportoptionen implementiert (vgl. Abb. 5.27). Ein Datenmodell, in dem alle erzeugten Daten gespeichert werden können, wird erst nachfolgend beschrieben,

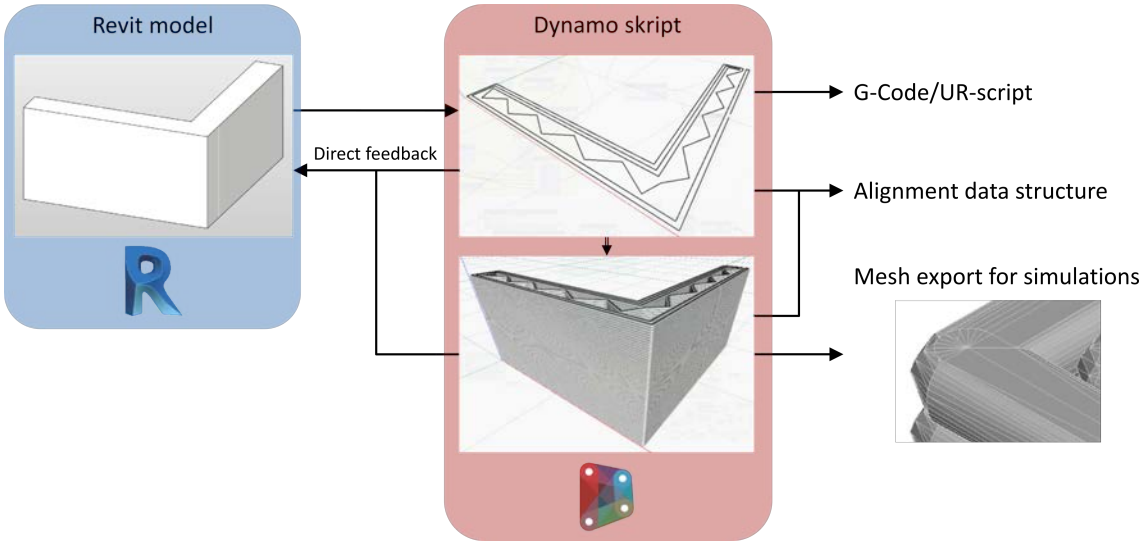


Abbildung 5.27: Verschiedene Exportoptionen nach Umwandlung des BIM-Modells in Druckpfade und das dazugehörige Volumenmodell.

in Abb. 5.27 wird dieses Datenformat aber bereits als Alignment Datenstruktur vorgestellt. Diese Option soll eigentlich also eigentlich keinen Export darstellen, sondern das Speichern in einem geeigneten Datenformat. Als Datenformat wird in dieser Arbeit das IFC-Format vorgeschlagen, insbesondere unter Verwendung der Straßenbauelemente, die mittlerweile in der aktuellen Version von IFC fest integriert wurde, daher auch die Bezeichnung „Alignment“. In Kapitel 7 wird auf dieses Datenformat genauer eingegangen und über eine mögliche Speicherfunktion berichtet.

Der Pfad kann wie bereits beschrieben über ein Python-Übersetzungsskript, das in einen Dynamo-Programmknoten eingebettet werden kann, direkt in Steuerungsprogramme übersetzt werden und und für den Druckprozess eingesetzt werden (wird in Abschnitt 5.7 beschrieben). Des Weiteren kann sowohl der Druckpfad als auch die entsprechende 3D-Geometrie per „direct feedback“ zurück nach Revit exportiert werden, um eventuell die gedruckte Geometrie im Kontext des gesamten Projekts zu visualisieren. Für Simulationen kann die 3D-Geometrie auch über eine Dynamo Erweiterung, dem Mesh Toolkit, in ein Mesh umgewandelt werden und als STL-Datei abgespeichert werden.

Kapitel 6

Laborversuch

Zur Validierung des in diesem Kapitel beschriebenen Pfadgenerierungsskriptes und den damit erstellten Pfaden wurden Druckversuche durchgeführt. Als Vorlage wurden hierzu die in Abschnitt 5.5 beschriebenen Beispielwände zehnfach verkleinert in verschiedenen Durchläufen gedruckt. Aufgrund von beschränkter Laborfläche wurde anstatt Beton-3D-Druck ein Extrusionsdruck mit Ton durchgeführt, da dieses Material deutlich einfacher zu handhaben ist, aber sehr ähnliche rheologische Eigenschaften wie Beton besitzt.

Als Extrusionsapparat kam in diesem Zusammenhang ein **Keramikdruckkopf** der Firma *StoneFlower3D* zum Einsatz. Mit diesem Druckkopf kann pastenförmiges Material extrudiert werden indem ähnlich zu einer Spritze aus einer Kartusche (Ram Extruder) die Paste herausgedrückt wird und über einen Schlauch zum Druckkopf geführt wird. Sowohl der Ram Extruder als auch der Druckkopf sind mit einem Schrittmotor ausgestattet, die beide gekoppelt über eine elektronische Steuereinheit betrieben werden. Der Schrittmotor des Ram Extruders dreht dabei eine Gewindestange, über die der Kolben in der Kartusche bewegt wird. Im Druckkopf betreibt der Schrittmotor eine kleine Förderschnecke, die dafür sorgt, dass das Material gleichmäßig aus der Düse austritt. Die Steuereinheit kann eigenständig, ohne Input-Signal, im Dauerbetriebsmodus betrieben werden und die Geschwindigkeit der Schrittmotoren über ein Potentiometer eingestellt werden. Ebenso kann die Kopplung der beiden Motoren über ein Potentiometer angepasst werden, so dass der Motor im Druckkopf mit ¼- bis 4-facher Relativgeschwindigkeit zu dem Ram Extruder Motor gefahren werden kann.

Als Bewegungsapparat zur Führung des Druckkopfes wurde der **UR10e** Roboter von der Firma *Universal Robots* eingesetzt. Der UR10e ist ein Knickarmroboter mit 6 Gelenken, einer Reichweite von etwa 1,3 m und einer Tragfähigkeit von 10 kg. Der Roboter ist auf einem Labortisch montiert und hat aufgrund des recht kleinen Laborzimmers einen stark eingeschränkten Arbeitsbereich, jedoch kann trotzdem ein Arbeitsbereich von mindestens 1 m x 1 m x 1 m angesteuert werden. Gesteuert wird der Roboter über einen Schaltkasten, mit dem, wie schon in Abschnitt 3.5 beschrieben, entweder über das angeschlossene Handgerät oder per TCP/IP Schnittstelle kommuniziert werden kann. Daten können ebenfalls per Netzwerk zum Roboter übertragen werden, oder über einen der USB-Anschlüsse.

Das Material, der Ton, kann in jedem beliebigen Bastelbedarf in verschiedenen Farben in Form von 10 kg-Hubel gekauft werden. Der gekaufte Ton ist allerdings noch etwas zu hart, um mit dem Extruder verwendet zu werden und muss daher mit etwas Wasser in eine weichere Konsistenz geknetet werden. Hierzu wird einfach eine bestimmte Menge Ton abgemessen und mit einer entsprechenden Menge Wasser in einer Schüssel sorgfältig

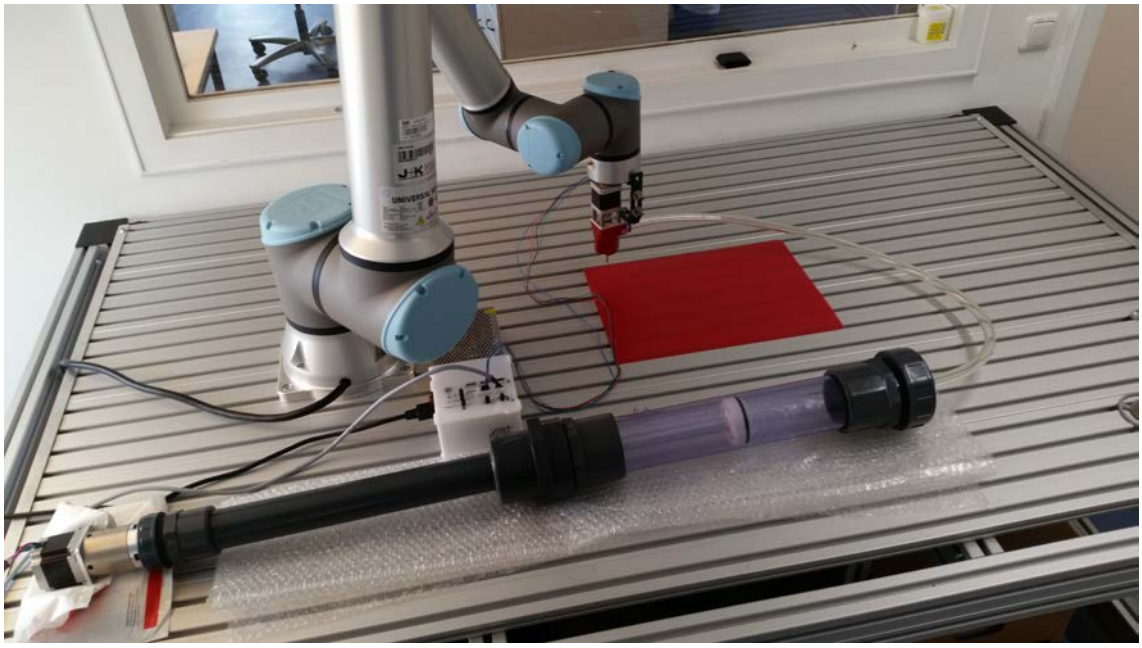


Abbildung 6.1: Versuchsaufbau für den Ton-3D-Druck eines aktivierten Wandelements in Modellgröße 1:10.

verrührt und anschließend, um Klümpchen zu vermeiden, durch ein Sieb gedrückt. Der so aufbereitete Ton ist gekühlt sehr lange haltbar, kann also in großer Menge vorbereitet werden.

Das gesamte System wird in Abb. 6.1 gezeigt. Zu sehen ist der Roboter mit Druckkopf am TCP, der Ram Extruder (vorne), die Steuereinheit des Extruders (weiße Box) und der gewählte Druckbereich (rote Matte).

6.1 Druckvorgang

Bevor der Druckvorgang aber gestartet wird, muss der vorbereitete Ton in die Kartusche gefüllt werden und der Ram Extruder mit dem Druckkopf über den Schlauch verbunden werden. Dann muss entsprechend der Druckpfadplanung eine Düse gewählt werden mit 1/10 der geplanten Düsengröße und diese am Druckkopf angebracht werden. Sind alle Kabel an den Schrittmotoren richtig angeschlossen kann nun der Extruder solange angetrieben werden, bis der Ton über den Schlauch zum Druckkopf vorgetrieben ist. Damit ist der Drucker vorbereitet und der eigentliche Druckvorgang kann nun initiiert werden, indem das zuvor beschriebene Programm gestartet wird. Hierzu wird am Handgerät auf die dafür vorgesehene Taste gedrückt und abgewartet, bis sich der Roboter in Bewegung setzt, um dann den Extruder manuell zu starten.

Nach wenigen gescheiterten Testläufen konnte das Equipment sehr schnell kalibriert werden und gute Druckergebnisse erzielt werden. Hierzu musste, wegen noch fehlender elektronischer Steuermöglichkeit, die Schrittmotorgeschwindigkeit manuell angepasst werden, sodass die Extrusionsgeschwindigkeit mit der Druckkopfgeschwindigkeit abgestimmt

ist (vgl. Abb. 5.6). Die aus Druckkopfgeschwindigkeit und Extrusionsgeschwindigkeit ermittelte Extrusionsrate ist so einzustellen, dass möglichst ein Ergebnis zwischen den in Abb. 5.6 dargestellten Querschnitten a) und b) erreicht wird. Abbildung 6.2 zeigt hier auf der rechten Seite einen nicht Kalibrierten Druckversuch und links einen Druckversuch mit gut eingestellter Extrusionsrate. Abbildung 6.3 veranschaulicht einen optimalen Druckverlauf, der schließlich zu einem erfolgreichen Testergebnis führt (Abb. 6.4).

6.2 Ergebnisse

In Abb. 6.2 sieht man auf der rechten Seite sehr deutlich, wie sich eine zu langsame Extrusionsgeschwindigkeit auf das Druckobjekt auswirkt. Mit der hier eingestellten Extrusionsrate fällt der Querschnitt des Filaments deutlich schmäler aus als vorhergesagt, so sehr, dass sich parallele Bahnen nicht berühren. Ein Objekt in voller Größe wäre so nicht druckbar bzw. nicht sinnvoll, da auf diese Weise keine Haftung zwischen den parallelen Bahnen erzeugt wird und so die strukturelle Stabilität stark eingeschränkt wird. Nach dem Trocknen ist dieses Modell ohne viel Kraftaufwand auseinandergebrochen.

Das Modell auf der linken Seite von Abb. 6.2 ist bereits deutlich besser ausgefallen, jedoch wurde die Pfadplanung hier nicht optimal durchgeführt. An der inneren Ecke sieht man hier, dass die Innenstruktur nicht die Außenwand berührt, da Ecke diagonal zu Ecke gezeichnet wurde und aus dem in Abschnitt 5.3.1 beschriebenen Grund hier eine Fehlstelle auftritt. Hierzu wurde das Pfadplanungsskript etwas angepasst, um die Bahnführung in einer solchen Ecksituation besser zu gestalten, woraus der in Abschnitt 5.5 beschriebene Mechanismus entstanden ist (vgl. Abb. 5.22). Ein entsprechender Drucktest mit dieser Veränderung wurde allerdings noch nicht durchgeführt.

Der in Abb. 6.3 veranschaulichte Druckverlauf hat wie erwartet insgesamt etwa 145 min gedauert. Eine Vorhersage zu dieser Zeit kann ganz einfach ermittelt werden, indem die Länge des geplanten Druckpfads ermittelt wird und durch die eingestellte Geschwindigkeit geteilt wird. Im Fall dieses Versuchs wurde ein Druckpfad mit der Länge 128,95 m erstellt und eine Geschwindigkeit von 1,5 $^{cm}/_{s}$ eingestellt, daraus folgt eine Bearbeitungszeit von 143,27 min. Die Divisionen für die Wegpunkterzeugung wurden in diesem Versuch alle 9 mm durchgeführt, dies ist auch bei ganz genauem Hinsehen auf Abb. 6.4 nicht

Abbildung 6.2: Kalibrierung des Druckvorgangs durch die Extrusionsrate.

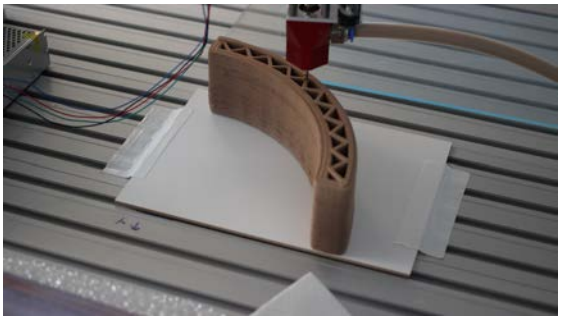


Abbildung 6.3: Optimaler Druckverlauf.

Abbildung 6.4: Erfolgreicher Druckversuch in voller Modellgröße.

Abbildung 6.5: Nach erfolgreichem Druck getrockneter Druckkörper. Hier ist ein deutlicher Volumenschwund erkennbar, vermutlich durch das Aushärten des Tons.

sichtbar. Es bleibt aber noch zu untersuchen ab welchem Grad der Diskretisierung hier ein merkbarer Effekt auftritt. Am Anfang des Druckversuchs wurde eine höhere Extrusionsgeschwindigkeit eingestellt, die nach etwa 3 cm Druckhöhe reduziert wurde auf einen optimalen Wert. Im Vergleich zu Abb. 5.6 entspricht die erste Einstellung eher dem Querschnitt b) und die zweite eher dem Querschnitt a). In Abb. 6.4 sieht man bei genauem Hinsehen, dass der untere Teil etwas bauchiger und dort vermutlich etwas breiter als das geplante Bauteil ist.

Eine genaue Messung der Größe wurde nach dem Trocknen jedoch nicht durchgeführt, da schon an der Unterlage gut erkennbar war, dass das Objekt um einiges geschrumpft ist (vgl. Abb. 6.5). Dieses Materialverhalten ist natürlich bei der Pfadplanung mit zu bedenken, jedoch wurde im Druckversuch Ton verwendet, der in dieser Hinsicht ganz andere Eigenschaften als Beton hat. Wie sich ein Objekt in tatsächlicher Größe und mit dem richtigen Material verhält muss noch evaluiert werden, jedoch nicht im Rahmen dieser Arbeit und auch nicht mit dem beschriebenen Versuchsaufbau.

Insgesamt kann aber zur Pfadplanung gesagt werden, dass diese mit der zuvor beschriebenen Herangehensweise sehr gut durchführbar ist. Die Bearbeitungszeit des Generierungsskriptes verläuft sehr schnell und kann von einem einfachen Office-PC in wenigen Sekunden durchgeführt werden. Ausgeführt wurde das Skript auf einem Lenovo Thinkpad T490 mit einem i7 Prozessor (Rechenleistung 1,9 GHz pro Kern). Für das gezeigte Druckbeispiel hat dieses System hierfür etwa 9 Sekunden gebraucht. Soll allerdings auch die 3D-Soll-Geometrie erstellt werden, ist das eine ganz andere Situation. Die vielen einzelnen Sweep- und Revolve-Operationen sowie die anschließende boolesche Vereinigung sind sehr zeitintensiv, wodurch sich die Ausführungszeit des Skripts mit demselben System auf mehrere Minuten verlängert hat und das schon bei einem verhältnismäßig kleinen Bauteil. Dazu ist zu erwähnen, dass viele Funktionen in Dynamo sowie in Grasshopper nur auf einem Prozessorkern berechnet werden und somit nicht die volle Leistung des Computers ausgenutzt wird, daher sollten hierfür eher spezialisierte Softwaretools eingesetzt werden. Jedoch gibt es noch ein weiteres Problem mit den Sweep-Körpern. Bei der booleschen Vereinigung tritt leider sehr häufig ein Fehler auf, wodurch diese

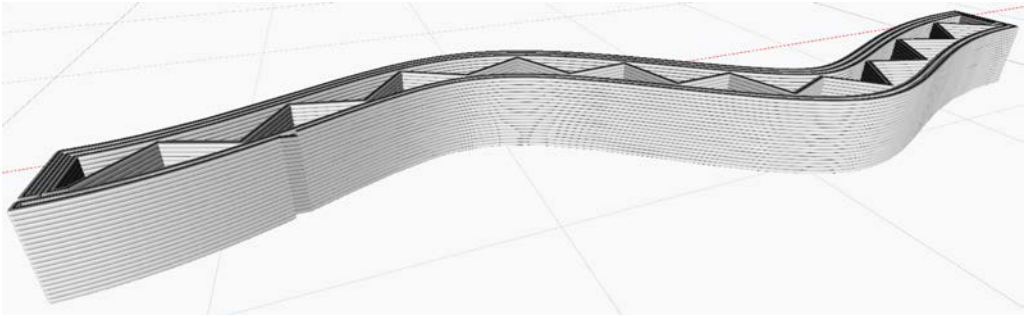


Abbildung 6.6: Erfolgreich vereinte Volumenkörper.

Prozedur fehlschlägt und kein zusammenhängender Volumenkörper generiert wird. Auch hier ist die Vermutung, dass eine auf diese geometrischen Operationen spezialisierte Software geeigneter wäre, um die 3D-Struktur zu erstellen. In dieser Hinsicht wurden auch die beiden anderen Softwarepakete Rhino und NX untersucht, aber leider ohne eine nennenswerte Verbesserung.

In einigen Fällen wurde aber eine erfolgreiche boolesche Vereinigung erreicht und ein zusammenhängender Volumenkörper erstellt. Ein solcher Volumenkörper ist in Abb. 6.6 zu sehen. Mit dem beschriebenen Mesh-Export als STL-Datei, konnte diese Geometrie verwendet werden, um eine Finite-Cell Wärmedurchgangssimulation durchzuführen. Auch wenn hier ein Erfolg erzielt wurde, so ist das Ergebnis dennoch sehr verbesserungsbedürftig. Zum einen liefert der Mesh-Export keine gute Vernetzung des Meshes, die Aspect Ratio lässt sich bei der Verwendung des Mesh-Toolkits in Dynamo nicht einstellen. Bessere Ergebnisse liefert hier Rhino/Grasshopper, da mit der entsprechenden Funktion in dieser Software die Aspect Ratio eingestellt werden kann. Hiermit wurde auch der STL-Export für die Simulation durchgeführt.

Auch wenn der Zusammenschluss aller Sweep-Geometrien funktioniert, so ist die Nachbearbeitung unverhältnismäßig aufwändig und die erzeugten Export-Geometrien nicht in bester Qualität, so dass damit Simulationen nur bedingt sinnvoll sind. Auch nach ausgiebiger Untersuchung weiterer Tools zur Erstellung von Sweeps und deren Vereinigung, konnte keine zufriedenstellende Lösung gefunden werden. Hier ist ein anderer Ansatz unbedingt notwendig.

Anstatt die Volumenbeschreibung über eine Oberflächenrepräsentation durchzuführen (wie zuvor mit der Sweep Operation), ist es auch möglich eine andere Strategie zu verfolgen. Wie in Abschnitt 4.1.2 bereits beschrieben, ist auch eine Beschreibung der Materialverteilung per *Volume Representation* möglich. Insbesondere kann hier eine Voxel-Darstellung sinnvoll sein, da mit dieser Methode explizit innere Strukturen beschrieben werden können und die Voxel (kleine Würfel) für z.B. Finite-Cell-Simulationen eine sehr gute Grundlage darstellen. Hier kann prinzipiell genauso vorgegangen werden, wie bei einer regulären Sweep-Operation. Entlang des Druckpfades könnten Wegpunktweise, jeweils in einem Abstand gleich der Auflösung des Voxel-Grids, mit dem vorhergesagten Querschnitt alle umschlossenen Voxel „ausgefüllt“ werden (vgl. Abb. 6.7).

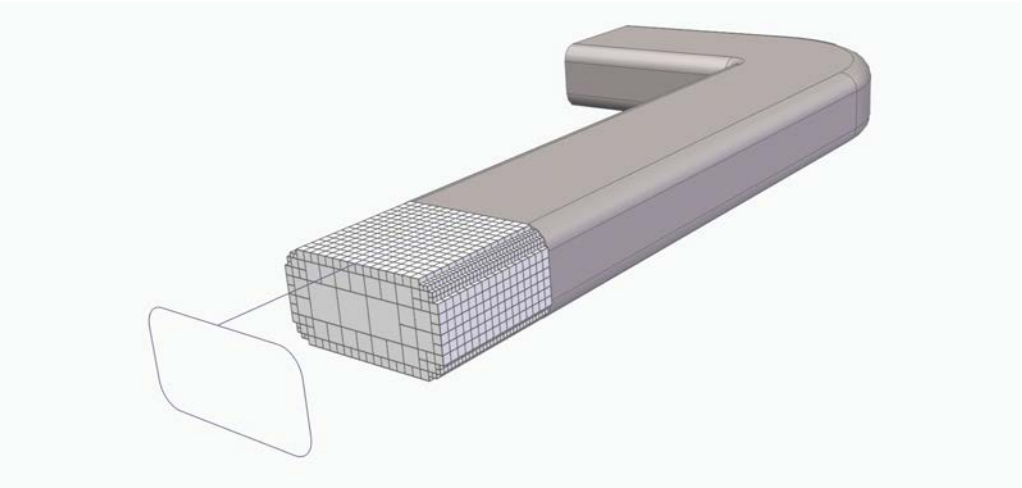


Abbildung 6.7: Darstellung eines möglichen Voxelmodells.

Auch wenn die eben beschriebene Durchführung der Pfadplanung mit anschließendem Ton-3D-Druck noch nicht optimal funktioniert, so ist die gezeigte Methode dennoch in der Lage einige der geforderten Fähigkeiten zu erfüllen. Es ist in jedem Fall möglich auf diese Weise Maschineninformationen aus einem BIM-Modell abzuleiten. Zudem konnten so einige hilfreiche Erkenntnisse gewonnen werden, unter anderem welche Informationen in einem Fertigungsmodell unbedingt verarbeitet werden müssen und in welcher Weise sich die verschiedenen Parameter gegenseitig beeinflussen. Im folgenden Kapitel wird daher ergänzend darauf eingegangen, wie die hier erstellten Informationen bestmöglich in den BIM-Workflow eingegliedert werden.

Zuletzt ist auch noch ein Softwareproblem zu nennen, dass mit der Verwendung von Dynamo aufgetreten ist. Die Geometrien der in Abb. 5.12 gezeigten Beispielobjekte konnten nicht alle in Dynamo eingelesen werden. Von den hier gezeigten Wandelementen konnte die schräge Freiformwand nicht verwendet werden, da Dynamo den Volumenkörper dieser Wand nicht extrahieren konnte. Nach einigen Versuchen hat sich hierzu herausgestellt, dass dieses Problem bei allen gekrümmten Wandelementen, die hierzu auch noch schräg geplant sind, auftritt. Das Problem ist, dass Revit und Dynamo unterschiedliche Geometriekernel verwenden und bei der Konvertierung der Daten in genau diesem Fall eine Inkompatibilität vorherrscht. Dieses Problem ist seit März 2018 ein immer noch offener Issue[1] auf github, damals für die Dynamo Version 2.0.0.4386 mit Revit 2018. In dieser Arbeit wurde die Dynamo Version 2.3.0.8352 mit Revit 2020 verwendet und immer noch existiert dieses Problem.

Auch eine Übertragung der Geometrie nach Rhino über das Datenformat IFC hat keine zufriedenstellenden Ergebnisse erzielt, da der IFC-Exporter von Revit diese Freiformwand nicht als B-Rep-Modell exportiert hat, sondern in Form eines Meshes und dieses auch noch in schlechter Qualität. Dieses Problem ist aber eben genau ein Hinweis darauf, dass nach Möglichkeit Datenkonvertierungen vermieden werden sollten, um ein optimales Ergebnis zu erhalten. Bei einer eventuellen Fortsetzung dieser Implementierung sollte

[1] https://github.com/DynamoDS/DynamoRevit/issues/1960

wenn möglich direkt die API von Revit verwendet werden, damit keine Konvertierung nötig ist. Ein weiterer Vorteil hierbei wäre, dass dann mit C#, also einer höheren Programmiersprache, gearbeitet wird, womit weitere Performance Vorteile einhergehen und bestimmte Codesequenzen deutlich übersichtlicher gestaltet werden können. Besonders interessant wäre es allerdings für die Pfadplanung, anstatt Revit oder ähnlicher Software, ein eigenes Tool zu entwickeln, dass direkt mit IFC-Dateien arbeiten kann. Ein solches Tool müsste entsprechend Geometriedaten aus einer IFC-Datei auslesen, diese in Druckerpfade umwandeln und diese Daten wieder richtig verlinkt in der IFC-Datei abspeichern.

Kapitel 7

Speicherung des Fertigungsinformationsmodells im IFC-Datenformat

Anhand der Informationen, die im vorhergehenden Kapitel erarbeitet wurden, ist die Idee entstanden, dass sich das IFC-Datenformat (vgl. Abschnitt 2.2) für eine Speicherung aller Fertigungsinformationen eignen könnte. Im Folgenden wird untersucht, ob sich dieses Datenformat tatsächlich eignet und damit eine Möglichkeit besteht, dass FIM direkt im BIM-Kontext eingesetzt werden kann.

Ein BIM-Modell ist eine sehr umfassende digitale Beschreibung eines Gebäudes mit besonderem Fokus auf die Interoperabilität der einzelnen Disziplinen, die an der Realisierung dieses digitalen Abbilds beteiligt sind (vgl. Abschnitt 2.1). Mit BIM kann auf der Grundlage von hochwertigen und miteinander gekoppelten geometrischen und semantischen Informationen viele Planungsprozesse automatisieren, wie z.B. Kollisionsprüfungen, die Erstellung von Leistungsverzeichnissen und das Ableiten von Bauplänen sowie Simulationsmodellen (PAOLINI et al., 2019).

Da auch die additive Fertigung grundsätzlich auf digitalen Modellen basiert, muss es prinzipiell möglich sein, BIM für die Planung von additiv gefertigten Bauteilen gewinnbringend einzusetzen. Das Datenformat IFC bietet hierzu einige Möglichkeiten, AM Parameter und Prozesse abzubilden. Im Folgenden wird zunächst beschrieben, welche Möglichkeiten IFC bereits bietet und welche Limitationen überwunden werden müssen und wie bestimmte Aspekte von AM integriert werden können. Allerdings konnte in dieser Arbeit noch keine Implementierung von einem solchen Datenmodell durchgeführt werden, es wird hier lediglich konzeptuell dargestellt.

7.1 Möglichkeiten mit IFC

Das Datenmodell, welches für die in dieser Arbeit vorgestellte Methodik, dem Fabrication Information Modelling (FIM), genutzt werden soll, muss in der Lage sein, alle entsprechend erzeugten Daten und Zwischenergebnisse (siehe auch Kapitel 4) abspeichern zu können. Im Folgenden werden entsprechend den Informationskategorien, die in Abschnitt 4.1 genannt werden, einzelne Elemente untersucht und eventuelle Möglichkeiten zur Abbildung in IFC vorgestellt. Besonders wichtig sind hierbei die essenziellen Prozessinformationen, die beiden anderen optionalen Kategorien müssen nicht zwingend umgesetzt werden, sondern können auch weiterhin über die in Abschnitt 5.8 beschriebenen Mechanismen

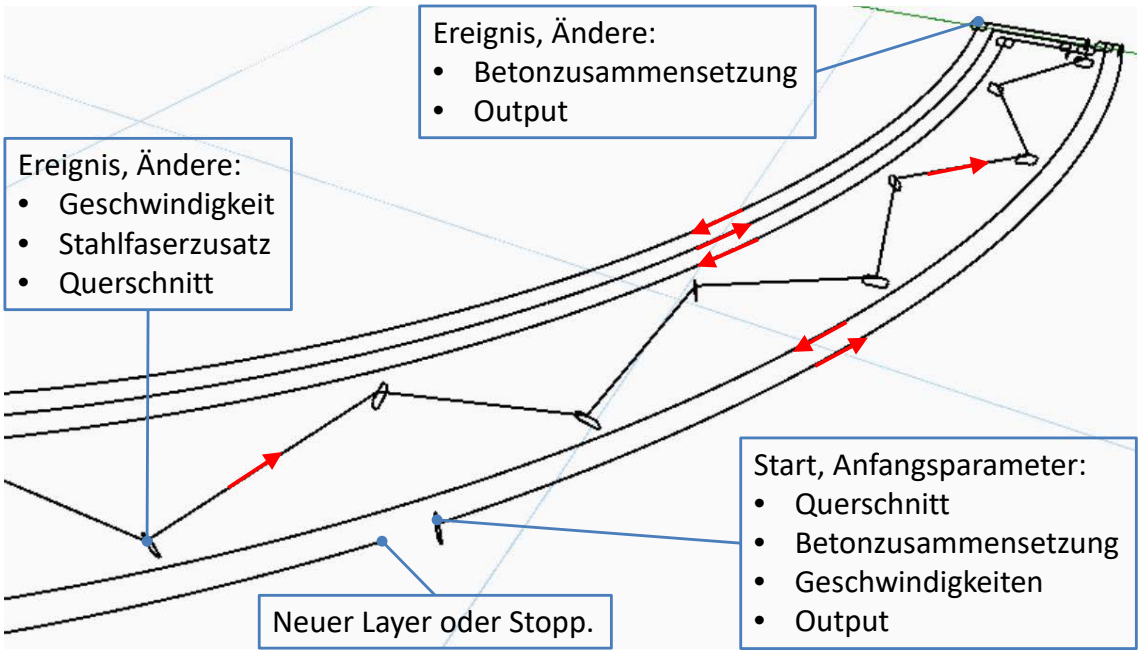


Abbildung 7.1: Lineare Referenzierung verschiedener 3D-Druck-Parameter entlang des Druckpfades. Die angegebenen Parameter könnten an bestimmten Wegpunkten über Ereignisse verändert werden oder gar über die gesamte Pfadlänge funktional abgebildet werden (vgl. Abb. 7.3).

exportiert werden. Generell ist in den folgenden Ausführungen die Idee, dass alle Fertigungsinformationen linear entlang dem Druckpfad referenziert dargestellt werden können, wie in Abb. 7.1 gezeigt wird.

7.1.1 Prozessinformationen

Unter diese Kategorie fallen sowohl die Material-, Prozess- und Maschinenparameter, der Druckpfad als auch der entsprechende Maschinensteuerungscode. Alle genannten Informationen sind hierbei von sehr unterschiedlicher Art und müssen gesondert voneinander betrachtet werden. Im Folgenden werden die entsprechenden Informationen einzeln beschrieben und im Anschluss die Zusammenhänge mittels eines Flussdiagramms visualisiert (vgl. Abb. 7.2).

Prozessparameter

Als übergeordneter Prozessparameter muss hier zunächst die AM-Methode genannt werden. Wie schon in Abschnitt 4.1.1 beschrieben, haben verschiedene AM-Methoden einen teilweise ganz eigenen Satz an beschreibenden Parametern. Daher ist es notwendig, die Methode als übergeordnete Entität zu definieren, mit der alle zu dieser Methode in Verbindung stehenden Parameter verknüpft werden können. Direkt mit der Methode wird eine Entität für das Maschinensystem verknüpft (nachfolgend beschrieben) und eine Relation zu dem Material des Bauteils erstellt.

Am wichtigsten ist hier aber die Entität, die den Druckpfad beschreiben soll, das `IfcAlinment`. Ein Alignment stellt hierbei eine Leitgeometrie dar, an der andere Strukturen linear referenziert werden können und wird durch eine Achsgeometrie, also einer Raumkurve, beschrieben. Diese Achse soll hier der im Dynamo-Skript erzeugten Polykurve, also dem geplanten Druckpfad entsprechen (vgl. Abschnitt 5.5) und kann z.B. über die Entität `IfcCompositeCurve`, der IFC-Version einer Polykurve, dargestellt werden.

Alle nicht konstanten Prozessparameter können dann entlang diesem Alignment linear referenziert abgebildet werden, z.B. über die Entität `IfcReferent`. Mit `IfcReferent` können „eventweise“ Parameterwerte angepasst werden, wobei zwischen zwei Events bzw. zwei Parameteranpassungen der Wert linear interpoliert wird (BORRMANN et al., 2017). Ein Anwendungsfall für dieses Konstrukt ist z.B. der Parameter Druckgeschwindigkeit und die damit zusammenhängende Extrusionsrate, wie in Abb. 7.4 verbildlicht wird. Andere Prozessparameter, wie z.B. die Schichthöhe oder die Größe der Extrusionsdüse, werden in dieser Arbeit als konstant über den gesamten Druckprozess definiert und können daher als einfache Properties definiert werden.

Letztlich muss noch die Maschinensteuerung erwähnt werden. Wie in Abschnitt 2.4.1 erläutert, wurde für das mit IFC verwandte Datenformat STEP bereits eine Erweiterung (STEP-NC) implementiert, mit der Maschinensteuerung durchgeführt werden kann. Aus diesem Grund ist es durchaus denkbar, dass dies auch für das Datenformat IFC durchgeführt werden kann. Jedoch gibt es bisher noch kaum Maschinensysteme, die mit STEP-NC gesteuert werden können, das Datenformat hat sich gegen den immer noch hauptsächlich eingesetzten G-Code nicht durchsetzen können. Ein starker Anreiz, eine ähnliche Implementierung für IFC durchzuführen, ist damit eher nicht gegeben. Zumal es mit den in Abschnitt 2.4.2 vorgestellten „robot frameworks“ (z.B. ROS) die Möglichkeit gibt, dass ein IFC-basiertes Steuerungsprogramm geschrieben werden könnte. Allerdings ist diese Aussage aktuell nur rein spekulativ und wurde in dieser Arbeit nicht bestätigt. Hier ist in jedem Fall eine weitere Untersuchung erforderlich und muss bei einer IFC-Implementierung des FIMs unbedingt durchgeführt werden.

Maschinenparameter

Diese Parameter sind allesamt unkompliziert, da alle entweder fixe Werte sind (Armlänge des Roboters, maximale Geschwindigkeit und Beschleunigung, etc.) oder limitierende Wertebereiche (maximale Achsauslenkungen). Aus diesem Grund könnten all diese Werte gesammelt in einem „Property Set“ abgespeichert werden.

Materialparameter

Diese Werte sind die am einfachsten zu realisierenden Parameter und müssen eigentlich gar nicht weiter erläutert werden. Materialmischungen und -eigenschaften werden auch in Projekten verarbeitet, die nichts mit AM zu tun haben. Entsprechende Entitäten für die Beschreibung des Materials sind daher bereits im IFC-Schema vorhanden (unter anderem

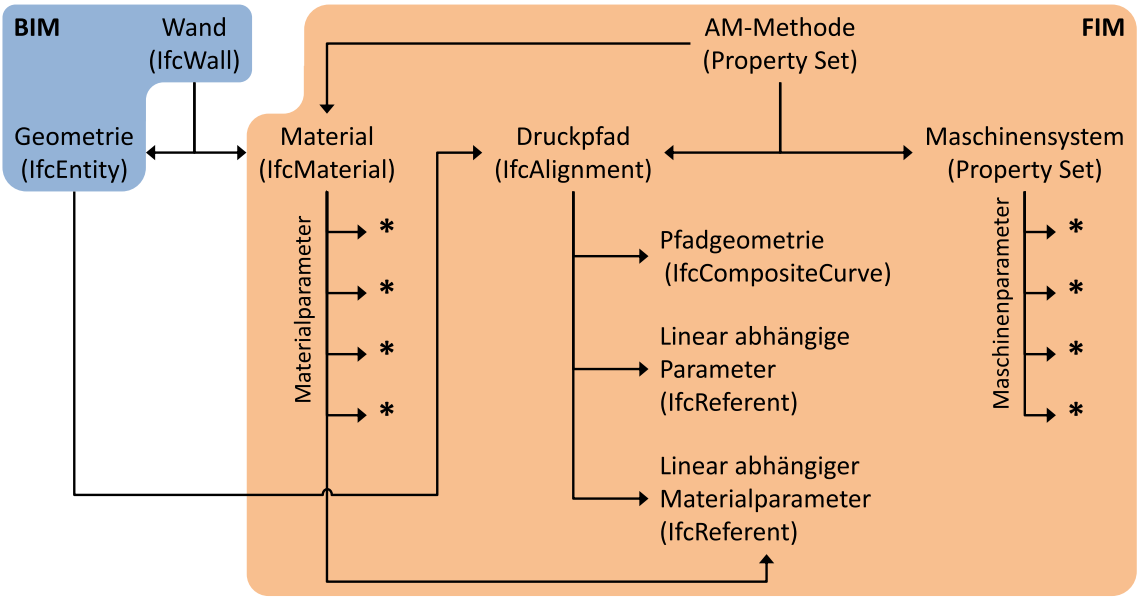


Abbildung 7.2: Vorgeschlagenes IFC-Schema in stark vereinfachter Form. Auf der Seite des BIM-Modells (hier blau dargestellt) wurde lediglich ein kleiner Ausschnitt dargestellt, nur die Geometrie.

`IfcMaterial`) und werden mit der Relation `IfcRelAssociatesMaterial` auf das Bauteil referenziert sowie durch `IfcMaterialProperties` genauer spezifiziert.

7.1.2 Volumenmodell und digitaler Zwilling

In Abschnitt 5.6 wurde für die Repräsentation der Materialverteilung nach dem Druckvorgang die Erstellung eines Volumenmodells über Sweep-Operationen untersucht. In IFC wäre diese Art von Volumenmodell durchaus darstellbar mit der Entität `IfcSweptAreaSolid`, es müsste hier nur ein zu sweepender Querschnitt verknüpft und eine Kurve gegeben werden, entlang der gesweept wird (Druckpfad). Wie aber in Abschnitt 6.2 beschrieben, gibt es mit der Erzeugung des Volumenkörpers auf diese Weise als auch mit der Umwandlung in ein simulationsfähiges Modell (Mesh) diverse Probleme. Daher wird eine Verwendung eines solchen Modells nicht weiter in Betracht gezogen.

Eine Alternative ist, wie ebenfalls in Abschnitt 6.2 beschrieben, ein Voxelmodell. Diese Art von Volumenmodell wird allerdings nicht im IFC-Standard unterstützt und kann daher nicht auf diese Weise gespeichert werden. Bei der Erstellung dieses Modells könnte allerdings Querschnittweise, also Scheibe für Scheibe, eine Referenzierung zum Druckpfad durchgeführt und so eine mögliche Schnittstelle zum IFC-Modell ermöglicht werden. Dies konnte aber im Rahmen dieser Arbeit nicht mehr durchgeführt werden.

7.2 Analogie zu Straßenbauprojekten

Ein 3D-Druckprozess kann im Prinzip ganz ähnlich zu einem Straßenbauprojekt geplant werden. Hier werden entlang einer geplanten Trasse verschiedene sogenannte *„Bänder“*

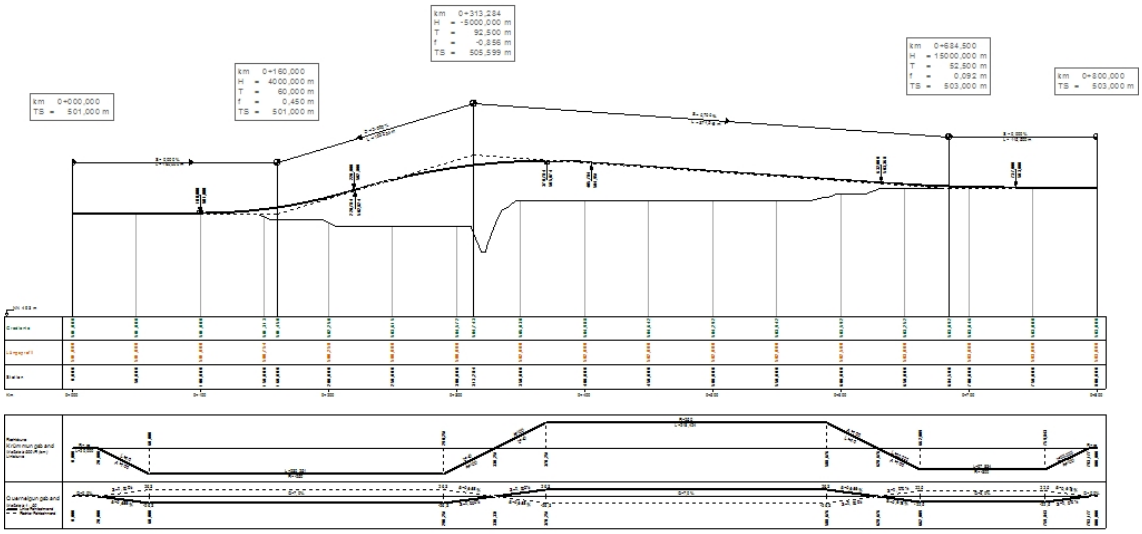


Abbildung 7.3: Beispiel Längsschnitt eines Straßenplans, hier werden entlang der Trasse Höhe, Querneigung und Krümmung aufgetragen. In einem 3D-Druckprojekt könnten so Geschwindigkeit, Exstrusionsrate, Materialmischung, etc. angegeben werden.

und Längsschnitte definiert, um bestimmte Eigenschaften der geplanten Straße abzubilden (vgl. Abb. 7.3). Z.B. gibt es hier das Querneigungsband, welches die Querneigung der Straße entlang der Mittelachse angibt, oder den Höhenplan, ein Längsschnitt, der jedem Punkt der Trasse eine Höhe zuweist. Die zuvor beschriebene lineare Referenzierung der Prozess und Materialparameter entlang des Kontrollfeatures Druckpfad kann prinzipiell ganz ähnlich gehandhabt werden wie für eine Straßenplanung.

Dies kann anhand eines Beispiels, wie eine IFC-Implementierung der Querneigung aussehen könnte, noch mal im Detail gezeigt werden. Abbildung 7.4 zeigt hierzu Querneigungswerte einer Schienentrasse. An vier referenzierten Punkten der Trasse (11-00 bis 14-00)

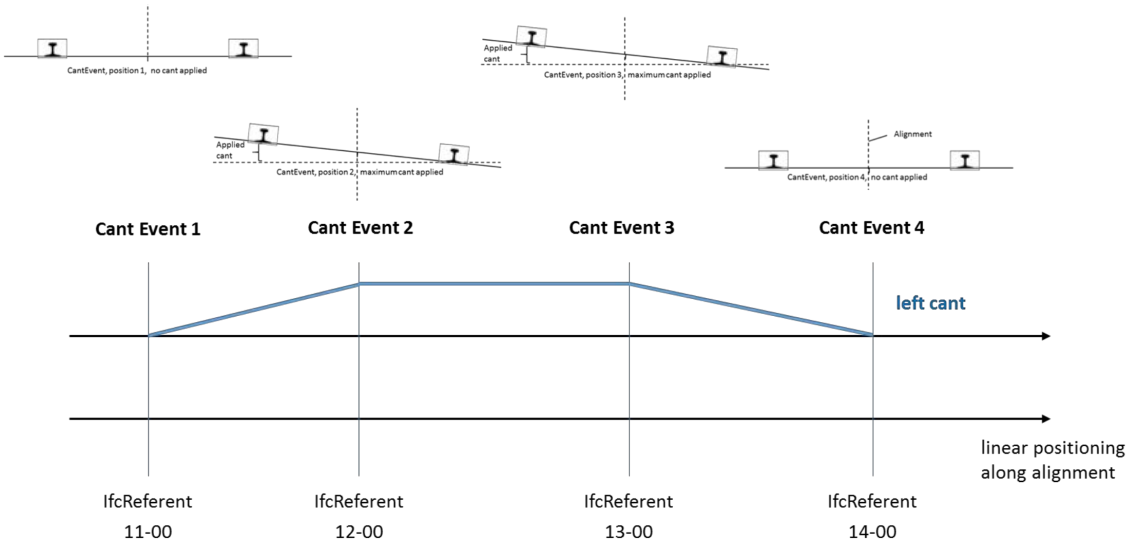


Abbildung 7.4: Straßenbau Beispiel: Querneigung einer Bahntrasse (BORRMANN et al., 2017).

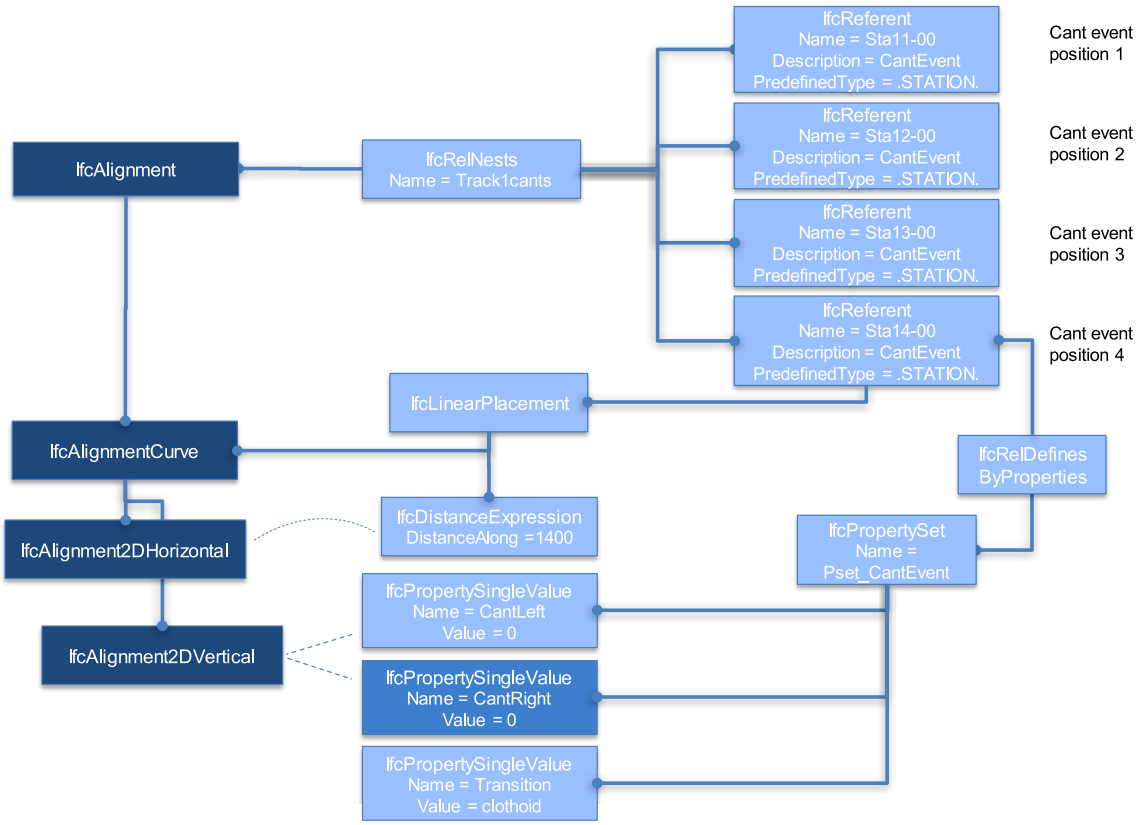


Abbildung 7.5: Straßenbau Beispiel: IFC-Realisierung der Querneigung (BORRMANN et al., 2017).

soll der Querneigungsverlauf geändert werden. Es wird hier ein Wert für die entsprechende Position gegeben und die anderen Werte zwischen den Positionen linear interpoliert. In Abb. 7.5 wird genau diese Situation abgebildet. Für jedes dieser Events, an denen die Querneigung geändert wird, wird jeweils ein `IfcReferent` Objekt erstellt und hier der entsprechende Querneigungswert eingestellt.

Im Fall der Prozessparameter ist ein ganz analoges Beispiel denkbar: Wie in Abschnitt 5.3.3 gezeigt, kann es zu Problemen bei der Druckausführung von Pfaden mit vielen starken Richtungsänderungen geben. Als eine mögliche Lösung wurde vorgeschlagen, dass an derartigen kritischen Stellen die Geschwindigkeit angepasst wird. Es könnten hier also ebenfalls in gleicher Weise wie beim Straßenbau Beispiel Änderungsevents mit `IfcReferent` Objekten realisiert werden.

Im IFC-Format wird aktuell sehr ausgiebig an Implementierungen für den Infrastrukturbereich gearbeitet und seit dem Release 4.1 (vgl. Abschnitt 2.2) wurden wie gerade dargestellt einige Erweiterungen geschaffen, die auch für den 3D-Druck interessant sein könnten. Unter anderem ist hier die schon zuvor beschriebene Entität `IfcAlignment` interessant, sie stellt ein Referenzierungssystem für lineare Bauwerke dar, also z.B. Straßen, und könnte somit verwendet werden, um den Druckpfad abzubilden. Parameterwerte können hier mit *Annotations* (`IfcAnnotation`) oder über die zuvor beschriebene Entität `IfcReferent` referenziert werden oder alternativ über *Offset-Kurven* (`IfcOffsetCurvesByDistance`) dargestellt werden. Die letztgenannte Entität, Offset-Kurve, kann auch dazu verwendet

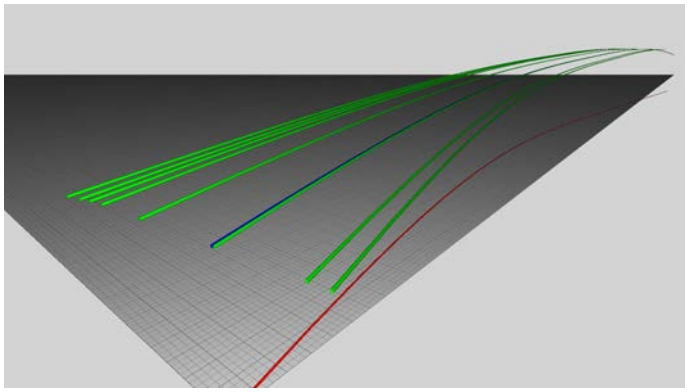


Abbildung 7.6: Offset-Kurven. Auf diese Weise könnten unter anderem die tatsächlich abgefahrene Bahn im Vergleich zum geplanten Pfad (Drucker-Feedback) oder die separaten Bahnen eines Druckers mit mobiler Basis oder mit Multidüsensystem dargestellt werden. (BUILDINGSMART, 2020b)

werden, um Feedback-Daten vom Drucker (z.B. gemessene Positionsdaten des Druckkopfs) abzuspeichern, wie z.B. der *„as-printed“*-Druckpfad für Soll-Ist Vergleiche. Oder es können während dem Druck Anpassungen auf Basis von Laserscans vorgenommen werden und als Offset gespeichert werden, wenn z.B. Druckfehler „on the fly“ ausgeglichen werden sollen. Zudem ist es möglich, dass mit diesen Offsets verschiedene TCP-Positionen dargestellt werden, um so beispielsweise die separate Bewegung einer mobilen Basis oder um einen Druckkopf mit mehreren Druckdüsen darzustellen. Genauso kann bei der Shotcrete-Methode die Düsenposition relativ zu der Position, an der das Material aufgetragen wird, dargestellt werden. Für Partikelbettmethoden, die ein Multidüsensystem verwenden, kann so jede Düsenposition über eine eigene Offset-Kurve beschrieben werden und zu jeder dieser Kurven eine eigene Parameterreferenzierung vorgenommen werden (vgl. Abb. 7.6).

Kapitel 8

Zusammenfassung und Ausblick

In dieser Arbeit konnten alle Fragestellungen, die in Kapitel 1 aufgestellt wurden, beantwortet werden, jedoch haben sich im Laufe der Bearbeitung immer wieder neue Fragen aufgetan. Im Folgenden werden einige dieser Fragen aufgegriffen und eine mögliche zukünftige Herangehensweise zu deren Lösung vorgeschlagen.

Das in Abschnitt 6.2 beschriebene Problem mit der Sweep-Geometrie muss von Grund auf neu angesetzt werden. Die in dieser Arbeit untersuchte Herangehensweise ist leider wenig erfolgversprechend und eignet sich kaum für den Export von Simulationsmodellen. Wie beschrieben wäre es eine bessere Idee, direkt eine Voxel-Repräsentation des Druckkörpers aus dem Druckpfad abzuleiten. Mit einem derartigen Modell können je nach Voxelgröße sehr detailliert Materialverteilungen darstellen. Wie genau aber das Voxelmodell erstellt und die entsprechenden Daten mit dem Druckpfad und den anderen im FIM erzeugten Informationen verlinkt werden kann, muss noch untersucht werden. Wird aber sowohl das auf dem Druckpfad basierende Volumenmodell, als auch der digitale Zwilling im Voxel-Format erstellt, ist es möglich, auf sehr einfache Weise beide Modelle, also das Soll und das Ist, miteinander zu vergleichen.

Als Robotersteuerungsprogramm wurde in dieser Arbeit Polyscope verwendet, das ausschließlich zur Steuerung von Robotern der Firma *Universal Robots* gedacht ist. Eine Verbesserung hierzu wäre es, mit ROS zu arbeiten, da hiermit weitestgehend herstellerunabhängig Steuerungscode entwickelt werden kann und das System eine immer größere Auswahl an Robotersysteme unterstützt. Zusätzlich würde es ROS ermöglichen, dass neben dem Robotersystem gleichzeitig im selben Programm auch weitere Peripheriegeräte gesteuert werden. Dies würde eine Koordination der im FIM erzeugten Prozessparameter seitens der Maschinensteuerung deutlich erleichtern.

Für weitere Laborversuche gibt es zudem noch die Möglichkeit, ein Raspberry Pi mit einer ROS-Installation auszustatten und zusätzlich die GPIO Ausgänge für die Steuerung des Ton-Extruders vorzubereiten. Damit kann gleichzeitig der im Laborversuch (vgl. Kapitel 6) verwendete UR10e Roboter sowie der Extruder über das PI gesteuert werden kann. Auf diese Weise könnten in Zukunft auch deutlich komplexere Pfadplanungen im Versuch getestet werden, z.B. mit adaptivem Geschwindigkeitsprofil. Zudem kann so sehr einfach weitere Peripherie angeschlossen werden, wie etwa ein Laserscanner oder eine Kamera zur Erstellung eines digitalen Zwillings. Mit einem solchen Modell, das die genaue Materialverteilung des Bauteils, so wie es gedruckt wurde, darstellt, können weitere Simulationen durchgeführt werden, um die Qualität zu prüfen oder und um weitere Erkenntnisse für zukünftige Projekte zu sammeln (siehe „Digital Twin“ in Abb. 8.1). In Kombination mit einem Laserscanner könnte auch eine Autokorrekturfunktion implementiert werden, um

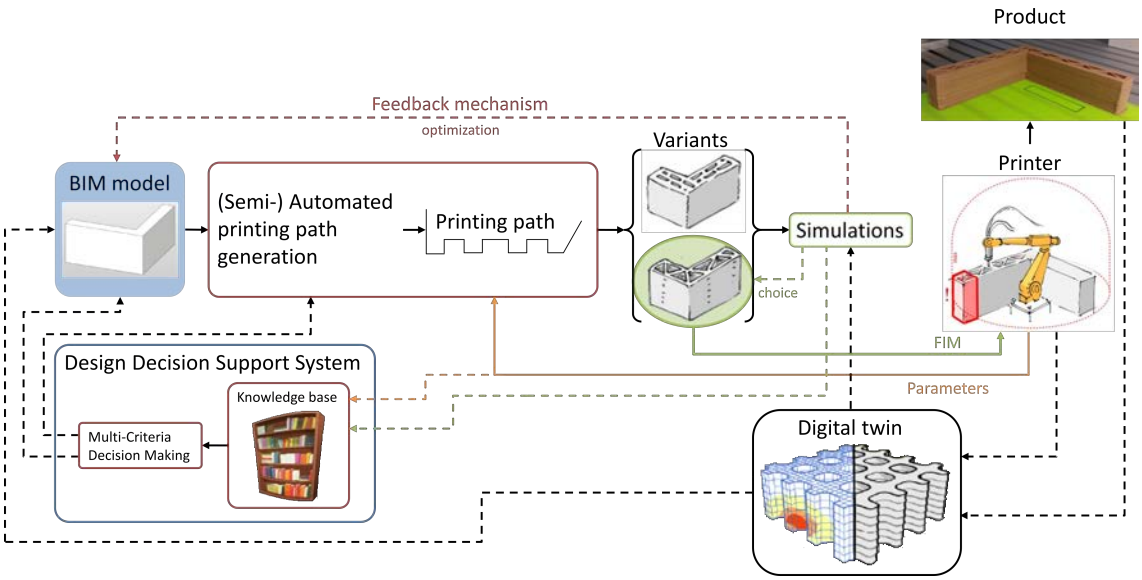


Abbildung 8.1: Erweiterungen zum Workflow aus Abschnitt 5.2.

z.B. die Extrusionsrate automatisch zu korrigieren, falls diese zu niedrig oder zu hoch geplant wurde.

In dieser Arbeit wurde zudem nur ein Ausschnitt der unzähligen Möglichkeiten von AM betrachtet. Es wurde hier nur die Betonextrusionsmethode (3DCP) genauer untersucht und allenfalls Parallelen zu anderen Methoden gezogen. Daher muss in Zukunft auch noch auf die vielen anderen Methoden (vgl. Abschnitt 2.3) eingegangen werden und versucht werden herauszufiltern, wo es Gemeinsamkeiten gibt, auf welche Unterschiede man achten muss und welche Auswirkung dies auf die Gestaltung eines Fertigungsmodells hat. Auch die vielen verschiedenen Robotersysteme (vgl. Abschnitt 2.3.4) sind hier genauer zu untersuchen und mit in die automatische Pfadplanung zu integrieren. Für das unterschiedliche Materialverhalten muss in Zukunft noch intensiver mit den entsprechenden Forschungsprojekten zusammengearbeitet werden, um die limitierenden Effekte bestimmter Materialparameter mit in die Planung einfließen zu lassen (siehe gelber Pfeil in Abb. 8.1).

Besonders wichtig ist aber auch zu untersuchen, wie sich das in dieser Arbeit gezeigte Pfadplanungstool in den Designprozess mit eingliedern lässt, wenn simulationsgestützt iterativ Anpassungen vorgenommen werden. Hierzu muss eingehend untersucht werden, wie bestmöglich eine geschlossene Kommunikationskette zwischen Design, Pfadplanung und Simulation ermöglicht werden kann (siehe „Feedback mechanism“ und DDSS in Abb. 8.1). Hierbei soll mithilfe einer Wissensdatenbank, also mit bereits gesammelter Erfahrung, die nötige Kommunikation so weit wie möglich reduziert werden sowie die Qualität des Austausches durch hochwertige Modelle gesteigert werden.

Literaturverzeichnis

ABUALDENIEN, J., SCHNEIDER-MARIN, P., ZAHEDI, A., HARTER, H., EXNER, H., STEINER, D., SINGH, M. M., BORRMANN, A., LANG, W., PETZOLD, F. et al. (2020). Consistent management and evaluation of building models in the early design stages. *ITcon*, *25*, 212–232.

Additive Manufacturing Processes Cheat Sheet [Draft]. (2020). *Additive Manufacturing Processes*, 1. https://doi.org/10.31399/asm.hb.v24.9781627082907

AHRBERG, R., BARFELS, L. & VOSS, J. (2011). *Programmierung von Werkzeugmaschinen*. https://doi.org/10.1007/978-3-8348-9898-2_15

BARTENSCHLAGER, J., HEBEL, H. & SCHMIDT, G. (2013). *Handhabungstechnik mit Robotertechnik: Funktion, Arbeitsweise, Programmierung*. Springer-Verlag.

BAULINKS. (2020). PERI druckt im Herbst 2020 Deutschlands erstes Einfamilienhaus aus Beton. Verfügbar 20. Oktober 2020 unter https://www.baulinks.de/webplugin/2020/1701.php4

BIM Blog. (2015). Verfügbar 20. September 2020 unter https://bimblog.typepad.com/autodesk_bim_blog/2015/08/was-ist-dynamo-f%C3%BCr-revit-.html

BIM FORUM. (2019). Level of Development (LOD) Specification Part I & Commentary. *Bim-Bep*, (April), 254. https://bimforum.org/lod/

BIMpedia. (2020). Verfügbar 20. September 2020 unter https://www.bimpedia.eu/-/1005-lod-___-level-of-development

BORRMANN, A., KOLBE, T., DONAUBAUER, A., STEUER, H., JUBIERRE, J. & FLURL, M. (2015a). Multi-Scale Geometric-Semantic Modeling of Shield Tunnels for GIS and BIM Applications. *Computer-Aided Civil and Infrastructure Engineering*, *30*(4), 263–281. https://doi.org/https://doi.org/10.1111/mice.12090

BORRMANN, A., AMANN, J., CHIPMAN, T., HYVÄRINEN, J., LIEBICH, T., MUHIC, S., MOL, L., PLUME, J. & SCARPONCINI, P. (2017). IFC Infra Overall Architecture Project Documentation and Guidelines. *BuildingSMART*, 1–53. https://www.buildingsmart.org/wp-content/uploads/2017/07/08%7B%5C_%7DbSI%7B%5C_%7DOverallArchitecure%7B%5C_%7DGuidelines%7B%5C_%7Dfinal.pdf

BORRMANN, A., KÖNIG, M., KOCH, C. & BEETZ, J. (2015b). *Building Information Modeling: Technologische Grundlagen und industrielle Praxis*. Springer-Verlag.

BOSSCHER, P., WILLIAMS, R. L., BRYSON, L. S. & CASTRO-LACOUTURE, D. (2007). Cable-suspended robotic contour crafting system. *Automation in Construction*, *17*(1), 45–55. https://doi.org/10.1016/j.autcon.2007.02.011

BRANDSTÖTTER, M., ANGERER, A. & HOFBAUR, M. (2014). An Analytical Solution of the Inverse Kinematics Problem of Industrial Serial Manipulators with an Ortho-parallel Basis and a Spherical Wrist, 7–11.

BUILDINGSMART. (2020a). IFC Formats. Verfügbar 20. September 2020 unter https://technical.buildingsmart.org/standards/ifc/ifc-formats/

buildingSMART. (2020b). Resource definition data schemas. Verfügbar 20. November 2020 unter https://standards.buildingsmart.org/IFC/DEV/IFC4_3/RC2/HTML/

Buswell, R. A., Leal de Silva, W. R., Jones, S. Z. & Dirrenberger, J. (2018). 3D printing using concrete extrusion: A roadmap for research. *Cement and Concrete Research*, *112*(October 2017), 37–49. https://doi.org/10.1016/j.cemconres.2018.05.006

Carlsson, B. (1984). The development and use of machine tools in historical perspective. *Journal of Economic Behavior & Organization*, *5*(1), 91–114.

Carneau, P., Baverel, O., Mesnil, R. & Roussel, N. (2019). An exploration of 3d printing design space inspired by masonry. *IASS Annual Symposium 2019 - Structural Membranes 2019*, (October), 1–10. https://doi.org/10.5281/zenodo.3563672

CC Corp. (2017). Introducing Contour Crafting Technology. Verfügbar 20. Oktober 2020 unter https://www.contourcrafting.com/

Chanatry, W. (2018). NX Continuous Release F.A.Q. Verfügbar 18. Oktober 2020 unter https://blogs.sw.siemens.com/nx-design/NX-Continuous-Release-F-A-Q/

Chen, H., Wei, Q., Zhang, Y., Chen, F., Shi, Y. & Yan, W. (2019). Powder-spreading mechanisms in powder-bed-based additive manufacturing: Experiments and computational modeling. *Acta Materialia*, *179*, 158–171. https://doi.org/10.1016/j.actamat.2019.08.030

Chiu, M. L. (2002). An organizational view of design communication in design collaboration. *Design Studies*, *23*(2), 187–210. https://doi.org/10.1016/S0142-694X(01)00019-9

Crotty, S. (2020). Revit Public Roadmap Update – April 2020. Verfügbar 20. September 2020 unter https://blogs.autodesk.com/revit/2020/04/20/revit-public-roadmap-update-april-2020/

Denavit, J. & Hartenberg, R. S. (1955). A kinematic notation for lower pair mechanisms based on matrices. *Journal of Applied Mechanics*, *77*, 215–221.

Dini, E. (n. d.). D-Shape is a particle bed 3D printing process at architectural scale. Verfügbar 20. Oktober 2020 unter https://d-shape.com/what-is-it/

dynamobim. (2016). New to Dynamo? Verfügbar 20. September 2020 unter https://dynamobim.org/learn/

Eastman, C. M., Eastman, C., Teicholz, P., Sacks, R. & Liston, K. (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons.

Fu, C., Aouad, G., Lee, A., Mashall-Ponting, A. & Wu, S. (2006). IFC model viewer to support nD model application. *Automation in Construction*, *15*(2), 178–185.

GeometryGym. (2020). IFC for Rhino/Grasshopper. Verfügbar 20. September 2020 unter https://geometrygym.wordpress.com/bim/#ifcrhino

Glossar. (2020). Verfügbar 20. September 2020 unter https://glossar.item24.com/glossarindex/artikel/item/extrudieren-kunststoff.html

Gramazio Kohler. (2020). Compas Fab. Verfügbar 20. Oktober 2020 unter https://gramaziokohler.github.io/compas_fab/latest/examples/01_fundamentals/02_coordinate_frames.html

GRASSHOPPER3D. (2020). Tutorials. Verfügbar 20. September 2020 unter https://www.grasshopper3d.com/page/tutorials-1

HAWKINS, K. P. (2013). *Analytic inverse kinematics for the universal robots UR-5/UR-10 arms* (Techn. Ber.). Georgia Institute of Technology.

HEHENBERGER, P. (2020). CNC-Technik und Programmierung. *Computerunterstützte Produktion* (S. 86–117). Springer.

HENKE, K., TALKE, D. & WINTER, S. (2016). *Additive Fertigung frei geformter Bauelemente durch numerisch gesteuerte Extrusion von Holzleichtbeton* (Techn. Ber.). Forschungsinitiative Zukunft Bau Bundesinstitut für Bau-, Stadt- und Raumforschung (BBSR). München, Germany.

HENKE, K. (2016). *Additive Baufertigung durch Extrusion von Holzleichtbeton* (Diss.).

HENKE, K., TALKE, D. & MATTHÄUS, C. (2020). Additive Manufacturing by Extrusion of Lightweight Concrete - Strand Geometry, Nozzle Design and Layer Layout. *RILEM Bookseries*, *28*, 906–915. https://doi.org/10.1007/978-3-030-49916-7_88

HOOPER, M. (2015). Automated model progression scheduling using level of development. *Construction Innovation*.

IFC Release Notes. (2020). Verfügbar 20. September 2020 unter https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ifc-release-notes/

ILIAN, BONEV. (2019). What are Singularities in a Six-Axis Robot Arm? Verfügbar 20. November 2020 unter https://www.mecademic.com/en/what-are-singularities-in-a-six-axis-robot-arm

JEONG, H., HAN, S. J., CHOI, S. H., LEE, Y. J., YI, S. T. & KIM, K. S. (2019). Rheological property criteria for buildable 3D printing concrete. *Materials*, *12*(4), 1–21. https://doi.org/10.3390/ma12040657

KAMARDEEN, I. (2010). 8D BIM modelling tool for accident prevention through design. *Association of Researchers in Construction Management, ARCOM 2010 - Proceedings of the 26th Annual Conference*, (September), 281–289.

KEBRIA, P. M., AL-WAIS, S., ABDI, H. & NAHAVANDI, S. (2016). Kinematic and dynamic modelling of UR5 manipulator. *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 004229–004234. https://doi.org/10.1109/SMC.2016.7844896

KHOSHNEVIS, B. (1999). Contour crafting-state of development. *Solid Freeform Fabrication Proceedings*, 743–750.

KLOFT, H., GEHLEN, C., HACK, N., HENKE, K., LOWKE, D. & MAINKA, J. (2020). *Additive Manufacturing in Construction (AMC) – The Challenge of Large Scale* (Techn. Ber.). https://www.tu-braunschweig.de/trr277/trr-277-home

KRUSE, A. (2019). *BIM-gestützte additive Fertigung* (Master thesis September).

KUKA. (2017). KUKA – Industrierobotik, mittlere Traglast. Verfügbar 20. Oktober 2020 unter https://www.kuka.com/-/media/kuka-downloads/imported/9cb8e311bfd744b4b0eab25ca883f6d3/kuka_pb_mittlere_tl_de.pdf

LANGENBERG, E. (2015). Mapping 20 years of 3D printing in Architecture. Verfügbar 20. Oktober 2020 unter https://www.elstudio.nl/?p=1904

LEE, J., BAGHERI, B. & KAO, H.-A. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23. https://doi.org/10.1016/j.mfglet.2014.12.001

Level of Development. (2020). Verfügbar 20. September 2020 unter https://www.hitechcaddservices.com/bim/support/level-of-development-lod/

LIEBICH, T., ADACHI, Y., FORESTER, J., HYVARINEN, J., RICHTER, S., CHIPMAN, T., WEISE, M. & WIX, J. (2013). Industry Foundation Classes IFC4 official release. Verfügbar 20. September 2020 unter https://standards.buildingsmart.org/IFC/RELEASE/IFC4/FINAL/HTML/

MACLEAMY, P. (2004). MacLeamy curve. *Collaboration, Integrated Information, and the Project Lifecycle in Building Design and Construction and Operation (WP-1202)*.

MATAERIAL. (2016). A radically new 3D printing method. Verfügbar 20. Oktober 2020 unter http://www.mataerial.com/

McNeel Wiki. (2020). Verfügbar 20. September 2020 unter https://wiki.mcneel.com/rhino/rhinohistory

MECHTCHERINE, V., BOS, F. P., PERROT, A., da SILVA, W. R., NERELLA, V. N., FATAEI, S., WOLFS, R. J., SONEBI, M. & ROUSSEL, N. (2020). Extrusion-based additive manufacturing with cement-based materials – Production steps, processes, and their underlying physics: A review. *Cement and Concrete Research*, *132*(December 2019), 106037. https://doi.org/10.1016/j.cemconres.2020.106037

MECHTCHERINE, V. & NERELLA, V. N. (2019). Beton-3D-Druck durch selektive Ablage: Anforderungen an Frischbeton und Materialprüfung. *Beton- und Stahlbetonbau*, *114*(1), 24–32. https://doi.org/10.1002/best.201800073

NÄTHER, M., NERELLA, V. N., KRAUSE, M., KUNZE, G., MECHTCHERINE, V. & SCHACH, R. (2017). *Beton-3D-Druck -Machbarkeitsuntersuchungen zu kontinuierlichen und schalungsfreien Bauverfahren durch 3D-Formung von Frischbeton* (Techn. Ber.).

NX. Answers for Industry. (2013). Verfügbar 20. September 2020 unter https://www.computerkomplett.de/assets/uploads/files/Produktbroschuere_NX_4639_tcm73-1423.pdf

O'CONNOR, A. C., DETTBARN, J. L. & GILDAY, L. T. (2004). Cost Analysis of Inadequate Interoperability in the US Capital Facilities Industry.

OGURA, H., NERELLA, V. N. & MECHTCHERINE, V. (2018). Developing and testing of Strain-Hardening Cement-Based Composites (SHCC) in the context of 3D-printing. *Materials*, *11*(8), 1–18. https://doi.org/10.3390/ma11081375

PA. (2019a). 3D Printed Earth Wall with Embedded Staircase. Verfügbar 20. September 2020 unter https://parametric-architecture.com/3d-printed-earth-wall-with-embedded-staircase-wasp-iaac/

PA. (2019b). 3D Printing Concrete by Luai Kurdi. Verfügbar 20. September 2020 unter https://parametric-architecture.com/3d-printing-concrete-by-luai-kurdi/

PAOLINI, A., KOLLMANNSBERGER, S. & RANK, E. (2019). Additive manufacturing in construction: A review on processes, applications, and digital planning methods. *Additive Manufacturing*, *30*(July), 100894. https://doi.org/10.1016/j.addma.2019.100894

PEASE, W. (1952). An automatic machine tool. *Scientific American*, *187*(3), 101–115.

POTT, A. & DIETZ, T. (2019). *Industrielle Robotersysteme*. https://doi.org/10.1007/978-3-658-25345-5

RAMLAB. (2017). RAMLAB unveils world's first class approved 3D printed ship's propeller. Verfügbar 20. Oktober 2020 unter https://ramlab.com/updates/ramlab-unveils-worlds-first-class-approved-3d-printed-ships-propeller/

RAMOS JUBIERRE, J. (2017). *Consistency preservation methods for multi-scale design of subway infrastructure facilities* (Dissertation). Technische Universität München. München.

Revit History. (2019). Verfügbar 20. September 2020 unter https://autodesk.blogs.com/between_the_lines/revit-history.html

RODRIGUES, T. A., DUARTE, V., MIRANDA, R., SANTOS, T. G. & OLIVEIRA, J. (2019). Current status and perspectives on wire and arc additive manufacturing (WAAM). *Materials*, *12*(7), 1121.

RODRIGUEZ, E. & ALVARES, A. (2019). A STEP-NC implementation approach for additive manufacturing. *Procedia Manufacturing*, *38*(2019), 9–16. https://doi.org/10.1016/j.promfg.2020.01.002

SCHEER, A.-W. (2013). *Wirtschaftsinformatik Studienausgabe: Referenzmodelle für industrielle Geschäftsprozesse*. Springer-Verlag.

SIEGERT, H.-J. & BOCIONEK, S. (2013). *Robotik: Programmierung intelligenter Roboter: Programmierung intelligenter Roboter*. Springer-Verlag.

SIEMENS. (2016). *Getting Started with NX Open*. https://docs.plm.automation.siemens.com/data_services/resources/nx/11/nx_api/common/en_US/graphics/fileLibrary/nx/nxopen/nxopen_getting_started_v11.pdf

Siemens Closes Acquisition of UGS. (2007). Verfügbar 20. September 2020 unter https://www.plm.automation.siemens.com/global/en/our-story/newsroom/siemens-press-release/43058

SINGH, H. & PATEL, K. (2020). How 5D BIM Improves Efficiencies in Construction Projects. Verfügbar 20. Oktober 2020 unter https://www.truecadd.com/news/how-5d-bim-fuels-the-growth-of-construction-industry

STEP TOOLS. (2007). The STEP-NC AP238 Standard. Verfügbar 20. Oktober 2020 unter https://www.steptools.com/stds/stepnc/

STEVENSON, K. (2020). A 9-Axis Concrete 3D Printer That Can Do Amazing Things. Verfügbar 20. September 2020 unter https://www.fabbaloo.com/blog/2020/3/16/a-9-axis-concrete-3d-printer-that-can-do-amazing-things

SUIKER, A. S., WOLFS, R. J., LUCAS, S. M. & SALET, T. A. (2020). Elastic buckling and plastic collapse during 3D concrete printing. *Cement and Concrete Research*, *135*(June), 106016. https://doi.org/10.1016/j.cemconres.2020.106016

TAKAHASHI, H. & MIYASHITA, H. (2017). Expressive fused deposition modeling by controlling extruder height and extrusion amount. *Conference on Human Factors in Computing Systems - Proceedings*, *2017-May*, 5065–5074. https://doi.org/10.1145/3025453.3025933

TAO, F., ZHANG, H., LIU, A. & NEE, A. Y. (2019). Digital Twin in Industry: State-of-the-Art. *IEEE Transactions on Industrial Informatics*, *15*(4), 2405–2415. https://doi.org/10.1109/TII.2018.2873186

TEICHERT, P. (2002). Carl Akeley—A tribute to the founder of shotcrete. *Shotcrete Magazine*, *4*(3), 10–12.

THOMPSON, S. M., BIAN, L., SHAMSAEI, N. & YADOLLAHI, A. (2015). An overview of Direct Laser Deposition for additive manufacturing; Part I: Transport phenomena, modeling and diagnostics. *Additive Manufacturing*, *8*, 36–62. https://doi.org/10.1016/j.addma.2015.07.001

UR. (2020a). Getting Joint angles of Inverse Kinematics URCap. Verfügbar 20. Oktober 2020 unter https://forum.universal-robots.com/t/getting-joint-angles-of-inverse-kinematics-urcap-another-way-for-get-inverse-kin/8504

UR. (2020b). Overview of client interfaces. Verfügbar 20. November 2020 unter https://www.universal-robots.com/articles/ur/overview-of-client-interfaces/

UR. (2020c). Parameters for calculations of kinematics and dynamics. Verfügbar 20. Oktober 2020 unter https://www.universal-robots.com/articles/ur/parameters-for-calculations-of-kinematics-and-dynamics/

UR. (2020d). Polyscope Manual. Verfügbar 20. November 2020 unter https://s3-eu-west-1.amazonaws.com/ur-support-site/44018/Software%7B%5C_%7DManual%7B%5C_%7Den%7B%5C_%7DGlobal.pdf

VILGERTSHOFER, S. & BORRMANN, A. (2017). Using graph rewriting methods for the semi-automatic generation of parametric infrastructure models. *Advanced Engineering Informatics*, *33*, 502–515. https://doi.org/10.1016/j.aei.2017.07.003

VisualARQ Features. (2020). Verfügbar 20. September 2020 unter https://www.visualarq.com/de/features/collaboration/ifc/

WECK, M. (2006). *Werkzeugmaschinen 4: Automatisierung von Maschinen und Anlagen*. Springer-Verlag.

WEGER, D., GEHLEN, C. & LOWKE, D. (2018). Additive Fertigung von Betonbauteilen durch selektive Zementleim-Intrusion.

WEI, C., LI, L., ZHANG, X. & CHUEH, Y. H. (2018). 3D printing of multiple metallic materials via modified selective laser melting. *CIRP Annals*, *67*(1), 245–248. https://doi.org/10.1016/j.cirp.2018.04.096

WINSUN. (2020). Coronavirus: WinSun 3D-druckt Quarantänestationen. Verfügbar 20. September 2020 unter https://3druck.com/case-studies/coronavirus-winsun-quarantaenestationen-0688273/

WOHLERS, T. & GORNET, T. (2014). History of additive manufacturing. *Wohlers report*, *24*(2014), 118.

WOLFS, R., BOS, F. & SALET, T. (2019). Hardened properties of 3D printed concrete: The influence of process parameters on interlayer adhesion. *Cement and Concrete Research*, *119*, 132–140. https://doi.org/10.1016/j.cemconres.2019.02.017

ZAHEDI, A. & PETZOLD, F. (2018). Utilization of simulation tools in early design phases through adaptive detailing strategies. *CAADRIA 2018 - 23rd International*

Conference on Computer-Aided Architectural Design Research in Asia: Learning, Prototyping and Adapting, *2*, 11–20.

ZHANG, W., MEHTA, A., DESAI, P. S. & FRED HIGGS, C. (2017). Machine learning enabled powder spreading process map for metal additive manufacturing (AM). *Solid Freeform Fabrication 2017: Proceedings of the 28th Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference, SFF 2017*, 1235–1249.

Kapitel 9

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die von mir eingereichte Abschlussarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort, Datum, Unterschrift