

Institut für Rechnergestützte Ingenieursysteme

Universität Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Bachelorarbeit Nr. 295

# **Verbesserung der Druckzeitabschätzung im 3D-Druck**

Bernd Jung

**Studiengang:** Informatik B.Sc.  
**Prüfer/in:** Prof. Dr. Dieter Roller  
**Betreuer/in:** Dipl.-Inf. Felix Baumann

**Beginn am:** 14. Dezember 2015  
**Beendet am:** 28. Juni 2016  
**CR-Nummer:** C.3, G.1.6



## Kurzfassung

Druckzeitabschätzungen im FDM 3D-Druck liegen oft weit über oder unter der tatsächlich benötigten Zeit. Eine häufig implementierte Methode, die Summe aus der berechneten Dauer einzelner Bewegungen zu bilden, bildet die physikalischen Gegebenheiten und die Auswirkungen softwareseitiger Optimierungen nur unzureichend ab. In dieser Arbeit werden mögliche Ursachen dieser Differenzen beschrieben. Außerdem wird ein im Verlauf dieser Arbeit entwickelter Ansatz zur Druckzeitabschätzung vorgestellt. Dabei werden Informationen aus bereits erfolgten Drucken verwendet, um eine Relation von inhärent in G-Code enthaltenen Eigenschaften zur dann bekannten Druckzeit zu finden, und daraus eine Berechnungsvorschrift zu abzuleiten, die als Repräsentation einer Regressions-Hyperebene dargestellt werden kann. Anschließend wird der Ansatz mit Daten von 21 verschiedenen, physikalisch gedruckten Testmodellen überprüft.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>13</b>
<b>2</b>	<b>Definitionen</b>	<b>17</b>
<b>3</b>	<b>Der Aufbau</b>	<b>19</b>
3.1	Testgerät . . . . .	19
3.2	Sensordaten . . . . .	19
3.3	Hilfswerkzeuge . . . . .	20
<b>4</b>	<b>Testmodelle</b>	<b>21</b>
<b>5</b>	<b>Einflüsse auf die Druckzeit</b>	<b>29</b>
5.1	Theorie . . . . .	29
5.2	Algorithmen . . . . .	30
5.3	Antriebe . . . . .	30
5.4	Taktgeber . . . . .	30
5.5	Filament . . . . .	31
5.6	Optimierungen . . . . .	31
5.7	Temperatur . . . . .	32
5.8	Folgerungen . . . . .	32
<b>6</b>	<b>Auswertung der Sensordaten</b>	<b>33</b>
6.1	Extraktion der tatsächlichen Druckzeit . . . . .	33
6.2	Erster Ansatz für eine Abschätzung . . . . .	34
<b>7</b>	<b>Existierende Ansätze</b>	<b>37</b>
<b>8</b>	<b>Neuer Ansatz: Regressions-Hyperebene</b>	<b>41</b>
8.1	Idee . . . . .	41
8.2	Die Rolle der Features . . . . .	42
8.3	Begründung für die Auswahl linearer Regressions-Hyperebenen . . . . .	44
8.4	Evaluation des Ansatzes . . . . .	45
<b>9</b>	<b>Zusammenfassung und Ausblick</b>	<b>55</b>
	<b>Literaturverzeichnis</b>	<b>57</b>



# Abbildungsverzeichnis

4.1	Testmodell maze-inside. Bildschirmfoto von gcode.ws . . . . .	23
4.2	Testmodell J10. Bildschirmfoto von gcode.ws . . . . .	23
4.3	Testmodell E10. Bildschirmfoto von gcode.ws . . . . .	23
4.4	Testmodell c-45. Bildschirmfoto von gcode.ws . . . . .	24
4.5	Testmodell I10. Bildschirmfoto von gcode.ws . . . . .	24
4.6	Testmodell C10. Das Modell ist flach, es hat nur 4 Lagen. Bildschirmfoto von gcode.ws . . . . .	24
4.7	Testmodell B10. Bildschirmfoto von gcode.ws . . . . .	25
4.8	Testmodell maze-outside. Die Struktur im Inneren ist gerade noch erkennbar. Bildschirmfoto von gcode.ws . . . . .	25
4.9	Testmodell mic-bot_v4. Bildschirmfoto von gcode.ws . . . . .	25
4.10	Testmodell galaxys5-battery. Bildschirmfoto von gcode.ws . . . . .	26
4.11	Testmodell E10 im Schnitt in der X,Y-Ebene. Der größte Teil der Verfahrswege besteht aus Stützstrukturen. Bildschirmfoto von gcode.ws . . . . .	27
4.12	Testmodell maze-inside im Schnitt in der X,Y-Ebene. Die gesamte Schicht besteht aus approximierten Kreisen. Verschiedene Farben zeigen unterschiedliche Verfahrgeschwindigkeiten an. Bildschirmfoto von gcode.ws . . . . .	27
4.13	Testmodell maze-outside im Schnitt in der X,Y-Ebene. Zu maze-inside ähnliche Form aber mit Stützstrukturen und kürzeren Kreisbögen. Bildschirmfoto von gcode.ws . . . . .	27
6.1	Bewegungsverlauf der X-Achse. Der Graph (rot) zeigt die erste Bewegung nach dem Setzen der Position. Eine Beschleunigung ist nicht erkennbar. Im Logfenster sind die abgeschrittenen Datenpunkte von rechts nach links, bis zur aktuellen Position des Fadenkreuzes (x:blau, y:grün) eingetragen. Jeder Datenpunkt hat einen Zeitstempel und einen Wert. Für die Einträge „difference per second“, wurde die Wertdifferenz durch die Zeitdifferenz in Sekunden geteilt. Ein Wert von 90 entspricht etwa 1 Millimeter. Bildschirmfoto von Graf	34
8.1	Relative Abweichungen zur tatsächlichen Druckzeit je Modell. Die Reihenfolge der Modelle entspricht Tabelle 8.1 . . . . .	51
8.2	Größte absolute relative Abweichung . . . . .	51
8.3	Kleinste absolute relative Abweichung . . . . .	51
8.4	Abweichungen zur tatsächlichen Druckzeit je Modell . . . . .	52
8.5	Größte absolute Abweichung . . . . .	52

8.6	Kleinste absolute Abweichung . . . . .	52
8.7	Größte Differenz zwischen einzelnen Abweichungen . . . . .	53
8.8	Durchschnittliche absolute Abweichung . . . . .	53
8.9	Abweichungen zur tatsächlichen Druckzeit für Modelle 15 bis 21 . . . . .	53



# Tabellenverzeichnis

8.1	Feature-Werte und tatsächliche Druckzeit der Modelle im Testfeld. Zeiträume sind in Sekunden, Winkel in Radianen und Geschwindigkeiten in Millimeter pro Minute angegeben. . . . .	48
8.2	Aus den Feature-Werten errechnete Koeffizienten der Hyperebene. Die Höhe der einzelnen Werte der Koeffizienten $x_2, x_3, \dots, x_m$ , zeigt den Einfluss der jeweiligen dazugehörigen Features . . . . .	49
8.3	Root Mean Squared Error im Testfeld je Verfahren zur Druckzeitabschätzung. Berechnet aus den Differenzen zur tatsächlichen Druckzeit. . . . .	50



# Verzeichnis der Listings

7.1	Auszug aus Repetier-Host/GCodeAnalyzer.cs. Teil der Vorausberechnung der Druckzeit aus G-Code Anweisungen. . . . .	37
7.2	Auszug aus den Kommentaren von GCoder/gcoder.py . . . . .	38



# 1 Einleitung

Die durch den 3D-Druck eröffneten Möglichkeiten, wurden bisher nur Ansatzweise entdeckt. Es können Formen erzeugt werden, die mit der spanenden Bearbeitung oder dem Spritzguss nicht erreicht werden können. Bei der spanenden Bearbeitung muss das Werkzeug an die zu bearbeitende Oberfläche herangeführt werden. Ist dies nicht möglich, weil eine zuvor bearbeitete Struktur den Zugang verhindert, kann noch versucht werden, das Gesamtteil aus mehreren Einzelteilen zusammen zu setzen. Die beim Spritzguss verwendete Form, muss nach erfolgtem Guss, von den Teilen wieder getrennt werden. Enthielte die Gussform filigrane, sich nach innen vergrößernde Elemente, könnte dies verhindert werden. Die Form der meisten bis heute gefertigten Teile, wurde den oben genannten Einschränkungen angepasst. Dadurch wurden notwendigerweise Kompromisse zwischen Funktion und Form gemacht. Durch 3D-Druck gefertigte Teile, können beinahe beliebige Formen eingefasst in anderen Formen enthalten. 3D-Druck wird momentan aber nur selten für die Produktion in hohen Stückzahlen, sondern vorrangig für Prototypen oder für sehr spezielle Teile verwendet, die mit den bisherigen Herstellungsverfahren nicht oder nur eingeschränkt erzeugbar sind [GH16, S. 96]. Bei Prototypen steht die schnelle Verfügbarkeit eines realen Modells im Vordergrund: um ein gedrucktes Modell zu erhalten, muss dieses lediglich als 3D-Modell entworfen werden und kann, nach der Umwandlung in ein für den Drucker erkennbares Dateiformat, direkt gedruckt werden [GNS16, S. 47] [Um16, S. 192 ff.]. In der medizinischen Forschung werden 3D-Drucker eingesetzt um aus schichtweise aufgetragenen Zellen, Organe zu drucken.<sup>1</sup>

Momentan gibt es nur eine geringe Anzahl von für den 3D-Druck verwendbaren Materialien. Diese Materialien wiederum können nur mit wenigen der bislang bekannten Druckverfahren verarbeitet werden, da die Funktionsweise jedes dieser Verfahren, besondere chemische oder physikalische Eigenschaften der verwendbaren Materialien ausnutzt. Die Eigenschaften eines gedrucktes Teils ändern sich mit dem verwendeten Material. Darum entscheiden die Anforderungen, die an das gedruckte Teil gestellt werden, über die zu verwendenden Materialien, die wiederum nur bestimmte Druckverfahren zulassen [GRS15, S. 305]. Sehr häufig werden Thermoplaste im Fused Deposition Modeling (FDM) Verfahren eingesetzt, etwa in Form von Acrylnitril-Butadien-Styrol (ABS) oder Polylactiden (PLA).

Der Preis von FDM 3D-Druckern ist bereits so niedrig geworden, dass sie häufig von Endverbrauchern gekauft werden. Dies hat den kreativen Umgang mit den Geräten gefördert und dazu geführt, dass sehr viele, sofort druckbare 3D-Modelle von den Benutzern selbst erstellt

<sup>1</sup><https://www.hw.ac.uk/news/archive/2013/printed-human-organs-testing-transplantation-11075.htm>

und im Internet kostenfrei verfügbar gemacht wurden. Diese Vielfalt an nützlichen, dekorativen oder verspielten Modellen stellt wiederum einen Anreiz dar, selbst einen 3D-Drucker zu erwerben [Fas12, S. 29 ff.].

Aufgrund der vielfältigen Möglichkeiten und der steigenden Nachfrage kann also davon ausgegangen werden, dass die Geräte zukünftig weiter verbessert werden, um die Qualität der gedruckten Teile zu erhöhen und die für den Druck eines Teils benötigte Zeit zu verringern. Damit ist auch zu erwarten, dass der Anteil des 3D-Drucks gegenüber anderen Herstellungsverfahren steigt und so für den privaten als auch den industriellen Einsatz wichtiger wird.

Obwohl FDM 3D-Drucker in großen Mengen hergestellt und verkauft werden, ist das Ergebnis der Vorausberechnung der Druckzeit eines Teils nicht immer genau. Die momentan für den Druck eines Teils benötigte Zeit reicht von einigen Sekunden bis vielen Stunden. Sie hängt unter anderem vom verwendeten 3D-Drucker, dem Druckverfahren und der Größe des zu druckenden Teils ab. Diese 3 Faktoren lassen sich weiter aufspalten in den physikalischen Aufbau des 3D-Druckers, der exakten Funktionsweise der verwendeten Software die das Verhalten der Maschine steuert, Bedingungen die künstlich geschaffen werden müssen um das zu verwendende Druckverfahren anwenden zu können und die exakte Geometrie sowie Ausrichtung des zu druckenden Teils.

Für eine Vorausberechnung der Druckzeit müsste der Einfluss aller dieser Faktoren mathematisch exakt nachgebildet werden, was nur mit einem extremen Aufwand möglich wäre. Darum wurden in dieser Arbeit Faktoren gesucht, die eine Verbesserung der Vorausberechnung der Druckzeit ermöglichen, ohne die physikalischen Gegebenheiten exakt nachzumodellieren. Um die daraus resultierenden Berechnungen zu überprüfen, wurden die tatsächlichen Druckzeiten mit einem FDM 3D-Drucker ermittelt.

## Gliederung

Die Gliederung dieser Arbeit ist wie folgt:

**Kapitel 2 – Definitionen:** definiert die in dieser Arbeit verwendeten Begriffe.

**Kapitel 3 – Der Aufbau:** beschreibt die verwendete Hard- und Software.

**Kapitel 4 – Testmodelle:** stellt die verwendeten Testmodelle vor.

**Kapitel 5 – Einflüsse auf die Druckzeit:** nennt und beschreibt Faktoren, die die Druckzeit beeinflussen können.

**Kapitel 6 – Auswertung der Sensordaten:** beschreibt die Verwendung der für die Arbeit zur Verfügung stehenden Sensordaten.

**Kapitel 7 – Existierende Ansätze:** stellt bereits angewandte Ansätze zur Druckzeitabschätzung vor.

---

**Kapitel 8 – Neuer Ansatz: Regressions-Hyperebene:** beschreibt einen im Verlauf dieser Arbeit entwickelten, neuen Ansatz zur Druckzeitabschätzung.

**Kapitel 9 – Zusammenfassung und Ausblick** bietet eine Zusammenfassung und Anregungen zur weiteren Nachforschung.





## 2 Definitionen

### 3D-Modell:

Repräsentation der Oberfläche eines physikalischen Objekts als dreidimensionale Vektorgraphik

### G-Code:

zeilenbasierte Anweisungsfolge für koordinatenbasierte Maschinen; folgt strukturell „DIN 66025“<sup>1</sup> beziehungsweise „The NIST RS274NGC Interpreter - Version 3“<sup>2</sup>

### Slicer:

Programm zum Konvertieren von 3D-Modellen in G-Code; Resultat sind Verfahrenswege in Lagen variabler Höhe

### Druckzeit:

- bezogen auf den G-Code: Menge von Anweisungen, die für den Druck eines Teils unbedingt notwendig ist, beinhaltet alle Anweisungen von „fahre zur Stelle des Beginns der ersten Lage“ bis zur letzten Bewegung weg vom gedruckten Modell
- bezogen auf einen Zeitraum: für den Druck eines 3D-Modells auf einem 3D-Drucker benötigte Zeit, entsprechend den für einen Anwendungsfall jeweils festgelegten ersten und letzten gültigen Anweisungen eines G-Codes

### Theoretische Druckzeit:

aus den zur Druckzeit gehörenden Anweisungen des G-Codes berechnet: Summe der Quotienten aus der Länge des Verfahrenswegs und dazugehöriger programmierter Verfahrensgeschwindigkeit

### Tatsächliche Druckzeit:

für den Druck eines 3D-Modells auf einem 3D-Drucker benötigte Zeit, entsprechend den zur Druckzeit gehörenden Anweisungen des G-Codes

### Druckzeitabschätzung:

eine Vorausberechnung der tatsächlichen Druckzeit unter Verwendung heuristischer Methoden, für ein 3D-Modell, das durch den Slicer in G-Code umgewandelt wurde

<sup>1</sup><http://www.din.de/de/mitwirken/normenausschuesse/nwm/normen/wdc-beuth:din21:1012276>

<sup>2</sup>[http://www.nist.gov/manuscript-publication-search.cfm?pub\\_id=823374](http://www.nist.gov/manuscript-publication-search.cfm?pub_id=823374)

## 2 Definitionen

---

Fused Deposition Modeling (FDM):

Druckverfahren, bei dem das aufzubringende Material von einer temperierten Düse geschmolzen extrudiert wird [GH16, S. 141]; im Deutschen auch „Schmelzschichtung“<sup>3</sup>

Filament:

Drahtförmiges Thermoplast, das der Düse eines FDM 3D-Druckers zugeführt wird um extrudiert zu werden [GH16, S. 141]

Retract:

Zurückziehen des Filaments um den Druck auf das bereits geschmolzene Thermoplast in der Extruderdüse zu reduzieren

Layer:

im FDM Verfahren: eine aufzubringende oder aufgebraute Schicht eines Modells aus Thermoplast; im Deutschen auch „Schicht“ oder „Lage“

<sup>3</sup><https://www.hs-niederrhein.de/design/werkstaetten/3d-druck-fdm>

# 3 Der Aufbau

## 3.1 Testgerät

Als Testgerät stand ein MakerBot Replicator 2X<sup>1</sup> zur Verfügung. Dieser 3D-Drucker arbeitet im Fused Deposition Modeling-Verfahren. Da sämtliche Versuche mit diesem Gerät ausgeführt wurden, wird zur Vereinfachung im Nachfolgenden auf diesen Drucker als „Testgerät“ Bezug genommen.

Der Drucker hat 3 bewegliche Achsen: X, Y und Z. Um eine Bewegung in +X- beziehungsweise +Y-Richtung auszuführen, wird der Druckkopf nach rechts beziehungsweise nach hinten verfahren. Für eine Bewegung in +Z-Richtung wird das Druckbett nach unten verfahren.

Ebenso umfasst der Begriff einer Bewegung des Druckers, in dieser Arbeit auch die Extrusion von Filament aus einer der beiden in der Maschine vorhandenen Extruder. Dies erleichtert die Angabe der jeweiligen Zeiten, da im G-Code häufig Anweisungen vorkommen die nur das Auspressen oder Zurückziehen des Filaments enthalten, jedoch mit einer Geschwindigkeitsvorgabe versehen sind, sodass sich daraus eine theoretische Dauer errechnen lässt.

## 3.2 Sensordaten

Um die tatsächlichen Bewegungen des Testgeräts mit den programmierten Bewegungen zu vergleichen, wurden in einem vorherigen Projekt Sensoren angebracht, die während des Drucks periodisch ausgelesen werden. Die Sensoren selbst wie auch die Position am Drucker, an der sie angebracht sind variierten über den Verlauf dieser Arbeit. Unter den aufgezeichneten Daten sind Werte eines Beschleunigungssensors, eines Gyroskops, eines Kompasses und eines Magnetfeldsensors. Das Auslesen der Sensorwerte erfolgt über die mit Prozessoren bestückten Entwicklungsplatinen Arduino<sup>2</sup> und Teensy<sup>3</sup>, die wiederum über einen USB-Hub mit einem Notebook verbunden sind und darüber abgefragt werden. Auf dem Notebook wird als Betriebssystem Linux ausgeführt. Der Drucker selbst ist ebenfalls über eine USB-Schnittstelle mit dem Notebook verbunden. Über diese Verbindung werden während des Druckens, ebenfalls

<sup>1</sup><http://store.makerbot.com/replicator2x>

<sup>2</sup><https://www.arduino.cc>

<sup>3</sup><http://www.pjrc.com/teensy>

periodisch, Statuswerte vom Drucker selbst abgefragt. Die von Arduino, Teensy und Drucker empfangenen Daten werden jeweils in eine separate Datei geschrieben. Diese Dateien enthalten je Zeile die Zeilennummer, das Datum, die Uhrzeit der Abfrage mit Millisekunden und darauffolgend, die ungefähr zeitgleich abgefragten Werte der angeschlossenen Sensoren.

### 3.3 Hilfswerkzeuge

Als Hilfe bei der Auswertung der Sensordaten, wurde eine Software (Graf) in der Programmiersprache C# entwickelt. Sie ermöglicht eine flexible Darstellung der Daten als Graph. Da die Dateien teilweise mehr als 250.000 Zeilen beziehungsweise Datenpunkte enthalten, war es nötig die Anwendung so zu konstruieren, dass der Graph verschoben sowie vergrößert und verkleinert dargestellt werden kann. Es sind außerdem auch Werte aus analogen Sensoren vorhanden, die so verrauscht sind, dass in Graf ein Tiefpassfilter integriert werden musste um sie analysieren zu können.

Der vom Slicer aus dem 3D-Modell erzeugte G-Code enthält sämtliche Anweisungen, die von der Maschine sequenziell ausgeführt werden sollen. Um die Anweisungsfolge besser analysieren zu können, wurde ein Programm (Gex) in der Programmiersprache C# entwickelt. Der dafür notwendige G-Code Parser, wurde mit Hilfe des ANTLR-Frameworks<sup>4</sup> erstellt. Damit kann die theoretische Ausführungszeit einer Maschinenbewegung aufgrund von Kontextinformationen wie zum Beispiel dem Anschlusswinkel an die nächste Bewegung, modifiziert werden, um die tatsächliche Ausführungszeit anzunähern. Außerdem kann damit die tatsächliche Druckzeit aus den aufgezeichneten Positionswerten des Druckers, eines bereits erfolgten Drucks ermittelt werden.

<sup>4</sup><http://www.antlr.org>

Lizenz: BSD

## 4 Testmodelle

Für die Berechnung der Druckzeitabschätzungen und die Ermittlung der tatsächlichen Druckzeit wurden verschiedene Modelle verwendet. Da viele der zur Verfügung stehenden Sensor- und Positionsdaten bei Beginn dieser Arbeit schon ausgelesen und gespeichert waren und in deren Verlauf aus den bis dahin verwendeten Modellen weitere Daten erzeugt wurden, war die Auswahl der Modelle zufällig. Dennoch sind es Modelle verschiedener Größe und Form, die durch jeweilige spezielle Merkmale, Besonderheiten beim Auswerten der Fahrwege durch die Firmware erkennbar machen können, da wie in Abschnitt 5.6 auf Seite 31 beschrieben wird, die Firmware die Ausführung des Druckes den speziellen Merkmalen entsprechend unterschiedlich planen könnte. Diese Merkmale sollten möglichst in jeweils unterschiedlichen Teilen, getrennt von anderen Merkmalen erscheinen, sodass eine markante Abweichung der erwarteten zur tatsächlichen Druckzeit leichter auf die jeweilige Besonderheit des gedruckten Teils zurückgeführt werden kann. Die erwünschten Merkmale sind:

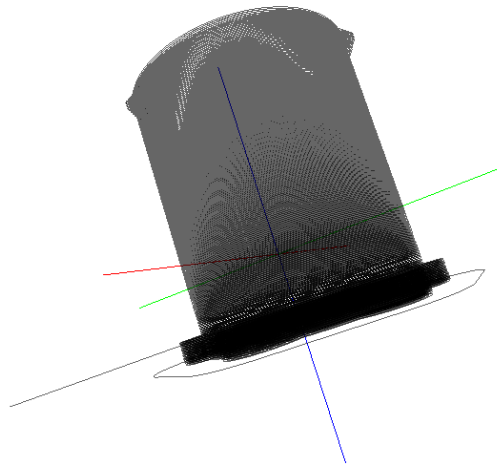
- je Modell entweder kurze oder lange, schnelle oder langsame Verfahwege in gleicher Ausrichtung zum Koordinatensystem der Maschine (einfache aneinander- beziehungsweise übereinandergelegte Linien)
  - welche Auswirkungen sind bezüglich der Beschleunigung in verschiedenen Achsen feststellbar?
- je Modell gleiche Anschlusswinkel (regelmäßiges Hexagon, Septagon, Oktagon, ...)
  - welche Auswirkungen haben die Anschlusswinkel auf die positive und negative Beschleunigung zwischen den Bewegungen?
- diese Modelle in unterschiedlicher Höhe
  - verändern sich diese Auswirkungen linear zu der Anzahl der Lagen?
- kurze Verfahwege folgend auf lange Fahrten mit hoher Geschwindigkeit
  - wie wirkt sich eine Veränderung der Geschwindigkeit aus?
- kurze Verfahwege die eine Kreisform approximieren
  - inwieweit wird die Gesamtbewegung optimiert?
- Modelle, die diese und weitere Merkmale in Kombination enthalten

→ gibt es einen linearen Zusammenhang zwischen dem Vorkommen der Merkmale und der tatsächlichen Druckzeit?

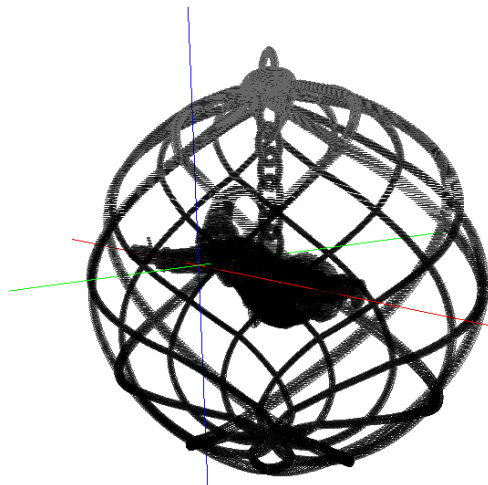
Leider war kein Modell darunter das überwiegend Verfahrswege enthält, die eine Kreisform approximieren, außerdem gab es kein Modell das ausschließlich in der Höhe variiert. Darum wurde noch das Modell maze-inside<sup>1</sup> aus Abbildung 4.1 in die Auswahl hinzugenommen. Dessen Besonderheit ist der hohe, zylindrische Rand, der wie in Abbildung 4.12 ersichtlich, fast ausschließlich aus approximierten Kreisbahnen besteht und einen großen Teil des Gesamtvolumens dieses Teils einnimmt. maze-inside ist eines der Teile, bei dem die Vorausberechnung der Druckzeit mit dem in Abschnitt 6.2 auf Seite 34 beschriebenen additiven Faktor eine große Abweichung von etwa 35 Minuten erzeugt.

Das Modell J10 aus Abbildung 4.2 enthält viele kurze Verfahrswege bei denen Material aufgebracht wird aber auch viele Fahrten in hoher Geschwindigkeit.

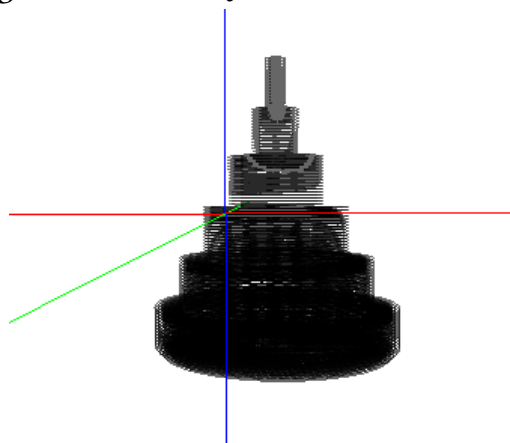
<sup>1</sup><http://www.thingiverse.com/thing:207662>



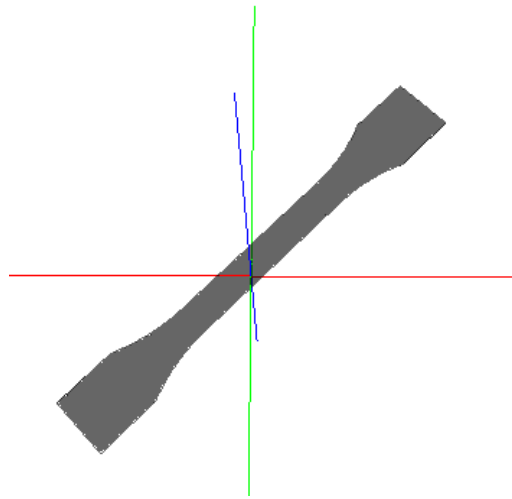
**Abbildung 4.1:** Testmodell maze-inside. Bildschirmfoto von gcode.ws



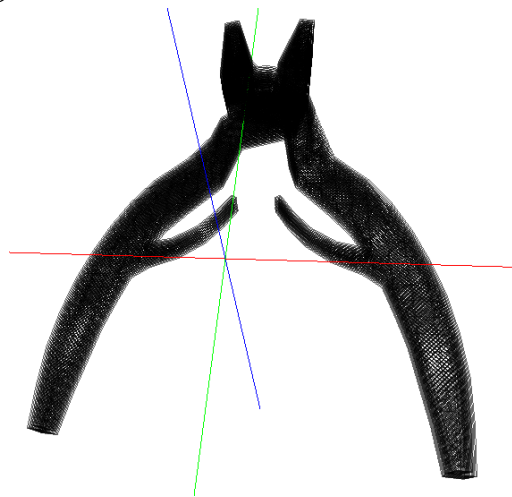
**Abbildung 4.2:** Testmodell J10. Bildschirmfoto von gcode.ws



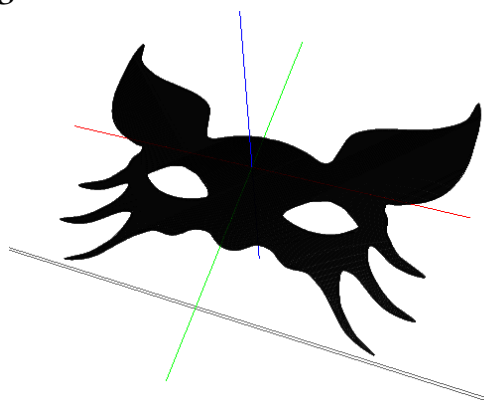
**Abbildung 4.3:** Testmodell E10. Bildschirmfoto von gcode.ws



**Abbildung 4.4:** Testmodell c-45. Bildschirmfoto von gcode.ws

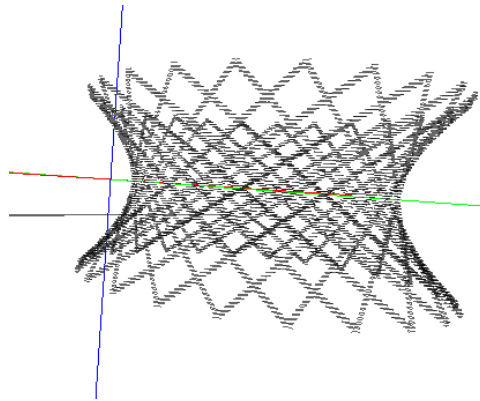


**Abbildung 4.5:** Testmodell I10. Bildschirmfoto von gcode.ws

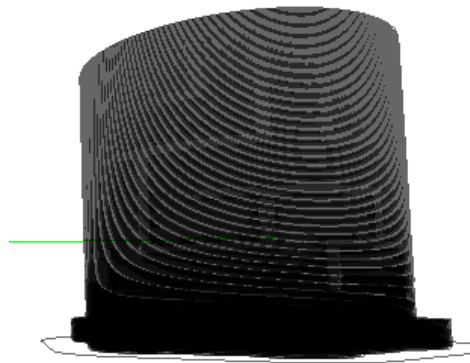


**Abbildung 4.6:** Testmodell C10. Das Modell ist flach, es hat nur 4 Lagen. Bildschirmfoto von gcode.ws

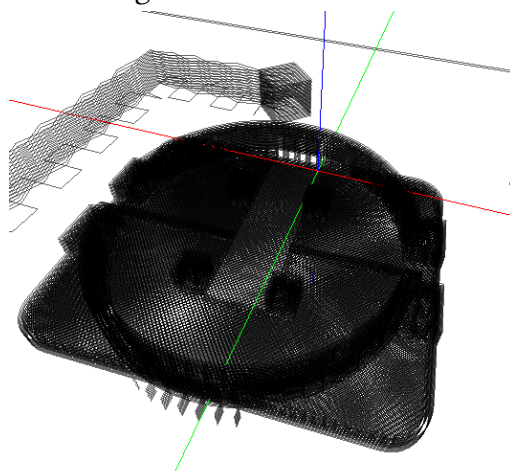




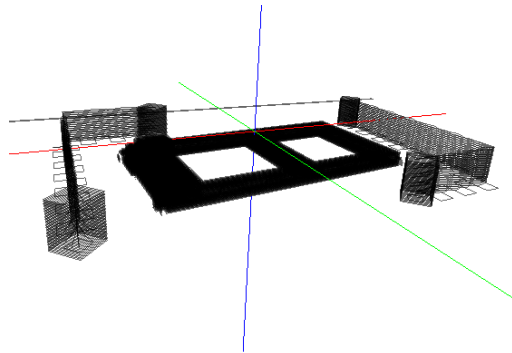
**Abbildung 4.7:** Testmodell B10. Bildschirmfoto von gcode.ws



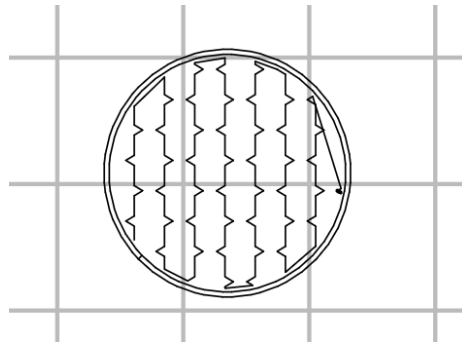
**Abbildung 4.8:** Testmodell maze-outside. Die Struktur im Inneren ist gerade noch erkennbar. Bildschirmfoto von gcode.ws



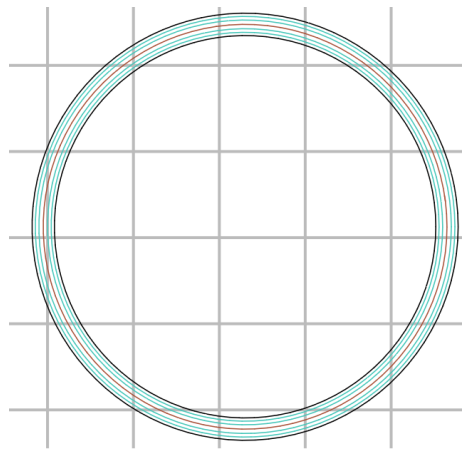
**Abbildung 4.9:** Testmodell mic-bot\_v4. Bildschirmfoto von gcode.ws



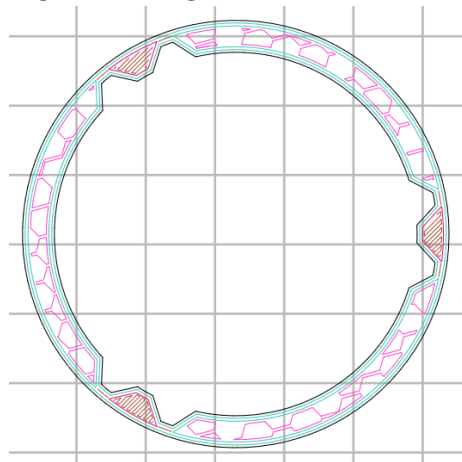
**Abbildung 4.10:** Testmodell galaxy5-battery. Bildschirmfoto von gcode.ws



**Abbildung 4.11:** Testmodell E10 im Schnitt in der X,Y-Ebene. Der größte Teil der Verfahrswege besteht aus Stützstrukturen. Bildschirmfoto von gcode.ws



**Abbildung 4.12:** Testmodell maze-inside im Schnitt in der X,Y-Ebene. Die gesamte Schicht besteht aus approximierten Kreisen. Verschiedene Farben zeigen unterschiedliche Verfahrgeschwindigkeiten an. Bildschirmfoto von gcode.ws



**Abbildung 4.13:** Testmodell maze-outside im Schnitt in der X,Y-Ebene. Zu maze-inside ähnliche Form aber mit Stützstrukturen und kürzeren Kreisbögen. Bildschirmfoto von gcode.ws



# 5 Einflüsse auf die Druckzeit

## 5.1 Theorie

Die folgende Auflistung an möglichen, die Druckzeit beeinflussenden Faktoren soll als Informationsfundament gelten, um die Umstände, denen eine Vorausberechnung der Druckzeit folgen müsste zu verdeutlichen.

Könnte ein Drucker, die für die Druckzeit entscheidenden Zeilen aus dem G-Code exakt in der programmierten Geschwindigkeit ausführen, wäre die theoretische Druckzeit die untere Grenze für die tatsächliche Druckzeit. Die Differenz zwischen diesen beiden Werten bestünde dann aus der Summe der Zeiten, die je Bewegung für die erfolgte positive und negative Beschleunigung zusätzlich entstanden sind.

Da die relative Bewegung des Druckkopfs zum Druckbett, das Resultat aus den Bewegungen der einzelnen Achsen ist, ist die resultierende Geschwindigkeit stets größer oder gleich der Geschwindigkeiten der einzelnen Achsen

$$(5.1) \quad v_{ges} = \sqrt{v_0^2 + v_1^2 + \dots + v_n^2}$$

Der größte Unterschied der resultierenden Geschwindigkeit zur Achse mit der höchsten Geschwindigkeit besteht für  $v_0 = v_1 = \dots = v_n$

$$(5.2) \quad v_{ges} = \sqrt{n+1} \cdot v_0 = \sqrt{n+1} \cdot v_1 = \dots = \sqrt{n+1} \cdot v_n$$

Um eine gleichmäßige Bewegung zu erzeugen, wird für eine Bewegung in mehreren Achsen, die Geschwindigkeit der einzelnen Achsen entsprechend synchronisiert. Ebenso ist auch die Gesamtbeschleunigung stets größer oder gleich der Beschleunigung der einzelnen Achsen

$$(5.3) \quad a_{ges} = \sqrt{a_0^2 + a_1^2 + \dots + a_n^2}$$

Zusätzlich muss die Menge des extrudierten Filaments der resultierenden Beschleunigung sowie der resultierenden Geschwindigkeit angepasst werden um eine gleichmäßige Schichtdicke zu gewährleisten.

### 5.2 Algorithmen

Wie in jedem Computersystem können bei der Berechnung der oben genannten resultierenden Geschwindigkeiten beziehungsweise Beschleunigungen Rundungsfehler, Auslöschung und Fehlerverstärkung auftreten [DR08, S. 35 ff.], die dann auch die Druckzeit beeinflussen. Ob diese inhärent vorhandenen Fehler allerdings die Druckzeit in einem bemerkbaren Bereich verändern, hängt von der Implementation der Firmware des jeweiligen Druckers ab. Die tatsächlichen Auswirkungen müssten über eine formale Analyse der jeweiligen Software ermittelt werden. Sie sollten im Regelfall aber minimal sein und von anderen Faktoren überschattet werden.

### 5.3 Antriebe

In dem verwendeten Testgerät sind Schrittmotoren eingebaut um X-, Y-, und Z -Achsen zu bewegen. Um eine bestimmte Drehzahl zu erhalten, wird bei Schrittmotoren die benötigte Anzahl an Schritten je Zeiteinheit berechnet und diese über eine Elektronik geregelt [Rum15, S. 1 f.]. Dies gilt auch für die Beschleunigung eines Schrittmotors, mit dem Unterschied, dass sich dabei die benötigte Anzahl an Schritten stets ändert. Da Schrittmotoren im Allgemeinen keine Abfrage ihrer aktuellen Position zulassen, sondern davon ausgegangen wird, dass die berechnete Position auch die tatsächliche Position ist, sollte sich hierüber keine Abweichung in der Ausführungszeit einer Bewegung ergeben. Ist der Schrittmotor allerdings mit einer zusätzlichen Positionserkennung, etwa einem Drehgeber ausgestattet, kann die tatsächlich benötigte Zeit für die Bewegung auf eine bestimmte Position von der theoretischen Zeit abweichen, falls der Motor nicht allen von der Elektronik vorgegebenen Schritten folgt.

### 5.4 Taktgeber

Das Testgerät zeigte nach einer manuell abgestoppten Druckzeit von 1 Stunde und 36 Minuten an, es hätte bereits 1 Stunde und 40 Minuten gedruckt, obwohl es nach dem Aufwärmen der Basisplatte und des Extruders anzeigte, es wären schon 7 Minuten vergangen. Dies entspricht einer Abweichung von etwa -2 Minuten pro Stunde. Falls die Schrittmotoren von dem gleichen Taktgeber gesteuert werden, entstünde ebenfalls eine Abweichung. Aufgrund des höheren Taktes ergäbe sich eine erhöhte Beschleunigung und eine erhöhte Endgeschwindigkeit der Schrittmotoren beziehungsweise der Achsen.

## 5.5 Filament

Bei FDM 3D-Druckern kann das Aufbringen von Material unterbrochen werden, um es an einer anderen Stelle fortzusetzen. Damit das bereits in der Düse erhitzte Thermoplast nicht heraus tropft, wird der Druck auf das Filament reduziert, indem es zurückgezogen wird. Dieser Vorgang wird Retract genannt. Dies geschieht sehr häufig an Positionen, an denen die aktuelle Lage fertig gedruckt wurde und der Druckkopf auf die Anfangsposition der nächsten Lage verfährt. Wenn der Druckkopf die Zielposition erreicht hat, muss das zurückgezogene Filament wieder nachgeführt werden. Die G-Code Dateien werden vom Slicer in der Regel so ausgegeben, dass der Druckkopf während dieses Zurückziehens und des darauffolgenden Nachführens an der Start- beziehungsweise Ziel-Position stehen bleibt um ein Schmieren zu verhindern. Wie weit das Filament dazu zurückgezogen werden muss, hängt vom jeweiligen Druckermodell ab. Für Berechnungen werden solche Vorgänge, in dieser Arbeit, als einfache Bewegungen behandelt, wie in Abschnitt 3.1 auf Seite 19 beschrieben.

Hat ein 3D-Drucker mehrere Druckköpfe und werden diese im Verlauf eines Druckes mehrfach gewechselt, kann dies ebenfalls einen signifikanten Einfluss auf die Druckzeit haben. So können auch Vorausberechnungen verfälscht werden, wenn die Extruderdüse, auf die gewechselt wird, entweder weiter vom nächsten Zielpunkt entfernt oder näher an diesem liegt. Im ersten Fall würde die erwartete Zeit überschritten, im zweiten Fall unterschritten.

## 5.6 Optimierungen

Vor dem Druckvorgang wird in der Regel die zu druckende Datei von der Firmware analysiert und ein Ausführungsplan erstellt<sup>1 2</sup>. Dabei werden je nach Firmware und zu druckendem Modell Bewegungsstrategien errechnet und so der Druck optimiert. Um das Modell dann zu drucken wird dieser Ausführungsplan von der Firmware abgearbeitet.

Falls aufeinanderfolgende Bewegungen die gleiche Geschwindigkeit haben und ihre Richtung nur geringfügig ändern, ist beim Wechsel auf die nächste solche Bewegung auch nur eine geringe Änderung der aktuellen Geschwindigkeit der Achsen erforderlich. Wenn die Länge der Bewegungen kurz genug ist, können sie sogar zu einer einzigen flüssigen Gesamtbewegung verschmolzen werden. Entscheidend ist auch die Ausrichtung der Bewegungen zum Koordinatensystem des Druckers. So kann es für zwei aufeinanderfolgende Bewegungen in zwei Achsen, mit gleicher Geschwindigkeit und einem Anschlusswinkel von 90°, je nach Ausrichtung sein, dass eine Achse von der Sollgeschwindigkeit  $s$  auf Geschwindigkeit 0 abgebremst werden muss, bevor die andere Achse von 0 auf  $s$  beschleunigt werden kann oder dazu, dass jede der

<sup>1</sup><https://github.com/MarlinFirmware/Marlin/blob/RC/Marlin/planner.cpp>

<sup>2</sup><https://github.com/makerbot/MightyBoardFirmware/blob/master/firmware/src/MightyBoard/Motherboard/StepperAccelPlanner.cc>

Achsen nur von  $\frac{s}{\sqrt{2}}$  auf 0 verzögert und dann lediglich auf diesen Wert wieder beschleunigt werden muss. Ob dies zu einer veränderten Druckzeit führt ist allerdings sehr stark von der verwendeten Firmware und deren Strategie Beschleunigungen zu berechnen abhängig.

### 5.7 Temperatur

In der Regel muss bei einem FDM 3D-Drucker die Extruderdüse auf eine vorgegebene Temperatur erhitzt werden um das aufzutragende Material zu schmelzen. Ebenso muss das Druckbett erhitzt werden um eine gleichmäßige Temperaturverteilung der ersten Lage zu gewährleisten. Beides zusammen dauerte auf dem Testgerät etwa 7 Minuten. Trotzdem wird in dieser Arbeit die Erwärmung der Extruderdüsen sowie des Druckbetts nicht zur Druckzeit gerechnet, da für einen zeitlich direkt darauffolgenden Druck, diese Zeit geringer ist oder nicht mehr aufgewendet werden muss, da Düsen und Bett immer noch annähernd die erforderliche Temperatur haben.

### 5.8 Folgerungen

Unter den vorgenannten, ist das Element mit dem größten Einfluss auf die Druckzeit, die verwendete Firmware. Durch die Anpassung und Optimierung der Eingabedaten kann ohne Kenntnis der genauen Funktionsweise der Firmware nicht sicher vorausgesagt werden, welcher Ausführungsplan aus einer Eingabedatei resultiert. Zudem könnten Abweichung entstehen, durch numerisch nicht stabile Algorithmen, durch einen ungenauen Taktgeber oder Verzögerungen durch Antriebe, Abnutzung oder Verschmutzung. Damit wäre das genaue Verhalten des Druckers sowie die daraus resultierende exakte Druckzeit, auf Eingabe eines beliebigen zu druckenden Modells, praktisch nicht im Voraus ermittelbar, sondern höchstens sehr genau annäherbar.

Dennoch hängt die Ausgabe der Firmware beziehungsweise das Verhalten des Druckers auf Eingabe eines Modells von den spezifischen Eigenschaften dieses Modells ab. Falls man davon ausgeht, dass dieser Umwandlungsprozess von Eingabedatei in Ausführungsplan durch eine determinierte Funktion modelliert werden kann, er auf die gleiche Eingabe also jeweils die gleiche Ausgabe erzeugt, sollte es möglich sein, durch eine passende Auswahl an Eigenschaften der Eingabedatei auf die jeweilige Ausgabe zu schließen.

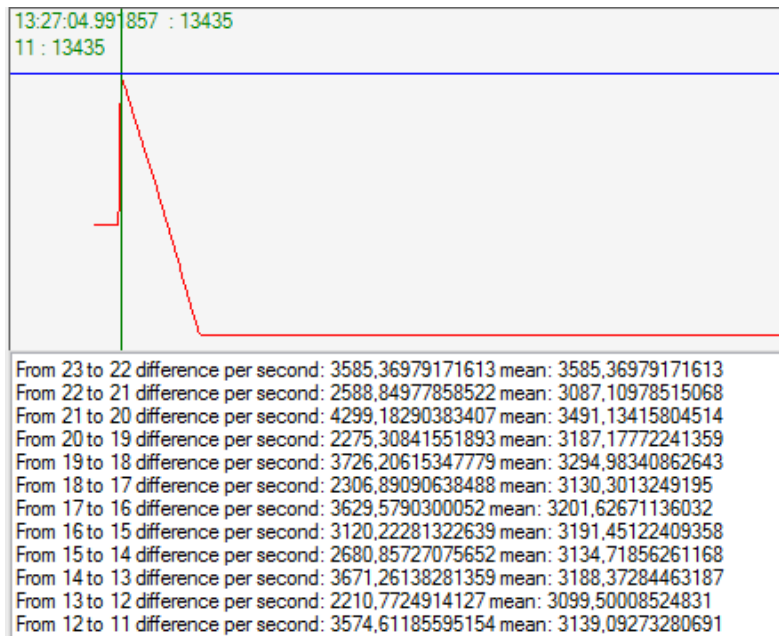


## 6 Auswertung der Sensordaten

Die Aufzeichnung der Sensordaten erfolgt in der Regel in einem Intervall von 100 bis 200 Millisekunden. In einzelnen Fällen kann der zeitliche Abstand zwischen zwei aufeinanderfolgenden Abfragen aber auch einige Sekunden betragen. Für eine Auswertung der Beschleunigung der Achsen ist die Frequenz der Datenpunkte daher generell zu niedrig. Dies ist erkennbar an der Wegdifferenz je Zeiteinheit zwischen aufeinanderfolgenden Datenpunkten, die im Einzelfall am erkennbaren Bewegungsbeginn und dessen Ende, für einen Schritt relativ klein ist, dann aber schon einen Wert annimmt, der nur wenig unter der theoretischen Verfahrgeschwindigkeit für diese Bewegung liegt. Ebenso können erwartete Ausschläge der Beschleunigungssensoren nicht gefunden werden, da diese in einer Aufzeichnungslücke stattfinden. Dies bedeutet, dass die Änderungsrate der Geschwindigkeit zu einem Zeitpunkt, nicht nachvollzogen und so ein evidenzbasierter Wert für die Beschleunigung der Bewegungen des Druckers nicht ermittelt werden kann. Dementsprechend ist die zeitliche Auflösung der einzelnen Werte aus Kompass und Magnetfeldsensor ebenfalls zu grob.

### 6.1 Extraktion der tatsächlichen Druckzeit

Dennoch sind die Sensordaten hilfreich, und tragen zum besseren Verständnis der Druckvorgänge bei. Insbesondere kann aus den direkt vom Drucker abgefragten Status- und Positionswerten, die tatsächliche Druckzeit der jeweils gedruckten Teile ermittelt werden. Dabei entspricht die Genauigkeit in etwa dem zweifachen des oben angesprochenen, variierenden Intervalls. Die Druckzeiten aus diesen Daten zu extrahieren ist Teil der Aufgabe des eigens für diese Arbeit geschriebenen Programms Gex. Dazu wird die Eingabedatei zeilenweise eingelesen und deren Form grob auf Korrektheit, das heißt Vorhandensein und richtige Position der Zeitmarke und die Anzahl der Datenpunkte überprüft. Dementsprechend fehlerhafte Einträge werden ignoriert. Zur Erkennung wird ausgenutzt, dass am Druckbeginn sowie an dessen Ende stets gleiche Positions- und Filamentkoordinaten in einer bestimmten Reihenfolge angefahren werden. Das Programm kann diese Muster beim Durchsuchen der Daten erkennen und so die Druckzeit mittels einer Subtraktion der entsprechenden Zeitmarken extrahieren. Falls Einträge fehlen besteht noch die Möglichkeit, dass der gesuchte Zeitpunkt mit Hilfe der letzten bekannten Position und der aus den vorhergehenden Einträgen berechneten Verfahrgeschwindigkeit extrapoliert werden kann. Sind zu viele Einträge fehlerhaft, kann die Druckzeit nicht mehr ermittelt oder aufgrund von Indizien nur noch geschätzt werden. Ein solcher Fall wird bei der Ausgabe der Druckzeit angemerkt und gegebenenfalls mit einer Einschätzung der möglichen



**Abbildung 6.1:** Bewegungsverlauf der X-Achse. Der Graph (rot) zeigt die erste Bewegung nach dem Setzen der Position. Eine Beschleunigung ist nicht erkennbar. Im Logfenster sind die abgeschrittenen Datenpunkte von rechts nach links, bis zur aktuellen Position des Fadenkreuzes (x:blau, y:grün) eingetragen. Jeder Datenpunkt hat einen Zeitstempel und einen Wert. Für die Einträge „difference per second“, wurde die Wertdifferenz durch die Zeitdifferenz in Sekunden geteilt. Ein Wert von 90 entspricht etwa 1 Millimeter. Bildschirmfoto von Graf

Abweichung versehen. Diese Fehlerbehandlung wurde implementiert um das Risiko der Ausgabe einer falschen Druckzeit zu verringern, da bereits vorhandene Dateien regelmäßig nicht verwendbare Einträge enthielten. Die Sensordaten wie auch die Skripte um sie auszulesen stammen aus einem vorherigen Projekt. Im weiteren Verlauf der Arbeit wurde die Aufzeichnung der Daten verbessert. Für eine Ermittlung der tatsächlichen Druckzeit aus den sonstigen Sensorwerten wäre es nötig einen Graphen zu bilden und diesen im entsprechenden Intervall zu integrieren. Neben einem erhöhten Rechenaufwand würde dies zu nicht quantifizierbaren Ungenauigkeiten führen, da keine kontinuierliche Aufzeichnung zur Verfügung steht.

## 6.2 Erster Ansatz für eine Abschätzung

Bei weiteren Analysen der Sensordaten sah es zunächst so aus als ob sich der Quotient

$$q = \frac{\textit{theoretische} - \textit{tatsächlicheZeit}}{\textit{programmierteGeschwindigkeit}}$$

je Bewegung, in einem hinreichend eng begrenzten Intervall befände und sich so je Bewegung die tatsächliche Zeit

$$tats\ddot{a}chlicheZeit = -q * programmierteGeschwindigkeit + theoretischeZeit$$

berechnen ließe. Dazu wurden die, zum Verfahrenweg einer bestimmten G-Code Anweisung passenden, vom Drucker zurückgemeldeten Positionswerte und deren Zeitmarken verglichen, ähnlich der in Abbildung 6.1 ersichtlichen Log-Einträge. Dies wurde nach der Implementation der obigen Formel in Gex aber nur bedingt bestätigt. Bei vielen Testmodellen aus Kapitel 4 auf Seite 21 ist die Berechnung bis auf wenige Minuten genau, für andere weicht sie aber um mehr als 30 Minuten ab. Dieser Ansatz wurde dann nicht weiter verfolgt, da außerdem ein Fehler beziehungsweise ein zeitlicher Versatz bei der Aufzeichnung der verwendeten Daten nicht ausgeschlossen werden konnte und sonst keine anderen, dafür verwendbaren Daten zur Verfügung standen.



## 7 Existierende Ansätze

Im Verlauf dieser Arbeit wurde nach bereits bestehenden Lösungen, Ansätzen und Programmen für eine Abschätzung der Druckzeit gesucht. Eine allgemeingültige Lösung wurde bislang nicht gefunden. Gefundene Ansätze und Implementierungen sind sich stets sehr ähnlich und gehen im Allgemeinen nicht über eine Summation der Quotienten aus Länge des Fahrwegs und Sollgeschwindigkeit hinaus.

Beispielhaft zeigt dies der Auszug aus `GCodeAnalyzer.cs`<sup>1</sup> in Listing 7.1. Die so erhaltene theoretische Zeit wird in manchen Algorithmen mit Konstanten multipliziert um die für die Beschleunigung und möglicherweise auch andere Faktoren verwendete Zeit zu berücksichtigen. In `gcode.ws`<sup>2</sup> wird der Fahrweg in Z-Richtung für die Berechnung nicht herangezogen und falls eine Zeile des geparsten G-Codes nicht beide Angaben in X- und Y-Richtung enthält, wird nur die theoretische Dauer der Filamentbewegung berücksichtigt.

Wie in Abschnitt 5.1 auf Seite 29 beschrieben, berechnet sich die resultierende Geschwindigkeit als Quadratwurzel aus der Quadratsumme der einzelnen Geschwindigkeiten. So berechnet sich auch die für eine Bewegung in mehreren Achsen benötigte Zeit als Quotient aus resultierender Weglänge und Sollgeschwindigkeit

$$(7.1) \quad t_{Min} = \frac{\sqrt{s_0^2 + s_1^2 + \dots + s_n^2}}{f}$$

---

**Listing 7.1** Auszug aus `Repetier-Host/GCodeAnalyzer.cs`. Teil der Vorausberechnung der Druckzeit aus G-Code Anweisungen.

---

```
[302] float dx = Math.Abs(x - lastX);
[303] float dy = Math.Abs(y - lastY);
[304] float dz = Math.Abs(z - lastZ);
[305] float de = Math.Abs(activeExtruder.e - activeExtruder.lastE);
[306] double time;
[307] if (dx + dy + dz > 0.001)
[308]     time = Math.Sqrt(dx * dx + dy * dy + dz * dz) * 60.0f / f;
[309] else
[310]     time = de * 60.0f / f;
[311] printingTime += time;
```

---

<sup>1</sup><https://github.com/repetier/Repetier-Host/blob/master/src/RepetierHost/model/GCodeAnalyzer.cs>

<sup>2</sup><http://gcode.ws>

---

**Listing 7.2** Auszug aus den Kommentaren von GCoder/gcoder.py

---

```
[549] # FIXME: this code has been proven to be super wrong when 2
[550] # subsequent moves are in opposite directions, as requested
[551] # speed is constant but printer has to fully decelerate
[552] # and reaccelerate
[553] # The following code tries to fix it by forcing a full
[554] # reacceleration if this move is in the opposite direction
[555] # of the previous one
```

---

Da die Sollgeschwindigkeit im G-Code in Millimeter pro Minute angegeben ist, muss die erhaltene Zeit noch mit 60 multipliziert werden um Sekunden zu erhalten

$$(7.2) \quad t_{Sek} = 60 \cdot \frac{\sqrt{s_0^2 + s_1^2 + \dots + s_n^2}}{f}$$

In Zeile 310 aus Listing 7.1 wird der in Abschnitt 5.5 auf Seite 31 besprochene Fall berücksichtigt, dass der Druckkopf steht während das Filament zurückgezogen oder nachgeführt wird.

Ein weiterer bereits implementierter Ansatz ist GCoder<sup>3</sup>. In GCoder wird versucht, G-Code Anweisungen zu erkennen, für die eine Beschleunigung notwendig ist und nur die für diese Beschleunigungen zusätzlich angefallenen Zeiten zur bisherigen Abschätzung hinzu zu addieren. Dabei wurde für aufeinanderfolgende Anweisungen mit der gleichen Sollgeschwindigkeit zunächst keine Beschleunigung berücksichtigt. Allerdings hat sich der Ansatz selbst nach einer Korrektur als nicht immer richtig herausgestellt, weshalb die Druckzeitabschätzung mit GCoder in dieser Arbeit nicht weiter verfolgt wird. Listing 7.2 zeigt einen Auszug aus den Kommentaren des Abschnitts zur Berechnung der Druckzeitabschätzung, der sinngemäß beschreibt, dass sich diese Art Beschleunigungen zu berücksichtigen als falsch herausgestellt habe, wenn zwei aufeinanderfolgende Bewegungen in entgegengesetzte Richtungen verlaufen, da sich die programmierte Geschwindigkeit zwar nicht ändere, der Drucker aber dennoch auf Geschwindigkeit 0 verzögern und dann wieder beschleunigen müsse.

Die sonstigen Implementierung gleichen sich in der Herangehensweise insofern, dass sie aus den G-Code Anweisungen die Weglänge und Sollgeschwindigkeit extrahiert und daraus die theoretische Dauer der Bewegungen berechnen. Zum Vergleich werden so letztlich nur die Druckzeitabschätzungen von GPX<sup>4</sup> und gcode.ws herangezogen, da ihre Ergebnisse für einige Modelle recht nah an der tatsächlichen Druckzeit liegen und sich zudem das Prinzip der Berechnung nicht wesentlich von anderen unterscheidet.

In der von GPX vorausgerechneten Zeit ist allerdings auch die Zeit für das Aufwärmen der Extruderdüsen und des Druckbetts enthalten, die in der Definition der Druckzeit in dieser Arbeit ausgenommen sind. Diese zusätzliche Zeit wird über je einen multiplikativen Faktor für Düse und Bett mit der Differenz aus Soll- und Isttemperatur berechnet, wobei der Standardwert für die

<sup>3</sup><https://github.com/kliment/Printrun/blob/master/printrun/gcoder.py>

<sup>4</sup><https://github.com/markwal/GPX>

---

Isttemperatur 24 Grad Celsius beträgt. Für eine Erwärmung von 24 auf 255 Grad Celsius werden damit je Extruderdüse 138,6 Sekunden errechnet, für den Aufwärmvorgang des Druckkopfs mit 2 Düsen also 277,2 Sekunden. Bei der Berechnung der Zeit für die Erwärmung des Druckbetts ist das Ergebnis unterschiedlich, je nachdem welche der zwei Extruderdüsen ebenfalls erwärmt wird oder keine der beiden. So führt die programmierte Erwärmung des Druckbetts, falls beide Düsen ebenfalls erwärmt werden, zu einer um 14 Sekunden niedrigeren Druckzeitabschätzung als dies der Fall ist, wenn die Erwärmung des Druckbetts nicht programmiert ist. Es ist nicht ganz klar ob dies durch einen Fehler im Programmcode ausgelöst wird. Da sich aber in Versuchen herausgestellt hat, dass diese Abweichung nur von der jeweiligen Konstellation der programmierten Erwärmungen abhängt und nicht auch von den sonstigen Anweisungen des G-Code, kann der jeweils richtige Wert als „Zeit für das Erwärmen der Düsen minus 14 Sekunden“ leicht berechnet werden. GPX berechnet außerdem 20 Sekunden für die Bewegungen, die im Header der meisten G-Code Dateien aus dem Testfeld programmiert sind. Für Vergleiche der verschiedenen Methoden zur Vorausberechnung der Druckzeit wird dementsprechend für das jeweilige Modell die entsprechende Zeit von der Ausgabe von GPX subtrahiert.





# 8 Neuer Ansatz: Regressions-Hyperebene

## 8.1 Idee

Im Verlauf dieser Arbeit wurde ein weiterer Ansatz zur Druckzeitabschätzung entwickelt. Dieser basiert auf der Annahme, dass 3D-Drucker, je nach Modell und Firmware, beim Drucken des selben G-Codes jeweils unterschiedliche Zeiten erreichen. Dadurch wäre die Variation an Druckern und Firmware zu groß um sie alle mit einer einzelnen Druckzeitabschätzung abzudecken. Außerdem wäre es unrentabel eine Druckzeitabschätzung für jede existierende Kombination zu entwickeln und diese stets aktuell zu halten. Stattdessen wird versucht aus bereits ausgeführten Drucken zu lernen, diese Informationen lokal zu sammeln und auf nachfolgende Drucke zu übertragen. Dabei wird die tatsächliche Druckzeit, mit bestimmten, aus dem ausgeführten G-Code extrahierten Eigenschaften verknüpft und gespeichert. Stehen ausreichend Daten zur Verfügung, sollen die Eigenschaften aus beliebigem G-Code extrahiert werden und zu einer Druckzeitabschätzung führen können.

Um das Verfahren anzuwenden, müssen inhärent im G-Code vorhandene Eigenschaften extrahiert werden, die die tatsächliche Druckzeit potentiell beeinflussen können. Diese Eigenschaften müssen jeweils als einzelne Zahlenwerte darstellbar sein und sollten so gewählt werden, dass ihre Zahlenwerte nicht linear abhängig sind. Solche Eigenschaften werden im Folgenden „Features“ genannt. Für je eine spezifische, als G-Code dargestellte Repräsentation eines Modells, entsteht so, zusammen mit der tatsächlichen Druckzeit ein Punkt im  $n + 1$ -dimensionalen Raum, wobei  $n$  für die Anzahl der Features steht. Sind mindestens  $n$  Modelle gedruckt worden und damit ihre jeweilige tatsächliche Druckzeit bekannt, kann zwischen die Punkte eine Regressions-Hyperebene gelegt werden. Dieser Vorgang zur Berechnung der Koeffizienten der Hyperebene wird Multiple Linear Regression (MLR) [Zie16, S. 243 f.] [FKL09, S. 24 f.] oder allgemeiner auch lineare Ausgleichsrechnung [SK09, S. 274 f.] genannt. Werden dann aus G-Code, dessen Druckzeit bislang nicht bekannt ist, diese Features extrahiert und in die durch MLR erhaltene Funktion eingesetzt, ist das Ergebnis eine Druckzeitabschätzung.

Sollte ein linearer Zusammenhang zwischen den gewählten Features und der tatsächlichen Druckzeit bestehen, ist die durch MLR erhaltene Ebene stets eindeutig bestimmt. Dann ließe sich auch die Druckzeit zukünftiger Drucke exakt vorausberechnen. Wie bereits in Kapitel 5 auf Seite 29 beschrieben, ist von einer linearen Abhängigkeit und damit einem exakten Ergebnis aber nicht auszugehen.

Um diesen Ansatz zu überprüfen wurde Gex um die Möglichkeit zur Extraktion der Features sowie der Berechnung der Hyperebene erweitert und auf einem Testfeld von bereits erfolgten Drucken mit bekannter, tatsächlicher Druckzeit ausgeführt. Dabei wurde die C#-Klasse `Matrix.cs`<sup>1</sup> verwendet. Allerdings stand leider nur ein begrenztes Testfeld an Modellen beziehungsweise G-Code zur Verfügung, sodass nicht ausgeschlossen werden kann, dass mit weiteren zum Testfeld hinzugefügten Modellen oder einem anderen Testfeld, andere Ergebnisse erzielt werden.

### 8.2 Die Rolle der Features

Nach Stańczyk und Jain [SJ15] bezeichnet der Begriff „Feature“ ein „Unterscheidungsmerkmal oder einen Aspekt von etwas und wird als Synonym für Charakteristik, Qualität oder Eigenschaft verwendet [...] speziell im Bereich Data Mining und Pattern Recognition“.

In diesem Fall werden Eigenschaften eines in druckbaren G-Code umgewandelten Modells gesucht, die explizit oder implizit die Druckzeit beeinflussen. Solche Eigenschaften werden einzeln genommen oder mit anderen zusammengefasst als Features bezeichnet. Diese Features sollten für die Druckzeitabschätzung über eine Hyperebene möglichst so gewählt werden, dass ihr Wert linear mit der Druckzeit zusammenhängt sowie den jeweiligen G-Code wesentlich repräsentiert.

Folgend werden die für diese Arbeit verwendeten Eigenschaften beziehungsweise Features zusammen mit dem Grund der jeweiligen Auswahl genannt. Es wurden mehrere Testdurchläufe mit unterschiedlicher Kombination der Features durchgeführt. Allerdings wurden nicht alle theoretisch möglichen Kombinationen getestet, sondern die Suche bei Erkennen eines annähernd linearen Bezugs zur tatsächlichen Druckzeit abgebrochen. Diese Kombination ist nachfolgend aufgelistet.

In die Auswertung mit einbezogene Features:

1. Summe aller Anschlusswinkel
  - ein Maß für Stärke und Häufigkeit der nötigen positiven und negativen Beschleunigungen
2. theoretische Druckzeit
  - wichtige Charakteristik des G-Codes, sorgt für eine grundsätzliche Annäherung an den tatsächlichen Wert
3. Anzahl der Retracts

<sup>1</sup><http://blog.ivank.net/lightweight-matrix-class-in-c-strassen-algorithm-lu-decomposition.html>

Lizenz: MIT

→ ein Maß für Verweilzeiten und implizit für die Anzahl der zu druckenden Layer

#### 4. theoretische Durchschnittsgeschwindigkeit

→ ein Maß für Größe und Häufigkeit der Sollgeschwindigkeiten

Für eine Bewegung in X-, Y- oder Z-Richtung, vor oder nach einem Retract, wird ein Anschlusswinkel von 90° angenommen. Die theoretische Druckzeit wird aus sämtlichen Bewegungen der drei Achsen berechnet oder, falls sonst keine andere Bewegung stattfindet, aus der Bewegung des Filaments.

Weitere Features, die bei Versuchen geprüft aber dann vernachlässigt wurden, da sie nicht die gewünschte Genauigkeit bringen beziehungsweise eine Ungenauigkeit induzieren, sind im Folgenden aufgelistet.

Weitere, nicht in die Auswertung mit einbezogenen Features:

- Summe aus der Länge aller Bewegungen
  - vermutlich ist der Zusammenhang mit der tatsächlichen Druckzeit hochgradig nicht-linear
- Summe aus der Länge aller Retracts
  - es gibt kurze und lange Retracts, dieser Wert soll deren unterschiedliche Dauer berücksichtigen
- Summe aus den Winkeln aller Bewegungen in Bezug zum Maschinenkoordinatensystem
  - möglicherweise variiert die Dauer eines Fahrwegs bei gleicher programmierter Geschwindigkeit, je nachdem ob er in einer oder in mehreren Achsen stattfindet, dieser Wert wäre ein Maß dafür
- Anzahl der Bewegungen
  - vermutlich ist der Zusammenhang mit der tatsächlichen Druckzeit hochgradig nicht-linear
- Anzahl der gedruckten Layer
  - vermutlich ist der Zusammenhang mit der tatsächlichen Druckzeit hochgradig nicht-linear
- Summe aus Länge aller Bewegungen in Z-Richtung
  - ein Maß für die Höhe des Modells, ist in vielen Fällen linear abhängig zur Anzahl der gedruckten Layer
- normalisierte Summe aus mittlerer Bewegungsdauer je Geschwindigkeit
  - repräsentiert unterschiedliche Beschleunigungen für unterschiedliche Endgeschwindigkeiten

Eine Methode, um die effektivste Kombination aus Features zu finden, wäre, algorithmisch sämtliche Kombinationen zu testen, um ein Minimum für den Root Mean Squared Error (RMSE) zu finden. Dieser ist für  $n$  Fehlerterme  $e_i$ , definiert als

$$(8.1) \quad RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2}$$

[Zie16, S. 257]. Die Fehlerterme entsprechen in diesem Fall der Differenz einer Druckzeitabschätzung zur tatsächlichen Druckzeit.

Für die Extraktion der Features ist es wichtig, dass nur der Teil des G-Codes verwendet wird, der ausgeführt wird, während die tatsächliche Druckzeit abgestoppt wird. Sind diese beiden Faktoren nicht sehr genau aufeinander abgestimmt, kann die Relation zwischen Feature-Werten und abgestopppter tatsächlicher Druckzeit inkonsistent werden.

### 8.3 Begründung für die Auswahl linearer Regressions-Hyperebenen

Laut Zielesny [Zie16] wird „MLR üblicherweise nicht als maschinelles Lernverfahren angesehen, da es prinzipiell nicht in der Lage ist, nichtlineare Funktionen zu erzeugen“. Es ist also nicht sehr wahrscheinlich damit exakte Druckzeitabschätzungen zu erzeugen. Selbst wenn die Feature-Werte und die tatsächliche Druckzeit eines bereits gedruckten Modells mit in die Berechnung der Funktion einbezogen war, ist es unwahrscheinlich, dass dieser Punkt auf der Regressionsebene liegt. Dies hat sich durch Anwendung des Verfahrens auf dem Testfeld auch bestätigt. Dennoch fiel die Entscheidung auf eine lineare Regression. Die Gründe dafür werden im Folgenden beschrieben.

Ein lineares Verfahren leidet nicht unter den generellen Nachteilen eines nichtlinearen:

- **linear:** Extrapolation ist möglich  
**nichtlinear:** eine Extrapolation wäre nicht aussagekräftig, da der tatsächliche Polynomgrad unbekannt ist
- **linear:** kein Overfitting  
**nichtlinear:** erhaltene Funktionen bilden oft die Trainingsdaten sehr genau nach, sind zwischen den Stützpunkten aber weit vom eigentlich richtigen Wert entfernt
- **linear:** schnelle Berechnung  
**nichtlinear:** Verfahren sind oft iterativ mit unbekannter Dauer

Zudem hat man die Möglichkeit über ein lineares Verfahren, ein „Gefühl für den Grad der Nichtlinearität“ [Zie16, S. 379] der Ausgangsdaten zu bekommen. Das hier beschriebene Verfahren kann also als ein Ansatz gesehen werden, um eine Druckzeitabschätzung über

maschinelles Lernen zu erhalten. So könnte man mit Hilfe der MLR versuchen die geeignetsten Features zu finden, eine Extrapolation ermöglichen und gleichzeitig die Interpolation mit nichtlinearen Methoden zu verbessern.

## 8.4 Evaluation des Ansatzes

Zur Berechnung der Hyperebene wird aus  $n$  Modellen und  $m$  Features eine Matrix  $A \in \mathbb{R}^{n \times m+1}$  gebildet, so dass je Zeile ein Modell und je Spalte ein Feature eingetragen sind. Die Werte der ersten Spalte sind konstant 1.

$$(8.2) \quad A_{n,m+1} = \begin{pmatrix} 1 & a_{mod_1,feat_1} & a_{mod_1,feat_2} & \cdots & a_{mod_1,feat_m} \\ 1 & a_{mod_2,feat_1} & a_{mod_2,feat_2} & \cdots & a_{mod_2,feat_m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_{mod_n,feat_1} & a_{mod_n,feat_2} & \cdots & a_{mod_n,feat_m} \end{pmatrix}$$

Zusammen mit den tatsächlichen Druckzeiten, die in einen Vektor  $\vec{b} \in \mathbb{R}^n$  eingetragen sind

$$(8.3) \quad \vec{b} = \begin{pmatrix} b_{mod_1} \\ b_{mod_2} \\ \vdots \\ b_{mod_n} \end{pmatrix}$$

stellt für einen Vektor  $\vec{x} \in \mathbb{R}^{m+1}$ ,

$$(8.4) \quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ x_{m+1} \end{pmatrix}$$

$A\vec{x} = \vec{b}$  das zu lösende Gleichungssystem dar.

Für den hier besprochenen Fall ist dieses Gleichungssystem in den überwiegenden Fällen überbestimmt und kann somit nicht direkt gelöst werden, beispielsweise mit der LR-Zerlegung. Da in diesem Ansatz leichte Nichtlinearitäten ausgeglichen werden sollen, wird  $\vec{x}$  nach dem Gaußschen Ausgleichsprinzip, mit Hilfe der MLR berechnet. Naiv kann dies implementiert werden als

$$(8.5) \quad \vec{x} = (A^T A)^{-1} \cdot A^T \vec{b}$$

Für die eigentliche Berechnung der Druckzeitabschätzung werden die  $m$  Features aus einem G-Code extrahiert, in einen Vektor  $\vec{k} \in \mathbb{R}^{m+1}$  eingetragen

$$(8.6) \quad \vec{k} = \begin{pmatrix} 1 \\ k_1 \\ k_2 \\ \vdots \\ k_m \end{pmatrix}$$

und dieser mit dem zuvor berechneten  $\vec{x}$  multipliziert.

$$(8.7) \quad t_{\text{Abschätzung}} = \vec{k}^T \cdot \vec{x}$$

Falls  $A$  nicht maximalen Rang hat, kann  $\vec{x}$  nicht eindeutig bestimmt werden. Dies ist der Grund, weshalb die Feature-Werte nicht linear abhängig sein sollen. Zudem wäre die Bedeutung solcher Features grundsätzlich in Frage gestellt. Auch sollte für kein Feature eine gesamte Spalte den Wert 0 haben, was allerdings bei einer algorithmischen Berechnung, damit umgangen werden kann, dass die gesamte Spalte so lange ignoriert wird, bis für einen neuen Eintrag dieser Wert ungleich 0 ist.

Bei der Berechnung der temporären Matrix  $C = A^T A \in \mathbb{R}^{m+1 \times m+1}$  aus Gleichung (8.5), fallen je hinzugekommenem Eintrag  $(m+1)^2$  viele, zusätzliche Multiplikationen und Additionen an. Da  $C$  aber symmetrisch positiv definit ist, reicht es wenn nur die untere Hälfte berechnet wird. Für  $d = A^T \vec{b}$  kann außerdem  $C\vec{x} = d$  effizient mit der Cholesky-Zerlegung gelöst werden.

Die für die Druckzeitabschätzung mit Gex benötigte Zeit, hängt in diesem Versuch von der Größe der Eingabedatei ab, da der G-Code geparkt werden muss und die Features daraus extrahiert werden müssen. Der gesamte Vorgang dauert dabei im Durchschnitt 0,000822 Millisekunden je Byte der Eingabedatei. Das Minimum ist  $0,000641 \frac{ms}{Byte}$ , das Maximum  $0,001033 \frac{ms}{Byte}$ , die Standardabweichung beträgt 0,0001090. Diese Werte sind aber nicht auf den allgemeinen Fall übertragbar, da die benötigte Zeit unter Anderem noch von der verwendeten Programmiersprache, vom Grad der Optimierung, dem verwendeten Computer und den zu extrahierenden Features abhängt. Für diese Arbeit sollten möglichst viele Aspekte des G-Codes ersichtlich gemacht werden, weshalb Berechnungen angestellt wurden, die für eine bloße Extraktion der Features nicht notwendig sind.

GPX benötigt für kleinere Dateien mit etwa 60.000 Bytes etwa 30 Millisekunden mehr Zeit als Gex, schreibt in dieser Zeit aber noch eine Ausgabedatei. Für die mit 13.803.628 Bytes größte Datei im Testfeld, benötigt GPX 4.437 Millisekunden weniger Zeit als Gex. Es konnte kein linearer Zusammenhang zwischen der Größe der Eingabedatei und der Laufzeit von GPX gefunden werden.

### 8.4.1 Testfeld

Leider war das für diese Arbeit zur Verfügung stehende Testfeld sehr begrenzt und deckt nicht oder in nicht ausreichend großem Maße die in Kapitel 4 auf Seite 21 angesprochenen, erwünschten Merkmale ab. Die hier beschriebenen Tests können in weiteren Versuchen mit einem größeren Testfeld gezielt durchgeführt werden um allgemeingültige Ergebnisse zu erhalten, was hier leider nicht garantiert werden kann.

Im Testfeld sind alle in Kapitel 4 auf Seite 21 vorgestellten Modelle. Die Werte der daraus extrahierten Features sind in Tabelle 8.1 eingetragen. Modelle, die in Tabelle 8.1 enthalten sind, aber in Kapitel 4 nicht aufgeführt werden, sind Variationen des Modells aus Abbildung 4.4, die jeweils um die Z-Achse gedreht und in einer anderen Richtung ausgefüllt sind.

Modell	tatsächliche Druckzeit	Summe aller Anschlusswinkel	theoretische Druckzeit	Anzahl der Retracts	theoretische Durchschnittsgeschwindigkeit
maze_inside	6996,7444	10916,4224	6086,2834	19	2932,5107
maze_outside	13993,5400	33566,9346	11165,6824	181	3031,6846
galaxys5_battery	2057,2702	10282,0409	1454,0426	401	4936,6601
o-45-l-dual-flip	393,2558	1031,1857	367,6960	38	2220,3717
pattern_c_60	232,6114	943,8625	219,3909	49	2091,8526
pattern_d_10	236,7479	2226,0599	225,0023	171	3089,4131
90-l-dual-norm	399,1671	341,4843	365,8815	17	2142,7922
C10	2229,3373	2280,0175	2001,3998	70	4039,1005
B10	825,1300	39361,1820	648,5287	2445	1774,2697
J10-l	5770,5704	176967,0095	3396,4622	9670	3725,6784
E10-l	867,9004	5926,8935	534,4362	315	3636,5791
mic-bot_v4	2049,6076	12473,2024	1470,3840	686	4753,2790
o-60-l-dual-flip	408,8235	1884,6544	376,6557	93	2362,4414
I10	1198,6653	6810,6687	746,2295	371	4795,2956
pattern_c_45	228,0073	396,7398	217,1024	7	1839,8710
o-10-l-dual-norm	415,9018	4455,6799	391,1044	342	3217,0771
pattern_c_30	227,9798	617,4252	218,0385	36	2043,1579
45-l-dual-flip	393,9011	880,4205	360,3129	24	2174,2055
pattern_c_10	229,0836	2215,4826	224,5102	171	3092,3875
45-l-dual-norm	395,0561	701,9164	370,6219	9	2123,3006
30-l-dual-flip	391,2669	4326,8817	362,2143	313	2792,0032

**Tabelle 8.1:** Feature-Werte und tatsächliche Druckzeit der Modelle im Testfeld. Zeiträume sind in Sekunden, Winkel in Radianten und Geschwindigkeiten in Millimeter pro Minute angegeben.



$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
-241.90175664	0.08758456	0.99752371	-1.37113096	0.11374252

**Tabelle 8.2:** Aus den Feature-Werten errechnete Koeffizienten der Hyperebene. Die Höhe der einzelnen Werte der Koeffizienten  $x_2, x_3, \dots, x_m$ , zeigt den Einfluss der jeweiligen dazugehörigen Features [BEPW16, S. 80].

## 8.4.2 Ergebnisse aus den Versuchen

Für jedes der Modelle wurden mit Hilfe der Regressions-Hyperebene zwei Druckzeitabschätzungen durchgeführt. Zuerst wurde die Hyperebene aus sämtlichen Modellen berechnet und dann unverändert für eine Druckzeitabschätzung aller Modelle verwendet. In den unten aufgeführten Diagrammen sind die so erhaltenen Graphen jeweils gelb und mit „hyp“ beschriftet. Danach wurden die Feature-Werte und die tatsächliche Druckzeit für das jeweils abzuschätzende Teil temporär entfernt und die Hyperebene neu berechnet, um die Genauigkeit für ein bislang unbekanntes Modell herauszufinden. Diese Graphen sind jeweils grün und mit „hypexcl“ beschriftet. Für einen Vergleich der erzielten Ergebnisse wurden noch die entsprechenden Werte von GPX (blau) und gcode.ws (orange-rot) in die Diagramme mit aufgenommen.

Für eine Druckzeit von 10 Stunden hat eine Abweichung von 5 Minuten ein anderes Gewicht als für eine Druckzeit von 10 Minuten. Darum werden in Abbildung 8.1, 8.2 und 8.3 die relativen Abweichungen zur tatsächlichen Druckzeit gezeigt. Die relative Abweichung  $d_{rel}$  zur tatsächlichen Druckzeit wurde errechnet als

$$(8.8) \quad d_{rel} = \frac{\text{errechnete} - \text{tatsächlicheDruckzeit}}{\text{tatsächlicheDruckzeit}}$$

Dass die in Abbildung 8.2 gezeigte, größte relative Abweichung aller 4 Abschätzungen sehr ähnlich ist, gleichzeitig aber die größte absolute Abweichung in Abbildung 8.5 für GPX und gcode.ws sehr viel größer ist, zeigt dass die Abschätzung durch die Hyperebene bei Modellen mit relativ hoher Druckzeit genauer war. Die größte relative Differenz im exklusiven Fall (hypexcl), ist bei Modell 19 aufgetreten, bei einer Abweichung von 68 Sekunden zu der tatsächlichen Druckzeit von 228 Sekunden. Dies gilt auch im nicht exklusiven Fall (hyp), bei einer Abweichung von 64 Sekunden. Abbildung 8.9, auf der die Abweichungen für die Modelle 15 bis 21 im Vergleich zu 8.4 nochmals vergrößert abgebildet sind, bestätigt diese Aussage.

Die gute Verwendbarkeit der für dieses Testfeld gewählten Features, zeigt der Verlauf der gelben Kurve in Abbildung 8.4, in Verbindung mit der maximalen absoluten Abweichung von 132 Sekunden in Abbildung 8.5 und der durchschnittlichen Abweichung von 50 Sekunden in Abbildung 8.8. Somit wurde ein annähernd linearer Zusammenhang zwischen den Features und der tatsächlichen Druckzeit gezeigt. Dabei auffällig ist die erhöhte Genauigkeit der Abschätzung nach Hinzunehmen der Modelle 2 und 10 zur Berechnung der Hyperebene, die für weiterhin hinzugefügte Modelle eine erhöhte Genauigkeit erwarten lässt.

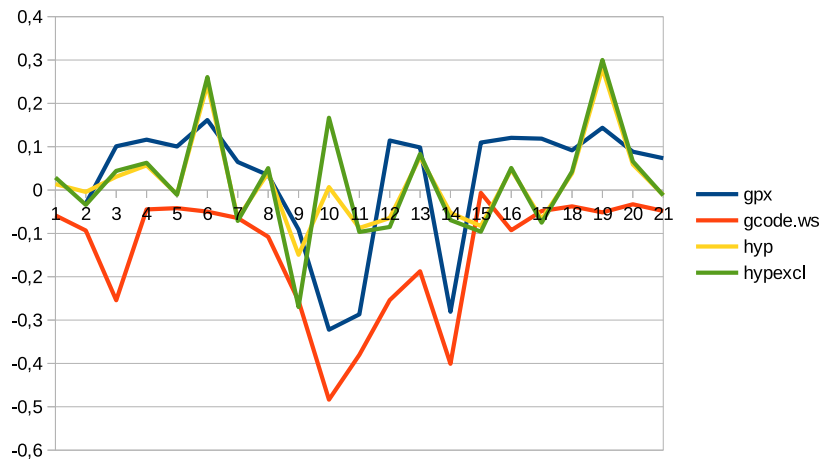
## 8 Neuer Ansatz: Regressions-Hyperbene

---

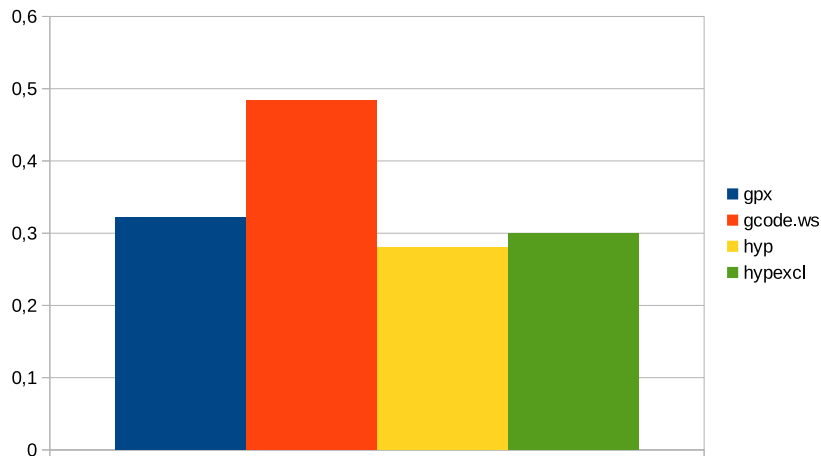
GPX	gcode.ws	hyp	hypexcl
446,653773	711,094754	61,788589	250,950245

**Tabelle 8.3:** Root Mean Squared Error im Testfeld je Verfahren zur Druckzeitabschätzung. Berechnet aus den Differenzen zur tatsächlichen Druckzeit.

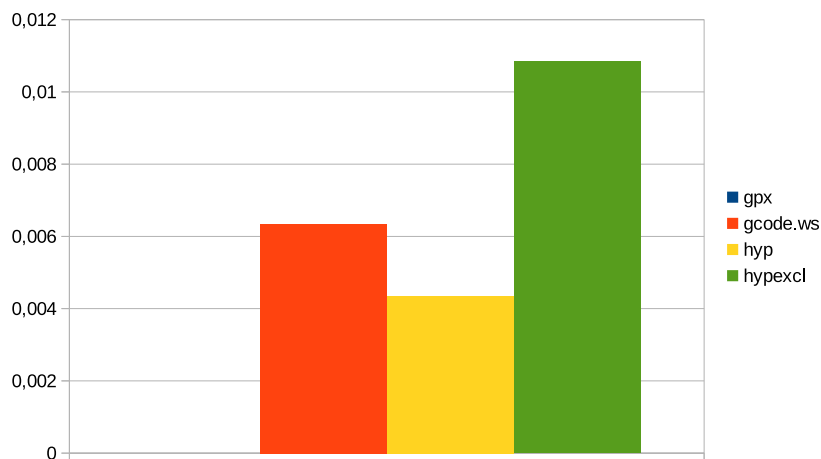
In Tabelle 8.3 ist der RMSE für die Druckzeitabschätzung mit der oben beschriebenen Feature-Kombination, extrahiert aus dem in Abschnitt 8.4.1 angegebenen Testfeld eingetragen.



**Abbildung 8.1:** Relative Abweichungen zur tatsächlichen Druckzeit je Modell. Die Reihenfolge der Modelle entspricht Tabelle 8.1

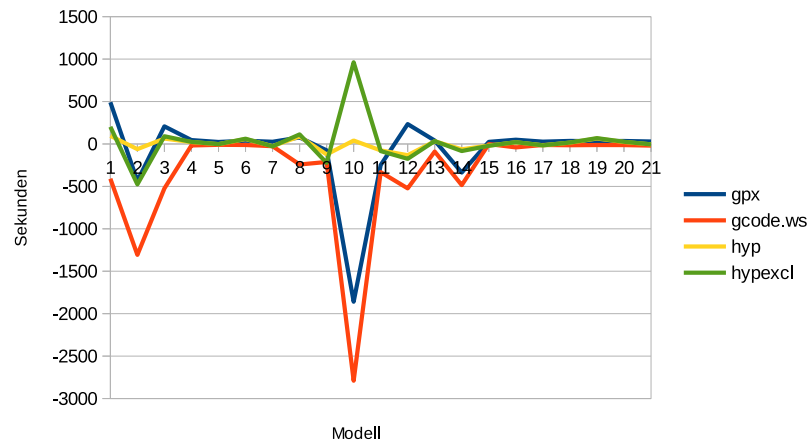


**Abbildung 8.2:** Größte absolute relative Abweichung

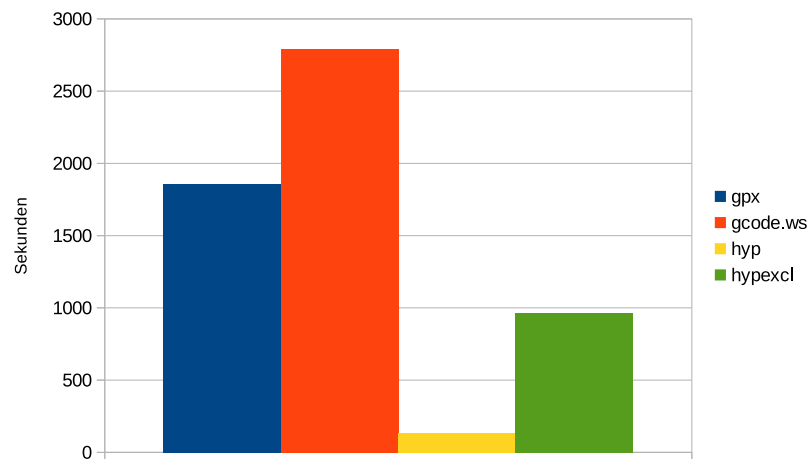


**Abbildung 8.3:** Kleinste absolute relative Abweichung

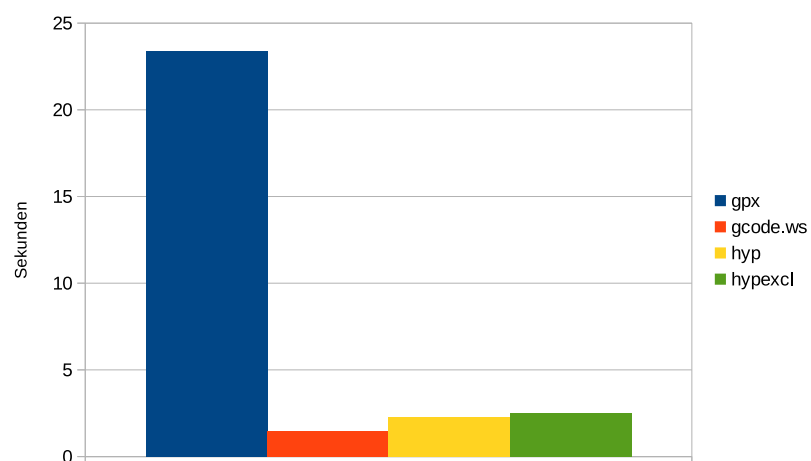
## 8 Neuer Ansatz: Regressions-Hyperbene



**Abbildung 8.4:** Abweichungen zur tatsächlichen Druckzeit je Modell



**Abbildung 8.5:** Größte absolute Abweichung



**Abbildung 8.6:** Kleinste absolute Abweichung

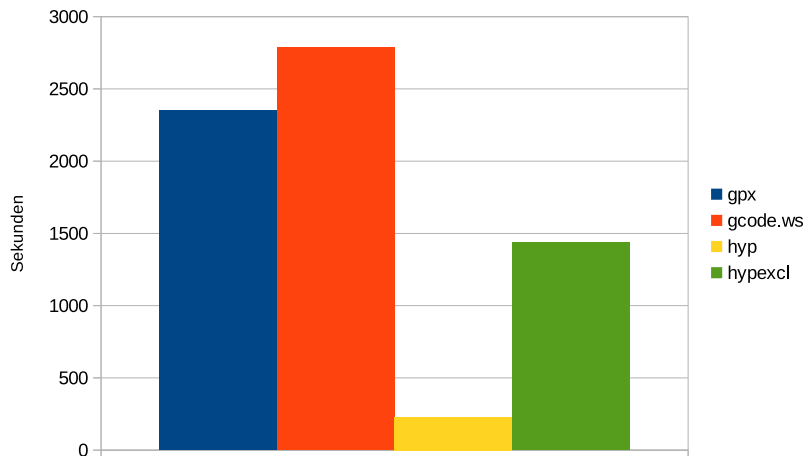


Abbildung 8.7: Größte Differenz zwischen einzelnen Abweichungen

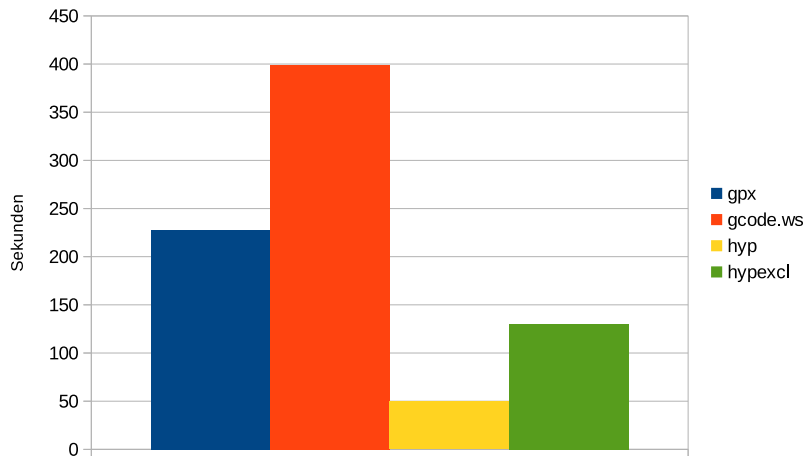


Abbildung 8.8: Durchschnittliche absolute Abweichung

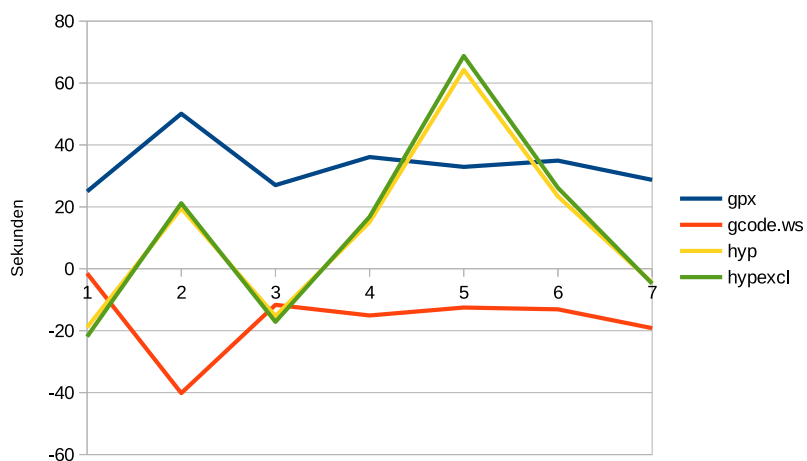


Abbildung 8.9: Abweichungen zur tatsächlichen Druckzeit für Modelle 15 bis 21



## 9 Zusammenfassung und Ausblick

Für eine exakte Berechnung der Druckzeit müssen sehr viele beeinflussende Faktoren berücksichtigt werden. Bisherige Ansätze versuchen den dafür nötigen Aufwand durch Implementieren von Heuristiken zu umgehen und erreichen dafür nur eine Druckzeitabschätzung. Sie verfolgen mit der Berechnung der theoretischen Druckzeit und deren Modifikation eine offensichtliche Strategie, vernachlässigen aber andere Faktoren, die keinen direkten Zusammenhang mit der Druckzeit haben und darum nicht einfach aufaddiert werden können. Außerdem ist der Einfluss dieser Faktoren vom jeweiligen Druckermodell und der darauf installierten Firmware abhängig und muss daher je Kombination neu evaluiert werden. Bei dem in dieser Arbeit entwickelten Ansatz über die Regressions-Hyperebene geschieht dies dadurch, dass die tatsächliche Druckzeit, mit diesen Faktoren ins Verhältnis gesetzt wird und für eine zukünftige Auswertung zur Verfügung steht. Der wichtigste Schritt besteht darin, geeignete Features zu finden, die den Einfluss dieser Faktoren proportional zur tatsächlichen Druckzeit repräsentieren. Hier wurde experimentell belegt, dass es Features gibt, die einen annähernd linearen Zusammenhang zur Druckzeit haben. Dies gilt zumindest für das hier verwendete, eingeschränkte aber dennoch recht unterschiedliche und damit verwendbare Testfeld. Da die eigentliche Funktion sehr wahrscheinlich nichtlinear ist, können auch Abweichungen für Modelle entstehen, die bei der Berechnung der Hyperebene mit einbezogen waren. Im Test war die größte Abweichung für ein bereits gedrucktes Modell 132 Sekunden. Dies wäre dadurch behebbar, dass zusammen mit den Feature-Werten ein Hash-Code gespeichert wird, über den eine nochmals gedruckte Datei identifiziert und dann die bereits bekannte Druckzeit ausgegeben wird.

Für eine weitere Untersuchung der Anwendbarkeit dieses Ansatzes könnte man die hier vorgestellten Versuche in einem größeren Testfeld mit systematisch gewählten oder selbst programmierten Versuchsmodellen nochmals durchführen. Die daraus gewonnenen Erkenntnisse können verwendet werden um weitere Features abzuleiten. Aus allen dann bekannten Features, kann die Kombination gesucht werden, für die der RMSE minimal wird. Die allgemeine Anwendbarkeit dieser Ergebnisse könnte dann auf weiteren 3D-Druckern beziehungsweise mit anderer Firmware geprüft werden. Zusätzlich könnte über eine Integration von nichtlinearen Methoden nachgedacht werden.

Im Weiteren wäre es vorstellbar diese Berechnung in die Firmware von 3D-Druckern zu integrieren, so dass die aktuellen, zur Druckzeitabschätzung benötigten Koeffizienten vom Slicer abgerufen werden können. Die für das Aufwärmen der Extruderdüsen und des Druckbetts benötigte Zeit, kann separat, durch eine lineare oder einfache polynomielle Funktion dargestellt werden. Nachdem beim Drucker die Koeffizienten dieser Funktion und die aktuellen

Temperaturen abgefragt wurden kann sie berechnet und zur Druckzeitabschätzung addiert werden. Ebenso könnte ein Online-Dienst eingerichtet werden, der die Feature-Werte und die tatsächliche Druckzeit speichert und bei einer Änderung die Koeffizienten neu berechnet. Vor dem Druck könnten die Koeffizienten zur Druckzeitabschätzung abgefragt werden, nach dem Druck kann dann die tatsächliche Druckzeit zusammen mit den Feature-Werten übermittelt werden.



# Literaturverzeichnis

- [BEPW16] K. Backhaus, B. Erichson, W. Plinke, R. Weiber. *Multivariate Analysemethoden*. Springer Berlin Heidelberg, 2016 (zitiert auf S. 49).
- [DR08] W. Dahmen, A. Reusken. *Numerik für Ingenieure und Naturwissenschaftler*. Springer Berlin Heidelberg, 2008 (zitiert auf S. 30).
- [Fas12] P. Fastermann. *3D-Druck/Rapid Prototyping*. Springer Berlin Heidelberg, 2012 (zitiert auf S. 14).
- [FKL09] L. Fahrmeir, T. Kneib, S. Lang. *Regression*. Springer Berlin Heidelberg, 2009 (zitiert auf S. 41).
- [GH16] A. Gebhardt, J.-S. Hötter. *Additive Manufacturing 3D Printing for Prototyping and Manufacturing*. Carl Hanser Verlag GmbH & Co. KG, 2016 (zitiert auf S. 13, 18).
- [GNS16] C. Gengnagel, E. Nagy, R. Stark. *Rethink! Prototyping*. Springer International Publishing, 2016 (zitiert auf S. 13).
- [GRS15] I. Gibson, D. Rosen, B. Stucker. *Additive Manufacturing Technologies*. Springer New York, 2015 (zitiert auf S. 13).
- [Rum15] E. Rummich. *Elektrische Schrittmotoren und -antriebe*. expert verlag GmbH, 2015 (zitiert auf S. 30).
- [SJ15] U. Stańczyk, L. C. Jain. *Feature Selection for Data and Pattern Recognition*. Springer Berlin Heidelberg, 2015 (zitiert auf S. 42).
- [SK09] H. R. Schwarz, N. Köckler. *Numerische Mathematik*. Vieweg+Teubner, 2009 (zitiert auf S. 41).
- [Um16] D. Um. *Solid Modeling and Applications*. Springer International Publishing, 2016 (zitiert auf S. 13).
- [Zie16] A. Zielesny. *From Curve Fitting to Machine Learning*. Springer International Publishing, 2016 (zitiert auf S. 41, 44).

Alle URLs wurden zuletzt am 27. 06. 2016 geprüft.



## **Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum, Unterschrift