
Hochschule Zittau/Görlitz

Praktikumsbeleg

# Untersuchung und Implementierung von Methoden der CAD-gestützten Roboterprogrammierung

*Dongliang Cao*

| | |
|---|---|
| Bearbeitungszeitraum | 3 Monate |
| Matrikelnummer | 217043 |
| Betreuer der Ausbildungsfirma | Dr.-Ing. habil. Fan Dai |
| Gutachter der Hochschule | Prof. Dr. Stefan Bischoff |

14. Mai 2019

**Danksagung**

Ich möchte mich beim Prof. Stefan Bischoff bedanken, dass er mein Betreuer an der Hochschule ist, und mich bei dem Praktikum unterstützt.

Ich möchte auch Dr. Dai Fan, meinem Betreuer bei ABB Forschungszentrum danken. In meinem Praktikum hat er mir bei theoretischer Fachkenntnisse und auch bei praktischer Anwendungsbereich viele Unterstützung gegeben. Er hat mir alles klar erklärt mit viel Geduld und Verständnis.

**Selbständigkeitserklärung**

Ich versichere hiermit, dass ich meine Praxisarbeit mit dem Thema „Untersuchung und Implementierung von Methoden der CAD-gestützten Roboterprogrammierung“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Mannheim, 14. Mai 2019
Ort, Datum

Dongliang Cao
Unterschrift

# Inhaltsverzeichnis

# 1 Einführung

## 1.1 Problemstellung

Die zunehmende Komplexität von Produkten und Maschinen sowie kurze Produktionszyklen bei kleinen Losgrößen stellen die Industriebranche vor große Herausforderungen. Sowohl die Programmierung von Industrierobotern im Online-Modus mit Handbediengerät als auch im Offline-Modus mit virtueller Simulation erfordert spezielle Kenntnisse in der Robotik und in fertigungsabhängigen Robotersteuerungssystemen und ist bei komplizierten Aufgaben sehr zeitaufwändig. Insbesondere klein- und mittelständische Unternehmen konfrontieren mit zusätzlichen Hindernissen wie höhe Investitionen für die Ausbildung der nicht qualifizierten Mitarbeitern und die wegen der Inbetriebnahme von Roboterzeller verlängerten Produktionszyklen.

## 1.2 Motivation

Um die Ingenieuren von den wiederholenden, eintönigen und zeitaufwändigen Programmierungsverfahren zu befreien und deshalb sich mehr auf das Projekt selbst und dessen notwendigen Operationen zu konzentrieren, benötigt neue Methode für die Erzeugung des Roboterprogrammes, die effizienter, intuitiver und benutzerfreundlicher ist.

Grundsätzlich gesagt, im Vergleich mit Online Programmierung kann OLP dank der Simulationsumgebung mehr Zeit und Bemühung für Benutzer sparen. Der Benutzter braucht nicht vor Ort den echten Roboter zu steuern, um die gewünschte Position zu erreichen und selbst zu programmieren, sondern sitzt sich vor den Bildschirm im Büro und gebt die notwendigen Positionen, Konfigurationen und Befehle in Simulationsumgebung. Nach der Simulation, die die Erreichbarkeit der gewünschten Positionen und die Kollisionsmöglichkeit überprüft, generiert das Roboterprogramm automatisch. Aber der Mangel für heutige Offline-Software ist auch eindeutig, wenn der Benutzer eine lange Reihe von Bewegungsanweisungen definieren möchten, braucht er entweder viele Zeit, um jede Position zu definieren oder weißt er nicht genaue die Positionen.

Die meiste Information über die wichtigsten Positionen für die Bewegungsanweisung beinhaltet normalerweise das Objekt, das direkt von dem Roboter bedient wird. Und die Information über das Objekt kann durch die Analyse der dahinter versteckten geometrische Daten nachzuziehen. Leider ziehen jetzt wenige Ansätze für die Kombination von CAD-Daten und Offline Programmierung.

Aus obigen Gründen möchte ich in meinem Praktikum ein Add-In-Programm

in Robotstudio entwickeln und ermöglicht die automatisch Erzeugung der Position für den Einschub von 2 Objekte.

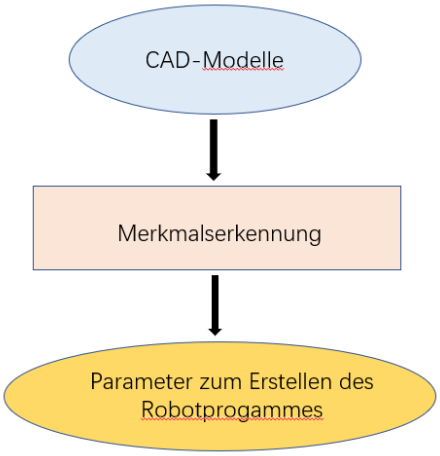


Abbildung 1: Bearbeitungsverfahren für CAD-Daten

## 1.3 Zielstellung

**Extraktion und Analyse der Information von CAD-Daten** Abruf und Analyse der geometrischen Informationen und Topologieinformationen des Objekts aus der CAD-Datei

**Filtern der Information von CAD-Daten** Ausschluss der irrelevanten Informationen und Herausfiltern der Informationen, die Beziehungen zwischen Objekten enthalten

**Feststellung der Einschubrichutung und des Einschubabstandes** Kalkulation und Bestimmung der potentiellen Einschubrichtungen mithilfe von gefilterten geometrischen Informationen und Topologieinformationen

**Visualisierung der Ergebnisse** Visualisierung der Ergebnisse in Benutzeroberfläche

# 2 Theoretische Grundlangen

## 2.1 Offline Programmierung

### 2.1.1 Definition

Die Offline-Programmierung (OLP) ist eine Roboterprogrammierungsmethode, bei der das Roboterprogramm unabhängig von der eigentlichen Roboterzelle erstellt wird. Das Roboterprogramm wird dann zur Ausführung auf den realen Industrieroboter hochgeladen. Bei der Offline-Programmierung wird die Roboterzelle durch ein grafisches 3D-Modell in einem Simulator dargestellt. Heutzutage helfen OLP Roboterintegratoren dabei, die optimalen Programm für den Roboter zu erstellen, um eine bestimmte Aufgabe auszuführen. Bei der Simulation des Roboterprogramms können Roboterbewegungen, Erreichbarkeitsanalysen, Kollisions- und Beinahe-Erkennung sowie Zykluszeitberichte berücksichtigt werden.[1]

**Vorteile**

1) Die Offline-Programmierung bricht die Produktion nicht ab, weil das Programm für den Roboter außerhalb des Produktionsprozesses auf einem externen PC geschrieben wird.

2) Integratoren und Endbenutzer können beim Entwurf einer Arbeitszelle Zeit und Geld sparen im Vergleich mit Online-Programmierung.

3) Die Fähigkeit zu analysieren, wie sich eine Arbeitszelle verhält, bevor Zeit und Geld in Geräte investiert werden, sorgt für eine reibungslose Umsetzung vom Konzept zur Realität.

**Vorgehensweise mit Offline-Programmierung**

1) **Erstellung der Arbeitszelle**
   Eine Arbeitszelle bzw. Roboterzelle bezieht sich auf eine Kombination von einem oder mehreren Robotern und das damit verbundene Werkzeuge, andere Werkstücke und Vorrichtungen. Wenn man eine Arbeitszelle erstellt, soll man zuerst alle Komponente in der Arbeitszelle importieren und danach platzieren. Außerdem muss der Roboter mit einer virtuellen Steuerung verbinden, um zu programmieren.Ein Werkzeug ist ein spezielles Objekt (z. B. eine Lichtbogenschweißzange oder ein Greifer), das an einem Werkstück arbeitet. Für korrekte Bewegungen in Roboterprogrammen müssen die Parameter des Werkzeugs in den Werkzeugdaten angegeben werden. Der wesentliche Teil der Werkzeugdaten ist der Werkzeugarbeitspunkt (TCP).

2) **Programmierung von Robotern**
Bevor der Benutzer ein Programm für seinen Roboter erstellen, solltet er die vorher genannte Arbeitszelle einrichten, in der sein Roboter arbeiten soll, einschließlich der Roboter, Werkzeug und Vorrichtungen.

das Verfahren für die Programmierung von Robotern kann sich hauptsächlich in 5 Schritte unterteilen.

- Erstellen von Positionen und Bahnen
- Prüfung der Positionsorientierung und Erreichbarkeit
- Synchronisieren des Programms mit der virtuellen Steuerung
- Ausführen von textbasierter Bearbeitung
- Kollisionserkennung

3) **Simulieren von Programmen**
Mit Simulationen werden vollständige Roboterprogramme auf einer virtuellen Steuerung ausgeführt.Durch Simulation kann die Zykluszeit berechnet,die Kollision erkannt, die E/A-Signale simuliert und auch die Ereignisse (Aktion mit einem Trigger verbunden) behandelt werden, um festzustellen, ob das Robotersystem die Erwartung von Endbenutzer erfüllt.

4) **Aufladung des Programmes in realer Steuerung**
Wenn die Simulation erfolgreich durchgeführt wird, kann der Benutzer das automatisch generiert Programm von Computer in realen Roboter aufladen.

### 2.1.2 Robotstudio

Robotstudio ist ein typisches Anwendungsbeispiel für die Offline-Programmierung (OLP), das von ABB entwickelt und unterstützt wird. RobotStudio ermöglicht der Benutzer das Arbeiten mit einer Offline-Steuerung. Dabei handelt es sich um eine virtuelle IRC5-Steuerung, die lokal auf dem PC ausgeführt wird. RobotStudio basiert auf dem so genannten Virtual Controller, einer exakten Kopie der Originalsoftware, die den Roboter in Produktionsprozessen steuert. So sind realistische Simulationen möglich, denn zum Einsatz kommen die Daten und Konfigurationen, die auch in der realen Produktion zum Einsatz kommen.[3] In meiner Arbeit, wird Robotstudio als eine Benutzeroberfläche zur Interaktion und Visualisierung von generierten Ergebnissen bedient. Außerdem wird Robotstudio Software Development Kit (SDK) als Programmbibliothek unter .NET Framework für die Entwicklung des Add-In-Programmes verwendet.

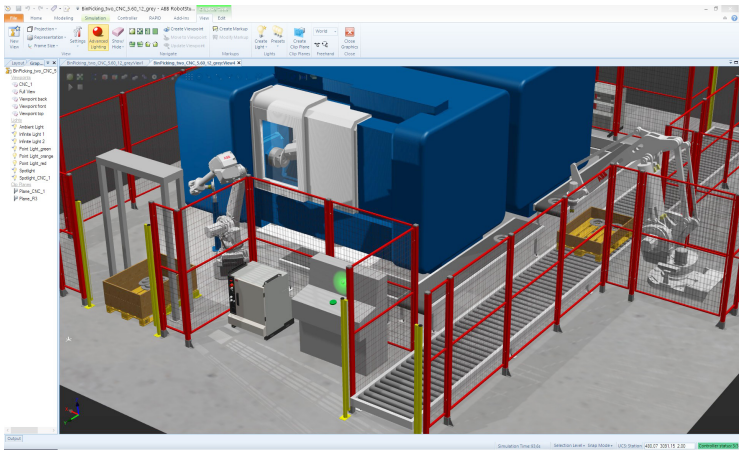

Abbildung 2: Roboterzelle in Robotstudio

## 2.2 Geometrie und Topologie

Geometrie und Topologie sind zwei wichtigste Begriffe für die mathematische Darstellung der räumlichen Information von Objekte in der realen Welt. Die Geometrie bezieht sich in diesem Artikel besonders auf die euklidische Geometrie, die sich mit Punkten, Geraden, Ebenen, Abständen, Winkeln usw. beschäftigt, sowie diejenigen Begriffsbildungen und Methoden, die im Zuge einer systematischen und mathematischen Behandlung dieses Themas entwickelt wurden.[2]Neben der mathematischen Beschreibung der räumlichen Lage und Form von einzelnen Komponenten, beschreibt die Topologie die Lagebeziehung zwischen Geoobjekten, wie z.B. die Knoten, Kante und Masche. In einfachen Systemen entsprechen den Punkten die Knoten, den Linien die Kanten und den Flächen die Maschen.[4]

Im meisten CAD-Datei wird boundary representation (b-rep), auf deutsch Begrenzungsflächenmodell, als Modellierungsform eines Flächen- oder Volumenmodells verwendet. Ein typisches Beispiel: Eine Fläche ist ein begrenzter Teil einer Oberfläche. Eine Kante ist ein begrenzter Teil einer Kurve und ein Scheitelpunkt liegt an einem Punkt. In der Welt des Datenaustauschs definiert the Standard for the Exchange of Product Model data (STEP) auch einige Datenmodelle für b-rep. Die allgemeinen generischen topologischen und geometrischen Modelle sind in der geometrischen und topologischen Darstellung nach ISO 10303-42 definiert.[5]

### 2.2.1 Geometrie

### 2.2.2 Topologie

## 2.3 CAD Modellierung

Verschiedene CAD-Software unterstützen unterschiedliche CAD-Dateien. In SolidWorks haben sie beispielsweise proprietäre Formate wie ein Teil, der als ".prt"definiert ist, und eine Baugruppe, die als ".asm"gespeichert ist. Die verschiedenen proprietären CAD-Dateiformate werden von unterschiedlichen CAD-Konstruktionssoftware wie Pro Engineer, SolidWorks und AutoCAD verwendet. Wegen der Vielfalt von CAD-Dateiformat, ein neutrales CAD-Dateiformat, mit dem die Daten zwischen verschiedenen CAD-Programmen ausgetauscht werden können, zu finden, ist für die Einarbeitung der CAD-Datei sehr wichtig.

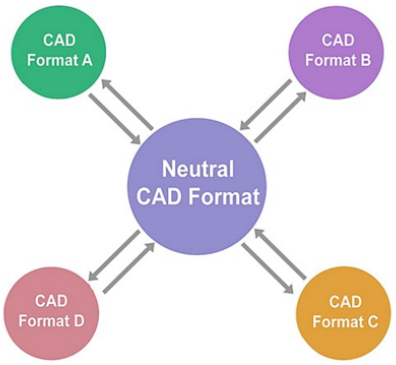


Abbildung 3: neutrales CAD-Dateiformat hilft Datenaustausch

Derzeit eines der beliebtesten CAD-Dateiformat ist STEP.Die Vorteile von STEP spiegelt sich hauptsächlich in den folgenden vier Punkten wieder.

1) Ermöglicht das Anzeigen und Ändern von Geometrie mithilfe eines beliebigen CAD-Tools, das STEP-Geometrie interpretieren kann, und unterbricht die Abhängigkeit zwischen CAD-Systemen und Produktdefinition

2) Enthält das Assembly-Schema, das alle Connector-Entitäten enthält

3) Nützlich zum Gruppieren von mechanischen Elementen in Bestimmte Ansicht

4) Einfache Ableitung der impliziten Informationsinhalte.

# 3 Aufgabenstellung

## 3.1 Addin-Entwicklung in Robotstudio

## 3.2 CAD-Modell Analyse Mithilfe von ACIS Modeler

## 3.3 Einschubrichtungerkennung

# 4 Methoden

# 5 Ergebnis und Disskusion

# 6 Zusammenfassung

# 7 Ausblick

# Literatur

[1] wikipedia:Off-line programming (robotics)

[2] wikipedia:geometrie

[3] ABB:Offline-Programmierung leicht gemacht!

[4] wikipedia:Geodaten

[5] wikipedia: boundary representation

# Abbildungsverzeichnis

# Abkürzungsverzeichnis

**OLP** Offline-Programmierung . . . . . 4

**TCP** Werkzeugarbeitspunkt . . . . . 4

**SDK** Software Development Kit . . . . . 5

**b-rep** boundary representation . . . . . 6

**STEP** the Standard for the Exchange of Product Model data . . . . . 6