

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Master Seminar Ausarbeitung - Thesis outline

Bernd Pohlmann

Entwicklung einer MATLAB-basierten
Simulationsumgebung für visuell geführte Roboter
2011

Bernd Pohlmann

Thema der Master Seminar Ausarbeitung - Thesis outline

Entwicklung einer MATLAB-basierten Simulationsumgebung für visuell geführte Roboter

Stichworte

Roboter, Simulation, Trajektorie, Matlab, VRML, Kinematik, 3D-Modell, OpenGL

Kurzzusammenfassung

Diese Arbeit beschäftigt sich mit der Entwicklung und Implementierung einer Roboter Simulation. Nach der Einleitung wird auf die Vorarbeiten der letzten Semester eingegangen, sowie die geplante Architektur der Anwendung vorgestellt. Anschließend werden die einzelnen Komponenten vorgestellt. Dazu gehören die Kinematik und Trajektorie sowie 3D-Modellierung und Bildverarbeitung. Abschließend findet eine Risikobetrachtung statt und die Arbeit endet mit einer kurzen Zusammenfassung.

Title of the paper

Developing a MATLAB-based simulation environment for visual controlled robots

Keywords

robot, simulation, trajectory, Matlab, VRML, kinematics, 3D-model, openGL

Abstract

This paper is about the development and implementation of a robot simulation. After a short introduction and motivation the prework of the last semesters are reviewed, followed by introducing the planned architecture of the application. After that the different modules are discussed. These include kinematics and trajectory as well as 3D modelling and image processing. Finally the risk assessment is on and the paper closes with a short summary.

Inhaltsverzeichnis

1 Motivation	1
2 Ziel der Arbeit	1
3 Vorarbeiten	2
3.1 Anwendungen 2	2
3.2 Projekt 1	4
4 Architektur und Komponenten	4
5 Vorgehensweise	6
5.1 Erzeugen des 3D-Modells	6
5.2 Anwendung der inversen Kinematik	7
5.3 Implementierung der Bildverarbeitung	8
5.4 Die Simulationsumgebung	9
6 Risikobewertung	10
7 Zusammenfassung	10

Abbildungsverzeichnis

4.1	Module der Umgebung (Klassendiagramm)	5
5.1	Der Katana-Fuß in zwei Detailstufen	7
5.2	DH-Matrix	8
5.3	Robotics Toolbox Example - Scitos G5	8

1 Motivation

Die Entwicklung von Algorithmen zur Steuerung von Robotern, insbesondere visuell geführten Manipulatoren, ist ein komplexes Fachgebiet. An der HAW Hamburg wird zur Zeit an diversen Projekten zu diesem Thema gearbeitet. Es gibt auf der einen Seite einfache Roboter wie die Carolo Cup Fahrzeuge, die im Wesentlichen über zwei Freiheitsgrade verfügen und auf der anderen Seite Knick-Arm Roboter mit ≥ 5 Freiheitsgraden. Die Plattform mit dem höchsten Schwierigkeitsgrad ist der Scitos G5, ein mobiler Assistenzroboter mit sieben Freiheitsgraden sowie Laserscanner, Ultraschallsensor und Kamera zur Navigation.

Da nicht alle Aspekte von einem Studenten bearbeitet werden können, ist man auf Simulation angewiesen, um parallel entwickeln zu können. Dabei ist die Visualisierung eine wichtige Aufgabe. Zur Zeit gibt es lediglich eine rein numerische oder eine graphisch nahezu unbrauchbare Methode, den Roboter und seine Bewegungsabläufe zu visualisieren. Dabei werden weder Techniken wie VRML¹ oder der Standard OpenGL² eingesetzt.

Ein weiterer wichtiger Aspekt, wenn nicht gar der wichtigste, bei der Entwicklung an einem Roboter ist die Sicherheit. Der Scitos G5 von MetraLabs³ ist für eine Maximallast von ca. 3kg zugelassen, wobei er dabei auf den Zehntel Millimeter genau ist. Er kann jedoch ohne Probleme 500N Kraft ausüben, was sowohl für den Menschen als auch für die Umwelt durchaus eine Bedrohung darstellen kann.

2 Ziel der Arbeit

Ziel der Arbeit ist die Entwicklung einer Simulationsumgebung für visuell geführte Roboter. Eingesetzte Sensoren können Kameras oder auch Laserscanner aller Art sein. Es soll ein 3D-Modell des Scitos G5 geschaffen und dieses mit den Algorithmen, welche zuvor in MATLAB entwickelt wurden, möglichst realitätsnah animiert werden. Auf Basis der Erfahrungen mit dem Katana 6M aus Projekt 1 und 2 soll eine Umgebung entstehen, welche mathematische Präzision und detailgetreue Simulation miteinander kombiniert. Durch den begrenzten Einsatz von Hardware sollen mehrere Detailstufen realisiert werden, so dass der gewünschten Performance und den (Echt-)Zeit-Anforderungen Rechnung getragen wird.

Zudem muss die Anwendung modular, möglichst durch eine Subsumption-Architektur⁴, aufgebaut werden. Zum einen müssen Umgebung, Gegenstände, Roboter und Algorithmen einfach austauschbar sein und zum anderen wird gefordert, dass sich einzelne Module dazuschalten bzw. ausschalten lassen.

Ein weiteres Ziel ist die Auswahl der richtigen Technologie, welche neben MATLAB zum

¹Virtual Reality Modelling Language

²Open Graphics Library

³<http://www.metralabs.com>

⁴Unterteilung eines Gesamtverhaltens für einen Roboter in viele Einzelkomponenten (Module)

Einsatz kommt. Da die Simulationsmöglichkeiten innerhalb von Simulink beschränkt sind, müssen andere Produkte in Betracht gezogen werden. Einerseits betrifft dies die graphische Umsetzung. Hier bietet MATLAB zwar die Darstellung von Ausgaben via OpenGL, die Simulation verwendet jedoch die in die Jahre gekommene VRML. Hier könnten externe Werkzeuge oder auch eine eigene Implementierung, vorzugsweise in JAVA oder C++, verwendet werden. Andererseits besteht die Möglichkeit, die MATLAB Module wie die Robotics- oder die Image Processing Toolbox in eine Bibliothek umzuwandeln. So kann MATLAB Funktionalität in anderen Programmen eingebunden werden, die eventuell eine bessere Umgebung für die Simulation bieten.

Unabhängig von der konkreten Implementierung muss die Software mindestens auf Microsoft Windows lauffähig sein, Plattformunabhängigkeit ist wünschenswert, aber nicht zwingend erforderlich. Die Umgebung muss für die Studenten der HAW leicht zu verstehen und zu verwenden sein, so dass zukünftige Entwicklung an und mit Robotern an der HAW parallel ohne Weiteres an ihr durchgeführt werden können.

3 Vorarbeiten

Während der zuvor absolvierten Semester wurde sich eingehend mit der Thematik der Funktionsweise und Simulation von Robotern auseinander gesetzt. Da für den Katana 6M zahlreiche Beispiele vorhanden waren und diese Technik an der HAW beherrscht wird, bietet er die optimale Plattform für Arbeiten mit dem Scitos 5G, welcher gleich drei Freiheitsgrade mehr besitzt. Im Folgenden wird auf die Untersuchungen von verwandten Projektarbeiten eingegangen sowie erste eigene Entwicklungen aus dem Projekt 1 vorgestellt.

3.1 Anwendungen 2

Während der Analyse diverser verwandter Arbeiten wurden einige relevante Projekte untersucht. Darunter befindet sich eine Arbeit von Studenten an der University of Malaysia, die sich mit der Simulation eines Mitsubishi Knick-Arm Roboters beschäftigt. Ziel war es, den Manipulator mit MATLAB in 3D darzustellen, sowie ihn durch direkte Kinematik zu steuern und das Ergebnis über VRML darzustellen. Daraus ergeben sich einige Aspekte, welche auch für diese Arbeit interessant sind:

- Erstellung des 3D-Modells / Reverse Engineering
- 3D Simulation mit Matlab und Simulink
- Verknüpfung von Simulation Roboter Steuerung

Das Team um Arya Wirabhuana verfügte nur über den Roboter selbst, nicht aber über Zeichnungen oder gar 3D-Teile aus einer CAD⁵ Anwendung. Um nun ein 3D-Modell zu erzeugen, wurde der Roboter in seine Einzelteile zerlegt, vermessen und wieder zusammengesetzt. Dies ist das einfachste Verfahren des Reverse Engineering. Mit den gewonnenen Daten wurde ein 3D-Modell für Simulink erzeugt. Als einziges unterstütztes Format wurde VRML genutzt, wobei die Geometrie dabei aus Performance Gründen stark vereinfacht werden musste. Die direkte Kinematik, oder auch Vorwärts-Kinematik war bereits vorhanden und wurde über ein Benutzerinterface mit Slidern zur Verfügung gestellt.

Das Ergebnis war eine kleine, vollständig in MATLAB integrierte Anwendung, die Eingaben direkt in die korrekte Pose des 3D-Modells umsetzen konnte. (Wirabhuana und bin Haron, 2004)

Aufgrund des stark vereinfachten Modells, der Tatsache das OpenGL VRML stark überlegen ist und der nicht vorhandenen Automatisierung, ist eine solche Implementierung für diese Arbeit sehr unwahrscheinlich. Grundsätzlich wurde jedoch gezeigt, dass MATLAB Simulink über die Möglichkeit verfügt, Roboter zu Simulieren. Die Grundbedingung für die Umsetzung einer eigenen Simulationsumgebung ist somit erfüllt.

Zwei weitere Projekte, welche sich mit visueller Orientierung beschäftigen, wurden ebenfalls betrachtet. Dies ist zum einen eine Implementierung von zwei Studenten der University of Essex, welche sich mit der Simulation von autonomen Fahrzeugen in MATLAB auseinandersetzt und zum anderen eine Bachelorarbeit der UNI Karlsruhe zum Thema Simulation eines Laserscanners, ebenfalls mit MATLAB. Die beiden Arbeiten zeigen weitere Möglichkeiten für die Ausprägungen einer Simulation. Besondere Aspekte sind hier die Orientierung mit Kameras sowie Laserscannern und Ultraschallsensoren sowie die Kollisionserkennung und -vermeidung. Da eine Kernaufgabe meiner Arbeit in der Einbindung der optischen Sensoren liegt, kommt diesen Arbeiten besondere Beachtung zu.

Im ersten der beiden Arbeiten steht die Navigation des Roboters im Vordergrund. Die gesamte Anwendung wurde vollständig modular aufgebaut, so dass einzelne Sensoren und/oder Algorithmen nach Belieben hinzu geschaltet werden können, um das Verhalten des 2 DOF⁶ Roboters in verschiedenen Szenarien simulieren zu können. Konkret werden dabei sog. Beacons an bestimmten Stellen in einem virtuellen Raum gesetzt, welche als Orientierungspunkte dienen. Der Roboter hat einen eingeschränkten Sichtbereich und kann dadurch nicht alle finden. Ziel ist es, den Raum zu kartografieren und dabei die zurückgelegte Bahn und die Beacons⁷ sichtbar zu machen.

Diese Arbeit dient als Inspiration für den Aufbau der eigenen Simulationsumgebung. Sie zeigt ebenfalls ein breites Spektrum an Methoden, welche zur Navigation eingesetzt werden können. (Shen und Hu, 30. Juli 2010)

⁵Computer Aided Design

⁶degree of freedom - Freiheitsgrad

⁷Befehlsgebung, Signal

Die Simulation eines Laserscanners, wie sie in der angesprochenen Bachelorarbeit implementiert wurde, ist ebenfalls interessant für meine eigene Arbeit, da die Zielplattform, der Scitos G5, ebenfalls über einen solchen verfügt. (Aufgabenbeschreibung⁸)

3.2 Projekt 1

In Projekt 1 wurde der Katana 6M Roboter der HAW virtualisiert und erste Tests und Analysen durchgeführt, um die Machbarkeit der Masterarbeit zu verifizieren.

Ziel war es zunächst, den Manipulator mit Kamera auf dem End-Effektor, der bereits in der Simulation mit der Robotics Toolbox (Corke, 2008) einwandfrei arbeitet, besser darstellen zu können. Da die dreidimensionale Darstellung nur aus Linien für Armeile und blauen Zylindern für die Gelenke, waren Auswertungen nur begrenzt möglich. Um die Möglichkeiten der Analyse zu verbessern, musste das Modell in eine der Realität nähere Form gebracht werden. MATLAB bietet an dieser Stelle die Virtual Reality Toolbox an, mit der VRML Modelle erzeugt und animiert werden können.

Die erste Aufgabe bestand nun darin, das 3D-Modell im STEP⁹ Format, welches von Neuronics bereitgestellt wurde, mithilfe von CAD und einem Konvertierungsprogramm in VRML umzuwandeln. Anschließend wurde ein Szenegraph aus fünf Transforms erstellt, welcher die exakte Nachbildung der Geometrie repräsentiert. Nun wurde die bereits existierende inverse Kinematik am Modell getestet und es stellte sich heraus, dass trotz Performance-Einbußen, eine brauchbare Animation entstanden ist.

Die Aufgabenstellung von Projekt 1 umfasste ebenfalls die Erstellung einer Architektur nach Vorgabe von Rodney Brooks¹⁰, sowie die Erstellung eines Algorithmus zur Umrechnung der Kamera-Koordinaten in Basiskoordinaten des Roboters.

Das Ergebnis der Arbeit zeigt, dass die Umsetzung einer Simulationsumgebung in MATLAB realisierbar ist. Es wurde jedoch auch deutlich, dass der in die Jahre gekommene VRML Standard Performance Schwächen zeigt und eine bessere Umsetzung mit OpenGL erreicht werden könnte.

4 Architektur und Komponenten

Wie bereits erwähnt, wird die Subsumtion Architektur als Vorbild für den Aufbau der Simulationsumgebung angestrebt. Dies bedeutet im Falle dieser Entwicklung jedoch nicht nur, dass einzelne Module an- oder abgeschaltet werden können, sondern auch, dass die Komponenten einfach durch andere zu ersetzen sind. Dies ist insbesondere wichtig, da die Simulationsumgebung für alle Studenten der HAW, welche sich mit Robotik beschäftigen, bereitste-

⁸<http://www.mrt.uni-karlsruhe.de/1737.php>

⁹Standard für den Austausch von Produkt Daten

¹⁰Entwickler der Subsumption Architektur, 1986

hen soll. Auf lange Sicht wird es eine Vielzahl von Roboterplattformen und Algorithmen für Kinematik, Trajektorie und Bildererkennung /-verarbeitung geben, für die nicht jedes mal die gesamte Anwendung angepasst werden soll. Dies führt im ersten Ansatz, welcher in Projekt 1 entstanden ist und in Projekt 2 weiter ausgearbeitet werden soll, zum in Abb. 4.1 dargestellten Diagramm. Die Grafik ist zur Zeit keinesfalls als vollständig zu betrachten, da sich die Architektur noch in der Entwicklung befindet. Das Modul *Simulations Umgebung* stellt den

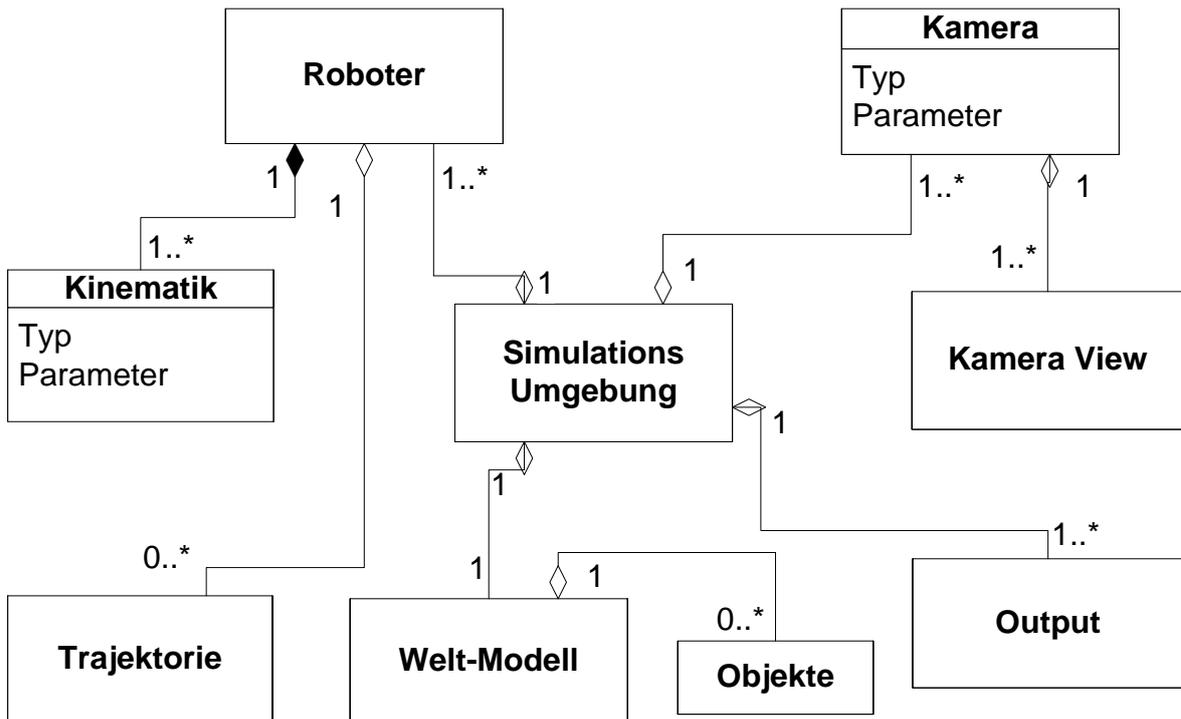


Abbildung 4.1: Module der Umgebung (Klassendiagramm)

Kern der Anwendung dar. Hier werden die Schnittstellen zu allen weiteren Modulen eingebettet. Dies ist auch, die einzige Komponente, welche nur einmal existiert. So wird es möglich sein, mehrere Roboter in einer *Welt* koexistieren und agieren zu lassen. Daneben wird es im Wesentlichen vier Module geben:

1. *Die Welt* stellt die Umgebung dar, in welcher sich die Roboter bewegen können. Diese kann von einem leeren Raum bis hin zur original getreuen Nachbildung des Living Place¹¹ alles beinhalten.
2. *Der Roboter* ist ein komplexes Modul bestehend aus Kinematik und Trajektorie. Hier wird sowohl das 3D-Modell bereitgestellt, als auch die Algorithmen für die Bewegungs-

¹¹http://livingplace.informatik.haw-hamburg.de/blog/?page_id=47

abläufe. Im Falle des Scitos G5 ist zum Beispiel denkbar, sowohl eine inverse Kinematik, welche die mobile Plattform und den Arm getrennt betrachtet als auch den Roboter als *eine kinematische Kette* zu implementieren. Bewegungsstrategien für die Bahnplanung / Trajektorie können ebenfalls eingebunden werden. Dieses Modul ist insbesondere für den autonomen Betrieb erforderlich.

3. *Die Kamera* steht stellvertretend für jede Form von visueller Sensorik. Der Kameraview für die Perspektive, also die Pose der Kamera. Sie wird im Kernmodul mit der Kinematik des Roboters vereint. So können Sensoren des Roboters exakt positioniert werden. Als visuelle Sensoren sind diverse Kameras und Laserscanner angedacht.
4. Ein weiteres Modul stellt *Output* dar. Hier werden die Ergebnisse dargestellt. Die Möglichkeiten reichen von textueller Ausgabe der Koordinaten des End-Effektors bis hin zu einer detaillierten 3D-Animation.

Die Verarbeitung der Sensordaten wird aller Voraussicht nach innerhalb der Simulationsumgebung stattfinden, um alle Module miteinander zu synchronisieren.

5 Vorgehensweise

Für diese Arbeit ist ein Basiswissen in allen erwähnten Disziplinen erforderlich. Die Themengebiete sind Simulation (mit MATLAB), Robotik, 3D-Bildverarbeitung und CAD. Im Folgenden werden die wichtigsten Teilschritte der Arbeit erläutert. Aufgrund der Tatsache, dass das endgültige Vorgehen zum Zeitpunkt dieser Arbeit noch nicht feststeht, werden hier nur die Hauptbestandteile vorgestellt.

5.1 Erzeugen des 3D-Modells

Wie bereits in Abschnitt 3.2 beschrieben, stellt Neuronics¹² bereits ein 3D-Modell für den Katana zur Verfügung. Ein solches ist für den Scitos G5 leider (noch) nicht verfügbar. Es wird jedoch davon ausgegangen, dass es zu Beginn der Arbeit bereitsteht.

Das 3D-Modell soll in drei Detailstufen erzeugt werden, da grundsätzlich gilt: Je höher die Details, desto niedriger die Performance in der Simulation bei konstantem Hardware Einsatz. Im ersten Schritt wird das STEP Modell in n Bauteile zerlegt. Anschließend werden davon Zeichnungen abgeleitet und die Parts vermessen. Auf dieser Grundlage wird dann ein Grundmodell konstruiert. Diese Schritte werden mit der kommerziellen CAD Software CATIA v5 von Dassault Systems¹³ ausgeführt. Das Grundmodell als solches stellt die kleinste Detailstufe dar. Es ist sehr grob (wenig Polygone) und enthält nur die Form ohne Bohrungen

¹²http://www.neuronics.ch/cms_en/web/index.php?identifier=homepage

¹³<http://www.3ds.com/de/products/catia/welcome/>

und andere Details. Es gibt keine Lichtquellen oder licht-sensitive Oberflächen. Die mittlere Detailstufe enthält die meisten sichtbaren Features, wie Rundungen, Ausformungen und Bohrungen. Die dritte und letzte Stufe ist ein Modell mit allen sichtbaren Details und implementierten Materialeigenschaften, wie Reflexion oder Oberflächenbeschaffenheit. Um die Performanceunterschiede messbar zu machen, wird eine kleine Animation mithilfe der Open Source Anwendung Blender¹⁴ verwendet. Das folgende Bild zeigt den Unterschied zwischen Stufe eins und zwei des Katana Fußes.

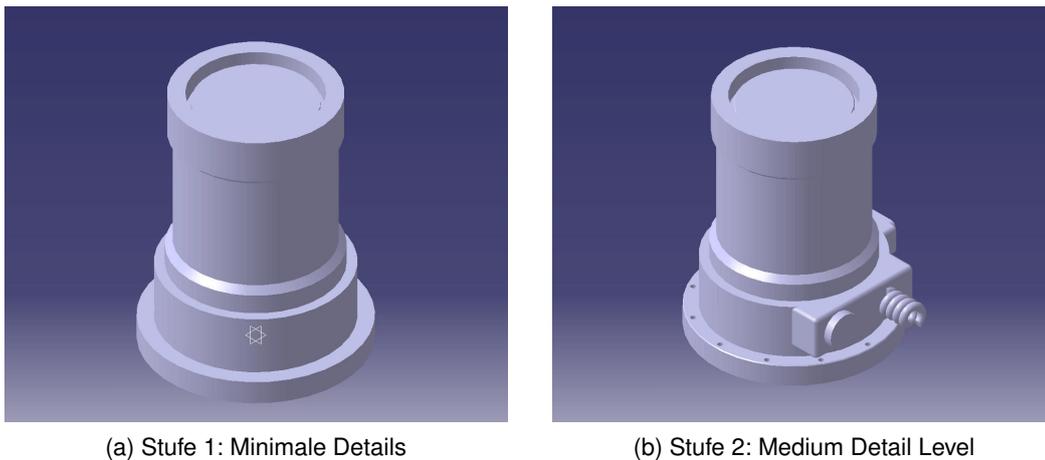


Abbildung 5.1: Der Katana-Fuß in zwei Detailstufen

5.2 Anwendung der inversen Kinematik

Die inverse Kinematik für den Katana 6M ist bereits entwickelt und getestet. Da dieser Knick-Arm Roboter nur über fünf Freiheitsgrade verfügt, ist sie hinreichend einfach. Der Scitos G5 verfügt jedoch über acht Freiheitsgrade, seine Plattform über drei (Translation von x und y sowie Rotation um die z-Achse) und der Greifarm über weitere fünf. Dies macht die Kinematik ungleich schwerer zu beherrschen, da diese dadurch redundant wird. Es gibt eventuell eine, keine oder aber unendlich viele Lösungen, um ein Ziel zu erreichen. (Bruno Siciliano, 2008) Diese Aufgabe wird zur Zeit in einer anderen Arbeit behandelt, die die Erstellung einer allgemeinen inversen Kinematik für Knick-Arm Roboter als Zielsetzung hat¹⁵.

Das Ergebnis wird mit hoher Wahrscheinlichkeit in dieser Arbeit verwendet. Zunächst wird jedoch nur die einfachere, direkte Kinematik angewandt, um den Roboter zu steuern. Dies heißt, dass die Gelenkwinkel vorgegeben werden, und daraus eine Zielpose erreicht wird.

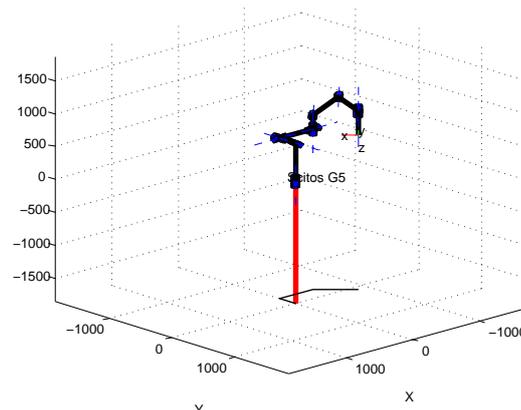
¹⁴<http://www.blender.org>

¹⁵Siehe hierzu die Ausarbeitungen von Christoph Schmiedecke

Dies wird mit der Robotics Toolbox von Peter Corke realisiert. Die sich aus den Abmessungen ergebende Denavit-Hartenberg Matrix (Abb. 5.2) wird mithilfe eines einfachen MATLAB Skriptes überprüft und anschließend eingebunden. Hierbei ist anzumerken, dass die Verfahrwege der Plattform als zwei Schubgelenke, je eines für x- und y-Richtung, dargestellt werden (Abb. 5.3).

α	a(mm)	θ	d(mm)
pi/2	0	0	582
0	0	0	0
pi/2	0	0	0
0	0	0	0
pi/2	0	0	0
pi/2	0	0	208
0	400	0	0
0	300	0	0
-pi/2	0	0	0
0	0	0	367.8

Abbildung 5.2: DH-Matrix

Abbildung 5.3: Robotics Toolbox Example
- Scitos G5

5.3 Implementierung der Bildverarbeitung

Ein überaus wichtiges Modul ist die Bildverarbeitung. Der Scitos G5 soll, wie zuvor der Katana mit einer Kamera mit Weitwinkelobjektiv in unmittelbarer Nähe zum End-Effektor ausgestattet werden. Zusätzlich verfügt die fahrbare Plattform über einen Laserscanner, der je nach Zeitaufwand ebenfalls in die Simulationsumgebung integriert werden soll.

Die Kamera wird dazu verwendet, die Umgebung wahrzunehmen und erfasst die Daten, welche über Algorithmen wie SIFT oder SURF robuste Merkmale bei bekannten Objekten wiedererkennen und so die relative Position des Roboters zum Objekt zu bestimmen. (Fries, 18. Aug 2010) Nur so kann der TCP so positioniert werden, dass der Roboter mit dem Objekt interagieren kann. In der Simulationsumgebung wird die Kamera als ein Viewpoint¹⁶ realisiert. Von diesem aus werden voraussichtlich 25 Fotos pro Sekunde geschossen, um für das *Auge* eine flüssige Bewegung zu erzeugen und eine *smooth* Bewegung des Roboters zu ermöglichen. Dabei können die intrinsischen Kamera-Parameter vernachlässigt werden, da diese ohnehin wieder korrigiert würden. Der View wird lediglich bezüglich des Bildausschnittes angepasst. Diese Daten werden von einem Algorithmus, wie er zur Zeit von Carsten Fries

¹⁶Betrachtungspunkt

implementiert wird, analysiert und entsprechend eine Bahn geplant, um sich wie gewünscht zum Objekt positionieren.

Da die Bildverarbeitung sehr rechenintensiv wird, entscheidet sich hier, welche Technik bei welchem Detaillierungsgrad eingesetzt wird. Zur Auswahl stehen VRML und OpenGL, wobei aus Aktualitäts- und Performancegründen voraussichtlich OpenGL genutzt wird. VRML wird jedoch trotzdem untersucht, da mit diesem Standard keine Berechnung und Darstellung außerhalb von MATLAB / Simulink stattfinden muss. Im Verlaufe von Projekt 2 wird sich entscheiden, zu wessen Gunsten das Aufwand / Nutzen Verhältnis zeigt. Auch für diesen Teil der Umsetzung besteht die Möglichkeit, eine bereits vorhandene Toolbox zu benutzen. Die sog. Image Processing Toolbox¹⁷ verfügt über zahlreiche Funktionen, welche eventuell einen Nutzen für die Arbeit bringen.

Der Grafikstandard der Umsetzung entscheidet auch darüber, ob eine eigene Implementierung in Java oder C++ erforderlich ist, oder ob Software von Dritt-Anbietern herangezogen wird.

5.4 Die Simulationsumgebung

Die Simulationsumgebung als Herz der Anwendung implementiert alle Schnittstellen zu den Modulen. Alle Daten werden hier zusammengeführt, so dass mithilfe der Subsumption Architektur einfach Funktionen aktiviert oder deaktiviert werden können. Hier wird ebenfalls die Darstellung der Simulation angesiedelt sein. Sie bietet neben dem Output Modul zur reinen Analyse von Bild- und Kinematikdaten eine Interaktionsmöglichkeit, so dass Bilddaten der Kamera als Input und die nächste Iteration der Zielpose des Roboters als Output übermittelt werden.

Optimal wäre an dieser Stelle die Erweiterung einer bestehenden oder die Konstruktion und Implementierung einer eigenen Toolbox. Damit würde die Integration in das bestehende Environment eine leicht zu bewältigende Aufgabe werden. Dies wird zur Zeit im Projekt 2 näher untersucht. Konkret geht es darum, mit einer Bibliothek wie beispielsweise OpenSG¹⁸ ein CAD Modell zu laden, den Grad der Tessellierung¹⁹ festzulegen und das Ergebnis zusammen mit dem Umgebungsmodell anderen MATLAB Funktionen zur Verfügung zu stellen; ganz nach dem Vorbild von Simulink und der 3D Animation Toolbox.

Sollte dies, wie angedacht, umsetzbar sein, sind bereits weitere Module denkbar. Die für die Sicherheit im realen Betrieb essenzielle Kollisionserkennung könnte zum einen durch Analyse der Sensordaten erfolgen und zum anderen könnte direkt innerhalb der 3D Darstellung selber ein Algorithmus aus der Starrkörpersimulation angewandt werden, um sich überlappende / durchdringende Geometrie erkennen zu können. Letztere Methode könnte zur Verifikation der Daten aus den Modulen genutzt werden.

¹⁷<http://www.mathworks.com/products/image/>

¹⁸<http://www.opensg.org>

¹⁹Technik, die sich mit der Zerlegung von Polygonen beschäftigt.

In der zentralen Komponente *Simulationsumgebung* wird zusätzlich zur Kommandozeile eine GUI²⁰ geschaffen, um dem Benutzer die Bedienung zu erleichtern. Die Interaktion über Eingabegeräte zwischen User und laufender Simulation ist jedoch nicht angedacht. Der wissenschaftliche Schwerpunkt dieser Arbeit wird auf der Subsumption Architektur und der Analyse der verschiedenen Algorithmen zur 3D-Darstellung liegen.

6 Risikobewertung

Da diese Arbeit mehrere, zum Teil sehr komplexe Themen vereint, bestehen bezüglich der Durchführung einige Risiken, die es im Vorhinein abzuschätzen gilt. Das Aufgabengebiet umfasst Teile der Robotik, 3D-Modellierung und Bildverarbeitung sowie Softwaredesign.

Das größte Risiko ist die Komplexität. Es besteht eine erhöhte Wahrscheinlichkeit, dass aufgrund der interdisziplinären Aufgabenstellung entweder nicht ausreichend in die Tiefe gegangen werden kann, um ein befriedigendes Ergebnis zu erzielen oder gar das Projekt als solches scheitert und nur eine Teilaufgabe behandelt werden kann. Es besteht ebenfalls die Möglichkeit, dass zwar die einzelnen Module separat korrekt funktionieren, jedoch das Gesamtsystem unvorhergesehene Fehler produziert. Eine umfassende Analyse ist somit unausweichlich.

Wie bei jeder Arbeit dieser Größe spielt der Faktor Zeit ebenfalls eine große Rolle. Hier wird das Ergebnis von Projekt 2 jedoch einen positiven Beitrag leisten, so dass es nur zu minimalen Engpässen kommen sollte.

Zwei weitere Faktoren, welche eher die Implementierung selbst betreffen sind der Detaillierungsgrad und die Performance: Ist die Anwendung zur Simulation von visuell geführten Robotern nah genug an der Realität, dass sich wirklich eine Produktivitätssteigerung in der Entwicklung an den Robotern der HAW herbeiführen lässt? Und ist die Anwendung leistungstark genug, dass, wenn schon nicht in Echt-Zeit, zumindest in annehmbarer Zeit simuliert werden kann?

All diese Faktoren bilden zusammen ein mittleres bis hohes Gesamtrisiko, dem schon zu Beginn der Arbeit mit äußerster Disziplin begegnet werden muss und auch wird.

7 Zusammenfassung

Die Masterarbeit zum Thema *Simulationsumgebung für einen visuell geführten Roboter* wird sowohl im Robotics-Laboratory der HAW als auch in Heimarbeit angefertigt. Sie hat es zum Ziel, eine Plattform für die Professoren, wissenschaftlichen Mitarbeiter und Studenten bereitzustellen, die das Testen von und mit den Robotern der HAW verbessert. Insbesondere der Scitos G5 Assistenzroboter, welcher im Rahmen des Living Place eingesetzt werden soll,

²⁰Graphical User Interface

wird in dieser Arbeit behandelt.

Die Grundlage bietet MATLAB mit seinen diversen Toolboxen sowie die Vorarbeiten aus den Veranstaltungen Anwendungen und Projekt 1 und 2. Die Arbeit setzt sich mit den Themen Robotik, Subsumption Architektur, 3D-Modellierung und Bildverarbeitung sowie Software-design auseinander. Es wird eine modulare Applikation entstehen, mit deren Hilfe nicht nur Roboter, sondern auch die Sensoren simuliert und analysiert werden können.

Roboter dringen mehr und mehr in den Bereich des Alltags ein und übernehmen immer mehr Aufgaben. Mit Blick in die Zukunft wird diese eine von mehreren Masterarbeiten der HAW, welche sich mit Robotern beschäftigen.

Literatur

- [Bruno Siciliano 2008] BRUNO SICILIANO, Oussama K.: *Handbook of Robotics*. 2008
- [Corke 2008] CORKE, Peter I.: *Robotics Toolbox for Matlab*. 2008. – URL <http://www.petercorke.com>
- [Fries 18. Aug 2010] FRIES, Carsten: Ausarbeitung - Anwendungen 2 / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-aw2/fries/bericht.pdf>, 18. Aug 2010. – Forschungsbericht
- [Shen und Hu 30. Juli 2010)] SHEN, Jiali ; HU, Huosheng: A Matlab-based Simulator for Autonomous Mobile Robots / University of Essex. URL <http://www.iproms.org/system/files/78.pdf>, 30. Juli 2010). – Forschungsbericht
- [Wirabhuana und bin Haron 2004] WIRABHUANA, Arya ; HARON, Habibollah bin: INDUSTRIAL ROBOT SIMULATION SOFTWARE DEVELOPMENT USING VIRTUAL REALITY MODELING APPROACH (VRML) AND MATLAB – SIMULINK TOOLBOX / University Teknologi Malaysia. URL http://saintek.uin-suka.ac.id/file_ilmiah/Paper%20RAPI%20UMS%20-%20Habib,%20Arya,.pdf, 2004. – Forschungsbericht