



Hochschule für Angewandte
Wissenschaften Hamburg
Hamburg University of Applied Sciences

AW1 Seminararbeit

Simulationsumgebung für
visuell geführte Roboter

von
Bernd Pohlmann

Bernd Pohlmann
Simulationsumgebung für
visuell geführte Roboter

AW1 Seminararbeit eingereicht im Rahmen des Anwendungen 1 Kurses
im Studiengang Master of Science
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Professor: Prof. Dr. Andreas Meisel

Eingereicht am 26.02.2010

Bernd Pohlmann

Titel der Seminararbeit

Simulationsumgebung für visuell geführte Roboter

Stichworte

Roboter, Kinematik, Denavit-Hartenberg, Matlab, Simulation

Kurzzusammenfassung

In dieser Ausarbeitung werden die Grundlagen erläutert, welche benötigt werden um die Bewegungen eines Roboters zu simulieren. Dabei wird insbesondere auf die Kinematik eingegangen. Es wird auf verschiedene Aspekte der Simulation im Bereich der Robotik hingewiesen sowie auf den Stand der aktuellen Entwicklung. Abschließend wird auf die zukünftigen Schritte bei der Entwicklung der Simulationsumgebung eingegangen.

Inhaltsverzeichnis

1	Einführung	1
1.1	Überblick	1
1.2	Problemstellung	1
1.3	Ziel	2
1.4	Motivation	2
2	Grundlagen	3
2.1	Robotik	3
2.1.1	Stand der Technik	3
2.1.2	Sicherheit	4
2.2	Kinematik	5
2.2.1	Koordinatensysteme	5
2.2.2	Position und Orientierung	5
2.2.3	Denavit-Hartenberg Transformation	6
3	Simulation	8
3.1	Simulationsumgebung	8
3.1.1	Mathworks MATLAB	9
3.1.2	Microsoft Robotics Developer Studio	9
3.1.3	SimRobot	9
3.2	Modell	10
4	Zusammenfassung & Ausblick	11
4.1	Zusammenfassung	11
4.2	Ausblick	11

Abbildungsverzeichnis

2.1	Moderne Roboter	4
2.2	Roboter im Sicherheitsbereich [Quelle: KUKA Roboter GmbH]	4
2.3	Die Denavit-Hartenberg Transformation [Quelle: wikipedia]	7
3.1	Roboter an der HAW[Quellen: Neuronics AG & MetraLabs GmbH]	8
3.2	Der Katana in der Simulation	10

1 Einführung

1.1 Überblick

Die Simulation hilft, Systeme zu analysieren, die für eine rein theoretische Betrachtung zu komplex sind. Die Experimente werden anstatt an einem realen Objekt an einem Modell durchgeführt. Hierbei ist das Modell auf die zu testenden Merkmale des Objektes reduziert. So können im Falle einer Computersimulation die Experimente an ein und dem selben Modell mit beliebigen Parametern durchgeführt werden, ohne dass, wie beispielsweise im Falle eines *realen* Objektes, äußere Einflüsse das Experiment beeinflussen. Nachdem die Simulation geeignete Ergebnisse hervorgebracht hat, müssen diese durch die Praxis verifiziert werden, denn die Simulation kann den Test mit einem Objekt nicht ersetzen.

Die Robotik ist ein interdisziplinäres Gebiet bestehend aus Informatik, Maschinenbau und Elektrotechnik. Sie befasst sich mit der Entwicklung und Steuerung von Robotern. Diese halten immer stärker Einzug in das tägliche Leben, ob in Form von Industrierobotern, Drohnen im zivilen und militärischen Bereich oder Assistenzroboter für den Haushalt. Eine große Herausforderung bei der Entwicklung stellt dabei die Interaktion mit dem Menschen dar.

Hierbei bietet auch die Simulation vielfältige Möglichkeiten, auf diese im Verlauf dieser Arbeit näher eingegangen wird.

1.2 Problemstellung

Ein beliebiger Roboter orientiert sich mithilfe einer Kamera und bewegt sich frei im Raum(Ebene). Seine Aufgabe ist es, Gegenstände zu erkennen, sich ihnen zu nähern und sie zu greifen.

Es gibt zur Zeit an der HAW keine Möglichkeit, die dazu notwendigen Algorithmen auf ihre Tauglichkeit zu testen, ohne sie direkt am Roboter auszuprobieren. Dies birgt die Gefahr, dass die Umwelt oder der Roboter selbst Schaden nimmt, wenn die Bewegungsabläufe nicht korrekt sind. Zudem ist es sehr aufwendig, den Roboter immer wieder neu zu programmieren und so iterativ das gewünschte Ergebnis zu erhalten. Ein weiterer Nachteil des direkten Tests ist, dass immer nur eine Person an dem Roboter arbeiten kann, was ebenfalls zeitliche Verzögerungen mit sich bringt.

Im Rahmen des iFlat¹ Projektes bekommt die HAW zu dem bisher existierenden einen weiteren, komplexeren Roboter. An diesem arbeiten voraussichtlich sechs Studenten parallel.

Es gilt nun, eine Lösung zu entwickeln, welche unabhängige Entwicklung und Tests möglich macht und dabei kostengünstig ist.

¹Versuchsgelände im Rahmen des *Living Place Hamburg*

1.3 Ziel

Ziel dieser Arbeit ist es, eine Einführung in die Komplexität der Robotik zu liefern und den Grundstein zu legen, in einer computergestützten Umgebung die Bewegungen eines visuell geführten Roboters zu simulieren. Im folgenden werden hierfür die Anforderungen festgelegt.

Die Simulationsumgebung

- muss auf jedem handelsüblichen Rechner lauffähig sein.
- muss mit Mathworks Matlab² umgesetzt werden.
- soll die Möglichkeit bieten, jeden beliebigen Roboter simulieren zu können.
- soll die Umgebung des Roboters inklusive Objekten zum Interagieren simulieren können.

Die vollständige Umsetzung findet im Rahmen des zweijährigen Masterstudiums statt.

1.4 Motivation

Roboter nehmen in unserer Gesellschaft immer komplexere Aufgaben wahr, vor allem der Bereich der Assistenzrobotik wächst stetig. Insbesondere auf diesem Gebiet, wo der Roboter mit Menschen agiert, muss sichergestellt sein, dass diese Co-Existenz ohne Zwischenfälle abläuft. Je mehr Aufgaben ein Assistenzroboter übernimmt, desto häufiger können Fehler passieren, welche im schlimmsten Falle, einen Menschen verletzen. Um nun das Risiko zu minimieren, müssen Verfahren eingesetzt werden, welche *realitätsnahe* Tests ermöglichen.

Ein weiterer Aspekt ist das Entwicklungstempo. Die Hersteller übertreffen sich mit immer neuen Systemen mit stetig wachsendem Potential. Um diese ausschöpfen zu können, ist es erforderlich, dass viele Entwickler mit unterschiedlichen Ideen und Ansätzen parallel in einer Umgebung einwickeln können, welche kostengünstig ist und sich jederzeit an neue Modelle und geforderte Funktionalitäten anpassen lässt.

Schließlich werden Roboter ihren Erbauern immer *ähnlicher*, sowohl in ihrem Verhalten als auch in ihrer Bauform. So nimmt ein Mensch den größten Teil seiner Orientierung über das Auge wahr. Wenn nun der Roboter ebenfalls diese Eigenschaft erhalten soll ohne dabei Sensoren zur Abstandsmessung o.ä. zu haben, muss diese visuelle Wahrnehmung sehr effizient sein.

Diese drei Aspekte machen die Entwicklung einer *Simulationsumgebung für visuell geführte Roboter* sehr attraktiv.

²wissenschaftliches Mathematikprogramm

2 Grundlagen

Dieses Kapitel befasst sich mit den Grundlagen, welche für das Verständnis der Arbeit notwendig sind. Um einen Roboter simulieren zu können, liegt der Fokus im Verständnis des Aufbaus sowie der Bewegungsabläufe und Positionierung/Orientierung.

2.1 Robotik

Der Begriff der Robotik wurde erstmals 1942 in einem Roman von Isaak Asimov¹ erwähnt und bedeutet nach seiner Definition das Studium der Roboter.[1]

Der Oberbegriff Roboter bezeichnet im Allgemeinen eine Maschine die durch gezielte Programmierung dazu in der Lage ist, ein oder mehrere definierte Aufgaben zu erfüllen. Hierbei unterscheidet sie sich von einfachen mechanischen Geräten dadurch, dass sie mithilfe von Elektronik und Sensorik *selbständig* Daten verarbeiten und auf Ereignisse reagieren kann.[2]

Das ursprüngliches Einsatzgebiet von Robotern liegt in der industriellen Fertigung. Hier können sie einfache, sich wiederholende Tätigkeiten mit höherer Frequenz und Präzision durchführen als der Mensch. Im Laufe der letzten Jahrzehnte hat sich ihr Aufgabengebiet jedoch deutlich erweitert. So geht die aktuelle Entwicklung immer mehr hin zu Service Robotern. Auf diesem Gebiet ist Japan die führende Nation und hat sich zum Ziel gesetzt, bis zum Jahre 2025, Roboter wie den von Honda entwickelten Asimo² im kommerziellen Markt zu etablieren.[3]

2.1.1 Stand der Technik

Heute sind Roboter viel mehr als nur Ersatz für menschliche Arbeitskraft. Die Bandbreite erstreckt sich von modernen Industrierobotern über Serviceroboter bis hin zu Medizintechnik Robotern und Exoskeletten.

Roboter wie der KUKA(Abb. 2.1a) übernehmen in der Automobilindustrie große Teile der Fertigung und werden auch in für Menschen unzumutbaren Umgebungen (z.B. Handhabung radioaktiver Stoffe) eingesetzt. Mobile Serviceroboter wie der Asimo(Abb. 2.1b) oder auch der in der HAW zum Einsatz kommende Scitos G5(Abb. 3.1b) unterstützen im Haushalt und können eine Vielzahl verschiedener Tätigkeiten ausführen; vom Darreichen von Gegenständen bis hin zum Absetzen von Notrufen.

Zwei weitere, noch recht junge Bereiche der Robotik bilden das Exoskelett³(Abb. 2.1c) und die Medizintechnik. Ein Beispiel für letztere ist ein von Shelten Yuen am Harvard Biorobotics Laboratory entwickeltes System, mit welchem sich am schlagenden Herzen Nähte befestigen lassen.[4]

¹russisch-amerikanischer Biochemiker und Schriftsteller(*1920 - +1992)

²<http://world.honda.com/ASIMO/>

³Stützstruktur für einen Organismus, die eine stabile äußere Hülle um diesen bildet

Roboter werden in vielen weiteren Bereichen, wie dem Militär, eingesetzt. Darauf wird in dieser Arbeit jedoch nicht näher eingegangen.



(a) KUKA KR210



(b) Honda Asimo



(c) Exoskelett

Abbildung 2.1: Moderne Roboter

2.1.2 Sicherheit

Beim Einsatz von Robotern im Umfeld des Menschen spielt die Sicherheit eine entscheidende Rolle. Egal, ob eine Roboterplattform stationär oder mobil ist und unabhängig davon wie groß die Leistung seiner Gliedmaßen/Gelenke ist, müssen Vorkehrungen getroffen werden, damit kein Schaden entsteht. So ist bei einem Industrieroboter sicherzustellen, dass sich bei eingeschaltetem Roboter kein Mensch in seinem Arbeitsbereich befindet, wenn dieser dadurch Gefahr läuft, durch Werkstücke oder Werkzeuge verletzt zu werden. Im Falle eines Assistenzroboters ist dies jedoch nicht durchführbar. Hier gilt es, den Roboter mit Sensoren auszustatten und so zu programmieren, dass er keine Schäden an seiner Umwelt herbeiführt. In diesem Zusammenhang sind die Robotergesetze von Asimov zu nennen. Auch wenn diese eher ethischer Natur sind, so wird doch im nicht-militärischen Bereich insbesondere auf die Einhaltung des ersten Gesetzes: "Ein Roboter darf keinen Menschen verletzen oder durch Untätigkeit zu Schaden kommen lassen." geachtet.

Die Sicherheit ist mit der ausschlaggebende Grund, warum eine Simulationsumgebung notwendig ist. Insbesondere im Studenumfeld der HAW, wo viele unerfahrene Personen an und mit dem Roboter arbeiten, bietet es sich an, die zu implementierenden Algorithmen für die Bewegungsabläufe erst in der Simulation zu testen und anschließend am realen Objekt zu verifizieren. Auch dem kostenintensiven Equipment der iFlat sowie dem Roboter selbst, kommt die Simulation zu gute.



Abbildung 2.2: Roboter im Sicherheitsbereich [Quelle: KUKA Roboter GmbH]

2.2 Kinematik

Kinematik beschreibt die Bewegung von Körpern in einem *Roboter-Mechanismus*. Dabei werden die Kräfte und Momente nicht berücksichtigt. Die Tatsache, dass Roboter von Beginn an auf Bewegung ausgelegt sind, macht Kinematik zum fundamentalen Aspekt vom Roboter Design über die Analyse und Kontrolle bis hin zur Simulation. Der besondere Fokus liegt dabei auf der Repräsentation von Position und Orientierung.

In der Kinematik wird von einer Reihe idealer Bedingungen ausgegangen:

- Die einzelnen Festkörper sind geometrisch perfekt in Position und Form.
- Die einzelnen Elemente haben über ihre Verbindungen idealen Kontakt und den Abstand '0'.
- Die einzelnen Körper sind in ihrer Bewegungsfreiheit uneingeschränkt.[5]

Um Position und Orientierung des Roboters bzw. seiner Armglieder exakt bestimmen zu können und den Roboter korrekt steuern zu können, muss zunächst ein Bezugssystem gewählt werden.

2.2.1 Koordinatensysteme

Das Bezugssystem, in welchem sich der Roboter bewegt, ist sein Welt-Koordinatensystem. Dieses ist im Falle eines stationären Roboters sein Arbeitsbereich, also die Umgebung, die er mit seinem End-Effektor bei maximaler Auslenkung aller Armglieder erreichen kann. Bei einem mobilen Roboter ist der Arbeitsbereich theoretisch unbegrenzt, jedoch praktisch durch seine Bewegungs-Reichweite beschränkt. Zusätzlich wird in jedes Gelenk ein Koordinatensystem gelegt, welches durch die Maße bzw. die Auslenkung des jeweiligen Gelenkes begrenzt wird. Am Ende der sog. kinematischen Kette aus Armgliedern und Gelenken befindet sich der End-Effektor. In der Regel ist dieser ein Werkzeug (z.B. ein Greifer), welches mit Werkstücken interagiert. Dieser hat ebenfalls ein Koordinatensystem, dessen Zentrum der Tool-Center-Point (TCP) bildet. Je nach Betrachtung kann man den Roboter auch relativ zum Werkstück-Koordinatensystem betrachten. Dies ist insbesondere dann relevant, wenn sich das Werkstück ebenfalls bewegt.[6]

Für diese Arbeit ist noch ein weiteres Koordinatensystem von Bedeutung. Da es sich um einen visuell geführten Roboter handelt, ist dies das Kamera-Koordinatensystem. Die Kamera ist in unmittelbarer Nähe des End-Effektors angebracht und *sieht* das Werkstück aus ihrem Center-Point.

All diese Koordinatensysteme (es handelt sich um kartesische Koordinaten) sind von Bedeutung, um die Position und Orientierung aller beteiligten Elemente bestimmen und letztendlich verändern zu können.

2.2.2 Position und Orientierung

Im Allgemeinen ist ein Roboter ein System aus Festkörpern mit Verbindungspunkten. Die Position und Orientierung der Gesamtheit der Elemente des Roboters bezeichnet man als *Pose*. Innerhalb seines Arbeitsbereiches kann ein Roboter je nach Konstruktion bestimmte Posen annehmen. Welche dies sind, wird durch die Freiheitsgrade (DOF⁴) bestimmt.

⁴Degree of freedom

Ein freier starrer Körper hat sechs Freiheitsgrade, drei translatorische und drei rotatorische. Für einen Roboter heißt das: Damit das Werkzeug in jeder Position jede beliebige Orientierung annehmen kann, muss der Roboter sechs Freiheitsgrade besitzen. Durch sinnvolle mechanische Konstruktion besitzt ein Roboter mit 6 Achsen auch 6 Freiheitsgrade.[7]Ein mobiler Assistenzroboter wie der Scitos G5 besitzt zusätzlich noch zwei weitere translatorische DOF, da er sich in der Ebene frei bewegen kann. Er ist dadurch kinematisch redundant. Dies bedeutet, dass es in diesem Fall unendlich viele Möglichkeiten gibt, mit dem End-Effektor einen Punkt im Arbeitsraum zu erreichen.

Um nun die Pose des Roboters zu bestimmen, gibt es zwei grundsätzliche Ansätze. Zum einen die direkte Kinematik: Die Winkel aller Gelenke werden angegeben und die daraus resultierende Pose errechnet. Im Gegensatz dazu geht die indirekte Kinematik wie folgt vor: Die zu erreichende End-Effektor Pose wird vorgegeben und die Gelenkwinkel werden entsprechend berechnet. Letzteres Verfahren ist mathematisch herausfordernder und im Falle kinematischer Unbestimmtheit (Redundanz) teilweise nur mit iterativen Methoden durchführbar. Näheres dazu finden Sie unter anderem in der Arbeit von C. Schmiedecke⁵. Im folgenden wird ein gängiges Verfahren für die direkte Kinematik vorgestellt.

2.2.3 Denavit-Hartenberg Transformation

Die Denavit-Hartenberg Transformation ist ein mathematisches Verfahren, mit dessen Hilfe die Koordinatensysteme innerhalb einer kinematischen Kette in einander überführt werden können. Bei einem z.B. 6-gelenkigen Manipulator wie dem KUKA beschreibt je eine homogene Matrix jedes der Gelenke in Bezug auf das vorhergehende Gelenk. Dabei beschreibt die Matrix A_1 die Position und Orientierung des ersten Gliedes (Fuß), A_2 die Position und Orientierung des zweiten Gliedes bezüglich Glied 1 usw. Die Kette zieht sich so bis zum sechsten Glied, dem End-Effektor, fort, so dass als Produkt

$$T = A_1 A_2 A_3 A_4 A_5 A_6$$

entsteht. Die Koordinatensysteme liegen fest in den Bewegungsachsen und müssen dabei nach der Denavit-Hartenberg-Konvention festgelegt werden:

- Die z_i -Achse wird entlang der Bewegungsachse des $(i+1)$ -ten Gelenks gelegt.
- Die x_i -Achse ist senkrecht zur z_{i-1} -Achse und zeigt von ihr weg.
- Die y_i -Achse wird so festgelegt, daß sich ein rechtshändiges Koordinatensystem ergibt.

Um nun die Beziehung zwischen den Koordinatensystemen herzustellen, müssen mehrere Translationen und Rotationen durchgeführt werden. Mit folgenden Schritten wird das Koordinatensystem i in $i-1$ überführt:

- Drehe um z_{i-1} mit dem Winkel θ_i , damit die x_{i-1} -Achse \parallel zu der x_i -Achse liegt
- Verschiebe entlang z_{i-1} um d_i bis zu dem Punkt, wo sich z_{i-1} und x_i schneiden
- Verschiebe entlang gedrehtem $x_{i-1} = x_i$ um eine Länge a_i , um die Ursprünge der Koordinatensysteme in Deckung zu bringen
- Drehe um x_i mit dem Winkel α_i , um die z_{i-1} -Achse in die z_i -Achse zu überführen

⁵ *Entwicklung einer dynamischen allgemeinen Kinematik*

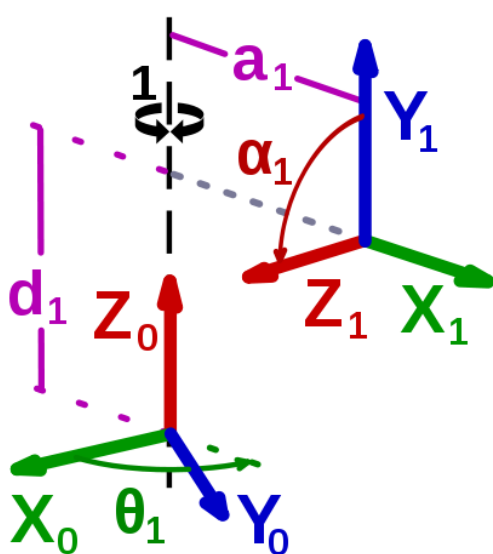
Fasst man alle vier homogenen Transformationen, so kann die Matrix A_i wie folgt berechnet werden:

$$A_i = Rot_{z_{i-1}, \theta_i} Trans_{(0,0,d_i)} Trans_{(a_i,0,0)} Rot_{x_i, \alpha_i}$$

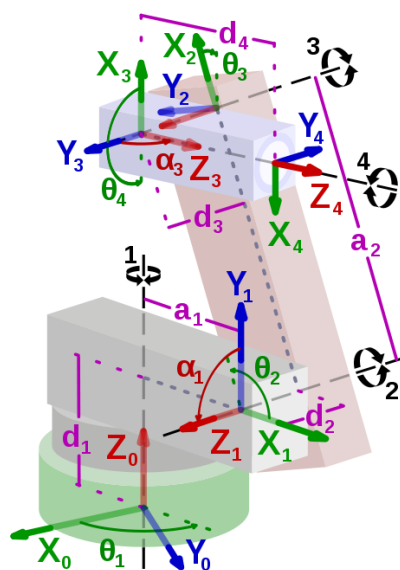
Die Gesamttransformation zwischen zwei Gelenken ist also eine einfache Matrizenmultiplikation mit:

$$\begin{pmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Um nun von TCP- in Welt-Koordinaten umzurechnen, wird die Matrizenmultiplikation $i-1$ mal ausgeführt.[8] Abb. 2.3 veranschaulicht dies.



(a) DH - Transformation(einfach)



(b) DH - Transformation(5-Achser)

Abbildung 2.3: Die Denavit-Hartenberg Transformation [Quelle: wikipedia]

3 Simulation

Dieses Kapitel handelt vom Erstellen eines Simulationsmodelles für einen Roboter. Es werden die Schritte abgehandelt, welche notwendig sind, um das Ziel einer Simulationsumgebung für einen Plattform unabhängigen, visuell geführten Roboter zu realisieren. Dabei wird zunächst auf die Roboter-Simulation im Allgemeinen eingegangen und im weiteren Verlauf explizit auf die an der HAW eingesetzten Methoden und Verfahren. Die Roboter, welche an der HAW zur Verfügung stehen sind der Katana (Abb. 3.1a), ein stationärer Knick-Arm-Roboter und zukünftig das mobile Gegenstück, der Scitos G5 (Abb. 3.1b). Beide verfügen über eine Kamera auf ihrem End-Effektor (nicht in den Abbildungen enthalten). Da die meisten Erfahrungen bisher mit dem Katana gemacht wurden, wird sich das folgende Kapitel auf die Simulation desselben konzentrieren.



(a) Katana



(b) Scitos G5

Abbildung 3.1: Roboter an der HAW [Quellen: Neuronics AG & MetraLabs GmbH]

3.1 Simulationsumgebung

Obwohl zu Beginn des Projektes bereits Mathworks MATLAB als Simulationsumgebung feststand, so ist es doch lohnenswert, zwei alternative Tools vorzustellen. Zuvor jedoch muss festgelegt werden, was genau simuliert werden soll und was aus Gründen der Komplexität bzw. der geringen Zeit entfallen kann. Im Verlauf des Master-Studienganges sollen folgende Features implementiert werden:

- Eine Abstraktion jedes beliebigen Roboters muss möglich sein. Bei mobilen Robotern muss der Bewegungsraum eingegrenzt werden.
- Das Modell des Roboters und seine Umgebung müssen voll dreidimensional dargestellt werden, um eine optimale Analyse zu ermöglichen.

- Da das Projekt zu groß ist, um alle Aspekte von einem einzelnen Studenten abarbeiten zu lassen, müssen verwandte Arbeiten einfach implementiert werden können, sei es eine inverse Kinematik oder die bildverarbeitende Komponente der Kamera.
- Es müssen einfache Objekte (Würfel, Zylinder, etc.) darstellbar sein.

Zum jetzigen Zeitpunkt ist keine Physik-Engine vorgesehen.

Da der Robotertyp austauschbar sein soll, entfallen alle Hersteller-spezifischen Simulationstools. Dies führt uns direkt zur Vorstellung theoretisch möglicher Applikationen.

3.1.1 Mathworks MATLAB

MATLAB ist eine hoch entwickelte Sprache für technische Berechnungen und eine interaktive Umgebung für die Algorithmenentwicklung, die Visualisierung und Analyse von Daten sowie für numerische Berechnungen. Es eignet sich für eine Vielzahl von Anwendungsgebieten wie die Signal- und Bildverarbeitung, Tests und Messungen sowie mithilfe ergänzender Toolboxen wie beispielsweise Simulink auch zur Simulation komplexer Zusammenhänge.[9] Es ist auch möglich, selber solche Erweiterungen zu schreiben, wie dies Peter Corke¹ mit der MATLAB Robotics Toolbox getan hat. Dieses sehr umfangreiche Addon liefert einen großen Teil der Simulationsumgebung, welche für dieses Projekt benötigt wird.

3.1.2 Microsoft Robotics Developer Studio

Microsoft Robotics Developer Studio(RDS), aktuell in der Version 2008 R2 stellt eine große Bandbreite an Unterstützung zu Verfügung, um Roboter oder Anwendungen für Roboter zu entwickeln. Microsoft RDS beinhaltet ein Programmiermodell, welches die Entwicklung asynchroner Automaten möglich macht. Die Software ist darauf ausgelegt, ein breites Spektrum verschiedener Roboter zu unterstützen.

Im Paket enthalten ist ebenso ein Set von graphischen Entwicklungswerkzeugen und diverse Simulationstools. Im *Visual Simulation Environment*, kurz VSE, können Robotik-Anwendungen in einer 3D Physik-Engine simuliert werden.[10]

3.1.3 SimRobot

SimRobot ist eine frei verfügbare Eigenentwicklung der UNI Bremen. Es ist eine Anwendung zur Simulation sensorbestückter Agenten in einer dreidimensionalen Umwelt. Sie kommt vollkommen ohne Robotik Hardware aus und soll für den Einsatz von Algorithmen in realen Robotersystemen vorbereiten. Die Simulation erlaubt die Definition von hierarchischen Objekten mit Körpern, Dreh und Teleskopgelenken, Fahr und Flugzeugen, Abstands, Farb und Lichtsensoren sowie zweidimensionalen Kameras und Facettenaugen. Damit wird auch, mit entsprechender Abstraktion, die Modellierung des Verhaltens natürlicher Systeme möglich. Durch den offenen Quellcode in der Programmiersprache C ist die Applikation leicht erweiterbar.[11]

¹<http://petercorke.com/Home.html>

3.2 Modell

Im folgenden wird ein bereits vorhandenes, einfaches Simulationsmodell mit der MATLAB Robotics Toolbox dargestellt. Es handelt sich dabei um eine Abstraktion des Katana Roboters.

Zunächst werden anhand der vom Hersteller angegebenen Abmessungen des Manipulators die Denavit-Hartenberg-Parameter angegeben. Daraus erstellt die Toolbox ein Modell, welches in einem zuvor definierten 3-dimensionalen Arbeitsraum zentral platziert wird. Anschließend können die Gelenke durch einfache Angabe von Winkeln in die gewünschte Position gebracht werden.

Herr Prof. Meisel hat eine inverse Kinematik für den Katana erstellt, was es ermöglicht, den modellierten Roboter ein Bewegungsmuster abarbeiten zu lassen. Folgende Abbildung zeigt einen Screenshot aus der Anwendung, welche den Roboter (mit bereits korrekten Denavit-Hartenberg Parametern) in seinem Arbeitsraum zeigt.

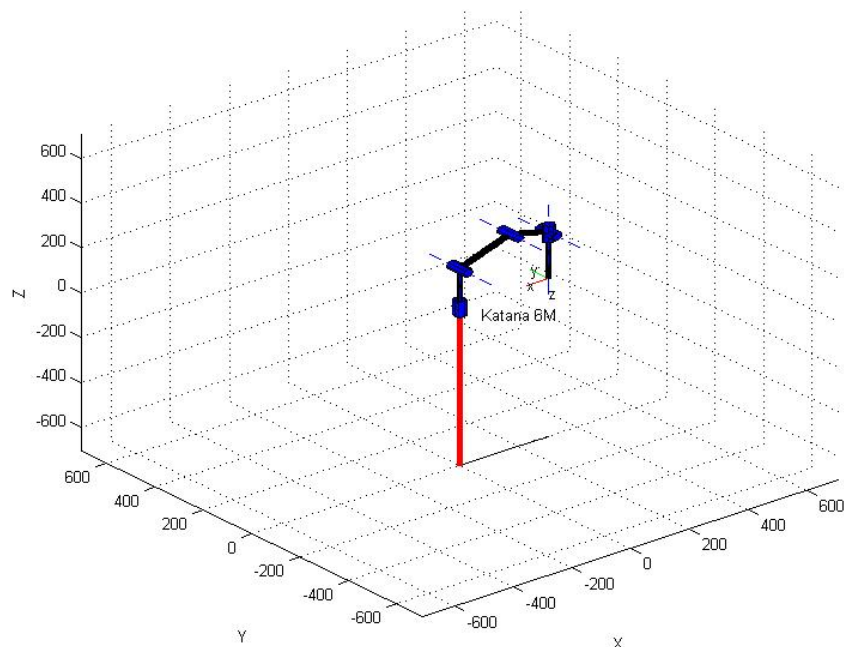


Abbildung 3.2: Der Katana in der Simulation

Zum jetzigen Zeitpunkt existieren noch keine Objekte in diesem. Die Kamera, Bildpunkte für diese etc. sind ebenfalls noch nicht vorhanden. Diese werden im weiteren Verlauf der Arbeit hinzugefügt sowie eine einfaches Modell in MATLAB implementiert, welches den Austausch von Modell und Algorithmen ermöglicht.

4 Zusammenfassung & Ausblick

4.1 Zusammenfassung

Die Grundlagen, welche in dieser Ausarbeitung erarbeitet wurden, bilden die Basis für die Umsetzung der Simulationsumgebung für visuell geführte Roboter. Dabei wurde bewusst nicht konkret auf das Kameramodell eingegangen, das dies in der Seminararbeit¹ *3D-Objekterkennung im Kontext eines Assistenzroboters* von Benjamin Wagner in vollem Umfang vorhanden ist. Die aus Herrn Wagners Arbeit hervorgehenden Ergebnisse werden mit hoher Wahrscheinlichkeit in die Simulationsumgebung eingebettet.

Wichtig ist jedoch, dass mit der Einführung in das Thema Kinematik der Grundstein für das Verständnis der Bewegungen sowie der Pose von Robotern gelegt wurde. Nun kann die MATLAB Robotics Toolbox effektiv genutzt und bei Bedarf angepasst werden. Die anderen, bereits am Markt verfügbaren Produkte, liefern weitere wichtige Anhaltspunkte, worauf bei der Umsetzung zu achten ist, bzw. welche Ergänzungen/Neuerungen notwendig werden könnten.

Der Umsetzung des Projektes stehen keine unlösbaren oder kostenintensiven Probleme im Wege und somit können nun weitere Schritte folgen.

4.2 Ausblick

Nach weiterer Einarbeitung in die Robotics Toolbox wird das bestehende Modell zunächst modularisiert. Anschließend werden folgende Schritte möglich:

- Einbringen von Objekten in den Arbeitsraum
- Austausch des Modells durch eine einfache mobile Plattform und später ein Modell des Scitos G5
- Modellierung eines Raumes mit diversen Objekten und Hindernissen
- Implementierung der Ergebnisse verwandter Arbeiten aus den Bereichen inverse Kinematik, Bahnplanung und Objekterkennung

Es ist auch denkbar, Teile anderer Simulationsumgebungen zu verwenden. Da MATLAB in Grenzen C Programmcode ausführen kann, wird in Betracht gezogen, Teile von SimRobot zu adaptieren. Dies wird im weiteren Verlauf der Arbeit genau geprüft. Sollte es zeitlich möglich sein, wird ebenfalls in Erwägung gezogen, eine Physik-Engine zu implementieren. Da die Möglichkeiten, welche sich in diesem Projekt bieten, nahezu unbegrenzt sind, bietet sich somit auch die Chance, daran mit zu entwickeln.

¹<http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master08-09-aw1/wagner/bericht.pdf>

Literaturverzeichnis

- [1] Encyclopedia Britannica Online *Robot* 20.02.2010
- [2] Craig, J.J., *Introduction to Robotics* Pearson Prentice Hall, 2005
- [3] Yumiko Myoken, <http://ukinjapan.fco.gov.uk/resources/en/pdf/5606907/5633632/next-generation-services-robots.pdf>,
Research and Development for Next Generation Service Robots in Japan, British Embassy, January 2009
- [4] v. Szentpétery, *Technology Review* Heise Zeitschriften Verlag, Ausgabe 12/08
- [5] Siciliano, Khatib, *Handbook of Robotics* Springer Verlag, 2008
- [6] Prof. Dr. Klaus Wüst, *Robotik - Vorlesung* an der FH Gießen-Friedberg, SS 2005
- [7] Prof. Dr. Christof Röhrig, *Kinematik stationärer Roboter - Vorlesung* an der FH Fortmund, WS 2008/2009
- [8] Prof. Dr. Jianwei Zhang, *Einführung in die Robotik - Vorlesung* an der UNI Hamburg April 2009
- [9] Mathworks Homepage, <http://www.mathworks.de/products/matlab/description1.html> 20.02.2010
- [10] Microsoft Homepage, <http://www.microsoft.com/robotics/#Product> 20.02.2010
- [11] Homepage der UNI Bremen,
http://www.informatik.uni-bremen.de/simrobot/index_d.htm 20.02.2010