



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

AW2 Ausarbeitung

Bernd Pohlmann

Robot Simulation Software: Simplicity & Precision

Bernd Pohlmann

Thema der AW2 Ausarbeitung

Robot Simulation Software: Simplicity & Precision

Stichworte

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

Kurzzusammenfassung

Diese Arbeit beschäftigt sich mit unterschiedlichen Möglichkeiten, Roboter am Computer zu simulieren. Nach einer kurzen Einleitung, in der beschrieben wird, wozu überhaupt die Simulation benötigt wird, wird auf Roboter Simulationssoftware im allgemeinen eingegangen. Anschließend werden drei Projekte aus verschiedenen Bereichen vorgestellt sowie ausgewertet. Im vorletzten Schritt werden daraus Ergebnisse abgeleitet sowie auf eigens umgesetzte Ansätze eingegangen. Abschließend wird die Ausarbeitung zusammengefasst und ein Ausblick auf das eigene Projekt gegeben.

Title of the paper

Robot Simulation Software: Simplicity & Precision

Keywords

robot, simulation, trajektorie, Matlab, VRML, kinematics, 3D-model

Abstract

This paper is about the various possibilities to simulate robots on a computer. After a short disclaimer that introduces the reasons for simulating robots, robot simulation software is described in general. In the next step three existing projects from different scopes are represented and analysed. In the following section some conclusions are made and the author's own implementations are shown. Finally this paper is summarized and a preview of further development is given to the reader.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Die Aufgabe	1
2	Einführung in das Thema	2
2.1	Robot Simulation Software	2
2.1.1	Physik	2
2.1.2	Sensorik und Kinematik	3
2.2	Darstellung im Modell	3
3	RSS Projekte	3
3.1	Industrieroboter Simulation mit MATLAB und VRML	4
3.2	Virtual Robot Controller Interface	5
3.3	MATLAB Simulation autonomer mobiler Roboter	6
4	Ergebnisse	8
4.1	Bewertung der Projekte	8
4.2	Einordnung der eigenen Arbeit	9
5	Zusammenfassung und Ausblick	10

Abbildungsverzeichnis

1	Der Mitsubishi RV-2AJ	4
2	Die Module des VRC [Quelle: Fraunhofer IPK]	6
3	Die Sicht des Simulators	8

1 Einleitung

Von der ersten Erwähnung des Wortes *Roboter* (1942) durch Isaac Asimov¹ bis zu heutigen modernen Robotern hat die Fiktion ihren Weg in die Realität gefunden. Roboter sind im 21. Jahrhundert nicht mehr wegzudenken; sie verrichten ihre Arbeit in der Industrie, im Haushalt und in der Medizin und haben vielerorts den Menschen durch ihre hohe Präzision und Geschwindigkeit bereits abgelöst.

Der Schwerpunkt dieser Arbeit liegt dabei auf der Behandlung der Industrieroboter, da sie die am häufigsten eingesetzte Robotergattung sind. Diese sind angeführt durch die Automobilindustrie in einer Evolutionsstufe angekommen, in der die Wirtschaftlichkeit ihres Einsatzes hauptsächlich dadurch beschrieben ist, wie schnell sich ein Roboter (neu) programmieren lässt und wie vielfältig seine Bewegungsabläufe sind. Ersteres soll an folgendem Beispiel verdeutlicht werden:

Im russischen VW Werk in Kaluga können bis zu 150.000 Fahrzeuge pro Jahr gebaut werden. Das entspricht circa 17 Autos/Stunde bei 24 Stunden Arbeit an 365 Tagen im Jahr. Wenn nun zur Programmierung eines Roboters 30 Minuten benötigt werden, werden acht Fahrzeuge nicht gebaut. Dies entspricht bei einem durchschnittlich ausgestatteten Golf VI eine Umsatzeinbuße von $8 \times 20.000 \text{ €} = 160.000 \text{ €}$. Bei Falsch-Programmierung kann, je nachdem wann der Fehler erkannt wird, ein ungleich höherer Schaden entstehen. (DPA, 28. Oktober 2006)

Anfang der neunziger Jahre, als ganze Fertigungsstraßen automatisiert wurden, wurde die (Roboter-) Simulation zu einem eigenen Entwicklungszweig. Musste zu Beginn der Roboter noch von Hand angelernt / programmiert werden, so wird mittlerweile die Trajektorie vollständig am Computer simuliert und erst nach erfolgreich abgeschlossenen Tests, das Programm binnen Sekunden auf den Roboter übertragen. Dies bringt kürzere Ausfallzeiten und somit höhere Produktivität.

1.1 Die Aufgabe

Diese Ausarbeitung beschäftigt sich mit der Simulation von Robotern in Forschung und Praxis. Im Rahmen dieser Ausarbeitung geht es darum, eine Simulationsumgebung für einen autonomen, mobilen, visuell geführten Roboter zu schaffen.

Dabei sind folgende Anforderungen vorab festgelegt worden:

- Verwendung von MATLAB zur Realisierung der Umgebung
- Maximale Modularität - Jede Komponente muss in wenigen Schritten austauschbar sein.
- Einfache Anwendbarkeit trotz komplexer mathematischer Zusammenhänge

Um die Aufgabe einzugrenzen, wird die Simulation die physikalischen Gegebenheiten (Kräfte, Momente) nicht enthalten. Um nun die Bedürfnisse erfüllen zu können, werden im Folgenden die Möglichkeiten beschrieben, sowie drei zum Teil sehr unterschiedliche Ansätze vorgestellt und bewertet.

¹(*1920 - †1992) russisch-amerikanischer Biochemiker, Sachbuchautor & Science-Fiction-Schriftsteller.

2 Einführung in das Thema

Die Simulation eines Roboters oder genauer, der Bewegung eines Roboters, liegen einfache Matrix-Operationen zugrunde. Dies gilt ebenso für das Modell einer Lochkamera, welche der Roboter zur Wahrnehmung nutzt. Die Herausforderung besteht bei einem solchen Projekt darin, all jene Faktoren zu berücksichtigen, die eine flüssige Bewegung ausmachen. Dabei spielen die Genauigkeit des Kamerabildes, die Freiheitsgrade des Roboters, sowie die Umgebung, in welcher der Roboter arbeitet eine Rolle. Sind beispielsweise Merkmale eines zu manipulierenden Objektes durch ein anderes verdeckt, muss eine Strategie vorhanden sein, den Roboter so zu bewegen, dass das Objekt vollständig sichtbar wird. Ein anderer Faktor, oder auch ein anderes Modul, stellt die Bahnplanung dar. Je nach Anzahl der Freiheitsgrade, gibt es keine eindeutige Kinematik und somit keine eindeutige Trajektorie.

All diese, und weitere, Module bilden das Backend für eine Simulation. Der Anwender will jedoch das Ergebnis der Simulation sehen, woraus sich zusätzliche Faktoren zur Darstellung des Roboters und der Umgebung ergeben.

2.1 Robot Simulation Software

Mit der zunehmenden Automatisierung der Fertigung in der (Automobil-) Industrie mussten Wege gefunden werden, Prozesse und Abläufe immer schneller an neue Gegebenheiten anzupassen. Für den Einsatz von Robotern bedeutete dies den Einsatz von Simulationsmechanismen. Hier bot sich die ebenfalls stark wachsende IT an, um am Rechner Arbeitsabläufe für Fertigungsstraßen zu planen.

Nach zunächst rein mathematischen Ansätzen, die ausschließlich die Machbarkeit und Korrektheit der eingegebenen Befehle überprüften, kamen nach und nach immer mehr visualisierende Anwendungen auf den Markt. Diese ermöglichen es, dem Anwender den detaillierten Bewegungsablauf zu sehen und somit früher Fehler zu erkennen.

Heute kann man Roboter Simulation in zwei große Teilbereiche aufgliedern: 1. Die Simulation von physikalischen Größen (Kräfte & Momente) und 2. Sensoren-Erfassung und Kinematik.

2.1.1 Physik

Die Simulation von physikalischen Größen ist Hauptbestandteil während der Design- und Testphase eines neuen Roboters. Sie dient dazu, maximale Lasten zu bestimmen sowie die Präzision unter Einwirkung verschieden dimensionierter Kräfte und Momente zu ermitteln. Ein Greifarmroboter wie der Katana kann für 500g maximale Last zugelassen sein, jedoch unter Einbußen der Genauigkeit deutlich mehr bewegen. Insbesondere bei diesem Typ Roboter spielt die Schwerpunktverlagerung beim Ausstrecken des Armes sowie seiner Rotation eine nicht unerhebliche Rolle. Dies kann am physikalischen Modell mit Anwendungen wie MATLAB Simulink simuliert werden.

Diese Arbeit wird sich jedoch nicht mit dem Bau eines neuen Roboters, sondern mit der Simulation eines bereits vorhandenen, voll spezifizierten Manipulators befassen. Daher liegt der Hauptaugenmerk auf der Sensorik und Kinematik.

2.1.2 Sensorik und Kinematik

Die Simulation der Sensorik, also wie der Manipulator seine Umgebung wahrnimmt, ist für autonome, mobile Roboter eine Kerndisziplin. Ohne Bestimmung der eigenen Lage zu dem zu manipulierenden Objekt sowie der Lage des Objektes selber, ist eine Trajektorie völlig nutzlos. Auch die Freiheitsgrade können nur bedingt etwas über die Durchführbarkeit einer Aktion aussagen. Maßgebend sind die Daten von Sensoren zur Entfernungsbestimmung. Diese sind unterteilt in

- optische, wie Kameras und Laserscanner sowie
- akustische, wie Ultraschall oder Radar.

Besonders interessant ist dabei die Kamera. Es ist beispielsweise möglich, sowohl den Roboter in seiner Umgebung visuell (aus einer Kameraperspektive) darzustellen als auch die Projektionsebene der Kamera zu modellieren. So kann aus dem Kamerabild mithilfe der Kinematik (z.B. über die Denavit-Hartenberg Parameter) die Position des Roboters sowie die Lage eines Objektes im Raum ermittelt werden, so dass die Anwendung eine Strategie errechnet, wie der Endeffektor in Reichweite des Objektes bewegt werden kann.

Durch seine hohe Genauigkeit eignet sich der Laserscanner ideal zur Hinderniserkennung und somit zur Kollisionsvermeidung. Ultraschall spielt aufgrund der fehlenden Präzision nur noch eine untergeordnete Rolle.

2.2 Darstellung im Modell

Neben den oft von den Roboter Herstellern bereitgestellten Simulationswerkzeugen, existieren zahlreiche Anwendungen wie Mathworks MATLAB (unterstützt durch diverse Add-ons wie Simulink oder die Epipolarline Toolbox) oder Microsoft Robotics Developer Studio, mit denen Modelle veranschaulicht werden können. Modelica als eigene objektorientierte Sprache zur physikalischen Modellierung und 3D Modellierungswerkzeuge wie Blender (OpenSource) kommen ebenfalls zum Einsatz. Auch die finite Element Methode kommt bei Belastungstests zum Einsatz.

Durch dieses breit gefächerte Spektrum ergeben sich unzählige Einstiegspunkte für die Auseinandersetzung mit dem Thema Roboter Simulation.

3 RSS Projekte

Im Rahmen der *Anwendungen 2* Vorlesung geht es darum, mit Blick auf die eigene Masterarbeit, verwandte Projekte zu analysieren und Parallelen sowie Abgrenzungen aufzuzeigen. In diesem Zusammenhang werden im Folgenden drei unabhängig voneinander entstandene Projekte vorgestellt. An dieser Stelle wird auf die Tatsache hingewiesen, dass der ferne Osten auf dem Gebiet der Robotik führend ist. Auch vor diesem Hintergrund sind zwei der vorgestellten Projekte betrachtet worden.

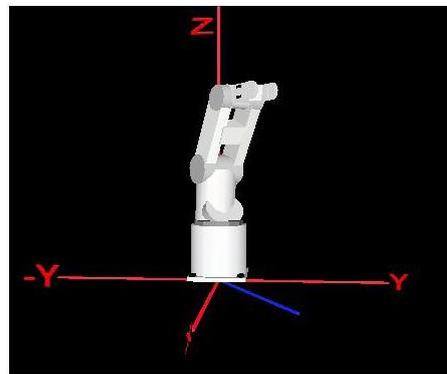
3.1 Industrieroboter Simulation mit MATLAB und VRML

Das Projekt "INDUSTRIAL ROBOT SIMULATION SOFTWARE DEVELOPMENT USING VIRTUAL REALITY MODELING APPROACH (VRML) AND MATLAB – SIMULINK TOOLBOX" beschäftigt sich mit der 3D-Simulation von Robotern am Beispiel eines Mitsubishi RV 2AJ Roboter Armes (Abb. 1a). Dabei handelt es sich um einen fest stehenden Manipulator mit sechs Gelenken und ebenso vielen Freiheitsgraden.

Ziel war es, den Roboter abstrahiert zu modellieren und darzustellen, sowie durch Eingabe der Gelenkwinkel bzw. der Position und Orientierung des Endeffektors das 3D-Modell in eine gewünschte Pose zu bringen. Dies wurde in drei Arbeitsschritten verwirklicht: 1. Spezifizierung der Kinematik, 2. Modellierung des Knick-Arm Roboters mit VRML, 3. Entwicklung einer GUI.



(a) in der Realität



(b) in der Simulation

Abbildung 1: Der Mitsubishi RV-2AJ

Zu 1: Die geometrische Konfiguration wurde vom Hersteller übernommen und die Gelenk Parameter wurden in einer Denavit-Hartenberg Matrix zusammengefasst. Somit ist es möglich, die Pose des Endeffektors bei Angabe der einzelnen Gelenkstellungen zu ermitteln. Anschließend wurde eine einfache inverse Kinematik entwickelt, um auch bei Eingabe der Pose im TCP² die einzelnen Gelenkstellungen zu bestimmen.

Zu 2: Um den Arm in 3D modellieren zu können, wurde er zerlegt und vermessen. VRML³ war das gewählte Mittel zur Umsetzung; zum einen, da es einen Standard darstellt und zum anderen da es eine Simulink Schnittstelle dafür gibt. Im Folgenden wird kurz auf die wesentlichen Merkmale eingegangen.

VRML ist eine Beschreibungssprache für 3D-Szenen, deren Geometrien, Ausleuchtungen, Animationen und Interaktionsmöglichkeiten. Ursprünglich als 3D Standard für das Internet entwickelt, ist sie *human readable*⁴ und wird in vielen 3D-Modellierungswerkzeugen verwendet. Ein VRML Modell besteht aus einer Basis (Group) und einer Anzahl von sog. Transforms und Shapes. Die Basis bildet den Ursprung des Modells, jedes Transform Element hat n

²Tool Center Point (Werkzeugmittelpunkt)

³Virtual Reality Modelling Language

⁴für den Menschen lesbar

Freiheitsgrade und ist mit einer Struktur (Shape) überzogen. Mehrere miteinander Verbundene Transforms ergeben zusammen mit den Shapes eine Baumstruktur, den Scene-Graphen, welcher das 3D-Objekt bottom-up darstellt. Diese Tatsache macht das Erstellen einer direkten Kinematik sehr einfach.

Zu 3: Das User Interface besteht aus zwei Teilen. Im ersten Teil, einem Browserfenster, befindet sich das 3D-Modell und im zweiten die Steuerung, welche über Slider und Eingabefelder die Parameter setzt, die anschließend über ein MATLAB Interface die neue Position des Roboters berechnet und mithilfe von Simulink das 3D-Modell aktualisiert. Im Hintergrund arbeitet dabei eine RSS Schnittstelle, welche in vorangegangenen Projekten entwickelt wurde.

Das System ist in dieser Version zwar auf drei bewegliche Gelenke (Hüfte, Schulter und Ellbogen) beschränkt, zeigt aber eine realitätsnahe Simulation des Roboter Armes (Abb. 1b). (Wirabhuana und bin Haron, 2004)

3.2 Virtual Robot Controller Interface

Der Virtual Robot Controller (VRC) ist Bestandteil der vom Fraunhofer IPK⁵ entwickelten *Realistic Robot Simulation* Schnittstelle. Das Projekt wurde Anfang der 90er vom IPK zusammen mit führenden Industrieunternehmen gegründet. Die Schnittstelle ermöglicht es, die originale Bewegungssoftware von Robotersteuerungen in Simulationssysteme zu integrieren. Mittlerweile ist sie zum weltweiten De-facto-Standard zur präzisen Simulation von Roboterbewegungen geworden. (IPK, 30. Juli 2010)

Die Idee für den VRC entstand aus folgendem Problem: Zu Beginn der offline Programmierung von Robotern wichen trotz hoher mathematischer Genauigkeit die simulierten Trajektorien sehr stark von den *realen* Bewegungen der Roboter ab. Es entstanden Bewegungsabläufe, welche bis zu 300mm neben den Sollwerten lagen, für die industrielle Fertigung also unbrauchbar. Die Differenz kam dadurch zustande, dass der Controller welcher die Motoren des Roboters steuert, ein abweichendes Bewegungsprofil hatte. Die logische Folgerung war die Aufnahme des Controllers in die Simulation, die Entstehung des VRC. Im VRC werden alle Schnittstellen abgebildet, über welche der reale Controller verfügt, inklusive aller I/O Devices. Ein vollständiger VRC setzt sich aus den in Abb. 2 gezeigten (Bau-)Gruppen zusammen. Auf das Element Zeitmanagement möchte ich kurz näher eingehen. *Virtuelle Zeit* ist das Äquivalent zu real-time in der Simulation und erlaubt volle Kontrolle über die zeitlichen Abläufe während der Simulation. Jedes I/O Signal zum Beispiel beinhaltet einen Zeitstempel, welcher es ermöglicht, die Signalisierung zwischen mehreren Controllern präzise anzupassen. Das virtuelle Zeitmanagement ist das Nadelöhr bei der präzisen Simulation von Robotern, unter anderem dann, wenn die Zeiten für Werkzeug- und Objektwechsel mit eingeplant werden müssen. Da dies in der Realität nicht beliebig schnell passieren kann, wird hier darauf geachtet, dass auch realistische Zeiten in die Simulation einfließen. Sollte dies nicht korrekt erfolgen, entsteht wieder die Situation, welche vor dem VRC der Fall war. In der Simulation kann alles beliebig beschleunigt werden und am realen Objekt ergeben sich große Abweichungen. Über den VRC werden ebenfalls Tolleranzen sowie Korrekturfahrten der einzelnen Gelenkmotoren realisiert.

⁵Institut für Produktionsanlagen und Konstruktionstechnik

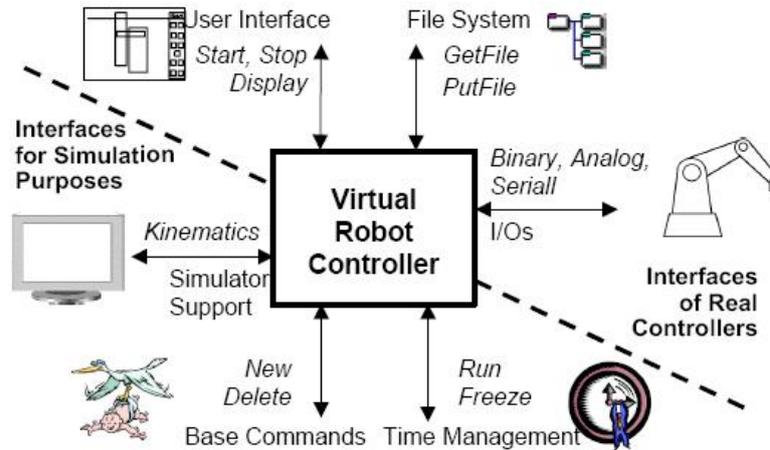


Abbildung 2: Die Module des VRC [Quelle: Fraunhofer IPK]

Um von der Simulation im RRS/VRC zum konkreten Anwendungsfall zu kommen, wurde folgender Arbeitsablauf festgelegt:

1. *Graphische Programmierung:* Der Benutzer definiert die Punkte und belegt diese mit Attributen wie Bewegungsart und -geschwindigkeit. Anschließend werden die VRCs festgelegt und der Simulator übersetzt die Controller spezifische Sprache. Dieser Prozess ist für den Anwender transparent und wird meist in früher Planungsphase angewandt. Das Ergebnis bildet die Basis für den nächsten Arbeitsschritt.
2. *Native Language Programming in der Simulation:* Hier wird in der Sprache des Controllers des Roboters der Arbeitsablauf einprogrammiert. In diesem Schritt ist die gesamte Bandbreite der VRC Funktionalität nutzbar. Es wird nun ausführlich das simuliert, was am Ende auf den realen Roboter in der Werkshalle aufgespielt wird.
3. *Native Language Programmierung am Roboter:* In diesem, letzten Schritt wird das zuvor generierte Programm auf den Roboter geladen und final getestet.

Diese drei Schritte beinhalten jeweils zahlreiche Iterationen des Modifizierens und Testens. Zwischen den einzelnen Schritten gibt es zusätzliche Datenkonsistenz Prüfungen. Die eigentliche Code-Generierung wird von Tools durchgeführt, welche der Roboter Hersteller bereitstellt.

Durch diese Vorgehensweise konnte in der Vergangenheit die Produktivität deutlich gesteigert werden, zum einen durch die Zeitersparnis beim Maschinenstillstand und zum anderen durch die verringerte Fehlerhäufigkeit durch ausgiebiges Testen während des Simulationsprozesses. (Bernhardt u. a., 30. Juli 2010)

3.3 MATLAB Simulation autonomer mobiler Roboter

Das dritte Projekt, welches in dieser Arbeit vorgestellt wird handelt ebenfalls von der MATLAB gestützten Simulation autonomer mobiler Roboter, jedoch mit Fokus auf Indoor Navigation. Die Arbeit "A Matlab-based Simulator for Autonomous Mobile Robots" von Jiali Shen and Huosheng Hu der University of Essex beschreibt die Komponenten eines selbst entwickelten

2D-Modells für die Entwicklung und Analyse von Navigationsalgorithmen. Ziel war es, eine einfach zu bedienende Applikation zu entwickeln, mit der beliebige Roboter mit unterschiedlichen Antriebsarten und Sensoren zur Abstandserkennung sowie Kameras in einer zuvor generierten Umgebung simuliert werden können.

Das System sollte unter anderem folgende Anforderungen erfüllen:

- Einfache Oberfläche zum Erstellen von Robotern und Umgebungen
- Live Beobachtungsmodelle zur Echtzeit-Analyse verschiedener Szenarien
- Simulation einer Kollisionserkennung und -reaktion

Es wird hier deutlich die Navigation in den Mittelpunkt gestellt. Die Anwendung wurde wiederum in drei Teile unterteilt: User Interface, Verhaltenssteuerung, Datenzusammenführung und Simulator Output.

User Interface

User Interaktion ist zum einen über Customer Configuration Files (CCF) möglich, welche die Erstellung von 2D-Umgebungen sowie eine Implementierung eigener Algorithmen zulassen und zum anderen durch Subroutinen, welche über den Simulator den Roboter steuern. Die Umgebungen bestehen aus Wänden, Korridoren, Türen, Objekten und sog. Beacons. Der Noise⁶ Generator kann ebenfalls konfiguriert werden.

Mit den Subroutinen kann der Benutzer das Verhalten z.B. bei einer Kollision oder beim Erkennen eines bestimmten Gegenstandes festlegen. In den CCF können noch viele weitere Werte (Robotermaße, Antrieb, Schlupf etc.) konfiguriert werden, auf welche in dieser AW2 Ausarbeitung jedoch nicht näher eingegangen wird.

Verhaltenssteuerung

Neben den im letzten Abschnitt erwähnten benutzerdefinierten Routinen, sind die Lokalisierung sowie ein Obstacle avoidance⁷ (OAM) - und ein Bahnplanungsmodul fester Bestandteil des Simulators. Das OAM kann über CCF ausgeschaltet werden. Hierbei ist jedoch zu beachten, dass das integrierte Module für die Reaktion auf eine Kollision davon unberührt bleibt, da sonst der Roboter in der Simulation durch Wände fahren könnte.

Datenzusammenführung

Wie der Name des Moduls *Data Fusion* bereits sagt, werden hier die Daten von CCD Kameras oder anderen Sensoren des Roboters zusammengeführt, um eine möglichst realitätsnahe Simulation zu schaffen. Dafür wurde das Observation Simulation Module, kurz OSM, geschaffen. Ausgehend von der Pose und der über die CCFs konfigurierten Anordnung der Beacons können so die tatsächlich im *Sichtbereich* liegenden Beacons sowie Distanz und Richtung

⁶in diesem Zusammenhang: Störung

⁷einem Gegenstands ausweichen

der momentanen Beobachtungsrichtung bestimmt werden. Mit den ebenfalls vorkonfigurierten Kameraparametern kann ebenfalls das Bild, welches der Roboter sieht, berechnet werden; es ist sogar möglich in eine Ansicht hineinzuzoomen. Die Daten vom OSM stellen somit das reale Bild in der Simulation bereit.

Simulator Output

Dieses Modul stellt verschiedene 2D-Ansichten auf die vom OSM bereitgestellten Daten bereit. Zum einen eine Karte mit der Umgebung und der gefahrenen Bahn als Linie dargestellt sowie zum anderen eine Sicht auf den Roboter und das was sich in seiner unmittelbaren Sensorreichweite befindet (Abb. 3). Bei Ausführung der Simulation entsteht ein Video, welches aus 25 permanent aktuell berechneten Bildern, welche über das OSM durch die Aufbereitung aller CCF Parameter und der Subroutinen errechnet werden, besteht. (Shen und Hu, 30. Juli 2010)

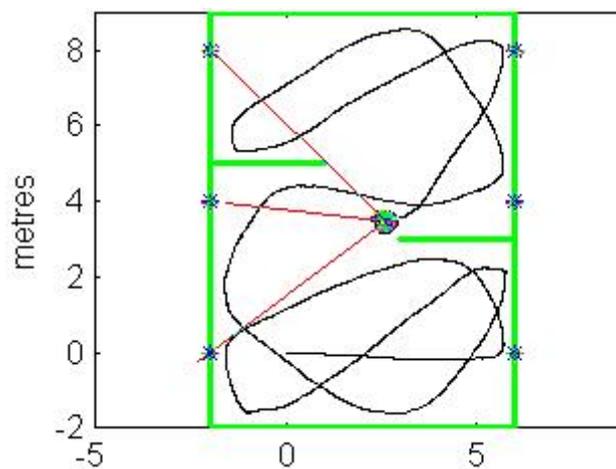


Abbildung 3: Die Sicht des Simulators

4 Ergebnisse

Alle drei vorgestellten Projekte haben zum Teil sehr unterschiedliche Ansätze für die Simulationsumgebung gewählt. Je nach Anwendungsgebiet wird der Roboter in 2D oder 3D in einer eigens geschaffenen Umgebung stark abstrahiert modelliert oder aber mit einem sehr hohen Detaillierungsgrad selbst der Controller des realen Roboters in der Simulation abgebildet. Unabhängig jedoch vom Einsatz als kommerzieller Standard oder in universitärer Forschung haben alle Projekte eines gemeinsam: Den Versuch, die Komplexität der Thematik für den Anwender einfacher handhabbar zu machen.

4.1 Bewertung der Projekte

Als kommerzielles Projekt hat das seit 1990 laufende RRS mit VRC vom Fraunhofer IPK einen deutlichen Technologievorteil. Es wird bis ins kleinste Detail jede Funktionalität des

realen Roboters simuliert, so dass in der Simulation getestete Trajektorien und andere Abläufe fehlerfrei von der Simulation auf den Roboter gespielt werden können. In den im Paper dargestellten Beispielen verfügen die Roboter jedoch nicht über Sensoren wie CCD Kameras o.Ä. und sind ebenfalls nicht autonom. Die Genauigkeit der Kinematik ist jedoch durch Umsetzung des VRC auf einem sehr hohen Niveau.

Das zuerst vorgestellte Projekt setzt den Fokus auf die Visualisierung des Roboters. Dabei sind durch die Verwendung von VRML der Detaillierung kaum Grenzen gesetzt, was eine sehr genaue Darstellung ermöglicht. Das verwendete Modul zur Gelenkkoordination des Roboters, heute direkt über die Robotics Toolbox⁸ von Peter Corke frei verfügbar in MATLAB verwendbar, bietet eine direkte Steuerung. Bei Bedarf können in wenigen Schritten Anpassungen an den aktuell zu simulierenden Roboter vorgenommen werden. Da das Projekt keinerlei Sensorik berücksichtigt, ist es für die Simulation autonomer Roboter in dieser Form nicht verwendbar.

Das zweite vorgestellte und ebenfalls in MATLAB realisierte Projekt bietet dort ein größeres Potential. Durch den modularisierten Aufbau von Umgebung, Roboter, Navigation, Kollisionserkennung und User Interface ist ein sehr hohes Maß an Konfigurierbarkeit gegeben. Der Anwender kann zusätzlich eigene Algorithmen einbinden sowie verschiedene mitgelieferte Module (de-) aktivieren. Auch die Analyse der Simulation ist gut umgesetzt, so dass Fehler oder zuvor nicht berücksichtigte Einflüsse auf die Navigation sofort auffallen. Das Simulationsmodell beinhaltet ebenfalls die Modellierung von CCD Kameras und Abstandssensoren, was für einen autonomen Roboter sehr wichtig ist. Lediglich die Darstellung in 2D lässt viel Raum für Verbesserungen. Die Simulation eines Roboters mit mehr als x- und y-Richtung als Positionsangabe ist schlicht nicht möglich. Dafür können aber die Antriebe sowie Störfaktoren wie Schlupf der Räder mit einbezogen werden.

4.2 Einordnung der eigenen Arbeit

Meine Arbeit, die Entwicklung einer Simulationsumgebung für visuell geführte Roboter (in MATLAB), beschäftigt sich mit der Darstellung eines beliebig gearteten Roboters in einer 3D-Umgebung sowie mit der Darstellung der Bildebene einer auf diesem montierten Kamera. Das Projekt wird mit dem Ziel verfolgt, an der HAW Hamburg für alle Studenten, die sich mit Greifarm-Robotern beschäftigen, eine Plattform bereitzustellen.

Dabei wird insbesondere auf folgende Modularität geachtet:

- *Roboter*: Kinematik, Trajektorie
- *Kamera*: Kameramodell, Darstellung der Projektion, Bildverarbeitung
- *Umgebung*: Dreidimensionale Darstellung von Roboter, Objekten und Environment

Sieht man diese Anforderungen im Kontext der vorgestellten Arbeiten, werden einige Parallelen deutlich. So wird ein 3D-Modell des Roboters (Projekt 1) benötigt, welcher vollständig konfigurierbar sein soll (Projekt 3) und gleichzeitig sehr hohe Erwartung an die Genauigkeit (Projekt 2) erfüllen soll; schließlich soll ein Assistenzroboter simuliert werden, welcher mit Menschen interagiert.

⁸http://www.petercorke.com/Robotics_Toolbox.html

Es handelt sich bei der Arbeit um ein rein universitäres Projekt, welches von vielen Zuarbeiten aus den einzelnen Modulen abhängig ist. Für die Qualität ausschlaggebend ist folglich zum einen die Handhabbarkeit als auch die Umsetzung von komplexen Algorithmen zusammen mit einem anschaulichen Modell, welches die Analyse dieser zulässt.

Aktuell befindet sich das Projekt in der Findungsphase. Viele Ideen sind in Prototypen umgesetzt worden sowie Teile der Umgebung bereits modularisiert. Durch die große Vielfalt an verwandten Arbeiten zu dem Thema sind im Verlaufe dieser Ausarbeitung weitere Erkenntnisse gewonnen worden sowie zahlreiche neue Ansätze entstanden.

Die Machbarkeit der Arbeit als solche ist gewährleistet. Im nächsten Schritt wird es darum gehen, den Grad der umzusetzenden Komplexität festzulegen um ein zufriedenstellendes Produkt zu schaffen; "Die Fusion von Simplicity & Precision"⁹.

5 Zusammenfassung und Ausblick

In dieser Ausarbeitung wurden mehrere Projekte aus dem Gebiet der Simulationsumgebungen für Roboter behandelt. Aus ihnen und weiteren Arbeiten sind viele Erkenntnisse entstanden, welche meine eigene Arbeit beeinflussen (werden). Die beiden mit MATLAB umgesetzten Projekte aus der universitären Forschung beschreiben grundlegende Möglichkeiten, auf denen weitere Arbeiten aufgebaut werden können. Der VRC setzt Maßstäbe bei der Präzision, ist jedoch viel zu komplex und umfangreich, um an der HAW eingesetzt zu werden. Dennoch liefert er wichtige Erkenntnisse über die Schwierigkeiten bei der Umsetzung von Funktionen aus der Simulation in die reale Welt.

Nachdem nun verwandte Arbeiten untersucht wurden und die eigene Arbeit dazu in Bezug gesetzt wurde, wird noch ein Ausblick auf kommende Aufgaben des Projektes "Simulationsumgebung für visuell geführte Roboter" gegeben: Die nächsten Schritte des Projektes umfassen unter anderem die Erstellung eines VRML Modells des Katana Roboters sowie des neuen 8 DOF¹⁰ Roboters der HAW und die Einbindung in die aktuelle Simulation. Die zweite, große Herausforderung ist die Entwicklung einer Bewegungsstrategie in der 3D Umgebung, welche von der bereits erwähnten Robotics Toolbox bereitgestellt wird. Für Projekt 1 kommt noch ein weiterer Punkt hinzu: Wiedererkennung simulierter markanter Punkte in der Bildebene der Lochkamera.

Alles in allem bietet diese Ausarbeitung wichtigen Input für die kommenden Projekte und die Masterarbeit.

⁹Einfachheit und Präzision

¹⁰Degree of Freedom - Freiheitsgrad

Literatur

- [Bernhardt u. a. 30. Juli 2010] BERNHARDT, Rolf ; SCHRECK, Gerhard ; WILLNOW, Cornelius: *THE VIRTUAL ROBOT CONTROLLER (VRC) INTERFACE*. 30. Juli 2010. – URL <http://www.realistic-robot-simulation.org/7public.pdf>
- [DPA 28. Oktober 2006] DPA: Verspäteter Start. In: *Stern* (28. Oktober 2006). – URL <http://www.stern.de/wirtschaft/news/unternehmen/vw-in-russland-verspaeteter-start-575093.html>
- [IPK 30. Juli 2010] IPK, Fraunhofer: *Realistik Robot Simulation*. 30. Juli 2010. – URL <http://www.ipk.fhg.de/geschaeftsfelder/automatisierungstechnik/prozessautomatisierung-und-robotik/forschung-und-entwicklung/rrs-realistic-robot-simulation>
- [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