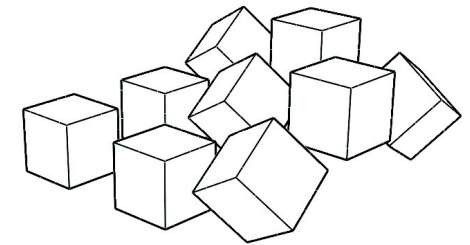
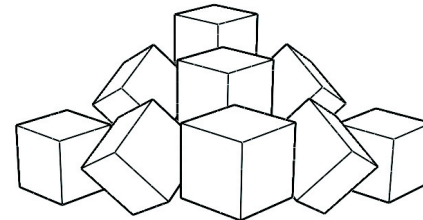
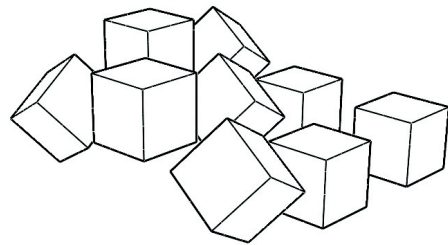


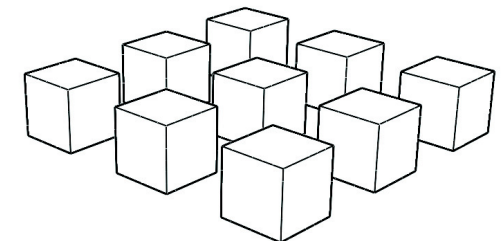
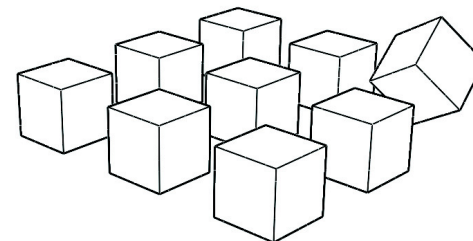
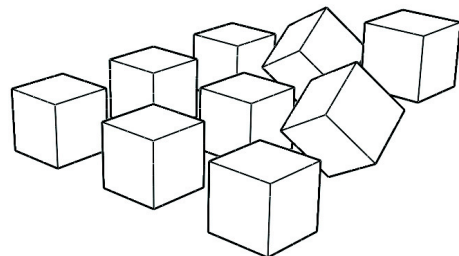
Robotertraum



Projekt Seminar betreut von Prof. Oliver Vogt

in Kooperation mit Prof. Kurt Geihs, FB 16 Elektrotechnik/Informatik

Ulf Cadenbach | 11./12. Semester Produktdesign | Kunsthochschule Kassel



Inhalt

Inhaltsverzeichnis	Seite 2
Hintergrund Roboter	Seite 3
Projekt Robotertraum	Seite 8
Phase 1: Ideenfindung	Seite 9
Phase 2: Entwürfe	Seite 11
Phase 3: Umsetzung	Seite 16
Phase 4: Simulation	Seite 18
Phase 5: Funktionsmodell	Seite 22
Phase 6: Weitere Aktivitäten	Seite 24
Fazit	Seite 26
Appendix	Seite 27
Quellen	Seite 32
Multimedia CD	Im hinteren Einband

Hintergrund

Roboter

aus Wikipedia, der freien Enzyklopädie

Roboter sind stationäre oder mobile Maschinen, die nach einem bestimmten Programm festgelegte Aufgaben erfüllen. Allerdings hat sich die Bedeutung im Laufe der Zeit gewandelt. Der Begriff Roboter (tschechisch: robot) wurde von Josef und Karel Capek Anfang des 20. Jahrhunderts durch die Science-Fiction-Literatur geprägt. Sein Ursprung liegt im slawischen Wort robot, welches mit Arbeit, Fronarbeit oder Zwangsarbeit übersetzt werden kann. 1921 beschrieb Karel Capek in seinem Theaterstück R.U.R. in Tanks gezüchtete menschenähnliche künstliche Arbeiter. Mit seinem Werk greift Capek das klassische Motiv des Golems auf. Heute würde man Capeks Kunstgeschöpfe als Androiden bezeichnen. Vor der Prägung dieses Begriffes wurden Roboter zum Beispiel in den Werken von Stanislaw Lem als Automaten oder Halbautomaten bezeichnet.

Definition

Während der Entwicklung von Handhabungsgeräten, die immer komplizierter wurden, kamen Entwickler auf die Idee, sie Roboter zu nennen. Spätestens ab diesem Zeitpunkt wurde das Wort Roboter, welches ursprünglich nur für humanoide Roboter verwendet wurde, fast beliebig für verschiedene Geräte benutzt. Entsprechend unterschiedlich ist die Definition eines Roboters von Land zu Land. So kommt es, dass z.B. 1983 von Japan 47000

dort installierte Roboter gemeldet wurden, von denen nach VDI-Richtlinie 2860 nicht einmal 3000 als Roboter gegolten hätten. [1]

Definition nach VDI-Richtlinie 2860

Industrieroboter sind universell einsetzbare Bewegungsautomaten mit mehreren Achsen, deren Bewegungen hinsichtlich Bewegungsfolge und Wegen bzw. Winkeln frei (d.h. ohne mechanischen Eingriff) programmierbar und gegebenenfalls sensorgeführt sind. Sie sind mit Greifern, Werkzeugen oder anderen Fertigungsmitteln ausrüstbar und können Handhabungs- und/oder Fertigungsaufgaben ausführen.

Definition nach Robot Institute of America (RIA)

Ein Roboter ist ein programmierbares Mehrzweck-Handhabungsgerät für das Bewegen von Material, Werkstücken, Werkzeugen oder Spezialgeräten. Der frei programmierbare Bewegungsablauf macht ihn für verschiedenste Aufgaben einsetzbar.

Definition nach Japan Robot Association (JARA)

* Manual Manipulator: Handhabungsgerät, das kein Programm hat, sondern direkt vom Bediener geführt wird.

* Fixed Sequence Robot: Handhabungsgerät, das wiederholt nach einem konstanten Bewegungsmuster arbeitet. Das Ändern des Bewegungsmusters ist relativ aufwendig.

* Variable Sequence Robot: Handhabungsgerät, wie vorher beschrieben, jedoch mit der Möglichkeit, den Bewegungsablauf schnell und problemlos zu ändern.

* Playback Robot: Der Bewegungsablauf wird diesem Gerät einmal durch den Bediener vorgeführt und dabei im Programmspeicher gespeichert. Mit der im Speicher enthaltenen Information kann der Bewegungsablauf beliebig wiederholt werden.

* Numerical Control Robot: Dieses Handhabungsgerät arbeitet ähnlich wie eine NC-gesteuerte Maschine. Die Information über

den Bewegungsablauf wird dem Gerät über Taster, Schalter oder Datenträger zahlenmäßig eingegeben.

* Intelligent Robot: Diese höchste Roboterklasse ist für Geräte gedacht, die über verschiedene Sensoren verfügen und damit in der Lage sind, den Programmablauf selbsttätig den Veränderungen des Werkstücks und der Umwelt anzupassen.

Roboter in Filmen

Da die Idee des Roboters ihren Ursprung im Theater hat, wurde diese auch später in vielen Filmen thematisiert. Im 1927 erschienenen Film Metropolis wurde erstmals ein Roboter in einem Film gezeigt. In der Star Wars Saga (1977 - 2005) spielen R2D2 und C3PO eine eher komödiantische Rolle. In der Serie Star Trek - The Next Generation (1987 - 1994) ist der Androide Data ein Führungsoffizier. Heute bringen Filme wie "Terminator" und "I, Robot" das Publikum immer noch über den Sinn dieser meist menschenähnlichen Maschinen zum Grübeln. Immer häufiger finden auch Industrieroboter (nicht humanoide Roboter) den Weg auf die Leinwand. In Filmen wie James Bond (Stirb an einem anderen Tag), Thunderbirds, Tomb Raider – Die Wiege des Lebens und Sakrileg wurden z.B. KUKA Industrieroboter in Szene gesetzt.

Roboter in der Literatur

Erstmals wurde der Begriff Roboter durch die Erzählungen Isaac Asimovs einem größeren Publikum bekannt. 1942 beschrieb er in "Runaround" (English) zum ersten mal die drei Robotergesetze die auch heute noch häufig in der Science Fiction Literatur rezipiert werden. 1950 veröffentlicht er mit I, Robot eine Sammlung von Kurzgeschichten zu diesem Thema. Auch später beschäftigt sich Asimov immer wieder mit dem Thema Roboter.

Robotik

Durch die häufige Thematisierung von Robotern in Film und Literatur, wurde auch die Wissenschaft aufmerksam auf diese Art der Maschinen. Das wissenschaftliche Gebiet, das sich mit der

Konstruktion von Robotern beschäftigt, heißt Robotik. Der Begriff wurde 1942 von Isaac Asimov in seinem Buch "Runaround" erstmals erwähnt. Ein allgemeines theoretisches wissenschaftliches Gebiet, welches sich mit Robotern beschäftigt, gibt es nicht. Sie sind meist Teilgebiete der Informatik.

Technische Grundlagen

Technisch realisiert werden Roboter hauptsächlich im Zusammenspiel der Disziplinen: Mechanik, Elektrotechnik und Informatik. Inzwischen hat sich aus der Verbindung dieser drei Disziplinen die Mechatronik entwickelt. Um autonome Systeme zu entwickeln, die eine gewisse Eigenständigkeit, beispielsweise bei der Wegfindung aufweisen, werden immer mehr wissenschaftliche Disziplinen in die Robotik eingebunden. Hier liegt ein Schwerpunkt bei der Verbindung von Konzepten der Künstlichen Intelligenz oder der Neuroinformatik (Teilgebiete der Informatik) und ihrer biologischen Vorbilder Biokybernetik (Teilgebiet der Biologie). Aus der Verbindung von Biologie und Technik entstand wiederum die Bionik.

Wichtigste Bestandteile eines Roboters sind die Sensoren, zur Erfassung der Umwelt, die Aktoren, zum agieren innerhalb der erfassten Umgebung, die Robotersteuerung und das mechanische Gestell des Roboters. Ein Roboter muss nicht unbedingt vollständig autonom handeln können, darum unterscheidet man autonome und ferngesteuerte Roboter.

Roboterarten

Der Begriff „Roboter“ beschreibt ein weitgefächertes Gebiet, weshalb man Roboter in viele Kategorien einordnet. Einige davon sind:

- * Autonome mobile Roboter
- * Humanoide Roboter
- * Industrieroboter
- * Portalroboter
- * Serviceroboter
- * Spielzeugroboter
- * Erkundungsroboter
- * Laufroboter
- * BEAM -> <http://de.wikipedia.org/wiki/BEAM>

Humanoide Roboter

Das Bild des Humanoiden Roboters in der Literatur wurde wie bereits erwähnt maßgeblich durch die Erzählungen Isaac Asimovs in den 40er Jahren des 20. Jahrhunderts geprägt. Humanoide Roboter waren lange Zeit technisch nicht realisierbar. Für die Entwicklung humanoider Roboter müssen viele wichtige Probleme gelöst werden. Sie sollen autonom in ihrer Umwelt agieren und möglichst auch interagieren können, wobei ihre Mobilität durch zwei Beine als Fortbewegungsmittel beschränkt ist. Außerdem sollen sie durch zwei künstliche Arme und Hände Arbeiten verrichten können. Seit 2000 (z. B. ASIMO von Honda [2]) scheinen die grundlegenden Probleme gelöst. Inzwischen werden regelmäßig neue Entwicklungen in diesem Bereich vorgestellt.

Die Humanoiden können auch der Gattung Laufroboter zugeordnet werden, während man einige Serviceroboter eher als Fahrroboter bezeichnen müsste.

Industrieroboter

1954 meldete George Devol erstmals ein Patent für Industrieroboter an. Heutige Industrieroboter sind in der Regel nicht mobil. In ihrer Form und Funktion sind sie speziell auf eine oder wenige Einsatzgebiete festgelegt. 1961 wurden sie erstmals bei General Motors in Produktionslinien eingesetzt. In Deutschland wurden Industrieroboter, beispielsweise für Schweissarbeiten in der Automobilindustrie, seit etwa 1970 eingesetzt.

Zu den Industrierobotern zählen auch die sogenannten Portalroboter, die beispielsweise bei der Produktion von Wafern, in Vergussanlagen oder in der Messtechnik eingesetzt werden. Heute werden auch viele Handlingaufgaben durch Industrieroboter ausgeführt.

Haushaltsroboter

Haushaltsroboter verrichten selbständig Arbeiten im Haushalt. Bekannte Anwendungen umfassen

- * Staubsaugroboter (beispielsweise von Electrolux, Siemens oder iRobot)
- * Rasenmäherroboter
- * Fensterreinigungsroboter

Spielroboter

Die meisten roboterähnlichen Spielzeuge sind keine Roboter, da ansonsten sämtlich selbst bewegende Gegenstände als Roboter anzusehen wären. Trotzdem gibt es Roboter die man als Spielroboter bezeichnet, da ihr automatisierter Funktionsumfang keinerlei arbeits- oder forschungstechnischen Nutzen haben. Ein Beispiel hierfür ist der Hund ähnelnde Lauf- und Spielroboter Aibo von Sony oder der Robosapien von WowWee. Neben seiner Funktion als Spielzeug, kann er jedoch auch professionell eingesetzt werden, was man beispielsweise in der Four-Legged

League beim jährlichen Roboterfußball sieht. Seine Produktion wurde trotzdem eingestellt.

Erkundungsroboter

Unter Erkundungsrobotern versteht man Roboter, die an Orten operieren, die für den Menschen (lebens-)gefährlich oder gar unzugänglich sind und ferngesteuert oder (teilweise) autark operieren. Dies gilt z. B. für Gebiete, in denen ein militärischer Konflikt ausgetragen wird. Aber auch für Gegenden, die bisher für den Menschen nur sehr schwer oder gar nicht erreichbar sind, wie z. B. die Mond- oder Mars-Oberfläche. Schon allein wegen der riesigen Entfernung der anderen Planeten ist eine Fernsteuerung von der Erde aus unmöglich, weil die Signale hin und zurück Stunden benötigen würden. In diesen Fällen müssen dem Roboter eine Vielzahl von möglichen Verhaltensweisen einprogrammiert werden, wovon er die sinnvollste wählen und ausführen muss.

Zur Erkundung enger Pyramidenschächte, in die Menschen nicht eindringen können, wurden schon mit Sensoren bestückte Roboter eingesetzt. Es wird auch darüber nachgedacht, einen sogenannten Cryobot, der sich durch Eis schmilzt, in den Wostoksee herab zu lassen. Dieser ist von der Außenwelt durch eine über 3 km dicke Eisschicht hermetisch abgeriegelt. Forscher vermuten in diesem ein unberührtes Ökosystem, was auf gar keinen Fall durch „oberirdische“ Mikroben kontaminiert werden soll.

Drohnen/Kampfroboter

Drohnen sind Roboter, die zu (meist militärischen) Aufklärungszwecken eingesetzt werden. Diese können sich in der Luft, zu Land oder auf/unter Wasser selbstständig oder autark bewegen. Beispiele hierfür sind die luftgestützte Global Hawk oder die landgestützte SWORDS. Diese können sowohl zur reinen Selbstverteidigung als auch zum aktiven Angriff auf Ziele Waffen mit sich tragen.

Rover/Lander

Unter einem Rover versteht man in der Raumfahrt Roboter, die sich mobil auf der Oberfläche anderer Himmelskörper (z. B. Mond/Mars) fortbewegen. Beispiele hierfür sind die sowjetischen Lunochod-Mondmobile oder die Zwillingroboter Spirit und Opportunity. Letztere können sich unabhängig von der Bodenkontrolle ihren Weg suchen. Auch nicht mobile Einheiten, sogenannte Lander, können als Roboter bezeichnet werden. Die Mondrover der Apollomissionen waren keine Roboter, weil sie direkt von Menschen gesteuert wurden.

Sonstige

Ebenfalls als Roboter bezeichnet man mobile Einheiten, die zum Aufspüren, Entschärfen oder Sprengen von Bomben oder Minen eingesetzt werden, wie z. B. der sogenannte TALON-Roboter. Auch gibt es Roboter, die in Trümmern nach verschütteten Menschen suchen können. Mittlerweile gibt es auch den sog. Killer-Roboter. [3] [4]

Zukünftige Entwicklung

2004 waren 2 Millionen Roboter im Einsatz, allein im Jahr 2008 werden 7 Millionen zusätzliche Installationen erwartet.[5] Die EU fördert Forschungsarbeiten, die bis 2010 ermöglichen könnten, dass Roboter für einfache Tätigkeiten in Krankenhäusern eingesetzt werden, z. B. Krankentransport, Essensausgabe, Reinigungsarbeiten. [6] Bis Ende 2008 soll ein Prototyp als Erntehelfer auf Obstplantagen entwickelt werden. [7]

Literatur

* Gero von Randow: Roboter - Unsere nächsten Verwandten.
Reinbek: Rowohlt 1997. ISBN 3-498-05744-8

Weblinks

Wiktionary: Roboter – Bedeutungserklärungen, Wortherkunft,
Synonyme und Übersetzungen
-> <http://de.wiktionary.org/wiki/Roboter>

Commons: Roboter – Bilder, Videos und Audiodateien
-> <http://commons.wikimedia.org/wiki/Category:Robots?uselang=de>

Roboternetz-Wiki
-> <http://www.roboternetz.de/wissen/index.php/Hauptseite>

Text zur Robotersicherheit (engl.)
-> <http://www.industrialsafetytalk.com/news/lca/lca106.html>

Chaosradio: Ein Überblick über die technische Entwicklung von
Robotern
-> <http://de.wikipedia.org/wiki/Chaosradio>

[1]
-> <http://www.roboter-info.de/>

Spielzeugroboter aus 70 Jahren
-> <http://www.hotbotz.de/>

Aufstand der Maschinen - Roboter fordern Rechte 2050:
Zukunftsstudie prophezeit neue Welt mit Menschenrechten für
Maschinen, [presetext.de](http://www.presetext.de), 22. Dezember 2006
-> <http://www.presetext.de/pte.mc?pte=061222020>

[2]robotlab
-> <http://www.robotlab.de/>

Quellen

1. Dr.-Ing. Michael Naval, Roboter-Praxis, Würzburg: Vogel
Verlag (1989), ISBN 3-8023-0210-9

2. heise.de: Hondas humanoider Roboter läuft schneller und
sicherer
-> <http://www.heise.de/newsticker/meldung/83316>

3. golem.de: Samsung entwickelt Killer-Roboter für die
Objektsicherung
-> <http://www.golem.de/0701/50167.html>

4. heise.de: Robocop soll die innerkoreanische Grenze schützen
-> <http://www.heise.de/newsticker/meldung/84272>

5. golem.de: Bill Gates: Ein Roboter in jedem Haushalt bis 2013
-> <http://www.golem.de/0612/49631.html>

6. golem.de: Roboterkrankenschwestern schwirren ab 2010
durch die Gänge: EU fördert Forschungsarbeit
-> <http://www.golem.de/0701/50150.html>

7. heise.de: USA: Roboter sollen Erntehelfer ersetzen.
-> <http://www.heise.de/newsticker/meldung/91636>

Von „<http://de.wikipedia.org/wiki/Roboter>“

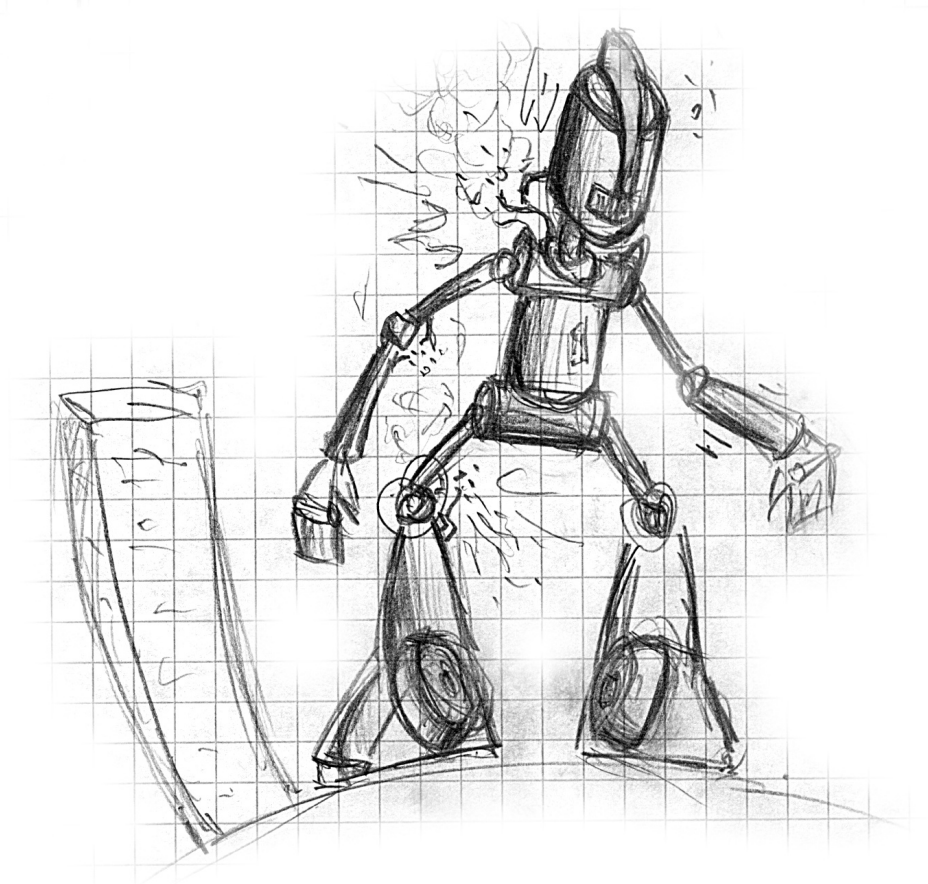
Projekt Robotertraum

Im Projektseminar "Robotertraum" versuchten wir uns zunächst eher spielerisch dem weitgefassten Feld der Robotik zu nähern. Dazu trafen wir uns jede Woche abwechselnd in der Kunsthochschule und in der Ingenieursschule bei unseren kooperierenden Informatikern.

So wurde gewährleistet, dass wir uns dem Thema Robotik von zwei Seiten her nähern konnten, einmal vom gestalterisch/künstlerischen Standpunkt der Produktdesigner und vom eher technischen/umsetzungsorientierten Standpunkt der Informatiker.

Während in der Kunsthochschule hauptsächlich mit Medien gearbeitet wurde; z.B. die Ergebnisse von umfassenden Netzrecherchen präsentiert wurden, Filmklassiker und Dokumentationen zum Thema gezeigt, oder aber auch "ganz praktisch" Spielzeugroboter ausprobiert und demontiert wurden, ging es bei den Informatikern etwas sachlicher zu. Hier wurde uns z.B. die Funktionsweise der RoboCup-Roboter der Uni Kassel am "lebenden Objekt" gezeigt und erklärt, grundlegende Funktionen der Sensorik und Datenverarbeitung autonom agierender Roboter näher gebracht und auf viele Fragen fundiert und realitätsbezogen geantwortet. Weiterhin wurden die Uni-Werkstätten besichtigt, um einen Einblick zu bekommen, was, wie und von wem realisiert werden könnte.

Nach dieser Phase des Eingewöhnens und "Beschnuppens" wurden die ersten eigenen Ideen, Entwürfe und Konzepte präsentiert, die von den Fachmännern der Informatik hinsichtlich ihrer Umsetzungsmöglichkeiten und der dafür notwendigen technischen Erfordernisse kommentiert wurden.



FB20_159 Robotertraum

Projektseminar 4.0 SWS

Studiengang: Produktdesign Diplom Uni

Dozent: Vogt

Termin: Mittwoch 10:00 - woch

Raum: Raum 0229 / Menzelstraße Atrium

Bemerkung zu o.g. Termin: Kooperation mit Prof. Kurt Geihs, FB 16 Elektrotechnik/Informatik

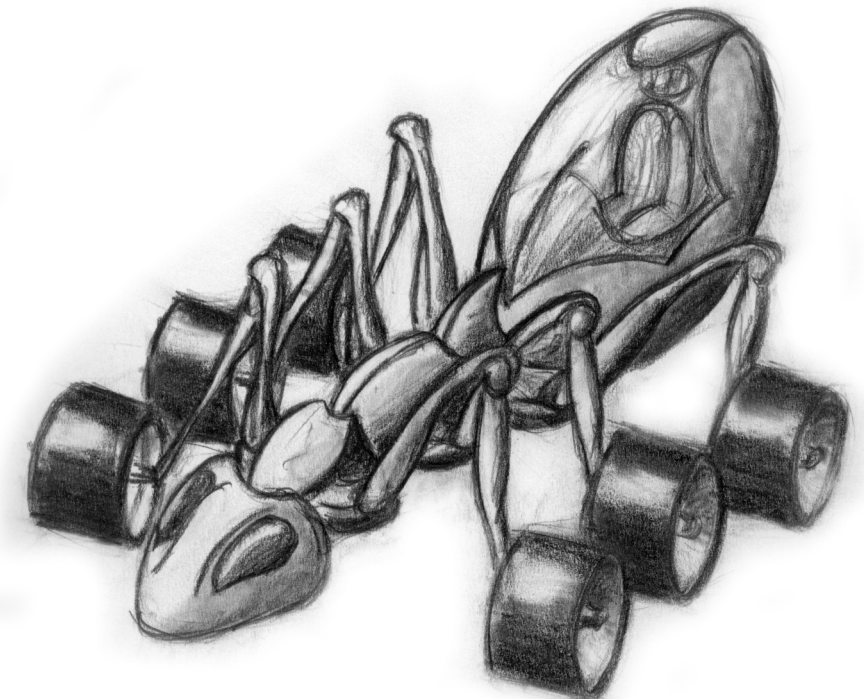
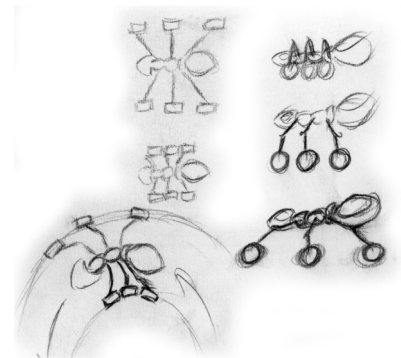
Phase 1: Ideenfindung

Zunächst einmal stellte sich mir die Frage, in welchen Bereichen der Robotik sich ein Designer sinnvoll einbringen kann. Da wären die "inneren" und "äusseren" Werte, die als Gesamtkonzept einen Roboter charakterisieren. Die Inneren stellen für mich die Sensorik, Elektronik und Steuerungssoftware dar, welche trotz marginaler Informatikkenntnisse wohl von vornherein ausserhalb meiner Kompetenzen liegen durften.

Bleiben noch die Äusseren, der Typus des Roboters (sprich, bewegt er sich auf Rädern, auf Beinen, bzw. überhaupt fort, hat er Greifer, Tentakeln oder besteht er aus einer einzigen glimmenden, berührungssensitiven Kugel... etc.pp.), die Anwendungsbereiche (Erkundung, Altenpflege, Automation von Prozessen, Kommunikation, etc.), die Gestaltung des Erscheinungsbildes für einen bereits bestehenden Typus (z.B. einen Roboter freundlich wirken zu lassen, um Ängste bei der Interaktion mit "ihm" zu minimieren oder die Aufgabe die ihm zukommt durch seine äussere Gestaltung visuell zu verdeutlichen) oder die Interaktion mit Menschen bzw. der Umwelt des Roboters.

Da ich seit jeher schon von den Errungenschaften der Bionik fasziniert bin, gerade auch, was die Umsetzung von Fortbewegungsstrategien von Tieren in technische/mechanische Konzepte angeht, war für mich relativ schnell klar, in welche Richtung meine Aktivitäten gehen würden. Trotz der durch meine Internetrecherchen bestärkten Erkenntnis, dass es fast nichts gibt, was es nicht gibt, wollte ich versuchen, einen neuen Typus von robotischer Fortbewegung zu entwickeln.

Erste Überlegungen brachten mich darauf, dass sowohl die Fortbewegung auf/mit Rädern, als auch die Fortbewegung auf/mit Beinen immer gewisse Vor- und Nachteile mit sich bringen. Über eine Kombination der beiden Fortbewegungsmöglichkeiten hatte ich mir bereits vor meinem Studium Gedanken gemacht (Zeichnungen aus dem Jahr 2001).



Räder sind zwar schnell, benötigen jedoch auch immer einen gewissen Grad an Haftung um überhaupt vorwärts zu kommen, können ohne Hilfsmittel keine Hindernisse, die grösser als ihr Radius sind, überwinden und sind zum Schwimmen sowieso gänzlich ungeeignet.

Beine sind zwar sehr viel universeller einsetzbar, eignen sich zur Überwindung von grösseren Hindernissen, wie auch zur Fortbewegung in widrigen Umgebungen, sogar zum Schwimmen, sind aber unheimlich schwierig zu koordinieren.

Während der Befehl für eine Vorwärtsbewegung bei der radbasierten Fortbewegung schlicht "drehe Rad (Räder) mit Geschwindigkeit X" (zwei Werte) bedeutet, wird es bei einer simplen Vorwärtsbewegung auf Beinen um Größenordnungen komplexer (z.B. "an Bein 1: bewege Gelenk A um 35° Richtung X, zeitgleich bewege Gelenk B um 60° in Richtung Y, danach bewege Gelenk C um ... " usw. den Rest kann man sich denken, oder auch nicht). Und dies war nur ein Bein von zwei, vier, sechs, acht oder noch mehr Beinen und nur ein Schritt vorwärts. Bei der Hindernisüberwindung oder bei der Fortbewegung in ungewohnter Umgebung wird das Ganze wieder einen Schritt komplizierter.

Worauf ich hinaus will dürfte klar sein, Fortbewegung mit Beinen erfordert ein großes Maß an Rechenkapazität, für minderkomplexe Lebensformen und "Low-Tech"-Anwendungen daher ungeeignet.

Dem gegenüber steht die Fortbewegung unter Einsatz des gesamten Körpers. Diese Art der Fortbewegung kommt in allen Bereichen der Tierwelt zum Einsatz und scheint "idiotensicher", weiterhin scheint sie in allen Umgebungen zu funktionieren. Einzeller wie Amöben, niedere Wirbellose wie Würmer, Schnecken, Quallen, aber auch höher entwickelte Wirbeltiere wie Schlangen, Plattfische nutzen sie und kommen hervorragend damit zurecht. Und das teilweise in grundsätzlich unterschiedlichen Lebensräumen.

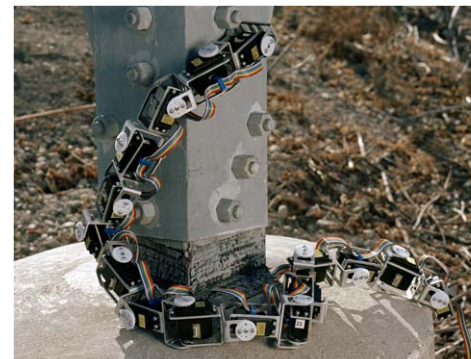
Beispielsweise unterscheidet sich die Fortbewegung einer Schlange auf Sand (Seitenwinder Klapperschlange) kaum von der Fortbewegung einer Wasserschlange im Wasser.

Ohne nun auf alle genannten Tiere näher eingehen zu wollen, eines ist ihren Fortbewegungsstrategien gemeinsam, sie versetzen ihren Körper bzw. Teile dessen in Wellenbewegungen.

Auch hierfür gab es bereits vielversprechende Ansätze, besonders interessant fand ich sog. Snakebots (Bild oben links), als auch die von Festo realisierten Projekte Air Ray (Bild unten links) und Aqua Ray (Bild rechts), die eine rochenartige Fortbewegung sowohl im Wasser, als auch in der Luft umsetzen. Ausführlichere Informationen dazu auf der beiliegenden CD.

Pfad: CD/ Ressourcen Projekt Robot Dreams/ Multimedia/ Schlangenartig

Pfad: CD/ Ressourcen Projekt Robot Dreams/ Multimedia/ Festo



Phase 2: Entwürfe

1. Entwurf

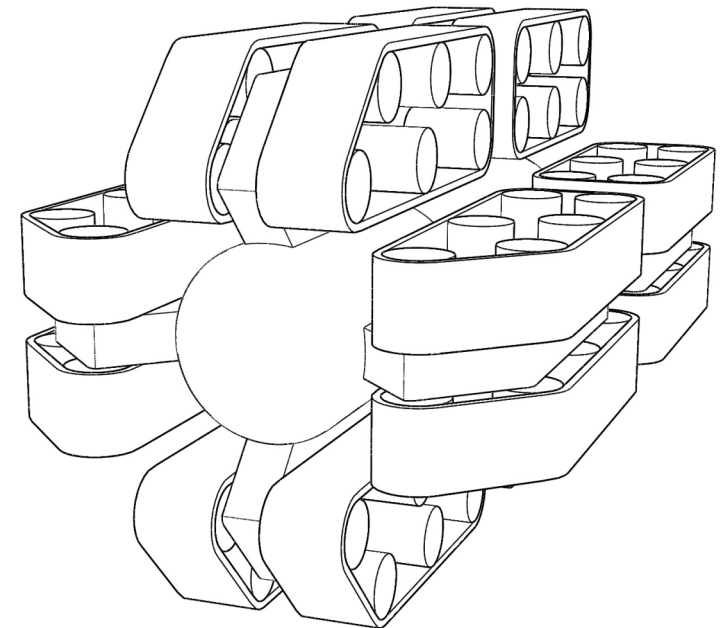
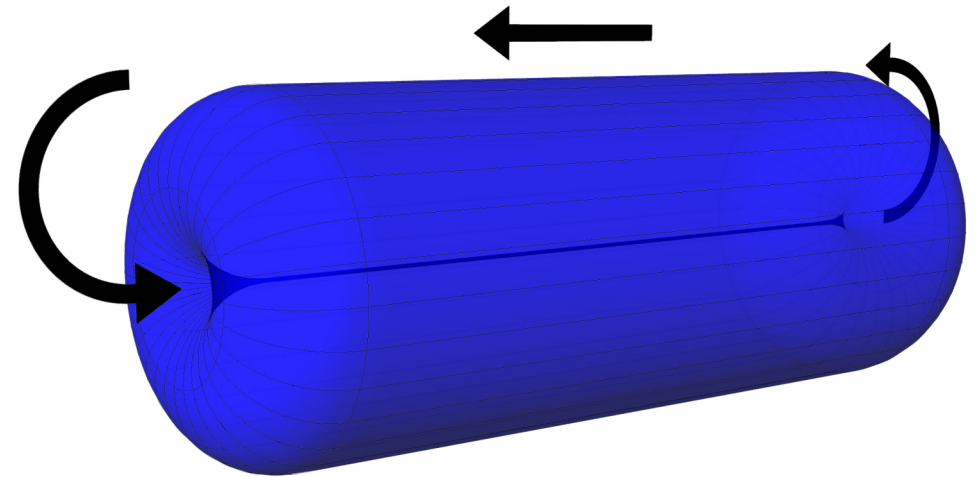
Die Grundidee war, die bestehenden Fortbewegungstypen "schlangenartige Bewegung" und "Fortbewegung auf/mit Ketten" zu kombinieren, um eine "wurmartige Fortbewegung" zu realisieren.

Würmer sind zwar ausserordentlich flexibel, bewegen sich aber im Endeffekt mehr durch Kontraktion und Relaxation ihres Körpers fort, als durch schlängelnde Bewegungen. Ein ähnlicher Effekt ist mit technischen Mitteln nur sehr kompliziert zu erreichen. Fakt ist jedoch, dass Würmer mit ihrer gesamten Körperoberfläche, bzw. mit ihrem gesamten Körperquerschnitt an einer Stelle Reibung erzeugen, an anderer Stelle wieder aufgeben und sich dadurch fortbewegen. Was allerdings immer noch technisch schwierig umzusetzen ist.

Die Lösung ist ein Kinderspielzeug, eine "Wasserwurst". Ich habe leider keinen adäquateren Namen dafür finden können.

Diese stellt im wesentlichen einen langgezogenen, wassergefüllten Torus dar (Bild oben).

Der interessante Effekt dabei ist, dass sich die äussere Oberfläche bei der "Fortbewegung" durch Kinderhände, am Ende nach innen stülpt, um zunächst die innere Oberfläche zu bilden, um dann aber am vorderen Ende wieder auszutreten und wiederum die äussere Oberfläche zu bilden. Dadurch ist es kaum möglich, dieses Objekt festzuhalten, es "flutscht" einem aus der Hand, da sich keine Möglichkeit bietet, dem immerwährenden Umstülpen dieses Körpers Einhalt zu gebieten.



Prinzipiell ist die nichts anderes als ein Rotationskörper der aus einem Ketten-/Raupenantrieb abgeleitet wird. Da solch eine "Wasserwurst" als Fortbewegungsmechanismus nun auch wieder technisch ausserordentlich kompliziert umzusetzen ist, muss man zwangsläufig zurück zur "Quadratur des Kreises" kommen. Ein Kettenantrieb auf drei, vier, oder sechs Seiten eines Objekts erfüllt einen ähnlichen Zweck.

Geschickt wäre auch, das Objekt in Sequenzen aufzuteilen, da ein umlaufender Ketten-/Raupenantrieb nur an einem starren Objekt funktionieren kann, es sei denn, die Ketten/Raupen wären äusserst flexibel.

Bilder des Entwurfs: vorherige Seite unten, diese Seite oben.

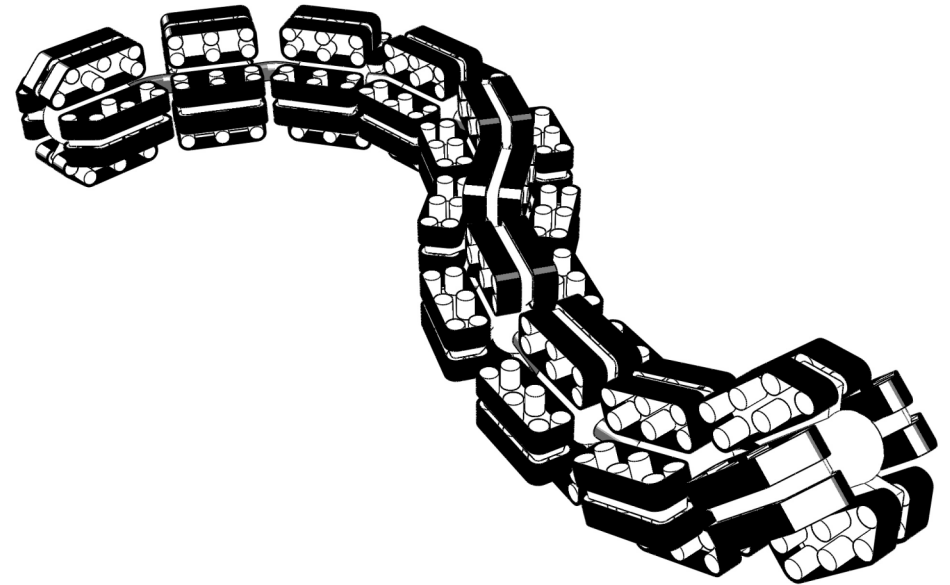
Vorteile:

Die Kombination der Vorteile schlangenartiger Fortbewegung und eines Kettenantriebes. Der Kettenantrieb gewährleistet eine relativ zügige Fortbewegung auf flachem bis mäßig unebenem Untergrund. Eine schlangenartige Fortbewegung ermöglicht das Überwinden von grösseren Hindernissen, sogar eingeschränkte "Kletterfähigkeiten", eventuell sogar Schwimffähigkeit. Ein "umlaufender" Kettenantrieb, d.h. auf 3, 4 oder mehr Seiten vermeidet das Problem des "Umfallens", der Roboter kennt kein oben oder unten und kann sich gleich gut fortbewegen, ob er nun auf der Unterseite, auf dem Kopf oder auf einer Seite liegt. Weiterhin kann eine zusätzliche Reibung genutzt werden, beispielsweise in engen Röhren. Der Roboter könnte sich "Verkeilen" und so zum Überwinden von Hindernissen Reibung von mehreren Seiten nutzen.

Einsatzgebiet:

Autonome Fortbewegung in gefährlicher und/oder unzugänglicher Umgebung, Erkundung

-Abbruch des Entwurfsprozesses, da Konzept schon existiert (Bild unten)



2. Entwurf

Nachdem ich beim ersten Entwurf nach einiger Arbeit feststellen musste, dass jemand anderes vor mir die gleichen Ideen hatte, ging ich dazu über, die Fortbewegung durch Einsatz des ganzen Körpers radikal auf ihre wesentlichen Elemente zu reduzieren. Jeder Körper besitzt eine (Ober-) Fläche, die in irgendeiner Art und Weise manipuliert wird. Bei Wirbeltieren geschieht dies meist durch eine in der Oberfläche liegenden Konstruktion, sprich Muskeln und Skelett. Mir fällt momentan kein Fall ein, in dem die Oberfläche (Haut) von Wirbeltieren als solches manipuliert würde.

Anders sieht es bei Wirbellosen aus, diese sind mangels fester, strukturgebender Komponenten in ihrem Körperbau in der Lage, ihre Form fast beliebig zu verändern. Amöben, Schleimpilze, Schnecken, Quallen, Kopffüßer (Sepien, Kraken). Gerade letztgenannte sind dafür bekannt, nicht nur die Farbe und Textur sondern auch die Struktur ihrer Haut willkürlich ändern zu können. Nun stellt sich die Frage, wie man diese Eigenschaften technisch umsetzen könnte.

Bei den Kopffüßern dürfte wahrscheinlich eine Matrix von Ringmuskeln in Zusammenarbeit mit ihrer unglaublich dehnbaren Haut zum Einsatz kommen. Ist jedoch technisch auch wieder kaum umsetzbar.

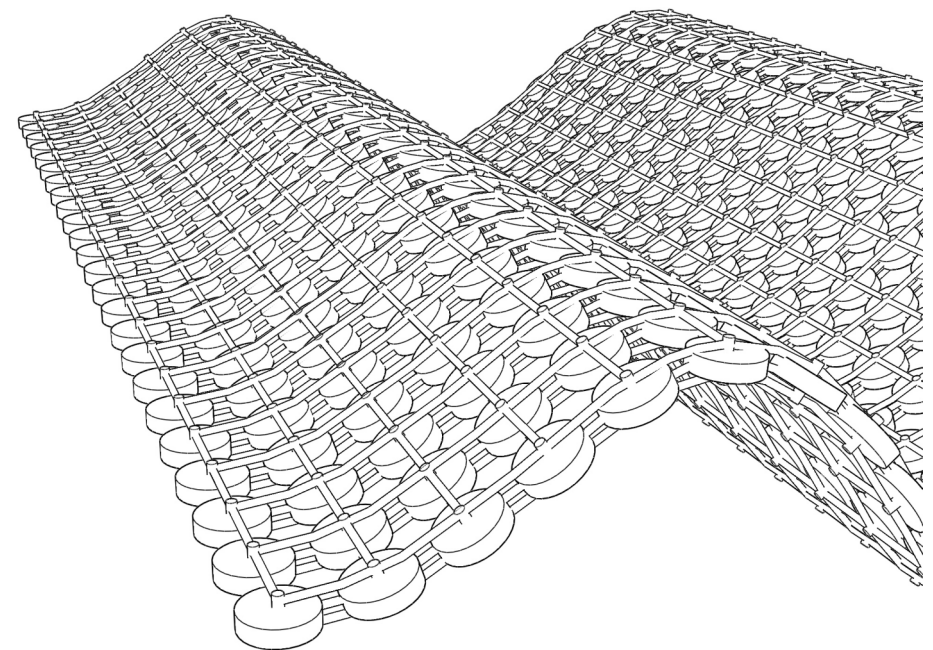
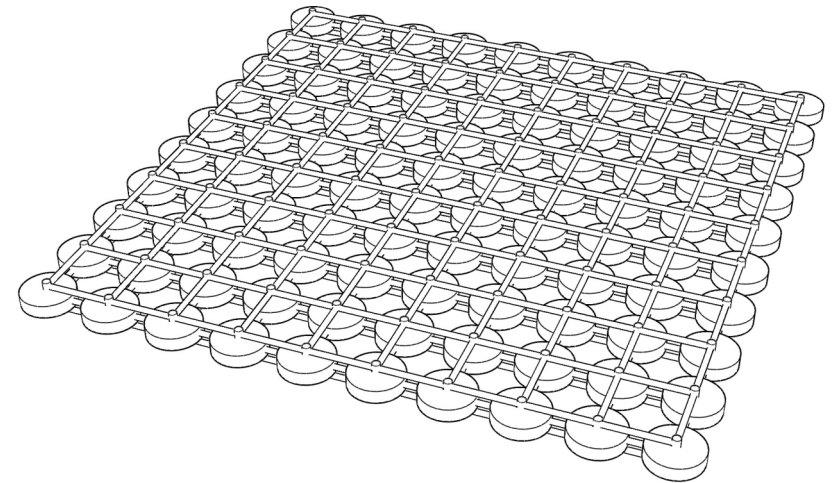
Die technische Umsetzung müsste aus der Informatik bekannten Prinzipien ähneln, z.B. einem zellulären Automaten.

-> http://de.wikipedia.org/wiki/Zellul%C3%A4rer_Automat

Ein Modul besitzt einen gewissen Zustand und manipuliert dadurch die Zustände seiner Nachbarmodule.

In diesem Falle würde es bedeuten, dass die Aktoren eines Moduls angesteuert würden und dadurch die Nachbarmodule des angesteuerten Moduls manipulieren würden.

Dies würde einen komplett neuen Typus in Form einer beweglichen Matrix, einer dynamischen Fläche aus simplen Einzelmodulen darstellen.



Die Fortbewegung entsteht durch die Imitation der Fortbewegungsweise von Schlangen, Schnecken, Rochen, Raupen oder Würmern, d.h. dadurch, den gesamten Körper in Wellenbewegungen zu versetzen, was eine Bewegung in unterschiedlichster Umgebung, auf festem Untergrund, auf instabilem Untergrund (Eis, Sümpfe, Treibsand, etc.), im Wasser, eventuell sogar in der Luft ermöglicht.

Einsatzgebiet:

Erkundung, d.h. autonome Fortbewegung in gefährlicher / unzugänglicher Umgebung, als reines Fortbewegungsmodul in anderen Konfigurationen

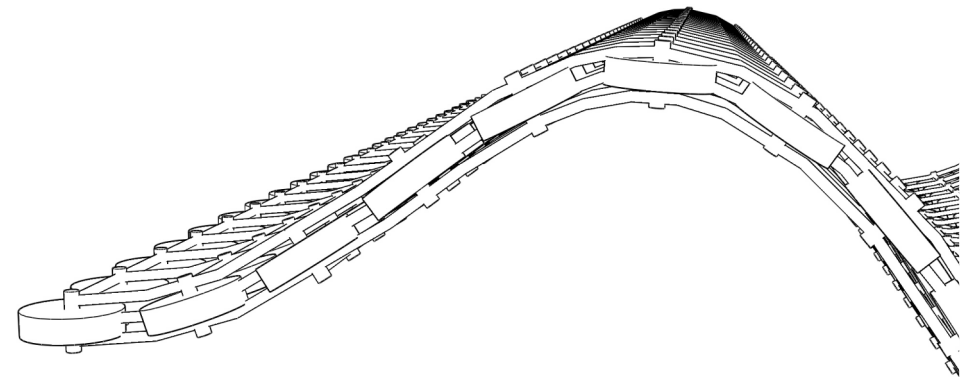
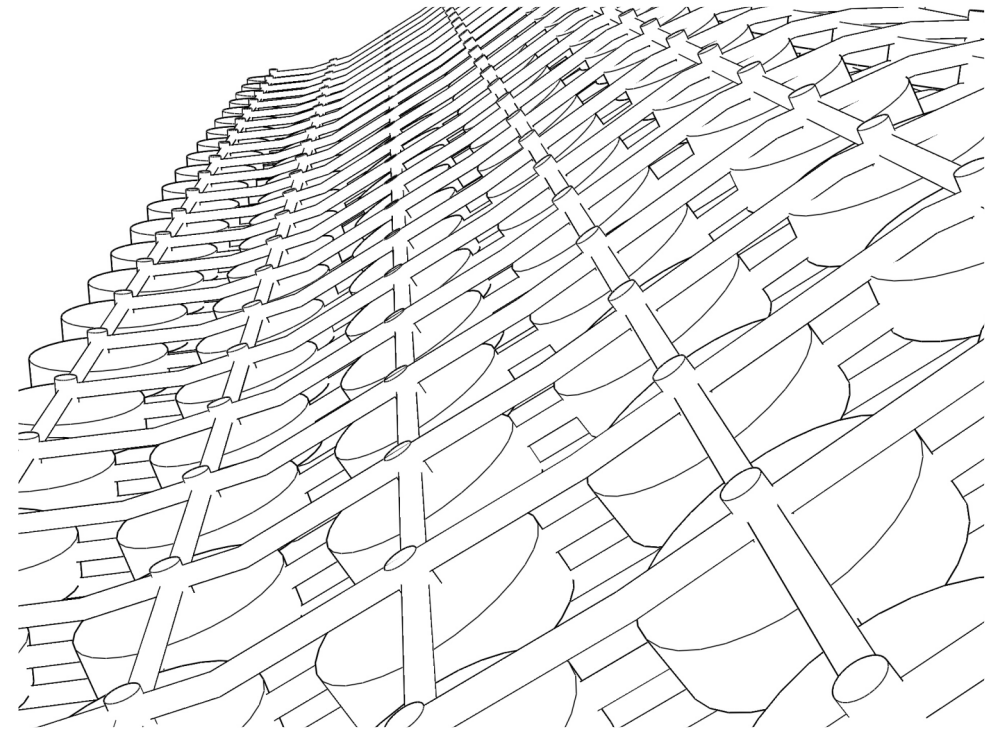
Darüber hinaus überall einsetzbar, wo Vorteile darin bestehen könnten, wenn eine Fläche ihre Form dynamisch verändern kann. Beispiele dafür wären die dynamische Anpassung an Strömungszustände bei Schiffsrümpfen, Rotoren (Windkraft) oder Segeln.

Vorteile:

Frei skalierbares System aus unkomplizierten Einzelelementen, daher gute Chancen für eine Realisierung eines Funktionsmodells / Prototypen.

Eine weitere Idee war, die Matrix ihre bestgeeignete Bewegungsweise anhand evolutionärer Algorithmen "selbst erlernen" zu lassen. Wie Karl Sims Kreaturen im Film "creatures" auf der CD.

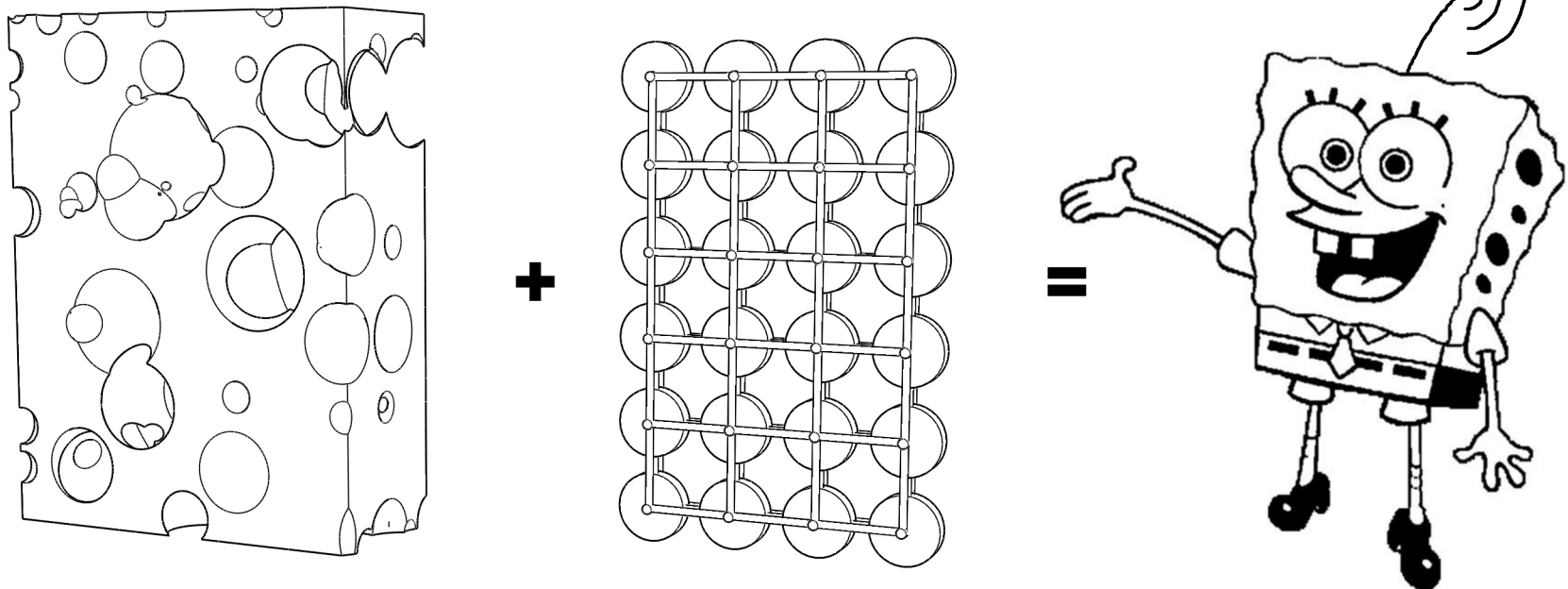
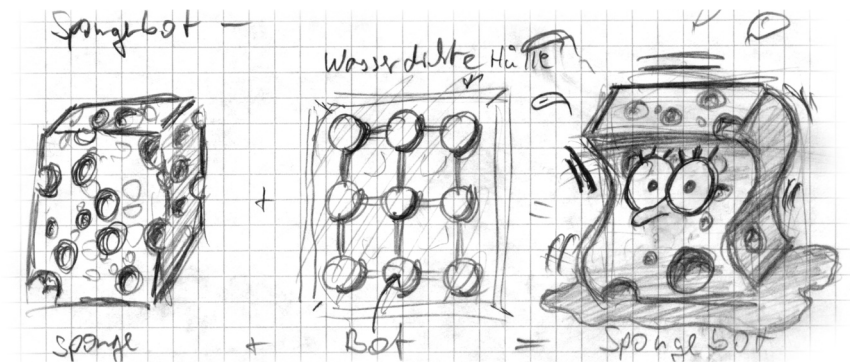
Pfad: CD/ Ressourcen Robot Dreams/ Multimedia /creatures.mpg



Der Spongebob

Durch die Aktivität in der Gruppe der "DrawBots", also malende/zeichnende Roboter entstand ferner die Idee, solch eine Matrix malen oder zeichnen zu lassen.

Dazu würde eine kleinere Matrix beispielsweise luft- und wasserdicht eingeschweisst und dann in einen Schwamm oder ähnlich saugfähiges bzw flüssigkeitsspeicherndes Material gesteckt. Dieses müsste in Farbe getränkt werden und könnte so auf einer Fläche anhand unterschiedlicher (Fort-)Bewegungen künstlerisch aktiv werden.



Phase 3: Umsetzung

Technische Überlegungen

Realisierung einer "echten" Skalierbarkeit

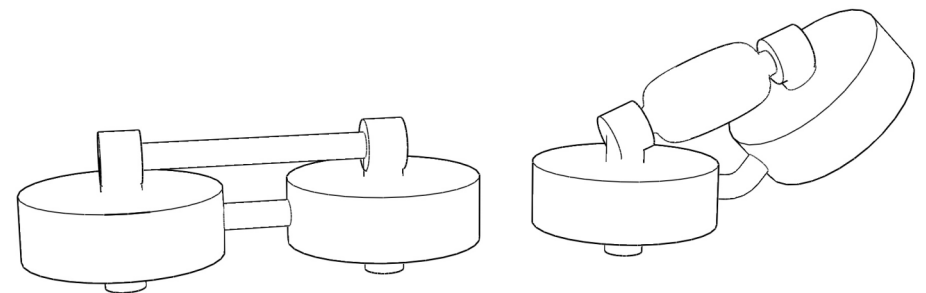
Jedes Modul kann zwei Nachbarmodule manipulieren. Sein Nachbar modul in "X-Richtung" und sein Nachbar modul in "Y-Richtung". Dadurch ergibt sich die Möglichkeit sowohl einer symmetrischen Skalierung (4,9,16,25,36,49,64, ..., 10.000, 100.000. usw. Module), als auch einer asymmetrischen Skalierung (2x1, 2x4, 3x18, 25x2, 100x1000, usw. Module).

Aufbau eines Moduls

Jedes Modul braucht zwei oder vier lineare Aktoren (Z.B. "Fluidic Muscles der Festo AG). Jedes Modul ist darüber hinaus mit einer flexiblen Verbindung mit seinem X-Nachbarn und seinem Y-Nachbarn verbunden.

Version mit zwei linearen Aktoren

X(Y)-Aktor kann X(Y)-Nachbarmodul anziehen oder wegdrücken. Der Aktor liegt dabei in Z-Richtung parallel zur flexiblen Verbindung. Durch Ausüben von Druck oder Schub und der Funktion der flexiblen Verbindung als Gelenk, wird das Nachbarmodul relativ zum agierenden Modul auf-/bzw. abwärts bewegt und gedreht (Bild).



Version mit vier linearen Aktoren

X1-Aktor übt (oben) Druck auf X-Nachbarmodul aus, während X2-Aktor (unten) Zug auf X-Nachbarmodul ausübt. Durch die flexible Verbindung zwischen den beiden Aktoren, die als Gelenk fungiert, wird das Nachbarmodul ebenfalls relativ zum agierenden Modul auf-/bzw. abwärts bewegt und gedreht. In Y Richtung gleiche Funktionsweise.



Alternativer Aufbau eines Moduls für ein Funktionsmodell

Da lineare Aktoren und eine entsprechende Ansteuerung nicht problemlos zu erwerben und einzusetzen sind, kam zunächst die Idee auf, Schrittmotoren einzusetzen. Problem dabei war jedoch, dass entsprechende NC-Steuerungen, die ein programmiertes Ansteuern der Motoren über den PC ermöglichen sollten, 1. extrem teuer sind und 2. selten über eine Möglichkeit verfügten mehr als 3 Motoren (X-, Y- und Z-Achse) anzusteuern.



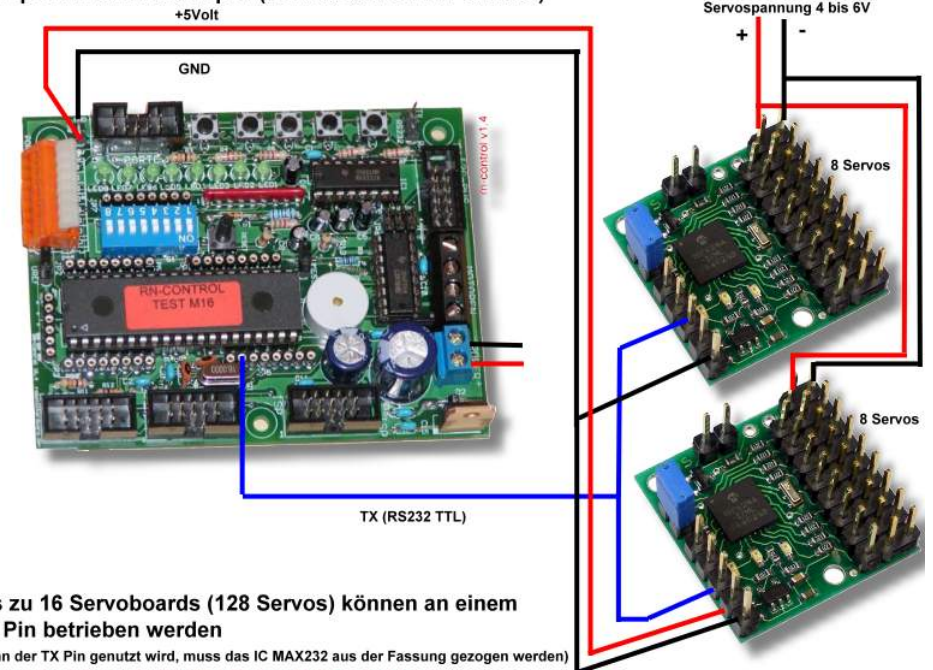
foto:robotikhardware.de

Bei der Recherche nach Alternativen stieß ich auf das sog. MicroServoBoard, ein etwa 3x3cm grosses Controllerboard, welches 8 Servomotoren (bzw. mehrere Boards in Reihe geschaltet, bis zu 128 Servomotoren) ansteuern kann. Die Bedienung bzw. Programmierung erfolgt über einer auf Basic basierenden Schnittstelle für den PC. Infos auf CD: Pfad: CD/ Ressourcen Robot Dreams/ Technik/ microseriellservo.pdf

Ich habe zunächst zum Ausprobieren Komponenten für ein 2x2(4) Module umfassendes Funktionsmodell bestellt. Vier Servos, ein MicroServoBoard, ein Kabel für die Ansteuerung des Boards über die Serielle Schnittstelle des Computers, die entsprechende Software, ein Netzteil für die Stromversorgung der Komponenten, sowie einige Kabel und Adapter.

Bilder (v.l.n.r): Servo, RS232-Kabel und Adapter (Serielles Kabel), Netzteil, MicroServoBoard, zwei MicroServoBoards hintereinandergeschaltet an einem Controllerboard

Beispiel Anschlussbeispiel (Servoboard an RN-Control)



Bis zu 16 Servoboards (128 Servos) können an einem TX Pin betrieben werden

(wenn der TX Pin genutzt wird, muss das IC MAX232 aus der Fassung gezogen werden)

Servos nicht werden des Betriebes anschließen oder Stecker ziehen!

foto:robotikhardware.de

Phase 4: Simulation

Vorgehensweise:

Damit eine Matrix von $X \times X$ Modulen sich auch tatsächlich fortbewegen kann, war zunächst eine Analyse der Systematik der Bewegungsabläufe nötig. Schließlich müssen (Zahlen-)Werte vorhanden sein, um die einzelnen Module und deren einzelne Motoren so anzusteuern, dass sich ein zusammenhängender Bewegungsablauf ergibt. Dafür wurde zunächst ein Modell und eine entsprechende (virtuelle) Umgebung benötigt. Da ich bereits für andere Projekte Skripte für die CAD-Software Rhinoceros 3.0 programmiert hatte, war dies auch bei diesem Vorhaben meine erste Wahl.

Virtuelles Modell:

Eine Matrix bestehend aus $3 \times 3(9)$ würfelförmigen Modulen. Die flexible Verbindung zwischen den Modulen wird durch Punkte zwischen den Würfeln simuliert, die Drehpunkte in X-, Y-, und diagonaler Richtung für jeden einzelnen Würfel relativ zu seinen Nachbarwürfeln definieren.

Die Funktionsweise der Aktoren wurde durch Rotationsbefehle ersetzt. Zur Veranschaulichung: Wird ein Würfel um einen Drehpunkt auf der Hälfte der Strecke zu seinem Nachbarwürfel um z.B. 45° nach oben gedreht, ist seine Bewegung relativ zu seinem Nachbarwürfel die gleiche, als würde er durch einen Akteur auf der Oberseite angezogen.

Wird ein Würfel um eine Achse zwischen seinen beiden

Nachbarwürfeln (X,Y) um z.B. 45° nach oben gedreht, ist seine Bewegung relativ zu seinen Nachbarwürfeln die gleiche, als würden beide Aktoren auf der Oberseite anziehen. Hört sich kompliziert an, ist es aber nicht.

Programmierung:

Neun Würfel in einer 3×3 Anordnung werden erstellt und eindeutig benannt (z.B. Wuerfel_1 bis Wuerfel_9). Jeder der neun Würfel bekommt vier Hilfspunkte zugewiesen, die sich jeweils auf der Hälfte der Strecke zu seinen Nachbarwürfeln befinden (solange noch kein Modul bewegt wurde). Diese Punkte werden ebenfalls eindeutig benannt (z.B. Wuerfel_1_nord, Wuerfel_1_sued, ..._ost, ..._west). Aus diesen Punkten lassen sich später die Rotationsachsen ableiten.

Ein Würfel und seine vier Hilfspunkte bilden ein Modul, sie sind "fest" miteinander verbunden, so dass sich die Punkte mit dem Würfel mitbewegen.

Über die Namen der Würfel können nun einzelne oder mehrere Würfel mitsamt ihrer zugehörigen Punkte ausgewählt werden. Ebenso können die einzelnen Punkte jedes Würfels ausgewählt werden, bzw. auch ihre Koordinaten im Raum abgefragt werden. Über die Punkte und deren Koordinaten ergeben sich für jeden einzelnen Würfel acht unterschiedliche Rotationsachsen relativ zu seinen Nachbarwürfeln.

Erst durch diese Ausgangskonfiguration ist es möglich, den realen, physikalischen Aufbau des Modells zu simulieren. Im fertigen Programm sieht die eine Bewegungsanweisung dann ungefähr so aus:

```
wu6  
wu8  
wu9  
angle=45  
strpoint1="w7s"  
strpoint2="w3o"  
selrotate
```

Die Würfel 6, 8 und 9 werden ausgewählt und um die Achse der Koordinaten der Punkte Würfel 7_süd zu Würfel 3_ost um 45° gedreht.

Durch eine Abfolge derartiger Anweisungen lassen sich nun auch komplexere Bewegungsabläufe des Modells simulieren.

Die erhaltenen bzw. erprobten Werte dieser Bewegungsabläufe lassen sich nun dazu verwenden (bzw. konvertieren), um Bewegungsabläufe am realen Modell zu testen.

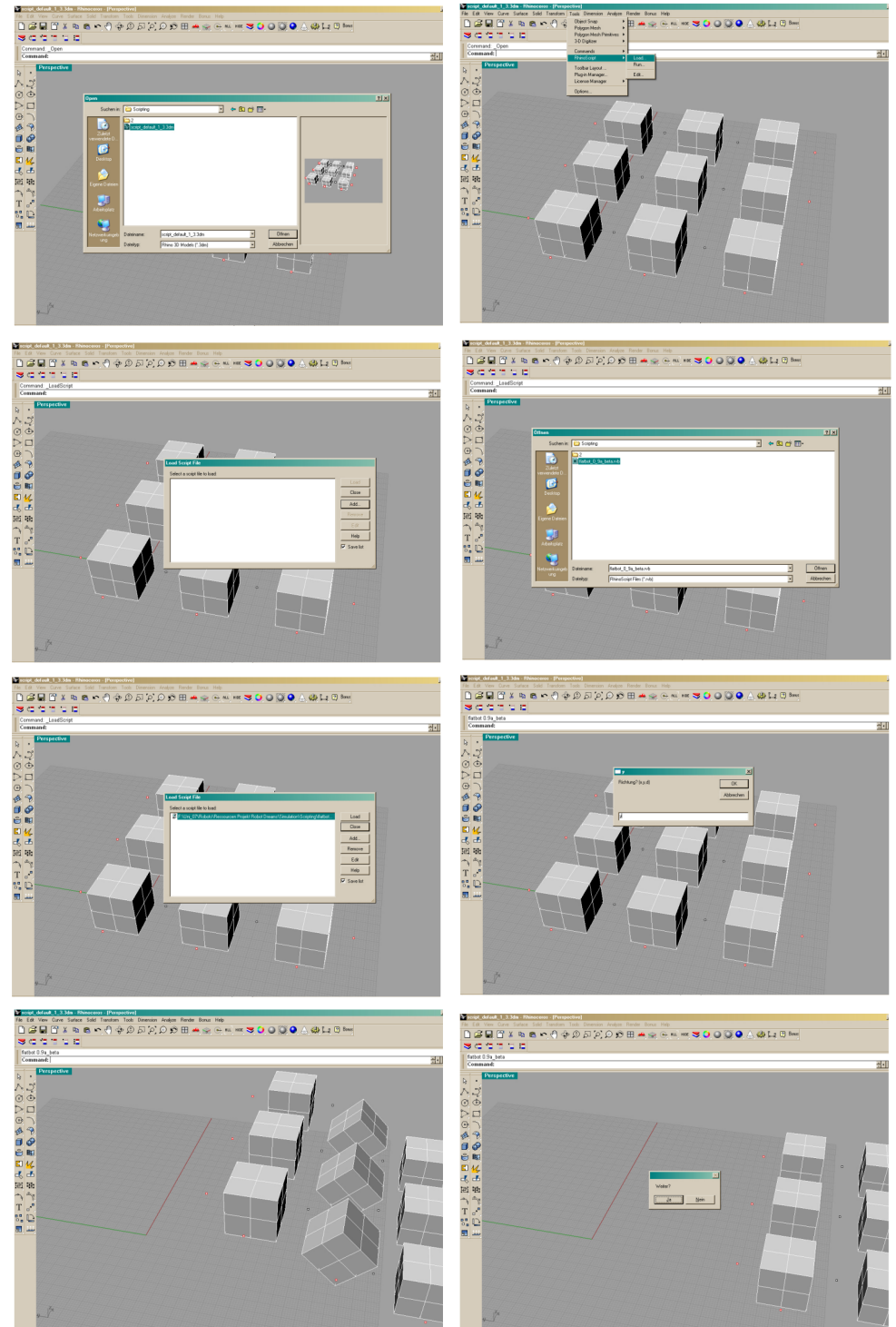
Im Appendix finden Sie den Quellcode für das Programm. Weiterhin befindet sich auf der beigelegten CD eine in Rhino 3.0 ausführbare Scriptdatei "flatbot_0_9a_beta.rvb".

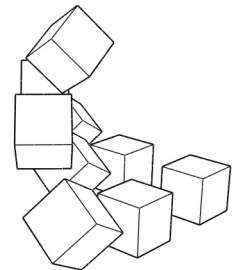
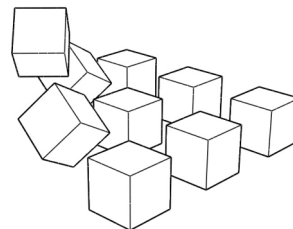
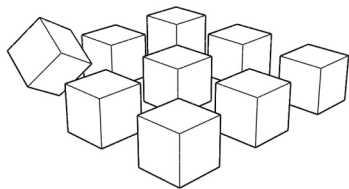
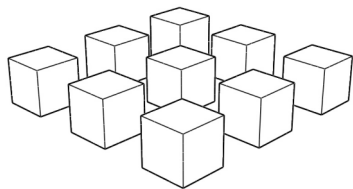
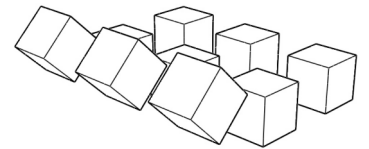
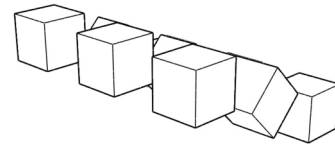
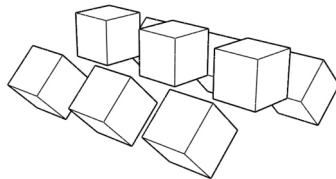
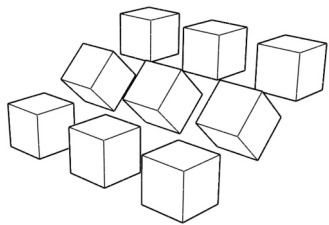
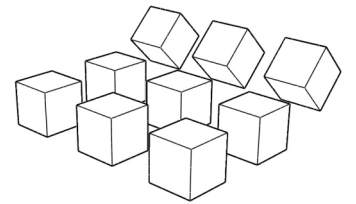
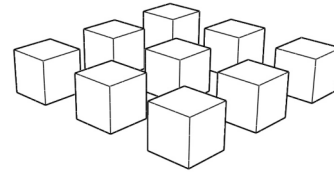
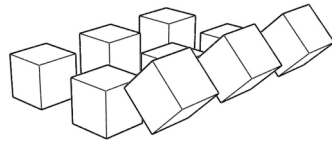
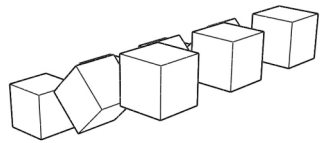
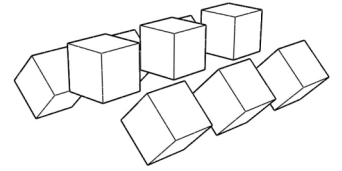
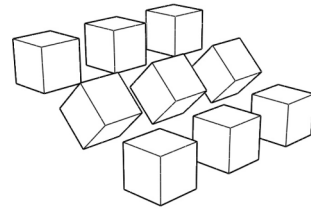
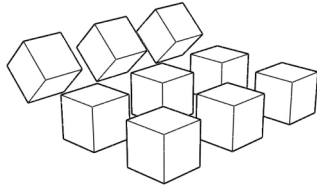
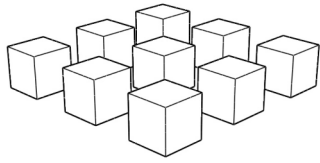
Pfad: CD/Ressourcen Projekt Robot Dreams/Simulation/Scripting

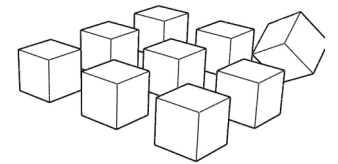
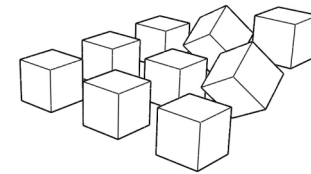
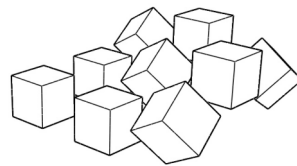
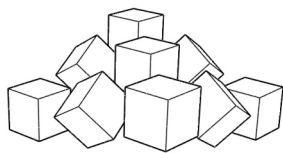
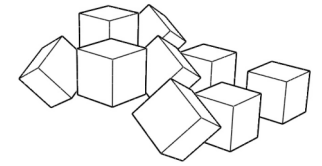
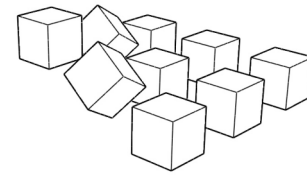
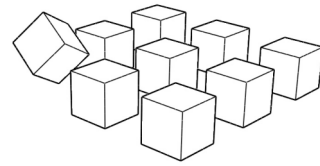
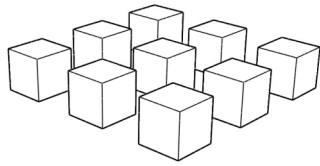
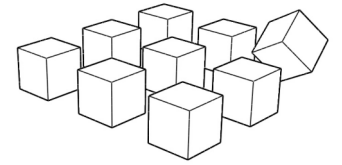
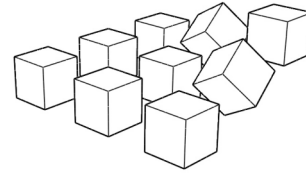
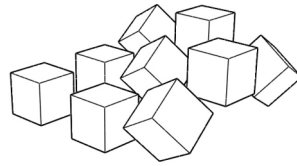
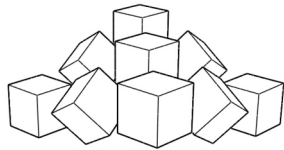
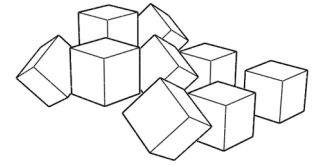
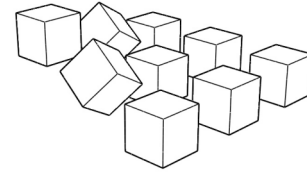
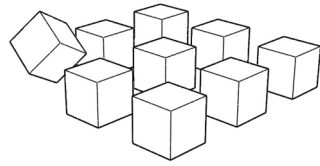
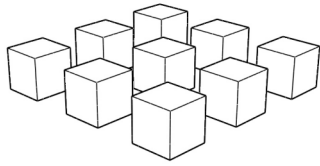
Um die Simulation auszuführen, muss zuerst die Rhino 3.0 Datei "script_default_1_3.3dm" im gleichen Ordner geöffnet werden, anschließend im Menü "Tools/RhinoScript/Load.../[Add...]" die Datei "flatbot_0_9a_beta.rvb" geladen werden. Diese lässt sich dann unter Menü "Tools/RhinoScript/Load.../" per Doppelklick oder Auswählen und Klick auf "[Load]" starten.

Es folgt eine Eingabeaufforderung, die nach der Bewegungsrichtung fragt (x,y,d), d.h. ein Bewegungsablauf des Modells in X-Richtung, Y-Richtung, oder diagonal. Danach kommt eine weitere Eingabeaufforderung, in der eingegeben werden kann, wieviele Schritte sich das Modell in die zuvor ausgewählte Richtung bewegen soll, bevor erneut nach Eingabewerten gefragt wird.

Auf den nächsten beiden Seiten sehen Sie eine Abfolge von Bewegungen, die mit dieser Simulation erzeugt wurden.







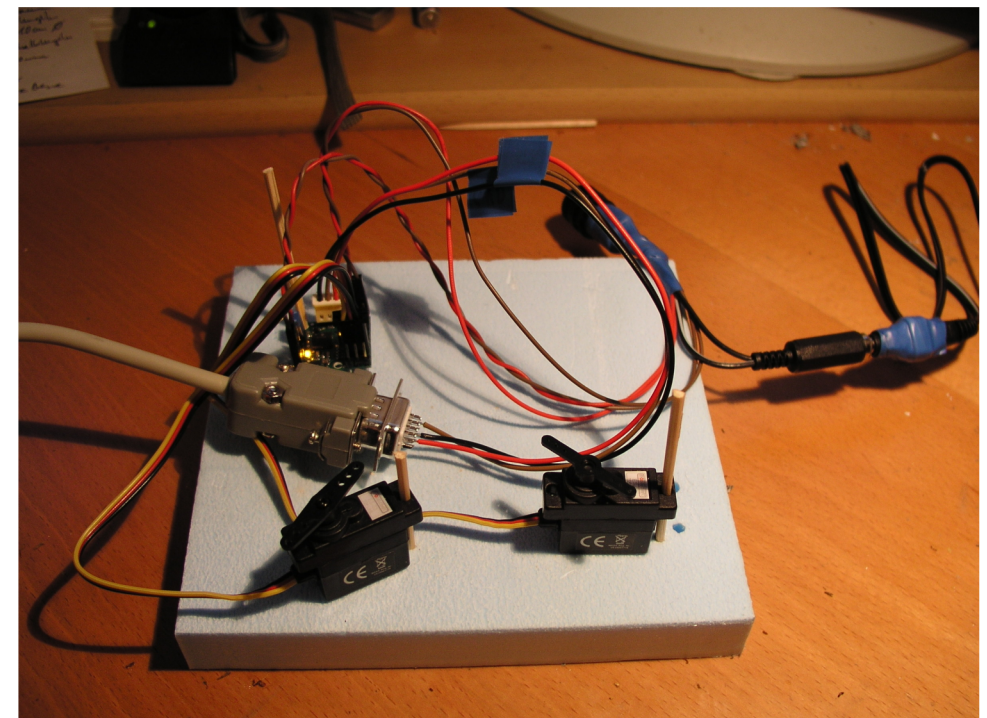
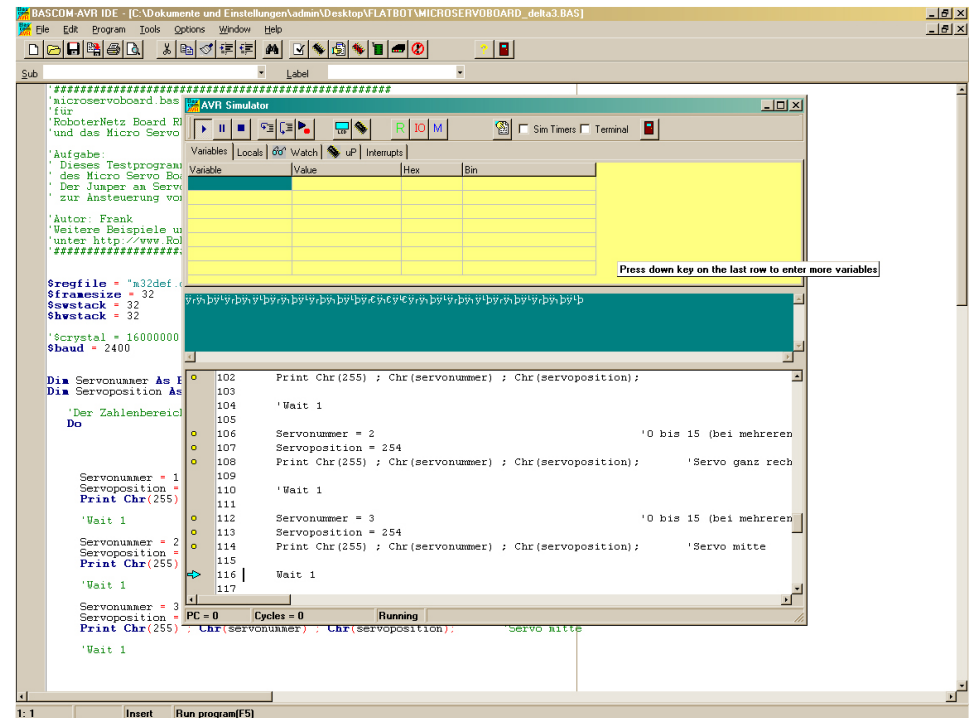
Phase 5: Funktionsmodell

Nach dem Eingang der bestellten Bauteile begann die Phase des Experimentierens am realen Modell. Zunächst einmal musste ein Versuchsaufbau realisiert werden, um die Funktion des PC Interfaces (Bild oben) zu testen. Im unteren Bild sind das MicroServoBoard, zwei daran angeschlossene Servos, die Stromversorgung (5V, 1,4A über ein stabilisiertes Netzteil) und der Anschluss an die RS232-Schnittstelle (Serieller Port) des PC zu sehen.

Das MicroServoBoard wird vom PC ähnlich angesprochen, wie ein Modem. Da es sich allerdings als etwas intolerant gegenüber höheren Übertragungsgeschwindigkeiten erwies und offenbar immer eine kleine "Aufwärmphase" benötigte, um sich mit den vom PC erhaltenen Daten zu synchronisieren, wurde es mit der niedrigsten wählbaren Übertragungsgeschwindigkeit von 2600 Baud angesprochen.

Nachdem ich den angeschlossenen Servos lange Zeit nicht mehr als ein kurzes Zucken entlocken konnte, folgte ein längere Phase Internet-Recherche, wobei sich herausstellte, dass das MicroServoBoard anscheinend für seine Macken bekannt ist und viele Faktoren berücksichtigt werden müssen, um eine halbwegs zuverlässige Funktion sicherzustellen.

Nach einigen Stunden Feinschliff an den Pausen zwischen zwei übermittelten Befehlen und Anpassung der Übertragungsgeschwindigkeit war ich schließlich soweit, dass ich zuverlässig zwei Servos hin und her bewegen konnte.



Ein kleines Video davon ist auf der beigelegten CD zu sehen:

Pfad: CD/ Ressourcen Projekt Robot Dreams/ Multimedia/
Funktionsmodelle/ Es tut sich was.MOV

Anschliessend folgten weitere Versuche mit mehr Servos. Nachdem dies alles einigermaßen funktionierte, begann ich mir Gedanken zu machen, wie ich mit den vorhandenen Mitteln Module erstellen konnte, die in etwa mit der von mir vorher erdachten Bauweise korrespondieren würden.

Das grösste Problem war dabei, die auf 180° beschränkte Drehbewegung der Servos in eine lineare Bewegung umzuwandeln. Hätten die Servos einen grösseren Drehbereich, hätte man dies über eine Winde lösen können, ein Modell was das Prinzip verdeutlichen sollte, hatte ich einmal während des Seminars vorgestellt (leider ist dieses irgendwo in den Tiefen der Werkstatt verschollen).

Blieb noch die Übersetzung mittels Zug- / Druckstangen, allerdings besaßen die Servos auch dafür einen denkbar ungeeigneten Aufbau.

Ein erster Versuch dazu ist als Video auf der CD zu sehen:

Pfad: CD/ Ressourcen Projekt Robot Dreams/ Multimedia/
Funktionsmodelle/ Funktionsmodell grosse Module.MOV

Reichlich klobig zwar, belegt aber die Funktionsweise schon recht gut.

Der nächste Schritt war nun, die Modulgrösse soweit zu verringern, dass sie im Prinzip (fast) nur noch aus den benötigten Teilen, sprich den Servos bestand.

Pfad: CD/ Ressourcen Projekt Robot Dreams/ Multimedia/
Funktionsmodelle/ erstes Funktionsmodell.MOV

Dies belegt die Plausibilität des Konzepts, zeigt jedoch auch die Einschränkungen durch die eingesetzten Mittel auf.

Für die nächstgrössere Anordnung wären ein weiteres MicroServoBoard, weitere zwölf Servos und eine stärkere Stromversorgung nötig gewesen.

Weiterhin hat sich gezeigt, dass das MicroServoBoard auch nur bedingt für diese Zwecke geeignet ist. Es ist zwar grundsätzlich in der Lage, mehrere Servos simultan zu bewegen, jedoch hat es sich als recht fehleranfällig erwiesen und ist darüber hinaus mit höheren Übertragungsgeschwindigkeiten, die für eine zufriedenstellende Ansteuerung einer grösseren Matrix nötig gewesen wären, überfordert.

Ein Beispielprogramm für BasCom, welches über das MicroServoBoard mehrere Servos ansteuert ist im Appendix zu finden.

Phase 6: Weitere Aktivitäten

Arduino Workshop

Nachdem ich durch meine Experimente und die Erfahrung, über Programmierung am PC in der realen, physikalischen Welt Objekte manipulieren zu können, "Blut geleckt" hatte, nahm ich noch am "Arduino Workshop - Einführung ins Physical Computing" von Olaf Val (Neue Medien) teil.

Ich erhoffte mir davon auch, neue Möglichkeiten zu finden, ein (besser) funktionierendes Funktionsmodell realisieren zu können. Im Laufe des Workshops lernten wir den grundsätzlichen Umgang mit den I/O (Eingabe/Ausgabe) -Möglichkeiten eines an einen Computer angeschlossenen MicroControllers.

Die Aufgaben rangierten vom Ein-/Ausschalten einer LED, über die Nutzung von Transistoren um einfache Schaltungen zu realisieren, über Relaisschaltungen, die einem das Ein/Ausschalten eines 220V-Verbrauchers über einen Tastendruck am PC ermöglichen, bis hin zu einfachen Eingabegeräten.

Prinzipiell ließe sich meine Versuchsanordnung auch über ein Arduino-Board (Bild) steuern, allerdings mit einer komplett anderen Vorgehensweise und es hätte bedeutet, noch eine neue Programmiersprache zu erlernen.

Leider war der Workshop auch erst im 2. Semester des Projekts, hätte sonst meines Erachtens nach zum Pflichtprogramm für die Projektteilnehmer gehört, da es genau den "Missing Link" zwischen "uns Designern" und unseren Kooperationspartnern aus der

Informatik bildete.

Weblinks:

www.arduino.cc

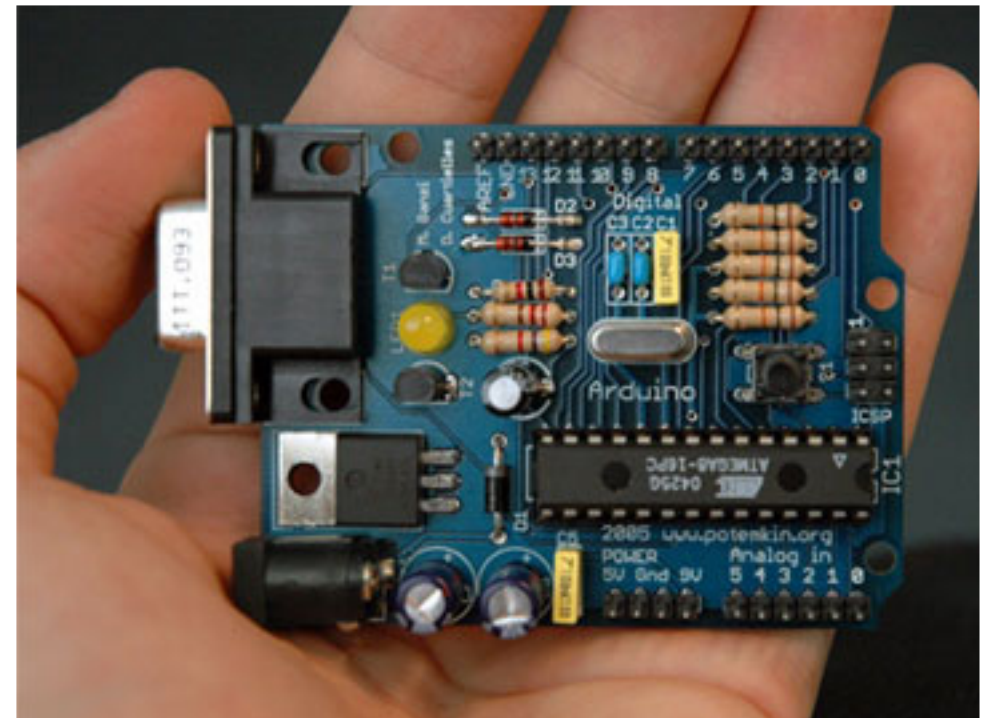
www.olafval.de

HannoverMesse 07

Während eines Messeauftritts für ein anderes Projekt, hatte ich Zeit, mir etliche Robotersysteme aus der Nähe zu betrachten. Direkt gegenüber vom Stand der Uni Kassel fuhr beispielsweise ein autonom agierender Roboter herum, der von seinen Spezifikationen her die perfekte Chassis für Annikas Litfaßsäulen-Bot gebildet hätte.

Weitere Impressionen auf der CD:

Pfad: CD/ Ressourcen Projekt Robot Dreams/Multimedia/Roboter HannoverMesse07



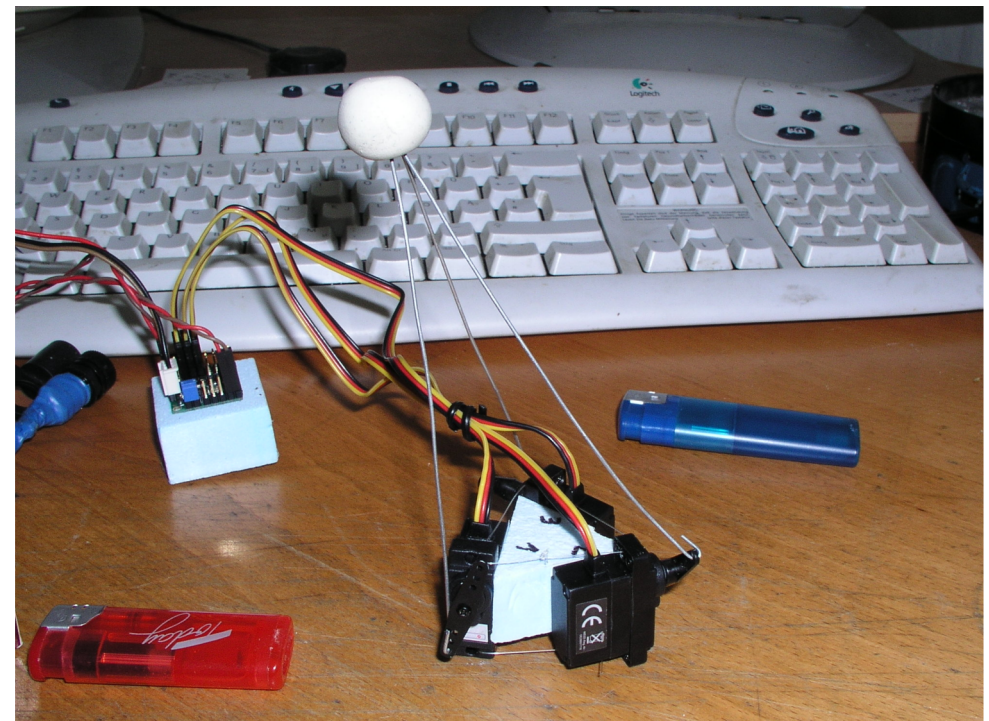
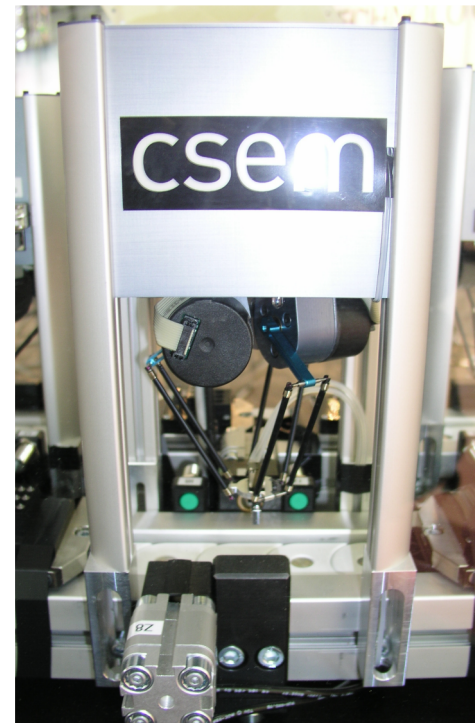
Den meisten Eindruck haben allerdings die sogenannten Delta-Roboter bei mir hinterlassen. Das Konzept, mit nur 3 Motoren schnelle und präzise Bewegungen in X,Y und Z Richtung zu ermöglichen, fand ich ausgesprochen interessant.

Pfade: CD/ Ressourcen Projekt Robot Dreams/ Multimedia/ Roboter HannoverMesse07/ CSEM
CD/ Ressourcen Projekt Robot Dreams/ Multimedia/ Roboter HannoverMesse07/ Delta-Roboter HannoverMesse.MOV

Im weiteren Verlauf kam es auch zu einem interessanten Gespräch mit Herrn Sébastien Perroud von der CSEM SA, die mit Ihrem kleinsten Delta-Robotern der Welt soeben den ersten Platz beim Swiss Technology Award erreicht haben.

Im Endeffekt hat mir das solange keine Ruhe gelassen, bis ich mit meinen vorhandenen Mitteln so ein Teil nachgebaut habe. Filme dazu auf der CD.

Pfade: CD/ Ressourcen Projekt Robot Dreams/ Multimedia/ Roboter HannoverMesse07/ Nachbau Delta-Roboter 1(-3).MOV



Fazit

Auch wenn im Endeffekt kein "fertiges Produkt" entstanden ist, haben meine Arbeiten die grundsätzliche Machbarkeit meines Konzepts belegt.

Allerdings wurden mir auch meine Grenzen aufgezeigt, einen tatsächlich funktionierenden und steuerbaren Prototypen konnte ich allein und mit meinen Mitteln in zwei (bzw. einem) Semester(n) unmöglich realisieren.

Dafür wären jeweils ein bis zwei Studenten nötig gewesen um sowohl den mechanischen/elektronischen Aufbau, als auch die Programmierung/Ansteuerung zu realisieren, weiterhin wäre mindestens das drei- bis vierfache an finanziellen Mitteln nötig gewesen.

Jedoch bin ich grundsätzlich immer noch an einer Weiterentwicklung interessiert, in Kooperation mit einer Firma, wie z.B. Festo, die über die nötigen Mittel, das nötige Know-How und entsprechendes Personal verfügt, wäre das durchaus machbar.

Darüber hinaus hat mir das Projekt wirklich grossen Spass gemacht und ich habe viel dazugelernt. Zum Beispiel, dass der Schritt von der virtuellen zur realen Welt kein grosser ist. Und dass dem zugrundeliegende Vorgänge nicht unbedingt kompliziert sein müssen.

Auf jeden Fall hat es mich in meinem Vorhaben bestärkt, mir ein Berufsfeld etwas abseits der "reinen Gestaltung von Produkten" hin zu eher systematisch / technischen Problemen zu suchen. Bzw. zu

versuchen in der "Grauzone" zwischen diesen Bereichen tätig zu werden.

Meiner bisherigen Erfahrung nach mangelt es immer noch massiv an einer "echten" Interdisziplinarität, auch wenn diese überall propagiert wird.

Weiterhin fand ich es etwas schade, dass das ursprünglich auf zwei Semester angelegte Projekt Seminar, nach dem ersten Semester offiziell mehr oder weniger eingestellt wurde. Meiner Meinung nach sind dabei etliche sehr interessante Konzepte entstanden, die, auch wenn schlussendlich mit gegebenen Mitteln nicht realisierbar, doch noch etwas weiter verfolgt hätten werden sollen.

Leider ergaben sich für dieses Projekt interessante Veranstaltungen, wie der Arduino Workshop oder die Hannover-Messe auch erst im zweiten Semester.

Appendix

Quellcode des Simulationsscripts für Rhino 3.0 (VBScript):

```
'
' RobotDreams
'
' Author: Ulf Cadenbach
' Year: 2007
'
' Subroutines:
' Purpose: Making a pattern (this version 3x3 boxes)
' move in various ways
' Specifications: VBScript, to be used in Rhinoceros 3.0
' This script:
'
Option Explicit

' Global_Variables

Dim arrObjects, strName, strobject, angle, strpoint1, strpoint2, count, help, ans, direction, schritte

Redim arobjects (100)

Dim Axis
Redim Axis (1)

Dim Ax1
Redim Ax1 (2)
Ax1 (0)=0
Ax1 (1)=0
Ax1 (2)=0

Dim Ax2
Redim Ax2 (2)
Ax2 (0)=10
Ax2 (1)=10
```

```
Axis (0)= Ax1
Axis (1)= Ax2

Dim Point
Redim Point(2)
Point(0)=0
Point(1)=0
Point(2)=0

' Main_Subroutines
'*****
'steuerung
Sub GetValues
dim m,n
direction=Inputbox("Richtung? (x,y,d)",direction)
schritte=Inputbox("Anzahl Schritte?",schritte)
If direction = "x" then
For m=0 to schritte
welle_west
Next
Elseif direction = "y" then
For m=0 to schritte
welle_sued
Next
Elseif direction = "d" then
For m=0 to schritte
Welle_diag
Next
Else MsgBox("Eingabe nicht eindeutig")
GetValues
End If

ans = MsgBox("Weiter?",VbYesNo)

If ans=VbYes then

GetValues

Else
MsgBox("Abbruch durch Nutzer")
```

Ax2 (2)=0

```
End Sub

'Objekte nach Namen auswählen
Function Obj_select (arobjects)
Dim obj_string, item, counter, arobjects2
Redim arobjects2(100)
counter=0

For Each item In arobjects

If Vartype (arobjects(counter)) = VBString

Then
obj_string =
Rhino.ObjectsByName(arobjects(counter), vbFalse)
Redim preserve arobjects2(counter)
arobjects2(counter)=obj_string(0)
counter=counter+1
End if

Next

Obj_select=arobjects2

End Function

'ausgewählte Objekte rotieren
Function Obj_rotate (arobjects,angle,axispoints)

Rhino.RotateObjects arobjects, axis(0) , angle, axispoints ,vbFalse

End Function

'3D-Koordinaten für die Drehpunkte
Function axispoints (strpoint1, strpoint2)

dim arrpoint1, arrpoint2, strpoints

redim strpoints(1)

strpoint1=Rhino.ObjectsByName(strpoint1, vbFalse)

arrPoint1=
Rhino.PointCoordinates(strpoint1(0))

If IsArray(arrPoint1) Then

Axis (0)= arrpoint1

'rhino.printex "strpoint1 ok"

End If

strpoint2=Rhino.ObjectsByName(strpoint2, vbFalse)
```

End If

```
arrPoint2=
Rhino.PointCoordinates(strpoint2(0))

If IsArray(arrPoint2) Then

Axis (1)= arrpoint2

'rhino.printex "strpoint2 ok"

End If

axispoints=Axis

End Function

'*****
'Würfel und zugehörige Punkte markieren
Sub wu1
Arobjects(help)="w1"
help=help+1
Arobjects(help)="w1w"
help=help+1
Arobjects(help)="w1n"
help=help+1
Arobjects(help)="w1o"
help=help+1
Arobjects(help)="w1s"
help=help+1
End Sub

Sub wu2
Arobjects(help)="w2"
help=help+1
Arobjects(help)="w2w"
help=help+1
Arobjects(help)="w2n"
help=help+1
Arobjects(help)="w2o"
help=help+1
Arobjects(help)="w2s"
help=help+1
End Sub

Sub wu3
Arobjects(help)="w3"
help=help+1
Arobjects(help)="w3w"
help=help+1
Arobjects(help)="w3n"
help=help+1
Arobjects(help)="w3o"
help=help+1
Arobjects(help)="w3s"
help=help+1
End Sub

Sub wu4
Arobjects(help)="w4"
help=help+1
Arobjects(help)="w4w"
help=help+1
Arobjects(help)="w4n"
help=help+1
```

```

Arrobjects(help)="w4o"
help=help+1
Arrobjects(help)="w4s"
help=help+1
End Sub

Sub wu5
Arrobjects(help)="w5"
help=help+1
Arrobjects(help)="w5w"
help=help+1
Arrobjects(help)="w5n"
help=help+1
Arrobjects(help)="w5o"
help=help+1
Arrobjects(help)="w5s"
help=help+1
End Sub

Sub wu6
Arrobjects(help)="w6"
help=help+1
Arrobjects(help)="w6w"
help=help+1
Arrobjects(help)="w6n"
help=help+1
Arrobjects(help)="w6o"
help=help+1
Arrobjects(help)="w6s"
help=help+1
End Sub

Sub wu7
Arrobjects(help)="w7"
help=help+1
Arrobjects(help)="w7w"
help=help+1
Arrobjects(help)="w7n"
help=help+1
Arrobjects(help)="w7o"
help=help+1
Arrobjects(help)="w7s"
help=help+1
End Sub

Sub wu8
Arrobjects(help)="w8"
help=help+1
Arrobjects(help)="w8w"
help=help+1
Arrobjects(help)="w8n"
help=help+1
Arrobjects(help)="w8o"
help=help+1
Arrobjects(help)="w8s"
help=help+1
End Sub

Sub wu9
Arrobjects(help)="w9"
help=help+1
Arrobjects(help)="w9w"
help=help+1
Arrobjects(help)="w9n"
help=help+1

```

```

Arrobjects(help)="w9o"
help=help+1
Arrobjects(help)="w9s"
help=help+1
End Sub
*****
Sub selrotate
    axispoints strpoint1,strpoint2
    arrobjects=obj_select(arrobjects)
    Obj_rotate arrobjects, angle ,axis
End Sub
*****
Sub reset
    redim arrobjects(100)
    help=0
End Sub
*****
Sub pause
    Dim i,j
    j=2500000
    For i=0 to j
        Next
End sub
*****
Sub Welle_Sued
    reset
    wu1
    wu2
    wu3
    angle=45
    strpoint1="w1s"
    strpoint2="w3s"
    selrotate
    pause
    wu1
    wu2
    wu3
    angle=45
    strpoint1="w1s"
    strpoint2="w3s"
    selrotate
    wu4
    wu5
    wu6
    wu7
    wu8
    wu9
    angle=-90
    strpoint1="w4s"
    strpoint2="w6s"
    selrotate
    pause
    reset
    wu1
    wu2
    wu3
    wu4
    wu5
    wu6
    wu7
    wu8
    wu9
    angle=45
    strpoint1="w4n"
    strpoint2="w6n"
    selrotate
    reset
    wu1
    wu2
    wu3
    wu4
    wu5
    wu6
    wu7
    wu8
    wu9
    angle=45
    strpoint1="w1n"
    strpoint2="w3n"

```

```

angle=-90
strpoint1="w1s"
strpoint2="w3s"
selrotate
pause
reset
wu1
wu2
wu3
wu4
wu5
wu6
wu7
wu8
wu9
angle=-90
strpoint1="w7n"
strpoint2="w9n"
selrotate
reset
wu4
wu5
wu6
wu7
wu8
wu9
angle=45
strpoint1="w4n"
strpoint2="w6n"
selrotate
pause
reset
wu7
wu8
wu9
angle=45
strpoint1="w7n"
strpoint2="w9n"
selrotate
pause
End Sub
*****
Sub Welle_West
    reset
    wu3
    wu6
    wu9
    angle=45
    strpoint1="w3w"
    strpoint2="w9w"
    selrotate
    pause
    reset
    wu3
    wu6
    wu9
    wu2
    wu5
    wu8
    angle=45
    strpoint1="w2w"

```

```

strpoint2="w8w"
selrotate

reset
wu3
wu6
wu9
angle=-90
strpoint1="w3w"
strpoint2="w9w"
selrotate

pause

reset
wu1
wu2
wu3
wu4
wu5
wu6
wu7
wu8
wu9
angle=45
strpoint1="w1w"
strpoint2="w7w"
selrotate

reset
wu2
wu3
wu5
wu6
wu8
wu9
angle=-90
strpoint1="w2w"
strpoint2="w8w"
selrotate

pause

reset
wu1
wu2
wu4
wu5
wu7
wu8
angle=-90
strpoint1="w2o"
strpoint2="w8o"
selrotate

reset
wu1
wu2
wu3

```

```

wu4
wu5
wu6
wu7
wu8
wu9
angle=45
strpoint1="w3o"
strpoint2="w9o"
selrotate

pause

reset
wu1
wu4
wu7
angle=-90
strpoint1="w1o"
strpoint2="w7o"
selrotate

reset
wu2
wu5
wu8
wu1
wu4
wu7
angle=45
strpoint1="w2o"
strpoint2="w8o"
selrotate

pause

reset
wu1
wu4
wu7
angle=45
strpoint1="w1o"
strpoint2="w7o"
selrotate

pause

End Sub

Sub Welle_diag

reset
wu1
angle=45
strpoint1="w4w"
strpoint2="w2n"
selrotate

reset
wu1
wu2
wu4

```

```

angle=45
strpoint1="w7w"
strpoint2="w3n"
selrotate

reset
wu1
angle=-90
strpoint1="w4w"
strpoint2="w2n"
selrotate

pause

reset
wu1
wu2
wu4
wu3
wu5
wu7
angle=45
strpoint1="w7s"
strpoint2="w3o"
selrotate

reset
wu1
wu2
wu4
angle=-90
strpoint1="w7w"
strpoint2="w3n"
selrotate

pause

reset
wu3
wu5
wu7
wu6
wu8
wu9
angle=-45
strpoint1="w7w"
strpoint2="w3n"
selrotate

reset
wu1
wu2
wu4
angle=-45
strpoint1="w7w"
strpoint2="w3n"
selrotate

reset
wu1
angle=90
strpoint1="w4w"
strpoint2="w2n"

```

```

selrotate

reset
wu6
wu8
wu9

angle=-90
strpoint1="w3o"
strpoint2="w7s"
selrotate

reset
wu9

angle=-45
strpoint1="w8s"
strpoint2="w6o"
selrotate

pause

reset
wu3
wu5
wu7
wu6
wu8
wu9
angle=-90
strpoint1="w7w"
strpoint2="w3n"
selrotate

reset
wu2
wu4
wu3
wu5
wu7
wu6
wu8
wu9
angle=45
strpoint1="w4w"
strpoint2="w2n"
selrotate

reset
wu9
angle=90
strpoint1="w8s"
strpoint2="w6o"
selrotate
pause

Rhino.Printex "check"

reset
wu6
wu8

```

```
wu9
angle=90
strpoint1="w3o"
strpoint2="w7s"
selrotate
```

```
reset
wu3
wu5
wu7
wu6
wu8
wu9
angle=45
strpoint1="w7w"
strpoint2="w3n"
selrotate
```

```
reset
wu6
wu8
wu9
angle=45
strpoint1="w7s"
strpoint2="w3o"
selrotate
```

```
reset
wu9
angle=-45
strpoint1="w8s"
strpoint2="w6o"
selrotate
pause
```

```
End Sub
*****
*****
```

```
' _____ Main_Program _____
```

```
*****
*****
```

```
Rhino.Printex " "
Rhino.Printex "*Programm gestartet*"
Rhino.Printex "flatbot 0.9a_beta"
```

```
GetValues
```

```
Rhino.Printex "*Programm beendet*"
```

```
'Rhino.Printex "Debug Info Arobjects nach zuweisung"
'Rhino.Printex "Debug Arobjects(0)" & Arobjects(0)
'Rhino.Printex "Debug Arobjects(1)" & Arobjects(1)
'Rhino.Printex "Debug Arobjects(2)" & Arobjects(2)
'Rhino.Printex "Debug Arobjects(3)" & Arobjects(3)
'Rhino.Printex "Debug Arobjects(4)" & Arobjects(4)
'If isarray (arobjects) Then
```

```
'Rhino.Printex "isarray arobjects=true"
'End if
```

```
'If isarray (point) Then
'Rhino.Printex "isarray point=true"
'End if
```

```
'Rhino.Printex "Debug Info Arobjects nach
_Testselect_und_Rotate"
'Rhino.Printex "Debug2 Arobjects(0)" & Arobjects(0)
'Rhino.Printex "Debug2 Arobjects(1)" & Arobjects(1)
'Rhino.Printex "Debug2 Arobjects(2)" & Arobjects(2)
'Rhino.Printex "Debug2 Arobjects(3)" & Arobjects(3)
'Rhino.Printex "Debug2 Arobjects(4)" & Arobjects(4)
```

'Quellcode des Testprogramms zur Ansteuerung	Servonummer = 2	Servoposition = 88	Chr(servoposition);
mehrerer Servos über BasCom (Basic):	Servoposition = 0	Print Chr(255) ; Chr(servonummer) ;	
#####	Print Chr(255) ; Chr(servonummer) ;	Chr(servoposition);	Wait 1
'microservoboard.bas	Chr(servoposition);		
'für		Wait 1	
'das Micro Servo Board	Wait 1		Servoposition = 18
		Servoposition = 78	Print Chr(255) ; Chr(servonummer) ;
'Aufgabe:	Servoposition = 251	Print Chr(255) ; Chr(servonummer) ;	Chr(servoposition);
' Dieses Testprogramm bewegt zwei Servos, die am	Print Chr(255) ; Chr(servonummer) ;	Chr(servoposition);	
Micro Servo Board angeschlossen sind,	Chr(servoposition);		Wait 1
' schrittweise hin und her.		Wait 1	
' Der Jumper am Servoboard muss gesteckt sein, da	Wait 1		
der MINI SSC MODE		Servoposition = 68	Servoposition = 8
' zur Ansteuerung von Servos genutzt wird	Servoposition = 128	Print Chr(255) ; Chr(servonummer) ;	Print Chr(255) ; Chr(servonummer) ;
	Print Chr(255) ; Chr(servonummer) ;	Chr(servoposition);	Chr(servoposition);
'Autor: Ulf Cadenbach	Chr(servoposition);		
#####		Wait 1	Wait 1
#####	Wait 1		
	Servonummer = 1	Servoposition = 58	Servoposition = 0
\$regfile = "m32def.dat"	Servoposition = 118	Print Chr(255) ; Chr(servonummer) ;	Print Chr(255) ; Chr(servonummer) ;
\$framesize = 32	Print Chr(255) ; Chr(servonummer) ;	Chr(servoposition);	Chr(servoposition);
\$swstack = 32	Chr(servoposition);		
\$hwstack = 32		Wait 1	Wait 1
	Wait 1	Servoposition = 48	Servoposition = 251
'\$crystal = 16000000	Servoposition = 108	Print Chr(255) ; Chr(servonummer) ;	Print Chr(255) ; Chr(servonummer) ;
'Quarzfrequenz	Print Chr(255) ; Chr(servonummer) ;	Chr(servoposition);	Chr(servoposition);
\$baud = 2400	Chr(servoposition);		
'Übertragungsrate	Chr(servoposition);	Wait 1	Wait 1
(niedrigste)		Servoposition = 38	Servoposition = 128
	Wait 1	Print Chr(255) ; Chr(servonummer) ;	Print Chr(255) ; Chr(servonummer) ;
Dim Servonummer As Byte	Servoposition = 98	Chr(servoposition);	Chr(servoposition);
Dim Servoposition As Byte	Print Chr(255) ; Chr(servonummer) ;		
	Chr(servoposition);	Wait 1	Wait 1
'Der Zahlenbereich 0 bis 254 umfasst einen 90 Grad			Loop
Drehbereich	Wait 1	Servoposition = 28	
Do		Print Chr(255) ; Chr(servonummer) ;	End

Quellen:

Hintergrund	S. 3 - 7 von wikipedia.org
Bildquellen	(jeweils v.l.n.r. v.o.n.u):
S. 8 und 9	Ulf Cadenbach
S. 10	unbekannt, Festo AG, Festo AG
S. 11	Ulf Cadenbach
S.12	Ulf Cadenbach, Univ of Michigan Mobile Robotics Laboratory
S. 13 -16	Ulf Cadenbach
S.17	Conrad AG, übrige www.robotikhardware.de
S. 19 - 22	Ulf Cadenbach
S. 24	www.arduino.cc
S. 25	Ulf Cadenbach

Copyright Notice:

Copyright for all images, text and source code published in this document is held by Ulf Cadenbach, unless stated otherwise. Any downloading, printing or any other method of electronic transmitting of images, text or source code without permission in writing on behalf of the artist/author is prohibited. To apply for copyright permission, please contact >me< stating the exact way in which the image, text or source code will be used.

Urheberrechtsbelehrung:

Die Rechte an allen Bildern, Texten und Programmcode in diesem Dokument, gehören Ulf Cadenbach, sofern nicht anderweitig angegeben. Jede Art der Verbreitung von Bildern, Texten oder Programmcode ohne schriftliche Genehmigung des Autors/Künstlers ist ausdrücklich vorbehalten. Für eine Genehmigung bitte ich Sie, mich zu kontaktieren und mir genau zu schildern, wie der entsprechende Text, Bilder oder Programmcode genutzt werden sollen.

Ulf Cadenbach | 12. Semester Produktdesign | Kunsthochschule Kassel | 2007
Kontakt: info@ulfcadenbach.de | +49 (0) 561 - 870 48 68