

**Entwicklung einer
heuristischen Verhaltensregelung
für die Visuomotorik
humanoider Roboter**

Dissertation

zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Benjamin Panreck

Berlin 2021

Erstgutachter: Prof. Dr. Dr. (h.c.) habil. Raúl Rojas

Zweitgutachter: Prof. Dr. Manfred Hild

Tag der Disputation: 10.02.2022

Abstract

Performing visuomotor actions is a common yet highly complex task for humanoid robots. Although computational power in autonomous robotic systems is rapidly increasing, the robustness and efficiency of simple principles remains and has been proven once again. Partly inspired by behavioral psychology, the present work develops task-specific heuristics to perform four selected visuomotor tasks adapted from an established developmental diagnostic test for children. A considerable ambition is to implement the development on a real humanoid robot. Because the two joints in each arm provide three degrees of freedom together, the two dimensions of freedom in shoulder can be directly coupled to one visual sensory channel each, the horizontal and vertical object position, via sensorimotor loops. The elbow joint is relaxed for selected processes to enable the robot's movements to be guided by environmental constraints. This approach has the benefit of working without prior knowledge, i.e. explicit models of the robot or the environment. Later, the robot self-optimizes its actions by generating, selecting and optimizing implicit models in the form of quadrics to compensate for its limited knowledge, such as the lack of visual distance estimation. Analogous to humans, the robot empirically learns from mistakes to improve its skills over time. The empirical knowledge is stored in offset maps assigned to the currently valid quadrics. Parts of the robot's computing unit has been upgraded and further developed beforehand to support efficient and flexible work with the humanoid robotic system.

With the ambition to implement complex behaviors in a real humanoid robot platform through simple heuristic principles and without predefined models, this work makes a substantial contribution to the field of hand-eye coordination of humanoid robots.

Key Words:

Hand-eye coordination, Heuristics, Humanoid Robot, offset maps, Quadrics, Sensorimotor loop, Environmental constraint, Behavior control, Visuomotor control

Zusammenfassung

Das Absolvieren von visuomotorischen Handlungen ist eine typische aber auch hochkomplexe Aufgabe für humanoide Roboter. Obwohl die Rechenleistung von autonomen Robotersystemen rasant steigt, wurde in der Vergangenheit immer wieder die Mächtigkeit von einfachen Prinzipien unter Beweis gestellt. Die vorliegende Arbeit entwickelt problemspezifische Heuristiken, welche teilweise aus der Verhaltenspsychologie inspiriert sind, um vier ausgewählte visuomotorische Aufgaben aus einem etablierten entwicklungsdiagnostischen Test für Kinder zu absolvieren. Ein wesentlicher Anspruch liegt darin, die Entwicklung auf einem realen humanoiden Roboter umzusetzen. Da der eingesetzte Roboter über nur drei Freiheitsgrade je Arm verfügt, war es möglich, die beiden Freiheitsgrade in der Schulter über sensomotorische Schleifen direkt mit je einem visuellen sensorischen Kanal, der horizontalen und vertikalen Objektposition, zu koppeln. Das Ellenbogengelenk wurde bei ausgewählten Prozessen relaxiert, um über Umgebungsbeschränkungen durch die Umwelt geführt zu werden. Diese Verfahren arbeiten ohne Vorwissen wie explizite Modelle des Roboters oder der Umwelt. Während der Bewegungen optimiert der Roboter seine Handlungen selbst, indem er implizite Modelle in Form von Quadriken generiert, auswählt und optimiert; so wird begrenztes Wissen, wie etwa eine fehlende visuelle Abstandsschätzung, kompensiert. Genau wie der Mensch, optimiert der Roboter sein Handeln empirisch aus Fehlern, um seine Fähigkeiten zu verbessern. Die Erfahrungswerte werden in Form von Offsetkarten gespeichert und den jeweils gültigen Quadriken zugeordnet. Vorab wurden Komponenten der Recheneinheit des Roboters neu entwickelt, um eine effiziente und flexible Arbeit mit dem humanoiden Roboter zu ermöglichen.

Mit dem Anspruch, die komplexen Verhaltensweisen mit einfachen heuristischen Prinzipien und ohne A-priori-Modelle auf einer realen Roboterplattform umzusetzen, liefert diese Arbeit einen relevanten Beitrag auf dem Forschungsgebiet der Hand-Auge-Koordination bei humanoiden Robotern.

Schlagwörter:

Hand-Auge-Koordination, Heuristik, Humanoider Roboter, Offsetkarten, Quadriken, Sensomotorische Schleife, Umgebungsbeschränkung, Verhaltensregelung, Visuomotorik

Inhalt

1	Einleitung	1
2	Stand der Forschung	8
2.1	Verhalten durch Heuristiken	8
2.2	Visual Servoing als Verfahren der Hand-Auge-Koordination	11
2.3	Einblick in die Roboteranwendung	15
2.4	Entwicklungstests als handlungsleitende Aufgaben	19
3	Humanoider Roboter für den praktischen Teil	21
3.1	Aufbau der Hardware	21
3.2	Verteilte Systemarchitektur der Elektronik	28
3.3	Weiterentwicklung der Roboterplattform	31
3.3.1	Taktmanagement der deliberativen Einheit	36
3.3.2	Videosignal von der Kamera bis zur Ausgabe	39
3.4	Zusammenfassung	46
4	Wahrnehmung des Körpers und äußerer Reize	47
4.1	Taktile Wahrnehmung	47
4.2	Visuelle Wahrnehmung	52
4.2.1	Aktives Sehen	56
4.2.2	Objekterkennung durch Farbselektion	58
4.2.3	Objektmerkmale durch Eckenerkennung	61
4.3	Zusammenfassung	64
5	Körperbeherrschung als Basis von Bewegungen	65
5.1	Regelung der Gelenke	67
5.2	Vorwärtskinematik als Referenz	77
5.3	Zusammenfassung	78
6	Visuomotorik mit Hilfe von sensomotorischen Schleifen	79
6.1	Objekt visuell verfolgen	82
6.2	Objekt greifen	83
6.3	Objekte stapeln	88
6.4	Linien nachzeichnen	93
6.5	Zusammenfassung	98
7	Verhaltensorientierung durch Quadriken	99
7.1	Verfahren zur Generierung von Quadriken	103
7.2	Bewegungen auf Quadriken	123
7.3	Nutzen der Quadriken für die Hand-Auge-Koordination	131
7.4	Nebenläufiges Lernen der Quadriken	135
7.5	Auswahl einer Quadrik als Optimierungsbasis	144
7.6	Zusammenfassung	149

8	Empirisches Optimieren der visuomotorischen Fähigkeiten	150
8.1	Abbilden der Korrekturen durch Offsetkarten	151
8.2	Online-Analyse des Greifvorgangs	153
8.3	Auswertung des Stapelns von Objekten	155
8.4	Auswertung der Zeichenergebnisse	156
8.5	Zusammenfassung	157
9	Übersicht der heuristischen Verhaltensregelung	158
10	Zusammenfassung dieser Arbeit	163
11	Ausblick	167
	Anhang	169
	Literatur	172

1 Einleitung

Roboter sind aus unserem heutigen gesellschaftlichen Leben nicht mehr wegzudenken. Um 1960 setzte General Motors als einer der ersten Großkonzerne einen Roboter in der Automobilproduktion ein [HM16]. Das Schweißen und Montieren von Autoteilen erfordert schnelle und genaue Bewegungen mit hohen Kräften. Die Tätigkeiten sind wiederkehrend und müssen bereits ab dem ersten Bauteil so exakt wie möglich ausgeführt werden, wobei die Fehlerquote gegen null gehen sollte, damit ein perfektes Produkt ausgeliefert werden kann. Dies erfordert exakt geplante Bewegungen der Roboterarme und speziell darauf angepasste Umgebungen, sowie eine Anpassung der Greifer an die Objekte. Die Beleuchtungsparameter müssen für die Kamerasysteme optimiert sein, sodass beispielsweise trotz reflektierender Oberflächen die Objektkonturen gut erkennbar sind. Zusammengefasst: Das robotische System wird als Ganzes an die Aufgabenstellung angepasst und erfüllt seine Aufgabe wie erwartet.

Mit zunehmender Rechenleistung und sinkenden Anschaffungskosten hat sich die Anzahl kommerziell verfügbarer Roboter in den letzten Jahren vervielfacht. Neben Computern, Smartphones und Sprachassistenten ziehen auch Roboter in viele unserer Haushalte ein. Sie verrichten Tätigkeiten wie Staubsaugen und Rasenmähen oder dienen als Unterhaltungsroboter. Laut einer repräsentativen Forsa-Umfrage aus dem Jahre 2016 glauben 76% der deutschen Bundesbürger, dass Roboter in der Zukunft auch in unserem persönlichen Alltag als Service-Roboter eine wichtige Rolle spielen werden. Etwas mehr als die Hälfte der Befragten könnte sich grundsätzlich vorstellen, einen Service-Roboter zu kaufen; ermöglicht der Service-Roboter im Alter ein längeres Verbleiben im vertrauten Wohnbereich, sind es sogar 83% der Befragten [For16]. Nach Aussage der *International Federation of Robotics* wurden 2019 weltweit mehr als 23,2 Millionen Service-Roboter für den privaten Gebrauch verkauft. Das entspricht einer Steigerung zum Vorjahr um 34% [IFR20].

Vermutlich kennt jeder Roboterbesitzer die Situation, dass der Roboter sich festfährt, den neuen Teppich beschädigt, statt dem Rasen die Lieblingsblume mäht oder nicht zurück zur Ladestation findet. Auch wenn solche Situationen lästig sind, nehmen wir diese in Kauf und genießen die Arbeitserleichterung und Zeitersparnis – oder finden zumindest Gefallen am Zuschauen. Obwohl Haushaltsroboter immer wieder in Situationen kommen, in denen der Mensch von außen eingreifen muss, werden solche Systeme zunehmend eingesetzt. Fehler werden ihnen zugestanden, denn uns ist bewusst, dass die zu bewältigenden Aufgaben nicht trivial sind – immerhin ist jeder Haushalt anders. Die Räumlichkeiten, in denen sich das System bewegt, unterliegen durch unsere alltägliche Nutzung einem kontinuierlichen Wandel, mit dem

der Roboter umgehen muss.

Im öffentlichen Raum begegnen uns zunehmend Roboter, die in ihrer Komplexität über die kompakten Haushaltsroboter hinausgehen. Sie begrüßen uns im Museum und bieten Informationen zu den Räumlichkeiten und Exponaten [FBE⁺09] [SKIH07]. Sie verkaufen uns Kaffee¹ oder kleinere Produkte, wenn der Handel geschlossen hat². Sie unterhalten uns in Einkaufszentren [AAHL17] und unterstützen die Arbeit in Pflegeeinrichtungen [TWJ⁺14]. Eines haben viele dieser komplexen Roboter gemeinsam: Sie gehören zur Gruppe der *Humanoiden Roboter*. Ein wesentliches Merkmal ist, dass ihre Gestalt in irgendeiner Form an die menschliche Gestalt angelehnt ist. Auf Grund der unglaublich vielen Einsatzmöglichkeiten humanoider Roboter werden sie oft für den jeweiligen Anwendungsfall angepasst. Sind sie stationär, benötigen sie nicht zwangsläufig Beine oder einen fahrenden Untersatz. Roboter, mit denen das zweibeinige (*bipedale*) Laufverhalten erforscht wird, erfüllen ihren Zweck gegebenenfalls auch ohne Arme oder einen Kopf.

Humanoide Roboter werden in immer mehr Bereichen eingesetzt. Dabei erfordert die Mobilität der Systeme das Mitführen von Energiespeichern und eine kompakte Bauweise aller Komponenten wie Elektronik, Motoren, Getrieben und Sensoren. Jedes zusätzliche Gelenk erhöht die Flexibilität des Roboters, aber auch die Komplexität der Bewegungen. Jeder zusätzliche Sensor bietet weitere Informationen über den Roboter selbst oder dessen Umwelt, erhöht aber die Menge an Informationen die verarbeitet werden müssen. Zwangsläufig beeinflusst jede Komponente den Energieverbrauch und damit die Betriebszeit sowie die Kosten des Systems. Im Gegensatz zu klassischen Industrierobotern müssen sich *mobile Roboter* an die Gegebenheiten der aktuellen Umgebung anpassen können. Sind sie zudem auch noch *autonom* bedeutet das, dass sie in allen Aspekten unabhängig von ihrer Umwelt sind.

¹Beispielsweise der *Robotic Coffee Master* des chinesischen Herstellers OrionStar.

²Beispielsweise der *Workerbotkiosk* des Berliner Herstellers pi4_robotics.

Zielsetzung der Arbeit

Der Titel dieser Arbeit *Entwicklung einer heuristischen Verhaltensregelung für die Visuomotorik humanoider Roboter* setzt sich aus mehreren Begriffen zusammen, die an sich schon ein sehr umfangreiches Forschungsfeld beschreiben. Dies ist typisch für das Gebiet der humanoiden Robotik, da viele Disziplinen ineinander greifen müssen, um das System erfolgreich in Betrieb nehmen zu können. Der Begriff *Verhalten* ist aus psychologischer Sicht eine umfassende Bezeichnung für alle Aktivitäten, Vorgänge und körperlichen Reaktionen – in diesem Fall des Roboters – die beobachtet oder gemessen werden können [MN07]. Als Besonderheit wird das Verhalten hauptsächlich durch *Heuristiken* bestimmt. Sie bezeichnen einfache und sparsame (ökonomische) Regeln, die es ermöglichen, komplexe Probleme mit minimaler Information zu lösen [Gig08]. Dabei finden sie allerdings nicht immer die optimale Lösung und können auch zu falschen Ergebnissen führen.

Die heuristische Verhaltensregelung, die den Schwerpunkt dieser Arbeit bildet, wird eine typische visuomotorische Aufgabe für humanoide Roboter übernehmen: das Greifen und Umsetzen von Objekten [BK19]. Die Umsetzung erfolgt – abstrakter formuliert – anhand von Handlungen, die eine Koordination zwischen visueller Wahrnehmung und Bewegungsapparat erfordern (*Visuomotorik*). Der Begriff der Wahrnehmung fasst sowohl den Prozess als auch das Ergebnis der Informationsgewinnung über die Sensoren zusammen. Dabei wird die Wahrnehmung im Allgemeinen durch Erfahrungen und Erwartungen beeinflusst. Ein bekanntes Beispiel hierfür sind optische Täuschungen [MN07].

Die Aufgaben sind angelehnt an das entwicklungsdiagnostische Testverfahren *ET 6-6-R* für Kinder von 6 Monaten bis 6 Jahren [PM13]. Dafür werden vier Tests aus den Bereichen Handmotorik und kognitiver Entwicklung an die Randbedingungen eines realen humanoiden Roboters angepasst.

- Verfolgen eines Gegenstandes mit den Augen
- Ergreifen eines Gegenstandes mit einer Hand, von der Kleinfingerseite her
- Stapeln von zwei Gegenständen aufeinander
- Nachzeichnen von horizontalen, vertikalen und diagonalen Linien

Der Mensch verfügt über fantastische motorische Fähigkeiten und kann deshalb eine Vielzahl komplexer Aufgaben problemlos und ohne bewusstes Nachdenken lösen. Möglicherweise erscheinen daher diese vier Aufgaben auf den ersten Blick trivial. Die

genannten Fähigkeiten für die ersten beiden Tests werden bei Kindern bereits ab einem Lebensalter von 6 bis 7,5 Monaten erwartet, das Stapeln zweier Gegenstände mit 9 bis 21 Monaten, und das Nachzeichnen von Linien mit 48 bis 60 Monaten, also 4 bis 5 Jahren. [PM13].

Kinder werden mit einfachen Greifreflexen geboren. Alles was man ihnen in die Hände legt, greifen sie. In den ersten drei Lebensjahren entwickeln sie eine individuelle Kontrolle über die Finger, und bis zum neunten Lebensjahr bereits ähnliche Greiffähigkeiten wie ein Erwachsener [TT10].

Werden diese Aufgaben auf einen humanoiden Roboter übertragen, heißt das, dass er sich an seine Umwelt anpassen (*adaptieren*), diese erkunden (*explorieren*) und verändern (*manipulieren*) muss. Dazu sind vielfältige Aspekte der Bildverarbeitung, der Sensorik für die Wahrnehmung des eigenen Körpers (*Propriozeption*), der Berührungserkennung (als Teil der *Exterozeption*), der Motorik und deren Koordination untereinander zu berücksichtigen. Im Folgenden wird vorgestellt, wie ein bereits existierender humanoider Roboter weiterentwickelt wird, wobei jeder Schritt von der Erfassung der Sensordaten über das Ausführen der Bewegungen bis hin zum Lernen aus der Handlung betrachtet und nah an der Hardware umgesetzt wird.

Diese Arbeit leistet damit einen weiteren Beitrag zur Entwicklung der Visuomotorik von humanoiden Robotern. Die Spezialisierung erfolgt durch den Einsatz von Heuristiken, die direkt an einem realen Robotersystem entwickelt werden – und zwar nicht nur nach den Grundsätzen der Informatik, sondern auch nach denen der Verhaltenspsychologie. Als weitere Besonderheit werden alle eingesetzten Algorithmen selbst implementiert und auf die wesentlichen Funktionen minimiert. Der Einsatz externer Bibliotheken wie etwa OpenCV zur visuellen Wahrnehmung wird dadurch eingespart. Passend zur verhaltenspsychologischen Ausrichtung der Heuristiken kommen zur Evaluierung die oben dargestellten vier entwicklungsdiagnostischen Tests zum Einsatz. Es wird gezeigt, wie diese Tests durch eine Relaxation des Ellenbogengelenks des Roboters, also einer Reduktion der aktiv geregelten Freiheitsgrade, erfolgreich absolviert werden. Dafür wird – entgegen dem aktuellen Forschungstrend – ohne explizite Modelle des Roboterkörpers oder der Umwelt gearbeitet. Die visuelle Wahrnehmung wird zusätzlich künstlich in der Auflösung reduziert, um zu verarbeitende Datenmengen zu begrenzen und eine Fokussierung auf die wesentlichen Informationen im Kamerabild zu ermöglichen. Im Laufe der Versuche und mit zunehmender Erfahrung des Roboters wird mit dem Ellenbogengelenk ein weiterer Freiheitsgrad aktiv geregelt, und dadurch das visuomotorische Verhalten durch das Feedback der vorherigen Versuche verbessert. Dazu baut der Roboter ein implizites Modell auf, welches die Wechselwirkung zwischen dem Roboter und dessen Umwelt

beschreibt und flexibel genug ist, sich an den stetigen Wandel der Parameter anzupassen.

Sensomotorische Schleifen zeichnen sich dadurch aus, dass Bewegungen (motorische Aktionen) durch Reize (sensorische Informationen) geregelt werden. Beim Menschen bedarf es nur etwas Übung und diese Handlungen erfolgen unbewusst (*kinästhetisch*). Michel nennt als Beispiel, dass bereits nach kurzer Übung das Schalten der Gänge eines Autos ausschließlich durch die Wahrnehmung der eigenen Bewegung erfolgen kann [MN07]. Im Rahmen dieser Arbeit werden sensomotorische Schleifen eingesetzt, um beispielsweise die horizontale und vertikale Objektposition im Kamerabild direkt mit je einem Schultergelenk des Roboters zu koppeln.

Die erstaunlich zuverlässigen Resultate, die in der Vergangenheit durch vereinfachte Algorithmen und Heuristiken erreicht wurden, sind die Inspiration dafür, bei der Entwicklung von visuomotorischen Fähigkeiten ebenfalls den Fokus auf Minimierung³ und den Einsatz von einfachen sensomotorischen Schleifen zu legen. Exemplarisch sind die Lokalisation entlang einer Route durch eine Folge von nur wenigen Bildpixeln [Mil13], der inzwischen in nahezu jedem Kamerasystem implementierte Gesichtserkennung von Viola und Jones [VJ01] sowie die Erkenntnis, dass ein Satz von einfachen unkorrelierten Entscheidungsbäumen (Random Forest) sich gegen 179 alternative, teils deutlich komplexere Klassifikatoren in der Genauigkeit durchsetzt [FDCBA14]. Als positiver Nebeneffekt sind die Erkenntnisse auch auf weniger leistungsfähige Roboter portierbar, sodass beim Betrieb auf eingebetteten Systemen weitere Ressourcen für andere Aufgaben zur Verfügung stehen.

³Bezogen auf die benötigte Rechenleistung, den Speicher, das Kamerasystem, die Anzahl an Freiheitsgraden, die Anzahl an Sensorwerten/-qualitäten und deren Auflösung.

Aufbau der Arbeit

Dieses Dokument gliedert sich in elf aufeinander aufbauende Kapitel. Nach dieser Einleitung folgt mit dem zweiten Kapitel eine Einordnung in den Stand der Forschung. Auf Grund der vielschichtigen Themen dieser Arbeit beschränkt sich dieses auf wesentliche Schwerpunkte wie Heuristiken, Visual Servoing, Roboter Manipulation und entwicklungsdiagnostische Tests in der Kinder- und Jugendpsychologie, und stellt den Bezug zur vorliegenden Arbeit her. Die theoretische Einführung zu mathematischen Verfahren oder Algorithmen erfolgt in den Kapiteln, wo diese relevant sind.

Das dritte Kapitel stellt die eingesetzte Roboterplattform vor. Neben der bestehenden Roboterplattform als Hardwareaufbau wurde begleitend zu dieser Arbeit die deliberative Einheit des Roboters neu entwickelt und diente als zentrale Basis für die hardwarenahe Entwicklung der heuristischen Verhaltensweisen. Besonderes Augenmerk lag auf dem Taktmanagement und der Verarbeitung der visuellen Signale. Der erste Schritt zur Erfüllung der entwicklungsdiagnostischen Aufgaben ist das Wahrnehmungssystem, welches im vierten Kapitel mit der Wahrnehmung von visuellen und taktilen Reizen vorgestellt wird.

Nach der Wahrnehmung bedarf es motorischer Fähigkeiten, und so widmet sich das fünfte Kapitel der Körperbeherrschung. Diese wird im Wesentlichen durch die reaktive Regelung der Roboter gelenke sowie die Einführung einer Vorwärtskinematik umgesetzt. Die Vorwärtskinematik hat für den Roboter selbst keine Funktion, dient jedoch im Rahmen dieser Arbeit als Referenz, sodass die Bewegungen im kartesischen Raum, dem Aktuationsraum, evaluiert werden können.

Das sechste Kapitel behandelt die Visuomotorik. Es ist das zentrale Kapitel dieser Arbeit: Hier wird das heuristische Verhalten des Roboters bei der Lösung der vier entwicklungsdiagnostischen Aufgaben zur Kopfbewegung und Hand-Auge-Koordination unter Nutzung von Umweltbeschränkungen dargestellt.

Die Weiterentwicklung der Hand-Auge-Koordination beschreibt das siebte Kapitel. Hier wird die passive Reaktion auf Umweltbeschränkungen zunehmend durch aktive Regelung mittels parallel erlernter Quadriken abgelöst.

Macht der Roboter Fehler oder erfolgen die Handlungen unpräzise, ist der Roboter in der Lage, sein Verhalten empirisch durch eigene Wahrnehmung oder die Unterstützung des Menschen selbst zu optimieren. Diesem Aspekt ist das achte Kapitel gewidmet. Dort wird beschrieben, wie die visuelle Wahrnehmung genutzt wird, um Offsetkarten zu bilden, die das Verhalten mit zunehmender Anzahl an Versuchen verbessert.

Um die heuristische Verhaltensregelung, welche im Verlauf der Arbeit Stück für Stück entwickelt wird, angemessen herauszustellen, werden die zentralen Heuristiken im neunten Kapitel in einer Übersicht zusammengefasst und die Parallelen zur Verhaltenspsychologie dargestellt.

Das zehnte und elfte Kapitel fassen die Ergebnisse dieser Arbeit zusammen und bieten einen Ausblick auf Wege und Möglichkeiten zur Anwendung und Weiterentwicklung dieser Ergebnisse.

Einen Überblick über die Entwicklungsschritte in dieser Arbeit bietet Abbildung 1. Die einleitenden sowie zusammenfassenden Kapitel sind in der grafischen Übersicht nicht berücksichtigt.

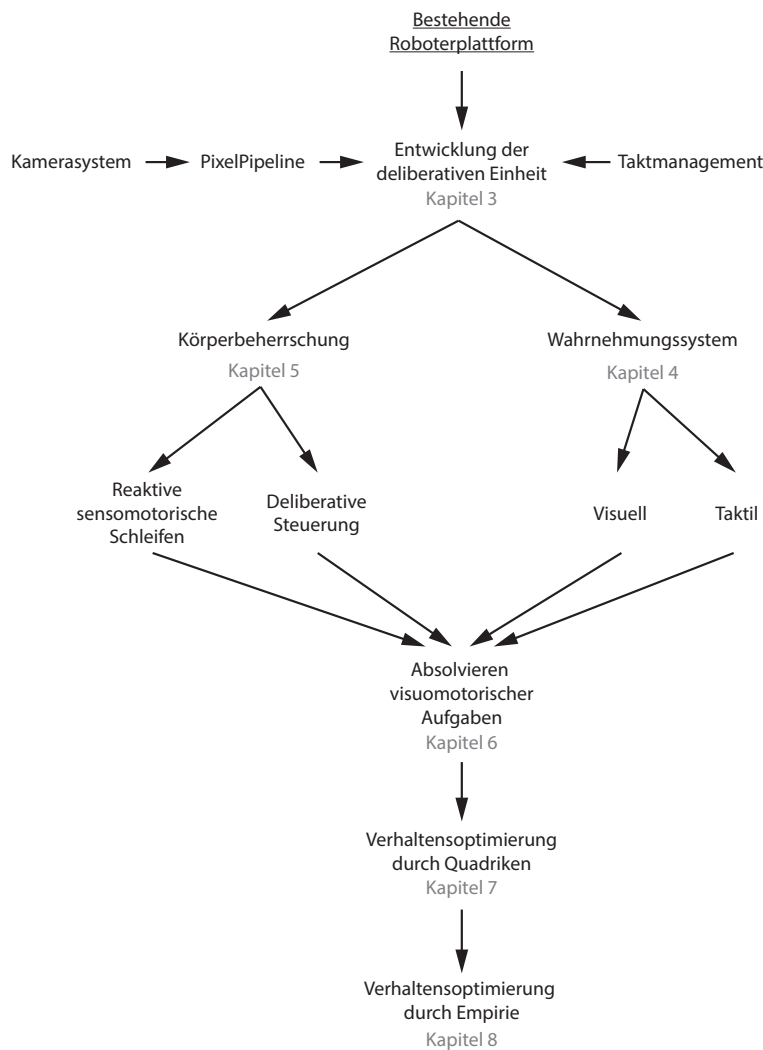


Abbildung 1: Gesamtübersicht der aufeinander aufbauenden Entwicklungsschritte dieser Arbeit. Einleitende und zusammenfassende Kapitel sind nicht aufgeführt.

2 Stand der Forschung

Die Forschung mit humanoiden Robotern ist auf Grund der universellen Einsetzbarkeit ihrer Ergebnisse sehr vielfältig. Entsprechend umfangreich ist der Stand der Forschung auf diesem Gebiet. Je nach Forschungsrichtung sind die Arbeiten der Robotik eng mit einer oder mehreren anderen Disziplinen wie der Mechanik, Elektronik, Verhaltenspsychologie, Bewegungssteuerung, Bildverarbeitung, Ethik und vielen weiteren verzahnt. Dieses Kapitels beschränkt sich daher auf die wichtigsten Themen mit einem direkten Bezug zu dieser Arbeit.

2.1 Verhalten durch Heuristiken

„Unter heuristischen Methoden sind jene systematischen, d.h. also auch nicht mathematischen Problemlöseverfahren zu verstehen, die mit Hilfe allgemeiner bzw. spezieller heuristischer Regeln (Prinzipien, Strategien, Verfahrensvorschriften) eine bestimmte Klasse von Problemen bzw. ganz spezielle Probleme zu lösen versuchen, jedoch im Einzelfall das Auffinden einer zulässigen Lösung und in jedem Fall das Auffinden einer optimalen Lösung nicht garantieren können.“

(Beier, 1974, entnommen aus [BH77] auf Seite 93)

Der Psychologe Gerd Gigerenzer bezeichnet Heuristiken als einfache und sparsame (ökonomische) Regeln, die es ermöglichen, komplexe Probleme mit minimaler Information in kurzer Zeit zu lösen [Gig08]. Eine einheitliche Definition dieses Begriffes gibt es nicht, jedoch folgen die meisten Definitionen einem vergleichbaren Tenor wie die beiden genannten von Beier und Gigerenzer.

Gigerenzer verwendet die Begriffe Heuristik, Faustregel und Instinkt synonym [Gig08]. Gerade der Begriff Instinkt ist ein interessanter Vergleich, denn auch dieser wurde nie eindeutig definiert. Für instinktives Verhalten, welches wir vor allem Tieren zuschreiben, haben wir in der Regel keine exakte Erklärung. Teller schrieb bereits 1875 ganz allgemein, dass der Instinkt Tieren zur Erhaltung und Fortpflanzung dient [Tel75].

Forscher haben zahlreiche Heuristiken identifiziert, die unsere alltäglichen Handlungen und unbewussten Entscheidungen bestimmen [GHP11]. Eine der bekanntesten motorischen Handlungen auf Basis von Heuristiken ist die Blickheuristik (*gaze-heuristic*) nach McLeod und Dienes [MD96]. Angenommen, ein mobiler Roboter soll einen Ball auffangen, welcher sich von hoch oben nähert. Er benötigt zum Beispiel

die ursprüngliche Entfernung und Anfangsgeschwindigkeit des Balls, seinen Luftwiderstand und Drall sowie die Windstärke und -richtung, um mit einer gewissen Sicherheit den Punkt der Landung zu berechnen. Mit etwas Glück ist sowohl die Berechnung als auch die Bewegung zum Landepunkt abgeschlossen, bis der Ball angekommen ist. Würde er einfach seine Laufrichtung und -geschwindigkeit so wählen, dass der Blickwinkel (d.h. der Winkel abweichend zur geraden Blickrichtung) konstant bleibt, weiß der Roboter zwar nicht vorab, wann und wo der Ball eintrifft, wird aber zum Zeitpunkt des Eintreffens am selben Ort sein. Er würde eine schnelle und sparsame Heuristik (*fast and frugal heuristic*) anwenden [GG06]. Sie benötigt keine der zuvor genannten Informationen zur Berechnung einer Flugbahn und lässt sich in einem robotischen System sehr effizient als sensomotorische Schleife⁴ implementieren, wie dies in Kapitel 6 erfolgen wird. Sie gilt zudem als kognitive Heuristik, wenn sie die folgenden drei Eigenschaften erfüllt [GHP11]:

Nutzung evolvierter Fähigkeiten

Evolvierte Fähigkeiten sind angeborene, durch die Evolution über viele Generationen erlangte Möglichkeiten, die während der Lebensspanne erlernt werden, aber nur erlernt werden können, weil die Evolution das vorbereitet hat. Beispielsweise das aufrechte Gehen, das visuelle Verfolgen von Objekten, das Wiedererkennen von Objekten, das Lesen oder die menschliche Sprache.

Nutzung von Umweltstrukturen

Heuristiken wirken immer im Zusammenhang mit der Umwelt, indem Sie deren Struktur nutzen oder verändern. Die Blickheuristik beispielsweise wandelt die komplexe Flugbahn eines Objektes in eine gerade Linie um, um eine Kollision herbeizuführen (Ball fangen) oder zu vermeiden (aufeinander zufliegende Flugzeuge steuern).

Beschreibung des Lösungsprozesses

Die Heuristik zeigt den Weg auf, wie ein Problem gelöst wird, und führt dann zum gewünschten Ergebnis.

Zusammengefasst: Kognitive Heuristiken beschreiben den Lösungsprozess und nicht das Ergebnis. Sie sind einfach, da sie auf evolvierten Fähigkeiten basieren. Sie können als intelligent bezeichnet werden, da sie Umweltstrukturen nutzen [GG06].

Möchte man ein Modell für eine Heuristik aufstellen, sollte dieses drei Punkte bein-

⁴In der Robotik werden Regelschleifen, die den aktuellen Zustand des Systems in der Umwelt durch Sensoren erfassen und daraus neue Motorsignale generieren, als sensomotorische Schleifen bezeichnet.

halten: erstens eine Verfahrensregel, wie die Heuristik abläuft; zweitens die Fähigkeiten, auf die die Heuristik aufbaut (und welche vom System vorgegeben sein müssen); und drittens die Art der Probleme, die mit der Heuristik gelöst werden können.

Basierend auf der Rückmeldung durch die Umwelt können sich Heuristiken adaptiv an Veränderungen anpassen. Payne [PBJ93] und Rieskamp [RO03] haben diese Prozesse beim Menschen nachgewiesen.

Bei Heuristiken mit mehreren Kriterien ist die Suche anhand von *Cues* (Kriterien) sehr mächtig. Dabei sind im Wesentlichen zwei Klassen zu unterscheiden. Das *One Reason Decision Making* und die *Tallying*-Heuristik.

One Reason Decision Making (ORDM)

Die *Cues* werden entsprechend einer zuvor festgelegten Priorität durchgegangen. Der erste zutreffende *Cue* diskriminiert die Suche und führt zur Entscheidung. Alle weiteren Kriterien niedrigerer Priorität werden ignoriert.

Tallying-Heuristik

Hier werden die Kriterien nicht zwangsweise priorisiert, sondern die Ergebnisse der *Cues* (1/0) werden aufaddiert. Die Suche ist beendet, wenn entweder alle Kriterien überprüft wurden (eine sogenannte erschöpfende Suche, siehe Einheitsgewicht-Modelle [Daw79]), wenn eine bestimmte Anzahl an signifikanten Kriterien überprüft wurde, oder wenn eine bestimmte Schwelle erreicht ist [Daw79]. Die Reihenfolge der Kriterien kann bewusst oder zufällig gewählt werden, oder die Kriterien werden nach der zeitlichen Abfolge der eintreffenden Informationen verarbeitet.

Sind die *Cues* stark unterschiedlich in ihrer Gewichtung und damit tendenziell nicht-kompensatorisch, ist das *One Reason Decision Making* eine gute Wahl. Bei sehr ähnlichen Gewichtungen, was zu kompensatorischen Kriterien führt, ist dagegen die *Tallying*-Heuristik mächtiger. Beide Varianten eignen sich nur für Kriterien (Prädiktoren) mit zwei Optionen. Bei mehr als zwei Optionen beziehungsweise kontinuierlichen Prädiktoren müssen diese vor der Anwendung dichotomisiert⁵ werden. Beispielsweise kann ein Objektgewicht unter 5 kg als „leicht“ und größer/gleich 5 kg als „schwer“ definiert werden.

In der Informatik kommen Heuristiken dann zum Einsatz, wenn eine exakte Berechnung der Lösung nicht möglich ist, da zu wenige Informationen vorliegen, oder wenn die Berechnung mit der vorhandenen Hardware nicht realisierbar oder zu zeitaufwändig ist. Manchmal ist weder die optimale Lösung noch eine Methode zur

⁵Bei der Dichotomisierung wird die Menge in zwei Teile ohne Schnittmenge aufgeteilt

Berechnung bekannt. Klassische Beispiele sind die klar definierten Probleme beim Schach und Tetris sowie das *Traveling Salesman Problem*⁶ [MF00].

Einen Überblick über heuristische Algorithmen in der Pfadplanung mobiler Roboter bieten Mac [MCTDK16] und Zafar [ZM18]. Dort erfolgt ein Vergleich zwischen der von ihnen als klassisch bezeichneten Pfadplanung und der heuristischen Pfadplanung mobiler Roboter. Mit dem Programm ABSOLVER II war es Prieditis [Pri93] möglich, durch sogenannte Abstraktionen leistungsfähige Heuristiken automatisch zu generieren, deren Entdeckung durch den Menschen Jahre hätte dauern können, wie zum Beispiel die erste effektive Heuristik zur Lösung des Zauberwürfels (*Rubik's Cube*). Oliver et. al. [OLC19] haben bei einem humanoiden *iCub*-Roboter ein proaktiv interferentes Verhalten⁷ für Greifaufgaben implementiert und damit ein beim Menschen bekanntes heuristisches Verhalten in der Robotik eingesetzt.

Wie diese Beispiele zeigen, werden Heuristiken in der Informatik hauptsächlich zur Durchsuchung von Datensätzen angewandt und haben wenig mit dem Verhalten eines Roboters zu tun. Für das Wahrnehmungssystem und motorische Prozesse in humanoiden Robotern werden Heuristiken derzeit kaum eingesetzt. Um eine effektive Heuristik zu einem Problem zu finden, sind besonders bei motorischen Aktionen Experimente mit einem physikalischen robotischen System notwendig.

2.2 Visual Servoing als Verfahren der Hand-Auge-Koordination

Verfahren, bei denen die Bewegung eines Roboter durch die Rückkopplung von visuellen Informationen geregelt wird, gehören zur Gruppe des *Visual Servoing* [SSVO10]. Dafür werden Bilddaten analysiert und das Ergebnis als Sensorgröße in den Regelkreis eines Systems eingebunden.

Servomotor (Servo) bezeichnet dabei eine Einheit aus einem Motor, einem Positionssensor, einer Regeleinheit und einem (optionalen) Getriebe. Der Servomotor erhält als Eingangswert (Führungsgröße) beispielsweise eine Position oder eine Geschwindigkeit, die er eigenständig einregelt. Beim *Visual Servoing* wird die Sensorgröße aus visuellen Merkmalen generiert.

Ist die Kamera am Endeffektor (Greifer) eines Roboters montiert, handelt es sich um

⁶Das *Traveling Salesman Problem* beschreibt die Herausforderung eines Handlungsreisenden. Ziel ist es, eine Reise zu planen, bei der alle Zielorte nur einmal bereist werden, die wieder am Anfangspunkt endet und dessen Gesamtstrecke minimal ist. Dieses Optimierungsproblem lässt sich auf andere Anwendungsfälle übertragen, wie beispielsweise die Routenplanung eines Paketdienstes oder das Layout von Leiterbahnen in der Elektronik.

⁷Proaktive Interferenz ist ein Begriff aus der Gedächtnispsychologie und bezeichnet die Beeinflussung von neuen Erfahrungen durch bereits Bekanntes. Ein typisches Beispiel aus dem Alltag: Man muss ausnahmsweise eine andere Buslinie nehmen als die täglich gewohnte, steigt aber versehentlich doch in den gewohnten Bus.

eine *End-point-closed-loop*- beziehungsweise *Eye-in-hand*-Konfiguration. Dies ist die bevorzugte Methode für industrielle Anwendungen, da der Zusammenhang zwischen Kameraposition und Greiferposition konstant ist. Humanoide Roboter, die die Kamera klassischerweise (aber nicht immer) in einem beweglichen Kopf verbaut haben, nutzen die *End-point-open-loop*-/*Eye-to-hand*-Konfiguration. Hier muss sowohl die Position des Greifers als auch die Position des zu greifenden Objekts über das Kamerabild ermittelt werden. Alternativ kann die Kamera unabhängig vom Roboter in der Umwelt fixiert sein.

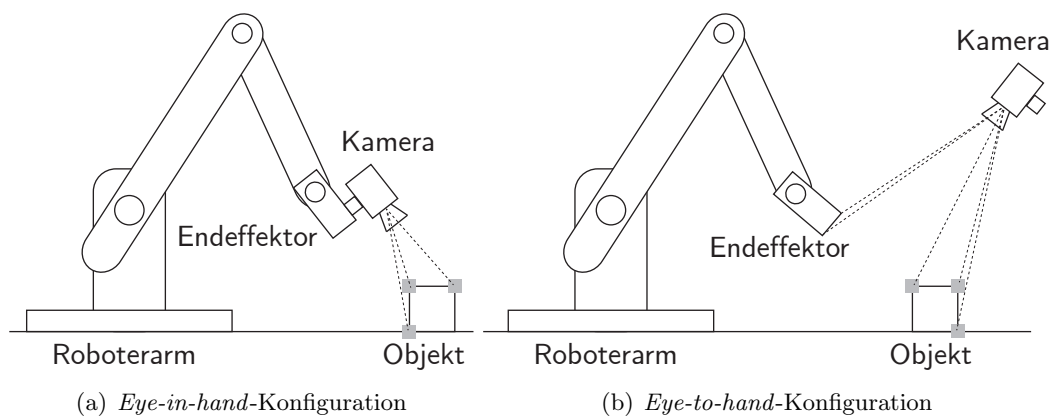


Abbildung 2: Arten des Visual Servoing, basierend auf der Position der Kamera. Die Darstellung erfolgt schematisch an einem Roboterarm und gilt analog für humanoide Roboter. (a) Bei der *End-point-closed-loop*-/*Eye-in-hand*-Konfiguration ist die Kamera starr mit dem Endeffektor des Roboters verbunden. (b) Bei der *End-point-open-loop*-/*Eye-to-hand*-Konfiguration wird sowohl der Endeffektor als auch das Objekt mit einer ortsunabhängigen Kamera betrachtet.

Neben der Position der Kamera wird auch zwischen positionsbasiertem *Visual Servoing* (*Position Based Visual Servoing*/PBVS) und bildbasiertem *Visual Servoing* (*Image Based Visual Servoing*/IBVS) unterschieden. Das PBVS berechnet die Führungsgröße für den Servo aus den Positionen des Endeffektors sowie des zu greifenden Objekts in den Raumkoordinaten des Roboters (Aktuationsraum) [WHB96]. Beim IBVS wird die Führungsgröße nicht aus den dreidimensionalen Raumkoordinaten berechnet, sondern aus den zweidimensionalen Bildkoordinaten des Endeffektors und des zu greifenden Objekts [WSN87]. Bedingt durch das Funktionsprinzip kann es sein, dass bei der positionsbasierten Regelung das Zielobjekt aus dem Kamerabild verschwindet, oder bei der bildbasierten Regelung zwar das Zielobjekt immer im Bild bleibt, jedoch eine Pose angesteuert wird, die der Roboter nicht anfahren kann. Um diese Nachteile zu kompensieren, können die beiden Verfahren zu einem hybriden Verfahren kombiniert werden [MCB99]. Beide Verfahren benötigen neben den Objektkoordinaten im Bild auch den Objektabstand. Da das IBVS-Verfahren recht

robust gegen Fehler in der Abstandsschätzung ist, kann diese während der Bewegung gelernt und optimiert werden [SNAC13] [ZJZ21].

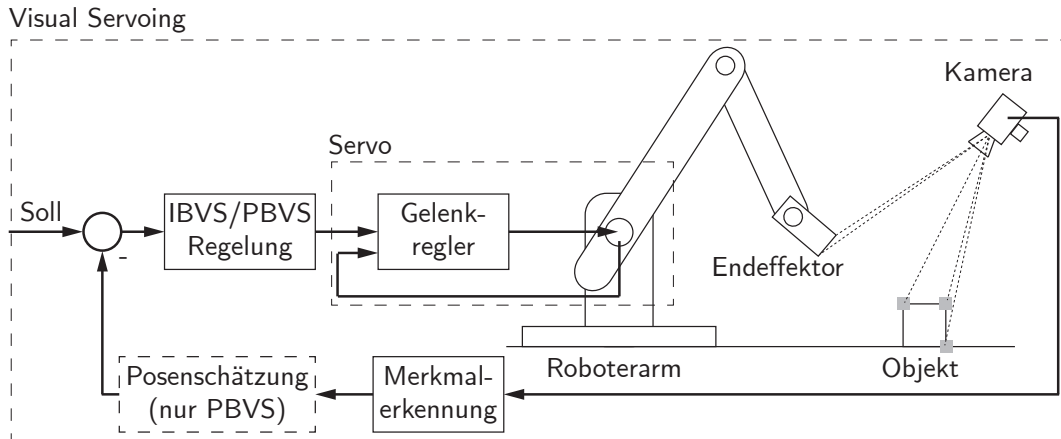


Abbildung 3: Schematische Darstellung der Regelkreise beim *Visual Servoing*, unabhängig von der Position der Kamera (hier in der *Eye-to-hand*-Konfiguration dargestellt). Das *Position Based Visual Servoing (PBVS)* nutzt einen inneren Servo-Regelkreis, der entsprechend seiner FührungsgröÙe das Roboter-gelenk regelt. Anhand der visuellen Merkmalerkennung wird mit der Posenschätzung die Lage der zweidimensionalen Bildmerkmale im dreidimensionalen Raum geschätzt und mit einem dreidimensionalen Soll-Wert verglichen. Die Differenz (Fehler) dient der PBVS-Regelung zur Korrektur der FührungsgröÙe des Servo-Regelkreises. Dies gilt auch analog für das *Image-Based Visual Servoing (IBVS)*, jedoch erfolgt hier keine Posenschätzung und die zweidimensionalen Bildmerkmale werden direkt mit einem zweidimensionalen Sollwert abgeglichen. Die Differenz ergibt die EingangsgröÙe für die IBVS-Regelung.

Grundlage für die Berechnungen der IBVS-/PBVS-Regelung sind spezielle Jacobi-Matrizen (*Jacobians*). Je nachdem, ob ein Modell des Roboters (*Roboter-Jacobian*) vorliegt, sind unterschiedliche *Jacobians* notwendig, um die gewünschten Gelenkwinkel des Roboters im Konfigurationsraum⁸ (*Joint Space*) zu berechnen. Die *Jacobians* sind nur so lange gültig wie die geometrische Konfiguration der Kamera konstant bleibt. Bewegt der Roboter beispielsweise den Kopf und damit auch die Kamera, müssen die *Jacobians* neu berechnet werden. Abbildung 4 zeigt die drei typischen *Jacobians*. In Anlehnung an Abbildung 3 besteht beim klassischen *Visual Servoing* der Zusammenhang, dass die Posenschätzung die Bild-*Jacobian*, die PBVS-Regelung die Roboter-*Jacobian* und das IBVS die gekoppelte Bild-Roboter-*Jacobian* berechnet, um die jeweils benötigten GröÙen zu ermitteln.

Die Literatur zum *Visual Servoing* konzentriert sich in der Regel auf kinematische Probleme und ignoriert dabei eine Reihe anderer grundlegender Probleme, die durch

⁸Der Konfigurationsraum hat so viele Dimensionen, wie der Roboter unabhängige Freiheitsgrade hat. Im Gegensatz zum Phasenraum stellt ein Punkt im Konfigurationsraum keine vollständige Beschreibung des Zustandes des mechanischen Systems dar.

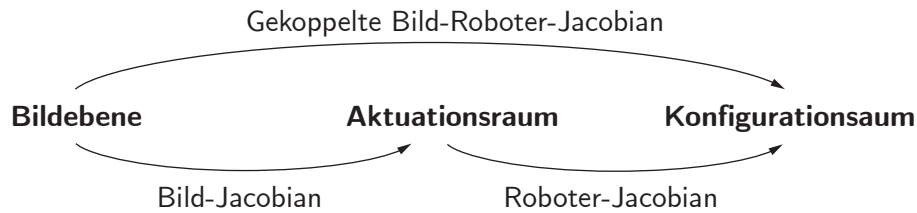


Abbildung 4: Verknüpfung von Bildebene, Aktionsraum und Konfigurationsraum mittels *Jacobians*. Angelehnt an [KC⁺02].

die Dynamik des Systems bedingt sind, wie zum Beispiel die Latenz des Bildverarbeitungssystems. In vielen Implementationen sind die Geschwindigkeit des Endeffektors und die Rückkopplungsverstärkungen streng begrenzt, um zu gewährleisten, dass die dynamischen Effekte vernachlässigt werden können [CG96]. Weitere Herausforderungen sind Nichtlinearitäten in optischen Kamerasystemen, zeitaufwändige Bildverarbeitungsalgorithmen, unzuverlässige Sensordaten und unzureichend genaue Modelle des Roboters und der Objekte im Kamerabild. Diese werden im Rahmen des klassischen *Visual Servoing* nicht betrachtet. Music et. al. [MBC14] haben in einer Simulation verschiedene Verfahren zum Schätzen der *Bild-Jacobian* für das IBVS-Verfahren in der *Eye-in-hand*-Konfiguration verglichen. Es hat sich gezeigt, dass im statischen Fall andere Verfahren brillieren als im dynamischen Fall und eine Umschaltung zwischen den Verfahren von Vorteil sein kann.

Eine junge Entwicklung ist das ereignisbasierte *Visual Servoing (Event Based Visual Servoing/EBVS)*, das insbesondere bei hohen Geschwindigkeiten und schlechten Lichtverhältnissen auf Basis von Ereignistriggern einer neuromorphen Kamera gute Ergebnisse erzielt [MAH⁺21]. Wie auf vielen Gebieten der Robotik wird auch beim *Visual Servoing* an Implementationen auf Basis von *CNNs*⁹ (*Convolutional Neural Networks*) geforscht – mit einerseits großem Lernaufwand, jedoch auch beachtlichen Erfolgen [TAK20].

Ein Beispiel ist der humanoide Roboter ARMAR-III vom KIT (Die letzte Weiterentwicklung ist der ARMAR-6 [AWK⁺19]). Er nutzt PBVS, indem die Handposition über einen roten Ball an der Hand detektiert und das zu greifende Objekt über ein zuvor angeleitetes Modell wiedererkannt wird [VWA⁺08]. Später wurde das PBVS für zweiarmige Greif- und Manipulationsaufgaben erweitert [BVA⁺12].

⁹*Convolutional Neural Networks/CNNs* sind eine spezielle Form von künstlichen neuronalen Netzen. Ein wesentliches Merkmal ist, dass es einen oder mehrere *Convolutional Layer* gibt, bei denen die Neuronen (häufig) in Form einer 2- oder 3-dimensionalen Matrix angeordnet sind. Auf diesen *Layer* wird dann eine diskrete Faltung in Form einer verhältnismäßig kleinen Faltungsmatrix angewandt, die systematisch über den *Layer* bewegt wird.

2.3 Einblick in die Roboteranipulation

Dieses Kapitel betrachtet den aktuellen Stand der Forschung zur Manipulation von Objekten durch Roboter. Dem Menschen gelingt es, sich vielen Aufgaben individuell anzupassen. Obwohl wir viele Parameter, wie Kräfte, Momente, Reibungen und Massen, die bei der Objektmanipulationen auftreten, nicht kennen, setzen wir dieses Wissen zur Bewältigung der Aufgaben durch Roboter doch oft unbewusst voraus.

Greifer und Griffplanung

Noch immer ist der meist eingesetzte Greifer der Zwei-Backen-Greifer ohne jede Sensorik [BK19]. Ein Greifer sollte möglichst leicht sein, da sein Eigengewicht die maximal zu hebende Last des Roboters verringert. Neben dem Gewicht ist auch der Bauraum bei der Integration aller Aktuatoren, Sensoren und mechanischen Komponenten in der räumlich begrenzten Hand zu berücksichtigen. Da viele Greifer aus festen Materialien bestehen, stellt dies oft den limitierenden Faktor in der Geschicklichkeit des Roboters dar, da die harten Greifelemente die zu greifenden Objekte schnell beschädigen können und, besonders bei universellen Greifelementen, nur wenig Berührungspunkte mit dem Objekt aufweisen. Die resultierenden Reibungskräfte sind gering und die Flächenpressung entsprechend hoch. Die Anzahl flexibler Greifer (*Softrobotics*) nimmt daher stetig zu [BK19]. Sie können die Greifobjekte umschließen, sind flexibler gegen Lastvariationen beim Greifen und verteilen die Griffkraft gleichmäßiger auf das Objekt [WADL⁺20]. Neben den flexiblen Elementen der Hardware wird alternativ oder in Kombination per Software eine erhöhte Flexibilität angestrebt, zum Beispiel durch variable Steifigkeiten in den Antriebseinheiten. Nachteilig beim Einsatz von flexiblen Greifern ist derzeit die vergleichsweise geringe Geschwindigkeit und Belastbarkeit [BK19]. Besonders jung ist die Entwicklung von flexiblen und sensitiven Roboterhäuten, welche eine erhöhte taktile Wahrnehmung mit vertretbarem Mehraufwand in der Verkabelung, Masse und Auswertung der Signale versprechen [LZ19]. Derzeit werden die daraus zu gewinnenden Informationen über Objektform, -größe und -orientierung sowie zur Oberflächenbeschaffenheit noch durch visuelle Wahrnehmung, kombiniert mit Basiswissen zu den Objekten, erschlossen beziehungsweise geschätzt [Bha19].

Besonders für humanoide Roboter ist aus ästhetischen Gründen eine anthropomorphe Hand interessant. Technisch gesehen ist deren Universalität und Komplexität allerdings recht schwer umzusetzen, besonders wenn diese über ähnliche Eigenschaften wie die menschliche Hand in Bezug auf Masse, Beweglichkeit, Kraft, Empfindlichkeit und Größe verfügen soll. Für den speziellen Anwendungsfall sind deshalb einfachere oder alternative Greiferprinzipien oft besser geeignet [BK19].

Bei der Planung der Griffpunkte eines Objekts für den jeweils eingesetzten Roboter greifer wird im Wesentlichen zwischen analytischen und datenbasierten Verfahren unterschieden. Bogh [BMAK14] hat Ergebnisse beider Verfahren untersucht und festgestellt, dass besonders bei analytischen Verfahren ein genaues Modell des zu greifenden Objekts notwendig ist, um anhand einer Metrik die optimalen Griffpunkte zu berechnen. Datenbasierte beziehungsweise empirische Ansätze bewerten und kategorisieren dagegen die vergangenen Griffversuche, um diese bei gleichen oder ähnlichen Situation abrufen zu können. Die zu greifenden Objekte lassen sich in bekannte Objekte, ähnliche Objekte und unbekannte Objekte einteilen. Während das Greifen bekannter oder ähnlicher Objekte anhand von vorhandenen Modellen oder Datenbankwissen zu den erkannten Objektmerkmalen erfolgt, werden bei unbekanntem Objekten hauptsächlich Heuristiken angewandt, um die Sensordaten den bekannten Griffarten zuzuweisen. Für Objekte, die den in der Datenbank erfassten ähneln, müssen gute Ähnlichkeitsmaße entwickelt werden, um die nächstbeste Griffart zu wählen. Datenbasierte Verfahren berechnen im Gegensatz zu den analytischen Verfahren keine Qualitätsmaße wie Gleichgewicht, Geschicklichkeit, Stabilität oder das dynamische Verhalten. Ihr Erfolg wird empirisch überprüft. Mit diesem Wissen können dann auch analytische Verfahren optimiert werden.

Lernmethoden basieren auf dem heuristischen Prinzip von Versuch und Irrtum, auf händisch gekennzeichneten Datensätzen und auf der Demonstration von Griffen durch den Menschen (*Learning from Demonstration*) [RSGD19]. Das Lernen kann sowohl in der Simulation als auch auf einem realen Roboter erfolgen. Auf einem realen Roboter muss mit systematischen und zufälligen Fehlern umgegangen werden, da dort im Gegensatz zur Simulation Rauschen in den Sensordaten sowie ungenaue Modelle der Kinematik, Dynamik, Sensorik oder der Objekte keine Ausnahme darstellen. Untersuchungen [BMAK14] haben gezeigt, dass die durch Simulation ermittelten Greifpositionen keine guten Resultate auf realen Robotern erzielen. Die der Simulation zu Grunde liegenden Modelle sind also bei weitem noch nicht gut genug, um die realen Bedingungen hinreichend genau abzubilden. Daher wird oft auf kostenintensive reale Systeme zurückgegriffen, so dass die Roboter das Greifen durch praktische Erfahrungen lernen.

Bei klar definierten und bekannten Objekten mit speziell auf die Aufgabe angepassten Greifsystemen sind Roboter recht zuverlässig und effizient im Greifen und Loslassen. Laut Billard [BK19] sollte der Trend zu austauschbaren Greifern gehen, die je nach Anwendungsfall wie Werkzeuge in Bearbeitungsmaschinen schnell und zuverlässig gewechselt werden können. Unüblich ist die Betrachtung, was und wie mit einem bestehenden Roboter und seinem Greifer gegriffen werden kann. Die robo-

tische Hand mit ihren Stärken und Schwächen steht bei den Betrachtungen meistens im Hintergrund.

Im Rahmen dieser Arbeit wird ein bestehender unteraktuierter Vier-Finger-Greifer eingesetzt. Dieser wird in Kapitel 3.1 auf Seite 26 vorgestellt. Die Entwicklung des Greifprozesses erfolgt aus der Gesamtbetrachtung der Fähigkeiten des Greifers, der Form der zu greifenden Objekte, sowie der Anforderungen der zu absolvierenden Aufgaben.

Wahrnehmung und Ausführung

Die Entwicklung zunehmend besserer visueller und taktiler Wahrnehmung führt in Kombination mit flexiblen Mechaniken und Antrieben zu immer natürlicher wirkenden Prozessen. Besonders durch Verfahren des maschinellen Lernens wird der Umgang mit Unsicherheiten verbessert, was zu adaptiven und robusteren Regelungen führt. Roboter können sich auf gewisse Variationen und Abweichungen in Objekteigenschaften einstellen. Da der Prozess typischerweise auf einen speziellen Bereich von Objekteigenschaften optimiert ist, hat dies jedoch seine Grenzen [BK19].

Die Objekterkennung kommt oft dann an ihre Grenzen, wenn die Objekte teilweise verdeckt sind. Eine Berechnung der Griffpositionen ist dann nicht mehr möglich, denn wenn die genaue Position und der Objekttyp nicht bekannt sind, können die im System hinterlegten Objekteigenschaften nicht abgerufen werden [NC21]. Grundsätzlich muss der Roboter über erweiterte kognitive Fähigkeiten verfügen. Es kommt dabei auf den Kontext der aktuellen Situation an, um für das jeweilige Objekt die optimale Griffposition vorausszusagen – beispielsweise muss ein gefüllter Becher anders gegriffen werden als ein leerer.

Wie beim Menschen erfolgt das Greifen bei Robotern multimodal. Aktuell werden visuelle und taktile Informationen vor allem seriell ausgewertet [BK19] – beispielsweise wird die Hand erst visuell zum Objekt geführt und der Greifprozess erfolgt dann durch taktile Wahrnehmung. Nur selten werden beide Modalitäten für die Erkennung, das Greifen und die Formrekonstruktion kombiniert. Jüngste Forschungsaktivitäten zielen auf ein kontinuierliches Lernen von visuomotorischen Fähigkeiten ab, wie dies für uns Menschen selbstverständlich ist. Burhan Hafez und Wermter [BHW21] stellen einen Ansatz vor, bei dem ein humanoider Roboter auf Basis von Demonstrationen durch den Menschen selbstorganisiert auf die zu erledigende visuomotorische Aufgabe schließt.

Interaktive Wahrnehmung

Werden Informationen aus einem einzelnen zweidimensionalen Bild extrahiert, sind oft zusätzliche Annahmen oder Vorwissen notwendig. Noë [Noë04] nennt als Beispiel einen Kreis. Erscheint ein idealer Kreis im Bild, ist anzunehmen, dass das Objekt parallel zur Kamera ist. Erscheint ein Oval, kann angenommen werden es sei ein Kreis, wenn der Rest des Bildes auf eine perspektivische Verzerrung hindeutet. Wirklich sicher ist das nur, wenn diese Randinformationen bekannt sind, oder das Objekt bewegt beziehungsweise aus einer anderen Perspektive betrachtet werden kann. Welcher Zusammenhang zwischen Handlungen in Form von Interaktionen mit der Umwelt und den daraus resultierenden Sensordaten besteht, wird auf dem Gebiet der interaktiven Wahrnehmung (*Interactive Perception*) erforscht. Bogh [BHS⁺17] definiert interaktive Wahrnehmung so, dass beim Ausüben einer beliebig gearteten Kraft auf die Umgebung folgende zwei Aspekte gegeben sein müssen: (1) Durch die Interaktion mit der Umwelt werden sensorische Informationen generiert, die sonst nicht zur Verfügung ständen. (2) Jedes Vorwissen über die Interaktion unterstützt die Interpretation des Zusammenhangs zwischen Sensorinformationen und Handlungsparametern (abhängig vom Zeitpunkt). Bei der Interaktion ist es üblich, dass Objekte mit dem Ziel des Informationsgewinns angehoben, verschoben oder auch nur berührt werden.

Das Prinzip der interaktiven Wahrnehmung wird in Kapitel 6 beim Gleiten der Roboterhand auf der Tischoberfläche angewendet, in Kapitel 7 beim Explorieren derselben, sowie beim initialen visuomotorischen Greifen von Objekten zur Positionsbestimmung im Konfigurationsraum.

Nutzung von Umgebungseinschränkungen

In der Robotik wird mit der Nutzung von Umgebungseinschränkungen (*Exploitation of Environmental Constraints*) die temporäre Reduktion der Freiheitsgrade eines Systems bezeichnet, um beispielsweise den Greifprozess zu vereinfachen. Es wird dabei angenommen, dass ein zu greifendes Objekt nicht im Raum schwebt, sondern auf einer Unterlage, wie beispielsweise einem Tisch, steht oder liegt. Bei sehr flachen Gegenständen ist das klassische Greifen von oben nicht möglich. Insbesondere mit weichen Greifelementen (*Softrobotics*) besteht jedoch die Möglichkeit, das Objekt an den Rand der Tischoberfläche zu ziehen und es von oben und unten zu greifen. Eine detaillierte Untersuchung dazu, wie die Umgebung Mensch und Roboter beim Greifen von Objekten unterstützt, hat Eppner in [EDAR⁺15] und [Epp19] durchgeführt.

Alternativ können Eigenschaften der Umwelt wie Reibung, Gravitation oder die Existenz von Kontaktflächen genutzt werden, um einen Mangel an Freiheitsgraden zu kompensieren. Dies wird als extrinsische Geschicklichkeit (*extrinsic dexterity*) [BK19] bezeichnet.

Diese Arbeit nutzt bei der Bewältigung der visuomotorischen Aufgaben (Kapitel 6) die Umgebungseinschränkung der Tischoberfläche, um ein Objekt zu greifen und Linien nachzuzeichnen. Im späteren Verlauf wird diese durch hinzugewonnenes Wissen (Kapitel 7) ersetzt.

2.4 Entwicklungstests als handlungsleitende Aufgaben

Entwicklungstests werden eingesetzt, um Entwicklungsverzögerungen sowie Entwicklungsstörungen in der (früh-)kindlichen Entwicklung abzubilden. Dabei ist bislang nicht einheitlich definiert, wann eine Entwicklungsverzögerung oder Entwicklungsstörung vorliegt.

Der Entwicklungstest ET 6-6-R von Petermann & Macha [PM13] ist konzipiert für Kinder von 6 Monaten bis 6 Jahren und untersucht die folgenden Entwicklungsbereiche: Körpermotorik, Handmotorik, kognitive Entwicklung, Sprachentwicklung und sozial-emotionale Entwicklung. Dieser Test setzt auf das Grenzsteinprinzip nach Michaelis et al. [MBNRKM13]. Die Anforderungen sind so formuliert, dass das Testergebnis ohne besondere Hilfsmittel im Alltag des Kindes abgefragt beziehungsweise beobachtet werden kann. Die sogenannten Grenzsteine sind Fertigkeiten, die für eine ungestörte Entwicklung zu einem definierten Lebensalter erworben sein sollten [PM13].

Der ET 6-6-R ist nicht der einzige Test im deutschen Sprachraum. So sind die *Bayley Scales of Infant Development (BSID-II)*, Denver-Skalen, Entwicklungsgitter, Griffiths-Entwicklungsskalen (GES), Münchener funktionelle Entwicklungsdiagnostik für das 1.–3. Lebensjahr (MFED 1–3) und der Wiener Entwicklungstest (WET) weitere ebenfalls etablierte und allgemein anerkannte Entwicklungstests in der Kinder- und Jugendpsychologie [PM13].

Die im Kapitel 6 zur Visuomotorik durchgeführten Aufgaben wurden an ausgewählte Aufgaben des ET 6-6-R angelehnt, da dieser eine große Bandbreite von Aufgaben umfasst, die auf die kognitiven und physischen Anforderungen an einen humanoiden Roboter angepasst werden können – auch über die Schwerpunkte dieser Arbeit hinaus. Grundsätzlich wird der Einsatz von standardisierten Tests in der Robotik angestrebt. Auf dem Gebiet der Objektmanipulation gibt es zwar eine Auswahl von Testdatensätzen und -szenarien, jedoch hat sich in der Forschung noch kein allgemeiner Standard durchsetzen können [BMAK14] [BK19]. Dies liegt sicher auch an der

enormen Vielfalt der Objekte, Aufgaben und Roboterplattformen. Mit Benchmarks wie GRASPA [BVPN20] wird versucht, durch flexible Metriken die Bewertung von Greifvorgängen individuell an die Fähigkeiten der jeweiligen Roboterplattform anzupassen; dies beschränkt sich jedoch ausschließlich auf das Greifen von ausgewählten Alltagsgegenständen ohne weiterführende Handlungen.

Diese Arbeit untersucht das Lösen von vier Aufgaben, die an den folgenden Tests des ET 6-6-R angelehnt sind, um die robotischen Handlungen mit ausgewählten Fähigkeiten eines Kleinkindes zu vergleichen:

- Verfolgen eines Gegenstandes mit den Augen
(angelehnt an T076)
- Ergreifen eines Gegenstandes mit einer Hand, von der Kleinfingerseite her
(*ulnar*) (angelehnt an T040)
- Stapeln von zwei Gegenständen aufeinander
(angelehnt an T085)
- Nachzeichnen von horizontalen, vertikalen und diagonalen Linien
(angelehnt an T159–T161)

3 Humanoider Roboter für den praktischen Teil

Dieses Kapitel stellt den humanoiden Roboter Myon vor. Er wurde im Forschungslabor Neurorobotik (NRL) der Beuth Hochschule für Technik Berlin entwickelt. Die Entwicklung begann 2008 im ALEAR-Projekt (*Artificial Language Evolution on Autonomous Robots*) [ALE] und setzt sich bis heute fort. Im Rahmen des Projekts wurde er für Experimente zur künstlichen Sprachevolution eingesetzt [SH12]. Aufgrund seines universellen Konzeptes [HSB⁺11] ist er auch für Forschungsvorhaben zur bipedalen Fortbewegung, Objektmanipulation, Wahrnehmung, Aufmerksamkeitssteuerung und der biologisch inspirierten Verhaltensregelung geeignet. Im Folgenden werden ausgewählte Konzepte des Roboters betrachtet, die für die weiteren Experimente relevant sind. Die Weiterentwicklung des Roboters im Rahmen dieser Arbeit wird in Kapitel 3.3 vorgestellt.

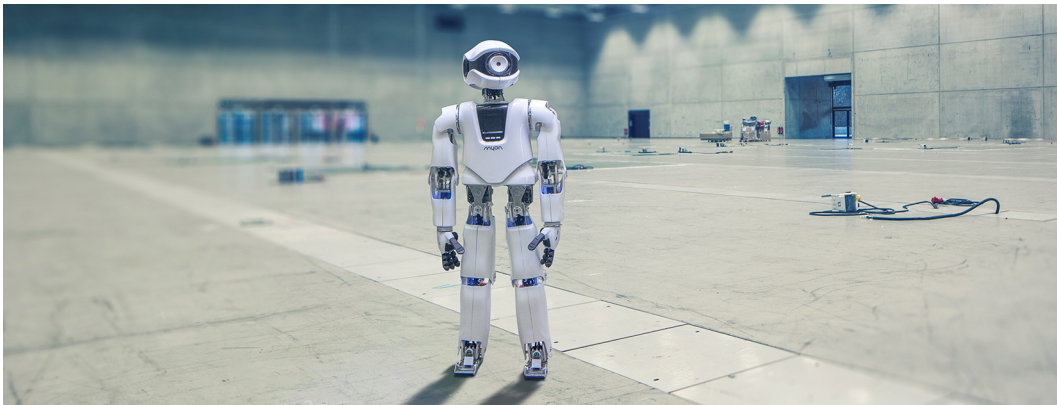


Abbildung 5: Roboter Myon im CityCube Berlin. Das Stehen wird aktiv durch eine sensomotorische Schleife geregelt. (Bildquelle: Messe Berlin)

3.1 Aufbau der Hardware

Für eine universell einsetzbare Forschungsplattform basierte die Entwicklung auf vier Designkonzepten: Modularität, Autonomie, Robustheit sowie Wartungs- und Bedienfreundlichkeit. Die Umsetzung der ersten beiden Konzepte grenzt Myon deutlich von anderen humanoiden Robotern ab.

Modularität

Modularität bedeutet hier, dass die Körperteile von Myon modular aufgebaut sind. Der Kopf, beide Arme, beide Hände, der Torso und beide Beine bilden jeweils ein eigenes Modul, die über einen speziell entwickelten Flansch mit dem Torso verbunden und auch im laufenden Betrieb abgenommen und ausgetauscht werden können.

Neben den Körperteilen selbst ist auch ein Großteil anderer Komponenten wie die Recheneinheiten (siehe Kapitel 3.2) und die Motoren mehrfach im gesamten Roboter verbaut und untereinander austauschbar. In montiertem Zustand arbeitet das dezentrale System als ein Gesamtsystem.

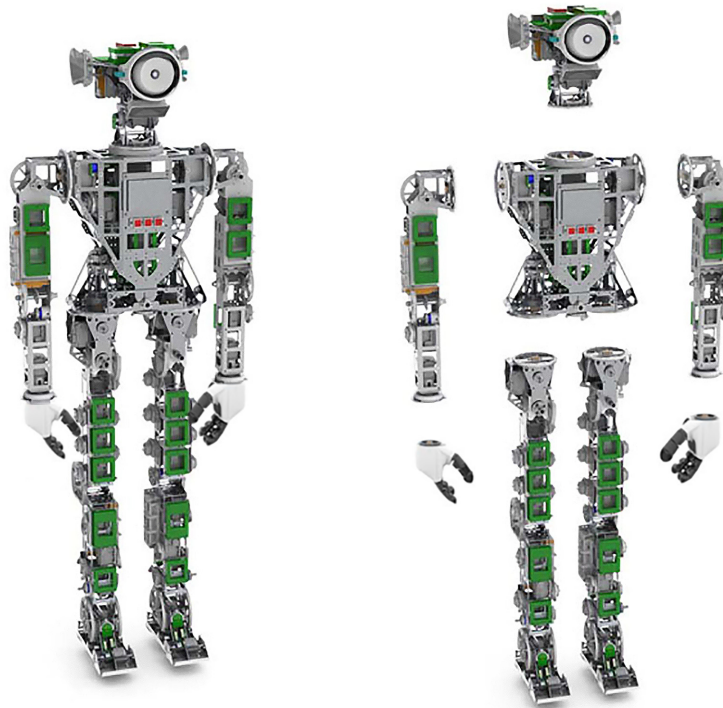


Abbildung 6: Modularer Aufbau des Roboters Myon ohne Verschalung. Links sind alle acht Module montiert. Rechts sind die Module über einen speziell für diesen Roboter entwickelten Flansch demontiert. Modifizierte Grafik aus [Sie15].

Autonomie

Da jedes Körperteil über eine eigene Energieversorgung und mehrere Recheneinheiten verfügt, sind isolierte Experimente mit nur einem oder einer Auswahl von Körperteilen problemlos möglich. Die Ergebnisse können bei Bedarf auf den vollständig zusammengesetzten Roboter portiert werden. Als Beispiel ist das stehende Bein aus [KBH11] zu nennen, dessen Prinzip anschließend für das aufrechte Stehen des gesamten Roboters eingesetzt wurde. Neben den einzelnen Körperteilen ist auch der teilweise oder vollständig zusammengesetzte Myon ein autonomes System. Die einzelnen Körperteile teilen die bereitgestellte Energie und Rechenleistung bei Bedarf mit dem restlichen System. Die einzigen Ausnahmen sind die Hände. Diese haben weder eine eigene Energieversorgung, noch eine eigene Recheneinheit.

Morphologie

Die Morphologie wird im Allgemeinen als die Lehre von Gestalt und Form definiert [Dud15]. In der Robotik bezeichnet der Begriff die Gestalt des mechanischen Aufbaus. Bei der äußeren Verschalung wurde ein geschlechtsneutrales (androgynes) Erscheinungsbild mit einer Körpergröße von 1,25 m gewählt. Dies entspricht etwa der Körpergröße eines siebenjährigen Kindes. Mit einem Gewicht von gut 16 kg ist Myon jedoch deutlich leichter als ein Kind dieses Alters. Bei der Wahl der Körpergröße musste ein Mittelweg zwischen Kostenoptimierung, guter Handhabung und der Möglichkeit, alle Komponenten zu verbauen, gefunden werden. Das Endoskelett, zu sehen in Abbildung 6, besteht aus gefrästem Aluminiumblech. Es trägt alle Komponenten des Roboters und ist auch ohne Verschalung bereits voll funktionsfähig. Das Exoskelett (die Verschalung) besteht aus zweischichtigem Vakuumthermogeformtem Kunststoff. Die innere Schicht besteht aus weiß gefärbtem, glasfaserverstärktem PC (Polycarbonat), die äußere Schicht aus transparentem PC. Die Verschalung schützt sowohl die empfindlichen Komponenten des Roboters als auch die mit ihm interagierenden Menschen.



Abbildung 7: Aufgrund seiner modularen und wartungsfreundlichen Bauweise wird Myon regelmäßig in Lehrveranstaltungen eingesetzt. Hier im Seminar „Wartung und Instandsetzung von Robotern“. (Quelle: photothek/Xander Heinel)

Kinematik

Die Kinematik¹⁰ des Roboters mit seinen 32 Freiheitsgraden (*Degrees Of Freedom/DOF*) wird in Abbildung 8 gezeigt. Ergänzend sind die Positionen der einzelnen Freiheitsgrade sowie ihre Funktion in der Tabelle 1 auf Seite 25 aufgeführt. Abgesehen von den vier Freiheitsgraden des Auges, die durch Miniaturservos angetrieben werden (je zwei für die Augenbewegung und für die Augenlider) werden alle Freiheitsgrade mit 44 Servomotoren vom Typ Dynamixel RX-28 des Herstellers

¹⁰Die Kinematik beschreibt die Bewegung von Körpern unter Vernachlässigung von Kräften und Massen. Es wird beschrieben *wie* ein Körper sich bewegt, nicht *warum*.

Robotis aktuiert. Jeder dieser Motoren verfügt über eine integrierte Elektronik mit einem Mikrocontroller, einer Winkelmessung der Abtriebswelle und einem Temperatursensor. Die Motoren sind so konfiguriert, dass ihnen die Spannung in Form einer Pulsweitenmodulation (PWM) als Führungsgröße vorgeben werden kann. Die verbauten Widerstandspotentiometer zur Winkelmessung erfassen einen Bereich von 300° . Einen mechanischen Anschlag gibt es nicht, sodass die Abtriebswelle kontinuierlich drehen kann. Der Totbereich von 60° wird nicht erfasst. Die integrierten Gleichstrommotoren werden nach dem 4-Quadranten-Prinzip per Pulsweitenmodulation gesteuert. Der Freilauf der Abtriebswelle wird über den *Coast*-Modus bereitgestellt und erlaubt, die Gelenke des Roboters einzeln zu relaxieren, da diese rücktreibbar und damit nicht selbsthemmend sind. Je nach Gelenk unterscheidet sich die geforderte Antriebsleistung. Dazu werden bis zu vier Servomotoren parallel auf ein Gelenk gekoppelt. Um zusätzlich das Drehmoment zu erhöhen, erfolgt die Kraftübertragung, kombiniert mit einer individuellen Übersetzung, durch zusätzliche Zugmittelgetriebe.

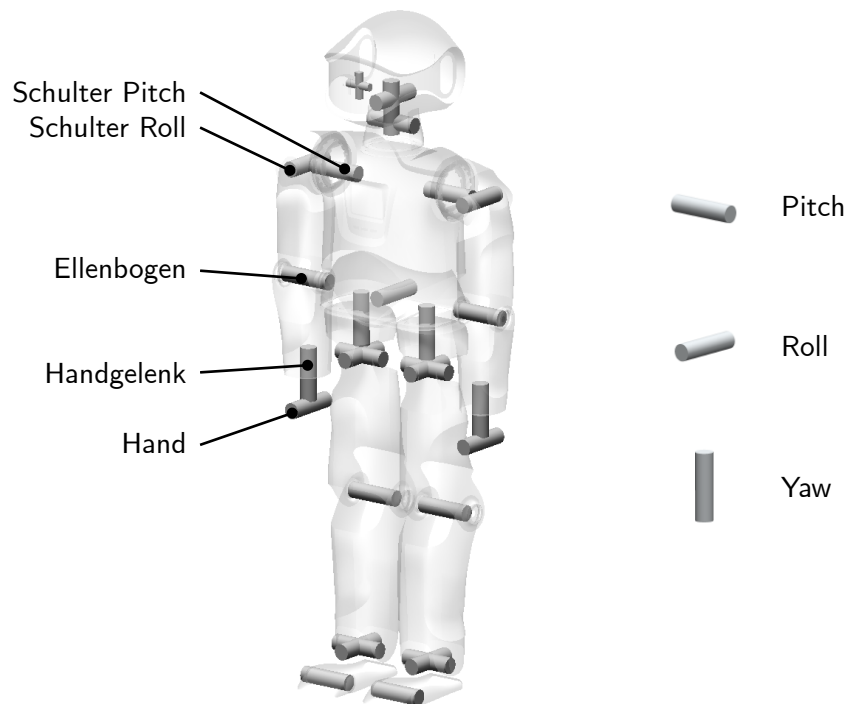


Abbildung 8: Darstellung der Freiheitsgrade (*DOF*) des Roboters Myon. Je nach Raumlage werden die Bewegungsrichtungen der Gelenke als *Pitch*, *Roll* und *Yaw* beschrieben. Die Bezeichnungen stammen aus der Flugsteuerung und werden im Deutschen als Nicken, Rollen und Gieren angesprochen. Da die Armgelenke bei der Visuomotorik besonders wichtig sind, wurden diese explizit beschriftet. Modifizierte Grafik aus [Sie15].

Tabelle 1: Gewichtsverteilung und Aktuierung des Roboters Myon. Entnommen aus [Sie15].

Körperteil	Masse	Gelenk	DOF	Anzahl Aktoren	Über- setzung
Kopf	1,89 kg	Auge und Augenlid	4	4	1 : 1
		Hals	3	3	1 : 1
Arm (2x)	1,47 kg	Schulter	1	1	1 : 2
		Ellenbogen	1	1	1 : 2
		Handgelenk	1	1	1 : 1
Greifer (2x)	0,24 kg	Finger	1	1	-
Torso	3,72 kg	Schulter (2x)	1	2	1 : 1,65
		Taille	1	1	1 : 4
		Hüfte (<i>Yaw</i> ; 2x)	1	1	1 : 2,73
Bein (2x)	3,55 kg	Hüfte (<i>Pitch</i>)	1	3	1 : 2,5
		Hüfte (<i>Roll</i>)	1	2	1 : 2,5
		Knie	1	3	1 : 2,5
		Fußgelenk (<i>Pitch</i>)	1	4	1 : 2,5
		Fußgelenk (<i>Roll</i>)	1	1	1 : 2,5
		Zeh (passiv)	1	-	-
Gesamt	16,13 kg		32	48	

Die Hände, die in der Robotik üblicherweise als Greifer bezeichnet werden, verfügen über drei Finger und einen Daumen. Wie Abbildung 9 zeigt ist die Roboterhand etwas kleiner als die Hand eines Erwachsenen.



Abbildung 9: Zum Vergleich geben sich ein Erwachsener und Myon die Hand. Der Vier-Finger-Greifer ist in seinen Abmessungen vergleichbar mit der Hand eines Jugendlichen und kann damit ähnlich große Objekte greifen wie ein Erwachsener. (Quelle: photothek/Xander Heidl)

Diese Hand gehört zur Gruppe der flexiblen Greifer (*Soft-Gripper*), da die vier Finger flexibel sind. Der Handkorpus besteht aus festem Kunststoff¹¹. Die Finger sind aus schwarz gefärbtem Polyurethan mit dichterreduzierendem Füllstoff gegossen. In der Handinnenfläche ist zur Erhöhung der Reibung ein vollflächiger Zellgummi eingelassen.



Abbildung 10: Darstellung der Zustände des Vier-Finger-Greifers: (links) offene Hand, (mitte) Zeigefinger gestreckt (Zeigen), (rechts) geschlossene Hand (Greifen/Faust). Von der offenen Hand zu den beiden Endpositionen sind beliebige Zwischenpositionen möglich. Entnommen aus [Sie15].

¹¹Der Handkorpus wird aus ABS im FDM-Druck hergestellt. ABS steht für Acrylnitril-Butadien-Styrol-Copolymer und ist ein thermoplastischer Kunststoff. FDM steht für *Fused Deposition Modeling* und ist ein dreidimensionales Druckverfahren, bei dem ein Kunststoffdraht aufgeschmolzen und in Schichten aufgetragen wird.

Der Aufbau der Hand ist minimalistisch. Die verfügbaren Zustände Zeigen und Greifen werden mit nur einem Motor (1 *DOF*) ausgeführt. In der Mittelstellung des Motors ist die Hand offen. Bei einer Drehung nach rechts schließen die Finger kontinuierlich und gleichförmig, bis die Hand zur Faust geschlossen ist. Dieser Modus ist zum Greifen von Objekten geeignet. Dreht der Motor von der Mittelstellung nach links, krümmen sich alle Finger mit Ausnahme des Zeigefingers. Damit kann der Roboter auf Objekte oder Lebewesen zeigen.

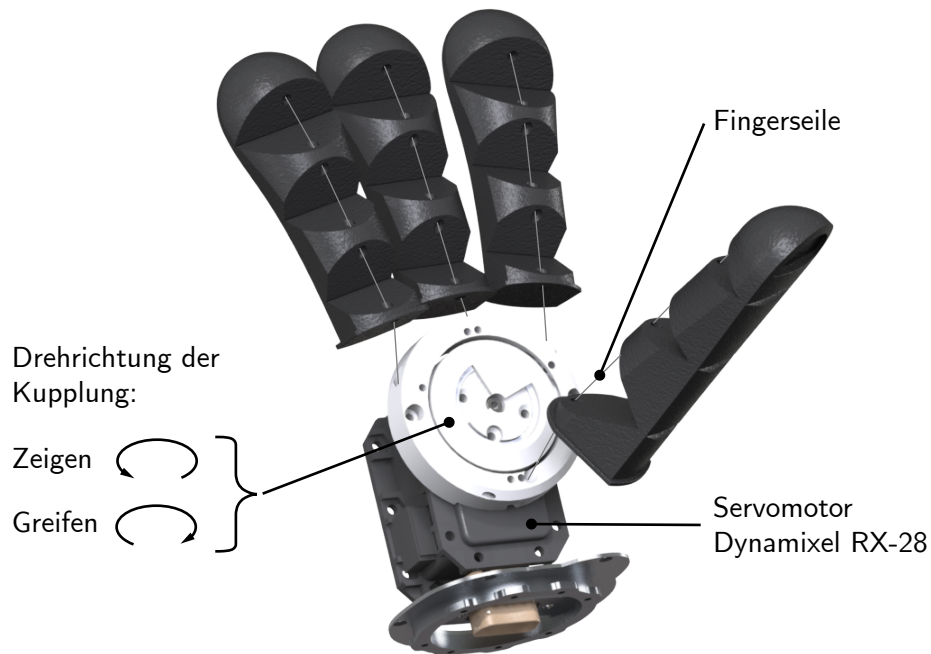


Abbildung 11: Aufbau des Vier-Finger-Greifers. Die Schale des Handkorpus ist in dieser Darstellung ausgeblendet. Befindet sich die Kupplung in Mittelstellung, ist die Hand wie gezeigt geöffnet. Eine Drehung nach rechts zieht alle Fingerseile und die Hand schließt. Eine Drehung nach links zieht alle Seile, außer dem des Zeigefingers. Modifizierte Grafik aus [Sie15].

Weitere Details zum mechanischen Aufbau des Roboters stellt Siedel in [Sie15] zur Verfügung.

3.2 Verteilte Systemarchitektur der Elektronik

Sowohl die Sensorvorverarbeitung als auch die Aufbereitung der Daten zur Motorsteuerung erfolgt in der verteilten Echtzeit-Systemarchitektur DISTAL (*Distributed Architecture for Large Neural Networks*) [Thi14]. Abbildung 12 zeigt anhand einer schematischen Darstellung des Roboters, wie die Elektronik, in Form sogenannter AccelBoard3D (AB3D), über den Körper verteilt ist.

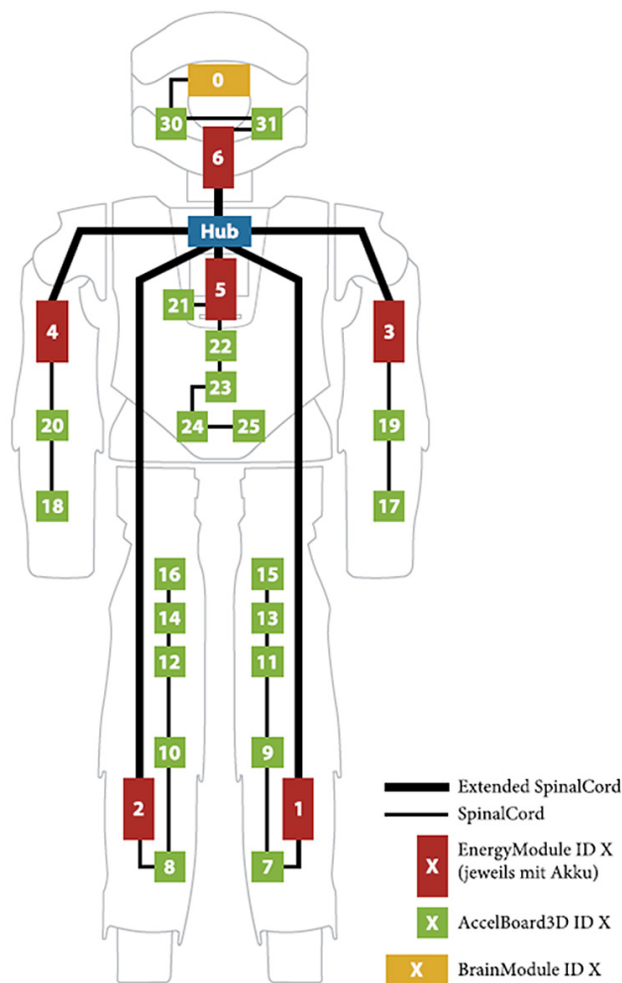


Abbildung 12: Schematische Darstellung des Roboters Myon. Dargestellt sind die Komponenten der verteilten Systemarchitektur DISTAL. Der *SpinalCord* ist der globale Datenbus, welcher ergänzt um eine Spannungsversorgung als *Extended SpinalCord* bezeichnet wird. Jedes Körperteil verfügt über einen eigenen Energiespeicher samt Elektronik (rot). Der *Hub* (blau) für den *Extended SpinalCord* verbindet die Energieversorgung und koordiniert den Datenaustausch zwischen den sechs Körperteilen auf dem *SpinalCord*. Das *BrainModule* (gelb) verarbeitet die Daten der exterozeptiven Sensorik und stellt die deliberative Einheit dar. Die 21 AccelBoard3D sind die dezentralen reaktiven Recheneinheiten; sie erfassen die Daten der propriozeptiven Sensorik und steuern die Servomotoren an. Entnommen aus [Thi14].

Jedes AB3D verfügt über einen Mikrocontroller (STM32F103CBT6), mit dem künstliche neuronale Netze berechnet werden können. Dank eines Systemtakts von 100 Hz, der über den globalen Datenbus (*SpinalCord*) für alle AB3Ds synchronisiert wird, ist das System hart echtzeitfähig. Die Signallaufzeit für jedes implementierte neuronale Netz lässt sich genau voraussagen. Über die neuronalen Netze lassen sich reaktive Strukturen wie beispielsweise sensomotorische Schleifen implementieren. Über den *SpinalCord* erfolgt neben der Synchronisation auch die Bereitstellung aller propriozeptiven Sensorwerte, die Ansteuerung aller 48 Motoren sowie, abhängig von den implementierten neuronalen Netzen, ein Datenaustausch zwischen den AB3Ds mit einer Aktualisierungsrate von 100 Hz. Tabelle 2 bietet eine Übersicht über die 238 Sensorwerte in sechs verschiedenen Sensorqualitäten.

Tabelle 2: Übersicht der propriozeptiven Sensoren des Roboters Myon. Vereinfacht aus [Sie15].

Qualität	Position und Messmethode	Mess- bereich	Auf- lösung	Anzahl
Winkel	Dynamixel RX-28 Potentiometer	300°	10 Bit	44
Winkel	Gelenke Potentiometer	340°	12 Bit	22
Kraft	Fußsohle Faraday-Kraftsensor	0–100 N	12 Bit	8
Beschleunigung	AccelBoard3D 3-Achs-Beschleunigungssensor	±8 g	10 Bit	3x21
Strom	AccelBoard3D 20 mΩ-Shunt-widerstand in Motorstromzuführung	0–8,6 A	12 Bit	36
Spannung	AccelBoard3D ADC des STM32	0–26,5 V	12 Bit	21
Temperatur	Dynamixel RX-28	0–85 °C	1 K	44
Gesamt				238

Die künstlichen neuronalen Netze sind grafisch über die Software *BrainDesigner* zu erstellen. Anschließend können sie nicht-flüchtig auf die 21 AB3Ds geflasht werden. Dabei wird berücksichtigt, dass die Teile des neuronalen Netzes, die sich auf ein spezielles Körperteil auswirken, auch auf den dort befindlichen Prozessoren berechnet werden. Die ausschließlich lokalen sensomotorischen Schleifen bleiben auch beim Trennen des Körperteils vom restlichen Körper intakt. Wie bei reaktiven Systemen üblich werden auf den AB3Ds keine Weltmodelle oder ähnliches verarbeitet, sondern ausschließlich auf aktuelle Sensordaten (Stimulus-Antwort-Prinzip) reagiert. Eine ausführliche Dokumentation zu DISTAL und *BrainDesigner* stellt Thiele in [Thi14] bereit.

Der Einsatz der neuronalen Netze wird in Kapitel 5 eingeführt. Da die Netze in ihrer Struktur strenggenommen lediglich an künstliche neuronale Netze angelehnt sind, werden sie nachfolgend als reaktive Netze angesprochen.

3.3 Weiterentwicklung der Roboterplattform

Neben der reaktiven Komponente, dessen reaktive Netze über *BrainDesigner* erstellt werden, befindet sich im Roboterkopf das sogenannte *BrainModule* (*BM*) (siehe Abbildung 13), das im Zuge dieser Arbeit entwickelt wurde. Das *BM* verarbeitet die Daten der exterozeptiven Sensoren (Kamera und Mikrofone), erzeugt die Audioausgabe und stellt die Rechenleistung für die deliberative Komponente zur Verfügung. Herzstück ist das *System-on-Chip* (*SoC*) Zynq XC7Z020 des Herstellers Xilinx, welches im Wesentlichen aus einem Dual-Core Cortex A9 Prozessor und einem Artix-7-FPGA (*Field Programmable Gate Array*) besteht. Des Weiteren sind zwei Mikro-SD-Steckplätze, ein Video-Encoder, ein Audio-Decoder, eine Echtzeituhr, ein On-Board-Speicher (1 GB SD-RAM und 32 GB eNAND) sowie diverse Anschlüsse für die Ein- und Ausgabe verbaut.

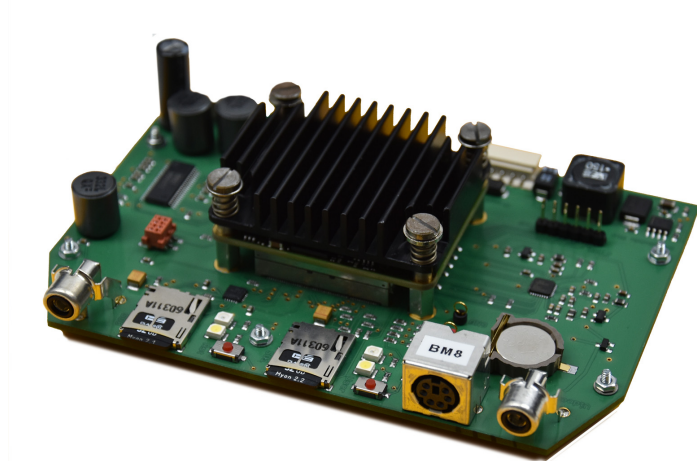


Abbildung 13: Das *BrainModule* ist im Kopf des Myon verbaut. Es verarbeitet die Daten der exterozeptiven Sensoren (Kamera und Mikrofone), erzeugt die Audioausgabe und stellt die Rechenleistung für die deliberative Komponente des Roboters zur Verfügung. Die passiv gekühlte Einheit in der Mitte ist das Herzstück, der *System-on-Chip* (*SoC*) Zynq XC7Z020 des Herstellers Xilinx. Die sichtbaren Anschlüsse sowie die Taster und SD-Karten-Steckplätze sind im eingebauten Zustand auf der Rückseite des Kopfes zugänglich.

Die Programmierung des Myon erfolgte ursprünglich über die von Xilinx zu diesem *SoC* bereitgestellte Entwicklungsumgebung¹². Die Hardwarebeschreibung des *FPGA* erfolgte in Verilog und VHDL¹³, die Programmierung der Prozessoren in C. Der gesamte Zyklus von einer Änderung im *FPGA* bis zur Aktualisierung des *SoC* dauerte je nach Rechenkapazität des Computers ca. 45 Minuten.

¹²Das *FPGA* des *SoC* wird über die Software Xilinx Vivado konfiguriert. Die Prozessoren werden über das Xilinx Software Development Kit *SDK* programmiert.

¹³VHDL steht für *Very High Speed Integrated Circuit Hardware Description Language*.

Daraufhin folgte die Entwicklung der Programmierumgebung *Behavior Design Environment (BDE)*. Diese ermöglicht die Programmierung in *Forth* ohne zusätzliche Computer und Compiler; es genügt das Anschließen eines Monitors und einer Tastatur an den Roboterkopf (siehe Abbildung 14). Optional zu *Forth* kann auch direkt eingebetteter Maschinencode ausgeführt werden, welcher für zeitkritische Operationen aus dem Assembler-Befehlssatz hergeleitet werden kann.

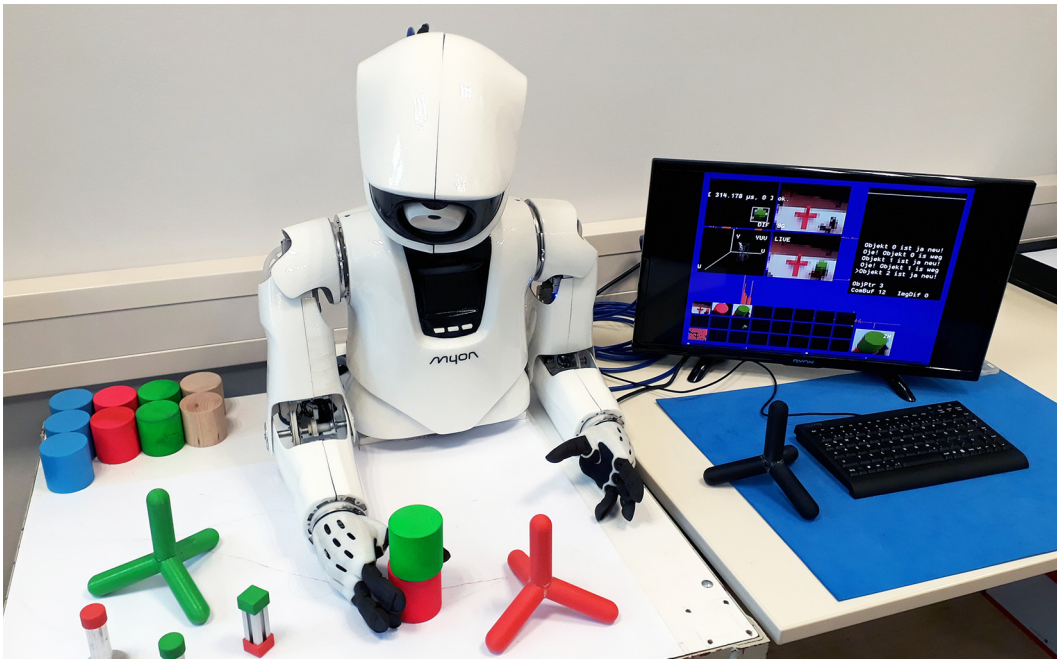


Abbildung 14: Programmierung des Roboters ohne zusätzlichen Computer, nur mit Tastatur und Bildschirm. Links ist der Oberkörper des Roboters fest auf einem Tisch montiert, rechts steht der Monitor und die Tastatur, welche direkt mit dem Roboterkopf verbunden sind. Der Monitor zeigt die Oberfläche der *Behavior Design Environment (BDE)*.

Forth wurde um 1970 von Charles H. Moore entwickelt und ist eine imperative¹⁴, stackbasierte Programmiersprache. Die Programmierung erfolgt mit der umgekehrt polnischen Notation. *Forth* ist sowohl Programmierumgebung als auch Betriebssystem. *Forth* ist selbstkompilierend, sodass prinzipiell beliebige Erweiterungen der Befehlsstruktur möglich sind. Ein auf *Forth* basierendes System ist in der Lage, sich selbst während des Betriebs zu erweitern. In der *BDE* steht ein vom gültigen Standard aus dem Jahre 2012 abgeleiteter Wortschatz zur Verfügung [Forb]. Dieser ist speziell auf die Möglichkeiten und Anforderungen der Hardware angepasst.

Forth ist zwar nicht sehr verbreitet, wird aber auch in heutigen kommerziellen Robotern in diversen Dialekten eingesetzt [Fora] [STR]. Mit dem Propeller 2 von Parallax

¹⁴Bei imperativen Programmiersprachen gibt die Reihenfolge der Anweisungen auch die Reihenfolge der Ausführung vor.

kam im November 2020 ein 32-bit Mikrocontroller mit acht identischen Prozessoren und nativer *Forth*-Unterstützung auf den Markt.

Grafische Benutzeroberfläche

Die *BDE* ist die zentrale Entwicklungsumgebung für das deliberative System des Roboters. Sie arbeitet mit einer 100 Hz-Hauptschleife und hat über den *SpinalCord* als intermediäre Schnittstelle Zugriff auf das reaktive System des Roboters inklusive der 238 Sensorwerte. Die 100 Hz-Schleifen aller 21 *AB3Ds* synchronisieren sich auf die 100 Hz des *BrainModules*.

Abbildung 15 zeigt eine typische Ausgabe auf dem angeschlossenen Bildschirm an der *BDE* während des Erkennens von vier verschiedenfarbigen Objekten im Kamerabild.



Abbildung 15: Typisches Fenster der *Behavior Design Environment (BDE)*. In diesem Beispiel werden vier farbige Objekte im Kamerabild erkannt und deren Positionen ermittelt. Unter Berücksichtigung der Auflösung der Videoausgabe von 720 x 576 Pixeln sind beliebige Darstellungen realisierbar.

Als besonders vorteilhaft hat sich erwiesen, dass während der Programmierung Teile des Algorithmus direkt ausprobiert werden können. Einen Unterschied zwischen Programmierung und Ausführung im klassischen Sinn gibt es nicht mehr, da der Roboter immer in Betrieb ist und die Algorithmen zu jedem Zeitpunkt ausgeführt werden können, auch während der Programmierung. Debugging-Tools sind nicht

notwendig, auch aufwändige Analyseroutinen entfallen hier.

Bedingt durch die proprietäre Entwicklungsumgebung lassen sich allerdings externe Softwarepakete wie OpenCV nur mit erhöhtem Aufwand einbinden. Die hardware-spezifische Implementierung für dieses System erfordert deshalb zwar deutlich höheren Programmieraufwand, führt jedoch auch zu wesentlich effizienterem Programmcode, wie dies für eingebettete Systeme mit begrenzten Ressourcen im Hinblick auf Rechenkapazität und Energie von entscheidender Bedeutung ist. In dieser Arbeit wurden alle Algorithmen selbst implementiert und auf Effizienz optimiert. Zur Programmierung der reaktiven Netze auf den AB3Ds ist derzeit noch der Einsatz eines Computers mit *BrainDesigner* erforderlich. Diesen ebenfalls durch die *BDE* zu ersetzen ist jedoch technisch bereits vorbereitet, sodass der Einsatz externer Computer zukünftig vollständig entfallen wird.

Im Rahmen dieser Arbeit wurde die Elektronik der *BrainModules* entwickelt sowie die Hardwarebeschreibung für das Taktmanagement und das visuelle System des Artix-7-FPGA im *SoC* durchgeführt, die in den folgenden Unterkapiteln näher erläutert werden.

Der strukturelle Aufbau sowie der Hauptsignalfluss der internen Struktur des *SoC* von und zur Peripherie wird in Abbildung 16 auf Seite 35 gezeigt.

Die Anwendung der Begriffe deliberative Einheit auf das *BrainModule* im Roboterkopf und reaktive Einheit auf die über den Körper verteilten *AccelBoards* entsprechen zwar strenggenommen nicht der üblichen Definition, da sie in der Regel die Verhaltensarchitektur beschreiben und eher auf Softwareelemente angewandt werden; jedoch lässt sich mit ihnen der verhaltensspezifische funktionelle Unterschied zwischen den genannten Hardwarekomponenten recht treffend beschreiben. Auf dem *BrainModule* werden die Zustandsautomaten¹⁵ für die visuomotorischen Handlungsabläufe berechnet (siehe Kapitel 6). Dort erfolgt auch die Berechnung, Überprüfung und Korrektur der expliziten Quadrikenmodelle (siehe Kapitel 7) sowie die Koordination der empirischen Verhaltensoptimierung (siehe Kapitel 8). Auf den einzelnen *AccelBoards* erfolgt die koordinierte Berechnung der lokalen sensomotorischen Schleifen. Beim Zusammenspiel des hybriden Systems werden jedoch auch einzelne sensomotorische Schleifen auf dem *BrainModule* berechnet, soweit dies für die jeweilige Aufgabenstellung effizienter ist.

¹⁵Endliche Zustandsautomaten (*Finite State Machines/FSM*) beschreiben eine endliche Menge von Zuständen, die ein System einnehmen kann. Es können je Zustand beliebig viele Zustandsübergänge definiert sein, um unter bestimmten Bedingungen den Wechsel in einen Folgezustand zu ermöglichen. Beispielsweise kann der Zustand „Hand offen“ bei der erfüllten Bedingung „Objekt gegriffen“ durch den Zustandsübergang „Hand schließen“ in den Zustand „Hand geschlossen“ überführt werden.

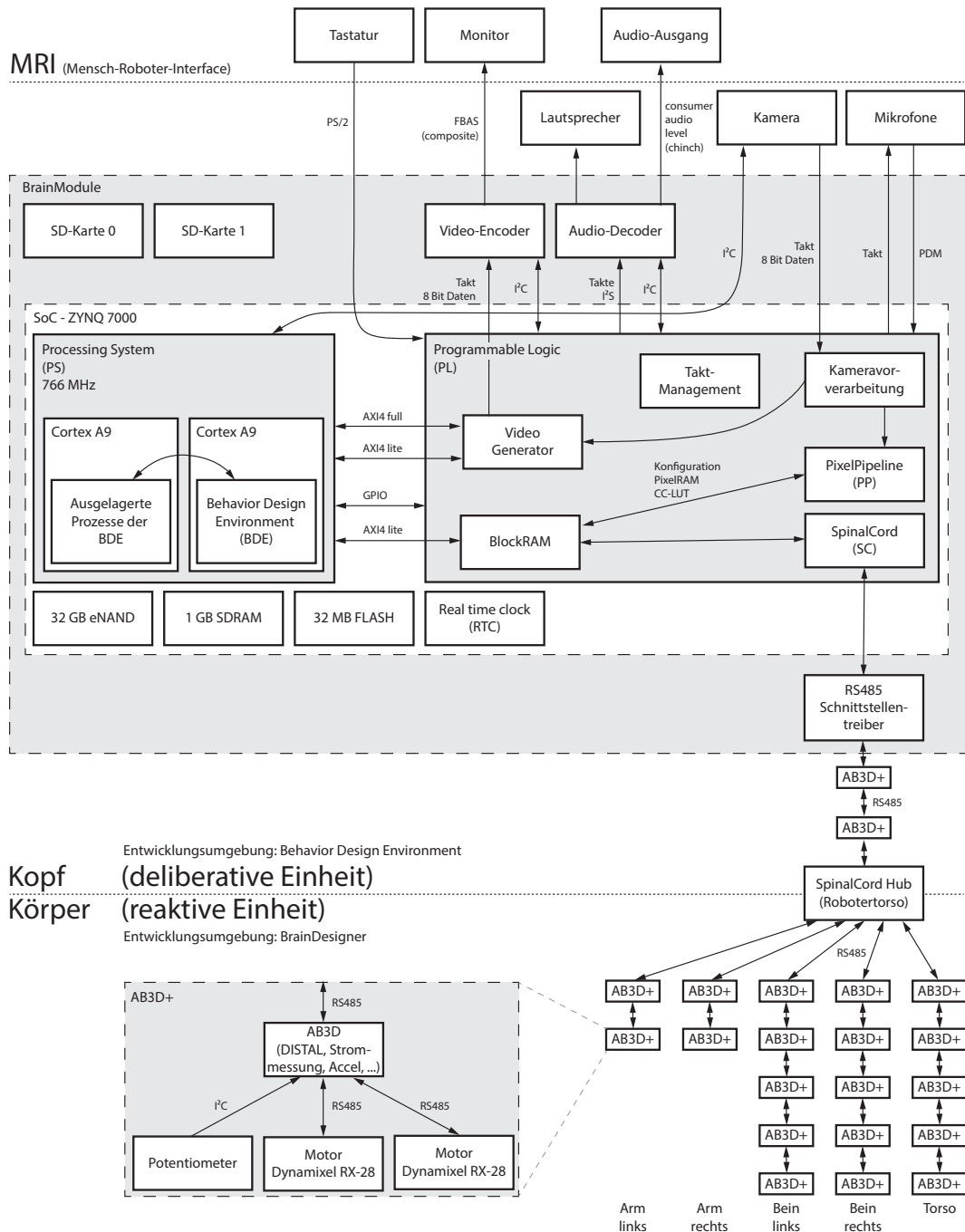


Abbildung 16: Schematische Darstellung des strukturellen Aufbaus und der Signalflüsse von den Komponenten des *BrainModules* zu den *AccelBoard3D (AB3D)* einschließlich Eingabe über Tastatur und Mikrophon beziehungsweise Ausgabe über Monitor, Lautsprecher und *Line-Audioausgang*. Im Interesse der Übersichtlichkeit werden nur die für diese Arbeit relevanten Details gezeigt.

3.3.1 Taktmanagement der deliberativen Einheit

Für komplexe Systeme wie einen humanoiden Roboter ist das *Timing* der Signale von entscheidender Bedeutung.

Durch korrupte Daten, *Glitchese*¹⁶ oder driftende Signale können in leichten Fällen zwar nur unkritische Informationen verloren gehen, in schlimmeren Fällen allerdings kann der Roboter umfallen und sich selbst oder seine Umgebung beschädigen.

Beim Myon gibt es nur einen extern nicht beeinflussbaren Takt: den Kameratakt. Die Taktung der *BDE*-Ausführung erfolgt analog zur bestehenden DISTAL mit 100 Hz. Daher leiten sich die zeitkritischen Takte streng vom 28,375 MHz-Takt der Kamera im Auge ab, damit das komplette visuelle System (siehe Kapitel 3.3.2) sowie die 100 Hz-Hauptschleife der *BDE* auf die Kamerabilder synchronisiert sind¹⁷. Fällt die Kamera oder deren Ausgabesignal aus, wird innerhalb von 10 ns auf einen Notfalltakt mit gleicher Frequenz umgeschaltet. Dieser basiert auf einem parallel existierenden 125 MHz Takt aus dem ebenfalls auf dem Board befindlichen Ethernet-Chip. Sobald der Kameratakt wieder für mindestens 256 ns stabil ist, stellt das System die Taktquelle zurück zur Kamera und synchronisiert sich erneut zum eintreffenden ersten Halbbild.

Zur Analyse der Performance der Notfalltaktumschaltung wurde die Signalleitung der Kamera mechanisch und nicht entprellt getrennt. Bei ungefähr 90% der Umschaltvorgänge hat der *MMCM*¹⁸ den *Locked*-Zustand nicht verlassen. Signifikante Frequenzabweichungen oder Jitter¹⁹ im Ausgangssignal waren nicht zu erkennen. Abbildung 17 zeigt zwei exemplarische Messungen, bei denen der Kameratakt auf den Notfalltakt und zurück geschaltet wird (CH1) mit dem zugehörigen Ausgabetakt von 27 MHz (CH2). Fällt der Kameratakt aus oder weicht signifikant von der Soll-Frequenz ab, erfolgt die Umschaltung auf den Notfalltakt so schnell, dass dies weder von der *BDE* noch von den DISTAL-Komponenten erfasst werden kann. An die *BDE* als deliberative Komponente erfolgt zeitgleich eine Fehlermeldung. Der sichere Betrieb des Roboters ist in dieser Hinsicht gewährleistet.

¹⁶Glitch bezeichnet kurzzeitig falsche Zustände in logischen Schaltungen, da die Signallaufzeiten in den einzelnen Logikgattern nie gleich sind. Alternativ wird dieser Effekt auch als *Race Condition* bezeichnet.

¹⁷Im 32 Bit breiten Timingvektor wechselt Bit 10 mit 100 Hz seinen Zustand. Auf dieses Bit ist die *BDE* synchronisiert.

¹⁸Beim *Mixed-Mode Clock Manager (MMCM)* handelt es sich um eine *PLL (Phase Locked Loop)* mit erweiterten Funktionen des *SoC*-Herstellers Xilinx.

¹⁹*Jitter* bezeichnet eine unerwünschte zeitliche Schwankung der Frequenz infolge fehlerhafter Synchronisation.

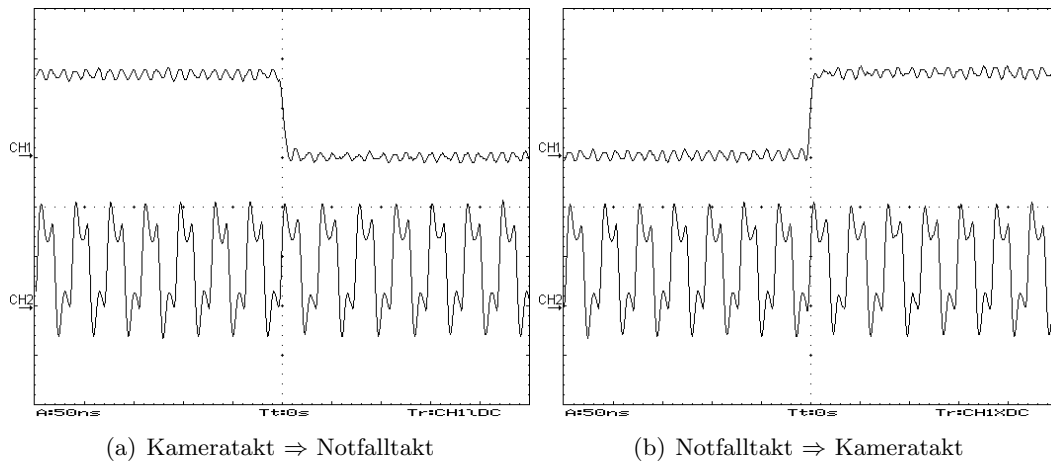


Abbildung 17: Darstellung des 27 MHz-Takts beim Abbruch des Kamerataktes zur Generierung des Haupttakts von 162 MHz. (CH1) Gültigkeit des Kamerataktes (*active high*), (CH2) 27 MHz Ausgabetakt

Abbildung 18 zeigt den Taktbaum in der *Programmable Logic (PL)*, welche im Wesentlichen das *FPGA* umfasst. Das *Processing System (PS)* mit den beiden Cortex A9 Prozessoren hat einen eigenen Takt von 766 MHz.

Aus dem 162 MHz Takt wird der 32 Bit breite Timingvektor erzeugt. Mit diesem Vektor ist es möglich, die *BDE* und alle dort implementierten Routinen starr mit der Hardware zu synchronisieren und beispielsweise zu detektieren, wann ein neues Kamerabild verfügbar ist und ob es sich dabei um das gerade oder ungerade Halbbild handelt. Alle Signale zwischen den Taktdomänen wurden zur Vermeidung von Metastabilitäten²⁰ über zwei serielle FlipFlops in die neue Taktdomäne überführt. Unabhängig davon gibt es einen 200 MHz Takt, der im *PS* des *SoC* erzeugt wird. Damit werden das *SpinalCord*-Modul sowie die Mikrofone im Roboterkopf getaktet. Die vollständige Synchronisation dieser Takte mit dem Kameratakt ist vorbereitet und wird folgen. Dazu wird der gesamte 200 MHz-Taktpfad (Abbildung 18) entfernt und an den Timingvektor angeschlossen.

²⁰Metastabilität bedeutet das zeitweise Verharren in einem undefinierten Zwischenzustand zwischen zwei stabilen Zuständen, beispielsweise bei einem Flipflop. In diesem Fall soll vermieden werden, dass sich der Eingang eines Data-FlipFlops (D-FF) zeitgleich zum Takteingang ändert.

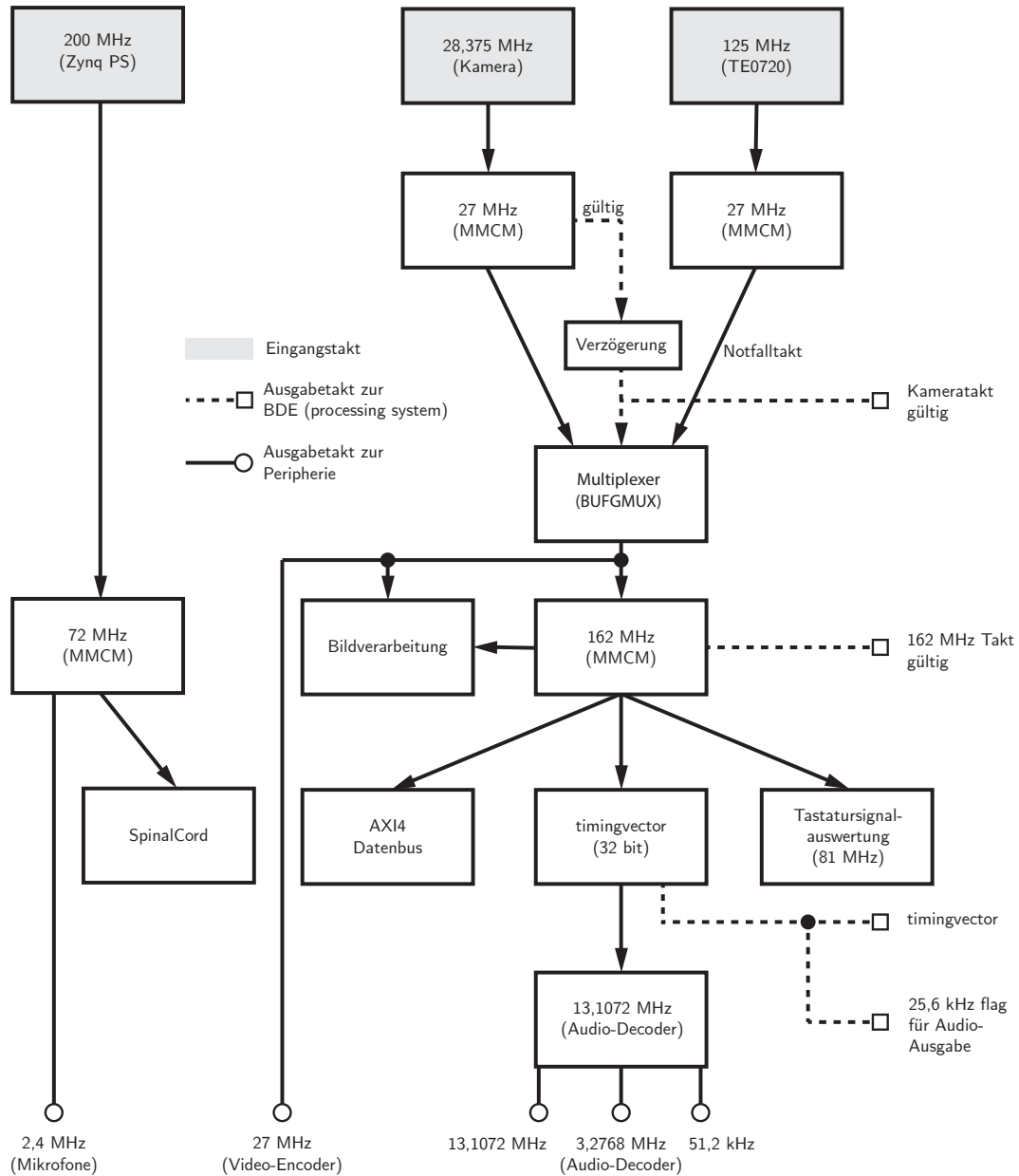


Abbildung 18: Taktmanagement im *FPGA* des *BrainModule*. Der Multiplexer (BUFGMUX) ist das zentrale Umschaltenelement zwischen dem Kameratakt und dem Notfalltakt. Der 27 MHz Takt aus dem Multiplexer wird für die Bildverarbeitung und als Takt für das Video-Encoder-IC genutzt, sowie als Quelltakt für den 162 MHz Takt, der den Großteil der signalverarbeitenden Elemente im *FPGA* taktet.

3.3.2 Videosignal von der Kamera bis zur Ausgabe

Für die neue Entwicklungsumgebung haben sich auch die Anforderungen an die Videoverarbeitung und -ausgabe geändert. Es gibt zum einen das Kamerasignal, welches als exterozeptives Sensorsignal für die *BDE* vorbereitet werden muss, und zum anderen die grafische Benutzeroberfläche der *BDE*, die auf einem externen Monitor ausgegeben werden muss. Die Verarbeitung der beiden Videosignale erfolgt im *FPGA* des *BrainModules* und läuft damit unabhängig von der *BDE*. Das Signal der *BDE* überlagert bei der Ausgabe auf dem Monitor das originale Kamerabild teilweise oder auch vollständig. Als schnelle Kontrollansicht und zur Überprüfung der Algorithmen hat sich diese Funktion als recht nützlich erwiesen. Als Schnittstelle zwischen Kamera und *BDE* wird die *PixelPipeline (PP)* eingesetzt. Eine besondere Herausforderung war der Umgang mit den unterschiedlichen Auflösungen und Bildraten. Abbildung 19 zeigt vereinfacht den Fluss der Videosignale mit den zugehörigen Auflösungen und Bildraten zwischen den einzelnen Modulen im *FPGA*. Zu unterscheiden ist eine Bildwiederholfrequenz von 25 Hz oder 50 Hz, wobei für die effektiv ausgegebene Bildrate entscheidend ist, ob das Bild in zwei Halbbildern (*interlaced/i*)²¹ oder als Vollbild (*non-interlaced* bzw. *progressive/p*) ausgegeben wird. Wie Abbildung 19 zeigt, arbeitet der Zweig von der Kamera bis zum Video-Encoder *interlaced*, jedoch die Verarbeitung in der *PixelPipeline* und die Ausgabe der *BDE*-Oberfläche *non-interlaced/progressive*.

Die Kamera ist vom Typ 21K-455DIG des Herstellers Videology [Vid B] [Vid C] mit dem Objektiv 32S3620N. Das Objektiv hat eine Brennweite von 3,6 mm, eine feste Apertur²² von F1,8 und einen minimalen Objektstand von 40 cm [Vid A]. Die Kamera arbeitet, wie bereits erwähnt, mit einem Nicht-Standard-Takt von 28,375 MHz. Die Auflösung des Bildsensors der Kamera beträgt 752 x 582 Pixel. Übermittelt werden davon jedoch nur 736 x 572 Pixel mit 25i. An das *FPGA* werden je Vollbild (inklusive Steuersignale und Schwarzschildern) 625 Zeilen übermittelt. Bei 50 Halbbildern pro Sekunde wird alle 20 ms ein Halbbild gesendet. Eine Zeile dauert 64 µs. Bei 28,375 MHz entspricht dies 1816 Werten pro Zeile mit je 8 Bit Informationstiefe. Dem 4:2:2 Format der Farbcodierung im YUV-Farbmodell folgend werden die Farbinformationen in der Reihenfolge Y0–V0–Y1–U0 übertragen.

Für je zwei Helligkeitswerte (Y) wird ein Farbwert (U, V) übertragen. Die Hellig-

²¹Dabei werden für jedes Bild zwei Halbbilder mit 50 Hz im Zeilensprungverfahren gesendet, d.h. das erste Halbbild besteht aus den ungeradzahligem, das zweite aus den geradzahligem Zeilen. Jedes Halbbild hat die halbe vertikale (Zeilenanzahl) und volle horizontale Auflösung (Linienanzahl). Das resultierende Vollbild wird mit 25 Hz ausgegeben.

²²Die Apertur gibt den Durchmesser der Öffnung eines optischen Systems an. Oft werden auch die Begriffe Öffnungsweite oder Blendenzahl benutzt.

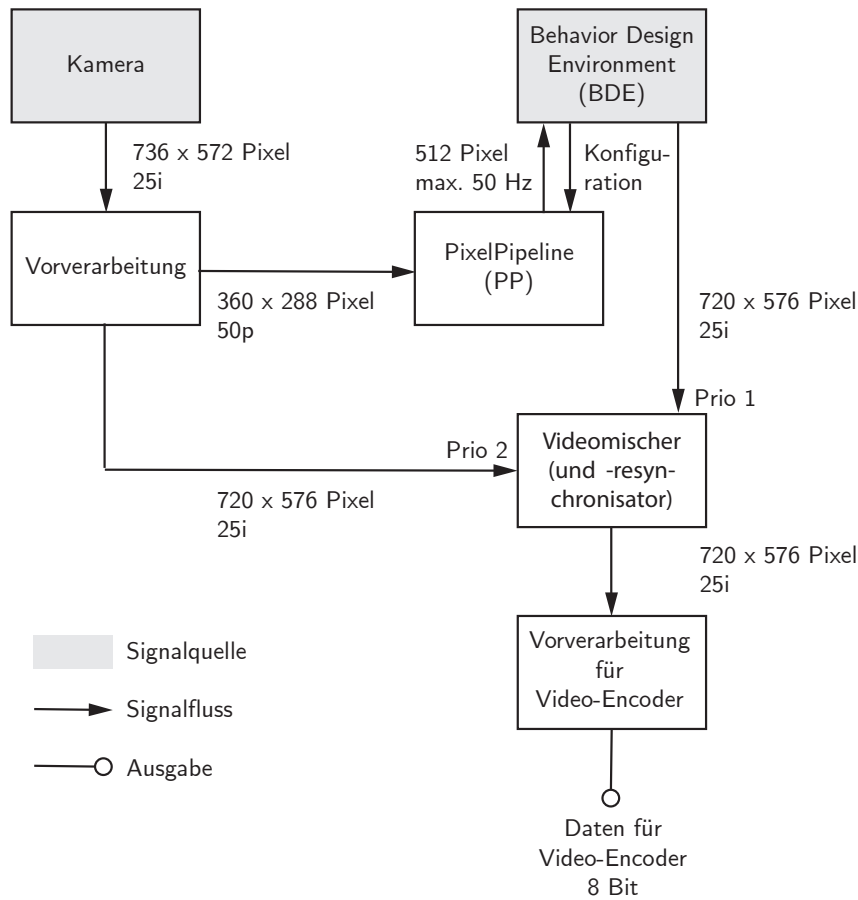


Abbildung 19: Übersicht der Signalflüsse des Videosystems. Signalquellen sind Kamera und *BDE*. Das Kamerabild steht der *BDE* über die *PixelPipeline (PP)* zur Verfügung. Der Videomischer überlagert das Bild der Kamera mit dem Bild der *BDE*. Sendet die *BDE* transparente Pixel, ist das Kamerabild in der Ausgabe sichtbar.

keitsauflösung der Kamera ist damit doppelt so hoch wie die Farbauflösung. Dies ergibt je Zeile 908 Pixel im Format YVYU.

Der ähnlichste vergleichbare Videostandard ist der von der International Telecommunication Union (ITU-R) [ITU07] festgelegte PAL-Videostandard BT.656-5. Dieser definiert eine aktive Bildauflösung²³ von 720 x 576 Pixel, eine Bildrate von 25i und einen Eingangstakt (Pixeltakt) von 27 MHz. Daraus resultieren 1728 Werte im Videosignal mit 864 Pixeln pro Zeile (einschließlich Steuersignalen und Schwarzschildern).

Die *PP* verarbeitet die Halbbilder unabhängig voneinander, bevor diese zu einem Vollbild mit halber Frequenz zusammengefügt werden. Um die Bildrate von 25i auf

²³Die aktive Bildauflösung beschreibt die tatsächlich sichtbaren Pixel des Bildes ohne Steuersignale und Schwarzschilder

50p zu erhöhen, wurde die Kamera per I²C-Schnittstelle²⁴ auf *non-interlaced* umkonfiguriert. Dadurch halbiert sich die aktive Auflösung auf 736 x 286 Pixel. Würden die Halbbilder ungeachtet des Zeilensprungverfahrens unabhängig voneinander verarbeitet, entstünde Zeilenjitter: der Bildinhalt würde zwischen geradem und ungeradem Halbbild um eine Zeile springen. Um das Seitenverhältnis von 5:4 beizubehalten, werden immer zwei Helligkeitswerte zu einem Wert gemittelt. Damit ist die Helligkeitsauflösung gleich der Farbauflösung und der *PP* steht ein Kamerabild mit 360 x 288 Pixeln und 50p zur Verfügung.

Um kompatibel zum ausgebenden Video-Encoder (ADV7390 [Ana18]) zu sein, welcher streng nach dem BT.656-5-Standard [ITU07] sendet, werden die linken und rechten acht Pixel des weitergeleiteten Kamerabildes verworfen. Die unterste Zeile des Bildes wird drei Mal übermittelt, da die Kamera weniger aktive Linien bereitstellt als der PAL-Standard definiert. Dazu wird jeweils eine Zeile zwischengespeichert, bevor diese ausgegeben wird. Die Ausgabe des Bildes auf dem Monitor erfolgt mit einer aktiven Bildauflösung von 720 x 576 Pixel *interlaced* und einer Zeile zeitlichem Versatz. Der Video-Encoder interpretiert die übermittelten YUV-Farbwerte als Werte aus dem $Y C_B C_R$ -Farbraum. Diese beiden Farbmodelle sind sich recht ähnlich, die Farbvektoren sind gemäß Standard lediglich verschieden skaliert. Die Umrechnung hätte in diesem Fall einen verschwindenden Mehrwert und erfolgt deshalb nicht.

Das Videosignal der grafischen Benutzeroberfläche der *BDE* wird über einen AXI4-full-Datenbus (AXI: *Advanced eXtensible Interface*) abgefragt. Im Videomischer wird dieses über das Kamerabild gelegt. Liefert die *BDE* auf einem Farbkanal 0x00 oder 0xFF, werden die betreffenden Pixel transparent und die korrespondierenden Pixel des Kamerabildes werden sichtbar²⁵, was für den Y-, U- und V-Kanal separat geprüft wird. Dies schränkt den Darstellungsbereich nicht ein, da diese beiden Extremwerte deutlich außerhalb des gültigen Intensitätsbereichs nach BT.656-5-Standard liegen. Auf diese Weise kann das originale Kamerabild vollständig oder partiell auf dem Bildschirm angezeigt werden, indem alle drei Kanäle 0x00 oder 0xFF senden. Ein Graustufenbild entsteht, indem nur der Y-Kanal von der Kamera durchgeleitet wird und die *BDE* auf dem U- und V-Kanal ein Grau (0x80)

²⁴I²C steht für *Inter-Integrated Circuit*. Das ist ein etablierter serieller Datenbus zur Kommunikation zwischen integrierten Schaltkreisen.

²⁵Das Kamerabild, welches 1:1 auf dem extern angeschlossenen Monitor ausgegeben wird, wird vollflächig von der Entwicklungsumgebung (*BDE*) überlagert, sodass das Kamerabild nicht mehr zu sehen ist. Bei den Pixeln, bei denen die *BDE* 0x00 oder 0xFF als Farbwert sendet, werden die Pixel nicht mehr von der *BDE* überlagert, sondern das Kamerabild auf dem Monitor sichtbar. Da ein Pixel aus drei Kanälen Y, U und V besteht, für die separat ein Wert 0x00 oder 0xFF gesendet werden kann, ist so ein Mischen des Kamerabildes und der *BDE*-Oberfläche möglich.

sendet. Das Grau kann alternativ auch mit farbigen Informationen aus der *BDE* überlagert werden, um zum Beispiel die Erkennung von Merkmalen im Kamerabild zu überprüfen.

Wie jedes optische System hat allerdings auch die im Myon verbaute Kamera Schwächen, mit denen umgegangen werden muss. So zeigt Abbildung 20 die tonnenförmige Verzeichnung des Kamerabildes. Die im sechsten Kapitel vorgestellten Heuristiken und sensomotorische Schleifen sind jedoch so konzipiert, dass die Verzeichnung nicht kompensiert werden muss.

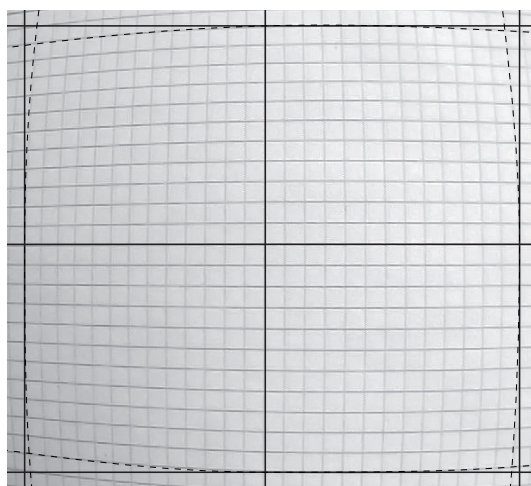


Abbildung 20: Aufnahme eines planaren Karo-Musters mit der Kamera im Roboterkopf. Die durchgezogenen Linien zeigen den idealen und die durchbrochenen Linien den tatsächlichen Verlauf des Karo-Musters. Die tonnenförmige Verzeichnung ist deutlich sichtbar.

PixelPipeline

Die *PixelPipeline/PP* ist die einzige – jedoch sehr mächtige – Schnittstelle zwischen der Kamera im Roboterkopf und der *BDE*. Mit ihr wird das Kamerabild vorverarbeitet. Dazu stehen Funktionen wie die affine Transformation, die Histogrammbildung, die Differenzbildgenerierung sowie eine Einheit zur Farbselektion zur Verfügung. Um das Kamerabild zu verarbeiten, muss die *PP* der Anwendung entsprechend konfiguriert werden.

Die Stärke der *PP* liegt darin, dass bestimmte Berechnungen im *FPGA* des *BrainModules* ausgeführt werden, ohne die Leistungsfähigkeit der *BDE* zu beeinträchtigen. Der *PP*-Durchlauf erfolgt simultan zur Vorverarbeitung des Kamerabildes. Sobald das Bild von der Kamera vollständig übermittelt wurde, liegt das Ergebnis der *PP* umgehend vor und kann von der *BDE* über einen geteilten Bereich im Speicher des *FPGA*, den *BlockRAM*, verarbeitet werden. Der *BlockRAM* ist auf eine Größe von 4 kB bzw. 512 Pixeln begrenzt. Wird ein größerer Bildausschnitt benötigt, muss dieser auf mehrere *PP*-Durchläufe verteilt werden²⁶. Da der *BlockRAM* über zwei unabhängige Schreib- und Lesepointer verfügt, kann er sowohl von der *PP* als auch von der *BDE* unabhängig gelesen und geschrieben werden. Mehrere *PP*-Durchläufe auf dem selben Bildausschnitt sind nicht möglich, da das Kamerabild nicht zwischengespeichert wird. Die Anforderung eines *PP*-Durchlaufs durch die *BDE* kann jederzeit erfolgen und dauert nur wenige Nanosekunden. Ein neues Bild kann immer am Ende eines Kamerabildes abgerufen werden. Dazu ist die *BDE* zeitlich starr mit der Kamera synchronisiert. Da die Kamera alle 20 ms ein Bild liefert, kann bei jeder zweiten Ausführung der 100 Hz-Schleife ein neuer Datensatz über die *PP* abgerufen werden. Wird kein Bild benötigt, muss auch keines abgerufen werden. Da die Kamera über einen globalen *Shutter* verfügt, tritt kein zu kompensierender *Rolling-Shutter*-Effekt auf.

Die Pixel des Kamerabildes können anhand ihrer Position im Bild über eine beliebige affine Abbildung im 512 Pixel großen Speicher abgelegt werden. Die Abbildungen der affinen Transformation umfassen die Operationen Drehung, Spiegelung, Skalierung, Scherung, Verschiebung oder Kombinationen derselben.

²⁶In diesem Fall wird für jeden *PP*-Durchlauf ein anderer Teilausschnitt ausgewählt und die einzelnen Resultate zusammengesetzt (gestitcht). Da jeder *PP*-Durchlauf 20 ms dauert, ist jedoch keine zeitliche Kohärenz der Teilausschnitte gegeben, sodass dies nur bei statischen Szenen oder bestenfalls bei langsamen Bewegungen sinnvoll ist. Allerdings kann dieses Verfahren auch gezielt eingesetzt werden, um einem bewegten Objekt im Bild zu folgen.

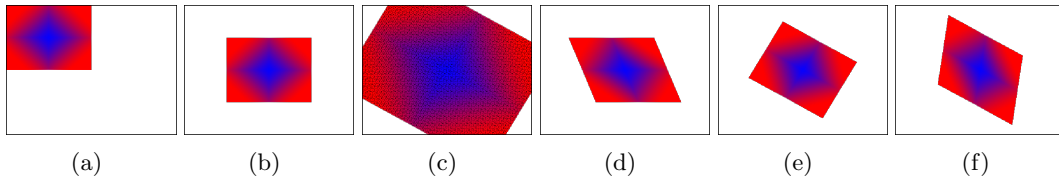


Abbildung 21: Beispiele verschiedener affiner Abbildungen mit der *PixelPipeline*. (a) Stauchung auf 50% Größe, (b) Stauchung auf 50% Größe mit Verschiebung, (c) Vollbild mit Rotation um 30° und Verschiebung, (d) Stauchung auf 50% Größe mit Verschiebung und Scherung, (e) Stauchung auf 50% Größe mit Verschiebung und Rotation um 30° , (f) Stauchung auf 50% Größe mit Verschiebung, Scherung und Rotation um 30° .

Grundsätzlich können die Kamerabilder auf zwei verschiedene Arten verarbeitet werden. Im YUV-Modus oder im *ColorClassifier*-Modus (CC).

YUV-Modus

Die eingehenden Pixelinformationen im YUV-Format werden unverändert verarbeitet. Werden mehrere Pixel aus dem Kamerabild auf den gleichen Ort im Speicher abgelegt, werden die Werte aufsummiert oder subtrahiert und ein Zähler inkrementiert. Wird addiert, lässt sich darüber zum Beispiel der mittlere Farbwert berechnen. Wird subtrahiert, können zum Beispiel die Unterschiede zwischen zwei Kamerabildern (Differenzbild) berechnet werden.

ColorClassifier-Modus

Mit frei konfigurierbaren Vergleichstabellen wird überprüft, ob ein Farbwert in der definierten Farbklasse (ein kubischer Bereich im YUV-Farbraum) liegt. Ist die Überprüfung wahr, wird an der zugehörigen Stelle im Speicher (abhängig von der affinen Abbildung) der Wert inkrementiert. Die 16 Farbklassen können über die *BDE* jederzeit beliebig definiert werden. Dieses Verfahren wurde durch Bruce [BBV00] inspiriert. Über eine zusätzliche Logiktafel können die kubischen Farbklassen optional kombiniert werden. Beispiele für die Konfiguration von Farbklassen zeigt Abbildung 22.

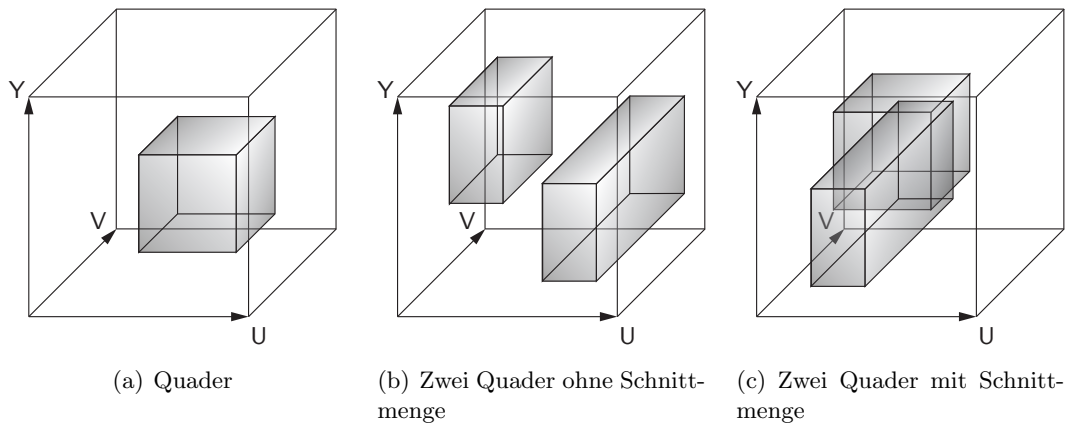


Abbildung 22: Beispiele für die *ColorClassifier*-Konfiguration. Die Quader schließen die zugehörigen Farbwerte einer Farbklass ein. Es können bis zu 16 verschiedene Farbklassen definiert werden. Eine Überlappung sowie eine logische ODER-Verknüpfung der Farbklassen ist möglich.

Die folgende Abbildung 23 zeigt ein Beispiel, bei dem der *CC* so konfiguriert ist, dass in der Ausgabe nur grüne Objekte aus dem Kamerabild gekennzeichnet werden.

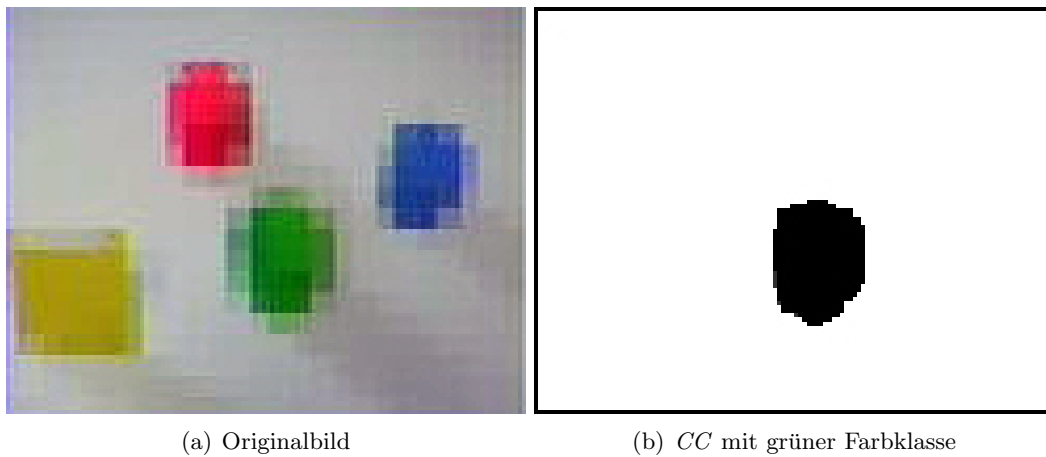


Abbildung 23: Extraktion grüner Objekte mittels Farbklass in der *Color-Classifer*-Modus der *Pixel Pipeline*.

Eine weitere Funktion ist der Histogramm-Modus. Im Histogramm-Modus entscheidet keine affine Abbildung über die Speicherstelle, sondern der aktuelle Wert des Y-, U- oder V-Kanals. Werden mittels *CC* alle Farben zugelassen, so wird bei einem Pixel mit beispielsweise der Luminanz $Y=120$ an der Speicherstelle 120 eine 1 addiert.

Abbildung 24 zeigt die Darstellung der Häufigkeitsverteilung der Y-, U- und V-Werte als Histogramm für jeden Farbkanal. Je Farbkanal wird ein *PP*-Durchlauf benötigt, sodass alle drei Histogramme nach insgesamt 60 ms vorliegen.

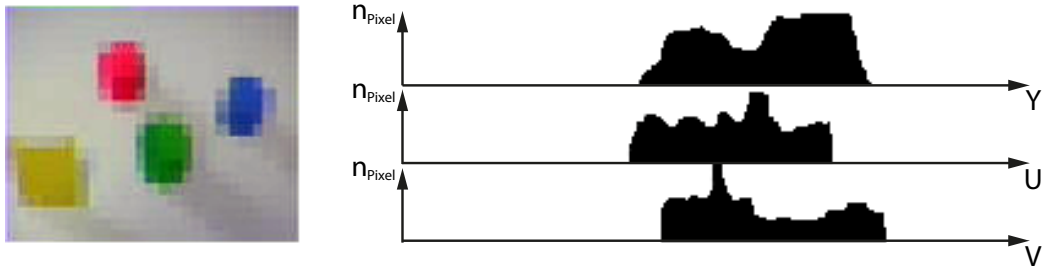


Abbildung 24: Histogramm über alle drei Farbkanäle. Je Histogramm erfolgt ein Durchlauf der *Pixel Pipeline*.

Eine vollständige grafische Übersicht über die Möglichkeiten mit der *PP* ist im Anhang auf Seite 169 zu finden.

3.4 Zusammenfassung

Die praktische Umsetzung dieser Arbeit erfolgt auf der bestehenden Hardware des humanoiden Roboters Myon. Da alle Verhaltensweisen auf dem Roboter entwickelt und getestet werden, ist eine agile Entwicklungsumgebung notwendig. Um den Anforderungen an das deliberative System des Roboters gerecht zu werden, wurde begleitend zu dieser Arbeit die zentrale Recheneinheit neu entwickelt und ein proprietäres Betriebssystem entwickelt. Die Entwicklung der Hardware und der Prozesse auf dem *FPGA* erfolgten im Rahmen dieser Arbeit.

4 Wahrnehmung des Körpers und äußerer Reize

Myon verfügt über 238 zeitgleich abrufbare Sensorwerte, die Informationen zum Zustand seines Körpers bereitstellen – die propriozeptive Sensorik. Zusätzlich hat er drei exterozeptive Sensoren im Kopf: die Kamera sowie je ein Mikrofon im linken und rechten Ohr. Über eine dedizierte Sensorik zur Erkennung von Berührungen verfügt Myon nicht. Diese muss er implizit über die Veränderung der Drehmomente (Motorstrom), steigende Regelfehler in den Gelenken oder extern induzierte Bewegungen (Winkel- und Beschleunigungssensoren) erschließen. Die aus der Verarbeitung der Sensorsignale gewonnenen Informationen ergeben in der Summe die Wahrnehmung (Perzeption) des Roboters.

Teil des Wahrnehmungsprozesses ist das Filtern und gezielte Modifizieren der Sensordaten, um entweder die Wahrnehmung auf bestimmte Reize zu fokussieren, oder auf Grund einer Erwartung den Verarbeitungsprozess anzubahnen und/oder zu beschleunigen. Berührungen können bei der Auswertung wichtiger sein als ein bewegtes Objekt. Wird eine Veränderung in der Bildmitte erwartet, wird die Aufmerksamkeit dorthin gelenkt. Sucht man nach einem roten Objekt, können andere Farben vernachlässigt werden. Dies bedeutet, dass ein Großteil der allozierten (Rechen-)Kapazitäten für diese Aufgabe genutzt wird. Die restlichen Abläufe, etwa die Verarbeitung anderer sensorischer Kanäle oder die Planung von Bewegungen, werden weniger detailliert oder weniger oft betrachtet.

4.1 Taktile Wahrnehmung

Die Haut ist das größte Sinnesorgan des Menschen und sein primärer sensorischer Kanal für Berührungen. Im Rahmen der begleiteten Bachelorarbeit von Miguel Morales [MM15] wurden Experimente durchgeführt, wie der Mensch Objekte und Formen erfühlen kann, wenn er nichts sieht, nichts hört und als Tastelement der Zeigefinger mit einem Kunststoffstab (siehe Abbildung 25) verlängert wird. Die Teilerperimente beinhalteten unter anderem das Erkennen von Alltagsgegenständen sowie das Wahrnehmen und anschließende Reproduzieren eines kleinen Aufbaus aus Klemmbausteinen mit zwei Probandengruppen zu jeweils fünf Personen (kleine Stichprobe) – Gruppe 1 mit dem Zeigefinger und Gruppe 2 mit dem verlängerten Zeigefinger. Gruppe 1 erzielte durchweg gute bis sehr gute Ergebnisse. Gruppe 2 erkannte weniger als die Hälfte der Alltagsgegenstände und konnte nur ungefähr die Hälfte der Aufbauten aus Klemmbausteinen einigermaßen erkennbar reproduzieren und lag damit erwartungsgemäß weit unter der Leistung von Gruppe 1, die die gewohnte Berührungsempfindlichkeit des Fingers nutzen konnte. Dieses nicht repräsentative



Abbildung 25: Verlängerung des menschlichen Zeigefingers mit einem Kunststoffstab für Experimente zur taktilen Wahrnehmung über die mechanische Kopplung zwischen Finger und Stab. Entnommen aus [MM15].

Experiment sagt zwar wenig darüber aus, was mit einem Roboter mit oder ohne Berührungssensorik möglich ist, zeigt aber, dass der Mensch ohne die gewohnte Berührungsempfindlichkeit der Haut stark eingeschränkt ist – zumindest so lange bis er gelernt hat, mit dieser Einschränkung umzugehen.

Da nur wenige Robotersysteme über großflächige Berührungssensoren verfügen, müssen taktile Informationen meist implizit erschlossen werden [NC21]. Dazu gibt es verschiedene Möglichkeiten. So lässt sich anhand des Motorstroms eine Veränderung im Drehmoment bestimmen. Verfügt der Roboter über rücktreibbare Gelenke, ist eine Kollision während der Bewegung oder im Stillstand durch eine unerwartete Veränderung im Motorstrom erkennbar. Wird zudem im Stillstand die Haftreibung im Gelenk genutzt, um die dem Motorstrom entsprechende Antriebskraft zu reduzieren, verbessert sich die Erkennung von extern induzierten Kräften. Dafür eignet sich die Ansteuerung im RASL-Modus, welche in Kapitel 5.1 eingeführt wird. Nachteilig ist, dass das Ein- und Ausschalten des RASL-Modus bei der Beobachtung des Motorstroms aktiv berücksichtigt werden muss.

Ist ein Gelenk relaxiert (Betrieb im *Coast*-Modus), kann dieses frei bewegt werden. Da eine Berührung immer auch eine Kraft ausübt, muss die Berührung nur stark genug sein, um eine Bewegung in einem oder mehreren Gelenken zu verursachen. Bei Myon wird anhand der 66 Winkelsensoren (siehe Tabelle 2 auf Seite 29) die Position der 32 Freiheitsgrade überwacht. Über die zeitliche Ableitung der Winkeldaten wird die Geschwindigkeit der Gelenke berechnet. Anhand dieser beiden Informationen lassen sich die Bewegungen, die aus einer Berührung verursacht wurden, unterscheiden und einordnen. Die Beschleunigungssensorik (Bei Myon sind es 21 Sensoren mit je drei Dimensionen, siehe Tabelle 2 auf Seite 29) ermöglicht im Stillstand (oder bei sehr langsamen Bewegungen) eine absolute Orientierungserkennung der Gliedmaßen. Jeder Sensor bestimmt seine Orientierung in der Umwelt. Der Winkel eines Gelenks berechnet sich aus der Orientierungsdifferenz der beiden über das Gelenk verbundenen Körperteile. Die aus den Beschleunigungswerten berechneten Resulta-

te sind träger als die aus der Winkelmessung über die Potentiometer ermittelten, da die Beschleunigungswerte gerade bei langsamen Bewegungen stärker geglättet werden müssen. Besonders die Ansteuerung über die PMS/PPMS (siehe Kapitel 5.1) erzeugt durch das zugrunde liegende *Chopping* mit einer Frequenz von 25 Hz Vibrationen im gesamten Roboterkörper, die von der Beschleunigungssensorik detektiert wird.

Abbildung 26 auf Seite 51 zeigt die Analyse der taktilen Wahrnehmung über den Servomotor im Ellenbogengelenk des Myon beim Abstellen eines Objekts. Die Kurven von oben nach unten zeigen den Motorstrom des Ellenbogenantriebs, den Integratorwert des PI-Reglers für das Ellenbogengelenk, den über das lokale Potentiometer gemessenen Winkel des Ellenbogens sowie die über die interne Drehmomentschätzung des Servomotors in der Hand bestimmte Griffkraft. Die Sensorwerte sind einheitenlos, da keiner der Sensoren kalibriert ist. Das Abstellen des oberen Objekts führt zu einem Vorzeichenwechsel im Integrator des PI-Reglers (roter Kreis). Die schwarze Kurve des Motorstroms (oben) ist der gleitende Mittelwert der Messwerte. Grau hinterlegt ist das rohe Messsignal, welches durch die PMS mit dem 25 Hz *Chopping* überlagert ist.

Der Versuch kann in folgende acht Phasen aufgeteilt werden:

1. Der Ellenbogen ist gebeugt, das Objekt angehoben.
2. Der Ellenbogen wird zunehmend gestreckt.
3. Der Ellenbogen wird langsamer gestreckt als in Phase 2. Der Integrator lädt sich zu Beginn der Phase etwas auf, um den Lagefehler aus dem Abbremsen der schnelleren Bewegung zu kompensieren.
4. Das Objekt ist abgestellt. Das Ellenbogengelenk muss sich nicht mehr gegen die Gravitation bewegen, da die Gewichtskraft durch das untere Objekt abgefangen wird. Der Integrator des PI-Reglers entlädt sich und erfährt einen Nulldurchgang. Bevor sich der Integrator negativ auflädt und das Objekt mit zusätzlicher Kraft nach unten drückt, wird die Hand geöffnet.
5. Durch das Öffnen der Hand muss das Armgewicht des Unterarms (ungefähr 300 g) wieder aktiv angehoben werden. Der Integrator lädt sich positiv auf, bis der Lagefehler des Ellenbogengelenks kompensiert ist.
6. Die Handfläche dreht nach oben, um das Objekt bei der Seitwärtsbewegung nicht herunterzustoßen.

7. Die Hand wird auf dem Tisch abgelegt.
8. Die Hand drückt auf die Tischoberfläche. Der Integrator lädt sich wieder negativ auf.

Da der Motorstrom immer positiv und aufgrund des Standby-Stroms der integrierten Elektronik nie null ist, zeigt der Vorzeichenwechsel im Integrator des PI-Reglers eindeutig den Lastwechsel an; Referenzwerte, Filterung und Kalibrierung sind nicht notwendig. Da die Bewegungen relativ langsam ausgeführt wurden, verhält sich der Stromverlauf qualitativ ähnlich wie der Integratorwert beziehungsweise wie die Motorspannung vor der PMS. Für dieses Verfahren müssen die Gelenke des Roboters rücktreibbar sein, sodass die Einflüsse der Umwelt auf die Motoren gekoppelt werden. Die Bewegungen dürfen außerdem nicht zu schnell sein, sodass der Integrator auf die Umwelteinflüsse reagieren kann. Beides lässt sich – im Gegensatz zu industriellen Anwendungen – gut mit der humanoiden Robotik vereinen.

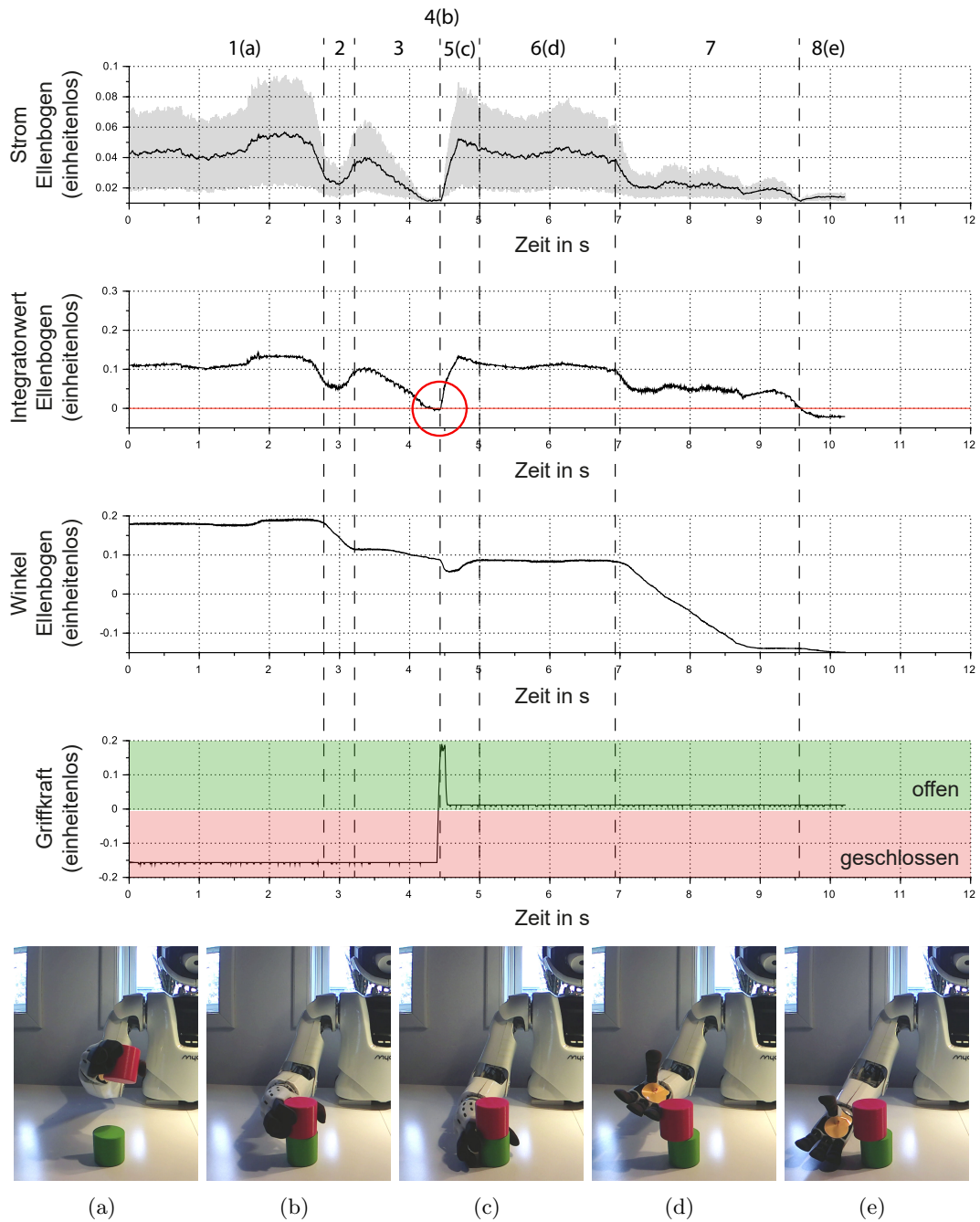


Abbildung 26: Analyse der taktilen Wahrnehmung über den Antrieb im Ellenbogen. Von oben nach unten: Motorstrom des Ellenbogenantriebs, Integratorwert des PI-Reglers für das Ellenbogengelenk, Winkel des Ellenbogens über das lokale Potentiometer, Griffkraft über die interne Drehmomentschätzung des Handservos. Alle Werte sind einheitenlos, da keiner der Sensoren kalibriert ist. Das Abstellen des oberen Objekts führt zu einem Vorzeichenwechsel im Integrator des PI-Reglers (roter Kreis). Die Buchstaben hinter den nummerierten Phasen verweisen auf die Fotos (a) bis (e).

4.2 Visuelle Wahrnehmung

Die visuelle Wahrnehmung erfolgt durch die Verarbeitung und Auswertung der Kamerabilder. Dies beinhaltet sowohl das Erkennen von Formen, Farben und Bewegungen als auch das Wiedererkennen von Elementen. Bei Myon beginnt der Verarbeitungsprozess bereits mit der *PixelPipeline (PP)* im *FPGA* des *BrainModules*. Die Beschränkung der Bildübertragung auf 512 Pixel zwischen der Kamera und der *BDE* ermöglicht die effiziente Nutzung einer selektiven Wahrnehmung. So kann entweder ein Vollbild in geringer Auslösung (zum Beispiel 32 x 16 Pixel, was ungefähr 0,5% der am Eingang der *PP* zur Verfügung stehenden Auflösung von 360 x 288 Pixeln entspricht) oder ein kleiner Ausschnitt aus dem Kamerabild in höherer Auflösung abgerufen werden. Technisch gesehen ist das Volumen an Daten beider Varianten gleich. Die Möglichkeiten in der Wahrnehmung sind jedoch sehr unterschiedlich. Das Vollbild schafft vor allem einen schnellen und sparsamen Überblick. Der Ausschnitt liefert einen variablen Detaillierungsgrad, der nur von der technisch maximal verfügbaren Auflösung limitiert ist, und ignoriert Informationen außerhalb des Ausschnitts.

Das unscharfe Vollbild ist vergleichbar mit dem peripheren Sehen des Menschen, das 99,99% der Netzhaut in Anspruch nimmt [GWB02] und bei dem mehrere Rezeptoren für eine einzelne retinale Ganglienzelle im visuellen Kortex gebündelt werden [SVS20]. Der detaillierte Bildausschnitt wiederum ist vergleichbar mit der *fovea centralis* des Menschen, dem Bereich des schärfsten Sehens, bei dem jeder visueller Rezeptor mit einer eigenen retinalen Ganglienzelle verknüpft ist [SVS20]. Obwohl das Sichtfeld des fovealen Systems verhältnismäßig klein ist, steht ihm beim Menschen gut die Hälfte der visuellen Neuronenmasse zur Verfügung [GWB02]. Taucht unerwartet ein Objekt in der Peripherie auf, wird das zentrale, scharfe Sehen unterdrückt, um die Aufmerksamkeit auf das neue Objekt zu richten. Die Position des Ausschnitts der *PP* kann alle 20 ms variiert werden und ist damit in der Lage, ein vergleichbares visuelles Abtastverhalten wie der Mensch mit schnellen Augenbewegungen (Sakkaden) zu erzeugen. Beim Betrachten einer neuen Szene erfolgen diese beim Menschen alle 250 bis 350 ms [CR08].

Abbildung 27 zeigt mehrere Objekte mit unterschiedlichen Formen und Farben aus der Perspektive des humanoiden Roboters. Die Lichtquelle kommt aus der Richtung der oberen rechten Bildecke. Je nach Objektform und Position wird deutlich, dass selbst Objekte mit einer homogenen Farbe aufgrund der Lichtverhältnisse über die Kamera inhomogen erfasst werden. Dabei lässt sich die Abschattung durch ein anderes Objekt von der eigenen Schattenbildung, welche im geometrischen Schattenraum entsteht, unterscheiden. So steht beispielsweise der grüne Zylinder im Schatten der

anderen Zylinder; der grüne Ball wird oben rechts beleuchtet und ist unten links dunkler, da dieser Teil im geometrischen Schattenraum liegt. Dennoch sind beide Objekte grün. Sie müssen voneinander unterschieden und jeweils vollständig als Objekt erkannt werden.



Abbildung 27: Blick über die Schulter des humanoiden Roboters. Auf dem Tisch liegen Objekte mit unterschiedlichen Farben und Formen. Da der Roboter keine eigene Lichtquelle besitzt, steht zur Objekterkennung nur die gegebene Beleuchtung zur Verfügung – Schattenbildung und Verdeckung sind ein typisches Problem.

Um den verschiedenen Anforderungen an das visuelle System gerecht zu werden, wurden die Ressourcen der *PP* in zehn Operationen aufgeteilt, welche in Tabelle 3 aufgeführt sind. Jede Operation benötigt 20 ms, bedingt durch die Bildrate der Kamera von $25i^{27}$. Nach 200 ms beginnt der Zyklus von vorn. Prinzipiell kann jede Operation während des Betriebs modifiziert oder ersetzt werden. Im ersten (1) und sechsten (6) Schritt, also alle 100 ms, wird als Sicherungsverhalten ein niedrig aufgelöstes Vollbild mit 25×20 Pixeln angefordert, um auf unvorhergesehene Situationen wie Bewegungen oder Veränderungen in der Belichtungssituation reagieren zu können. Im zweiten (2) und dritten (3) Schritt wird ein definierter Bildausschnitt

²⁷Nach der Konfigurationsänderung der Kamera, beträgt die Bildrate 50p bei halbiertter Auflösung. Weitere Details wurden in Kapitel 3.3.2 auf Seite 39 ff. vorgestellt.

mit 25 x 20 Pixeln abgebildet, wovon sowohl ein YUV-Bild (2) als auch alle acht definierten Farbklassen (3) abgerufen werden. Die im vierten (4) und fünften (5) Schritt generierten Ortshistogramme zeigen die horizontale und vertikale Häufigkeit der Pixel für jede Farbkategorie. Der siebente (7) Schritt dient dem Aufbau eines Vollbildes mit nahezu maximaler Auflösung, indem ein 25 x 20 Pixel großer Ausschnitt mit voller *PP*-Auflösung angefordert wird. Dieser Ausschnitt wird über 196 Bilder zu einem maximal aufgelösten Vollbild mit 350 x 280 Pixeln zusammengesetzt (*stitched*). Gibt es Bildbereiche, auf die sich (2) und (3) zum betreffenden Zeitpunkt fokussieren, können diese öfter aktualisiert und in das zusammengesetzte Bild übertragen werden. Das resultierende Vollbild kann durch übergeordnete Routinen genutzt werden, um zu Zeiten von freien Ressourcen („Langeweile“) nach interessanten Merkmalen im Bild zu suchen. Eine vollständige Aktualisierung dauert 39,2s – werden ausgewählte Orte zeitlich höher aufgelöst, entsprechend länger. Im achten (8), neunten (9) und zehnten (10) Schritt werden die Häufigkeitshistogramme des Y-, U- und V-Kanals abgerufen.

Tabelle 3: Nutzung der *PixelPipeline*. Alle 20 ms kann die Konfiguration geändert werden. Die Tabelle zeigt, dass hier zehn verschiedene Konfigurationen nacheinander ausgeführt werden, die sich nach jeweils 200 ms wiederholen.

Reihenfolge	Konfiguration
1	YUV 25 x 20 Pixel Vollbild
2	YUV 25 x 20 Pixel + affine Transformation
3	CCC0 25 x 20 Pixel + affine Transformation
4	Ortshistogramm einer Farbkategorie auf der horizontalen Achse
5	Ortshistogramm einer Farbkategorie auf der vertikalen Achse
6	YUV 25 x 20 Pixel Vollbild
7	YUV 350 x 280 Pixel (<i>stitched</i>)
8	Histogramm auf dem Y-Kanal
9	Histogramm auf dem U-Kanal
10	Histogramm auf dem V-Kanal

Die Kamera verfügt über interne Funktionen, die über die deliberative Einheit (*BDE*) des Roboters gesteuert werden können. Genutzt werden primär der manuell getriggerte Weißabgleich, die manuelle Einstellung der Verschlusszeit (*Shutter*) sowie die analoge Signalverstärkung des Bildsensors (*Gain*).

Der Weißabgleich ist eine fundamentale Einstellung, um zuverlässig Farben zu erkennen. Das Einschalten der Raumbelichtung, der Stand der Sonne, wechselnde Bewölkung oder die Änderung des Standortes beeinflussen die Farbwahrnehmung. Daher muss eine homogene weiße Fläche als Referenz für einen regelmäßigen Weißabgleich zur Verfügung stehen. Für ausgewogene Farbwiedergaben und einer dynamischen Anpassung an die Variation der Lichtverhältnisse sind sowohl ein *Auto-Shutter*, der die Belichtungszeit des optischen Sensors einstellt, sowie ein *AutoGain*, das die analoge Verstärkung der Sensorsignale vor der Analog-Digital-Wandlung bestimmt, sehr praktisch. Sobald jedoch eine starke Lichtquelle wie eine Taschenlampe, ein Scheinwerfer oder die Reflexion an einer glänzenden Oberfläche ins Kamerabild kommen, wird die Helligkeit drastisch reduziert. Im Mittel ist das Bild zwar ausgewogen, jedoch sind dann die Farbinformationen der nicht überbelichteten Bereiche stark reduziert oder gehen sogar komplett verloren. Alle bislang erkannten Objekte können jetzt nur noch über die räumliche Anordnung wiedererkannt werden. Die Begrenzung der Farbtiefe auf 8 Bit je Farbkanal lässt nur wenig Spielraum für eine softwareseitige Anpassung der Helligkeit. Damit die aktuellen Farbklassen der *PP* weiterhin Gültigkeit haben, wird die Belichtungseinstellung der Kamera manuell über die *BDE* konfiguriert. Dabei wird der *Shutter* zur groben Einstellung und das *Gain* zur anschließenden Feinjustierung genutzt. Dies ermöglicht eine verlustfreie Variation an den Rändern des Farbspektrums, die Erhaltung der aktuellen Farbklassen sowie die gezielte Nutzung von *HighKey*²⁸- oder *LowKey*²⁹-Bildern.

Dazu wird der Histogramm-Modus der *PP* genutzt, wie er in Abbildung 24 auf Seite 46 beispielhaft gezeigt wurde. Die Ausgabe der Histogramme für alle drei Kanäle erfordert drei *PP*-Durchläufe und dauert folglich 60 ms.

Die visuelle Wahrnehmung der Umgebung und des Roboters selbst umfasst sowohl die adaptive Anpassung des visuellen Systems als auch die Extraktion von visuellen Merkmalen. Im Rahmen dieser Arbeit werden Merkmale anhand von Farben sowie Ecken und Kanten im Bild extrahiert und als Basis der visuellen Wahrnehmung genutzt. Auf die Nutzung freier Bibliotheken wie beispielsweise OpenCV wurde bewusst verzichtet, um minimalistische Prinzipien zu gewährleisten.

²⁸gezielte Überbelichtung

²⁹gezielte Unterbelichtung

4.2.1 Aktives Sehen

Je stärker ein humanoider Roboter dem menschlichen Vorbild ähnelt, desto wahrscheinlicher ist es, dass er über eine Kamera verfügt und diese in einem zum Torso beweglichen Kopf verbaut ist. Wird die Kopfbewegung genutzt, um die Qualität der Wahrnehmungsergebnisse über die Kamera zu verbessern, wird dies als *aktives Sehen* (*Active Vision*) bezeichnet. Aloimonos [AWB88] formulierte eine Veränderung geometrischer Parameter des Sinnesapparats als Ausgangspunkt für aktives Sehen. In dieser Arbeit wird als Sinnesapparat das System aus der Kamera im Auge, den drei Freiheitsgraden im Hals sowie der affinen Abbildung der PixelPipeline zusammengefasst.

In den folgenden Kapiteln wird die individuelle Kopfbewegung zur adaptiven Anpassung der Blickrichtung als wesentlicher Teil des Wahrnehmungsprozesses dargestellt. Eine häufig eingesetzte Vorgehensweise ist die Zentrierung der interessanten Objekte in der Bildmitte. Dies hat drei Vorteile: (1) Das Objekt kann vollständig erfasst werden, ohne dass es zuvor vollständig wahrgenommen beziehungsweise dessen Grenze zum Hintergrund erkannt wurde. (2) In allen Richtungen wird die maximal mögliche Umgebung erfasst, und (3) der Abbildungsfehler ist minimal, da die Verzeichnung des Kameraobjektives erst zu den Bildecken hin signifikant wird (siehe Abbildung 20 auf Seite 42).

Ein Anwendungsbeispiel hierfür ist der automatische Weißabgleich der Kamera. Da sich die Umwelt von mobilen Robotern stets verändert, ist es von Vorteil, wenn eine weiße Referenzfläche nativ mitgeführt wird. Da Myon weiß verschalt ist, sind seine Armschalen eine brauchbare Referenz und der Weißabgleich kann jederzeit intrinsisch motiviert³⁰ ausgeführt werden. Beispielsweise wenn die ihm durch die Farbe bekannten Objekte verschwinden, obwohl keine Bewegung im Kamerabild wahrgenommen wird.

Dafür übernimmt ein übergeordnetes System kurzzeitig die Kontrolle über einen der beiden Arme und der Arm sowie die Blickrichtung nähern sich einander an, bis der Arm mittig im Kamerabild zu sehen ist. Wie Abbildung 28 zeigt, kann es auch hier auf Grund gerichteter Lichtquellen zur Schattenbildung kommen. Der Weißabgleich, ausgeführt auf der Kamerahardware, erfolgt über einen kleinen Bereich in der Bildmitte und dauert ungefähr 5 s. Um einen Weißabgleich im Schatten zu vermeiden, wird links und rechts von der Mitte geprüft, welche Seite heller ist. Diese einfache Heuristik entscheidet, ob der Arm weiter nach links oder rechts bewegt wird, bis links

³⁰Ein Individuum führt eine intrinsisch motivierte Handlung aus, wenn es diese ohne äußere Einflüsse ausführt. Beim Menschen kann die Motivation Spaß, ein Interesse an etwas oder eine darin gesehene Sinnhaftigkeit sein.

und rechts von der Bildmitte die gleiche Helligkeit erkannt wird. Für Abbildung 28 wurde der Arm leicht nach rechts, also zur Körpermitte, bewegt. Im Anschluss wird der Arm wieder in seine vorherige Pose gebracht und die Kontrolle an das untergeordnete System zurückgegeben.



Abbildung 28: Weißabgleich der Kamera an der Verschalung des linken Arms. Der Ausschnitt oben links zeigt das Bild der Kamera im Auge. Die Arminnenseite ist auf Grund des Schattens durch die hier gegebene Belichtungssituation nicht geeignet für einen Weißabgleich.

Da Myon über eine monokulare Kamera ohne Tiefeninformationen verfügt, hat Wurst in der begleiteten Bachelorarbeit [Wur16] untersucht, wie sich mit einer horizontalen Auflösung von 360 Pixeln der Abstand von Objekten durch eine Kopfdrehung um die Hochachse (*Yaw*) schätzen lässt. Abhängig vom Objektabstand, der Winkeldifferenz der Kopfdrehung und vor allem von der Qualität, mit der die Verschiebung des Objektes im Kamerabild detektiert wird, wurde der Objektabstand im einstelligen Zentimeterbereich genau erkannt. Diese Interpolierungen sind sehr wertvoll, um grundsätzlich einzuschätzen, ob die Objekte überhaupt im Arbeitsbereich der Hand liegen (siehe Abbildung 44 auf Seite 81) und wie sie ungefähr angeordnet sind. Für ein zuverlässiges Greifen der Objekte ist diese Schätzung jedoch zu ungenau. Shichao et. al. bestätigen ungenaue 3D-Informationen auf Basis eines erweiterten *Dual Quaternion*-Ansatzes, bedingt durch ein monokulares Kamerabild [DMY⁺21].

4.2.2 Objekterkennung durch Farbselektion

Objekte können über verschiedenartige Merkmale erkannt und voneinander unterschieden werden: Form, Größe, Position, Farbe, Muster, Textur und vieles mehr. Je detaillierter das gesuchte Objekt beschrieben wird und je mehr Informationen zu beispielsweise Kanten, Ecken und Farben vorliegen, desto genauer kann ein Objekt identifiziert werden. Diese Arbeit setzt auf eine einfache und zuverlässige Objekterkennung unter der Randbedingung, dass die Objekte anhand ihrer Farbe erkannt und verfolgt werden.

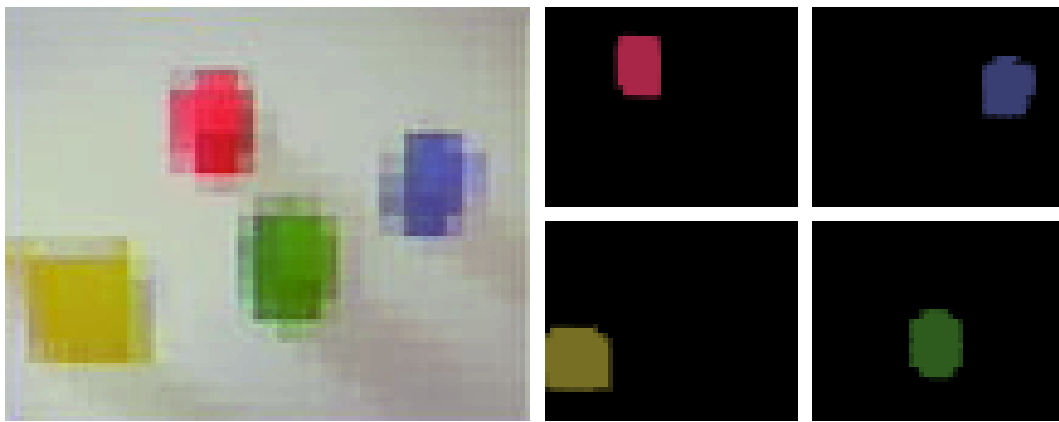


Abbildung 29: Erkennung von Objekten mittels Farbklassen. Für jedes Objekt wird eine eigene Farbklasse angelegt, sodass der jeweils relevante Bereich im Farbraum abdeckt ist.

Wird ein Objekt ins Sichtfeld gestellt, erfährt das Bild eine deutliche Änderung. Sobald keine Änderung mehr stattfindet, wird die Differenz sowohl im Bild selbst als auch in den Histogrammen der einzelnen Kanäle analysiert. Dazu wird das aktuelle Bild mit einem stark tiefpassgefilterten Bild verglichen. Ermittelt wird der größte zusammenhängende Bereich im Histogramm der Farbkanäle; dieser wird für die Definition der neuen Farbklasse in den *ColorClassifier* geschrieben. Damit leichte Helligkeitsschwankungen toleriert werden, wird der Luminanzkanal (Y) nach oben und unten etwas erweitert; die damit einhergehende insignifikante Erweiterung des inbegriffenen Farbraums wird dabei in Kauf genommen. Das YUV-Differenzbild zeigt, wo sich das neue Objekt befindet beziehungsweise ob das Objekt entfernt wurde. Abbildung 30 zeigt, wie Myon das Hinzukommen und Entfernen von Objekten erkennt und zu jedem neuen Objekt eine Nahaufnahme speichert. Dieses visuelle Kurzzeitgedächtnis hat eine Kapazität von acht Elementen und ist vergleichbar mit dem *Visual Short Term Memory (VSTM)* aus der *Theory of Visual Attention (TVA)* nach Bundesen [Bun90]. Ist im *VSTM* kein freier Platz vorhanden, muss ein anderes Objekt entfernt werden. Werden Objekte bewusst entfernt, wird die Farbklasse

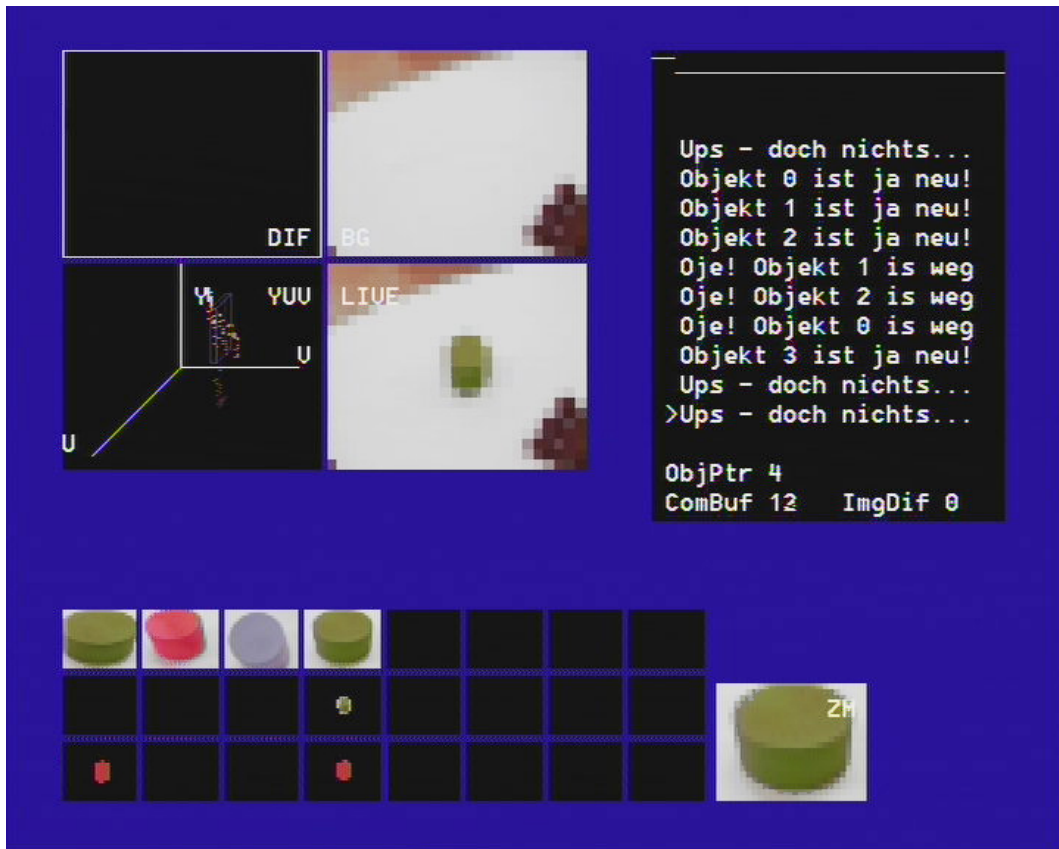
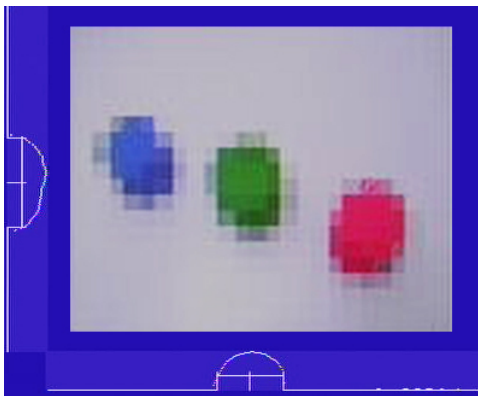


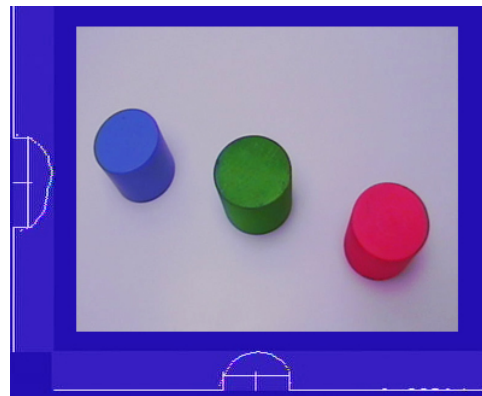
Abbildung 30: Erkennung von neuen und entfernten Objekten im Kamerabild. Zum betreffenden Zeitpunkt befindet sich nur ein grünes Objekt im Bild; drei andere Objekte sind bereits aus dem Bild entfernt worden. (Oben links) Vier Ansichten: das aktuelle Kamerabild; ein tiefpassgefiltertes Bild, um Bildänderungen zu verdeutlichen; ein Differenzbild zwischen den beiden zuvor genannten Bildern; die Darstellung der Pixel des zuletzt erkannten Objekts im YUV-Farbraum. (Oben rechts) Der innere Monolog des Roboters, inspiriert durch den virtuellen Roboter PSI nach Dörner [Dö01] [Dör02]. (Unten) Darstellung der acht Objekte im Kurzzeitgedächtnis mit der Position, einer Nahaufnahme und der Resultate der dazu definierten Farbklasse. Zu jedem Objekt im Kurzzeitgedächtnis wird eine Nahaufnahme gespeichert. Diese steht noch so lange zur Verfügung, bis sie von der Aufnahme eines neuen Objekts überschrieben wird.

des Objekts direkt verworfen. Die Nahaufnahme bleibt jedoch so lange im Speicher, bis das *VSTM* an dieser Speicherstelle überschrieben wird, damit Objekte, die nach kurzer Zeit wieder ins Bild gestellt werden, über einen Vergleich der gespeicherten Nahaufnahmen mit einer gewissen Wahrscheinlichkeit als bekannte Objekte wahrgenommen werden können. Der rechts in Abbildung 30 gezeigte innere Monolog des Roboters ist durch den virtuellen Roboter PSI nach Dörner [Dö01] [Dör02] inspiriert. Er hat zwar für diese Arbeit keine direkte Bedeutung, soll aber für folgende Arbeiten zur Interaktion und Selbstreflexion dienen.

Um Objekten visuell grob zu folgen, ist die maximale *PP*-Auflösung von 512 Pixeln ausreichend. Diese Arbeit nutzt typischerweise eine Auflösung von lediglich 500 Pixeln aus 25×20 Pixeln, um das Aspektverhältnis der Kamera von 5:4 zu erhalten und die Inhalte nicht zu verzerren. Muss die Positionsauflösung erhöht oder die Position des Objektschwerpunkts berechnet werden, ist es sinnvoll, die Ortshistogramme für eine Farbklasse zu bilden. Dies ist effizienter und wesentlich schneller, als die Auflösung eines Bildes zu erhöhen. Hierfür steht die volle Auflösung mit 360×288 Pixeln zur Verfügung, wie sie von der *PP* bereitgestellt wird. Die Berechnung erfordert jeweils einen *PP*-Durchlauf für *x*- und *y*-Richtung und dauert daher insgesamt 40 ms. Für die Erkennung der Objektkanten werden die Histogramme zunächst normiert und dann ausgehend vom Maximum bei einer Intensität von 0,4 in beide Richtungen abgesteckt. Die Objektmittle wird als arithmetischer Mittelwert der beiden Objektkanten berechnet. Eine beispielhafte Ausgabe in der *BDE* zeigt Abbildung 31.



(a) Darstellung des auf 25×20 Pixel reduzierten Vollbildes



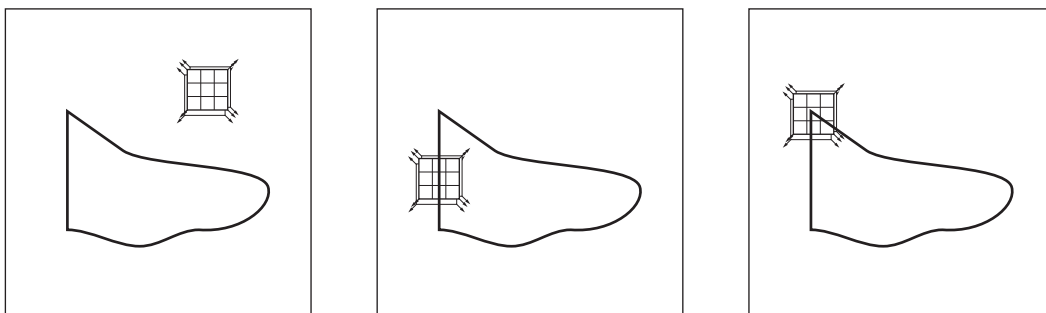
(b) Darstellung des originalen Kamerabildes

Abbildung 31: Erkennung von Objekten mittels Ortshistogrammen nach Farbklassen mit voller Kameraauflösung – hier am Beispiel eines grünen Objektes.

Mit der *PP* als Flaschenhals der visuellen Information ist die Wahrnehmungsfähigkeit stärker eingeschränkt als dies technisch notwendig wäre. Dies ermöglicht jedoch, dass das Wahrnehmen von Sinneseindrücken in das aktive Handeln eingebettet werden kann. Der Roboter nimmt nur das wahr, was für sein Handeln zum betreffenden Zeitpunkt relevant ist und vermeidet damit, eine Fülle unnötiger Informationen verarbeiten zu müssen.

4.2.3 Objektmerkmale durch Eckenerkennung

Für die Aufgabe des Nachzeichnens in Kapitel 6.4 wird zur Erkennung der Zeichenbereiche sowie der Linien eine Eckenerkennung eingesetzt. Hierfür hat sich der *Harris-Detektor* bewährt, da dieser effizient zu implementieren und rotationsinvariant ist. Der Detektor von Harris und Stephens [HS88] baut auf dem Detektor von Moravec [Mor80] auf, der zu einer der ersten bekannten Detektoren für Ecken und Kanten gehört.



(a) Der Fensterinhalt ändert sich bei keiner der Bewegungen → Leerer Bereich.

(b) Der Fensterinhalt ändert sich nicht bei Bewegungen in vertikaler Richtung → Kante.

(c) Der Fensterinhalt ändert sich bei Bewegungen in allen Richtungen → Ecke.

Abbildung 32: Diese drei Beispiele zeigen ein einfaches fiktives Kamerabild. Der Harris-Detektor analysiert das Bild mit einem Ausschnitt (Fenster) aus 3 x 3 Pixeln. Entscheidend ist, wie die Bewegung des Fensters mit der Änderung der Pixelwerte innerhalb des Fensters zusammenhängt.

Abbildung 32 illustriert das Funktionsprinzip: Der Detektor analysiert das Bild, indem jeweils nur ein Ausschnitt von 3 x 3 Pixeln („Fenster“) betrachtet wird. Abhängig davon, welche der neun Pixel sich bei einer Verschiebung des Fensters ändern, wird erkannt, ob das Fenster einen leeren Bereich, eine Kante oder eine Ecke abdeckt.

Die Intensität der Veränderung $E(u, v)$ bei einer Verschiebung um (u, v) Pixel lässt sich für ein Graustufenbild $I(x, y)$ mit folgender Formel[HS88] berechnen:

$$E(u, v) = \sum_{x, y} w(x, y) (I(x + u, y + v) - I(x, y))^2 \quad (1)$$

Dabei ist $w(x, y)$ eine Fensterfunktion, die bestimmt, wie die einzelnen Pixel des Fensters in die Berechnung eingehen. Diese Funktion ist in der vorliegenden Arbeit mit $w(x, y) = 1$ definiert, kann aber zum Beispiel auch eine diskrete Gaußverteilung sein, sodass der mittlere Pixel einen höheren Einfluss hat als die umliegenden Pixel.

Da die Ableitung ein Maß für die Änderung ist, wird für kleine Verschiebungen (u, v) die Berechnung mittels Taylor-Entwicklung erster Ordnung vereinfacht. Dies ergibt:

$$E(u, v) \approx \sum_{x,y} w(x, y) (I_x^2 u^2 + I_y^2 v^2 + 2I_x I_y uv). \quad (2)$$

In Matrixform gilt:

$$E(u, v) \approx \begin{pmatrix} u & v \end{pmatrix} \mathbf{M} \begin{pmatrix} u \\ v \end{pmatrix} \quad (3)$$

mit

$$\mathbf{M} = \sum_{x,y} w(x, y) \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \quad (4)$$

Die Eigenwerte (λ_1, λ_2) von \mathbf{M} sind ein Maß für die größte und kleinste Änderung. Sind beide Eigenwerte klein, so gab es nur geringe oder keine Änderungen im Fenster. Ist ein Eigenwert deutlich größer als der andere, dann gab es in eine Richtung eine große Änderung, in die Richtung orthogonal dazu jedoch nicht. Dies ist ein Merkmal für eine Kante. Sind beide Eigenwerte groß, so ist das ein Zeichen für eine Ecke. Die Nutzung der Eigenwerte von \mathbf{M} macht dieses Verfahren zusätzlich rotationsinvariant [SMB00]. Abbildung 33 illustriert den Zusammenhang zwischen den Eigenwerten und den Merkmalen im Fenster.

Dieser Zusammenhang lässt sich zu einem Maß R_H formalisieren.

$$R_H = \det(\mathbf{M}) - k(\text{trace}(\mathbf{M}))^2 \quad (5)$$

$$R_H = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (6)$$

mit der empirischen Konstante $k \in \{x \mid 0,04 \leq x \leq 0,06\}$.

Es gilt:

$R_H \ll 0$	Kante
$R_H \gg 0$	Ecke
$R_H \approx 0$	Fläche

Im Anschluss an die Berechnung der R-Matrix empfiehlt sich eine *Non-Maximum Suppression/NMS*, um überlappende Detektionen zu eliminieren.

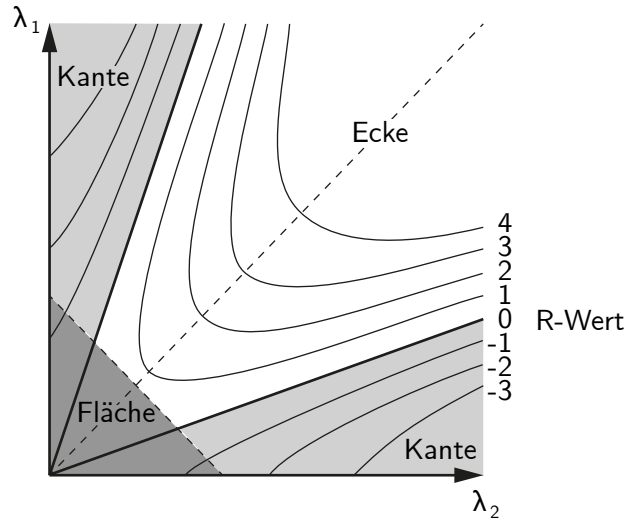


Abbildung 33: Zusammenhang zwischen Ecken, Kanten und Flächen zu den Eigenwerten λ_1 und λ_2 von \mathbf{M} (Nachempfunden aus [HS88]).

Da Ecken in realen Kamerabildern nur selten perfekt scharf sind, ist der Harris-Detektor größenvariant. Je weiter man das Bild vergrößert, desto mehr wird aus einer Ecke eine Rundung (runde Kante). Da in dieser Arbeit bewusst die Bildauflösung stark reduziert wird, stellt dies kein Problem dar.

Eine verbreitete Weiterentwicklung des Harris-Detektors ist der *Shi-Tomasi-Detektor*. Er steht beispielsweise in der freien Bibliothek OpenCV (*Open Source Computer Vision*) zur Verfügung. Wie die zugehörige Publikation *Good Features to Track* [Shi94] beschreibt, liegt dessen Stärke in einer optimierten Eckenerkennung für die Verfolgung von Merkmalen. Der Unterschied liegt in der Auswertung der Eigenwerte zum Maß R_S :

$$R_S = \min(\lambda_1, \lambda_2) \quad (7)$$

Übersteigt R_S einen definierten Wert, bedeutet das, dass beide Eigenwerte größer sind als dieser Wert. Wie Abbildung 33 zeigt, ist dies ein valides Maß für eine Ecke. Im Folgenden der Harris-Detektor eingesetzt. Obwohl dieser allgemein als weniger zuverlässig in der Eckenerkennung angesehen wird, liefert er bei der Aufgabe zum Nachzeichnen von Linien auf 25 x 20 Pixel großen Bildern ausreichend gute Ergebnisse. Abbildung 34 zeigt fünf Beispiele. Jeder rote Punkt zeigt einen erkannten Pixel und überlagert das auf Graustufen reduzierte Kamerabild. Da die zu erkennenden Linien im folgenden Kapitel 6 zur Visuomotorik immer eine quadratische Umrandung haben, die in den Randbereichen der Kamerabilder zu sehen ist, werden alle

vom Bildrand ins Bild kommende Inhalte inhibiert. Eine Erkennung von Kanten an den Bildrändern (Vergleiche Abbildung 34 a–b) wird damit unterbunden. Die positiven Auswirkungen einer *Non-Maximum Suppression/NMS* zeigen die Abbildungen 34 c–d, wo eine Ecke im Bild auch nur als ein Pixel erkannt wird.

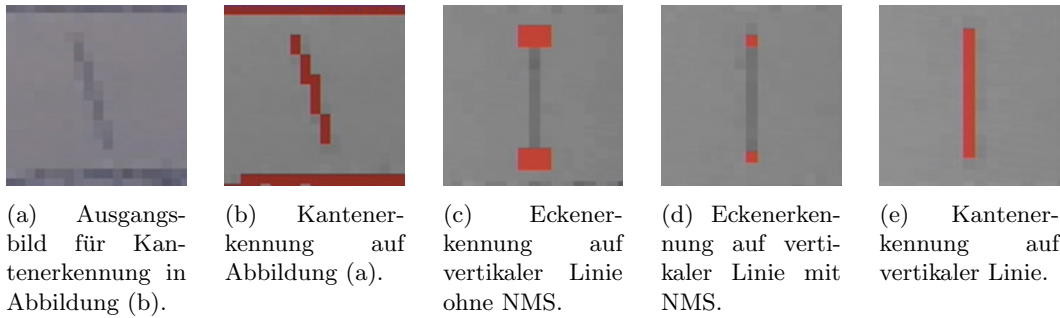


Abbildung 34: Anwendung des Harris-Detektors auf fünf Kamerabilder mit jeweils 25 x 20 Pixeln. Das Bild wurde in Graustufen gewandelt.

4.3 Zusammenfassung

Die Wahrnehmung des Körpers und äußerer Reize erfolgt über zwei Sinneskanäle – taktil und visuell. Für die taktile Wahrnehmung einer Kollision hat sich am Beispiel des Ellenbogengelenks gezeigt, dass die Analyse des I-Anteils des PI-Reglers im Gelenk valide Informationen über den Belastungszustand und Kollisionen liefert. Insbesondere durch den Nulldurchgang bei einem Lastwechsel ist dieses Verfahren unabhängig von einer Kalibrierung und von Referenzwerten.

Die visuelle Wahrnehmung basiert auf der selektiven Wahrnehmung, welche durch die Konfiguration der *PixelPipeline* gezielt erzwungen wird. Die Bildausschnitte beziehungsweise die Art der Bildinformationen müssen aktiv angefordert werden. Durch die Anpassung der Kameraausrichtung, der affinen Abbildung durch die *PixelPipeline* sowie durch die dynamische Konfiguration der Kameraparameter (*Shutter*, *Gain* und Weißabgleich) handelt es sich um einen aktiven Sehprozess (*Active Vision*). Dank der weißen Verschaltung des Roboters kann der Weißabgleich des Kamerabildes jederzeit intrinsisch motiviert ausgeführt werden. Die Erkennung von Objekten erfolgt anhand von YUV-Differenzbildern sowie Intensitäts- und Ortshistogrammen, die in einem visuellen Kurzzeitgedächtnis gespeichert werden. Die Erkennung von Ecken wird durch eine effiziente Implementation eines Harris-Detektors realisiert.

5 Körperbeherrschung als Basis von Bewegungen

Jedes robotische System weist eine materielle Gestalt auf. Mit anderen Worten: es hat einen Körper. Um mit diesem Körper die Umwelt zu erkunden (explorieren) oder sie sogar zu verändern (manipulieren) muss es zunächst die Fähigkeit besitzen, den eigenen Körper zu kontrollieren. Dazu müssen die einzelnen Gelenke Bewegungen ausführen, die zum gewünschten Ziel führen.

Je nach Ansatz können diese Bewegungen zuvor geplant, also deliberativ, oder reaktiv auf eine Änderung in den sensorischen Daten ausgeführt werden. Die folgenden Unterkapitel stellen isolierte sensomotorische Schleifen vor, auf denen die visumotorischen Handlungen aus Kapitel 6 aufbauen werden. Da diese Handlungen zwar in einem größeren Kontext geplant werden, jedoch immer auch Reaktionen auf Änderungen einzelner Sensordaten sind, werden beide Ansätze zu einer hybriden Variante kombiniert.

Künstliche neuronale Netze als Programmiermethode

Der Begriff der künstlichen neuronalen Netze bezeichnet ein biologisch inspiriertes mathematisches Modell als Basis für eine Vielzahl von Algorithmen der künstlichen Intelligenz (KI) und der Robotik [AJO⁺18]. Das Konzept wurde erstmals von McCulloch und Pitts im Jahr 1943 [MP43] vorgestellt.

In dieser Arbeit wird die reaktive Bewegungssteuerung des Roboters in Anlehnung an künstliche neuronale Netze (siehe DISTAL in Kapitel 3.2) umgesetzt.

Ein typisches Modell hat n Eingänge x_i ($i = 1, \dots, n$) und einen Ausgang y . Jeder Eingang wird individuell über einen Faktor w_i gewichtet. Aus der Summe der gewichteten Eingänge, welche auch als Neuronenaktivität bezeichnet wird, wird mit der Transferfunktion f das Ausgabesignal y berechnet.

Die Updategleichung eines zeitdiskreten Neurons³¹ lautet:

$$y_{t+1} := f \left(\sum_{i=1}^n w_i x_{i,t} \right), \quad w_i, x_{i,t} \in \mathbb{R}, t \in \mathbb{N} \quad (8)$$

³¹In dieser Arbeit werden Mikrocontroller zur Berechnung der neuronalen Netze eingesetzt. Die Berechnungen erfolgen daher ausschließlich Zeit- und Werte-diskret (16 Bit).

Als Transferfunktionen werden in dieser Arbeit die Identitätsfunktion, der Tangens hyperbolicus (\tanh), die Signumfunktion (sgn) sowie Umschalter und weitere Spezialfälle eingesetzt. Die Transferfunktionen sind in diesem Fall über einen ausgewählten Assembler-Befehlssatz der eingesetzten Mikrocontroller auf den AB3Ds programmierbar.

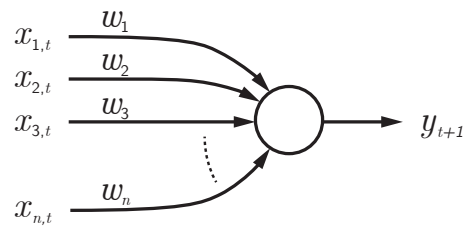


Abbildung 35: Grafische Darstellung eines einzelnen Neurons. Die Eingänge x werden mit den Gewichten w multipliziert und summiert. Der Ausgabewert y wird anhand einer Transferfunktion, angewandt auf die gewichtete Eingangssumme, berechnet.

Die grafische Darstellung eines Neurons (Abbildung 35) erfolgt in der Regel durch einen Kreis. Dieser symbolisiert die Summenbildung und die Transferfunktion. Die Eingänge werden jeweils mit einem in das Neuron gerichteten Pfeil, die gewichtete Synapse, dargestellt, der Ausgang durch einen Pfeil aus dem Neuron heraus. Gibt es mehrere Ausgangspfeile, haben diese den gleichen Wert. Gewichte werden als Wert an die Eingangspfeile geschrieben. Die Darstellung ist aufgebaut wie ein gerichteter Graph.

Werden mehrere Neuronen verkettet, entsteht ein neuronales Netz. Ist der gerichtete Graph zyklisch, spricht man von einem rekurrenten neuronalen Netz. Falls nicht, handelt es sich um ein *Feed-Forward-Netz*.

Die in dieser Arbeit eingesetzten neuronalen Netze werden mit einer Frequenz von 100 Hz aktualisiert. Da es sich um eine Struktur handelt, die wesentlich vielfältiger und flexibler als ein klassisches künstliches neuronales Netz ist, werden sie im Folgenden als reaktive Netze bezeichnet. Jedes Element des Netzes verzögert die Weiterleitung des Signals um einen Taktzyklus (10 ms). Das gilt beispielsweise für ein Neuron samt Transferfunktion oder eine Multiplikation, jedoch nicht für die Synapsen. Ein *Feed-Forward-Netz* mit fünf seriellen Neuronen (Ebenen) benötigt damit 50 ms, bis der Ausgang auf eine Änderung im Eingang reagiert. Dieser Ablauf sichert die Echtzeitfähigkeit der reaktiven Netze³².

³²Bei Bedarf kann durch eine manuelle Modifikation der Neuronenstruktur von diesem Timing abgewichen werden und es können mehrere Neuronen je 10 ms Taktzyklus in Echtzeit berechnet werden.

5.1 Regelung der Gelenke

Zur Regelung der teilweise mit mehreren Aktoren angetriebenen (multiaktuierten) Gelenke des Myon wird das antagonistische modellfreie Steuerungskonzept von Siedel [Sie15] eingesetzt und mit verschachtelten (kaskadierten) Reglern angesteuert.

Für die Arm- und Greiferbewegung stehen je Arm sechs Motoren zur Verfügung: zwei für Schulter-*Pitch* sowie je einer für Schulter-*Roll*, Ellenbogen, Handgelenk und die Greifbewegung selbst. Für das Schulter-*Pitch*-Gelenk mit den zwei starr parallel gekoppelten Motoren ist eine antagonistische Steuerung möglich. Dies bedeutet, dass die beiden Motoren, während sie das Gelenk aktuieren, immer auch leicht gegeneinander arbeiten, um eine Vorspannung im Gelenk zu erzeugen. Siedel [Sie15] hat folgende Vorteile identifiziert:

- Das Getriebespiel (auch als Lose bezeichnet) kann aufgehoben werden.
- Dynamische Bewegungen werden gedämpft.
- Der Totbereich der eingesetzten Aktoren kann kompensiert werden. Werden die Motoren so eingebaut, dass die Nullstellung der integrierten Winkelsensoren um mehr als 60° versetzt ist, kann durch eine Sensorfusion der volle Bereich von 360° sensorisch abgedeckt werden, ohne dass ein externer Winkelsensor notwendig ist.
- Der schnelle, gegenphasige Wechsel zwischen Haft- und Gleitreibung von zwei Aktoren führt zu einer Reduktion der Gesamtreibung und somit zu einem verbesserten Antriebsverhalten (Keine *Stick-Slip*-Effekte sowie ein erhöhtes Abtriebsmoment).
- Da die antriebsinterne Haftreibung höher ist als die Gleitreibung, kann die Energieaufnahme in Ruhepositionen deutlich verringert werden, ohne dass sich das Gelenk bewegt. Dies hat neben dem geringeren Energiebedarf zudem den positiven Nebeneffekt, dass sich die Aktoren weniger stark erwärmen und die Position länger gehalten werden kann.

Siedels Steuerung basiert auf drei Konzepten: Grundlage ist die PPMS (Phasenverschobene PulsModulierte Steuerung) und die DAAV (Dynamische Asymmetrische Anpassung der Vorspannung) in Kombination mit dem RASL-Modus (Reduktion der Antriebsleistung bei Statischer Last) bei einer Frequenz von 25 Hz.

Ziel der PPMS ist es, das Regelverhalten zu verbessern, indem die Nichtlinearität

im Umkehrpunkt der Bewegung reduziert sowie der negative Einfluss der Getriebe-
reibung verringert oder sogar kompensiert wird. Dazu werden die Motoren phasen-
verschoben über eine Pulsmodulierung angesteuert. Mit der DAAV werden Nichtli-
nearitäten im Umkehrpunkt des Motors durch eine dynamische und asymmetrische
Anpassung der Vorspannung realisiert.

Der zusätzliche RASL-Modus reduziert die Antriebsleistung im statischen Lastfall,
also wenn die Winkelgeschwindigkeit nahezu null ist, zum einen um Energie zu spar-
en, zum anderen um die Erwärmung der Aktoren zu reduzieren. Insbesondere bei
rücktreibbaren Gelenken, wie alle beim Myon, ist die Erwärmung der Aktoren ein
entscheidender Faktor für die Betriebszeit eines Gelenks.

Laut Siedel [Sie15] wird mit der hybriden Steuerung, im Gegensatz zur einfachen
proportionalen Steuerung, dank reduzierter Reibungseffekte das Antriebsmoment
um 30% gesteigert und der Energieverbrauch um 22% gesenkt.

Für die vier einfach aktuierten Gelenke im Schulter-*Roll*-Gelenk, im Ellenbogen, im
Handgelenk und im Greifer wird statt der PPMS die PMS (PulsModulierte Steue-
rung) mit RASL-Modus eingesetzt. Bei der PMS handelt es sich im Wesentlichen um
die Ansteuerung mittels *Chopping*. Die DAAV ist bei einem Motor nicht einsetzbar,
da eine Vorspannung im Gelenk mindestens zwei Motoren erfordert.

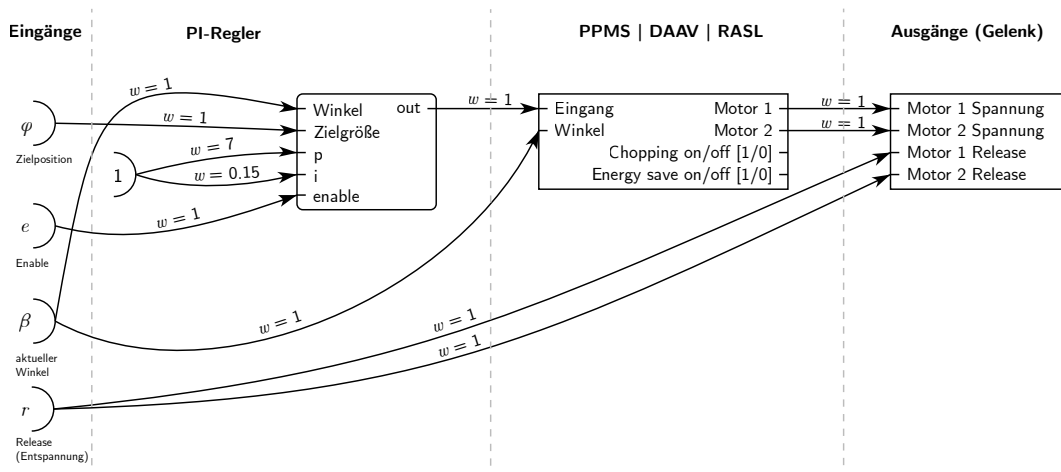


Abbildung 36: Reaktiver PI-Regler mit hybrider Steuerung am Beispiel des Schulter-*Pitch*-
Gelenks mit zwei Motoren in vereinfachter Darstellung. Je nach Heuristik kann entweder das re-
aktive oder das deliberative System den Sollwinkel des Gelenks vorgeben; den Regler inhibieren,
sodass das Gelenk in der aktuellen Stellung verharrt; sowie das Gelenk relaxieren.

Die Pulsmodulation sowie die Regelung der Gelenke erfolgt ausschließlich auf dem
reaktiven System und beeinflusst damit nicht die Visuomotorik. Die Regelung der
antagonistischen Steuerung erfolgt durch klassische PI-Regler.

Regelung des Handgelenks

Der Winkel des Handgelenks wird reaktiv unter Zuhilfenahme der Beschleunigungssensorik im Unterarm geregelt, sodass unabhängig von der Handposition im Raum die Orientierung der Handfläche immer gleich bleibt. Welchen Winkel die Handfläche zur Gravitationsrichtung einnimmt, wird deliberativ dem Greifprozess angepasst. Benötigte Größen aus dem reaktiven System sind der Wert der vertikalen Beschleunigung im Unterarm a sowie der aktuelle Winkel des Handgelenks φ_W . Das deliberative System kann die Regelung über ein binäres *Flag* e aktivieren oder deaktivieren, sowie die Reaktivität der Regelung v und den Offset der Handgelenkstellung s definieren. Der Offset s des Handgelenkwinkels ist abhängig vom absoluten Winkel des Handgelenks φ_W . Da dieser Wert über das servointerne Potentiometer bestimmt wird, ist dieser auch beim Austausch des Arms gleich und unterliegt keinem signifikanten Drift.

Formal ist der Vorgang wie folgend beschrieben:

$$f_t = v_{t-1} (-0,3 + s_{t-1} + \varphi_{W,t-1} - 2,5 a_{t-1})$$

$$m_t = \text{PMS} (p_{t-4})$$

$$p_t = p_{t-1} + 3 k_{t-1} - 2 k_{t-2}$$

$$k_t = e_{t-1} (2 (\varphi_{W,t-2} - \varphi_{W,t-1}) + f_{t-1})$$

m ist der Ausgabewert, der überlagert durch die PMS an den Motor gesendet wird. Der Ausgabewert f ist ein Fehlermaß zur Kontrolle durch das deliberative System. Die Zeitindizes t zeigen, mit welcher Verzögerung die Signale in die jeweilige Berechnung eingehen. Diese sind für eine erfolgreiche Reproduktion dieses und der folgenden reaktiven Netze von entscheidender Bedeutung. Neben konstruktiv bedingter Verzögerungen, die durch die sequentielle Berechnung der Neuronen auf der Zielhardware entstehen³³, gibt es auch gezielte Verzögerungen. Diese treten zum Beispiel bei Rückkopplungen in Regelschleifen auf.

Abbildung 37 zeigt die Struktur zur Regelung der Handgelenkdrehung im reaktiven System als grafisches Netz. Die als Kreis dargestellten Neuronen haben eine lineare Transferfunktion. Oft werden diese auch durch z^{-1} gekennzeichnet.

³³Dies gilt insbesondere für die Berechnung der reaktiven Netze in dieser Arbeit. Grundsätzlich könnte, unter Berücksichtigung der sich ändernden Timings, davon abgewichen werden.

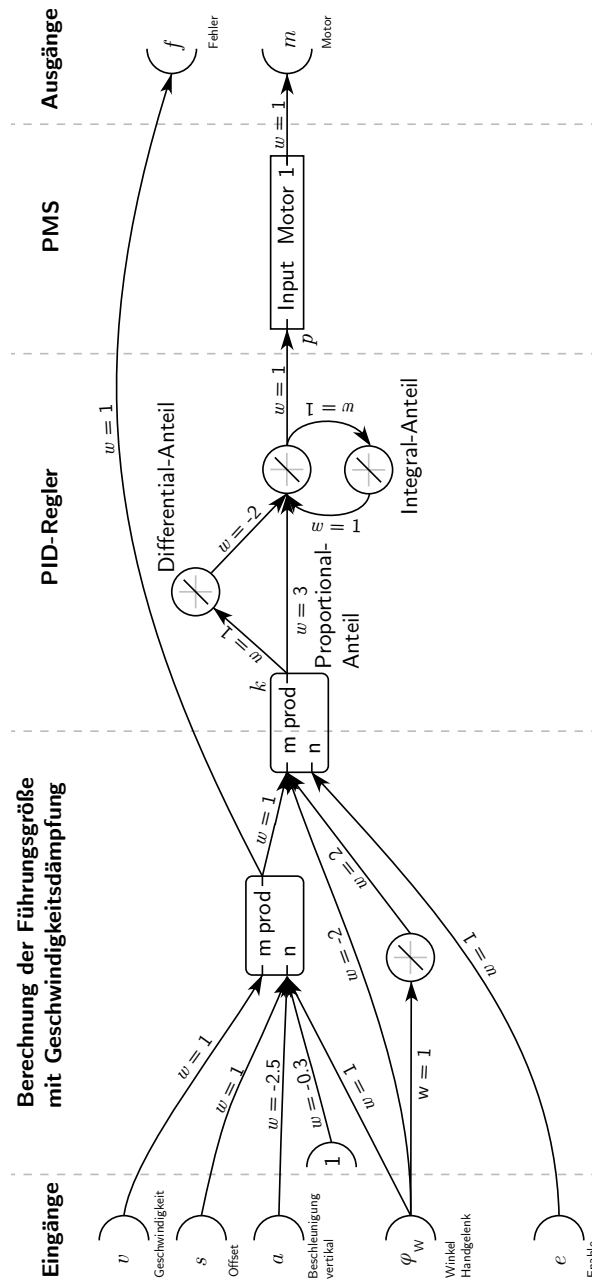


Abbildung 37: Reaktives Netz zur Regelung des Handgelenks mit einer Aktualisierungsrate von 100 Hz. Das Netz kann zwischen den Eingängen und den Ausgängen in drei Abschnitte unterteilt werden. Der (1) Berechnung einer Führungsgröße für einen (2) PID-Regler mit einer (3) Puls-Modulierten Überlagerung (PMS). Das deliberative System wird über ein Fehlermaß f über den aktuellen Zustand der auf Beschleunigungswerten basierten Regelung informiert.

Regelung der Griffkraft

Auch die Regelung der Finger erfolgt im reaktiven System. Dabei gibt es die Zustände „Hand offen“ ($o = 1$) und „Hand geschlossen“ ($o = 0$). Das Öffnen erfolgt positionsgeregelt, bis die Finger vollständig geöffnet sind. Das Schließen erfolgt kraft- und positionsgeregelt. Die Finger beugen bis zu einer definierten Griffkraft i_m , welche über den Motorstrom i definiert ist, oder bis sie maximal geschlossen sind. Sowohl der Zustand o als auch die maximale Griffkraft i_m können über das deliberative System gesteuert werden. Dieses erhält zudem eine Rückmeldung, ob die maximale Griffkraft i_a oder die maximale Griffposition p erreicht ist. Die aktuelle Griffkraft i sowie die aktuelle Griffposition φ_H ist parallel über den *SpinalCord* verfügbar. Dieser Motor wird ohne PMS angesteuert.

Formal ist das reaktive Netz wie folgt definiert:

$$\begin{aligned}
 p_t &= \begin{cases} 1 & \text{für } (-g_{t-1}) > (C - 0,07) \\ 0 & \text{für } (-g_{t-1}) \leq (C - 0,07) \end{cases} \\
 i_{a,t} &= (h_{t-1} - i_{m,t-1}) > 0 \\
 m_t &= 4e_{t-1} (g_{t-1} - \varphi_{H,t-1}) \\
 h_t &= 0,9 h_{t-1} + 0,1 i_t \\
 a_t &= \begin{cases} g_{t-1} & \text{für } (p_{t-1} + i_{a,t-1}) > 0 \\ C & \text{für } (p_{t-1} + i_{a,t-1}) \leq 0 \end{cases} \\
 b_t &= \begin{cases} a_{t-1} & \text{für } o_{t-1} > 0 \\ R & \text{für } o_{t-1} \leq 0 \end{cases} \\
 g_t &= \begin{cases} \varphi_{H,0} & \text{für } t = 0 \\ g_{t-1} & \text{für } (t > 0) \text{ und } (g_{t-1} = b_{t-1}) \\ g_{t-1} + \left((0,01 - (0,007o_{t-2}))e_{t-2} \right) & \text{für } (t > 0) \text{ und } (g_{t-1} < b_{t-1}) \\ g_{t-1} - \left((0,01 - (0,007o_{t-2}))e_{t-2} \right) & \text{für } (t > 0) \text{ und } (g_{t-1} > b_{t-1}) \end{cases}
 \end{aligned}$$

mit den Konstanten für die geöffnete Handstellung $R = 0$ und die geschlossene Handstellung $C = 0,55$. Diese sind abhängig vom absoluten Winkel des Fingermotors φ_H . Da dieser Wert über das servointerne Potentiometer bestimmt wird, ist dieser auch beim Austausch der Hand gleich und unterliegt keinem signifikanten Drift. Abbildung 38 zeigt die Struktur zur Regelung der Griffposition und -kraft im reaktiven System als grafisches Netz.

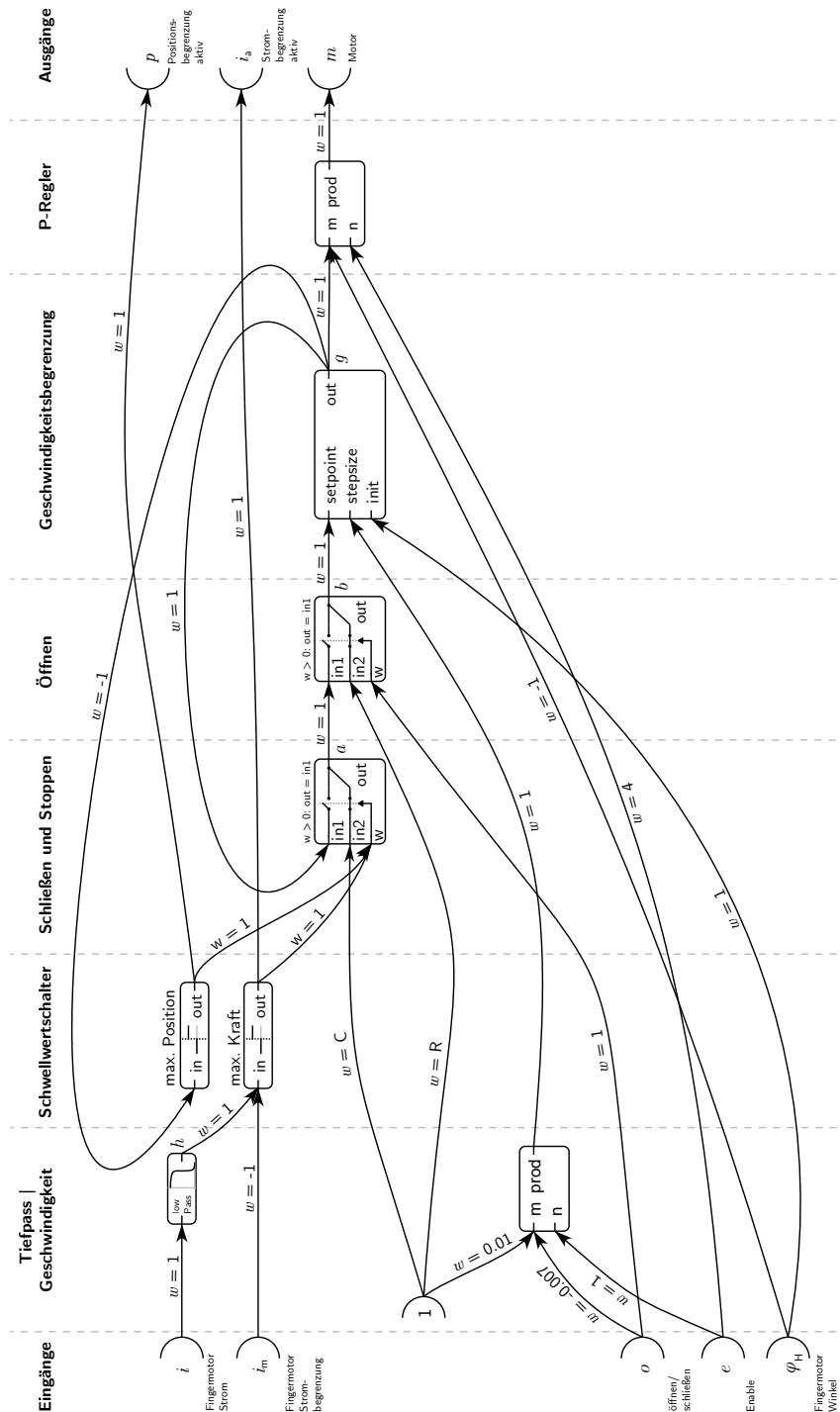


Abbildung 38: Reaktives Netz zur Regelung der Finger (Griffkraft und -position) mit einer Aktualisierungsrate von 100 Hz. Das Netz kann zwischen den Eingängen und den Ausgängen in sechs Abschnitte unterteilt werden. Einen (1) Tiefpass zur Glättung der Motorstromwerte, sowie der Berechnung der richtungsabhängigen Greifgeschwindigkeit, (2) zwei Schwellwertschalter zur Erkennung der Positions- und Kraftgrenzen, den (3) Schließvorgang, den (4) Öffnungsvorgang, die (5) Geschwindigkeitsbegrenzung sowie einen abschließenden (6) Proportionalregler. Das deliberative System wird über eine aktive Strom- oder Positionsbegrenzung informiert.

Regelung der Kopfneigung (Roll)

Für die Kopfneigung ist das *Roll*-Gelenk im Hals zuständig; es kann auf drei verschiedene Arten über das reaktive System geregelt werden.

Es kann (1) eine Wunschneigung über das deliberative System vorgegeben werden, die Neigung über (2) die Beschleunigungssensorik im Kopf so geregelt werden, dass das Kamerabild immer horizontal (parallel zum Horizont) bleibt oder (3) mittels der Information der visuell erfassten Objektbreite den Kopf so neigen, dass die Objektbreite im lokalen Minimum gehalten wird. Dies ist sinnvoll, um die Bildachsen (x und y) an den Schultergelenken (*Pitch* und *Roll*) auszurichten, indem der Unterarm des Roboters als Objekt interpretiert wird. Ist die Breite des Unterarms minimal, so liegt dieser vertikal im Bild und eine Bewegung des Schulter-*Pitch* Gelenks entspricht nahezu der vertikalen Bewegung (auf der y-Achse) im Kamerabild. Auf Grund der Anordnung der Halsgelenke impliziert eine Kopfdrehung um die Hochachse (*Yaw*), mit Ausnahme des exakt horizontalen Blicks im *Pitch*-Gelenk, immer auch eine Drehung des Kamerabildes um die *Roll*-Achse. Der Drehpunkt liegt unterhalb des Bildmittelpunkts. Eine Kopfdrehung (*Yaw*) um ca. $\pm 10^\circ$ bei einem konstanten Nickwinkel (*Pitch*) von -24° führt zu einer absoluten *Roll*-Bewegung (Rotation des Horizontes) von ca. $\pm 4,5^\circ$. Abbildung 39 zeigt eine Messung der absoluten Kopfneigung (*Roll*) über die Kopfdrehung (*Yaw*) bei einem Blick nach unten ($\varphi_P = -24^\circ$).

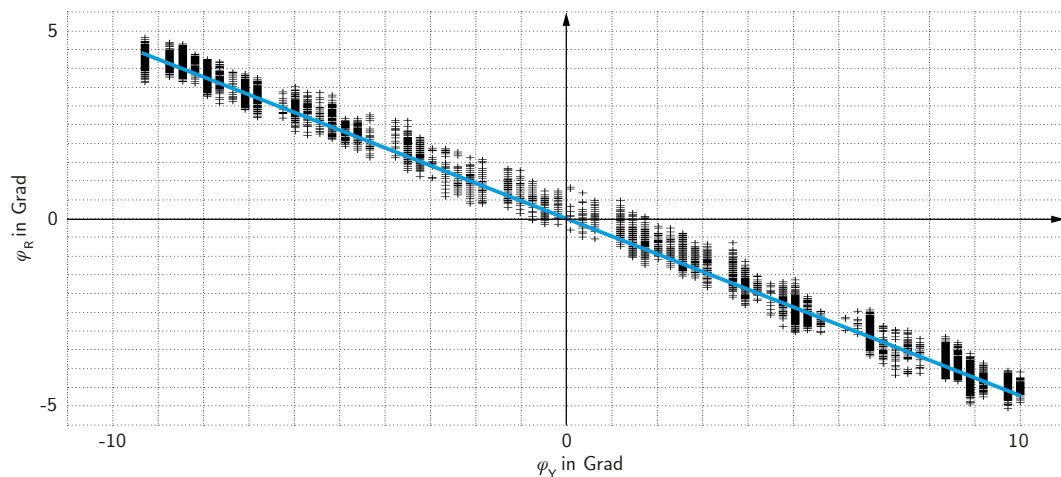


Abbildung 39: Kopfneigung um das *Roll*-Gelenk φ_R aufgezeichnet über die Kopfdrehung um das *Yaw*-Gelenk φ_Y bei einem Blick nach unten ($\varphi_P = -24^\circ$).

Dies wird bei Bedarf mittels der Beschleunigungssensoren im Kopf (Variante 2) durch das reaktive System des Roboters kompensiert, indem das *Roll*-Gelenk so geregelt wird, dass die vertikale Achse des Kamerabildes immer mit der Erdbeschleuni-

gung fluchtet. Abbildung 40 zeigt die Struktur zur Regelung der Kopfneigung (*Roll*) im reaktiven System.

Formal lassen sich die drei Varianten auch in drei Formelsätze aufteilen. Die Ausgabewerte der Einheit zur horizontalen Ausrichtung ω_a sowie der Einheit zur Ausrichtung für die minimale Objektbreite h speisen ihr Signal bei Bedarf in die Einheit für die deliberative Regelung ein und dominieren dessen Verhalten.

Deliberative Positionsregelung:

$$\begin{aligned}
 m_t &= \text{PMS} \left((c_{t-6} (e_{s,t-8} - e_{a,t-7})) - (e_{a,t-6} \omega_{a,t-6}) \right) \\
 c_t &= 0,12 (c_{t-2} + 12 (n_{t-2} + h_{t-2} - \varphi_{R,t-2})) \\
 n_t &= \begin{cases} \varphi_{r,0} & \text{für } t = 0 \\ n_{t-1} + 0,001 \operatorname{sgn}(\varphi_{s,t-1} - n_{t-1}) & \text{für } t > 0 \end{cases}
 \end{aligned}$$

Horizontale Ausrichtung:

$$\begin{aligned}
 \omega_{a,t} &= \omega_{a,t-2} - j_{t-2} + 3j_{t-1} \\
 j_t &= e_{a,t-1} (2\varphi_{R,t-2} - 2\varphi_{R,t-1} - 0,025 (12,5a_{t-2} + \varphi_{R,t-2}))
 \end{aligned}$$

Ausrichtung für minimale Objektbreite:

$$\begin{aligned}
 \omega_{o,t} &= 0,5 (k_{t-1} (e_{o,t-3} > 0)) \\
 \varphi_{o,t} &= \varphi_{o,t-2} + 0,004 \omega_{o,t-1} \\
 h_t &= \varphi_{o,t-1} e_{o,t-1} \\
 k_t &= \begin{cases} 1 & \text{für } t = 0 \\ k_{t-1} (-\operatorname{sgn}(i_{t-3} - i_{t-4})) & \text{für } t > 0 \end{cases} \\
 i_t &= 0,5 b_{t-1} + 0,5 i_{t-1}
 \end{aligned}$$

Abbildung 40 zeigt die Struktur zur Regelung des Hals-Roll-Gelenks im reaktiven System als grafisches Netz.

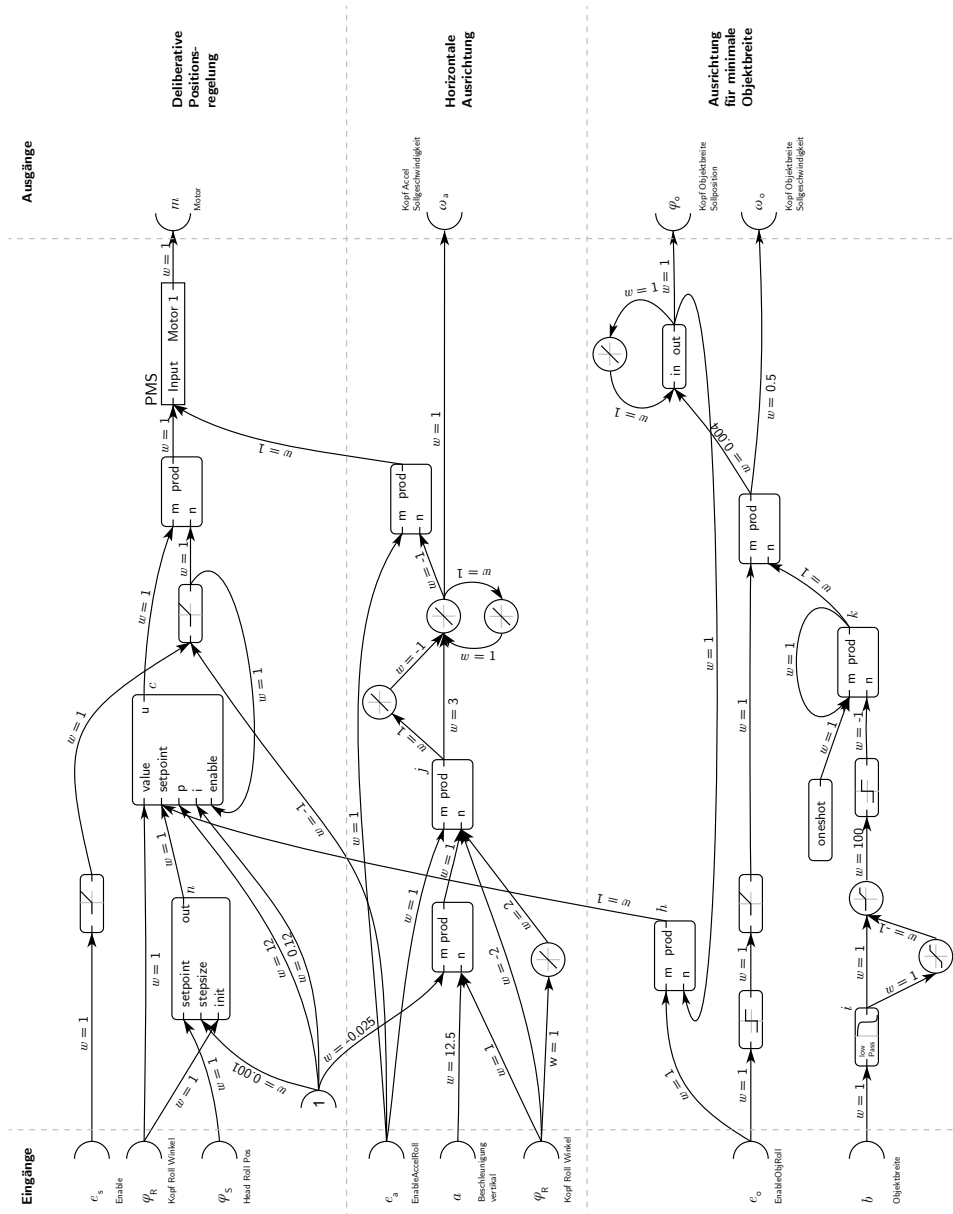


Abbildung 40: Reaktives Netz zur Regelung der Kopfneigung (*Roll*). Dieses kann aufgrund einer Winkelvorgabe aus dem deliberativen System, reaktiv zur horizontalen Ausrichtung mittels Beschleunigungssensorik oder hybrid anhand der visuellen Information der Breite des Unterarms erfolgen, sodass der Kopf mit der vertikalen Bildachse parallel zum Unterarm geneigt wird. Zur besseren Übersichtlichkeit ist der Eingang ϕ_R doppelt dargestellt.

Für den Fall, dass die *Roll*-Bewegung des Kopfes gezielt angesteuert werden soll (Variante 1), was zum Beispiel für eine natürliche Interaktion mit dem Menschen notwendig sein kann, wird die Bilddrehung alternativ über eine affine Rotation der *PixelPipeline* kompensiert. Da dies jedoch das Blickfeld zusätzlich einschränkt (siehe Abbildung 41), wird die Ausrichtung über das Halsgelenk bevorzugt.

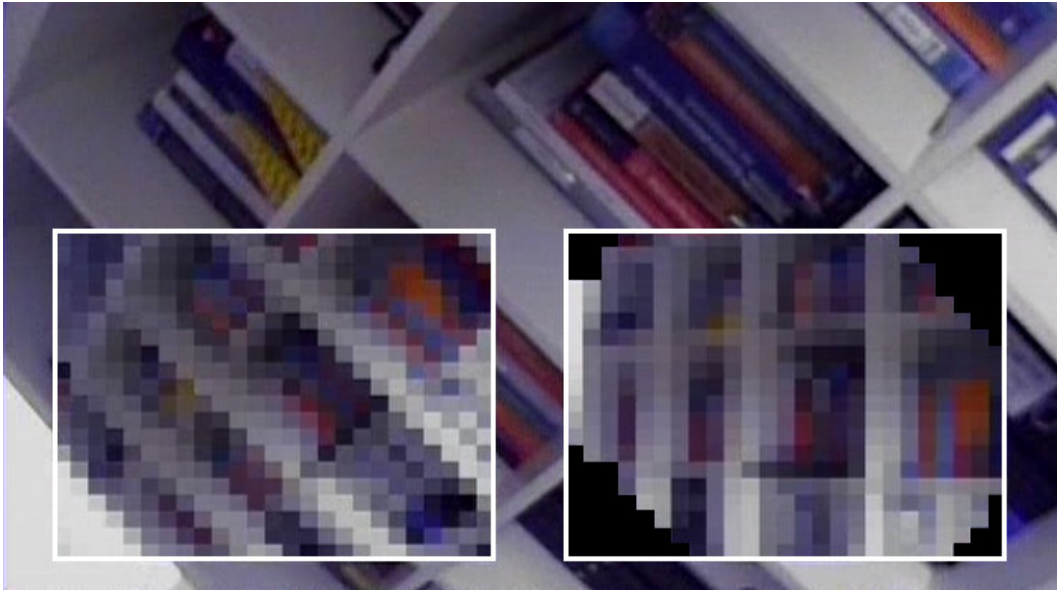


Abbildung 41: Kompensation der Kopfneigung um das *Roll*-Gelenk durch eine entgegengesetzte Rotation in der *PixelPipeline*. (Hintergrund) generisches Kamerabild; (unten links) Bild über *PixelPipeline* ohne Rotation; (unten rechts) Bild über *PixelPipeline* mit Kompensation der Kopfneigung mittels der Beschleunigungssensorik im Kopf.

5.2 Vorwärtskinematik als Referenz

Die Bewegungssteuerung in dieser Arbeit basiert ausschließlich auf Daten im Konfigurationsraum. Zur Überprüfung der Daten und Ergebnisse wurden diese in den Aktuationsraum überführt. Die notwendige Vorwärtskinematik dient der Visualisierung der Handposition im Raum und zur Ermittlung des ungefähren Handabstandes zum Auge und wurde nach den Denavit-Hartenberg-Konventionen aufgestellt.

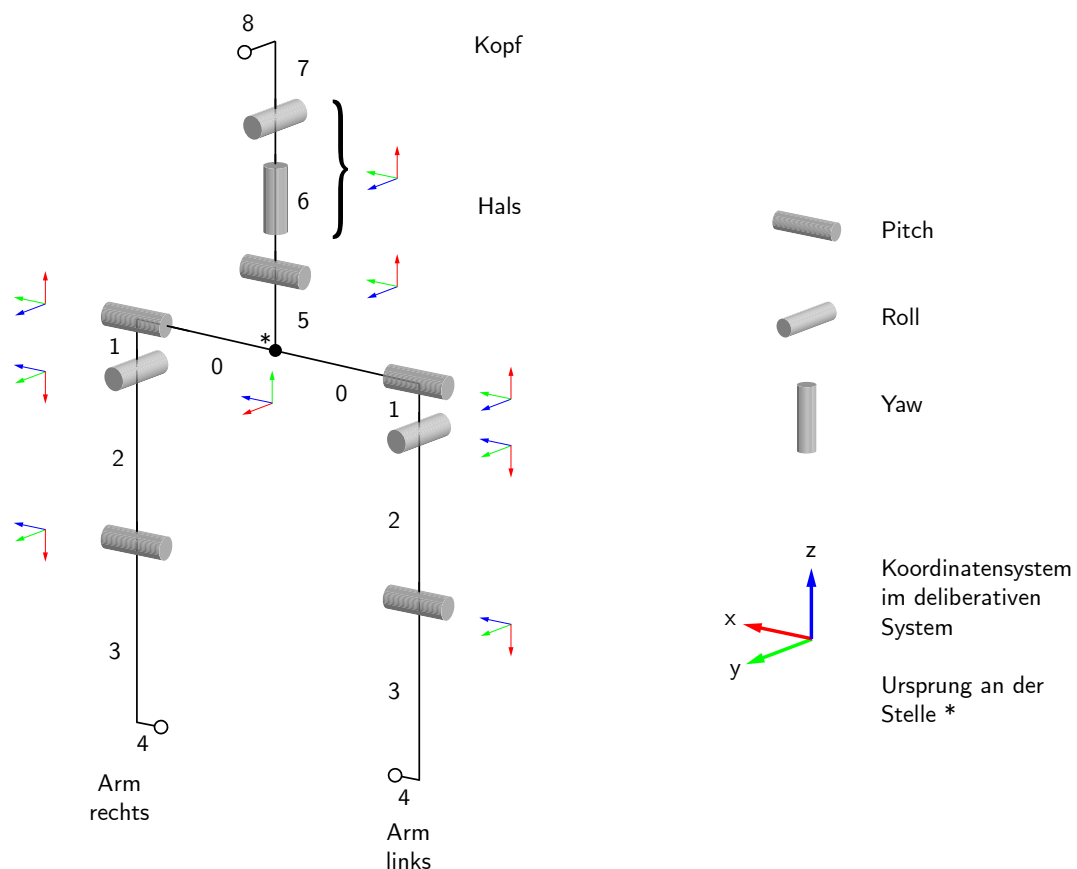


Abbildung 42: Übersicht der Koordinatensysteme für die Vorwärtskinematik der Arme und des Kopfes mittels Denavit-Hartenberg-Konvention. Die Achsen der Koordinatensysteme sind farb-codiert. Die neun Verbindungen (*Linkages*) sind mit 0 bis 8 durchnummeriert, wobei die beiden Arme spiegelsymmetrisch um die mittlere Sagittalebene modelliert sind. Die Elemente 4 und 7 haben kein separates Gelenk und sind statisch als Verschiebung vereinfacht. Das *Roll*- und das *Yaw*-Gelenk im Hals können über die DH-Konventionen als ein Gelenk (6) modelliert werden.

Da dazu jedoch absolute Informationen, wie Körperteillängen, Winkel und Gelenkübersetzungen notwendig sind, entspricht dieses Vorgehen nicht dem Anspruch dieser Arbeit, Prinzipien zu entwickeln, die grundsätzlich ohne explizite Modelle auskommen. Im Laufe der Arbeit werden Körperteile gegen andere ausgetauscht,

die nicht exakt die gleichen sensorischen Absolutwerte liefern; zudem ist die Mechanik des Myon recht nachgiebig. Deshalb wird die Vorwärtskinematik ausschließlich zur Plausibilitätsprüfung berechnet. Die bewusst tolerierte Ungenauigkeit ist bei der Auswertung der Grafiken zu berücksichtigen. Abbildung 42 zeigt eine Übersicht der Koordinatensysteme für die Vorwärtskinematik.³⁴

Grundsätzlich existieren bereits aktive Lernverfahren, die über die visuelle Rückkopplung von Markerpositionen an der Roboterhand die DH-Parameter der Armkinematik schätzen können, wie Cunha et. al. am Beispiel des Roboters *iCub* in einer Simulation zeigen [CVB⁺21].

5.3 Zusammenfassung

Die Regelung der Gelenke erfolgt im reaktiven System des Roboters. Dazu werden die reaktiven Netze auf die AB3Ds über den Körper verteilt und lokal berechnet. Die wesentlichen Parameter stehen dabei jedem AB3D sowie der deliberativen Einheit (dem *BrainModule*) über den *SpinalCord* zur Verfügung. Die deliberative Einheit ist ohne Rechenaufwand in der Lage, zentrale Parameter zu steuern und das Verhalten zu beeinflussen. Reaktive und deliberative Einheit fusionieren zu einer hybriden Architektur.

Die Reglerstrukturen der Handgelenkdrehung, des Greifens sowie der Kopfneigung um das *Roll*-Gelenk ermöglichen ein lokales sensomotorisches Verhalten. Die Regelung der verbleibenden Gelenke, die für diese Arbeit von Relevanz sind (Kopf-*Pitch*, Kopf-*Yaw*, Schulter-*Pitch*, Schulter-*Roll*, Ellenbogen), erfolgen zunächst über klassische PI-Regler im reaktiven Netz mit einer Führungsgröße aus der deliberativen Einheit. In Kapitel 7.2 wird auch für das Ellenbogengelenk eine erweiterte sensomotorische Schleife im reaktiven Netz eingeführt.

Um die späteren Datenpunkte im Konfigurationsraum besser evaluieren zu können, wird eine Vorwärtskinematik nach der Denavit-Hartenberg-Konvention eingesetzt, um die Punkte in den Aktuationsraum zu überführen. Die Vorwärtskinematik hat für das Verhalten des Roboters keine Relevanz.

³⁴Die der Vorwärtskinematik zugrundeliegenden Abmessungen und Winkelwerte des Roboters Myon stehen im Anhang auf Seite 170 f. zur Verfügung.

6 Visuomotorik mit Hilfe von sensomotorischen Schleifen

Die Visuomotorik führt die bislang vorgestellten und entwickelten Teilsysteme zusammen, da sie alle Komponenten, von der Hardwareplattform mit der Entwicklungsumgebung über die Wahrnehmung und die Körperbeherrschung bis hin zur Manipulation der Umwelt umfasst.

Begleitet wird dieses Kapitel von den vier ausgewählten Aufgabenstellungen des ET 6-6-R [PM13]: ein Objekt visuell verfolgen, ein Objekt greifen, Objekte stapeln und Linien nachzeichnen. Die drei zuletzt genannten Aufgaben gehören zur Kategorie der Hand-Auge-Koordination, einem Teilgebiet der Visuomotorik.

Als zentralen Tenor dieser Arbeit sollen diese Aufgaben ohne explizite Modelle des Roboters oder der Umwelt ausgeführt werden. Ziel ist es, die Aufgaben mit minimalem Rechenaufwand und maximaler Zuverlässigkeit und Flexibilität erfolgreich zu absolvieren. Genauigkeit und Schnelligkeit stehen dabei im Hintergrund. Der Erfolg beim Greifen eines Objekts ist dann gegeben, wenn das gegriffene Objekt nicht von selbst aus der Hand fällt; das Stapeln von Objekten gilt dann als erfolgreich, wenn das obere Objekt dauerhaft auf dem unteren Objekt stehen bleibt; beim Nachzeichnen erfolgt die Erfolgsbewertung mithilfe einer Vergleichsschablone aus dem ET 6-6-R [PM13]. Für die genannten visuomotorischen Anforderungen werden die Beine des Roboters nicht benötigt; dank seiner Modularität können diese einfach demontiert werden. Der Torso mit Armen, Händen und Kopf wurde fest auf einem Tisch montiert. Beine würden die Experimente unnötig komplex machen, da zusätzlich der aufrechte Sitz oder das aufrechte Stehen sichergestellt werden müssten. Auf Grund der dezentralen Rechenarchitektur erfordert das spätere Zusammenführen keine aktive Migration. Abbildung 43 zeigt den Versuchsaufbau sowie die benötigten Utensilien. Zum Greifen und Stapeln stehen drei Zylinderkörper mit 6 cm Durchmesser und 6 cm Höhe in grün, rot und gelb zur Verfügung. Das Nachzeichnen erfolgt anhand des Aufgabenblatts zum Untertest *Nachzeichnen* aus dem ET 6-6-R [PM13]. Das gelbe Quadrat mit 5 cm Kantenlänge wird eingesetzt, um dem Roboter sowohl die Zeichenvorlage als auch das Zielfeld zu demonstrieren. Da für alle Versuche die gleiche Hand genutzt wird, musste der Stift für einen sicheren Griff modifiziert werden. Er hat an der dicksten Stelle einen Durchmesser von 6 cm und an der Stiftspitze einen orangen Ring für die visuelle Erkennung.

Die in Kapitel 4.2 besprochene visuelle Wahrnehmung ist der wesentliche Wahrnehmungskanal für die folgenden visuomotorischen Handlungen nach dem Prinzip des *Visual Servoing*. Da diese Aufgabe ohne a priori definierte Modelle wie beispielswei-

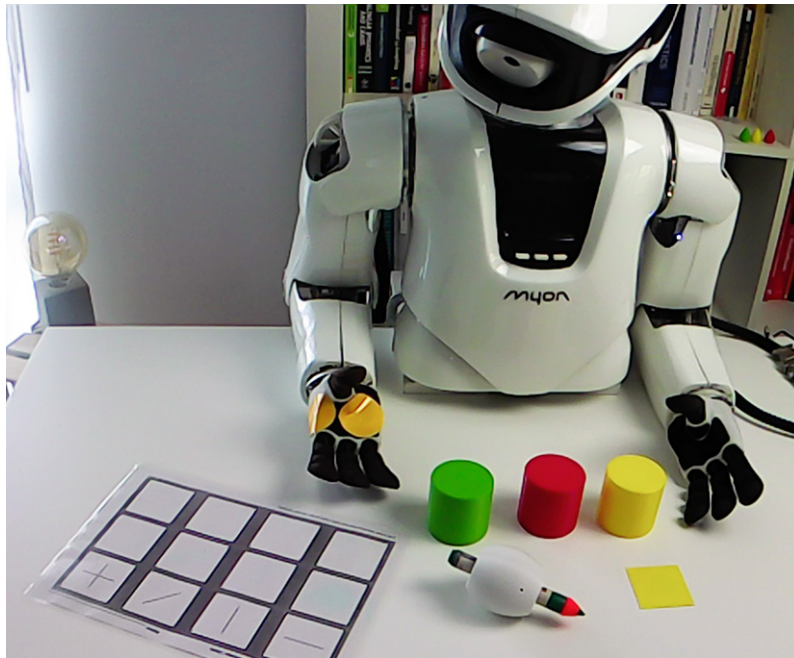


Abbildung 43: Versuchsaufbau zum Absolvieren visuomotorischer Aufgaben durch einen humanoiden Roboter. Zubehör: grüner, roter und gelber Zylinder, Stift, gelbe quadratische Karte, Testbogen zum Nachzeichnen.

se einer *Robot-Jacobian* ausgeführt werden soll, wird nach dem bildbasierten Ansatz (IBVS) gearbeitet. Die Kamera im Roboterkopf betrachtet die Hand sowie das Objekt unabhängig und ist damit außerhalb der Regelschleife (*Open-loop/Eye-to-hand-Konfiguration*). Die drei Freiheitsgrade im Hals erlauben eine beliebige Ausrichtung der Kamera. Um diese Hardwarekonfiguration mit den klassischen Verfahren des *Visual Servoing* zu betreiben, wäre für jede Kopfausrichtung die Berechnung einer spezifischen Bild-Roboter-*Jacobian* erforderlich. Ohne exakte Modelle des optischen Systems inklusive einer Abstandsschätzung ist die direkte Berechnung nicht möglich und die *Jacobians* müssten für jede Kopfausrichtung neu gelernt werden. Die Visuomotorik in dieser Arbeit baut auf weniger rechenintensive Prinzipien aus robusten sensomotorischen Schleifen als Kopplung zwischen dem Kamerabild und den Gelenken auf.

Bei der Wahl der Aufgaben wurden morphologisch bedingte Einschränkungen des Roboters berücksichtigt. Bei einer Tischoberfläche auf Hüfthöhe des Roboters (Unterkante des Torso) ist der per Roboterhand erreichbare Bereich, wie in Abbildung 44 gezeigt, eingeschränkt. Der grau markierte Bereich (im Folgenden als Arbeitsbereich bezeichnet) zeigt die Fläche, die die Hand erreichen kann, ohne von der Tischoberfläche abzuheben. Die Übergabe eines Objekts von einer zur anderen Hand

ist grundsätzlich möglich, jedoch nur in einem kleinen Bereich der Tischoberfläche. Dies liegt darin begründet, dass Myon je Arm vier Freiheitsgrade besitzt: zwei in der Schulter, einen im Ellenbogen und einen Freiheitsgrad im Handgelenk (vergleiche Abbildung 8 auf Seite 24). Im Vergleich dazu hat der Mensch je Arm sieben Freiheitsgrade: drei in der Schulter, einen im Ellenbogen, einen im Unterarm und zwei im Handgelenk, dazu zwei unterstützende Freiheitsgrade im Schultergürtel. Die reduzierte Anzahl der Freiheitsgrade führt zu einer deutlichen Einschränkung in der Bewegungsfreiheit des Myon.

Auch wenn der Begriff *Einschränkung* zunächst negativ konnotiert ist, möchte diese Arbeit die damit verbundene Vereinfachung des Konfigurationsraums als für die Robotik als durchaus erstrebenswert herausheben. Burhan Hafez und Wermter [BHW21] beispielsweise haben zur Vereinfachung gezielt die Armbewegung des Roboters NICO (Neuro-Inspired COmpanion) [KSM⁺17] auf die vier Freiheitsgrade, die ebenfalls beim Roboter Myon zur Verfügung stehen, begrenzt.

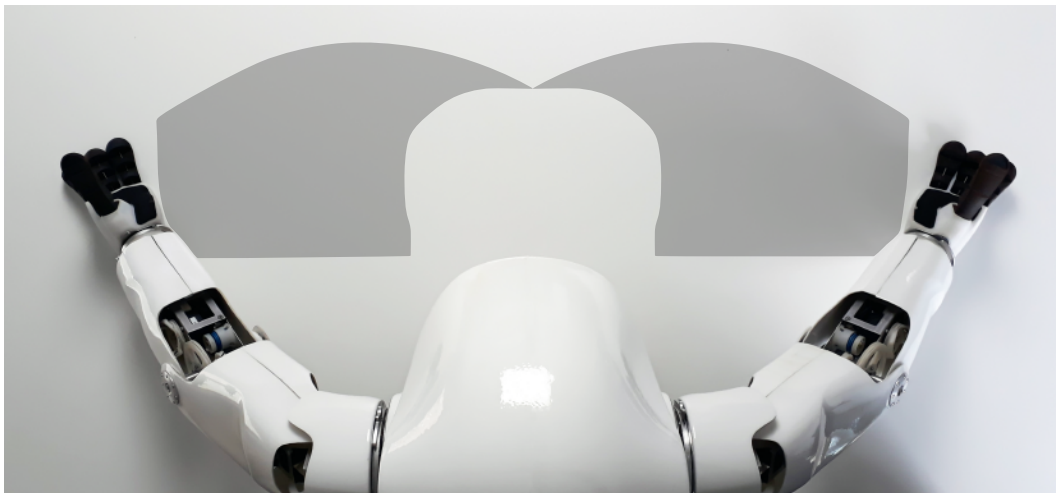


Abbildung 44: Arbeitsbereich der Hände in der Tisebene. Die graue Fläche kennzeichnet den Bereich, den die Hand erreichen kann, ohne mit dem Torso zu kollidieren oder mit dem Handballen von der Tischoberfläche abzuheben.

Alle Bewegungen, sowohl die der Hand als auch die des Kopfes, orientieren sich an dem Gesetz von Fitts [Fit54], welches die Dauer von menschlichen Zeigebewegungen unter Berücksichtigung des Abstandes und der Größe des Ziels vorhersagt. Demzufolge beginnt die Bewegung schnell und erfolgt grob in die Richtung des Ziels. Sie wird mit abnehmendem Abstand langsamer und genauer. Dieses Verhalten ist laut Fitts unabhängig davon, ob ein (reales) Zeigen mit dem Finger oder ein (digitales) Zeigen mit dem Zeiger einer Computermaus beschrieben wird.

6.1 Objekt visuell verfolgen

Innerhalb des Entwicklungstests ET 6-6-R besteht der Test T076 für 6–7,5 Monate alte Säuglinge (Grenzstein G-06-KOG) im Verfolgen eines Objekts mit den Augen [PM13]. Dabei ist das Objekt im Abstand von 20 cm zu den Augen horizontal und vertikal um jeweils 15 cm in beide Richtungen zu bewegen. Anschließend erfolgt eine kreisförmige Bewegung mit einem Radius von 10 cm.

Um dieser Anforderung gerecht zu werden, die auch eine Grundfertigkeit für die folgenden Aufgaben darstellt, wurden zwei einfache und unabhängig voneinander agierende sensomotorische Schleifen implementiert. Als sensorischer Kanal steht die horizontale und vertikale Bildposition des zu verfolgenden Objekts zur Verfügung. Diese werden aus den Ortshistogrammen des *ColorClassifiers* (siehe Abbildung 31 auf Seite 60) generiert. Ziel ist es, den Kopf so zu bewegen, dass das Objekt, genauer der *Point Of Interest/POI*, in der Bildmitte gehalten wird. Alle drei Gelenke zur Kopfbewegung (*Pitch*, *Roll*, *Yaw*) haben Einfluss auf die Objektposition im Bild. Für eine möglichst robuste sensomotorische Schleife wird die horizontale Objektposition ausschließlich an das Gieren (Kopfdrehung um die Hochachse; *Yaw*) gekoppelt, und die vertikale unabhängig davon auf das Nicken (*Pitch*). Die Kopfneigung (*Roll*) wird mittels der Beschleunigungssensorik über die sensomotorische Schleife im reaktiven Netz horizontal gehalten. Abbildung 45 zeigt den Verlauf der Halswinkel *Pitch* und *Yaw* des Roboters bei der Durchführung des Tests, welcher problemlos absolviert wurde und wesentlich von der Störsicherheit der Objekterkennung abhängt.

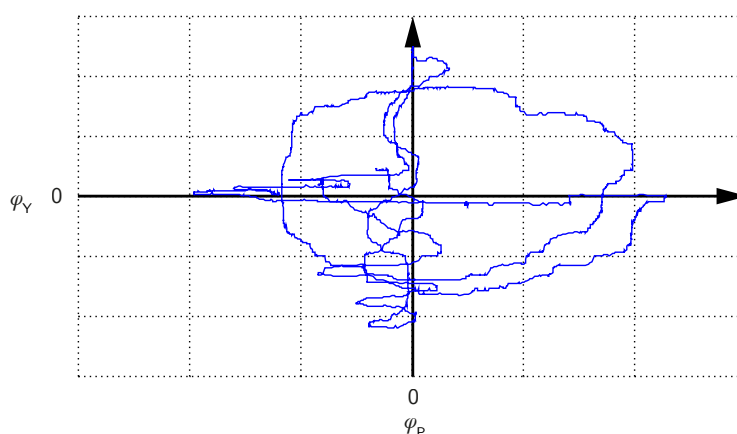


Abbildung 45: Visuelles Verfolgen eines Objekts gemäß ET 6-6-R (Test T076) [PM13]. Der Kurvenverlauf zeigt die beiden Kopfwinkel der *Pitch*- und *Yaw*-Bewegung des Roboters während des Tests.

6.2 Objekt greifen

Innerhalb des Entwicklungstests ET 6-6-R besteht der Test T040 für 6–7,5 Monate alte Säuglinge (Grenzstein G-06-HM) im Greifen eines Gegenstandes von der Kleinfingerseite her (ulnar) [PM13].

Während für das visuelle Verfolgen von Objekten nur deren Position im Bild relevant ist, beginnt ab hier ein Prozess, der unter begrenztem Wissen (*Bounded Rationality*) ausgeführt werden muss. Wie aus den vorherigen Kapiteln bekannt, basiert die Informationslage primär aus dem Kamerabild. Dabei gibt es für das zu greifende Objekt zwei Bildkoordinaten und eine zweidimensionale Objektgröße in Pixeln. Unbekannt ist die Morphologie der Objektunterlage (hier des Tisches), die absolute Objektgröße, der Objektabstand, das Objektgewicht, die absolute Handposition und so weiter. Über Distanzsensoren oder eine Tiefenkamera verfügt das System nicht. Um dennoch ein zuverlässiges Greifen zu ermöglichen, werden Heuristiken eingesetzt, die gewisse Annahmen voraussetzen. Grundsätzlich ist anzunehmen, dass das Objekt auf einer Oberfläche abgestellt ist, welche für den Greifprozess in irgendeiner Form als Randbedingung (*Environmental Constraint*) genutzt werden kann. Dafür wird das Sichtfeld mittig auf das zu greifende Objekt gerichtet. Je nachdem, ob sich das Objekt links oder rechts vom Torso befindet, wird der Ellenbogen des entsprechenden Armes relaxiert, sodass die Hand auf der Tischoberfläche aufliegt.

Für die folgenden Ausführungen wird ausschließlich der rechte Arm eingesetzt. Für den linken Arm gilt entsprechend alles gespiegelt an der mittleren Sagittalebene des Roboters. Zur Erkennung der Hand wurden zwei Varianten implementiert:

Variante 1: Durch willkürliche Bewegungen der Schultergelenke sind Bewegungen in der rechten Bildhälfte zu erkennen, und die Position der Hand kann durch das Ende der vom unteren Rand beginnenden Bewegung geschätzt werden. Die Erkennung der Hand erfolgt mit stark reduzierter Auflösung von horizontal 16 Pixeln und vertikal 32 Pixeln, die durch einen separaten Aufruf der *PP* in die rechte Bildhälfte gelegt werden. Die grobe Auflösung führt zwar zu groben Bewegungen, hat jedoch zum einen den Vorteil, dass die Erkennung schneller zu berechnen ist, zum anderen führen die groben Bewegungen zu einer Exploration der Tischoberfläche, die für das im folgenden Kapitel 7 gezeigte Lernen eines impliziten Modells der Oberfläche genutzt werden kann. Die andere Bildhälfte wird genutzt, um die Armbewegung von der Kopfbewegung oder der Umwelt zu unterscheiden. Abbildung 46 zeigt die auf dieser Variante basierenden Bilder.

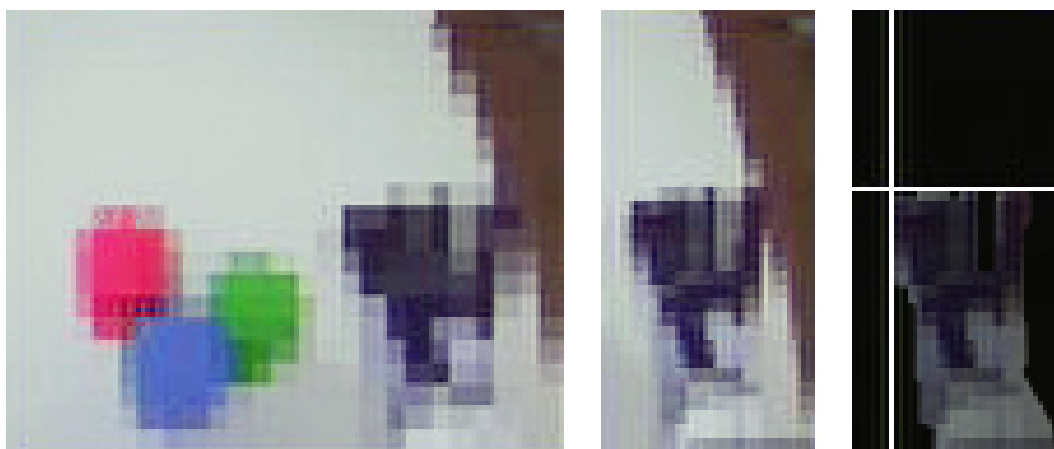


Abbildung 46: Visuelles Ermitteln der Handposition durch Differenzbildung der rechten Bildhälfte. (links) Vollbild in 25 x 20 Pixel Auflösung, (mitte) Rechte Bildhälfte in 16 x 32 Pixel Auflösung, (rechts) Zeitliches Differenzbild in 16 x 32 Pixel Auflösung der rechten Bildhälfte mit eingezeichneter Position der Hand. Die Handposition wird durch den Pixel ganz links in der obersten Zeile einer Veränderung definiert.

Variante 2: Die Erkennung der Hand erfolgt über eine für diese charakteristische Farbe (in diesem Fall gelb). Die Position wird, wie auch bei der Objektverfolgung, über den ColorClassifier mittels horizontaler und vertikaler Ortshistogramme ermittelt (siehe Abbildung 31 auf Seite 60).

Variante 1 zeichnet aus, dass sie ohne jede Modifikation des Armes auskommt und – abgesehen von Umgebungen mit schwarzem Hintergrund – zuverlässig funktioniert. Eine 5 Hz-Aktualisierung (also alle 200 ms) führt bei schnellen Bewegungen zu deutlichem Überschwingen der Bewegung. Hier ist ein gangbarer Mittelweg in der Aktualisierungsrate zu finden: zum einen schnell genug, um Bewegungen gut erfassen zu können; zum anderen langsam genug, um Veränderungen in den Differenzbildern erkennen zu können, die sonst vom Bildrauschen überdeckt werden. Die Geschwindigkeit der Armbewegungen entspricht im Rahmen dieser Arbeit in etwa der des Menschen, wenn er ganz ruhig und entspannt ein Objekt greift.

Variante 2 erfordert die farbliche Markierung der Hand, sodass diese dort zuverlässig und reproduzierbar vom Hintergrund separiert werden kann. Diese Variante ist unabhängig von der Aktualisierungsrate immer gleich zuverlässig und hat zudem die etwa 200-fache Ortsauflösung im Vergleich zu Variante 1. Die Positionsbestimmung erfolgt hier auch ohne Handbewegung.

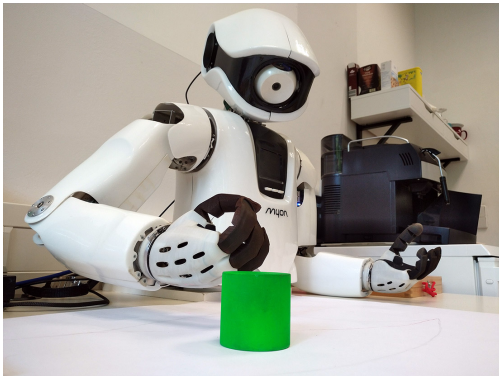
Da der Rechenaufwand für beide Varianten ähnlich ist, jedoch die höhere Genauigkeit von Variante 2 die resultierenden Bewegungen deutlich flüssiger macht, wird diese Variante im Folgenden eingesetzt.

Für die Kopfneigung (*Roll*) sind alle drei in Kapitel 5.1 gezeigten Optionen geeignet, um die Aufgabe erfolgreich abzuschließen. Am schnellsten konvergieren Hand- und Objektposition, wenn das *Roll*-Gelenk so ausgerichtet wird, dass die Bewegung in Schulter-*Pitch* einer vertikalen Bewegung im Bild sowie die Bewegung in Schulter-*Roll* einer horizontalen Bewegung im Bild mit minimalen Quereinflüssen entsprechen. Zur Erkennung des Unterarms wurde sowohl die schwarze Trennfuge der Unterarmschalen als auch ein parallel zur Trennfuge aufgeklebter Farbstreifen erfolgreich eingesetzt. Abbildung 49a auf Seite 89 zeigt die Erkennung des Armwinkels im Bild durch die horizontale Breite der Trennfuge. Dafür wurden in der *PP* alle Bildzeilen auf eine Zeile abgebildet (gestaucht). Je senkrechter die Fuge im Bild liegt, desto schmaler wird die gestauchte Abbildung. Nachteilig ist, dass Myon eine unnatürlich wirkende Kopfhaltung einnimmt, die bei einer Mensch-Roboter-Interaktion eventuell nicht erwünscht ist. Dies ist ebenfalls in Abbildung 49a ersichtlich.

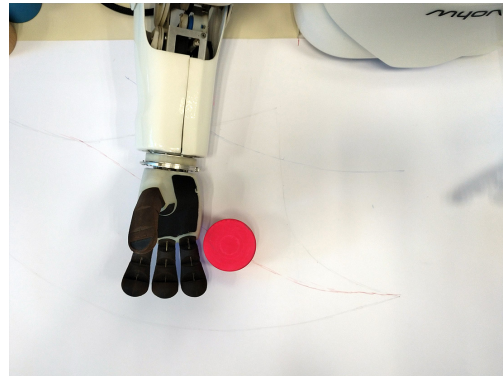
Für den Greifprozess mit sensomotorischen Schleifen unter Berücksichtigung der morphologischen Gegebenheiten des Myon hat sich folgende Heuristik bewährt:

- Objekt durch Kopfbewegung im Kamerabild zentrieren und die Hand ins Blickfeld bewegen (Abbildung 47a)
- Hand (mit der Handfläche nach oben) von außen neben das Objekt bewegen (Abbildung 47b)
- Handfläche zum Objekt drehen (Abbildung 47c)
- Zugreifen (Abbildung 47d)

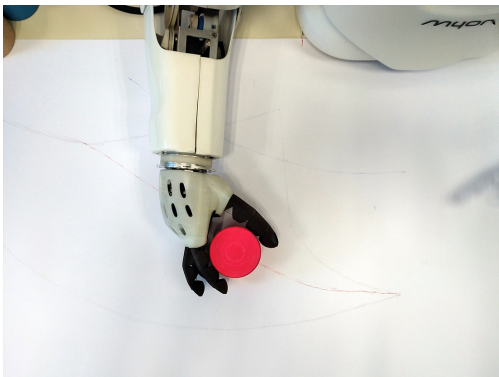
Beim Greifprozess müssen Objekt und Hand zeitgleich in die Bildmitte bewegt werden. Dazu werden die zu erkennenden Farbklassen der *PP* alle 200 ms (also nach 10 Kamerabildern) umgeschaltet. Daraus ergibt sich eine Aktualisierungsrate von 2,5 Hz für Objekt- und Handposition.



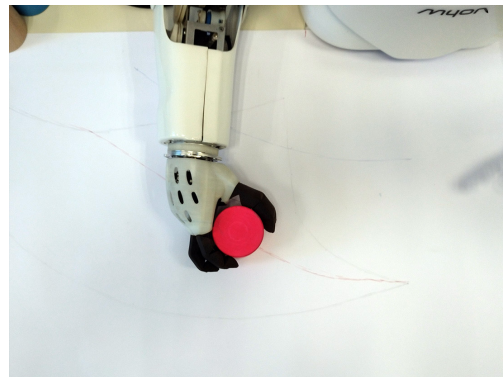
(a) Objekt im Kamerabild zentrieren und Hand im Blickfeld



(b) Hand neben dem Objekt mit der Handfläche nach oben



(c) Hand neben dem Objekt mit der Handfläche zum Objekt eingedreht



(d) Objekt erfolgreich gegriffen

Abbildung 47: Ablauf der Aufgabe „Objekt greifen“ mit dem rechten Arm des humanoiden Roboters.

Dafür werden sieben unabhängige sensomotorische Schleifen eingesetzt, die in Abbildung 48 dargestellt sind. Zusätzlich wird das Ellenbogengelenk relaxiert. Die mit * gekennzeichneten Eingänge kommen vom deliberativen System. Der Ursprung der Bildkoordinaten ist in die Bildmitte verschoben.

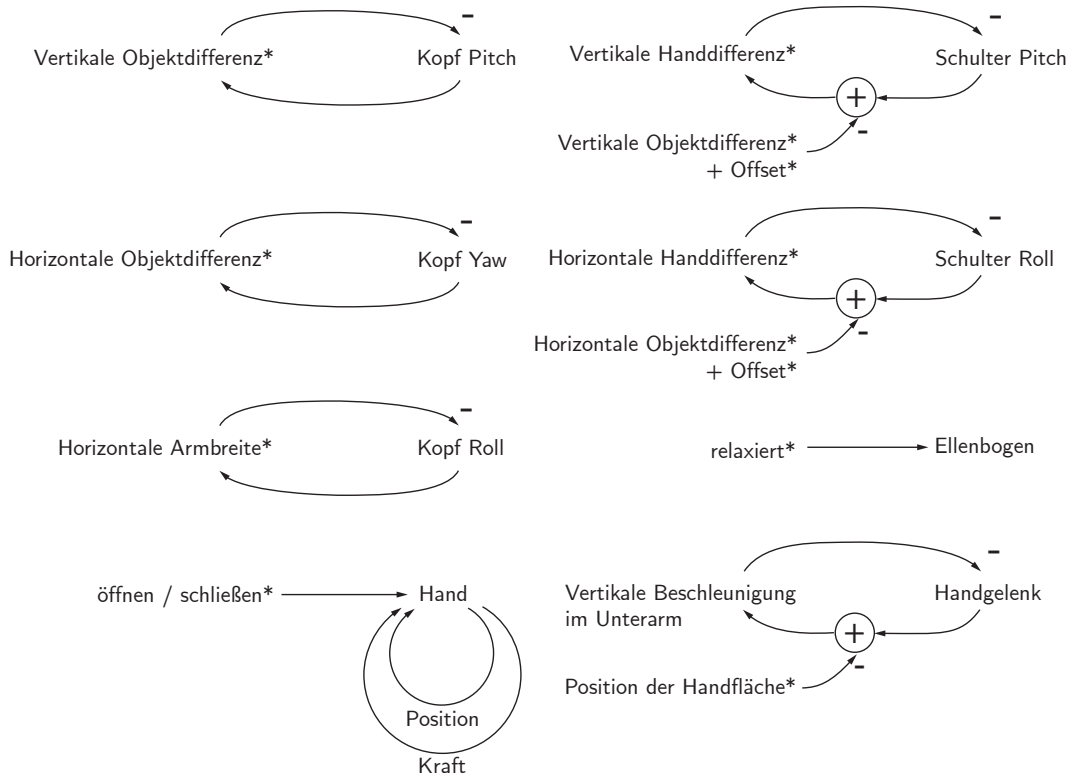


Abbildung 48: Überblick über die sensomotorischen Schleifen für „Objekt greifen“. Die Schleifen haben auf der linken Seite den sensorischen Teil (Eingang) und auf der rechten Seite den motorischen Teil (Ausgang). Die als Differenz bezeichneten Werte sind die Differenzen der aktuellen Position zur Bildmitte. Bei exakt mittiger Position sind die Differenzen null. Die mit * gekennzeichneten Eingänge sind Werte, die vom deliberativen System bereitgestellt werden.

6.3 Objekte stapeln

Innerhalb des Entwicklungstests ET 6-6-R besteht der Test T085 für 9–21 Monate alte Kleinkinder (Grenzstein G-18-KOG) im Stapeln zweier Objekte.

Das Stapeln von Objekten basiert auf den Heuristiken für „Objekt greifen“ wie in Kapitel 6.2 gezeigt. Die beiden Objekte werden im Folgenden als Basisobjekt (unten) und Stapelobjekt (oben) bezeichnet. Grundsätzlich gibt es auch hier zwei Varianten. Die erste Variante basiert auf der Heuristik, die Objekte im Bild zentriert; die zweite basiert auf Punkten im Konfigurationsraum des Arms. Damit das Basisobjekt, das Stapelobjekt und die Hand zeitgleich im Bild erkannt werden können, wird der Ursprung der Bildkoordinaten als virtuelle Bildmitte leicht ins obere linke Viertel verschoben – siehe Abbildung 49 auf Seite 89.

Im Folgenden wird die Schrittfolge für beide Varianten in Stichpunkten beschrieben:

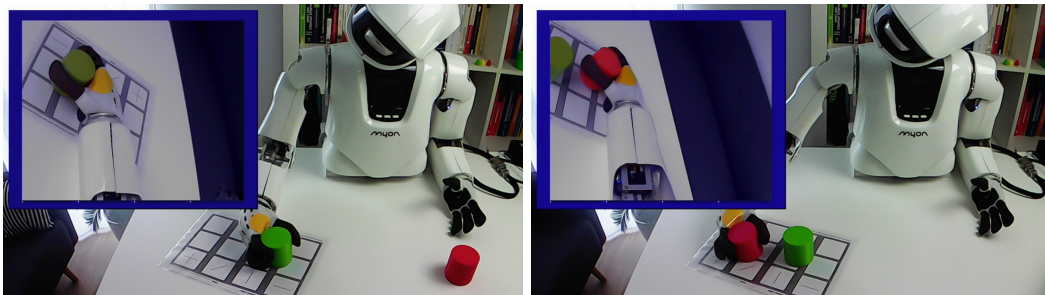
Variante 1:

- Stapelobjekt durch Kopfbewegung im Kamerabild zentrieren und greifen (Abbildung 49b)
- Stapelobjekt anheben durch Beugen des Ellenbogens (Abbildung 49c)
- Basisobjekt im Bild zentrieren und Kopfposition fixieren
- Stapelobjekt auf Basisobjekt stellen indem der Ellenbogen langsam gestreckt wird und das Stapelobjekt mittels sensomotorischer Schleife kontinuierlich in der Bildmitte gehalten wird. Das Basisobjekt wird visuell vom Stapelobjekt überlagert. (Abbildung 49d)
- Stapelobjekt loslassen und Hand ablegen (Abbildungen 49e,f)

Variante 2:

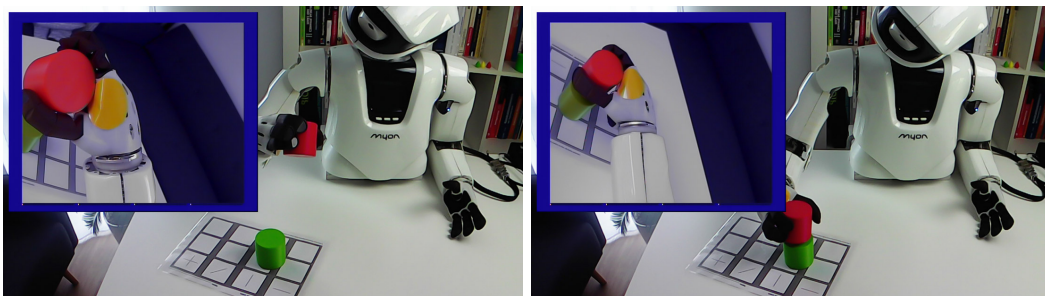
- Zentrieren und Greifen des Basisobjekts, um dessen Koordinaten im Konfigurationsraum des Arms zu erlernen. (Abbildung 49a)
- Zentrieren und Greifen des Stapelobjekts (Abbildung 49b)
- Stapelobjekt anheben durch Beugen des Ellenbogens und Anfahren der Schulter-*Pitch*- und Schulter-*Roll*-Position des Basisobjekts im Konfigurationsraum. (Abbildung 49c)

- Stapelobjekt auf Basisobjekt stellen indem der Ellenbogen langsam gestreckt wird (Abbildung 49d)
- Stapelobjekt loslassen und Hand ablegen (Abbildungen 49e,f)



(a) Greifen des Basisobjekts

(b) Greifen des Stapelobjekts



(c) Stapelobjekt anheben

(d) Stapelobjekt abstellen

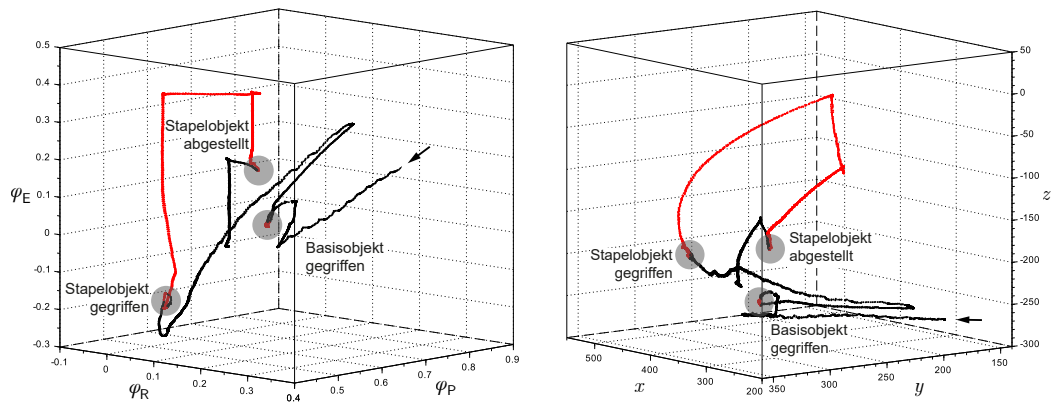


(e) Stapelobjekt loslassen

(f) Aufgabe abgeschlossen

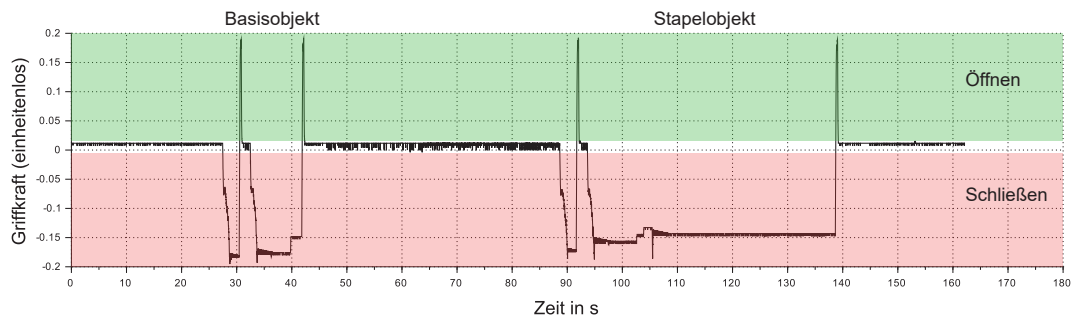
Abbildung 49: Ablauf der Aufgabe „Objekte stapeln“ mit dem rechten Arm des humanoiden Roboters. Oben links ist jeweils das korrespondierende Kamerabild des Roboters zu sehen. Das Raster auf dem Tisch ist für diese Aufgabe irrelevant und wird für das spätere Nachzeichnen von Linien benötigt.

Die Variante 1 beschreibt den Prozess einer typischen Blickheuristik, wie sie in Kapitel 2.1 vorgestellt wurde. Das Stapelobjekt wird so bewegt, dass der Blickwinkel konstant bleibt bis es zu der gewünschten Kollision mit dem Basisobjekt kommt. Die sensomotorischen Schleifen werden beim Übergang der Teilaufgaben lediglich gestoppt beziehungsweise werden die Geschwindigkeiten, Offsets und Farbklassen variiert.



(a) Handposition im Konfigurationsraum des rechten Arms. Der Pfeil zeigt den Startpunkt der Bewegung. Ist der Verlauf rot dargestellt, hat die Hand das Objekt gegriffen. Die Gelenkwinkel sind einheitenlos.

(b) Handposition im Aktuationsraum des Robotertorso. Der Pfeil zeigt den Startpunkt der Bewegung. Ist der Verlauf rot dargestellt, hat die Hand das Objekt gegriffen. Die Werte sind in Millimeter angegeben.



(c) Verlauf der Griffkraft über die Zeit. Liegt die Kurve im grünen Bereich wird die Hand geöffnet. Im roten Bereich schließt die Hand oder übt eine Kraft aus um das Objekt zu halten. Dazwischen, also um die Null herum, ist die Hand geöffnet. Im roten Bereich ist gut sichtbar, dass die Hand immer zweimal hintereinander geschlossen wird, um einen sicheren Griff zu gewährleisten.

Abbildung 50: Position der Hand im Konfigurationsraum sowie im Aktuationsraum des Torso und Verlauf der Griffkraft beim Objekte stapeln. Die Abfolge dieses Experiments wurde händisch weiter geschaltet. Dies führt neben den ruhigen Armbewegungen zu einer zusätzlichen Verzögerung des Ablaufes. Die Handlung ist bei etwa 140s beendet.

Eine weitere Herausforderung ist das Abstellen des Stapelobjekts auf dem Basisobjekt, da die Höhe des Basisobjekts unbekannt ist. Das Aufsetzen des Stapelobjekts auf dem Basisobjekt wird, wie in Kapitel 4.1 beschrieben, über den Integratorwert des Ellenbogen-PI-Reglers erkannt.

Abbildung 50 zeigt (a) die Position der Hand im Konfigurationsraum, (b) die Position der Hand im Aktuationsraum sowie (c) den Verlauf der Griffkraft während ein Objekt nach Variante 2 gestapelt wird. Immer wenn die Hand ein Objekt gegriffen hat, ist der Verlauf in den Abbildungen (a) und (b) rot dargestellt. Die Darstellung im Aktuationsraum erfolgt über die Online-Berechnung der Vorwärtskinematik (siehe Kapitel 5.2). Sie hat ausschließlich informativen Charakter und wird für den Stapelprozess nicht herangezogen. Das Kontrahieren des Ellenbogens wird über eine separate sensomotorische Schleife durchgeführt (siehe Abbildung 51).

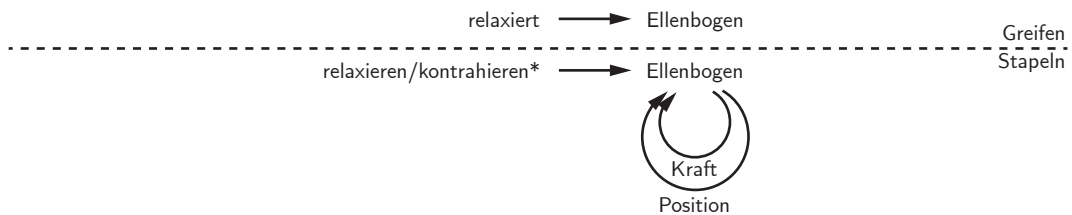


Abbildung 51: Überblick über die sensomotorischen Schleifen im Ellenbogen für „Objekt greifen“ und „Objekt stapeln“. Die Schleifen haben auf der linken Seite den sensorischen Teil (Eingang) und auf rechten Seite den motorischen Teil (Ausgang). Der mit * gekennzeichnete Eingang wird vom deliberativen System bereitgestellt. Details der Regelschleife siehe Kapitel 5.1.

Da bei Variante 2 das Stapelobjekt im Konfigurationsraum an der gleichen Position von Schulter-*Pitch* und Schulter-*Roll* abgestellt wird, an der auch das Basisobjekt steht, gibt es auf Grund des unterschiedlichen Ellenbogenwinkels einen Versatz in der absoluten Position. Dies wird in Abbildung 52 deutlich und entspricht einem typischen Versatzfehler. Bei der Blickheuristik in Variante 1 tritt dieser Versatz ebenfalls auf, basiert hier jedoch auf der optischen Parallaxe. Das Stapeln mittels Blickheuristik funktioniert besonders gut bei Basisobjekten, die nah am Körper platziert sind und erspart das bei Variante 2 notwendige Referenzgreifen des Basisobjekts. Variante 2 ist zuverlässiger bei Objekten, die weiter vom Körper entfernt sind. Eine Analyse der Ursachen erfolgt in Kapitel 8.3 auf Seite 155.

Unabhängig vom Versatz sind diese einfachen sensomotorischen Schleifen in der Lage, den Roboter zum Stapeln der Objekte zu befähigen.



(a) Versatzfehler in der Frontansicht



(b) Versatzfehler in der Draufsicht

Abbildung 52: Versatzfehler vor dem Stapeln. Das Stapelobjekt wird je nach ausgeführter Variante auf Grund der optischen Parallaxe oder systematisch falscher Schulterwinkel versetzt auf das Basisobjekt abgestellt.

6.4 Linien nachzeichnen

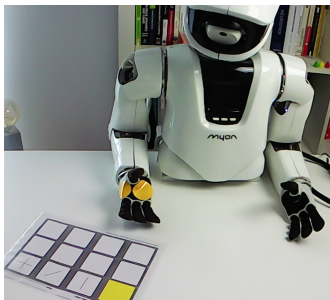
Innerhalb des Entwicklungstests ET 6-6-R bestehen die Tests T159 bis T161 für Kinder im Vorschulalter ab ca. 48 Monaten (T150 und T160) beziehungsweise ab ca. 60 Monaten (T161) im Nachzeichnen einer horizontalen Linie (T159), einer vertikalen Linie (T160) und einer diagonalen Linie (T161) [PM13].

Um die drei Linientypen nachzuzeichnen, muss der Roboter diese zunächst erfassen und dann an einem definierten Ort nachzeichnen. Wie zuvor gibt es keine Informationen über die Umwelt, wie zum Beispiel zur Tischoberfläche, zur Position, Größe und Rotation des Aufgabenblatts, der Inhalte der Aufgaben und so weiter. Einzige Annahme ist, dass die Felder der Vorlage und zum Nachzeichnen mit einem gelben Objekt präsentiert werden, und dass der Zeichenstift eine orange markierte Spitze zur visuellen Erkennung nach Kapitel 4.2.2 hat.

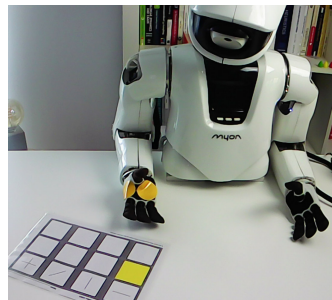
Im Folgenden wird der Ablauf des Nachzeichnens beschrieben:

- Dem Roboter wird der Stift in die Hand gegeben; die Roboterhand greift kraftgeregelt zu.
- Der Roboter zentriert seinen Blick auf das gelbe Quadrat soweit vorhanden. Das Quadrat wird vom Menschen auf die jeweilige Vorlage gelegt (horizontale, vertikale oder diagonale Linie, Abbildung 53a).
- Die Kopfbewegung wird fixiert und der Roboter reduziert sein Blickfeld mittels affiner Abbildung der PP auf den Ausschnitt des gelben Quadrats, um die Auflösung dort zu maximieren und umliegende Informationen zu reduzieren. Diese Funktion ist vergleichbar mit der der *Fovea centralis*, dem Bereich des schärfsten Sehens bei Säugetieren.
- Das gelbe Quadrat wird vom Menschen auf den Bereich gelegt, in dem die jeweilige Linie nachgezeichnet werden soll (Abbildung 53b).
- Der Roboter analysiert die noch immer im Bild befindliche Linie der gezeigten Vorlage. Die Informationen werden basierend auf dem aktuellen Bildausschnitt mit 25 x 20 Pixeln gespeichert.
- Der Roboter weitet sein Blickfeld wieder auf den maximal möglichen Bildausschnitt und zentriert das Bild durch eine Kopfbewegung mittels sensomotorischer Schleife auf das verschobene gelbe Quadrat – das Zielfeld.

- Das gelbe Quadrat wird wieder mittels affiner Abbildung der PP in maximaler Größe dargestellt.
- Der Mensch entfernt das gelbe Quadrat und das leere Zielfeld wird sichtbar.
- Der Stift wird anhand seiner orangefarbenen Spitze mittels sensomotorischer Schleifen, basierend auf den Ortshistogrammen des Vollbilds, in die Bildmitte bewegt.
- Die Kontrolle des Arms wird durch eine lokale sensomotorische Schleife übernommen, welche die Position im aktuellen 25×20 Pixel großen Bildausschnitt analysiert. Die Stiftspitze wird zum Startpunkt der Vorlagelinie bewegt. Während die Stiftspitze dem Aufgabenblatt durch eine Handgelenkdrehung angenähert wird, wird die Position adaptiv angepasst, bis die Stiftspitze das Aufgabenblatt berührt. Der Startpunkt ist beim rechten Arm immer die Ecke ganz links, damit die Hand die gezeichnete Linie nicht nachfolgend verdeckt.
- Die Stiftspitze wird mittels linearer Interpolation (Bresenham-Algorithmus) zum Endpunkt der Vorlagelinie bewegt.
- Der Stift wird vom Aufgabenblatt abgehoben und zur Seite bewegt.



(a) Gelbes Quadrat auf der Vorlage der horizontalen Linie



(b) Gelbes Quadrat auf einem der Zielfelder



(c) Myons Perspektive. Das gelbe Quadrat liegt auf der Vorlage der diagonalen Linie.

Abbildung 53: Vorlage und Zielfeld für die Aufgabe „Nachzeichnen“ einer horizontalen Linie. Das Zielfeld kann beliebig gewählt werden.

Damit die Stiftspitze unabhängig von der Position auf dem Aufgabenblatt immer den Kontakt zum Blatt behält, ohne dass der Stift zerdrückt wird, erfolgt eine Anpassung der Rotation des Handgelenks über die in Kapitel 5.1 vorgestellte sensomotorische Schleife im reaktiven System auf Basis der Beschleunigungswerte im Unterarm. Sowohl für die Erkennung des gelben Quadrats zur Berechnung der affinen Abbildung als auch zur Extraktion der Start- und Endpunkte der Vorlagelinie wird der

in Kapitel 4.2.3 vorgestellte *Harris-Detektor* eingesetzt. Der Zoom erfolgt wegen der niedrigen Auflösung in zwei Stufen, um eine gute Positionierung zu erreichen. Abbildung 54a zeigt die erste Zoomstufe: links das ursprüngliche Bild mit den rot markierten *Harris*-Eckenmerkmalen – die für die affine Abbildung relevanten Merkmale sind grün, gelb und blau hervorgehoben; rechts der darauf folgende Bildausschnitt. Analog gilt dies auch für die zweite Zoomstufe in Abbildung 54b.

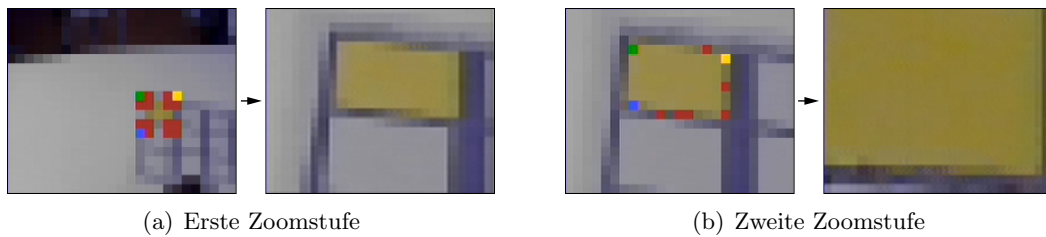


Abbildung 54: Zoomstufen basierend auf den erkannten Ecken des gelben Quadrats. Der Zoom vom Vollbild bis zum Quadrat erfolgt in zwei Stufen.

Die Analyse der Linien mithilfe der *Harris*-Eckenmerkmale wird in Abbildung 55 für die drei bekannten Linientypen des Entwicklungstests (a–c) sowie für eine gemalte Freihandlinie (d) gezeigt. Das Kamerabild wird dafür in ein Graustufenbild konvertiert, sodass der Detektor nur auf dem Luminanzkanal (Y) arbeitet. Die detektierten Eckenmerkmale sind rot markiert. Das Merkmal mit dem minimalen Abstand zur Ecke unten links ist blau markiert und wird als Startpunkt gemerkt. Das Merkmal mit dem minimalen Abstand zur Ecke oben rechts ist gelb markiert und wird als Endpunkt gemerkt.

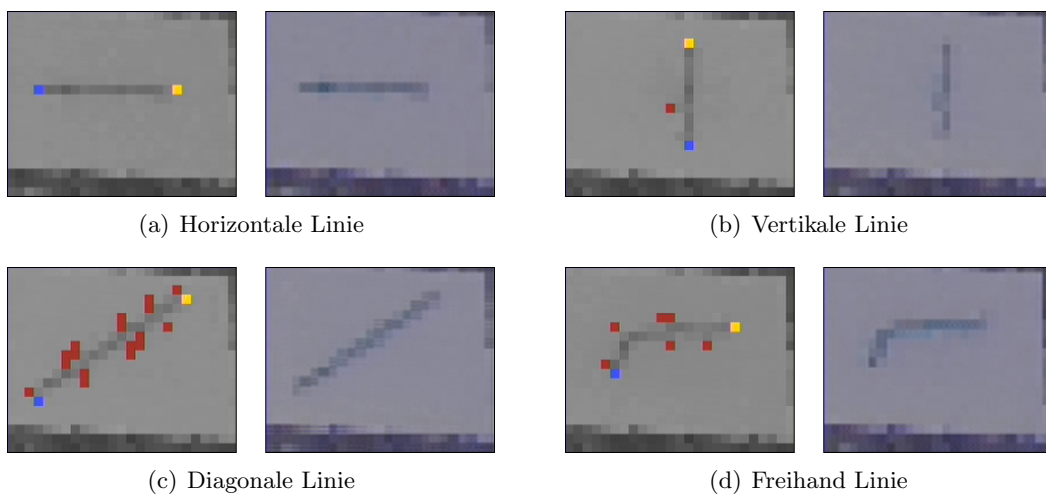
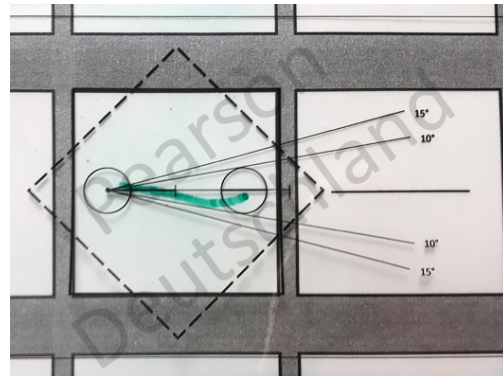


Abbildung 55: Erkannte Endpunkte der Linien. (links) Graustufenbild mit Harris-Eckenmerkmalen, (rechts) Originalbild der *PP*.

Abbildung 56a zeigt ein gutes Resultat der nachgezeichneten vertikalen Linie. Die Auswertung der gezeichneten Linie erfolgt mit der Testschablone des ET 6-6-R, wie in Abbildung 56b gezeigt. Das hier gezeigte Beispiel erfüllt die Kriterien bezüglich der Winkelabweichung und tolerierten Radien für den Start- und Endpunkt der Linie.



(a) Blick über die rechte Schulter des Roboters nach dem Nachzeichnen einer vertikalen Linie.

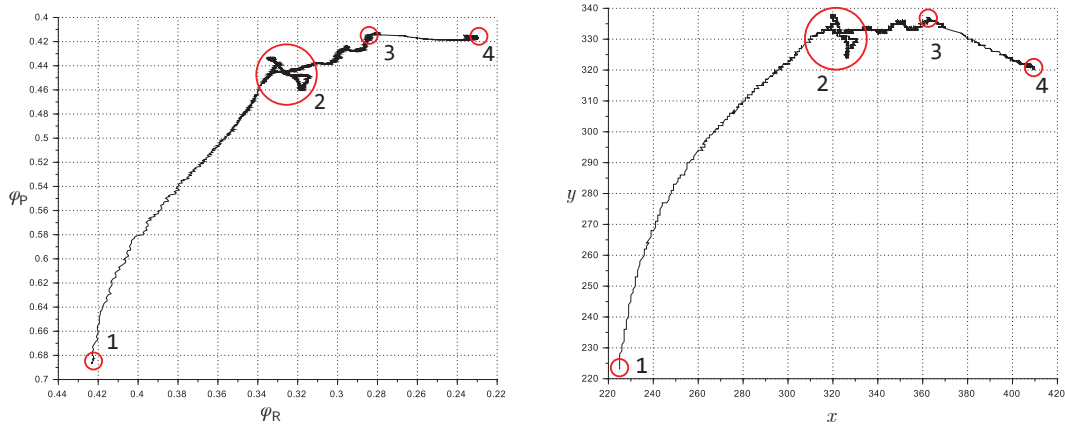


(b) Auswertung der durch den Roboter gezeichneten vertikalen Linie mit der Testschablone. Die Abbildung wurde zur besseren Lesbarkeit der Winkelskala um 90° im Uhrzeigersinn gedreht.

Abbildung 56: Resultat der durch den Roboter gezeichneten vertikalen Linie und Auswertung mit der ET 6-6-R-Testschablone [PM13].

Abbildung 57 zeigt den Verlauf der Hand (a) im Konfigurationsraum und (b) im Aktuationsraum beim Nachzeichnen einer horizontalen Linie. Das Ellenbogengelenk im Konfigurationsraum sowie die z-Koordinate im Aktuationsraum wurden für die Darstellung vernachlässigt. Der Ellenbogen folgt passiv der Tischoberfläche, sodass die z-Koordinate konstant ist.

Die Aufgabe *Nachzeichnen* wurde damit ohne Umweltinformationen allein auf Basis von sensomotorischen Schleifen und einfacher Heuristiken erfolgreich umgesetzt.



(a) Handposition im Konfigurationsraum des rechten Arms. Die Gelenkwinkel sind einheitenlos.

(b) Handposition im Aktuationsraum des Robotertorso. Die Werte sind in Millimeter angegeben.

Abbildung 57: Verlauf der Hand beim Nachzeichnen einer horizontalen Linie. (1) Beginn der Bewegung, (2) Erreichen des Anfangspunktes der Linie. Der Stift wird auf dem Aufgabenblatt aufgesetzt und die Position über das visuelle *Feedback* korrigiert. (3) Erreichen des Endpunktes der Linie. Der Stift wird vom Aufgabenblatt abgehoben. (4) Die Hand ist aus dem Sichtbereich bewegt worden – die Aufgabe ist abgeschlossen.

6.5 Zusammenfassung

Die visuomotorischen Handlungen, geleitet durch die vier ausgewählten entwicklungsdiagnostischen Tests, führen alle bisherigen Inhalte der Arbeit zusammen und werden durch den Roboter ausgeführt. Gerade die im Vergleich zum Menschen wenigen Freiheitsgrade im Arm des eingesetzten Roboters bieten gute Voraussetzungen für eine Verhaltensregelung auf Basis einfacher sensomotorischer Schleifen, die durch Heuristiken umgeschaltet, inhibiert oder kombiniert werden können. Die Aufgabe, ein Objekt visuell durch die Kopfbewegung zu verfolgen, erscheint trivial und kann ohne Probleme absolviert werden. Sie stellt jedoch eine wesentliche Kompetenz für die Lösung der folgenden drei Aufgaben dar. Beim „Objekt greifen“ erfolgt die Erkennung der Hand bevorzugt über eine Farbmarkierung auf dem Handkorpus. Diese Variante ist wesentlich genauer und vergleichbar rechenintensiv wie die Erkennung über erkannte Bewegungen der Hand im Kamerabild. Für das Greifverhalten werden sieben unabhängige sensomotorische Schleifen implementiert, die auch für die weiteren Aufgaben eingesetzt werden. Das „Objekt stapeln“ erfolgt basierend auf den Erkenntnissen des Greifens. Dabei stehen auch hier zwei Varianten zur Verfügung. Bei Basisobjekten nah am Roboterkörper eignet sich das Stapeln auf Basis der Blickheuristik sehr gut, da hier das vorherige Greifen des Basisobjektes wegfällt und der Parallaxefehler beim Abstellen des Stapelobjektes hinreichend klein ist. Steht das Basisobjekt weiter vom Körper weg, ist die Variante auf Basis der gespeicherten Objektkoordinaten im Konfigurationsraum zuverlässiger. Das „Nachzeichnen von Linien“ unterscheidet sich im Ablauf von den vorherigen Aufgaben. Zentrale Merkmale sind der zweistufige Zoom der *PP* auf die Vorlage beziehungsweise auf das Zielfeld, um die visuelle Aufmerksamkeit auf diese Bereiche zu fokussieren sowie die Erkennung der Vorlagelinie anhand der beiden Linienenden mit dem *Harris*-Eckendetektor. Das Nachzeichnen erfolgt durch eine lineare Interpolation der lokalen Bildkoordinaten. Die absolute Position der Linien ist für dieses Verhalten irrelevant. Grundsätzlich wurden alle vier Tests erfolgreich und reproduzierbar durchgeführt. Durch das Handeln unter begrenztem Wissen (*Bounded Rationality*) kommt es zwar sporadisch zu Fehlern bei der Ausführung, doch werden diese, wie in den kommenden Kapiteln aufgezeigt, zunehmend durch erweitertes Weltwissen reduziert.

7 Verhaltensoptimierung durch Quadriken

Bei den bislang vorgestellten Verfahren der Hand-Auge-Koordination in Kapitel 6 war die aktive Nutzung von Umgebungseinschränkungen, insbesondere der Tischoberfläche, grundlegender Bestandteil: Durch das relaxierte Ellenbogengelenk wurde die Hand über die Tischoberfläche geführt und der Konfigurationsraum des Arms auf die beiden Schultergelenke reduziert. Der Schutz des Handballens und der Tischoberfläche sowie ein ungünstiger Reibungskoeffizient zwischen Hand und Tischoberfläche sind nur ein Teil von Gründen, aus denen der Kontakt zwischen Hand und Tisch unerwünscht sein kann.

Wie der Mensch verfügt auch ein Roboter typischerweise nicht über eine propriozeptive Wahrnehmung der eigenen Körperproportionen. Der Mensch kompensiert dies, indem er durch die Interaktion mit der Umwelt ein implizites Modell seines Körpers bildet. Dieses Modell wird gemeinhin als Körpergefühl bezeichnet. Verändert sich die Morphologie des eigenen Körpers, sei es durch temporäre körperliche Einschränkungen oder sogar dauerhaft, zum Beispiel durch eine Amputation, ist der Mensch in der Lage, dieses Körpergefühl durch eine fortdauernde Interaktion mit der Umwelt an die neuen Gegebenheiten anzupassen. In der Robotik ist es üblich, dem System dieses Körpergefühl als explizites Modell mit allen Maßen, Massen, Drehmomenten, Getriebeübersetzungen und sonstigen Randbedingungen einzuprogrammieren. Da die Daten aus dem Konstruktionsprozess bereits vorliegen, erfordert dies in der Regel nur einen minimalen Mehraufwand für die Entwickler. Gibt es Veränderungen in der Morphologie oder dem Antriebsstrang, zum Beispiel durch Greiferwechsel, andere Raddurchmesser durch Abnutzung oder Ablagerungen, verschmutzte Getriebe, Lagerverschleiß oder durch sonstige Umbauten beziehungsweise Schäden, müssen die Modellparameter aktiv angepasst werden. Anhand dieser expliziten Körpermodelle lassen sich komfortabel Vorwärts-Kinematiken berechnen, beispielsweise mittels Denavit-Hartenberg-Transformation (siehe Kapitel 5.2), um die Position einzelner Glieder zuverlässig vom Konfigurationsraum in den Aktuationsraum zu überführen. Für die Pfadplanung wird bevorzugt die inverse Kinematik berechnet, anhand derer die einzelnen Gelenkwinkel für geforderte Raumkoordinaten des Greifers bestimmt werden [LW08]. Enorm viele Beiträge in der Fachliteratur beschäftigen sich damit, diese Bewegungen trotz vorhandener Singularitäten³⁵ und mit der idealen Trajektorie auszuführen.

³⁵Bei einer Roboter-Kinematik tritt eine Singularität dann auf, wenn es für einen bestimmten Raumpunkt mehrere Möglichkeiten gibt, diesen zu erreichen; die Abbildung dieses Punktes vom Aktuationsraum in den Konfigurationsraum ist nicht eindeutig.

Da die vorgestellten Verfahren allesamt einen hohen Rechenaufwand erfordern und zudem recht unflexibel sind, wird in dieser Arbeit ein alternativer Weg eingeschlagen.

Eine Möglichkeit, wie der Roboter durch Selbstexploration ein schlankes und implizites Modell bilden und jederzeit aktualisieren kann, ist die Abbildung von Bewegungen mittels Quadriken. Es handelt sich dabei nicht um ein direktes Körpermodell, sondern um die Darstellung des Zusammenhangs zwischen der Arm-Hand-Einheit und der Umwelt. Dass sich Quadriken grundsätzlich für die Armkoordination eines humanoiden Roboters eignen, wurde in *Quadric-Representing Neurons (QRENS)* [HKH11] bereits gezeigt. Dieses Konzept wurde hier weiterentwickelt und in die Praxis überführt. Während der Aktuationsraum alle durch die Hand erreichbaren Punkte im kartesischen Raum (x, y, z) umfasst, enthält der Konfigurationsraum alle von den drei Armgelenken erreichbaren Punkte $(\varphi_P, \varphi_R, \varphi_E)$. Als Arbeitsbereich ist die Teilmenge des Aktuationsraums definiert, die zusätzlich unter einer Randbedingung gültig ist. Eine Randbedingung ist beispielsweise, dass die Hand 6 cm oberhalb der Tischoberfläche sein muss.

Der Einsatz von Quadriken ist vor allem in der Computergrafik üblich. Mit Ihnen können dreidimensionale Strukturen vereinfacht werden. Ein klassisches Verfahren ist der Garland-Heckbert-Algorithmus, bei dem das Objekt inkrementell durch Dreiecke vereinfacht wird, ohne dass dessen Fehler (in Form einer sogenannten Fehlerquadrik) größer wird [GH97]. Jüngste Publikationen stellen probabilistische Quadriken [TK20] vor, die selbst bei verrauschten Daten homogene Dreieck-Netze generieren können, wo klassische Verfahren versagen und die Oberfläche unnötig hochauflösend darstellen müssen.

Nicholson et. al. [NMS18] und Liao et. al. [LWQZ20] nutzen Quadriken als Objektmodelle für einen semantischen *SLAM*-Algorithmus (*Simultaneous Localization And Mapping*) zur Lokalisation. Die Attraktivität von Quadriken liegt in diesem Fall in der kompakten Darstellung der Größe, Position und Orientierung von dreidimensionalen Objekten. Die Kombination von *SLAM*-Methoden sowie die Nutzung von *CNNs* liefert beachtliche dreidimensionale Repräsentationen von Objekten in Form von einfachen Quadriken aus einem monokularen Kamerabild [HLLR19].

Grundsätzlich erfolgt der Einsatz von Quadriken in der Robotik primär zur Modellierung von zu greifenden Objekten; diese lassen sich in der Regel gut mit Funktionen zweiten Grades beschreiben. Sind die Möglichkeiten der Modellierung mit Polynomen zweiten Grades nicht genau genug, lassen sich mehrere Quadriken kombinieren oder sogenannte Superquadriken einsetzen. Diese ähneln formal den Quadriken, jedoch werden die Quadrierungen durch beliebige andere Potenzen ersetzt

und ermöglichen damit die Modellierung von beispielsweise würfelförmigen Objekten mit variabel scharfen Kanten (Superellipsoid) [OSG20]. Dies vervielfacht die Einsatzmöglichkeiten bei geringfügig komplexerer formaler Beschreibung. Ein aktuelles Beispiel zeigt [VPPN18].

Quadriken

Eine Quadrik Q ist die Lösungsmenge einer quadratischen Gleichung mit mehreren Unbekannten [OSG20]. Formal gilt

$$Q = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b} \mathbf{x} + c = 0\}, \quad (9)$$

wobei $\mathbf{A} \in \mathbb{R}^{n \times n}$ symmetrisch ist, $\mathbf{b} \in \mathbb{R}^n$ und $c \in \mathbb{R}$.

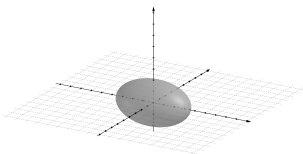
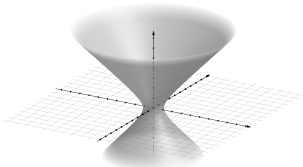
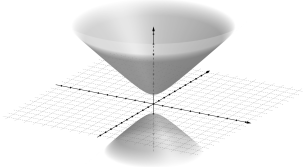
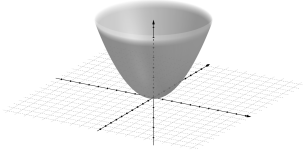
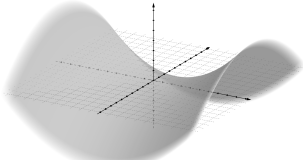
Durch eine Hauptachsentransformation kann jede Quadrik auf eine der 17 möglichen Normalformen transformiert und dadurch kategorisiert werden. 14 Normalformen stellen geometrische Gebilde dar, die bis zur dritten Ordnung ($n = 3$) gut visualisiert und optisch verglichen werden können. Um eine Quadrik zu normalisieren, werden nacheinander die gemischten Terme der zweiten Ordnung (nicht-diagonale Elemente von \mathbf{A} sind alle null) und, falls möglich, auch die linearen Terme ($\mathbf{b} = 0$) und die Konstante ($c = 0$) entfernt. Ein paar Beispiele nicht entarteter Quadriken im \mathbb{R}^3 , die in allen Richtungen gekrümmte Flächen haben, zeigt Tabelle 4 mit den Parametern $a, b, c > 0$.

Mit Quadriken dritter Ordnung lassen sich ausgewählte Armbewegungen des Roboters Myon sehr gut abbilden [Sel05]. Die Gelenke Schulter-*Pitch* φ_P , Schulter-*Roll* φ_R und Ellenbogen φ_E repräsentieren jeweils eine der drei Dimensionen. Solange sich die Position der Hand im Konfigurationsraum auf der Quadrik befindet, z.B. parallel zur Tischoberfläche im Aktuationsraum, ergibt die entsprechende implizite³⁶ quadratische Funktion null.

Im Vergleich zu Kubiken (Polynomen dritten Grades) und Polynomen noch höheren Grades haben Quadriken wesentlich weniger singuläre Punkte und ausgewählte Normalformen sowie auf Grund der einfacheren Struktur im \mathbb{R}^3 ein wesentlich geeigneteres Extrapolationsverhalten für die hier relevante Anwendung. Die Extrapolationseigenschaften werden im Folgenden beim Vergleich von *Fitting*-Verfahren genauer betrachtet.

³⁶Definiert die Gleichung $f(x, y, z) = 0$ die Variable z als Funktion von x und von y , handelt es sich um eine implizite Darstellung der Funktion z . Gleiches ist ebenfalls für x und y als Funktion der übrigen Variablen gültig [MW06].

Tabelle 4: Normalformen nicht entarteter Quadriken im \mathbb{R}^3 mit $a, b, c > 0$. Nicht entartete Quadriken haben in allen Richtungen gekrümmte Flächen. Entartete Quadriken haben in einer oder mehreren Richtungen gerade Elemente oder mindestens eine Variable mit dem Wert null.

Ellipsoid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0$	
Einschaliges Hyperboloid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} - 1 = 0$	
Zweischaliges Hyperboloid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} + 1 = 0$	
Elliptisches Paraboloid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 2z = 0$	
Hyperbolisches Paraboloid	$\frac{x^2}{a^2} - \frac{y^2}{b^2} - 2z = 0$	

Um die Komplexität der Bewegungsplanung zu reduzieren, wird je nach Anwendung eine Quadrik ausgewählt und die Bewegung im Konfigurationsraum auf den zugehörigen Arbeitsbereich, hier als Schnittmenge des Aktuationsraums und der Quadrik, beschränkt. Für diese Schnittmenge ist die Quadrik invariant. Durch die Definition als implizite Funktion Q müssen nur zwei der drei Gelenkwinkel vorgegeben werden. Das dritte Gelenk ergibt sich automatisch. Auf Grund der fehlenden Distanzinformation im Kamerabild empfiehlt es sich, das Ellenbogengelenk implizit über die beiden Schultergelenkwinkel zu definieren und damit die Nutzung der Tischoberfläche als Umweltbeschränkung zu ersetzen.

7.1 Verfahren zur Generierung von Quadriken

Durch die Bewegung der Roboterhand im von der Quadrik abzubildenden Bereich entstehen mit jeder Abtastung und Speicherung der Armwinkel weitere Daten in einer Punktwolke. Um Quadriken zu berechnen, die den gesamten Arbeitsbereich umfassen, sollte auch die zugrunde liegende Punktwolke diesen Bereich abdecken. Für eine effiziente Berechnung auf einem eingebetteten System ist die folgende Notation der Quadriken als Matrizenprodukt vorteilhaft:

$$Q = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{w}^T \mathbf{f}_n(\mathbf{x}) \stackrel{!}{=} 0\} \quad (10)$$

wobei $\mathbf{f}_n : \mathbb{R}^n \rightarrow \mathbb{R}^m$ die Erweiterung von \mathbf{x} ist, inklusive aller quadratischer Terme und der Konstanten 1, $\mathbf{w} \in \mathbb{R}^m$ ist ein Gewichtsvektor mit $m = \frac{n(n+3)}{2} + 1$.

Durch die drei relevanten Armgelenke ist $n = 3$, und der Parametervektor hat die Dimension $m = 10$. Es folgt

$$\begin{aligned} \mathbf{x} &= (x_1 \ x_2 \ x_3)^T, \\ \mathbf{f}_3(\mathbf{x}) &= (x_1^2 \ x_2^2 \ x_3^2 \ x_1x_2 \ x_2x_3 \ x_1x_3 \ x_1 \ x_2 \ x_3 \ 1)^T, \\ \mathbf{w} &= (w_1 \ w_2 \ \dots \ w_{10})^T. \end{aligned}$$

In dieser Arbeit werden die Winkelwerte der drei Gelenke Schulter-*Pitch* φ_P , Schulter-*Roll* φ_R und Ellenbogen φ_E genutzt. Diese werden anhand von Potentiometerwerten ermittelt, die absolute Winkel zwischen den Gelenken bei geringem Rauschen bereitstellen.

Dies ergibt folgende Vektoren:

$$\begin{aligned}\boldsymbol{\varphi} &= (\varphi_P \varphi_R \varphi_E)^T, \\ \mathbf{f}_3(\boldsymbol{\varphi}) &= (\varphi_P^2 \varphi_R^2 \varphi_E^2 \varphi_P \varphi_R \varphi_E \varphi_P \varphi_E \varphi_P \varphi_R \varphi_E 1)^T, \\ \mathbf{w} &= (w_1 w_2 \dots w_{10})^T.\end{aligned}$$

Da die impliziten Modelle auf jedem Roboter individuell, schnell und adaptiv generiert werden können, ist eine Kalibrierung der Potentiometer nicht notwendig.

Alternativ könnten die Beschleunigungssensoren in Unterarm, Oberarm und Torso genutzt werden, was einen entscheidenden Vorteil hätte: Die gelieferten Beschleunigungswerte sind auf die Umwelt bezogen Absolutwerte, die bei jedem Roboter – unabhängig von einer genauen Kalibrierung – gleich sind. Die Potentiometer liefern hingegen ein rauschärmeres Signal und sind unanfälliger gegenüber Vibrationen durch die Motoren und gegenüber äußeren Bewegungen durch parallel ausgeführte Aktionen anderer Extremitäten. Insbesondere die Gelenkansteuerung mittels PMS/PPMS ist deutlich durch eine parasitäre Überlagerung von Vibrationen des *Choppings* in den Werten der Beschleunigungssensorik zu erkennen, weshalb diese hier nicht eingesetzt werden.

Da es sich bei diesem Problem um ein unrestringiertes Optimierungsproblem handelt, gibt es viele Möglichkeiten den Gewichtungsvektor \mathbf{w} zu berechnen. Wie Punktwolken auf Quadriken abgebildet werden können, wurde bereits intensiv untersucht (Ausgewählte Beispiele sind [Agi81], [CZ98], [LH91] und [RK10]). Die Qualität der unterschiedlichen Verfahren im Allgemeinen wurde von Dai [DN98] und Petitjean [Pet02] verglichen. Mit der Stabilität des Least-Squares-Verfahrens, welches im Folgenden noch eingeführt wird, hat sich Halir [HF98] beschäftigt.

Im Rahmen dieser Arbeit werden das *Least-Squares*-Verfahren (LS) als Referenz, das rekursive *Least-Squares*-Verfahren (RLS), sowie das Gradientenabstiegsverfahren (GA) auf Basis des LS eingesetzt. Ziel der erneuten Experimente ist es, die Verfahren speziell auf die Anforderungen bei der Bewegung eines Roboterarms zu vergleichen. Das beinhaltet die effiziente Berechnung auf einem eingebetteten System mit hinreichend genauen Resultaten, die Stabilität der resultierenden Regelung auf Basis der Quadriken sowie die Extrapolationsfähigkeit der resultierenden Quadriken über die zugrunde liegenden Stützstellen hinaus.

Fehlerberechnung für die Optimierung

Alle drei Verfahren basieren auf der Minimierung der Fehlerquadratsumme. Sie können eingesetzt werden, wenn ein Modell, wie die Quadriken, lineare Ausgangsparameter besitzt, und wenn mindestens so viele Trainingsdatenpaare vorliegen wie es zu optimierende Parameter gibt, mit der Bedingung, dass diese Trainingsdaten nicht linear abhängig ist.

Der quadratische Fehler E über n Punkte berechnet sich anhand folgender Formel.

$$E(\mathbf{w}, \boldsymbol{\varphi}) = \sum_{i=1}^n (\mathbf{w}^T \mathbf{f}_3(\boldsymbol{\varphi}_i))^2 \quad (11)$$

Zusätzlich wird der quadratische Fehler E um eine Ridge-Regularisierung ergänzt. Diese wird dazu eingesetzt, die Beträge von \mathbf{w} minimal zu halten. Es gilt

$$E_R(\mathbf{w}, \boldsymbol{\varphi}) = \zeta \sum_{i=1}^n (\mathbf{w}^T \mathbf{f}_3(\boldsymbol{\varphi}_i))^2 + (1 - \zeta)(\mathbf{w}^T \mathbf{w}), \quad (12)$$

mit $\zeta \in \{0 \dots 1\}$. Der Ausdruck $(\mathbf{w}^T \mathbf{w})$ entspricht der Summe der Wichtigungsquadrate.

Das Gewicht w_{10} wird konstant auf $w_{10} = 1$ gesetzt und damit eine Optimierung von \mathbf{w} zu einem Nullvektor vermieden. Wird vom Ausdruck 1 subtrahiert, wird $w_{10} = 1$ eliminiert und die verbleibenden neun variablen Wichtigungen w_k mit $k \in \{1 \dots 9\}$ werden mittels der genannten Verfahren optimiert.

Least-Squares-Verfahren

Das *Least-Squares*-Verfahren (LS) basiert auf der Matrixinversion der Datenmatrix. Da diese in der Regel nicht quadratisch ist, ist sie auch nicht invertierbar. Daher wird hier die Moore-Penrose-Inverse (Pseudoinverse, formal durch das Symbol $^+$ beschrieben) berechnet.

Die Gewichte \mathbf{w} berechnen sich mit der Matrix $\boldsymbol{\phi}$ aus n Messwertvektoren

$$\boldsymbol{\phi} = (\boldsymbol{\varphi}_1 \boldsymbol{\varphi}_2 \cdots \boldsymbol{\varphi}_n) \quad (13)$$

und der punktweisen Basiserweiterung \mathbf{f}_3 auf die Matrix $\boldsymbol{\phi}$ zu

$$\mathbf{w}^T = \mathbf{f}_3(\boldsymbol{\phi})^+ (-\mathbf{1} \cdot \mathbf{1})^T \quad (14)$$

mit dem Vektor $\mathbf{1}$ aus n Einsen bei n Messwertvektoren.

Hiermit lässt sich komfortabel die optimale Lösung des potentiell überbestimmten Gleichungssystems berechnen. Da es sich um ein Offline-Verfahren handelt, bei dem vor der Berechnung alle Datenpaare vorliegen müssen, und dieses zudem mit zunehmender Anzahl an Datenpunkten in seiner Rechenkomplexität zunimmt, ist dieses Verfahren nur bedingt für die Echtzeitberechnung auf einem robotischen System geeignet.

Rekursives-Least-Squares-Verfahren

Das erste hier vorgestellte Online-Verfahren ist das rekursive *Least-Squares*-Verfahren (RLS) [Hay14]. Dabei werden in jedem Zeitschritt t die Updategleichungen für den Gewichtsvektor \mathbf{w}_t sowie für die Matrix \mathbf{P}_t berechnet. Die als Rekursionsformel bezeichnete Gleichung zur Berechnung der Matrix \mathbf{P}_t ist die Schätzung der Inverse der Autokorrelationsmatrix $(\mathbf{f}_3(\varphi))(\mathbf{f}_3(\varphi))^T$. Der Rechenaufwand bleibt in jedem Schritt gleich. Bei diesem Verfahren entfällt die Matrixinversion und wird durch geschickte Umformung zu einer Division durch einen Skalar. Die folgende mathematische Beschreibung basiert auf den Ausführungen von [Sch17].

Verstärkungsvektor γ_t :

$$\gamma_t = \frac{\mathbf{P}_t \cdot \mathbf{f}_3(\varphi)_{t+1}}{\lambda + \mathbf{f}_3(\varphi)_{t+1}^T \cdot \mathbf{P}_t \cdot \mathbf{f}_3(\varphi)_{t+1}} \quad (15)$$

Korrekturterm β_t :

$$\beta_{t+1} = (-1) - \mathbf{f}_3(\varphi)_{t+1}^T \cdot \mathbf{w}_t \quad (16)$$

Updategleichung \mathbf{w}_{t+1} :

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \gamma_t \cdot \beta_{t+1} \quad (17)$$

Rekursionsformel \mathbf{P}_{t+1} :

$$\mathbf{P}_{t+1} = \frac{1}{\lambda} [\mathbf{P}_t - \gamma_t \cdot \mathbf{f}_3(\varphi)_{t+1}^T \cdot \mathbf{P}_t] \quad (18)$$

Fehler E_t über aktuelle Messwerte von $\mathbf{f}_3(\varphi)$:

$$E_t = (\mathbf{w}^T \cdot \mathbf{f}_3(\varphi)_t)^2 \quad (19)$$

Fehler $E_{t,m}$ über alle m Messwerte von $\mathbf{f}_3(\varphi)$:

$$E_{t,m} = \sum_{t=0}^{t=m} (E_t) \quad (20)$$

Zusätzlich wird ein Vergessensfaktor $\lambda = 0,9 \dots 1$ eingesetzt. Geht λ gegen 1, verbessert sich die Elimination von Störeinflüssen. Im Gegensatz dazu folgt das Verfahren mit einem Vergessensfaktor kleiner 1 zunehmend besser zeitveränderlichen Parametern. Ist anzunehmen, dass die relevante Quadrik sich verändert hat, in dem die Hand zum Beispiel nicht mehr auf der Tischoberfläche bewegt wird, oder der Roboter sich in seiner Umwelt bewegt hat, kann ein starkes Vergessen der vergangenen Schritte explizit erwünscht sein.

Bei dem hier genutzten Vergessensfaktor handelt es sich um ein exponentielles Vergessen, da die j Zeitschritte in der Vergangenheit mit λ^j vergessen werden.

Die Initialisierung des Gewichtsvektors \mathbf{w} kann beliebig gewählt werden und erfolgt im Rahmen dieser Arbeit durch einen Nullvektor. Die \mathbf{P} -Matrix wurde nach Schröder [Sch17] mit Zufallswerten zwischen 100 und 1000 als obere Dreiecksmatrix initialisiert.

Gradientenabstiegs-Verfahren

Das Gradientenabstiegsverfahren (GA) ist ein weiteres Online-Verfahren, bei dem bei jeder Berechnung die zuvor zufällig gewählten Gewichte der Quadrik anhand der partiellen Ableitungen nach den einzelnen Gewichten angepasst werden, sodass der resultierende Fehler mit den Iterationen abnimmt.

Die partiellen Ableitungen ergeben sich zu

$$\frac{\partial E_R(\mathbf{w}, \varphi)}{\partial \mathbf{w}} = 2 \zeta \mathbf{f}_3(\varphi) (\mathbf{w}^T \mathbf{f}_3(\varphi))^T + 2 (1 - \zeta) \mathbf{w} \quad (21)$$

Damit der resultierende Fehler minimal wird, werden die partiellen Ableitungen zum Wert Null optimiert. Dadurch hat der Faktor 2 in den beiden Summanden keinen Einfluss und wird im Folgenden weggelassen.

Aufgelöst ergibt sich

$$\frac{\partial E_R(\mathbf{w}, \varphi)}{\partial w_1} = \zeta \varphi_{\text{ShP}}^2 \mathbf{C} + (1 - \zeta) w_1, \quad (22)$$

$$\frac{\partial E_R(\mathbf{w}, \varphi)}{\partial w_2} = \zeta \varphi_{\text{ShR}}^2 \mathbf{C} + (1 - \zeta) w_2, \quad (23)$$

⋮

$$\frac{\partial E_R(\mathbf{w}, \varphi)}{\partial w_9} = \zeta \varphi_{\text{El}} \mathbf{C} + (1 - \zeta) w_9, \quad (24)$$

wobei der Ausdruck $\mathbf{C} = \mathbf{w}^T \mathbf{f}_3(\varphi)$ für jedes k von w_k gleich bleibt und somit nur einmal je Schritt berechnet werden muss.

Der neue Wichtungsvektor \mathbf{w} wird berechnet durch

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \Delta \mathbf{w}_t \quad (25)$$

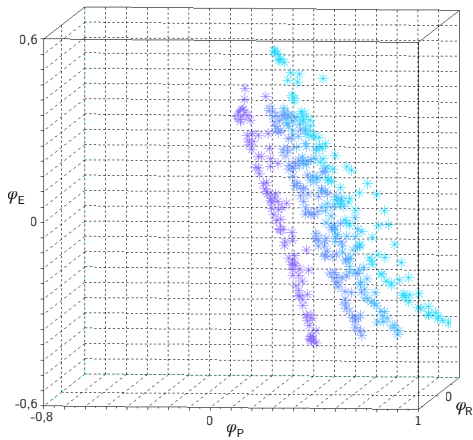
mit

$$\Delta \mathbf{w}_t = -\varepsilon \frac{\partial E_R(\mathbf{w}, \varphi)}{\partial \mathbf{w}}. \quad (26)$$

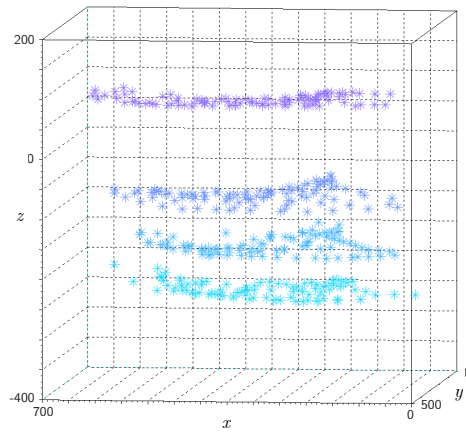
Die Lernrate ε liegt im Bereich $0 < \varepsilon \ll 1$. Der Startwert beträgt im Rahmen dieser Arbeit $\varepsilon = 0.1$ und wird immer wenn der Fehler E_R zunimmt halbiert. Nach 200 Iterationen wird die Berechnung unabhängig von der aktuellen Lernrate abgebrochen und die Quadrik als hinreichend genau angenommen. Zu Beginn wird der Gewichtsvektor \mathbf{w} mit Zufallswerten zwischen -1 und 1 initialisiert. Bei den Verfahren im folgenden Kapitel 7.4 zur Optimierung der Quadriken während der Hand-Auge-Koordination wird die maximale Anzahl an Iterationen zuvor festgelegt oder beim Unterschreiten einer definierten Lernrate abgebrochen. Die Gewichte werden nicht mehr zufällig, sondern auf Basis vorhandener Quadriken initialisiert.

Beispielquadriken

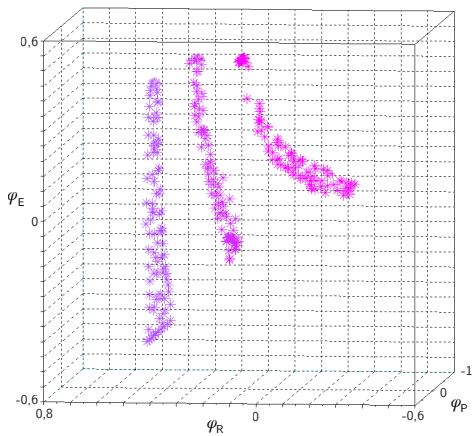
Zur Analyse der resultierenden Quadriken sind die Gelenkwinkel für verschiedene Anwendungen aufgezeichnet und ausgewertet worden. Dafür wurden zunächst vier Messreihen aufgenommen, deren Punkte auf einer definierten Höhe oberhalb der Tischoberfläche liegen (Abbildungen 58(a) und 58(b)) sowie drei Messreihen, deren Punkte parallel zur mittleren Sagittalebene des Roboters liegen (Abbildungen 58(c) und 58(d)) und damit vertikal verlaufende Ebenen im Aktuationsraum darstellen. Die Auswertung basiert auf 100 Messpunkten je Messreihe und Quadrik. Die Messreihe wird bei den iterativen Verfahren (RLS und GA) zwei Mal in Folge genutzt, sodass jeder Punkt zwei Mal, jedoch in zufälliger Reihenfolge berücksichtigt wird (200 Iterationen). Je Verfahren (RLS und GA) werden 30 Gewichtsvektoren mit unterschiedlichen Startwerten berechnet. Mit den Abbildungen 59–65 auf den Seiten 111–117 sind sieben prototypische Ebenen dargestellt. Die grünen Markierungen in den Abbildungen (a–c) zeigen die der Berechnung zugrunde liegenden Messpunkte. Der Vergessensfaktor beim RLS ist mit 0,9 sehr niedrig gewählt, was das Rauschen im Fehler erhöht, jedoch die Anzahl an totalen Fehlläufern reduziert und diese schneller eliminiert. Die Regularisierung beim GA ist mit 0,99 als praktikabler Mit-



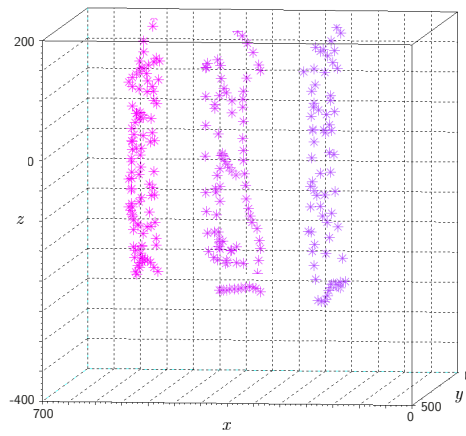
(a) Vier Messreihen mit je 100 Punkten, die jeweils auf einer definierten Höhe oberhalb der Tischoberfläche liegen, dargestellt im Konfigurationsraum des rechten Armes. Die Gelenkwinkel sind einheitenlos.



(b) Vier Messreihen mit je 100 Punkten, die jeweils auf einer definierten Höhe oberhalb der Tischoberfläche liegen, dargestellt im Aktuationsraum des Roboters. Die Achsen sind in mm skaliert.



(c) Drei Messreihen mit je 100 Punkten, die parallel zur Sagittalebene des Roboters liegen, dargestellt im Konfigurationsraum des rechten Armes. Die Gelenkwinkel sind einheitenlos.



(d) Drei Messreihen mit je 100 Punkten, die parallel zur Sagittalebene des Roboters liegen, dargestellt im Aktuationsraum des Roboters. Die Achsen sind in mm skaliert.

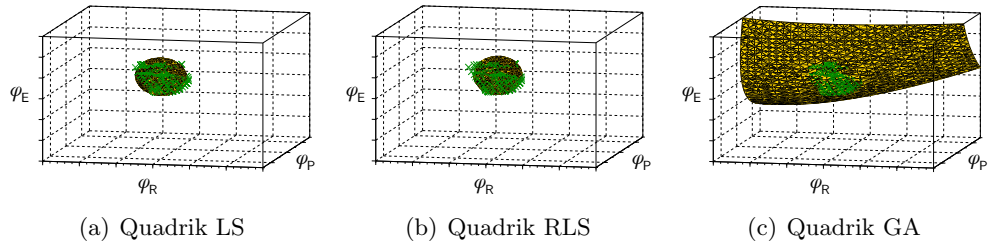
Abbildung 58: Sieben Punktwolken mit definierter Position als Grundlage für die Berechnung von Beispielquadriken im Aktuationsraum des rechten Armes. Die Punktwolken in den Abbildungen b und d entsprechen denen in den Abbildungen a und c. Sie wurden lediglich mittels der Vorwärtskinematik des Roboters vom Konfigurationsraum in den Aktuationsraum überführt.

telweg gewählt. Dieser führt zu kleinen Gewichten, einer nur leicht verlangsamten Konvergenz, sowie zu einer kaum vergrößerten Streuung im Fehler innerhalb der 30 parallelen Optimierungen.

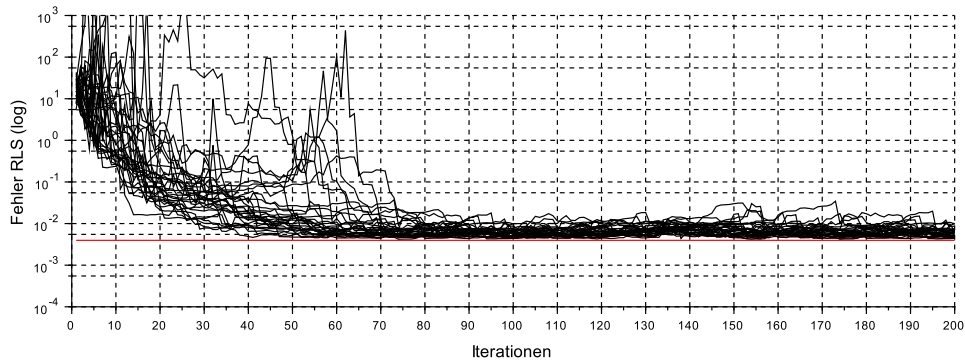
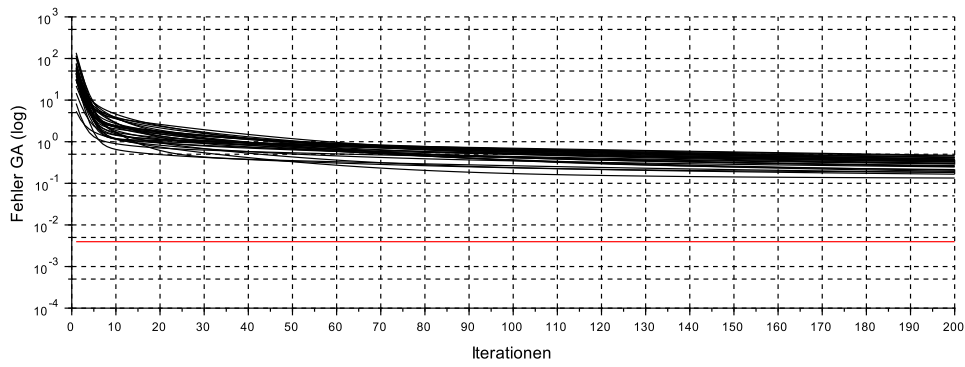
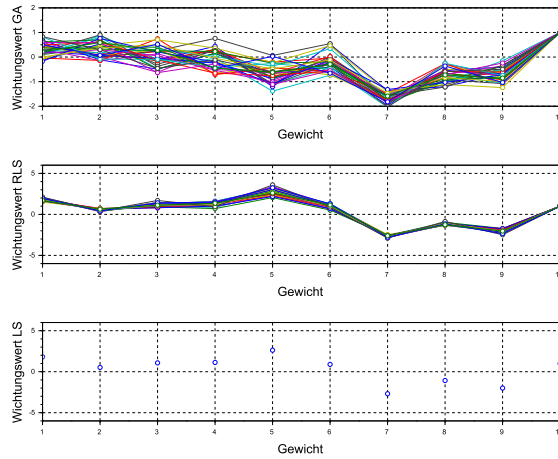
Das LS-Verfahren scheidet für die Anwendung auf dem Roboter aus, da es sich um ein Offline-Verfahren handelt, welches alle Daten gesammelt benötigt und die

optimale Lösung in einem Durchgang berechnet. Für die Praxis sind dagegen Online-Verfahren relevant, die die Schätzung der Quadrik während des Betriebes optimieren.

Die dargestellten Beispiele zeigen, dass das RLS-Verfahren nach gut 100 Iterationen sehr gute Ergebnisse erzielt und Quadriken generiert, welche sowohl qualitativ als auch quantitativ der idealen Lösung aus dem LS-Verfahren mit 100 Messwerten recht nahe kommen. Der Gradientenabstieg konvergiert selbst nach 200 Iterationen deutlich langsamer als das RLS-Verfahren. Deutlich beim GA ist auch das geringe Rauschen zwischen den Fehlern der auf den 30 parallelen Gewichtsvektoren basierenden Quadriken. Im Vergleich zum RLS-Verfahren hat der GA nach bereits 10 bis 30 Iterationen eine gute Schätzung der Quadrik erreicht.

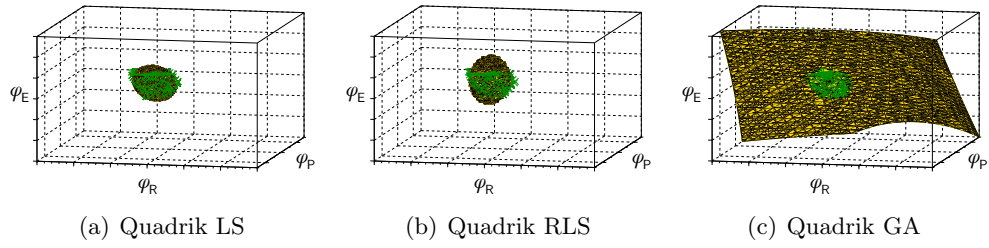


(d) Versuchsaufbau

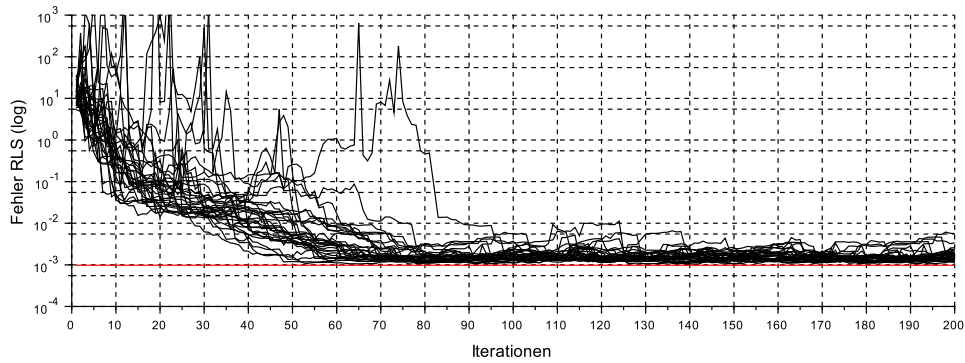
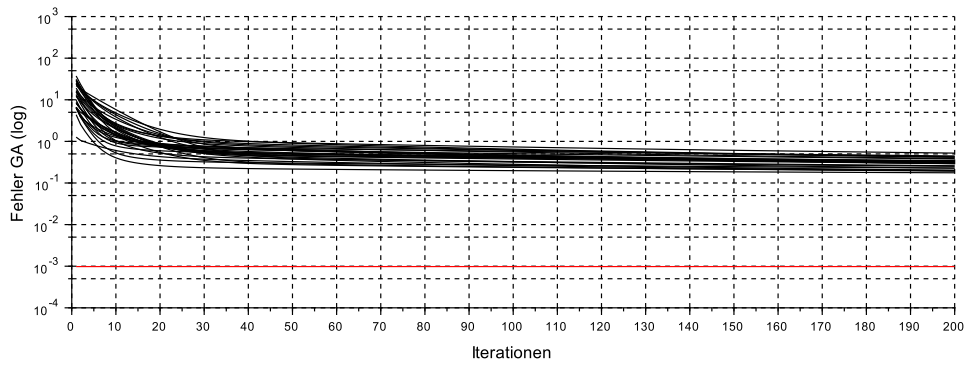
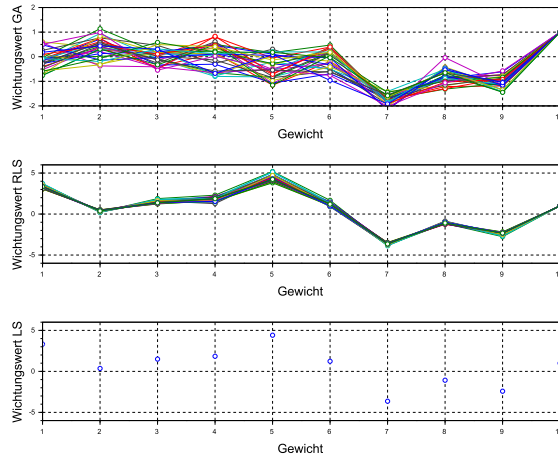


(f) Fehler über Iterationen: (oben) GA, (unten) RLS, (rote Linie) LS

Abbildung 59: Quadrik des gesamten Aktionsraumes mit der Bedingung, dass die Hand permanent Kontakt zur Tischoberfläche hat.

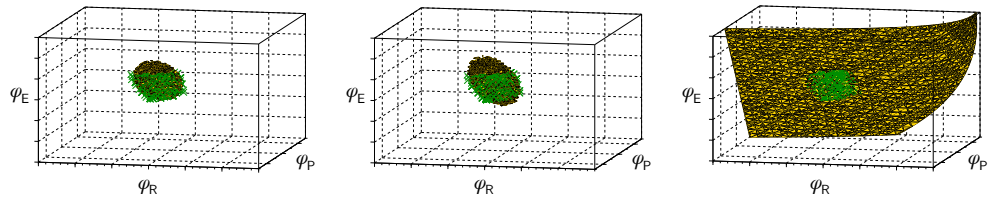


(d) Versuchsaufbau



(f) Fehler über Iterationen: (oben) GA, (unten) RLS, (rote Linie) LS

Abbildung 60: Quadrik des gesamten Aktionsraumes mit der Bedingung, dass die Hand einen konstanten Abstand von 6 cm zur Tischoberfläche hat.



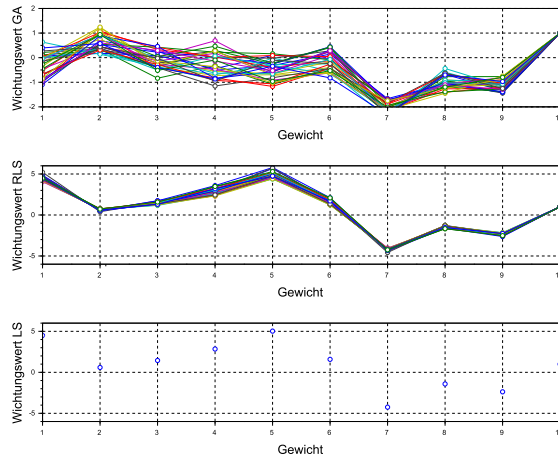
(a) Quadrik LS

(b) Quadrik RLS

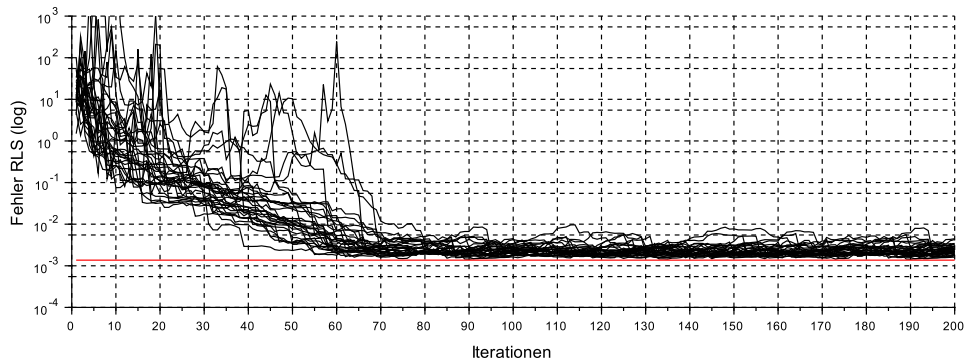
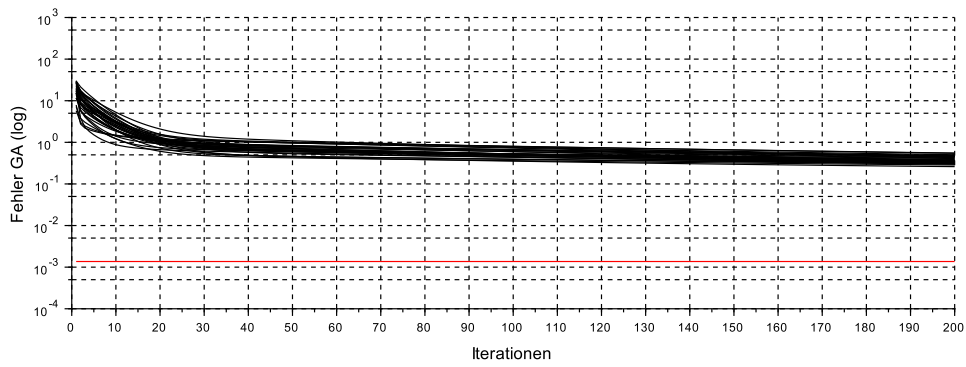
(c) Quadrik GA



(d) Versuchsaufbau

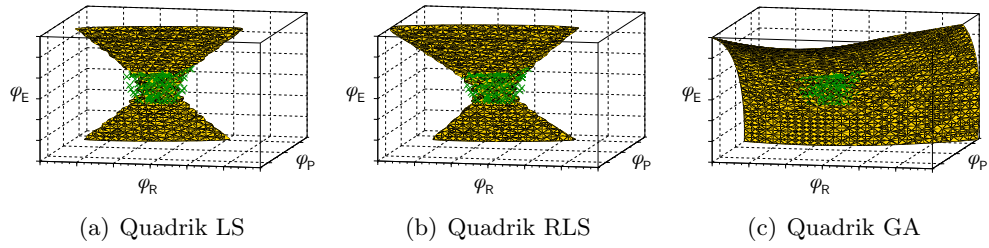


(e) Gewichtsvektoren

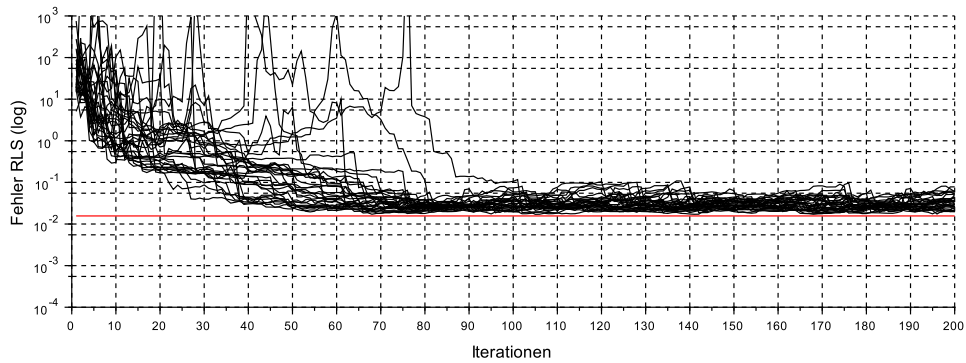
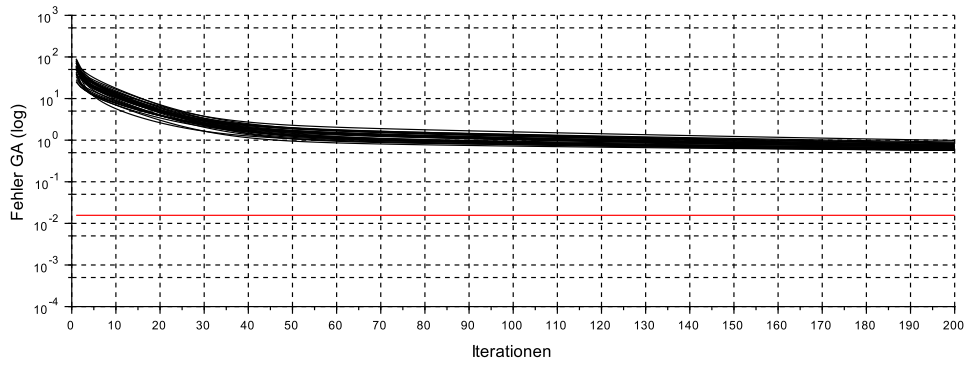
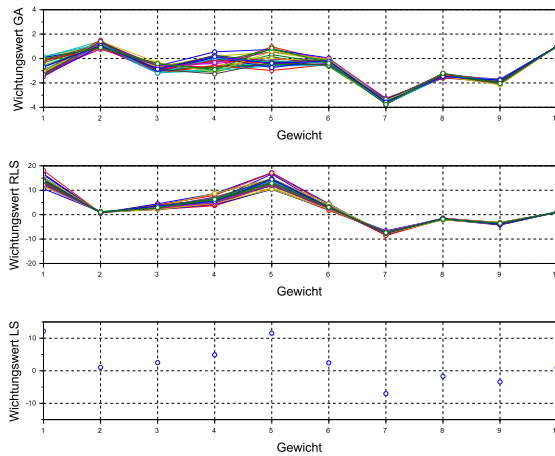


(f) Fehler über Iterationen: (oben) GA, (unten) RLS, (rote Linie) LS

Abbildung 61: Quadrik des gesamten Aktionsraumes mit der Bedingung, dass die Hand einen konstanten Abstand von 12 cm zur Tischoberfläche hat.

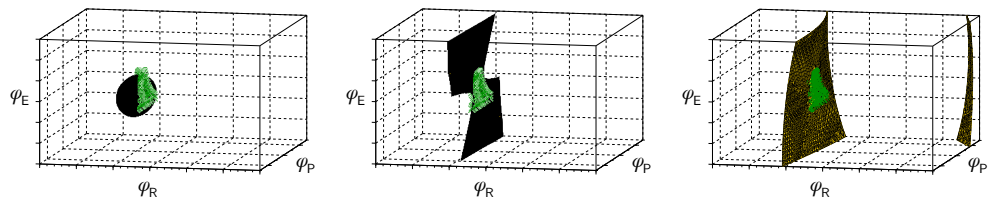


(d) Versuchsaufbau



(f) Fehler über Iterationen: (oben) GA, (unten) RLS, (rote Linie) LS

Abbildung 62: Quadrik des gesamten Aktionsraumes mit der Bedingung, dass die Hand auf Schulterhöhe und parallel zur Tischoberfläche ist.



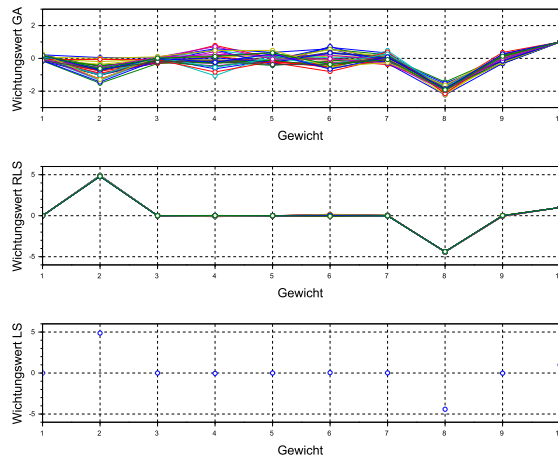
(a) Quadrik LS

(b) Quadrik RLS

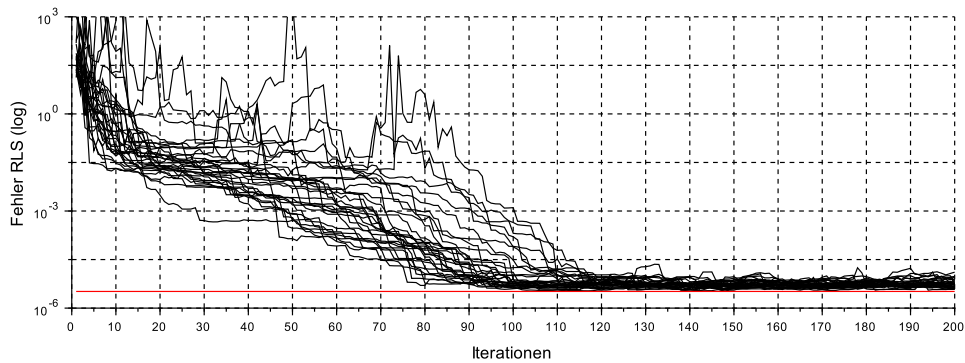
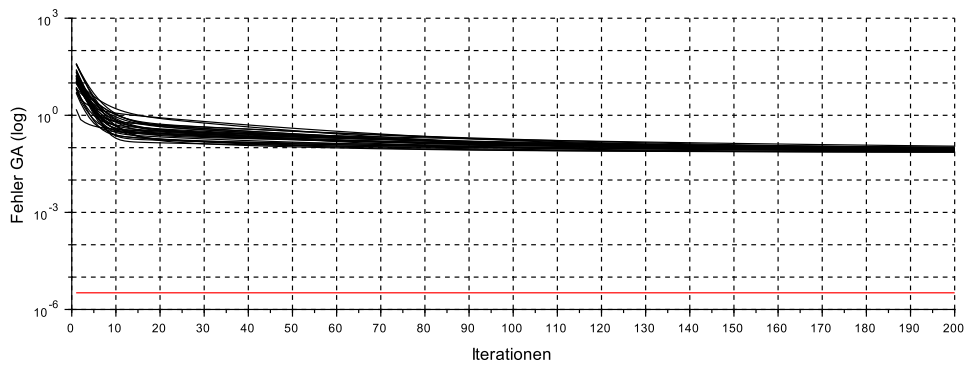
(c) Quadrik GA



(d) Versuchsaufbau

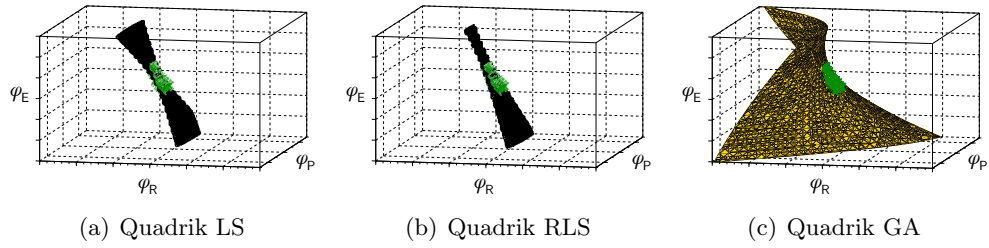


(e) Gewichtsvektoren

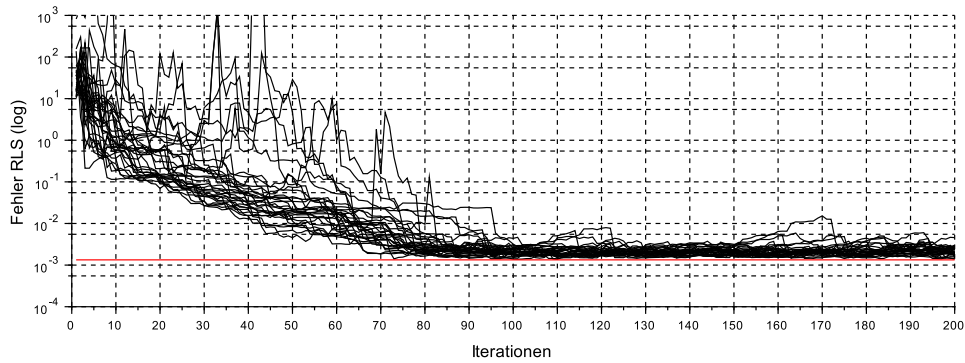
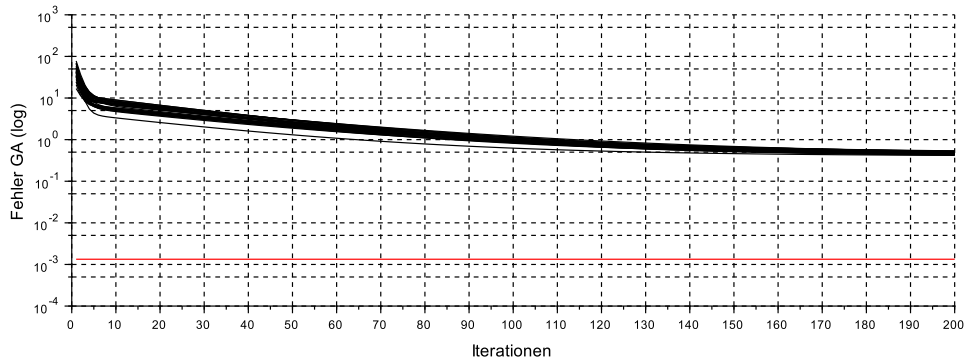
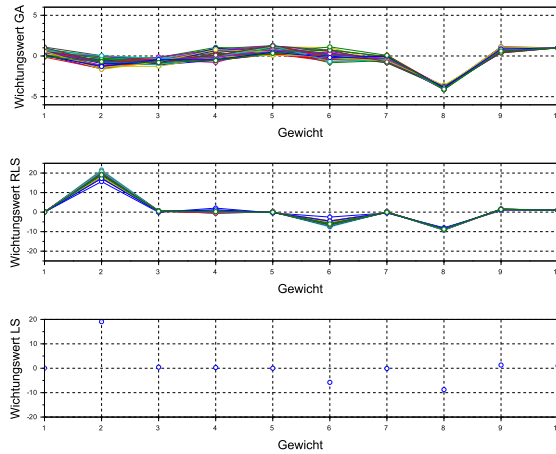


(f) Fehler über Iterationen: (oben) GA, (unten) RLS, (rote Linie) LS

Abbildung 63: Quadrik des gesamten Aktionsraumes mit der Bedingung, dass die Hand die graue vertikale Wand zur Körpermitte berührt.

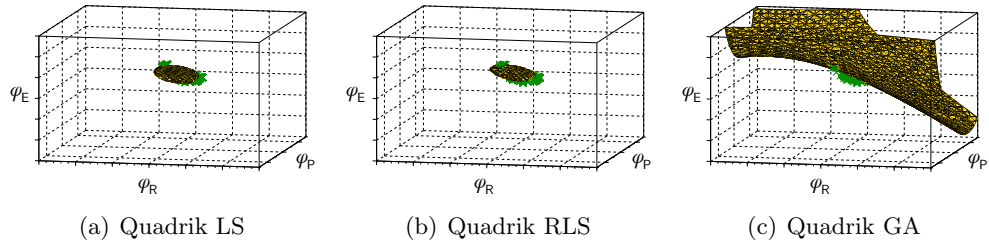


(d) Versuchsaufbau

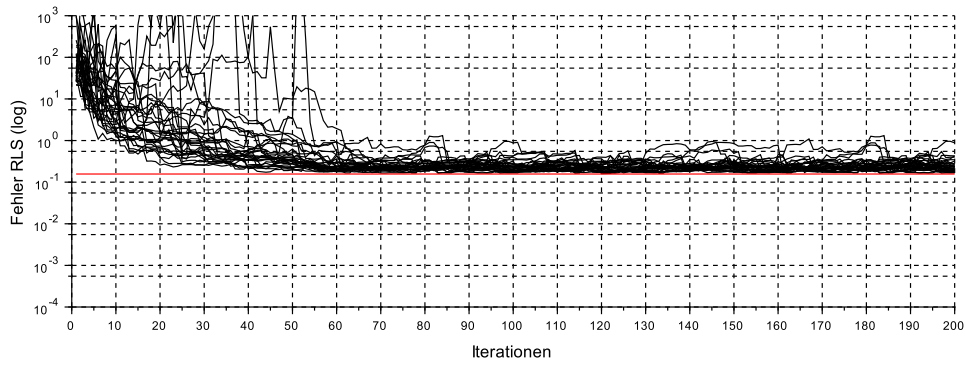
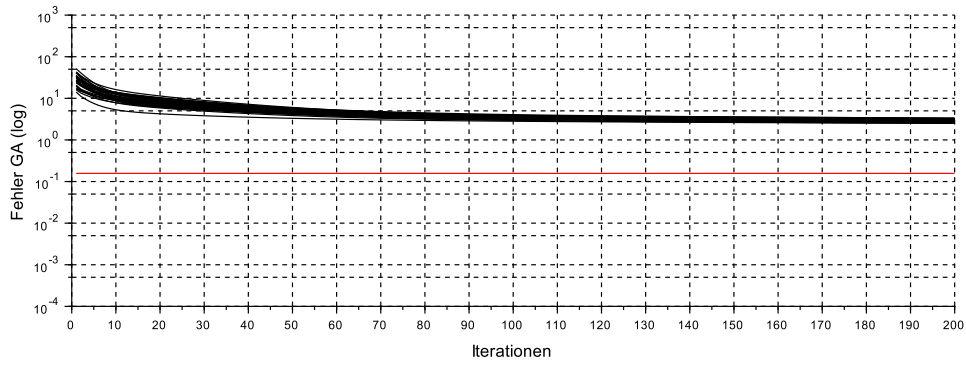
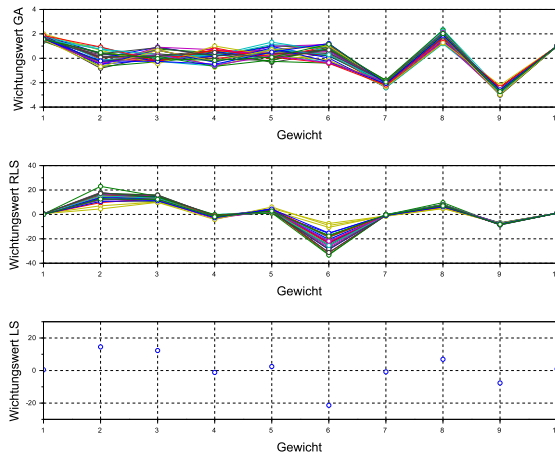


(f) Fehler über Iterationen: (oben) GA, (unten) RLS, (rote Linie) LS

Abbildung 64: Quadrik des gesamten Aktionsraumes mit der Bedingung, dass die Hand die graue vertikale Wand mittig im Aktionsraum berührt.



(d) Versuchsaufbau



(f) Fehler über Iterationen: (oben) GA, (unten) RLS, (rote Linie) LS

Abbildung 65: Quadrik des gesamten Aktuationsraumes mit der Bedingung, dass die Hand die graue vertikale Wand außen im Aktuationsraum berührt.

Obwohl der resultierende Fehler des GA deutlich größer ist als beim RLS, zeigt die Darstellung der Quadriken (Abbildungen 59(c)–65(c)), dass sich die Resultate besser für die Anwendung in der Ansteuerung der Robotergelenke eignen. Quadriken der Normalform eines Ellipsoiden, die wie hier optimal in den Messpunkten liegen (Abbildungen 59(a)–65(a)), haben folgendes Problem: Wird die Hand im Konfigurationsraum auf der Oberfläche der Quadrik geregelt und kommt in die Randbereiche der zugrundeliegenden Messpunkt wolke, liegen diese Bereiche in den scharfen Rundungen der Quadrik. Bereits leichte Bewegungen über den Bereich der Lernphase hinaus oder geringfügige Sensorfehler führen dazu, dass die Bewegung auf der „Rückseite“ des Ellipsoiden fortgesetzt wird. Dieser Bereich führt zu einer Bewegung außerhalb des gültigen Arbeitsbereichs und kann zu Schäden am Roboter oder der Umwelt führen. Quadriken der Normalform eines Hyperboloiden oder eines Paraboloiden (wie mittels GA generiert) bieten in der Umgebung der Messpunkte eine gleichmäßige Extrapolation der Bewegung im Konfigurationsraum und eignen sich daher deutlich besser für diese Anwendung. Tabelle 5 bietet eine Übersicht über die Normalformen der generierten Quadriken.

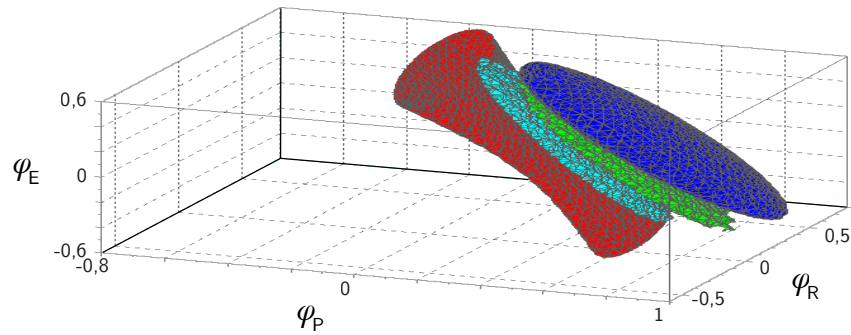
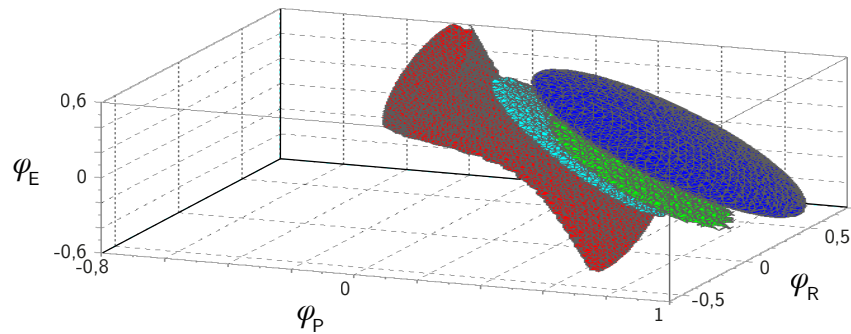
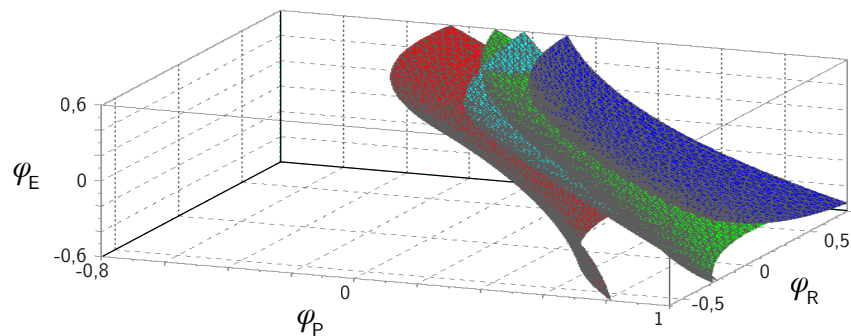
Abbildung 66 auf Seite 120 zeigt die vier Quadriken, die im Aktuationsraum parallel zur Tischoberfläche liegen, zugehörig zu den jeweiligen Optimierungsverfahren dargestellt im Konfigurationsraum. Sie basieren auf den Messpunkten in Abbildung 58(a). Die Grenzen der Darstellung entsprechen den mechanischen Anschlüssen des rechten Roboterarms. Daher entspricht der dargestellte quaderförmige Bereich dem vollständigen Konfigurationsraum. An den resultierenden Quadriken des LS und RLS, die der Normalform eines Ellipsoiden, bzw. eines einschaligen Hyperboloiden entsprechen und zudem in ihren Dimensionen recht kompakt sind, wird folgendes deutlich: Abhängig von der Richtung, aus der man sich der Quadrik im Konfigurationsraum nähert, sind unterschiedliche Abschnitte der Schalen am nächsten, die alleine durch den Funktionswert des Polynoms Q nicht zu unterscheiden sind. Beim Ellipsoid beispielsweise wird der Raum in zwei Mannigfaltigkeiten unterteilt, je eine innerhalb und eine außerhalb des Körpers, welche durch das Vorzeichen des Polynoms Q leicht zu unterscheiden sind. Wird die Hand im Konfigurationsraum über ein beliebiges Verfahren der Schale der Quadrik angenähert, ist hieraus nicht zu erkennen, ob es sich um den zuvor an die Messpunkte optimierten Abschnitt der Schale handelt oder nicht. Da der GA zu einem sehr schwach gekrümmten einschaligen Hyperboloiden oder einem zweischaligen Hyperboloiden mit nur einer Schale tendiert, liegt innerhalb des Konfigurationsraums je Quadrik nur eine Schale, die den Raum in zwei eindeutig unterscheidbare Mannigfaltigkeiten unterteilt. Eine Bewegung zur Schale hin führt immer zur angestrebten Ebene im Aktuationsraum,

Tabelle 5: Übersicht der Normalformen der Quadriken, abhängig vom Optimierungsverfahren

Anwendung	Abbildung (Seitenzahl)	Normalform		
		LS	RLS	GA
Hand auf der Tischoberfläche	59 (111)	Ellipsoid	Ellipsoid	Zweischaliges Hyperboloid
Zwischen Hand und Tischoberfläche 6 cm Abstand	60 (112)	Ellipsoid	Ellipsoid	Einschaliges Hyperboloid
Zwischen Hand und Tischoberfläche 12 cm Abstand	61 (113)	Ellipsoid	Ellipsoid	Einschaliges Hyperboloid
Hand auf Schulterhöhe	62 (114)	Einschaliges Hyperboloid	Einschaliges Hyperboloid	Einschaliges Hyperboloid
Wand innen	63 (115)	Elliptischer Zylinder	Einschaliges Hyperboloid	Einschaliges Hyperboloid
Wand mittig	64 (116)	Einschaliges Hyperboloid	Einschaliges Hyperboloid	Einschaliges Hyperboloid
Wand außen	65 (117)	Ellipsoid	Ellipsoid	Einschaliges Hyperboloid

dem zuvor definierten Arbeitsbereich.

Abbildung 67 auf Seite 121 zeigt Vergleichsgrafiken der Gewichtsvektoren für horizontale Bewegungen (Flächen parallel zur Transversalebene des Roboters) und vertikale Bewegungen (Flächen parallel zur Sagittalebene des Roboters). Erwartungsgemäß liegt die Variation in den Parametern bei den horizontalen Flächen hauptsächlich im Schulter-*Pitch*-Gelenk und leicht im Ellenbogen. Zwischen den Ebenen mit vertikaler Bewegung variieren die Parameter primär im Schulter-*Roll*-Gelenk sowie im Ellenbogen. Das Schulter-*Pitch*-Gelenk hat kaum Einfluss auf die Definition der Quadrik, da es das „variable“ Gelenk für die darauf ausgeführte Bewegung ist.

(a) Erzeugt mittels *Least-Squares*-Verfahren (LS)(b) Erzeugt mittels *Recursive-Least-Squares*-Verfahren (RLS)

(c) Erzeugt mittels Gradientenabstiegs-Verfahren (GA)

Abbildung 66: Grafische Darstellung von vier Quadriken, die im Aktuationsraum parallel zur Tischoberfläche liegen, dargestellt im Konfigurationsraum zugehörig zu den jeweiligen Optimierungsverfahren. Sie basieren auf den Messpunkten in Abbildung 58(a). Die Grenzen der Darstellung entsprechen den mechanischen Anschlägen des rechten Roboterarms – dem Konfigurationsraum. (blau) Hand auf der Tischoberfläche, (grün) Hand in 6 cm Höhe, (türkis) Hand in 12 cm Höhe, (rot) Hand auf Schulterhöhe.

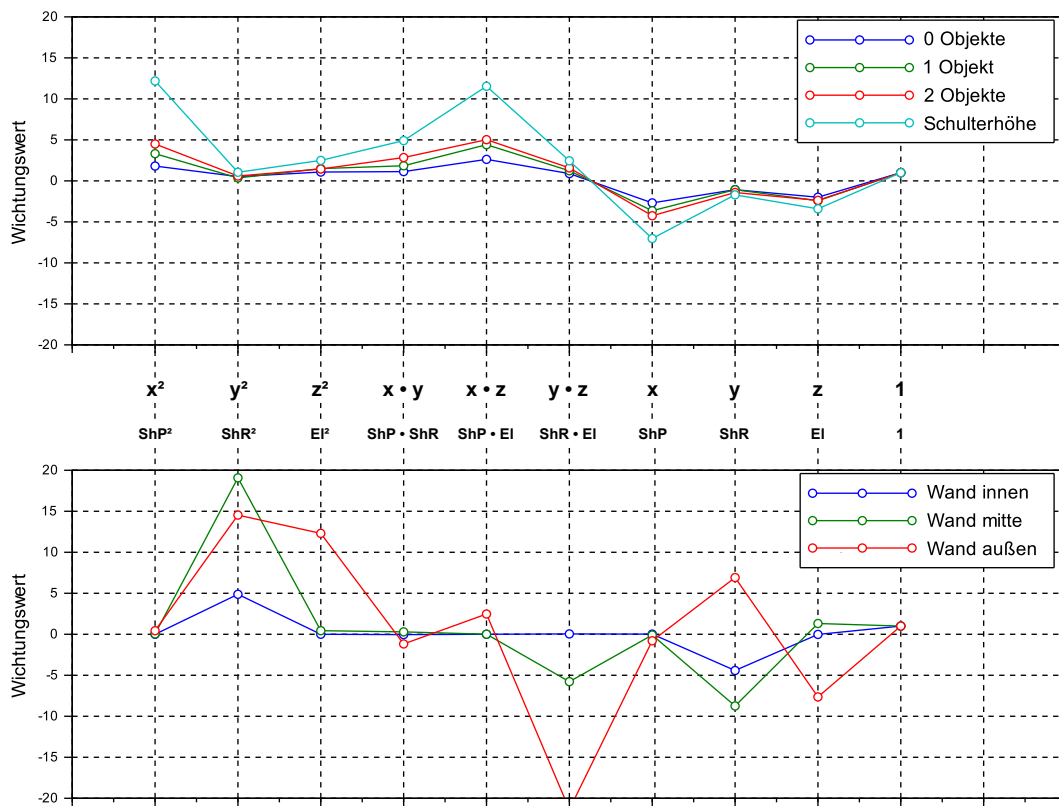


Abbildung 67: Repräsentative Gewichte für horizontale (oben) und vertikale Bewegungen (unten), generiert mit dem LS-Verfahren.

Tabelle 6 zeigt die Ausführungszeiten des Gradientenabstiegs auf der Roboterplattform. Dabei wurden exemplarisch 10, 30 und 100 Iterationen direkt nacheinander berechnet, um den Einfluss der Speicherzugriffe sowie der Pufferspeicher (*Caches*) zu berücksichtigen. Im Vergleich dazu dauert eine Inversion einer 9×9 -Matrix mittels QR-Zerlegung auf der Zielhardware $\approx 86,232$ ms. Dies entspricht ca. 1033 Durchgängen des Gradientenabstiegs mit jeweils 30 Iterationen beziehungsweise ca. 353 Durchgängen des Gradientenabstiegs mit jeweils 100 Iterationen. Dies ermöglicht die Berechnung von ungefähr 300-mal so vielen Quadriken mit einer bis zu 10-fach besseren Abdeckung des Arbeitsbereichs bei gleicher Rechenzeit.

Tabelle 6: Ausführungszeiten des Gradientenabstiegs auf der Roboterplattform mit unterschiedlicher Anzahl Iterationen

	10 Iterationen	30 Iterationen	100 Iterationen
Eingangsvektor zusammenstellen	7,55 μ s	19,61 μ s	58,96 μ s
Gradientenabstieg	17,55 μ s	35,40 μ s	98,95 μ s
Fehlerberechnung	12,36 μ s	28,46 μ s	86,44 μ s
Summe	37,46 μs	83,47 μs	244,35 μs
Mögliche Ausführungen pro Slot (10 ms)	≈ 267	≈ 120	≈ 41

Da der Gradientenabstieg effizient auf der Roboterhardware zu berechnen ist, die resultierenden Quadriken sehr gut für die Abschätzung der Zwangsbedingung zwischen der Roboterhand und der Umwelt geeignet sind, und die Extrapolationseigenschaften der resultierenden Quadriken im Bezug zu dieser Anwendung mit Abstand am besten sind, wird dieses Verfahren im Folgenden eingesetzt.

7.2 Bewegungen auf Quadriken

In diesem Kapitel wird die Berechnung der Armbewegung auf einer Quadrik im deliberativen und reaktiven Teil des Roboters betrachtet. Der deliberative Teil ist sehr flexibel, muss seine Ressourcen jedoch mit vielen anderen Berechnungen teilen. Diese werden zentral auf dem *BrainModule* (siehe Kapitel 3.3) im Kopf des Roboters durchgeführt. Die reaktiven Netze, welche im Aufbau an künstliche neuronale Netze erinnern, sind in ihrer Struktur statisch, beeinflussen jedoch bis zur Nutzung der maximalen Neuronenzahl nicht die Ausführungszeiten anderer Berechnungen. Diese werden dezentral, verteilt auf die AB3Ds (siehe Kapitel 3.1), berechnet.

Abbildung 68 zeigt die vier prototypischen Quadriken, die parallel zum Tisch mit unterschiedlichen Abständen zur Tischoberfläche orientiert sind. Die Quadriken wurden an den mechanischen Anschlüssen der Gelenke beschnitten und mit den zugrundeliegenden Messpunkten als grüne Kreuze überlagert. Die Transformation der Quadriken in den Aktuationsraum erfolgte anhand der Vorwärtskinematik nach Kapitel 5.2. Die Betrachtung der Quadriken im Aktuationsraum (Abbildung 68b,d) zeigt, dass die Punkte auf den Quadriken außerhalb der zugrunde liegenden Messpunkte zwar grundsätzlich mechanisch zulässige Werte einnehmen, diese jedoch nicht mehr parallel zur Tischoberfläche sind und daher auch nicht angesteuert werden dürfen.

Deliberative Ausführung im Roboterkopf

Bei der deliberativen Ausführung wird im Konfigurationsraum mittels Gradientenabstieg der nächst-kürzeste Weg der Hand auf die Quadrik berechnet. Die grundlegenden Formeln entsprechen denen in Kapitel 7.1, jedoch wird hier nicht reguliert; die partiellen Ableitungen sind nicht mehr nach den Gewichten zu bilden, sondern nach den zu ermittelnden Gelenkwinkeln φ .

Die resultierenden Formeln lauten:

$$\frac{\partial E(\mathbf{w}, \varphi)}{\partial \varphi_P} = 2 (\mathbf{w}^T \mathbf{f}_3(\varphi)) (2w_1\varphi_P + w_4\varphi_R + w_5\varphi_E + w_7) \quad (27)$$

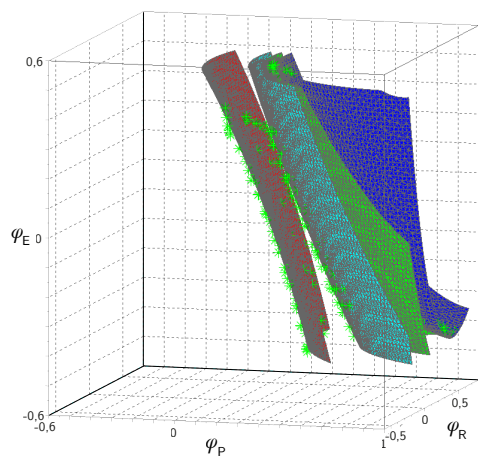
$$\frac{\partial E(\mathbf{w}, \varphi)}{\partial \varphi_R} = 2 (\mathbf{w}^T \mathbf{f}_3(\varphi)) (2w_2\varphi_R + w_4\varphi_P + w_6\varphi_E + w_8) \quad (28)$$

$$\frac{\partial E(\mathbf{w}, \varphi)}{\partial \varphi_E} = 2 (\mathbf{w}^T \mathbf{f}_3(\varphi)) (2w_3\varphi_E + w_5\varphi_P + w_6\varphi_R + w_9) \quad (29)$$

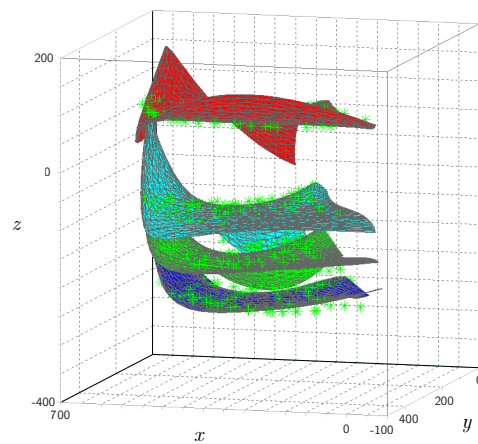
Die Updatefunktion für das jeweilige Gelenk $\varphi_i \in \{\varphi_P, \varphi_R, \varphi_E\}$ lautet

$$\varphi_{i,t+1} = \varphi_{i,t} - \varepsilon_a \frac{\partial E(\mathbf{w}, \boldsymbol{\varphi})}{\partial \varphi_i} \quad (30)$$

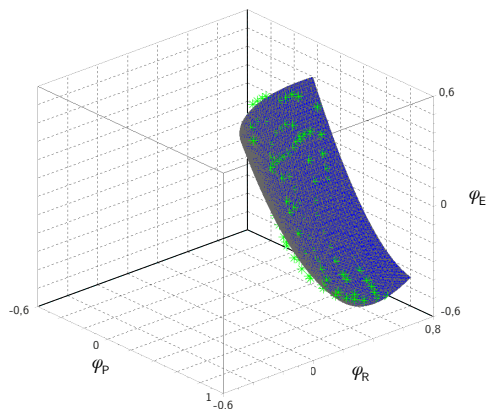
mit der Schrittweite $\varepsilon_a \in \{0 \dots 1\}$.



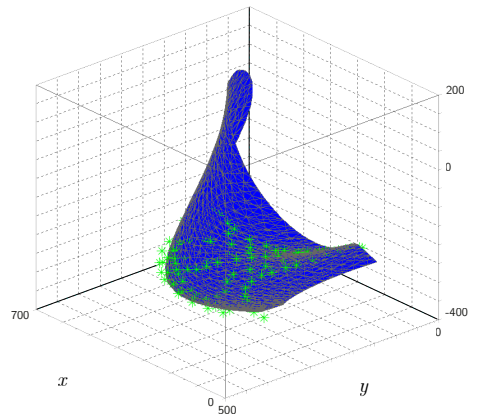
(a) Vier Quadriken im Konfigurationsraum. Die Gelenkwinkel sind einheitenlos.



(b) Vier Quadriken im Aktuationsraum. Die Achsen sind in mm skaliert.



(c) Quadrik der Tischoberfläche im Konfigurationsraum. Die Gelenkwinkel sind einheitenlos.



(d) Quadrik der Tischoberfläche im Aktuationsraum. Die Achsen sind in mm skaliert.

Abbildung 68: Tischparallele Quadriken in unterschiedlichen Abständen zur Tischoberfläche. Die Darstellung im Konfigurationsraum erfolgt anhand der Messdaten und den daraus generierten Quadriken. Der Wertebereich ist durch die mechanischen Anschläge der einzelnen Gelenke begrenzt. Die Darstellung im Aktuationsraum erfolgt durch eine Vorwärtskinematik. Die grünen Kreuze zeigen die zugrunde liegenden Messpunkte. (blau) Hand auf der Tischoberfläche, (grün) Hand in 6 cm Höhe, (türkis) Hand in 12 cm Höhe, (rot) Hand auf Schulterhöhe

Ausgehend von hier kann unter zwei verschiedenen Betriebsmodi unterschieden werden:

Kooperation

Werden alle drei Gelenke φ auf der Quadrik fixiert, ist es möglich, dass der Mensch Myons Hand problemlos auf der Quadrik bewegt und dessen Verlauf erkundet bzw. erfühlt und somit Punkte auf der Quadrik aktiv demonstriert.

Ansteuerung

Wird zum Beispiel nur die Bewegung des Ellenbogengelenks mittels Gradientenabstieg berechnet, so ist die Ansteuerung der Hand auf der Quadrik über die Festlegung der beiden Schultergelenke möglich. Voraussetzung ist, dass es für die vorgegebenen Schulterwinkel einen gültigen Wert für den Ellenbogen gibt und der Fehler E möglichst klein ist.

Abbildung 69 zeigt den Pfad der Hand (blau), wenn alle drei Gelenke mittels Gradientenabstieg auf die Quadrik zubewegt werden. Der Startpunkt lag bei $P_0 = (-1, -0,5, -0,5)$.

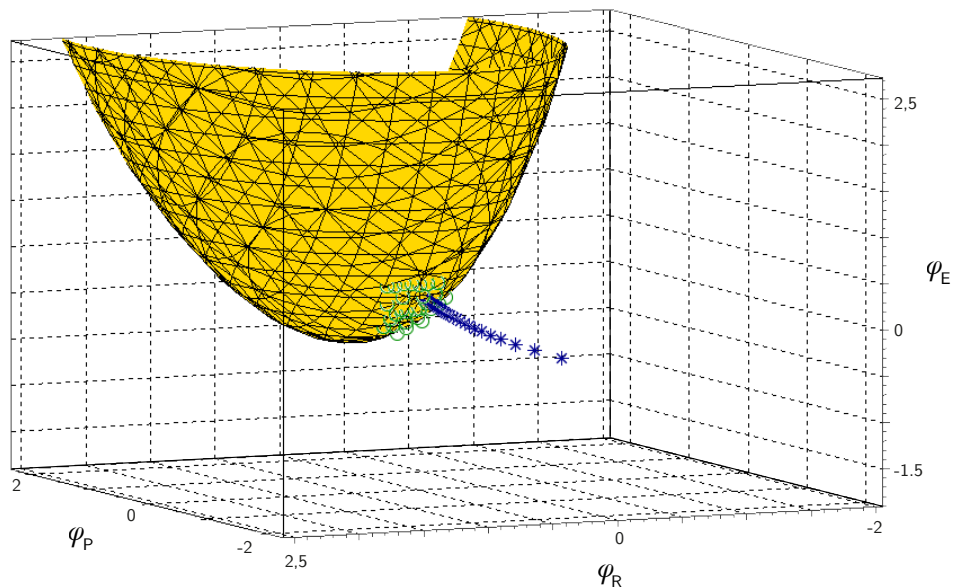


Abbildung 69: Diese Abbildung zeigt den Pfad der Hand im Konfigurationsraum des rechten Arms. Dieser beginnt entfernt von der Quadrik im Punkt $P_0 = (-1, -0,5, -0,5)$. Die Rückführung zur Quadrik erfolgt mittels GA auf allen drei Gelenken: (blau) Pfad der Hand; (gelb) Quadrik; (grün) der Quadrik zugrunde liegende Messpunkte. Die Gelenkwinkel sind einheitenlos.

Dezentrale reaktive Ausführung

Bei der reaktiven Ausführung kommt eine ähnliche Implementation wie die von Hild eingeführten *Quadic-Representing Neurons (QRENS)*[HKH11] zur Anwendung. Abbildung 70 zeigt den Aufbau des reaktiven Netzes zur Berechnung einer Quadrik mit der darauf folgenden Positionsregelung in Form einer kognitiven sensomotorischen Schleife (*Cognitive Sensorimotor Loop/CSL*) im HOLD-Modus [KBH11]. Diese hält den Ellenbogenwinkel möglichst auf der Quadrik.

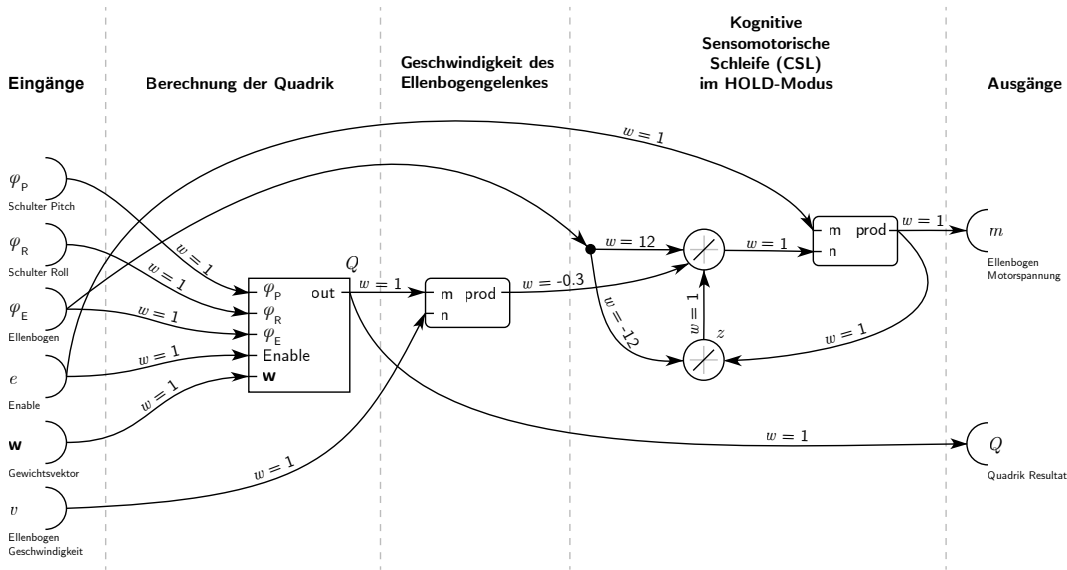


Abbildung 70: Grafische Repräsentation der reaktiven Struktur zum Auslesen der propriozeptiven Sensorwerte sowie der Motoransteuerung des Ellenbogens auf Basis des Polynoms Q einer quadratischen Funktion.

Formal ist dieser Prozess wie folgend beschrieben:

$$\begin{aligned}\varphi_t &= (\varphi_P \ \varphi_R \ \varphi_E)_t^T \\ \mathbf{f}(\varphi_t) &= (\varphi_P^2 \ \varphi_R^2 \ \varphi_E^2 \ \varphi_P\varphi_R \ \varphi_R\varphi_E \ \varphi_P\varphi_E \ \varphi_P \ \varphi_R \ \varphi_E \ 1)_t^T \\ \mathbf{w}_t &= (w_1, w_2, \dots, w_{10})_t^T \\ Q_t &= (e(\mathbf{w}^T \mathbf{f}(\varphi)))_{t-2}\end{aligned}$$

$$m_t = e_{t-1} (12\varphi_{E,t-2} + z_{t-1} - 0,3v_{t-3}Q_{t-3})$$

$$z_t = m_{t-1} - 12\varphi_{E,t-1}$$

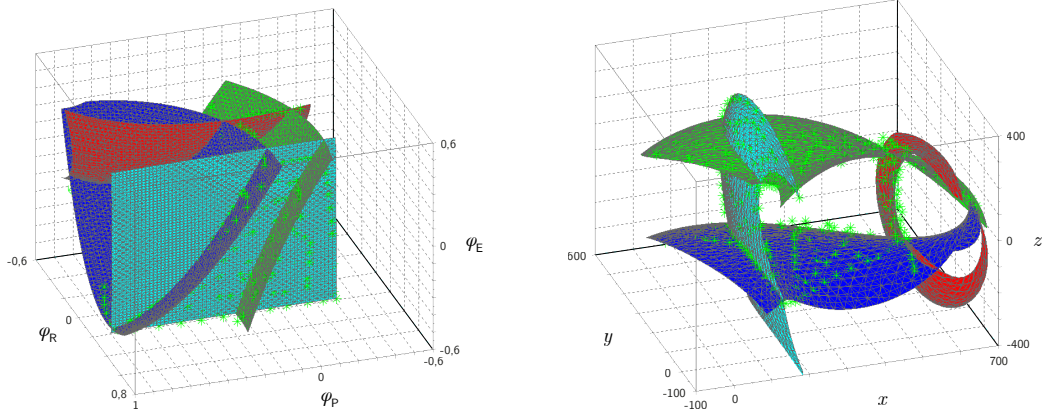
Die Quadrik wird als implizite Funktion aus den drei Armgelenken berechnet, und der Ellenbogen basierend auf dem vorzeichenbehafteten Ergebnis des Polynoms Q angesteuert. Dieses Ergebnis wird unabhängig von der Ansteuerung eines Gelenks mit dem Wert m an das deliberative System übermittelt, um jederzeit die Abweichung der aktuellen Pose im Konfigurationsraum von der Quadrik ermitteln zu können. Das Besondere dieser Art der Ansteuerung ist, dass die deliberative Einheit im Kopf des Roboters keinerlei Rechenaufwand durch die Regelung des Ellenbogens hat.³⁷ Sie kann jedoch trotzdem die Parameter der Quadrik (e, v, \mathbf{w}) während des Betriebs beliebig variieren. Die nachgeschaltete Gelenkregelung basiert auf dem in Kapitel 5.1 vorgestellten Verfahren.

Kombination von Quadriken

Quadriken können zu verschiedenen Zwecken kombiniert werden: So kann beispielsweise ein virtueller Raum als Teilmenge des Konfigurations-/Aktuationsraums aus mehreren Quadriken eingeschlossen werden, indem man dem Roboter durch wenige Stützpunkte die Grenzen zeigt, und diese dann durch mehrere Quadriken verbunden werden. Wird eine Quadrik erreicht, geht die Lösung der Funktion Q gegen null und die Bewegung kann auf reaktiver Ebene inhibiert werden. Dabei ist die Anzahl an Quadriken flexibel. Beispiele sind eine kubische Begrenzung mit sechs Quadriken, eine ebene Begrenzung mit einer Quadrik oder die Nutzung der inneren Mannigfaltigkeit eines Ellipsoiden mit ebenfalls nur einer Quadrik. Abbildung 71 zeigt eine Begrenzung durch vier Quadriken im Konfigurations- sowie im Aktuationsraum. Diese ermöglichen die Ausführung der Bewegung auf einer der vier zuvor gezeigten horizontalen Quadriken beschränkt auf die gültigen Bereiche, zu denen auch Messpunkte vorliegen.

Zur effizienten Umsetzung sind in der reaktiven Struktur sechs Quadriken fest implementiert, wobei durch die deliberative Einheit jede separat parametrisiert und aktiviert werden kann. Die Bewegung wird bei Näherung an eine oder mehrere Quadriken inhibiert und nur dann wieder ermöglicht, wenn eine Bewegung von den Begrenzungen weg führt. Den Aufbau einer einzelnen Begrenzungsquadrik zeigt Abbildung 72 auf Seite 128. Der als Gradient g bezeichnete Eingang des Netzes dient der Auswahl der Mannigfaltigkeit. Das Vorzeichen gibt vor, welches Vorzeichen das charakteristische Polynom im gültigen Bereich des Konfigurationsraums einnimmt.

³⁷Das reaktive System stellt je Gelenk Ressourcen für ungefähr 20 Quadriken zur Verfügung, ohne zusätzliche zeitliche Ressourcen zu verbrauchen.



(a) Vier Quadriken im Konfigurationsraum mit den zugrunde liegenden Messpunkten. Die Gelenkwinkel sind einheitenlos.

(b) Vier Quadriken im Aktuationsraum mit den zugrunde liegenden Messpunkten. Die Achsen sind in mm skaliert.

Abbildung 71: Vier Quadriken zur Begrenzung des Aktuationsraums im Konfigurationsraum sowie die Darstellung derselben Quadriken im Aktuationsraum mittels Berechnung der Vorwärtskinematik. Die grünen Kreuze zeigen die zugrunde liegenden Messpunkte. (blau) Hand auf der Tischoberfläche, (grün) Hand oberhalb der Schulter und parallel zur Tischoberfläche, (türkis) Vertikale Begrenzung nach außen, (rot) Vertikale Begrenzung nach innen/zur Körpermitte.

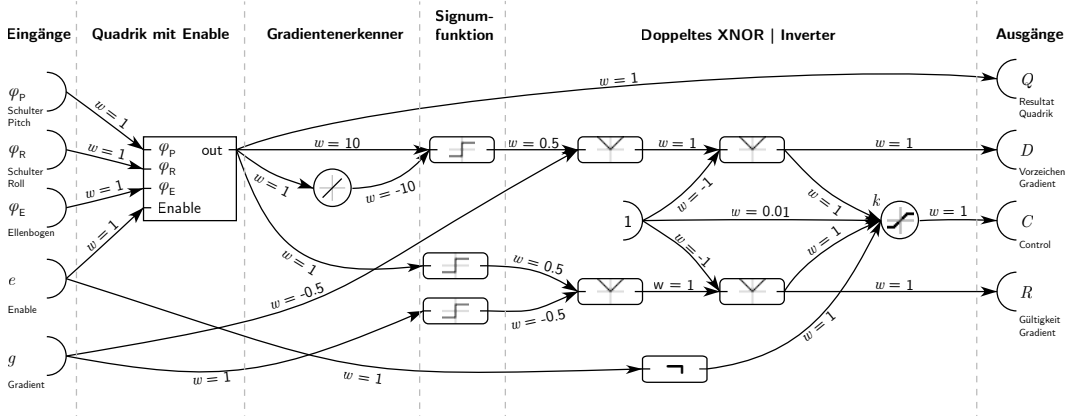


Abbildung 72: Reaktives Netz für die Erkennung einer Begrenzung als Teilmenge des Konfigurationsraums durch Quadriken. Die vom deliberativen System übertragenen Gewichte w sind nicht dargestellt.

Formal ist eine einzelne Begrenzungsquadrk wie folgend beschrieben:

$$\begin{aligned}
\boldsymbol{\varphi}_t &= (\varphi_P \ \varphi_R \ \varphi_E)_t^T \\
\mathbf{f}(\boldsymbol{\varphi}_t) &= (\varphi_P^2 \ \varphi_R^2 \ \varphi_E^2 \ \varphi_P \varphi_R \ \varphi_R \varphi_E \ \varphi_P \varphi_E \ \varphi_P \ \varphi_R \ \varphi_E \ 1)_t^T \\
\mathbf{w}_t &= (w_1, w_2, \dots, w_{10})_t^T \\
Q_t &= (e (\mathbf{w}^T \mathbf{f}(\boldsymbol{\varphi})))_{t-2} \\
C_t &= \begin{cases} 1 & \text{für } k_{t-1} \geq 1 \\ k_{t-1} & \text{für } -1 < k_{t-1} < 1 \\ -1 & \text{für } k_{t-1} \leq -1 \end{cases} \\
k_t &= 0,01 + D_t + R_t + (1 - e_{t-1}) \\
D_t &= | |0,5 (\operatorname{sgn}(Q_{t-3} - Q_{t-4}) - g_{t-2})| - 1| \\
R_t &= | |0,5 (\operatorname{sgn}(Q_{t-3}) - \operatorname{sgn}(g_{t-3}))| - 1|
\end{aligned}$$

mit $e \in \{0, 1\}$.

Die sechs Begrenzungsquadriken werden durch multiplikative Verkettung miteinander verknüpft. Die Ansteuerung der drei Armgelenke mit $\boldsymbol{\varphi}^*$ wird bei Bedarf durch das Produkt inhibiert:

$$\begin{aligned}
\boldsymbol{\varphi}_t^* &= d_{t-1} \boldsymbol{\varphi}_{t-1} \\
d_t &= \begin{cases} 0 & \text{für } c \leq 0 \\ c_{t-1} & \text{für } 0 < c < 1 \\ 1 & \text{für } c \geq 1 \end{cases} \\
c_t &= C_{1,t-1} C_{2,t-2} C_{3,t-3} C_{4,t-4} C_{5,t-5} C_{6,t-5}
\end{aligned}$$

Abbildung 73 zeigt das zugehörige reaktive Netz.

Das Umschalten zwischen zwei Quadriken ist ebenfalls möglich. Dieses Verfahren kann beispielsweise eingesetzt werden, um die Hand von der Tischoberfläche in eine höhere Ebene wechseln zu lassen. Dafür können zwei Quadriken im reaktiven Netz implementiert werden, zwischen denen über das *Enable-Flag* e analog überblendet

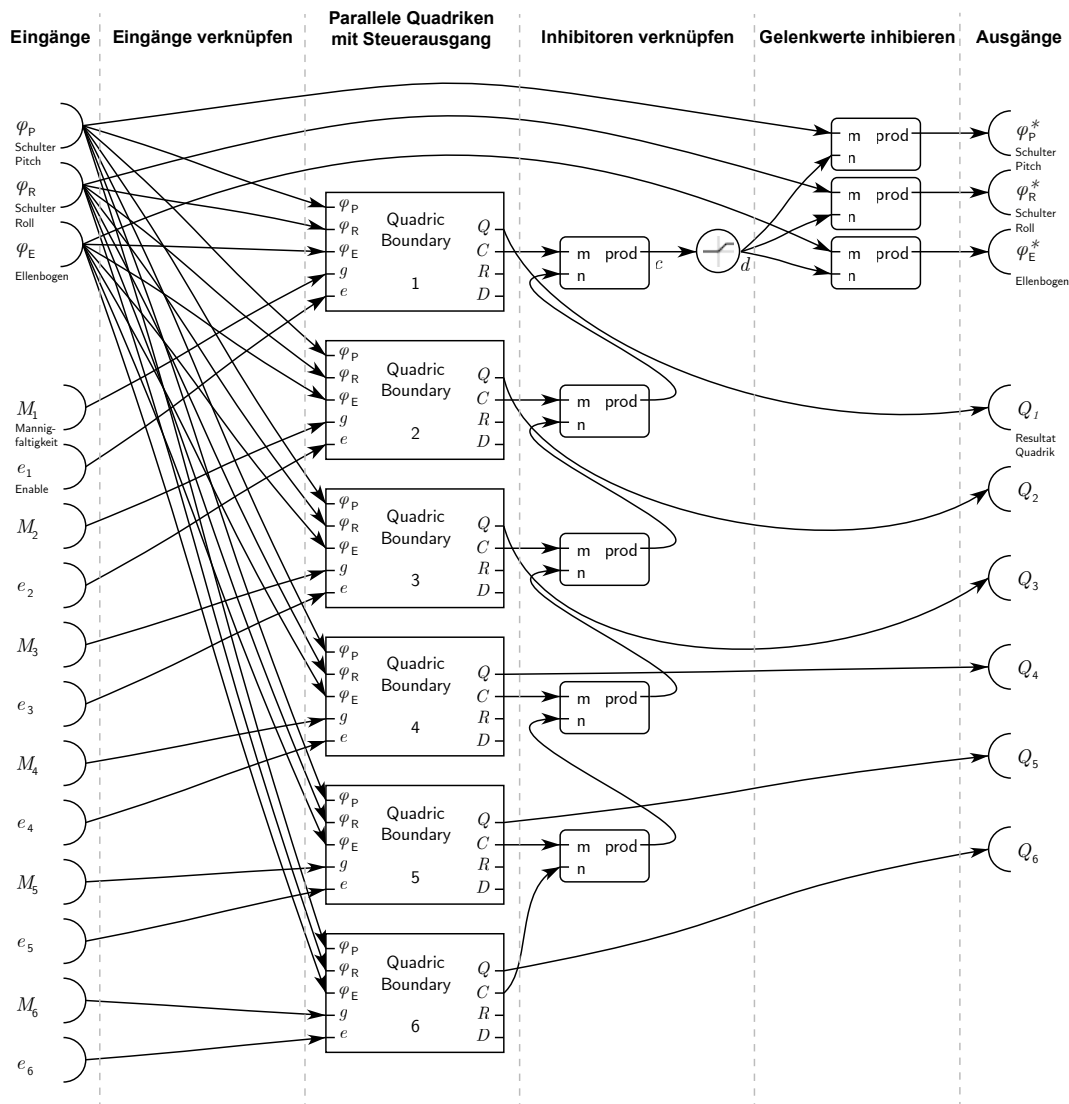


Abbildung 73: Reaktives Netz mit sechs verknüpften Begrenzern von Teilmengen des Konfigurationsraums durch Quadriken. Die vom deliberativen System übertragenen Gewichte sind im Interesse einer besseren Übersichtlichkeit nicht dargestellt. Alle Synapsen haben das Gewicht $w = 1$.

wird. Alternativ ist auch eine Quadrik ausreichend, indem diese per *Enable-Flag* e während der Parameteränderung \mathbf{w} inhibiert wird. Entscheidend ist, dass das Vorzeichen des Ergebnisses der Funktion Q für die gewünschte Mannigfaltigkeit bekannt ist. Dieses ist in diesem Fall in Form des Vorzeichens des Parameters v (siehe Abbildung 70) zu berücksichtigen, da sich ansonsten der Punkt im Konfigurationsraum durch die folgende Regelschleife (*CSL* im *Hold-Modus*) von der Zielquadrik entfernt anstatt sich ihr zu nähern. Insbesondere bei sich schneidenden Quadriken ist dies zu berücksichtigen.

7.3 Nutzen der Quadriken für die Hand-Auge-Koordination

Im Vergleich zu den im vorigen Kapitel 6 gezeigten Verfahren liegt beim Einsatz von Quadriken zusätzliches Wissen über den Zusammenhang zwischen Roboter und Umwelt vor. Die Quadriken können die zuvor vorausgesetzten *Environmental Constraints* ersetzen, sodass bei bereits explorierter Umwelt kein Kontakt zu dieser mehr notwendig ist – beim Beispiel des Greifens zwischen Hand und Tischoberfläche: Ist die Tischquadrik bekannt, wird der reaktiven Einheit des Roboters die Regelung des Ellenbogengelenks übertragen und die Hand kann – ohne Rechenaufwand für das visuomotorische System in der deliberativen Einheit – oberhalb der Tischoberfläche gehalten werden. Dies gilt analog auch für das Nachzeichnen von Linien.

Objekt greifen

Neben dem Vorteil, dass für das Greifen nur noch das Greifobjekt selbst berührt werden muss, was zu reduzierter Reibung und vermindertem Verschleiß führt, erweitern sich die Einsatzbereiche dahingehend, dass nun auch gegriffen werden kann, wenn andere Objekte im Weg stehen. Erfolgte zuvor die Annäherung an das Greifobjekt noch ausschließlich von der Seite, ist dies mit der gewonnenen Abstandsinformation von Hand zu Tischoberfläche sowie der in Kapitel 2.1 vorgestellten Blickheuristik nun auch von oben möglich. In Abbildung 74 ist zu erkennen, wie sich Hand und Greifobjekt auf der Blickachse optisch überlagern.

Ist die Quadrik der Tischoberfläche bekannt, kann das deliberative System die Steuerung des Ellenbogens übernehmen. Der vom reaktiven System berechnete Funktionswert der Funktion Q dient dabei als Abstandsmaß. Am Vorzeichen ist ersichtlich, ob sich die Hand in der „oberen“ oder der „unteren“ Mannigfaltigkeit im Konfigurationsraum befindet. Der Übergang zwischen den beiden Mannigfaltigkeiten erfolgt an der Tisch-Quadrik. Für den Aktuationsraum bedeutet dies, dass das Vorzeichen angibt, ob sich die Hand ober- oder unterhalb der Tischoberfläche befindet. Wird die Blickheuristik fortdauernd angewandt, kommt es bei der kontinuierlichen Streckung des Ellenbogens unweigerlich zur Kollision von Hand und Objekt. Wird die Blickheuristik aber so angepasst, dass das Objekt mittig auf der Blickachse liegt und die Hand seitlich daran anschließt, bleibt die Hand an der idealen Greifposition stehen, und zwar ganz ohne explizite Pfadplanung.

Auf Grund der Flexibilität der Finger wäre ohne dieses implizite Maß der Kontakt der Hand mit dem Tisch erst durch den Kontakt des Handballens über die mechanische Kopplung der Gelenke detektierbar, was das Greifen von oben verhindert.



Abbildung 74: Greifen eines Objekts von oben mittels Blickheuristik und Abstandsinformation durch den Funktionswert der Funktion Q der Tisch-Quadrik.

Objekte stapeln

Die beiden bereits vorgestellten Verfahren zum Stapeln von Objekten (Kapitel 6.3 auf Seite 88) basieren auf der Hinderniserkennung über den I-Anteil des Reglers im Ellenbogen-Gelenk.

Dabei wurde das Basisobjekt gegriffen und durch Beugen des Ellenbogens angehoben. Anschließend wurde der gebeugte Ellenbogen solange gestreckt, bis das Stapelobjekt auf dem Basisobjekt steht. Dieser Zeitpunkt wurde über einen Lastrichtungswechsel beziehungsweise einen Vorzeichenwechsel im I-Anteil des Ellenbogen-Reglers geschätzt.

Ist die Quadrik des Tisches oder noch besser die Quadrik oberhalb des Basisobjektes bekannt, kann das implizite Abstandsmaß mit dem bestehenden Verfahren gekoppelt werden. Diese Sensorfusion ermöglicht eine schnellere Bewegung, da erkannt wird, wenn das Stapelobjekt das Basisobjekt verfehlt. Dies ist beispielsweise der Fall, wenn das Basisobjekt nach der optischen Verdeckung durch Dritte entfernt wird.

Abbildung 75 zeigt den Verlauf der Hand beim Stapeln von Objekten im Konfigurationsraum. Die gelbe Oberfläche entspricht der mittels GA generierten Quadrik der Tischoberfläche. Abbildung 76 zeigt, wie sich die Funktionswerte der vier tischparallelen Quadriken aus dem Satz der Beispielquadriken bei der Ausführung der Aufgabe „Objekt stapeln“ verhalten. Dafür wurden die aktuellen Gelenkwinkel

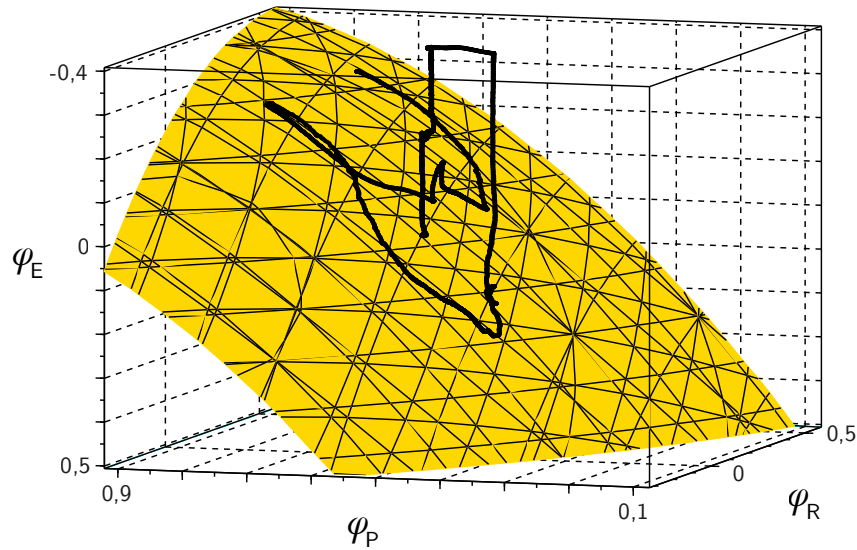


Abbildung 75: Verlauf der Handposition beim Stapeln von Objekten im Konfigurationsraum. Die gelbe Oberfläche entspricht der mittels GA generierten Quadrik der Tischoberfläche. Die Gelenkwinkel sind einheitenlos.

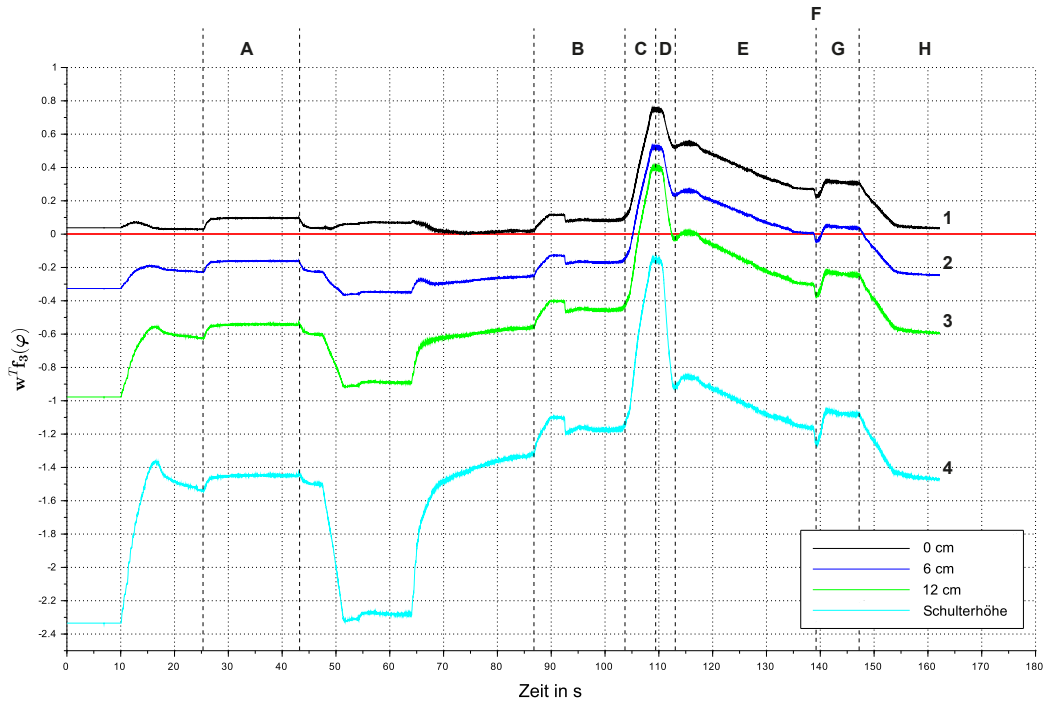


Abbildung 76: Verlauf der vier tischparallelen Quadriken, berechnet mit dem GA während eines Stapelprozesses. Die rote Linie zeigt den Funktionswert null als Referenz. (A) Basisobjekt greifen, (B) Stapelobjekt greifen, (C) Ellenbogen beugen, (D) Schulter auf Position vom Zeitpunkt A bewegen, (E) Ellenbogen strecken, (F) Objekt loslassen, (G) Hand vom Objekt entfernen, (H) Hand auf dem Tisch ablegen.

kel φ in Formel 10 auf Seite 103 eingesetzt. Die Kurven 1–4 zeigen den Verlauf der Funktionswerte der vier Quadriken parallel zur Transversalebene des Roboters mit zunehmendem Abstand zur Tischoberfläche. Die rote Linie zeigt den Wert null als Referenz. Funktionswert und Referenz decken sich immer dann, wenn die Position der Hand exakt auf der jeweiligen Quadrik liegt.

Nachzeichnen von Linien

Beim Nachzeichnen von Linien finden alle Bewegungen in der Nähe der Tischoberfläche statt; aktive Bewegungen des Ellenbogengelenks zum Heben und Senken der Hand sind nicht notwendig. Da es sich beim Zeichnen selbst um vergleichsweise langsame Bewegungen handelt, welche möglichst präzise ausgeführt werden sollen, hat die Reibung zwischen Hand und Tischoberfläche sowie der gegebenenfalls auftretende *Stick-Slip*-Effekt, wenn die Haftreibung periodisch in die Gleitreibung übergeht, einen besonderen Stellenwert. Wird der Kontakt zwischen der Hand und der Tischoberfläche durch Quadriken kurz oberhalb der Tischoberfläche vermieden, reduzieren sich die Reibungseffekte auf die Reibung in den Gelenken selbst.

7.4 Nebenläufiges Lernen der Quadriken

Da sich ein mobiler Roboter relativ zu seiner Umwelt bewegt, diese sich ebenfalls verändert, und auch die Werte der propriozeptiven Sensoren bei Temperaturschwankungen, Verschleiß oder durch anderweitige Manipulationen variieren, ist es erstrebenswert, die Gültigkeit der Quadriken regelmäßig zu überprüfen und bei Bedarf neue Quadriken lernen zu lassen. Dazu bedarf es aktueller Daten. Für die Beispiele in Kapitel 7.1 wurden die Daten der Punktwolken manuell aufgenommen, um die jeweiligen Arbeitsbereiche vollständig abzudecken.

Die Daten, die bei der Hand-Auge-Koordination ohnehin zur Verfügung stehen, sollten so genutzt werden, dass jederzeit eine verbesserte Quadrik vorgeschlagen, eine vorhandene Quadrik angepasst, oder eine neue Quadrik generiert werden kann. Liegen bereits ähnliche Quadriken vor, können deren Gewichte als gute Initialgewichte für eine neue Quadrik gewählt werden.

Zu Beginn der Experimente wurde der rechte Arm des Roboters durch einen anderen Arm ausgetauscht; dieser ist grundsätzlich baugleich, liefert jedoch andere Sensorwerte, da die Sensoren nur grob absolut eingestellt werden. Diese Varianz ist vergleichbar mit einem Ortswechsel oder einer variierenden Umgebung, wie sie im realen Einsatz eines humanoiden Roboters zu erwarten ist. Dies hat zur Folge, dass die vorhandene Tischquadrik keine Gültigkeit mehr hat und an die neuen Gegebenheiten angepasst werden muss.

Da durch die 100 Hz Updaterate der reaktiven Einheit pro Sekunde 100 neue Messpunkte je Sensorwert zur Verfügung stehen – unabhängig davon ob sich der Arm schnell, langsam oder gar nicht bewegt –, müssen geeignete Stützpunkte für das nebenläufige Lernen während der Bewegung identifiziert werden. Dazu wurde ein reaktives Netz implementiert, das für definierte vektorielle Abstände im Konfigurationsraum Punkte zwischenspeichert und das deliberative System über einen neuen Stützpunkt informiert. Als Distanz wurde der konstante Wert $D = 0,03$ gewählt.

Das reaktive Netz lässt sich gut formal beschreiben, in dem die Armgelenke als Vektor φ zusammengefasst werden. Der Vektor φ^* speichert den jeweils letzten gültigen Stützpunkt.

$$\varphi_t = (\varphi_{P,t} \varphi_{R,t} \varphi_{E,t})^T$$

$$\varphi_t^* = \begin{cases} \varphi_t & \text{für } s_t > 0 \\ \varphi_{t-1}^* & \text{für } s_t = 0 \end{cases}$$

$$\Delta\varphi_t = \varphi_t - \varphi_t^*$$

$$s_t = (\Delta\varphi_{t-2}^T \Delta\varphi_{t-2}) > D_{t-2}^2$$

$$S_t = 0,9 S_{t-2} + s_{t-3}$$

Das netzinterne Triggersignal $s \in (0, 1)$ wird nach der Aktualisierung von φ^* umgehend zurückgesetzt. Das Triggersignal $S \in (0 \dots 1)$ ist mit dem deliberativen System verbunden und steht durch die Implementation des Monoflops etwas länger zur Verfügung. Die Amplitude des Triggerimpulses S beträgt nach n Zyklen: $S = 0,9^n$ ($T = n \cdot 10$ ms).

Dieses Netz wird in Abbildung 77 grafisch dargestellt.

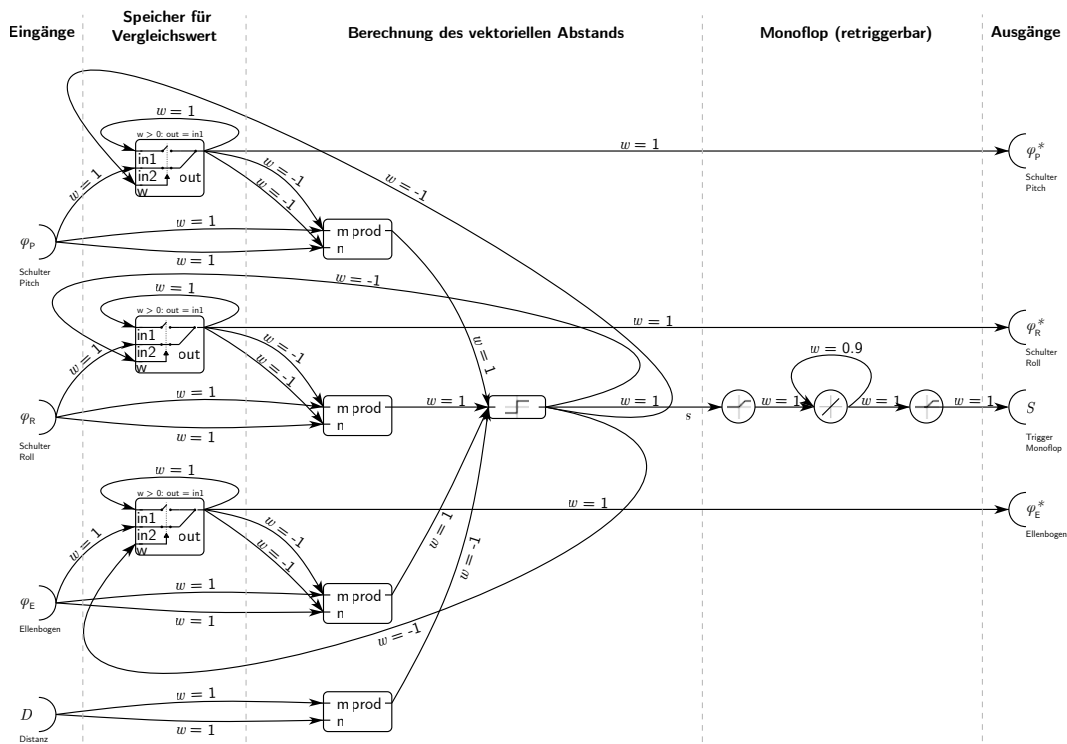


Abbildung 77: Reaktives Netz zur Analyse der Armbewegungen. Wird die vom deliberativen System vorgegebene Distanz (D) im Konfigurationsraum überschritten, löst das einen Triggerimpuls (s, S) aus, und die aktuellen Gelenkwinkel φ werden in φ^* zwischengespeichert.

Wie Abbildung 78 auf Seite 138 zeigt, waren diverse Versuche, die Quadrik mit dem bekannten Gradientenabstiegsverfahren anzupassen, während der Roboter eine Linie nachzeichnet, ohne Erfolg. Dazu wurde der Gewichtsvektor (a) zufällig, (b) mit dem Gewichtsvektor der vorherigen Tischquadrik, (c) mit dem Gewichtsvektor der Quadrik auf Schulterhöhe sowie (d) mit dem Gewichtsvektor der vertikalen Quadrik zur Körpermitte hin initialisiert. Es wurden 100 Iterationen auf 23 resultierenden Stützstellen mit einer Anfangslernrate von $\varepsilon = 0,1$ ausgeführt. Die gelben Flächen sind die resultierenden Quadriken. Erwartungsgemäß ist die Verteilung der Messpunkte zu homogen, um die Tischoberfläche zu repräsentieren, und die Optimierung führt zu sehr speziellen Quadriken, die ausschließlich für diese eine Aufgabe gültig sind.

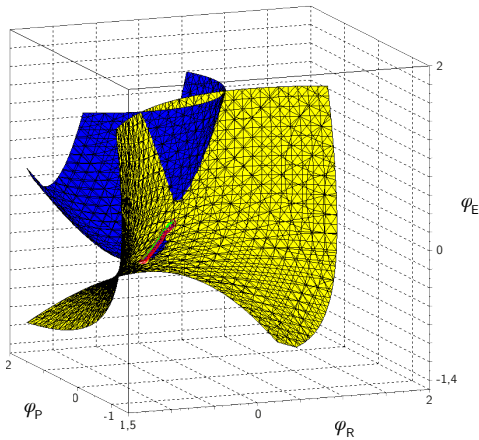
Eine leichte Modifikation im Gradientenabstiegsverfahren führt dennoch zu hinreichenden Resultaten. Dazu wird der Gradientenabstieg auf dem Parametervektor \mathbf{w} mit dem Vektor \mathbf{g} zusätzlich gewichtet und das zehnte Gewicht $w(10)$, welches bislang auf konstant 1 gesetzt wurde, bei der Optimierung einbezogen. Basierend auf der Formel 26 (Seite 108) wird die Formel für den Gradientenvektor wie nachfolgend dargestellt angepasst.³⁸

$$\mathbf{g} = \frac{1}{10} (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 10)^T \quad (31)$$

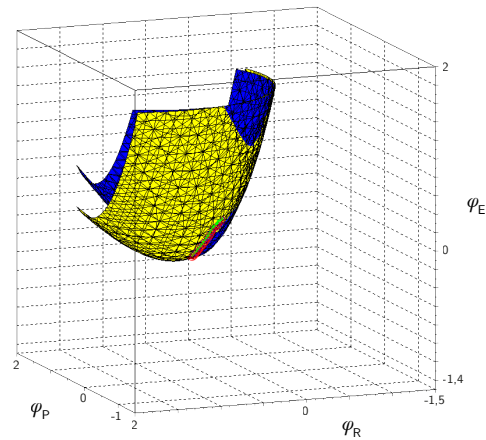
$$\Delta \mathbf{w} = (-\varepsilon \cdot \mathbf{g}) \circ \frac{\partial E_R(\mathbf{w}, \varphi)}{\partial \mathbf{w}} \quad (32)$$

Dadurch erfolgt die Optimierung primär auf dem bislang konstanten Gewicht w_{10} , welches die Referenzquadrik im Wesentlichen im Raum verschiebt; abgeschwächt um eine Zehnerpotenz werden die anderen neun Gewichte nur leicht optimiert, um die Lage im Raum und die Krümmung anzupassen.

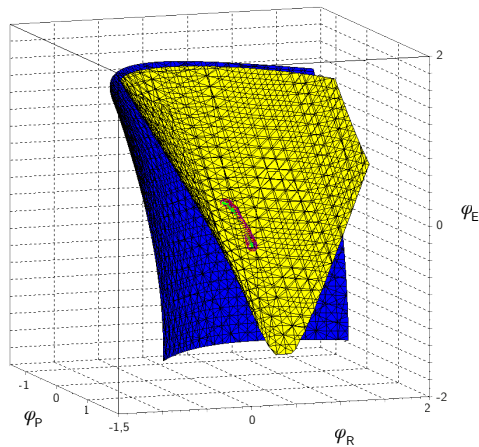
³⁸Der \circ Operator steht für das Hadamard-Produkt. Dieses führt eine elementweise Multiplikation von Matrizen oder Vektoren aus.



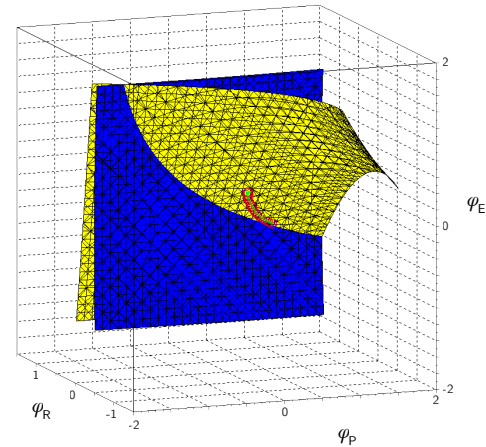
(a) GA mit zufällig gewählten Anfangsgewichten



(b) GA basierend auf der prototypischen Quadrik der Tischoberfläche



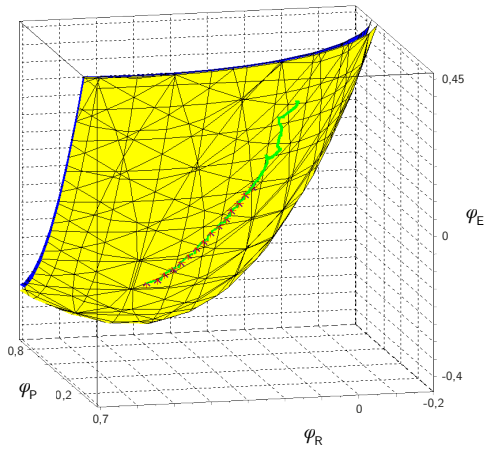
(c) GA basierend auf der tischparallelen prototypischen Quadrik auf Schulterhöhe



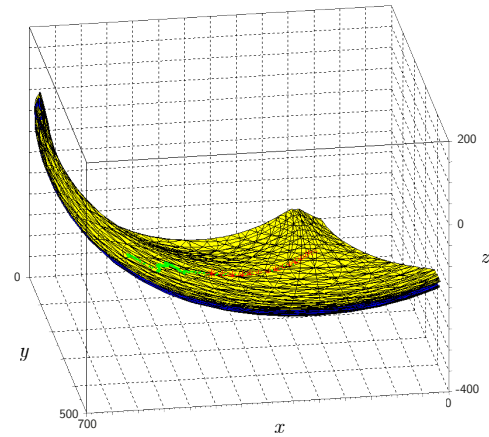
(d) GA basierend auf der vertikalen prototypischen Quadrik zur Körpermitte hin

Abbildung 78: Lernen von Quadriken beim Nachzeichnen. (blau) Referenzquadrik – außer bei (a), hier ist die Tischquadrik zum Vergleich dargestellt; (gelb) gelernte Quadrik; (grün) Verlauf der Messpunkte; (rot) ausgewählte Messpunkte als Stützstellen für den Gradientenabstieg

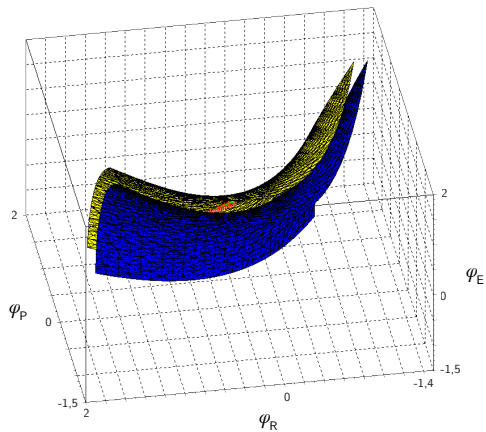
Abbildung 79 auf Seite 139 zeigt das Resultat, sowohl mit der bekannten Tischquadrik als auch mit der Quadrik auf Schulterhöhe als Ausgangsquadrik für den Ablauf des Nachzeichnens. Besonders ist zudem, dass nur die ersten 10 Stützpunkte von insgesamt 25 verfügbaren Punkten für die Optimierung benutzt wurden. Dies bedeutet in der Praxis, dass die Hand bis zum Erreichen des Aufgabenblattes bereits eine hinreichend genaue Quadrik adaptiert hat und die Aufgabe ab dort ohne Kontakt zur Tischoberfläche fortsetzen kann.



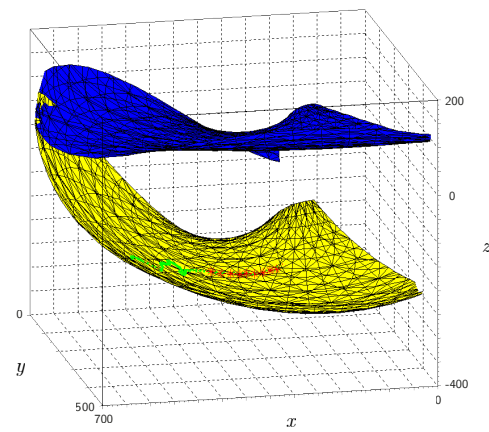
(a) Darstellung im Konfigurationsraum mit der Referenzquadrik in Tischhöhe. Zur besseren Sichtbarkeit der Hand-Trajektorie (grün) ist nur ein Teil des Konfigurationsraums dargestellt. Die Gelenkwinkel sind einheitenlos.



(b) Darstellung im Aktuationsraum mit der Referenzquadrik in Tischhöhe. Die Achsen sind in mm skaliert.



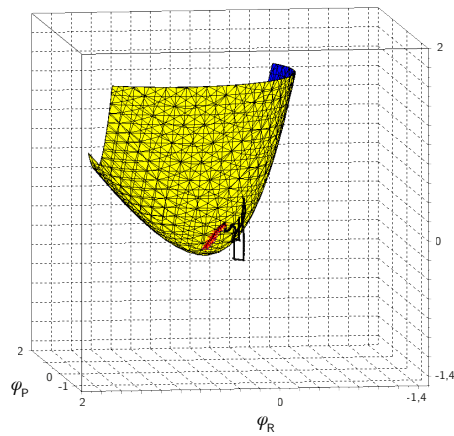
(c) Darstellung im Konfigurationsraum mit der Referenzquadrik in Schulterhöhe. Die Gelenkwinkel sind einheitenlos.



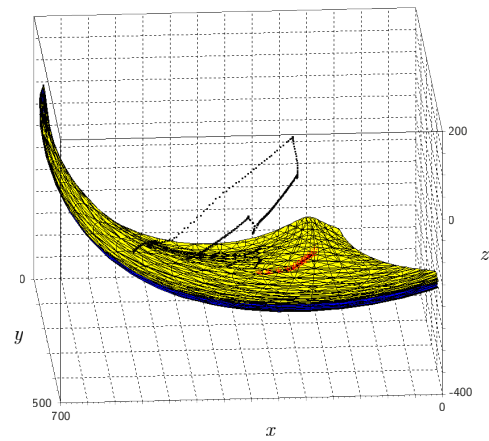
(d) Darstellung im Aktuationsraum mit der Referenzquadrik in Schulterhöhe. Die Achsen sind in mm skaliert.

Abbildung 79: Nebenläufiges Optimieren von Quadriken beim Nachzeichnen, dargestellt im Konfigurationsraum sowie im Aktuationsraum. (blau) Referenzquadrik, auf deren Gewichtsvektor mittels GA-Verfahren die neue Quadrik generiert wird; (gelb) optimierte Quadrik; (grün) Trajektorie der Hand mit einer Abtastfrequenz von 100 Hz; (rot) die zehn ersten Messpunkte der Hand-Trajektorie mit einem Abstand von 0,03 im Konfigurationsraum.

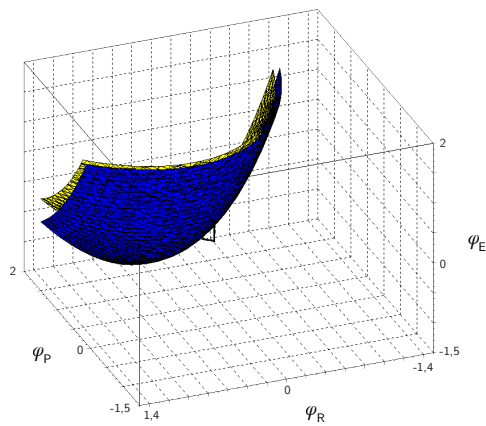
Die Optimierung für das Stapeln von Objekten wurde analog zum Nachzeichnen vorgenommen. Auch hier konnte bereits nach 14 Stützstellen eine repräsentative Quadrik für die Tischoberfläche adaptiert werden, noch bevor die Hand das Basisobjekt greift. Grundsätzlich gilt: Je größer die Fläche der konvexen Hülle ist, die durch die Stützstellen aufgespannt wird, desto genauer ist die resultierende Quadrik. Alle Punkte innerhalb der konvexen Hülle verfälschen die Schätzung. Die Resultate sind in Abbildung 80 dargestellt.



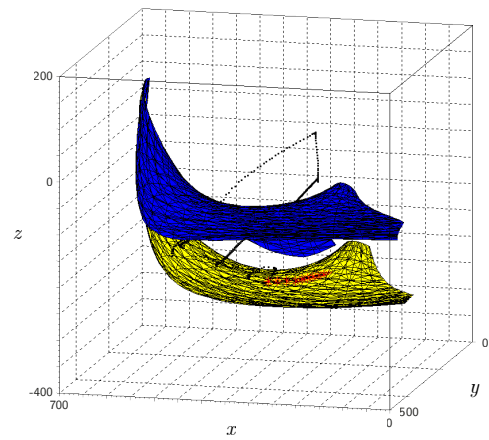
(a) Darstellung im Konfigurationsraum mit der Referenzquadrik in Tischhöhe. Die Gelenkwinkel sind einheitenlos.



(b) Darstellung im Aktuationsraum mit der Referenzquadrik in Tischhöhe. Die Achsen sind in mm skaliert.



(c) Darstellung im Konfigurationsraum mit der Referenzquadrik in 12 cm Höhe. Die Gelenkwinkel sind einheitenlos.



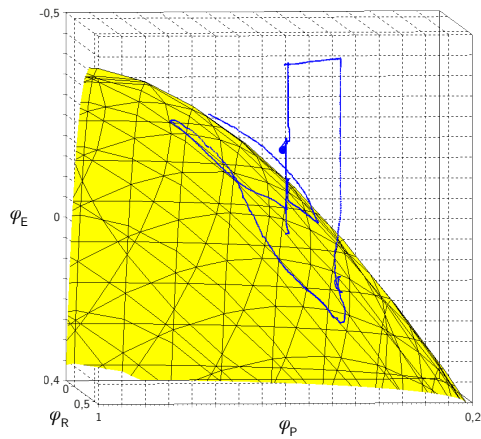
(d) Darstellung im Aktuationsraum mit der Referenzquadrik in 12 cm Höhe. Die Achsen sind in mm skaliert.

Abbildung 80: Nebenläufiges Optimieren von Quadriken beim Stapeln von Objekten, dargestellt im Konfigurations- und Aktuationsraum. (blau) Referenzquadrik, nach dessen Gewichtsvektor das GA-Verfahren die neue Quadrik generiert; (gelb) optimierte Quadrik; (schwarz) Trajektorie der Hand mit einer Abtastfrequenz von 100 Hz; (rot) die 14 ersten Messpunkte der Hand-Trajektorie mit einem vektoriellen Abstand von 0,03 im Konfigurationsraum.

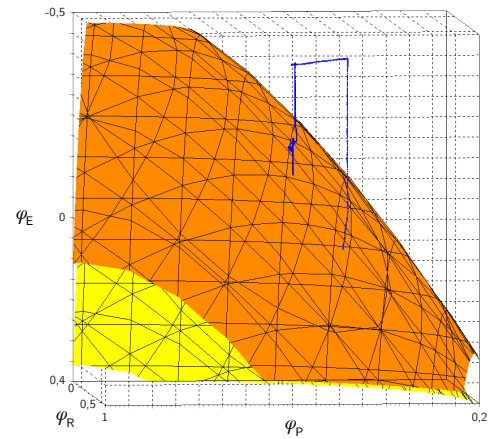
Beim Lernen einer Quadrik für das Abstellen des Stapelobjekts auf dem Basisobjekt liegen deutlich weniger Stützpunkte vor, die für die Optimierung genutzt werden können. Abbildung 81 auf Seite 143 zeigt den Verlauf der Gelenkwinkel (blau). Die dickere blaue Kugel zeigt den Punkt, an dem die Kollisionserkennung im Ellenbogengelenk das Aufsetzen des Stapelobjekts auf dem Basisobjekt erkannt hat. Auf Basis der Tischquadrik (gelb) wird, wenn nur eine oder sehr nah beieinander liegende Stützstellen vorliegen, die Quadrik über das konstante Gewicht w_{10} lediglich verschoben. Basierend auf dem Gewichtungsvektor \mathbf{g} aus Formel 31 zeigt Formel 33 den modifizierten Gewichtungsvektor \mathbf{g} für die Optimierung mit einer oder zwei Stützstellen zur Berechnung des Gradientenabstiegs nach Gleichung 32.

$$\mathbf{g} = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1) \quad (33)$$

Liegen drei Trajektorien oder mehr vor, ist auch die räumliche Struktur der Quadrik optimierbar. Formel 31 hat dabei gute Ergebnisse geliefert, wie Abbildung 82 auf Seite 143 zeigt.

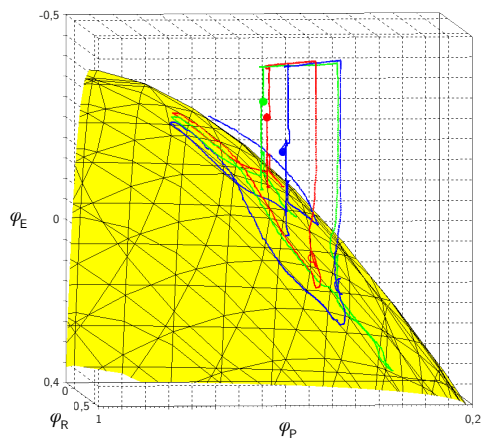


(a) Die gelb dargestellte Quadrik entspricht der Tischoberfläche.

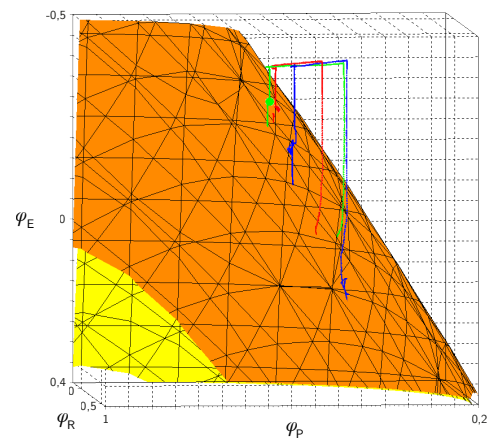


(b) Die orange dargestellte Quadrik wurde mittels GA auf den Kollisionspunkt optimiert.

Abbildung 81: Verlauf der Gelenkwinkel beim Objekte stapeln. Die dickere blaue Kugel zeigt den Punkt, an dem die Kollisionserkennung im Ellenbogengelenk das Aufsetzen des Stapelobjekts auf dem Basisobjekt erkannt hat. Die Gelenkwinkel sind einheitenlos.



(a) Die gelb dargestellte Quadrik entspricht der Tischoberfläche.



(b) Die orange dargestellte Quadrik wurde mittels GA auf die drei Kollisionspunkte optimiert.

Abbildung 82: Verlauf der Gelenkwinkel beim Stapeln von Objekten. Die drei Trajektorien (blau, rot, grün) bilden jeweils einen Stapelvorgang ab. Die dickeren Kugeln in der Farbe der jeweiligen Trajektorie zeigen den Punkt, an dem die Kollisionserkennung im Ellenbogengelenk das Aufsetzen des Stapelobjekts auf dem Basisobjekt erkannt hat. Die Gelenkwinkel sind einheitenlos.

7.5 Auswahl einer Quadrik als Optimierungsbasis

Beim Einsatz von Quadriken zur Regelung eines Armgelenkes oder als Optimierungsbasis ist jederzeit die nächstbeste Quadrik aus dem Speicher aller bekannter Quadriken – mit anderen Worten, aus dem Gedächtnis des Roboters – zu bestimmen. Welche Quadrik sich am besten eignet, um die aktuelle Bewegung zu beschreiben, hängt wesentlich von der jüngsten Trajektorie der Hand zur aktuellen Position ab. Das ist unabhängig davon, wofür die Quadrik eingesetzt werden soll, sei es zur Regelung beim Greifen von Objekten oder beim Nachzeichnen, als initiale Gewichte beim Lernen einer neuen Quadrik, oder zur Auswahl von Begrenzungen des Aktuationsraums. Wurde eine Quadrik als gültig identifiziert, ist während der Armbewegung kontinuierlich zu prüfen, ob diese weiterhin gültig ist oder zumindest die beste Näherung darstellt. Dazu steht ein Ringbuffer mit den letzten 30 Stützpunkten zur Verfügung. Immer wenn das in Abbildung 77 auf Seite 136 dargestellte reaktive Netz einen neuen Stützpunkt meldet, wird dieser in den Ringbuffer übernommen und dabei der älteste Stützpunkt überschrieben.

Die Analyse typischer Bewegungen des Roboterarms hat gezeigt, dass diese in der Regel durch Quadriken in Form leicht gekrümmter Flächen modelliert werden können. Sie werden durch Hindernisse wie der Tischoberfläche im Aktuationsraum bestimmt und zusätzlich durch die Gelenkanschlüsse im Konfigurationsraum begrenzt. Analysiert man die Verteilung der Punktwolke aus den gemessenen Stützstellen nach ihrer Ausdehnung in drei orthogonalen Achsen, ist die Ausdehnung bei validen Stützpunkten in zwei Richtungen deutlich größer als in der dritten Richtung - sie bilden näherungsweise eine Fläche. Diese Beziehung wird genutzt, um die zur Verfügung stehenden Stützstellen für die Auswahl einer gültigen Quadrik zu bewerten.

Bei den drei vorgestellten Aufgaben können die Stützstellen auf zwei Arten zur Verfügung stehen: (1) durch die jüngste Bewegung des Armes und die permanente Analyse des vektoriellen Abstandes durch das reaktive Netz; der Ringbuffer ist permanent mit Werten für 30 Stützpunkte gefüllt, von denen die jeweils älteren keine Relevanz mehr haben können; (2) durch wenige Punkte, die beim Abstellen des Stapelobjektes auf das Basisobjekt extrahiert werden.

Werden die Stützstellen auf Basis der vergangenen Bewegung definiert, erfolgt die Bewertung anhand einer *ORDM*-Heuristik (*One-Reason-Decision-Making*). Dazu kann das System ausgehend vom aktuellsten Stützpunkt so lange in die Vergangenheit schauen, bis der einhüllende Quader der Punktwolke im Volumen stark zunimmt³⁹, oder mit anderen Worten, bis sich die Punkte von einer angenäherten Ebene

³⁹Der einhüllende Quader ist dazu in seiner räumliche Lage so orientiert, dass sein Volumen stets minimal ist.

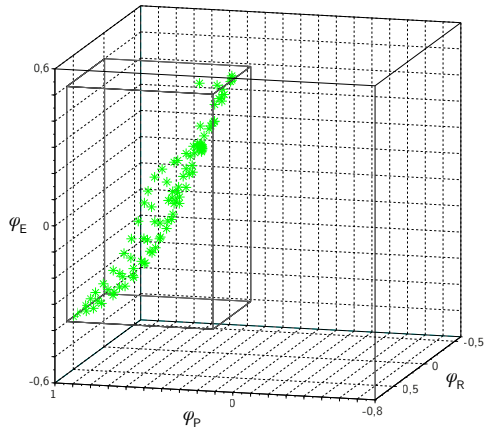
ne zunehmend in einen einhüllenden Quader verändern. Formal eignet sich dafür die Berechnung der Hauptkomponentenanalyse (*Principal Component Analysis/PCA*) auf der Punktwolke. Als gutes Abbruchkriterium für die *ORDM*-Heuristik hat sich bewährt, wenn der relative Eigenwert der dritten Hauptkomponente 5% überschreitet. Ergibt die *PCA* mit fünf oder mehr Stützstellen keine sinnvollen Werte, kann keine Schätzung auf Basis der vergangenen Bewegung erfolgen, da die Bewegung keinem Muster folgt, welches sich in Form einer Quadrik abbilden lässt und anschließend sinnvoll zu nutzen wäre. Je mehr Stützstellen gültig sind, desto genauer wird die folgende Schätzung. Zusätzlich ist es von Vorteil, wenn die Eigenwerte der ersten beiden Hauptkomponenten eine vergleichbare Größenordnung besitzen, um eine gute Schätzung zu ermöglichen; die Punkte besitzen dann in zwei Richtungen eine vergleichbare Varianz. Dies ist bei Bewegungen unmittelbar vor dem Berühren des Stapelobjekts beziehungsweise dem Startpunkt der Linie beim Nachzeichnen unwahrscheinlich, da hier vor allem direkte Bewegungen ausgeführt werden⁴⁰. In diesem Fall ist beispielsweise die Erkennung der Hand durch Bewegungen im Kamerabild mit stark reduzierter Auflösung von Vorteil, wie diese in der Variante 1 beim Objekt greifen vorgestellt wurde. Abbildung 46 auf Seite 84 zeigt diesen Vorgang, der zwangsweise zu stärkeren Armbewegungen bei der Anbahnung des Greifens führt und damit Stützstellen von höherer Qualität für die Auswahl einer gültigen Quadrik liefert.

Auf Basis der als relevant eingestuften Stützstellen erfolgt die Berechnung des kleinsten quadratischen Fehlers. Dazu wird die etablierte Formel 11 auf Seite 105 auf alle Quadriken im Speicher angewandt. Je nach Größe des Fehlers wird die Quadrik mit dem niedrigsten Fehlerwert als gültig erklärt oder dient als Initialwert für eine Optimierung auf Basis der Stützstellen. Der Schwellwert (obere Grenze) für eine gültige Quadrik ist mit $E_T = 0,05$ definiert und empirisch ermittelt worden.

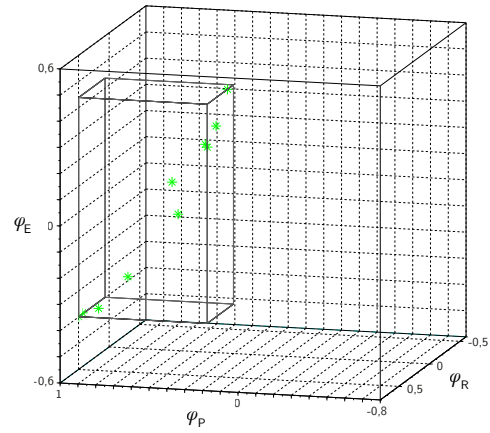
Abbildung 83 zeigt Experimente zur Hauptkomponentenanalyse auf ausgewählten Punktwolken. Die folgende Tabelle 7 führt die Art der zugehörigen Bewegungen des Roboterarms auf, die zu den Punktwolken geführt haben und stellt die relativen Eigenwerte der drei Hauptkomponenten dar. Erwartungsgemäß entsprechen alle Punkte, die aus den Datensätzen der Quadriken entnommen wurden, vollständig dem Kriterium für gültige Stützstellen. Eine Ausnahme bildet die in Abbildung 83(c) dargestellte Kombination aus zehn Punkten der Tischoberfläche sowie fünf Punkten einer orthogonal zu dieser verlaufenden vertikalen Ebene. Der zuvor definierte Grenzwert von 5% des relativen dritten Eigenvektors der *PCA* wurde mit fast 8% gezielt überschritten. Würde man von den 15 Punkten die letzten fünf Punkte

⁴⁰Dies führt eher dazu, dass die zweit- und drittgrößte Hauptkomponente vergleichbar groß sind.

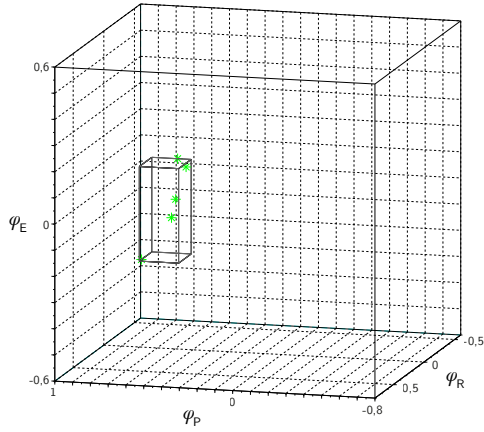
von der *PCA* ausschließen, wäre das Kriterium wieder erfüllt und es liegen gültige Stützstellen zur weiteren Verarbeitung vor. Die beiden letzten Datensätze wurden während des Nachzeichnens und dem Stapeln von Objekten aufgezeichnet.



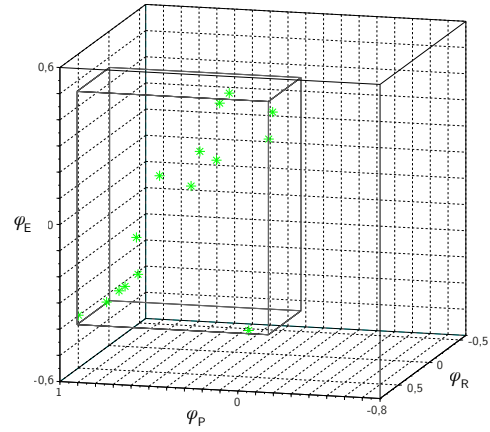
(a) Tischoberfläche (100 Stützpunkte)



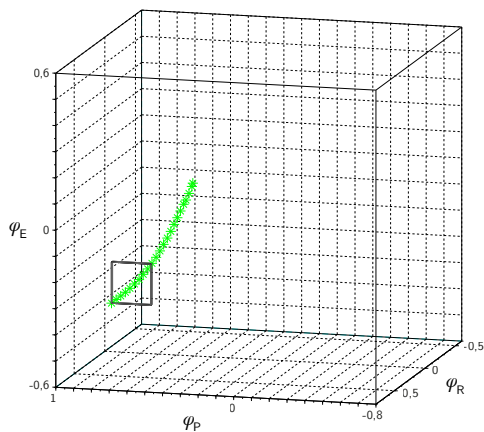
(b) Tischoberfläche (10 Stützpunkte)



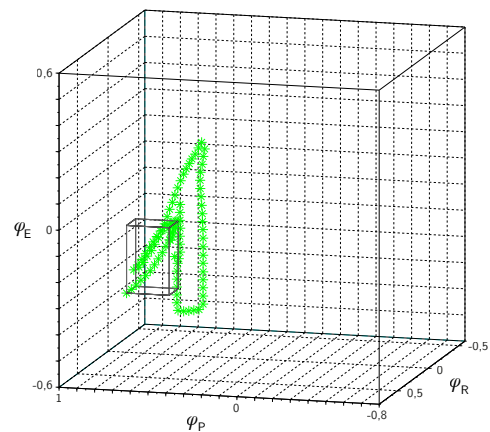
(c) Tischoberfläche (5 Stützpunkte)



(d) Tischoberfläche (10 Stützpunkte), kombiniert mit vertikaler Ebene (5 Stützpunkte)



(e) Nachzeichnen (erste 10 Stützpunkte)



(f) Stapeln (erste 14 Stützpunkte)

Abbildung 83: Experimente mit der Hauptkomponentenanalyse von ausgewählten Punktwolken (eingeschlossen im inneren Quader) zur Verifikation der Heuristik für zulässige Stützstellen zur Wahl der nächst passenden Quadrik aus dem Gedächtnis des Roboters. Die Gelenkwinkel sind einheitenlos und bilden den gesamten Konfigurationsraum ab.

Tabelle 7: Quellen der Punktwolken in Abbildung 83 sowie die relativen Eigenwerte der drei Hauptkomponenten, berechnet mittels Hauptkomponentenanalyse. Die Eigenwerte wurden mit absteigender Größe in einem Vektor zusammengefasst und normiert.

Abbildung	Quelle der Punktwolke	Eigenwertvektor
83(a)	Tischoberfläche (100 Stützpunkte)	(0,659 0,331 0,01)
83(b)	Tischoberfläche (10 Stützpunkte)	(0,803 0,191 0,006)
83(c)	Tischoberfläche (5 Stützpunkte)	(0,716 0,283 0,001)
83(d)	Tischoberfläche (10 Stützpunkte), kombiniert mit vertikaler Ebene (5 Stützpunkte)	(0,586 0,336 0,078)
83(e)	Nachzeichnen (erste 10 Stützpunkte)	(0,91 0,089 0,001)
83(f)	Stapeln (erste 14 Stützpunkte)	(0,981 0,018 0,001)

Die Berechnung des normierten Eigenwertvektors erfordert bei der Anwendung auf dem eingebetteten System des Roboters keine vollständige *PCA* und lässt sich für einen effizienten Einsatz reduzieren. Im Rahmen dieser Arbeit wurde der Vektor in folgenden Schritten berechnet:

Anhand einer Matrix aus $5 \leq N \leq 30$ Stützstellen mit je einem Spaltenvektor φ der drei Armgelenkwerte wird (1) die *Scatter*-Matrix⁴¹ \mathbf{S} berechnet:

$$\mathbf{S} = \sum_{k=1}^N ((\varphi_k - \bar{\varphi})(\varphi_k - \bar{\varphi})^T) \quad (34)$$

$\bar{\varphi}$ ist der mittlere Gelenkwertvektor, der den arithmetischen Mittelwert für jedes Armgelenk beinhaltet.

$$\bar{\varphi} = \frac{1}{N} \sum_{k=1}^N \varphi_k \quad (35)$$

⁴¹Die bekanntere Kovarianz-Matrix entspricht der *Scatter*-Matrix, wird jedoch vor der Matrixmultiplikation zusätzlich mit $\frac{1}{N-1}$ multipliziert. N ist die Anzahl an Stützstellen. Die resultierende Abweichung in den Eigenwerten ist aufgrund der folgenden Normierung irrelevant.

Im Anschluss erfolgt (2) die Berechnung der Eigenwerte über den numerischen QR-Algorithmus, sowie (3) das Sortieren der Eigenvektoren in absteigender Größe. Durch (4) die Normierung des Eigenwertvektors ist ein Vergleich der drei Eigenvektoren unabhängig vom Betrag möglich.

$$\mathbf{E}_{norm} = \begin{pmatrix} \frac{E_1}{\sum \mathbf{E}} \\ \frac{E_2}{\sum \mathbf{E}} \\ \frac{E_3}{\sum \mathbf{E}} \end{pmatrix} \quad (36)$$

7.6 Zusammenfassung

Um die Komplexität der Regelung der Armbewegung zu reduzieren, wird der Konfigurationsraum zeitweise um die Dimension des Ellenbogengelenkes reduziert. Die Regelung dieses Gelenks erfolgt in der entsprechenden Zeit über eine auf Quadriken basierende sensomotorische Schleife. Eine Quadrik ist die Lösungsmenge eines Polynoms zweiten Grades und bildet ein implizites Modell für den Zusammenhang zwischen Roboterhand und Umwelt. Zur Analyse der Generierung dieser Quadriken wurden Punktwolken der Gelenkwinkel im Konfigurationsraum des rechten Armes für sieben verschiedene Arbeitsbereiche aufgenommen. Die evaluierten Optimierungsverfahren sind das *Least-Squares*-Verfahren, das rekursive *Least-Squares*-Verfahren sowie das Gradientenabstiegsverfahren. Letzteres hat sich für diese Anwendung als das beste Verfahren herausgestellt, sowohl in der Eignung der resultierenden Quadriken als auch in der effizienten Implementation auf dem eingebetteten System. Die quadrikenbasierte Regelung kann sowohl über das deliberative als auch über das reaktive System erfolgen. Das reaktive Netz stellt zudem die Möglichkeit bereit, den Arbeitsbereich im Aktuationsraum durch bis zu sechs Quadriken abzugrenzen, sowie das deliberative System zu informieren, wenn sich die Hand um eine definierte Strecke im Konfigurationsraum bewegt hat. Diese Informationen werden eingesetzt, um Stützstellen für eine Online-Optimierung der Quadriken während der Ausführung der Aufgaben zu ermöglichen. Um die Trajektorie der Stützstellen auf geeignete Punkte zu reduzieren, wird eine Hauptkomponentenanalyse (PCA) auf dem eingebetteten System implementiert und die Analyse durch eine *ORDM*-Heuristik organisiert. Die Nutzung von Quadriken hat auf alle drei Verfahren der Hand-Auge-Koordination eine positive Auswirkung.

8 Empirisches Optimieren der visuomotorischen Fähigkeiten

Die visuomotorischen Aufgaben aus Kapitel 6 konnten mittels einfacher sensomotorischer Schleifen und Heuristiken erfolgreich gelöst werden. Die Quadriken in Kapitel 7 optimierten diese Verfahren durch erweitertes Weltwissen und ersetzten damit die Umgebungsbeschränkung (Berührung der Oberflächen) durch ein implizites Modell, das den Bezug zwischen Hand und Umwelt für den jeweiligen Arbeitsbereich beschreibt. In diesem Kapitel werden Methoden aufgezeigt, mit denen die Qualität der Verhaltensabläufe durch empirische Optimierung weiter verbessert werden kann. Im Gegensatz zum vorherigen Kapitel, das die Optimierung bei der Ausführung (auf der motorischen Seite der Handlungsabläufe) beschreibt, stellt dieses Kapitel vor, wie die Informationen der visuellen Wahrnehmung zur Optimierung genutzt werden können.

Im Rahmen dieser Arbeit beschränkt sich die visuelle Wahrnehmung auf die Lokalisation von Objekten mit unterschiedlichen Farben bzw. anhand von Eckenmerkmalen im Koordinatensystem des Kamerabildes. Die meisten der vorgestellten Verfahren haben zum Ziel, die Objekte visuell zusammenzuführen (Hand zu Gegenstand beziehungsweise Gegenstand zu Gegenstand) oder ein Objekt in eine definierte Position im Kamerabild zu navigieren (Gegenstand, Hand oder Stiftspitze). Abhängig von der Position der Hand, des Gegenstandes, des Stiftes oder des Zeichenblattes in der Bildebene variieren jedoch auf Grund der Parallaxe die Zielpositionen für die Lösung der jeweiligen Aufgabe im Kamerabild. Je weiter die Handlung vom Körper entfernt ausgeführt werden soll, desto unwahrscheinlicher ist ihr Erfolg, da der Roboter daneben greift beziehungsweise die Linie versetzt zeichnet. Bisher wurde dieses Problem zwar festgestellt, jedoch noch nicht versucht, dieses durch gezielte Maßnahmen zu lösen.

Bekannt aus der theoretischen Einführung des *Visual Servoing* in Kapitel 2.2 (Seite 11) werden die Parallaxefehler üblicherweise durch die Anwendung von direkten Körpermodellen und Abstandsinformationen kompensiert oder treten gar nicht erst auf. Entsprechend dem Ziel dieser Arbeit werden dagegen im Folgenden einfache Prinzipien vorgestellt, mit denen diese Fehler ohne explizite Modelle durch empirische Verhaltensoptimierung reduziert werden können.

8.1 Abbilden der Korrekturen durch Offsetkarten

Die visuelle Wahrnehmung eines Objektes wird auf einen Bildpunkt mit einer horizontalen und einer vertikalen Position beschränkt. Ob die Bildkoordinate einen Objektschwerpunkt oder eine ausgewählte Ecke des Objektes beschreibt, hängt von der jeweiligen Aufgabe und der Seite des involvierten Armes ab. Unter der Voraussetzung, dass nur eine Hand genutzt wird, lässt sich eine visuomotorische Greifpose durch insgesamt zehn Parameter beschreiben.

Die zehn Parameter lauten:

- 2x Werte der Bildkoordinate (horizontal x /vertikal y)
- 3x Halsgelenke (*Yaw/Pitch/Roll*)
- 3x Armgelenke (Schulter-*Pitch* φ_P /Schulter-*Roll* φ_R /Ellenbogen φ_E)
- 1x Drehung des Handgelenks
- 1x Zustand der Hand (offen/geschlossen)

Da durch die sensomotorischen Kopplungen bereits Zwangsbedingungen zwischen den Parametern bestehen, lässt sich die Beschreibung der visuomotorischen Greifpose weiter vereinfachen. Dies betrifft die drei Halswinkel, da die Blickrichtung (*Pitch* und *Yaw*) durch die Zentrierung des Objektes im Kamerabild vorgegeben und auch die *Roll*-Bewegung je nach Anwendung fest an den Unterarm oder das Gravitationsfeld gekoppelt ist. Mit der sensomotorischen Ausrichtung des Handgelenkes zum Gravitationsfeld fällt auch die Drehung des Handgelenks weg. Geht man davon aus, dass beim Greifen die Hand immer geschlossen ist, bleiben noch fünf Parameter übrig: die beiden Werte der Bildkoordinate sowie die drei Armgelenke.

Unterstützt durch Quadriken, die die Regelung des Ellenbogengelenkes für definierte Arbeitsbereiche bestimmen, wurden für jede Quadrik vierdimensionale Offsetkarten eingeführt. Diese Karten lassen sich mit den ersten beiden Dimensionen Schulter-*Pitch* φ_P und Schulter-*Roll* φ_R als projizierter Konfigurationsraum darstellen. Für jeden Punkt dieser Darstellung existieren zwei weitere Dimensionen: die Offsetwerte für die horizontale und die vertikale Bildkoordinate. Abbildung 84 visualisiert dies durch zwei Grafiken, je eine für den horizontalen und vertikalen Offset. Die graue Fläche stellt die Referenz dar, der Offset wird durch dunklere (positive Korrektur) und hellere Bereiche (negative Korrektur) angezeigt.

Die Idee ist, dem aktuellen Punkt im Konfigurationsraum, der durch die beiden

Schultergelenke sowie die assoziierte Quadrik eindeutig beschrieben ist, Korrekturdaten der visuellen Wahrnehmung (Bildkorrekturdaten) zuzuordnen, um auftretende Parallaxefehler positionsabhängig zu kompensieren und dadurch das Greifen im Aktuationsraum zu optimieren.

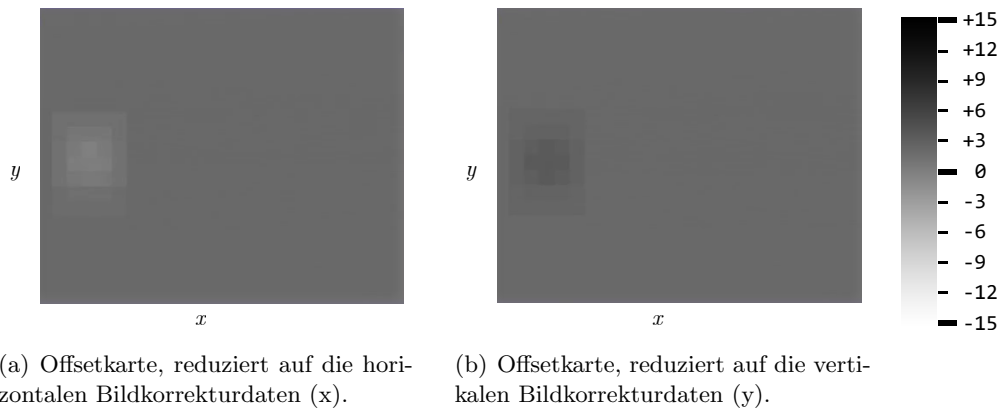


Abbildung 84: Visuelle Darstellung einer Offsetkarte für die empirisch ermittelten Korrekturdaten in den visuell bestimmten Objektpositionen. Die beiden Bildachsen beschreiben die Gelenke Schulter-*Pitch* φ_P und Schulter-*Roll* φ_R . Von der Fläche abweichende Grautöne zeigen die Korrektur in den Objektpositionen: im dunkleren Bereich erfolgt eine positive Korrektur, im helleren eine negative. Diese Offsetkarten wurden nach einer einmaligen Korrektur beim Nachzeichnen exportiert.

Die Anwendung der Offsetkarten wurde auf einen Arbeitsbereich, den die Roboterhand auf Tischhöhe erreichen kann, mit 50 cm Breite und 40 cm Tiefe (aus der Sicht des Roboters) großzügig bemessen. Erfolgt die Bewegung oberhalb der Tischoberfläche, vergrößert sich der relevante Bereich im Konfigurationsraum bis auf Schulterhöhe vernachlässigbar wenig, bedingt durch die Morphologie des Roboterkörpers. Bei 25 Unterteilungen des Wertebereichs für das *Roll*-Gelenk und 20 Unterteilungen für das *Pitch*-Gelenk sind Korrekturwerte mit einer Auflösung im Aktuationsraum von ca. 2 cm möglich. Die scheinbar geringe Auflösung ist hinreichend genau und vermeidet unerwünschte Schwingungen durch die bei höherer Auflösung periodisch anfallende Korrektur der Bilddaten: Erfolgt an einem Punkt eine Korrektur der Bildkoordinaten, führt dies zu einer Korrektur der Handposition im Aktuationsraum; da für die neue Handposition jedoch noch keine Bildkorrekturdaten vorliegen, wird die Hand zurück in die Ausgangsposition bewegt, für die bereits Korrekturdaten vorliegen – es kommt zu einer oszillierenden Bewegung. Um dieses Verhalten zu vermeiden, sollte die Korrekturauflösung für beide Schultergelenke so niedrig wie möglich sein. Bei einer Korrektur der Bildpixel werden die umliegenden Punkte im Konfigurationsraum ebenfalls mit halber Intensität korrigiert, um die Gradienten zwischen den Korrekturpunkten zu reduzieren. Auf der eingesetzten Roboterhardware ergibt sich

ein Speicherbedarf von unter 1 kB je Offsetkarte. Beim Generieren neuer Quadriken (40 Byte) auf Basis vorhandener Quadriken werden auch die assoziierten Offsetkarten kopiert.

8.2 Online-Analyse des Greifvorgangs

Das Greifen eines Gegenstandes erfordert insbesondere bei der eingesetzten Roboterplattform eine sehr genaue Positionierung der Hand. Neben der propriozeptiven Analyse des Greifens über die Griffkraft aus Kapitel 5 (Seite 71), ist unterstützend auch eine Abschätzung von Fehlerfällen exterozeptiv über die visuelle Wahrnehmung möglich. Bedingt durch die Greif-Heuristik werden die Hand und der Gegenstand zwangsläufig angenähert. Die visuelle Fehlerabschätzung ist daher nur kurz vor oder während des Greifvorganges erforderlich, wenn sich die Hand zum Objekt eindreht und die Finger schließt. Da dieser finale Prozess stark von der Morphologie und dem Funktionsprinzip des jeweiligen Endeffektors (Hand oder Greifer) beeinflusst wird, ist die folgende Analyse nur für die in der Arbeit eingesetzte Roboterhand (Vier-Finger-Greifer) gültig.

Hier wurden vier Fehlertypen identifiziert, die einzeln oder in Kombination auftreten können⁴²:

- Die Hand ist zu nah am Greifobjekt und verschiebt dieses horizontal noch vor dem Eindrehen der Hand (beim rechten Arm nach links). Ursache: Die horizontale Korrektur ist zu klein. Die Anpassung des Korrekturwertes muss bereits während der Ausführung vorgenommen werden, damit das Greifobjekt nicht aus dem Arbeitsbereich geschoben wird.
- Das Greifobjekt befindet sich neben dem Handkorpus⁴³. Der Daumen setzt beim Eindrehen der Hand auf dem Greifobjekt auf und verdeckt dieses partiell. Die Hand hebt ab und der Ellenbogen wird in Folge der Handgelenkdrehung gebeugt. Erfolgt bereits eine Regelung des Ellenbogens mit einer Quadrik auf Tischhöhe, ist eine Vorzeichenänderung im I-Anteil des Ellenbogenreglers zu detektieren. Ursache: Die vertikale Korrektur ist zu groß.

⁴²Die folgenden Beschreibungen der Handposition zum Objekt beziehen sich auf die visuelle Wahrnehmung, also das Kamerabild, aus der Perspektive des Roboters.

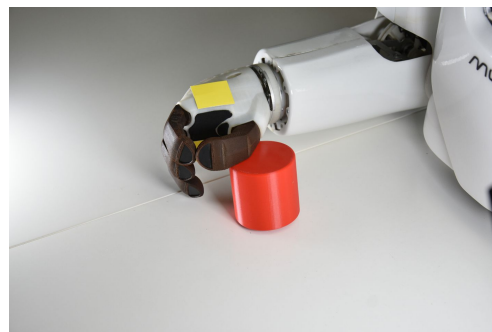
⁴³Als Handkorpus wird hier der Hauptteil der Hand (im Bild weiß) ohne die Finger (schwarz) bezeichnet; er enthält den Antriebsmotor und das Zugmittelgetriebe.

- Die Hand ist auf Tischebene zu weit entfernt vom Greifobjekt. Der Daumen setzt beim Eindrehen der Hand auf dem Greifobjekt auf und verdeckt dieses partiell. Das Greifobjekt wird beim Schließen der Hand zur Hand hin verschoben (beim rechten Arm nach rechts). Ursache: Die horizontale Korrektur ist zu groß.
- Das Greifobjekt befindet sich neben den Fingerspitzen. Es wird beim Schließen der Hand in horizontaler Richtung von der Hand weg verschoben (beim rechten Arm nach links). Die Finger können dabei das Greifobjekt partiell verdecken. Ursache: Die vertikale Korrektur ist zu klein.

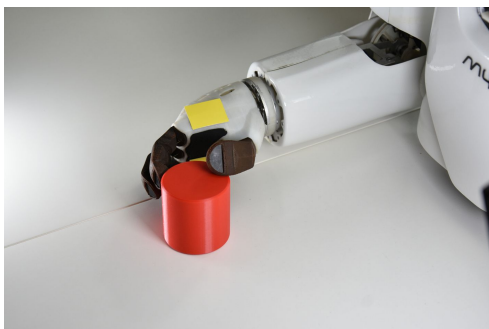
In Abbildung 85 sind sowohl ein idealer Griff als auch die drei Fehlerarten dargestellt, die nicht bereits während der Annäherung der Hand an das Greifobjekt auftreten.



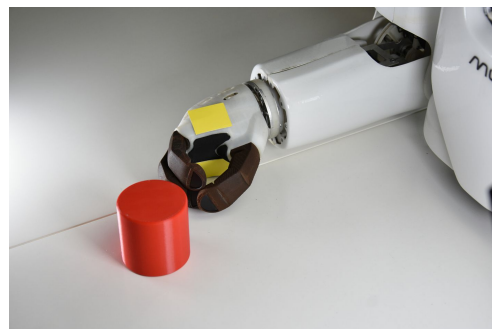
(a) Idealer Griff



(b) Objekt neben dem Handkorpus



(c) Hand und Objekt auf Tischebene zu weit voneinander entfernt



(d) Objekt neben den Fingerspitzen

Abbildung 85: Darstellung unterschiedlicher Fehlerarten beim Greifen eines Gegenstandes. Diese Fehlerarten sind nur für den hier gezeigten Greifer in Form einer Vier-Finger-Hand gültig.

Die Analyse erfolgt als *Take-The-Best(-Leave-The-Rest)*-Heuristik⁴⁴, priorisiert in der genannten Reihenfolge. Sobald eines der Kriterien in absteigender Priorität zutrifft, wird die Analyse beendet, die Korrektur durchgeführt und die Heuristik beginnt die Analyse unbeeinflusst von der vorhergegangenen Korrektur von vorn. Die Priorisierung erfolgte sowohl anhand des Fehlergebnisses – der erste Fehlertyp führt dazu, dass das Objekt aus dem Arbeitsbereich geschoben wird – als auch an der Zuverlässigkeit der Fehlererkennung. Besonders beim vierten Fehlertyp muss es nicht immer zu einer Verschiebung des Gegenstandes kommen. Ist das Greifen ohne Erfolg und keiner der vier Fehlerarten wurde erkannt (erschöpfende Suche), erfolgt das Greifen erneut ohne eine Anpassung der Offsetkarten.

Außerdem besteht noch die Möglichkeit, dass der Mensch den Roboter auf den Fehler hinweist, wenn die eigenständige Erkennung fehlerhaft oder nicht möglich war. Wird einer der Fehlerarten erkannt, erfolgt die Anpassung des Korrekturwertes mit einem konstanten Inkrement von +0,008 beziehungsweise einem konstanten Dekrement um –0,006. Die unterschiedlichen Beträge für Inkrement und Dekrement ermöglichen es, dass Änderungen groß genug sind, direkt einen wirkungsvollen Effekt zu haben, jedoch bei zunehmender Anzahl an Sprüngen um den idealen Offsetwert die Genauigkeit weiter zunehmen kann.

8.3 Auswertung des Stapelns von Objekten

Je nach eingesetzter Variante hat der Versatz zwischen Basis- und Stapelobjekt verschiedene Ursachen.

Variante 1 basiert auf der Blickheuristik. Das Stapelobjekt wird dem Basisobjekt durch eine Bewegung auf der Blickachse angenähert, bis es auf diesem abgestellt wurde. Der Versatz zwischen den beiden Objekten entspricht der Flucht der Blickachse der Kamera und wird in Abbildung 52 auf Seite 92 demonstriert. Als systematischer Fehler ist dieser abhängig von der Position und der Höhe des Basisobjektes und lässt sich durch Offsetkarten in Kombination mit einer gültigen Quadrik als Höhenreferenz gut kompensieren.

Bei Variante 2 wird an der Position des Basisobjektes die Schulterposition (φ_P und φ_R im Konfigurationsraum) zwischengespeichert und diese für das Absetzen des Stapelobjektes wieder angefahren. Je nach Position des Basisobjektes führt dies zu Versatzfehlern, bedingt durch die Morphologie des Roboterarmes. Da bei dieser Variante keine visuellen Informationen benötigt werden, ist der Einsatz von Offsets im

⁴⁴Basierend auf einer definierten Rangfolge werden die Kriterien sequentiell, wie bei einem einfachen Entscheidungsbaum, untersucht. Sobald ein Kriterium einen Treffer erzielt, gilt dies als beste Wahl und die Suche wird abgebrochen.

Kamerabild zur Korrektur nicht möglich. Allerdings handelt es sich auch hier um einen systematischen Fehler, sodass optional Offsetkarten – statt für Korrekturen der visuellen Informationen – auch für Korrekturen der Schulterpositionen im Konfigurationsraum eingesetzt werden können. Im Gegensatz zur visuellen Korrektur werden die Offsets hier nicht kontinuierlich aus der Offsetkarte, sondern einmalig auf die gespeicherte Position des Basisobjektes angewandt.

Die Fehleranalyse beim Stapeln von Objekten kann sowohl während des Abstellens (online), als auch im Anschluss (offline) durchgeführt werden. Für die Online-Korrektur bietet sich das sogenannte *Learning from Demonstration* an. Dazu wird die genaue Position beim Abstellen des Objektes durch den Menschen demonstriert. Der Roboter schaltet die Ansteuerung der Armgelenke dafür in den Kooperationsmodus der gültigen Quadrik, wie er in Kapitel 7.2 eingeführt wurde.

8.4 Auswertung der Zeichenergebnisse

Beim Nachzeichnen von Linien wird die Vorlagelinie mittels eines modifizierten *Harris*-Eckendetektors analysiert und die Enden der Linie als absolute Positionen im vergrößerten Bildausschnitt gemerkt. Diese Positionen werden beim Nachzeichnen auf dem vergrößerten Zeichenbereich wieder angefahren und beim Zeichnen linear interpoliert. Die Analyse wird direkt auf die Vorlagelinie angewandt. Beim Zeichnen werden diese Koordinaten jedoch nicht genau auf die Stiftspitze übertragen, sondern auf die farblich abgesetzte Markierung am Stift (siehe Abbildung 86) oberhalb der Stiftspitze. Der resultierende Versatz ist abhängig von der jeweiligen Position der Hand, kann aber zuverlässig über Offsetkarten korrigiert werden. Im Gegensatz zur Optimierung des Stapelns von Objekten kann hier eine Korrektur vollautomatisch erfolgen: Der Roboter kann nach dem Zeichnen mit dem *Harris*-Eckendetektor, der zuvor schon bei der Analyse der Vorlagelinie (vergleiche Abbildung 55 auf Seite 95) angewandt wurde, die selbst gezeichnete Linie analysieren. Die Abweichung von der Vorlage wird dann zum Wert an der jeweiligen Position in der Offsetkarte addiert. Dazu müssen zuvor die beiden Gelenkpositionen für Beginn und Ende der Linie zwischengespeichert werden. Die Korrektur erfolgt immer nach dem Zeichnen, also offline. Abbildung 84 auf Seite 152 zeigt die Offsetkarte nach einer automatischen und einmaligen Korrektur. Die Vorlagelinien aus dem entwicklungsdiagnostischen Test sind nur ca. 35 mm lang. Die Korrektur von Anfangs- und Endpunkt erfolgt daher gegebenenfalls bei zwei benachbarten Unterteilungen in den Offsetkarten. Für längere Linien empfiehlt es sich, die Korrektur linear zu interpolieren, wie dies auch beim Zeichnen der Linien erfolgt.

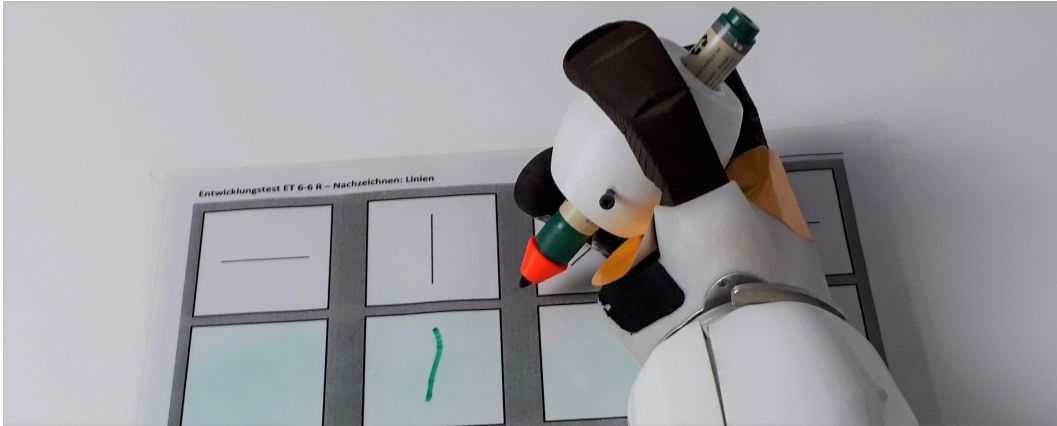


Abbildung 86: Nachgezeichnete vertikale Linie. Die Positionierung erfolgt anhand des orangen Ringes an der Stiftspitze.

8.5 Zusammenfassung

Die Offsetkarten sind ein einfaches aber wirkungsvolles Mittel, das Ergebnis der Aufgaben zu verbessern. Die dynamische Anpassung der Korrekturwerte beim Greifen und Nachzeichnen kann grundsätzlich vollständig autonom durch den Roboter erfolgen. Mit zunehmender Erfahrung werden die Abläufe schrittweise optimiert. Beim Stapeln von Objekten ist der Roboter für die zuverlässige Bewertung der Prozesse noch auf den Menschen angewiesen; unterstützend wirkt sich jedoch aus, dass die Positionskorrektur des Stapelobjektes nur innerhalb des durch die Quadrik vorgegebenen Bereiches möglich ist. Abhängig von der gültigen Quadrik variieren auch die Offsetkarten. Da es sich um unterschiedliche Domänen handelt, in denen die verschiedenen Handlungen ausgeführt werden, kann nicht eine Offsetkarte für alle drei Handlungen genutzt werden. Beim Greifen und Stapeln in den visuell basierten Varianten korrigieren die Offsetkarten die Position der visuellen Wahrnehmung. Beim Stapeln, basierend auf der Position des Basisobjektes im Konfigurationsraum, erfolgt auch die Korrektur bei der Positionierung des Stapelobjektes im Konfigurationsraum. Beim Nachzeichnen werden die visuell wahrgenommenen Fehler ebenfalls für die Korrektur im Konfigurationsraum eingesetzt.

9 Übersicht der heuristischen Verhaltensregelung

Dieses Kapitel bietet eine Übersicht über die heuristische Verhaltensregelung, die in dieser Arbeit entwickelt wurde. Für eine heuristische Verhaltensregelung bedarf es einfacher und aus ökonomischer Sicht sparsamer Regeln, die Probleme mit minimaler Information lösen – den Heuristiken [Gig08]. Der Einsatz dieser führt zu einem Verhalten, welches durch beobachtbare körperliche Reaktionen des Roboters erkennbar wird [MN07].

In den vorherigen Kapiteln wurde eine Reihe von lokalen⁴⁵ und globalen⁴⁶ sensomotorischen Schleifen eingeführt, bei denen kontinuierlich motorische Korrekturen auf Basis von sensorischen Informationen vorgenommen werden. Auf deliberativer Ebene wurden diese durch Zustandsautomaten, Algorithmen der Bildverarbeitung und implizite Modelle ergänzt. Obwohl Zustandsautomaten und Ablaufsteuerungen in der Robotik sowie in der Informatik ein verbreitetes Mittel zur Implementierung von Heuristiken sind, wurden in dieser Arbeit vor allem sensomotorische Schleifen eingesetzt und beide Varianten in einer hybriden Roboterarchitektur kombiniert. Zu vielen dieser Vorgänge lassen sich Parallelen zu Heuristiken der Verhaltenspsychologie herstellen. Bereits mehrfach eingesetzt wurde die Blickheuristik. Ihr Prinzip ist beim Menschen (zum Beispiel beim Fangen eines Balls) sowie beim Tier (zum Beispiel Hund fängt Frisbee) mehrfach belegt und auch aus dem Alltag gut nachzuvollziehen. Implementiert durch mehrere isolierte sensomotorische Schleifen, erzielt die Blickheuristik auch beim Roboter sehr gute Ergebnisse trotz minimalem Aufwand und wenig Randinformationen. Die Schleifen müssen lediglich die Gelenke so ansteuern, dass der Blickwinkel zu einem Objekt konstant gehalten wird, während sich das Objekt vom Auge entfernt oder sich auf dieses zubewegt. Es handelt sich zudem um eine kognitive Heuristik⁴⁷ [GHP11], da die Blickheuristik evolvierte Fähigkeiten nutzt, das Objekt in der Umwelt bewegt wird, und die sensomotorischen Schleifen nur den Lösungsprozess beschreiben, ohne ein Ziel zu definieren.

Tabelle 8 listet die grundlegenden Heuristiken auf, die für die Ausführung der visomotorischen Handlungen (Kapitel 6) eingesetzt werden. Gemäß der Definition eines Heuristikmodells (siehe Kapitel 2.1 auf Seite 8 ff.), werden jeweils (1) die Verfahrensregeln wie die Heuristik abläuft, (2) die Fähigkeiten auf denen die Heuristik aufbaut und (3) die Art der Probleme die damit gelöst werden können gelistet. Da die Heuristik nicht das Ziel beschreibt, sondern nur den Weg zur Lösung, ist in der rechten Spalte ergänzend aufgeführt für welche Aufgaben diese eingesetzt werden.

⁴⁵in den Körperteilen; durch das reaktive System

⁴⁶über den ganzen Körper verteilt; durch die hybride Architektur

⁴⁷Die Definition von kognitiven Heuristiken kann in Kapitel 2.1 ab Seite 8 nachgeschlagen werden.

Tabelle 8: Übersicht der grundlegenden Heuristiken, die für die Ausführung der visuomotorischen Handlungen eingesetzt werden. Da die Heuristik nicht das Ziel beschreibt, sondern nur den Weg zur Lösung, ist in der rechten Spalte ergänzend aufgeführt für welche Aufgaben diese eingesetzt werden.

Verfahrensregeln	Fähigkeiten	Art der Probleme	Eingesetzt für
<ul style="list-style-type: none"> · Kopf <i>Pitch</i> bewegen, sodass vertikale Objektposition in der Bildmitte · Kopf <i>Yaw</i> bewegen, sodass horizontale Objektposition in der Bildmitte 	<ul style="list-style-type: none"> · Objektposition (x,y) aus Kamerabild extrahieren · Kopfgelenke bewegen (<i>Pitch</i>, <i>Yaw</i>) 	Objekt visuell fixieren/verfolgen	<ul style="list-style-type: none"> · Objekt verfolgen · Objekt greifen · Objekte stapeln · Linie nachzeichnen
<ul style="list-style-type: none"> · Beschleunigung im Kopf durch <i>Roll</i>-Bewegung in vertikaler Achse maximieren 	<ul style="list-style-type: none"> · Kopfeigung zum Gravitationsfeld erfassen · Kopf bewegen (<i>Roll</i>) 	Kopfeigung horizontal halten (<i>Roll</i>)	<ul style="list-style-type: none"> · Objekt verfolgen · Objekte stapeln · Linie nachzeichnen
<ul style="list-style-type: none"> · Kopf <i>Roll</i> bewegen, sodass Unterarm vertikal im Bild liegt 	<ul style="list-style-type: none"> · Unterarmorientierung aus (Objektbreite) Kamerabild extrahieren · Kopf bewegen (<i>Roll</i>) 	Unterarm vertikal im Kamerabild halten	<ul style="list-style-type: none"> · Objekt greifen · Objekte stapeln
<ul style="list-style-type: none"> · Handgelenkdrehung an Position des Unterarms im Gravitationsfeld anpassen (Offset für bewusste Drehungen; absolut zum Gravitationsfeld) 	<ul style="list-style-type: none"> · Lage des Unterarms im Gravitationsfeld erfassen 	Handgelenkdrehung unabhängig von der Armbewegung konstant halten	<ul style="list-style-type: none"> · Objekt greifen · Objekte stapeln · Linie nachzeichnen
<ul style="list-style-type: none"> · Schulter <i>Pitch</i> bewegen, sodass Objekt vertikal in Bildmitte (+ Offset aus Offsetkarte) · Schulter <i>Roll</i> bewegen, sodass Objekt horizontal in Bildmitte (+ Offset aus Offsetkarte) 	<ul style="list-style-type: none"> · Objekt-/Handposition (x,y) aus Kamerabild extrahieren · Arm bewegen (Schulter <i>Pitch</i>, Schulter <i>Roll</i>) 	Objekt/Hand an Objekt heranführen	<ul style="list-style-type: none"> · Objekt greifen · Objekt stapeln (Variante 1) · Linie nachzeichnen
<ul style="list-style-type: none"> · Ellenbogengelenk regeln, sodass der Funktionswert der Quadrik null ergibt (Bewegungsrichtung ist abhängig vom Vorzeichen des Funktionswerts) 	<ul style="list-style-type: none"> · Ellenbogengelenk regeln · Funktionswert einer quadratischen Funktion aus den drei Armgelenken berechnen 	Hand in Ebene parallel zum Tisch bewegen	<ul style="list-style-type: none"> · Objekt greifen · Objekte stapeln · Linie nachzeichnen
<ul style="list-style-type: none"> · Ellenbogen strecken, bis Nulldurchgang im I-Anteil (Kollision) oder bis Quadrik erreicht (Nulldurchgang im Funktionswert) 	<ul style="list-style-type: none"> · I-Anteil des Ellenbogenreglers auslesen · Optional: Funktionswert einer quadratischen Funktion berechnen 	Stapelobjekt auf Basisobjekt abstellen	<ul style="list-style-type: none"> · Objekte stapeln
<ul style="list-style-type: none"> · Schulter <i>Pitch</i> bewegen, sodass gespeicherter <i>Pitch</i>-Winkel gehalten wird (+ Offset aus Offsetkarte) · Schulter <i>Roll</i> bewegen, sodass gespeicherter <i>Roll</i>-Winkel gehalten wird (+ Offset aus Offsetkarte) 	<ul style="list-style-type: none"> · Arm bewegen (Schulter <i>Pitch</i>, Schulter <i>Roll</i>) · Zielposition der Schultergelenke kennen 	Schulterposition halten	<ul style="list-style-type: none"> · Objekte stapeln (Variante 2)

Der „adaptive Werkzeugkasten“, wie ihn Gigerenzer et. al. [GHP11] bezeichnen, enthält weitere Heuristiken mit Parallelen zu dieser Arbeit; die wichtigsten werden im Folgenden kurz aufgeführt und im Kontext erläutert.

Minimalist-Heuristik

Eine Entscheidung kann auf mehreren Kriterien basieren. Diese werden in zufälliger Reihenfolge überprüft; sobald eines der Kriterien erfüllt ist, wird die Handlung beendet. Diese Heuristik wurde beispielsweise beim Schließen der Hand durch Erreichen einer definierten Griffkraft oder durch das vollständige Schließen eingesetzt.

Take-The-Best-Heuristik

Werden die Kriterien der Minimalist-Heuristik anhand Ihrer Validität in absteigender Reihenfolge überprüft, handelt es sich um die *Take-The-Best-Heuristik*. Dies wird beispielsweise bei der Auswahl der Stützstellen mittels *PCA* (siehe Kapitel 7.4) genutzt, indem die Validität von der jüngsten bis zur ältesten Stützstelle definiert ist; sowie bei der empirischen Griffkorrektur. Da der erste Ausreißer die Suche terminiert, sind diese Heuristiken klassische Vertreter aus der Gruppe des *One-Reason-Decision-Making*.

Take-The-Last-Heuristik

Diese Heuristik entspricht im Wesentlichen der *Take-The-Best-Heuristik*, jedoch wird hier zuerst das Kriterium betrachtet, welches im letzten Durchlauf zur Entscheidung geführt hat, unabhängig von der Qualität der vorherigen Entscheidung. Dies wurde bei der Ermittlung einer gültigen Quadrik in Kapitel 7.5 eingesetzt. Erst nach der Überprüfung der zuletzt gültigen Quadrik werden die weiteren bekannten Quadriken durchgegangen. Die Validität ist in diesem Fall anhand des Alters von alt nach jung definiert.

Mangel & Systematische Entwicklung

Liegen sehr viele Quadriken im Speicher des Roboters, wird die Wahl einer gültigen Quadrik zunehmend schwieriger, da immer mehr Datensätze überprüft werden müssen und sich die einzelnen Quadriken weniger unterscheiden. Um den Rechenaufwand zu reduzieren, wird künstlich ein Mangel herbeigeführt, indem Quadriken, die sich wenig unterscheiden, in Ruhephasen des Roboters in ihrer Anzahl reduziert werden. Aus diesem Mangel erfolgt die systematische Entwicklung durch das Ableiten neuer Quadriken. Um diesen Mangel zu erzeugen, werden die vorhandenen Quadriken anhand der Anzahl an Stützstellen,

sowie anhand der Zeitspanne seit der letzten Nutzung durch eine *Tallying*-Heuristik analysiert und aussortiert.

Rekognitionsheuristik

Erfordert die Aufgabe eine Entscheidung zwischen zwei Objekten, von denen entweder nur eines bekannt ist oder mehr Informationen über dieses eine vorliegen als über das andere, wird die Entscheidung zugunsten des bekannteren Objekts ausfallen. Dieses Prinzip wird eingesetzt, wenn sich der Roboter zwischen zwei ähnlichen Quadriken entscheiden muss. Auch wenn rein rechnerisch (anhand der Fehlerberechnung) die „unbekanntere“ Quadrik besser geeignet erscheint (weniger zugrunde liegende Stützstelle), wird die „bekanntere“ Quadrik gewählt (mehr zugrunde liegende Stützstellen).

Versuch und Irrtum (*Trial and Error*)

Die wichtigste und vermutlich auch bekannteste Heuristik ist das Prinzip von Versuch und Irrtum: Es werden so lange zulässige Möglichkeiten ausprobiert, bis eine Lösung gefunden ist. Fehler werden bewusst in Kauf genommen. Dieses Prinzip wird bei jeder Optimierung der visuomotorischen Handlungen in dieser Arbeit angewandt, um möglichst schnell Erkenntnisse zur Handlungsoptimierung zu erlangen.

Je nach Verhaltensmuster werden die Heuristiken über einen Zustandsautomaten umgeschaltet. Alle Prozesse, die kein strukturelles Handlungswissen oder Informationen der visuellen Wahrnehmung benötigen, sind isoliert im reaktiven System implementiert und können vom deliberativen System lediglich inhibiert werden. Alle weiteren visuomotorischen Prozesse sind in letzter Instanz ebenfalls im reaktiven System implementiert, haben jedoch weitere Abhängigkeiten zum deliberativen System, wie beispielsweise die Objektkoordinaten in der Bildebene. Sie sind daher als hybride Prozesse implementiert.

Abbildung 87 fasst dieses Kapitel in einer Übersichtsgrafik zusammen.

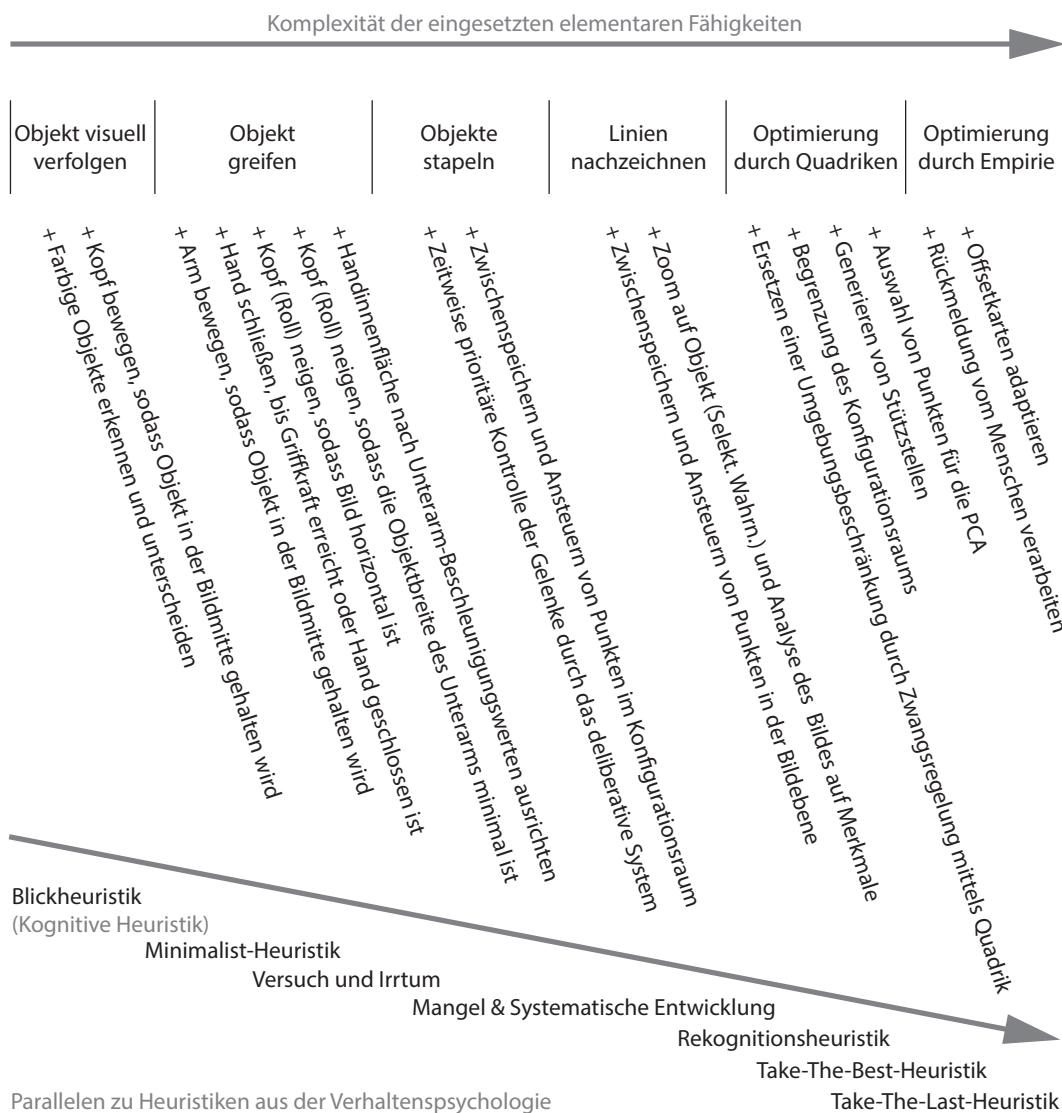


Abbildung 87: Übersicht zum strukturellen Aufbau der heuristischen Verhaltensregelung. Die Komplexität der visuomotorischen Aufgaben sowie der dafür geforderten Fähigkeiten des Roboters nimmt von links nach rechts zu. Unten sind die Heuristiken aus der Verhaltenspsychologie aufgeführt, von denen die Entwicklung der Heuristiken dieser Arbeit inspiriert wurde.

10 Zusammenfassung dieser Arbeit

Diese Arbeit untersucht, ob humanoide Roboter ohne explizite Modelle ihres Körpers oder der Umwelt visuomotorische Handlungen ausführen können. Statt auf Modellen basieren diese Handlungen auf sensomotorischen Schleifen, kombiniert mit heuristischen Strategien. Geleitet vom Prinzip der Einfachheit – alle Komponenten und Abläufe möglichst wenig komplex und zuverlässig zu gestalten – wird der vollständige Prozess von der Weiterentwicklung der Roboterplattform bis zur Selbstoptimierung der Handlungsabläufe an einem realen Roboter abgebildet. Die Anforderungen an die visuomotorischen Fähigkeiten des Roboters wurden aus entwicklungsdiagnostischen Tests für Kinder abgeleitet.

Damit alle Teilaspekte auch in der Praxis einfach und zuverlässig umgesetzt werden, erfolgt die Entwicklung direkt auf einer autonomen Roboterhardware. Dazu wurde der bestehende Roboter Myon weiterentwickelt; die wesentlichen Änderungen beziehen sich auf die Elektronik im Kopf des Roboters. Diese stellt die Rechenkomponenten und *Hardware*-Schnittstellen für den deliberativen Teil der Softwarearchitektur zur Verfügung und erfasst die Daten der exterozeptiven Sensorik. Durch die Neuentwicklung der Programmierumgebung, dessen Anteile im *FPGA* im Rahmen dieser Arbeit entstanden sind, entfallen alle Synthese- und Kompilierprozesse. Die Programmierung und Ausführung des Programmcodes für höhere Verhaltensweisen des Roboters geht seitdem nahtlos ineinander über, und alle sensorischen Daten sowie internen Zustände können jederzeit eingesehen und überwacht werden. Dies hat die Entwicklungszeiten drastisch reduziert. Das neue Taktmanagement im *FPGA* stellt sicher, dass alle Sensoren und Aktoren ohne Datenverlust oder korrupte Werte ausgelesen beziehungsweise angesteuert werden können, da sie phasenstarr synchronisiert sind. Sollte die zentrale Taktquelle ausfallen, wird zur Gewährleistung eines sicheren Betriebs des Roboters auf einen Notfalltakt umgeschaltet und das deliberative System informiert. Die *PixelPipeline* als Schnittstelle zwischen Kamera und Entwicklungsumgebung ist der zentrale sensorische Kanal für alle visuomotorischen Verfahren. Diese wurde bewusst – unabhängig vom gewählten Bildausschnitt – auf eine maximale Auflösung von 512 Pixeln am Ausgang beschränkt. Für eine bestimmte Funktion beziehungsweise Aufgabe muss jeweils zwischen einem großen Bildausschnitt mit niedriger Auflösung und einem kleinen hoch aufgelösten entschieden werden. Dadurch muss das deliberative System nie mehr als 512 Pixel verarbeiten; so werden die Ausführungszeiten begrenzt, was für die Echtzeitfähigkeit des Systems entscheidend ist.

Die Wahrnehmung von Berührungen und Kollisionen erfolgt mittels Analyse des I-Anteils der PI-Gelenkregler; ein Richtungswechsel der Last wird durch einen Vorzeichenwechsel angezeigt. Das macht diesen Parameter besonders attraktiv, da eine Kalibrierung nicht notwendig ist und damit (neben weiteren berücksichtigten Faktoren) das Austauschen von Modulen, beispielsweise eines Arms, problemlos möglich ist.

Bei der visuellen Wahrnehmung wird bewusst auf etablierte Bibliotheken der Bildverarbeitung verzichtet, um das System – dem Grundprinzip dieser Arbeit folgend – so einfach wie möglich zu halten. Die visuelle Wahrnehmung von Objekten erfolgt zunächst anhand von Bewegungen und, sofern dem System bekannt, anhand der Objektfarbe. Dazu werden alle Objekte erkannt, die aktiv ins Sichtfeld des Roboters gestellt oder wieder entfernt werden, und in einem der acht Objektspeicher abgelegt. Dies entspricht einer Art Kurzzeitgedächtnis. Mit der *PixelPipeline* wird das Bild auf alle acht Objektfarben zeitgleich und parallel zur Übertragung durch die Kamera analysiert. Ist die Auflösung von bis zu 512 Pixeln nicht ausreichend, werden die Farbklassen auf Ortshistogramme in horizontaler und vertikaler Richtung angewandt und nach 40 ms steht das Ergebnis mit technisch maximaler Auflösung zur Verfügung. Diese Erkennung ist zuverlässig, da sie auch teilverdeckte beziehungsweise teilverschattete Objekte erkennt. Die Suche nach Ecken, wie sie für das Nachzeichnen von Linien benötigt wird, erfolgt unabhängig vom betrachteten Bildausschnitt auf 500 Pixel reduziert und hat eine konstante Ausführungszeit. Ist das Bild zu hell, zu dunkel oder benötigt es einen aktuellen Weißabgleich, werden die notwendigen Anpassungen direkt auf der Kamerahardware ausgeführt.

Um die Ansteuerung der Gelenke möglichst einfach zu gestalten, werden komplexere Vorgänge, wie das kraftgeregelte Greifen von Objekten oder die horizontale Ausrichtung des Handgelenks beziehungsweise des Kopfes in der *Roll*-Bewegung dezentral auf den reaktiven Komponenten des Roboters berechnet. Die Ausführung erfolgt unabhängig vom deliberativen System, kann aber durch dieses modifiziert und inhibiert werden.

Durch Reduktion der Gelenkfreiheitsgrade ist es möglich, einfache visuelle Merkmale eines Objekts direkt mit dem korrespondierenden Robotergelenk zu koppeln, wie beispielsweise die Positionsdifferenz zum Mittelpunkt der vertikalen Bildachse mit dem Schulter-*Pitch*-Gelenk. Fehlende Informationen, wie die Tiefenschätzung in der visuellen Wahrnehmung, werden durch die Nutzung von Umgebungsbeschränkungen kompensiert, wie beispielsweise die Relaxation des Ellenbogens, damit die Roboterhand während der Bewegung auf der Tischoberfläche verbleibt. Bereits mit diesen einfachen Vorgehensweisen können die entwicklungsdiagnostischen Testaufgaben er-

folgreich absolviert werden. Dazu werden empirisch verschiedene Strategien abgeleitet, wie an die jeweilige Aufgabe angepasste sensomotorischen Schleifen, die durch Heuristiken inhibiert oder getriggert werden können.

Um das Gleiten der Hand über die Tischoberfläche langfristig zu vermeiden, werden Quadriken in den Konfigurationsraum des Armes gelegt. Diese quadratischen Funktionen bilden ein implizites Modell, welches Informationen über den Zusammenhang zwischen der Position der Roboterhand und den Umweltstrukturen bietet. Sie können beispielsweise vorgeben, wie die Gelenke positioniert werden müssen, damit die Hand während der Bewegung in einem definierten Abstand parallel zur Tischoberfläche gehalten wird. Dabei hat sich entsprechend der einschlägigen Literatur gezeigt, dass das *Recursive-Least-Squares*-Verfahren (*RLS*) schneller als das Gradientenabstiegsverfahren mit der mathematisch idealen Lösung des *Least-Squares*-Verfahrens konvergiert. Aus praktischer Sicht wird allerdings das Gradientenabstiegsverfahren bevorzugt, da dieses mit einer akzeptablen Anzahl an Iterationen ausreichend genaue und vor allem zuverlässigere Quadriken für die Bereiche außerhalb der bekannten Positionen im Konfigurationsraum liefert sowie auch den gewünschten Effekt im Aktuationsraum besser extrapoliert. Diese impliziten Modelle werden während der Aufgabenausführung gelernt und optimiert. Die eingesetzten Verfahren erfordern weder A-priori-Wissen noch kalibrierte Sensordaten. Da eine Armquadrik durch einen Vektor mit nur zehn Einträgen definiert ist, belegt jedes Modell verschwindend wenig Speicher: auf der eingesetzten 32-Bit-Architektur nur 40 Byte je Quadrik.

Gerade beim Nachzeichnen von Linien, aber auch beim Greifen und Stapeln von Objekten, hängt die Genauigkeit der Resultate von der Position der Hand im Aktuationsraum ab. Aufgrund der ortsfesten und in drei Achsen schwenkbaren Kamera entstehen Parallaxefehler in der Positionserkennung. Da die Ausrichtung des Roboterkopfes durch die sensomotorische Kopplung bei gleicher Objektposition immer gleich ist, sind diese Fehler systematisch und können kompensiert werden. Die Ausrichtung des Roboterkopfes ist durch einen Punkt im Konfigurationsraum der drei Halswinkel eindeutig definiert, wie die Objektposition durch einen Punkt im Aktuationsraum eindeutig definiert ist. Dazu wird beim Greifen und Stapeln von Objekten eine visuelle Analyse durchgeführt und die Rückmeldung vom Menschen (*Learning from Demonstration*) berücksichtigt, um ein adaptives Offsetgitter im Arbeitsbereich aufzubauen und mit der aktuell gültigen Quadrik zu verknüpfen. Beim Nachzeichnen erfolgt diese Korrektur vollständig autonom, indem die Merkmale der gezeichneten Linie mit den Merkmalen der Vorlage verglichen werden. Eine ORDM-Heuristik unterstützt den Roboter dabei, aus einer laufenden Bewegung gültige oder

zumindest ähnliche Quadriken aus seinem „Gedächtnis“ zu identifizieren. Ist dort keine gültige Quadrik vorhanden, werden die Gewichte der ähnlichsten Quadrik als initiale Gewichte für eine dynamische Optimierung einer neuen Quadrik gesetzt. Die Optimierung erfolgt solange bis ihre Gültigkeit gegeben ist.

Die in dieser Arbeit entwickelten und eingesetzten Verfahren benötigen keine hochauflösenden visuellen Sensoren oder präzise Hardwareaufbauten. Sie sind weitestgehend auf andere Roboterplattformen übertragbar und funktionieren dank der farb-basierten visuellen Objekterkennung auch recht zuverlässig bei teilweiser Verdeckung des Objekts. Variationen in der Belichtungssituation können über die aktive Kompensation der Kameraempfindlichkeit korrigiert werden. Erfolgt eine Störung durch Objekte gleicher Farbe, kann diese die sensomotorischen Prozesse zwar vorübergehend dominieren; erfolgt die Störung jedoch nur kurzzeitig, wird das Verfahren im Anschluss fortgesetzt. Besonders beim Nachzeichnen von Linien erfolgt die visuelle Wahrnehmung fokussiert auf das Aufgabenblatt; Störungen können dort unmittelbar und einfach durch eine Inhibition der Farbwahrnehmung in den grob aufgelösten Vollbildern unterbunden werden.

11 Ausblick

Zur Entwicklung der heuristischen Verhaltensregelung wurden geeignete Aufgabenstellungen identifiziert, ein bestehender humanoider Roboter weiterentwickelt und der Erfolg der Konzepte nach dem Prinzip der Einfachheit auf der Roboterhardware verifiziert.

Während zum Zeitpunkt der Neuentwicklung des *BrainModules* der eingesetzte *SoC* noch ganz neu auf dem Markt war, gibt es zum Zeitpunkt der Abgabe dieser Arbeit bereits viele alternative Module, die sich für diesen Anwendungsfall sehr gut eignen. Besonders deutlich ist der Trend, zunehmend mehr parallele Recheneinheiten in einem Modul zu vereinen, sowie die Entwicklung von Modulen mit dedizierten Einheiten zur Berechnung von künstlichen Neuronalen Netzen. Ungeachtet der herausragenden Fortschritte in der Elektronikentwicklung hat das System mit der existierenden Hardware noch viel Optimierungspotential. Dies betrifft beispielsweise die konsequente Synchronisation der verbleibenden Taktsignale, insbesondere des *SpinalCord*, das Übertragen von Bytecode vom deliberativen System in die Prozessoren der AB3D, sowie die Verbesserung der Nutzerfreundlichkeit und die Entwicklung einer nativ implementierten Quelltextversionierung in der Entwicklungsumgebung (*BDE*).

Zur Weiterentwicklung der Verhaltensregelung sind bereits drei Maßnahmen in Planung. (1) Die Anwendung der Verhaltensweisen auf weitere und komplexere Problemstellungen, (2) die Entwicklung einer umfassenderen Objekterkennung über Farben, Ecken und Kanten hinaus, sowie (3) die Erhöhung der Ausführungsgeschwindigkeit. Derzeit ist die Geschwindigkeit besonders durch die mit nur 2,5 Hz recht langsam ausgeführte Objekterkennung in der visuellen Wahrnehmung beschränkt. Technisch ist die fünffache Erkennungsrate für die Positionsbestimmung von zwei unterschiedlich farbigen Objekten möglich, ohne Änderungen im *FPGA* vornehmen zu müssen. Da dann, abgesehen von der Objektposition, alle anderen visuellen Merkmale wegfallen (siehe Tabelle 3 auf Seite 54), erfordert dies eine umfassende Umstrukturierung des *Frameworks*. Weicht man in bestimmten Punkten vom Bestreben ab, möglichst einfache Lösungen einzusetzen, sind komplexere Verfahren der Objekterkennung zu prüfen, die die Zuverlässigkeit und Flexibilität der Verhaltensweisen zu erhöhen versprechen. Ein interessanter Ansatz ist das lernende Segmentierungsverfahren für gegriffene Objekte vom Boerdijk et. al. [BSDT21], welches zudem robust gegen bewegte Hintergrundobjekte ist, selbst wenn sich das gegriffene Objekt nicht mehr bewegt.

Die Umsetzung und Verifikation der empirischen Optimierung auf der Roboterhardware ist bereits erfolgt. Ausgiebige Experimente, die die Qualität und Geschwindigkeit der Optimierung validieren, stehen bisher noch aus. Hat der Roboter Leerlaufphasen, vergleichbar mit einer Phase der Langeweile beim Menschen, kann der Roboter intrinsisch motiviert bestimmte Bereiche im Aktuationsraum anfahren, um mehr über die Umwelt zu erfahren und die Quadriken zu optimieren. Gerade die Definition von potentiell interessanten Punkten im Aktuationsraum ist hier von Interesse für weitere Entwicklungen.

Erlernt der Mensch neue motorische Fähigkeiten, beginnt dies häufig mit langsamen und verkrampften Bewegungen, die die volle Aufmerksamkeit erfordern. Mit zunehmender Erfahrung werden die Abläufe flüssiger und intuitiver. Gleiches kann auch bei den visuomotorischen Abläufen dieser Arbeit untersucht werden, indem beispielsweise die lineare Interpolation zwischen den Endpunkten der Linie beim Nachzeichnen durch gezielte Impulse auf die Schultergelenke ersetzt wird, vergleichbar mit den Experimenten, die Werner [Wer13] für das Aufrichten eines Pendels durchgeführt hat. Eine weitere wesentliche Voraussetzung dieser Arbeit – und der Grundstein zur Motivation der direkten Kopplung visueller Koordinaten in der Bildebene mit den Armgelenken im Konfigurationsraum – ist die Reduktion der Gelenkfreiheitsgrade. Theoretisch können die Freiheitsgrade bei komplexeren Robotermorphologien künstlich durch Fixierung eines Gelenks oder durch definierte Kopplung mehrerer Gelenke reduziert werden, um die Ansteuerung zu vereinfachen. Wann und wie ausgewählte Gelenke in der Praxis zielführend fixiert oder definiert gekoppelt werden können, ist ebenfalls eine interessante Fragestellung, die aus dieser Arbeit folgt.

Ergänzend dazu sind bereits erste Arbeiten zur Erweiterung der exterozeptiven Wahrnehmung des Roboters erfolgt. Dazu wurden metallische Dehnungssensoren im *AerosolJet*-Verfahren sowie im Mikropipettier-Verfahren gedruckt und getestet, mit dem Ziel, diese direkt in die Verschalung sowie auf die flexiblen Finger der Hand zu drucken. Die ersten Ergebnisse sind in [PH18] veröffentlicht.

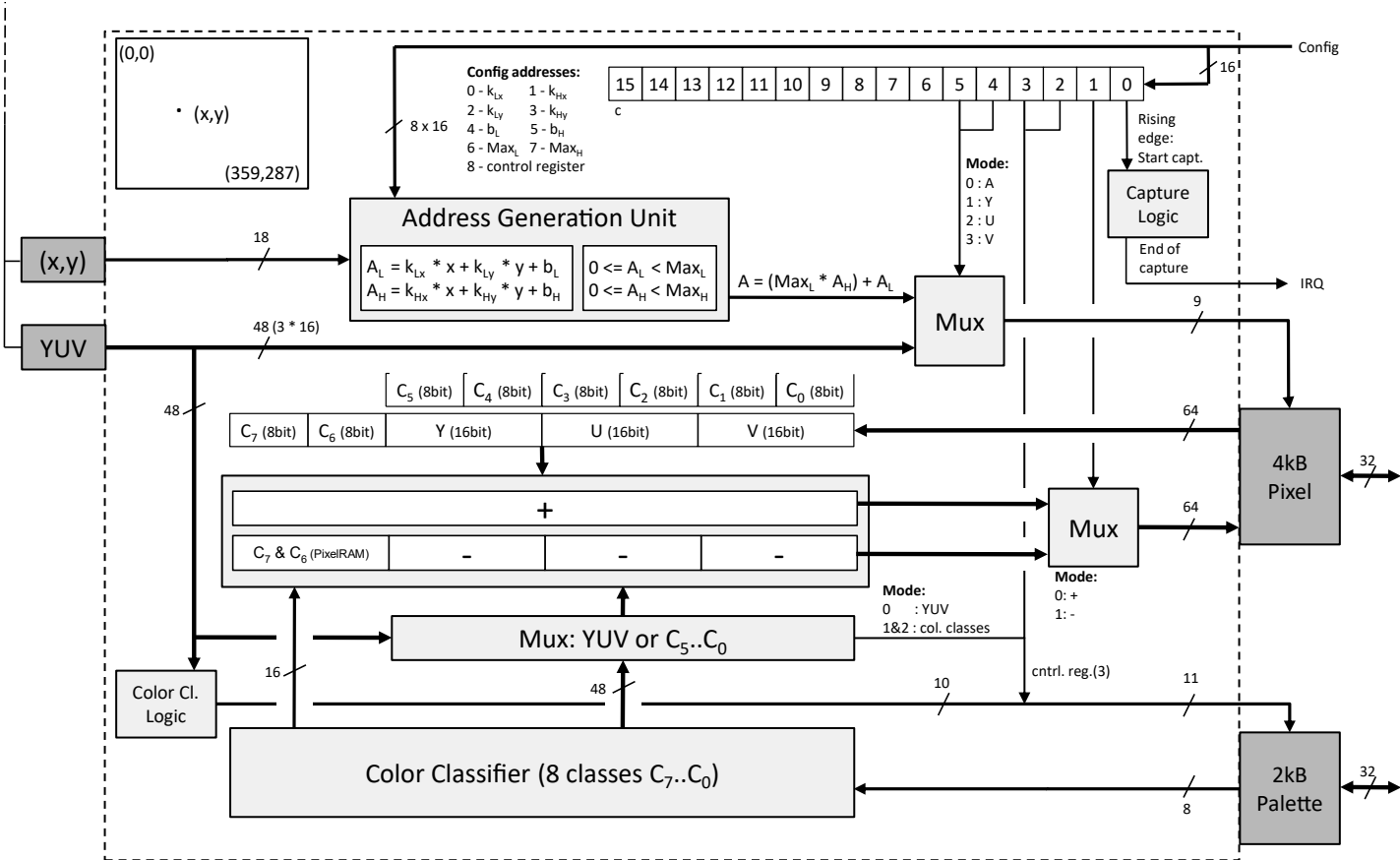


Abbildung 88: Übersichtsgrafik zu den Konfigurationsmöglichkeiten der *PixelPipeline*.

Abmessungen und Gelenkwinkel

Für die Reproduktion der Inhalte dieser Arbeit sind die Abmessungen und Winkelwerte des humanoiden Roboters Myon hilfreich. Die absoluten Winkelwerte sind Richtwerte für den Zusammenbau des Roboters. Eine explizite Kalibrierung ist nicht erfolgt.

Die Abstände zwischen den Gelenken sowie die Längen der Gliedmaßen sind in Tabelle 9 aufgeführt und beziehen sich in der Nummerierung auf Abbildung 42 auf Seite 77.

Tabelle 9: Überblick über die Abstände zwischen den Gelenken sowie die Längen der Gliedmaßen, bezogen auf Abbildung 42.

Nummer	0	1	2	3	4	5	6	7	8
Länge in mm	195,38	10,56	215,00	280,00	60,00	100,61	47,26	44,77	113,32

Die Bewegung der Gelenke ist über mechanische Anschläge begrenzt. Tabelle 10 auf Seite 171 bietet einen Überblick über die Gelenkstellungen bei einem einheitenlosen Winkelwert von null, dem Wert in den mechanischen Anschlägen sowie die ungefähren Gelenkwinkel in Grad. Der maximale Wertebereich des Potentiometers ist auf Werte von -1 bis 1 normiert und bildet einen Winkel von 360° ab.

Tabelle 10: Überblick über die Gelenkstellungen bei einem einheitenlosen Winkelwert von null, dem Winkelwert in den mechanischen Anschlägen sowie die ungefähren Gelenkwinkel in Grad. Die Gelenke im unteren Torso sowie in den Beinen sind nicht aufgeführt.

Körperteil	Gelenk	Pose	Potentiometerwert	Anderer Anschlag	Potentiometer Übersetzung	Anschlag aus Null-Position		Motor Übersetzung
Kopf	Pitch	Kopf waagrecht	0		1:1	35° unten	43° oben	1:1
	Roll	Kopf waagrecht	0		1:1	55° links	55° rechts	1:1
	Yaw	Kopf waagrecht	0		1:1	90° links	90° rechts	1:1
Arm rechts	Schulter Roll	oberer Anschlag	-0,5	0,71	1:0,756	25° innen	119° außen	1:2
	Ellenbogen	oberer Anschlag	0,5	-0,5	1:0,756	127°		1:2
	Handgelenk	Handfläche oben	-1	-1	1:1			1:1
Arm links	Schulter Roll	oberer Anschlag	0,5	-0,71	1:0,756	25° innen	119° außen	1:2
	Ellenbogen	oberer Anschlag	-0,5	0,5	1:0,756	127°		1:2
	Handgelenk	Handfläche oben	1	1	1:1			1:1
Torso	Schulter Pitch (rechts)	unterer Anschlag (hinten)	1	-0,8	1:0,759	180° vorne	55° hinten	1:1,65
	Schulter Pitch (links)	unterer Anschlag (hinten)	-1	0,8	1:0,759	180° vorne	55° hinten	1:1,65
Greifer links		Zeigen	-0,7		1:1			1:1
		Greifen	0,7		1:1			1:1
		Hand offen	0		1:1			1:1
Greifer rechts		Zeigen	0,7		1:1			1:1
		Greifen	-0,7		1:1			1:1
		Hand offen	0		1:1			1:1

Literatur

- [AAHL17] AALTONEN, I. ; ARVOLA, A. ; HEIKKILÄ, P. ; LAMMI, H.: Hello Pepper, May I Tickle You? Children’s and Adults’ Responses to an Entertainment Robot at a Shopping Mall. In: *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. New York, NY, USA : Association for Computing Machinery, 2017 (HRI ’17). – ISBN 9781450348850
- [Agi81] AGIN, G. J.: Fitting Ellipses and General Second-Order Curves. / Carnegie-Mellon Univ Pittsburgh PA Robotics Inst. 1981. – Forschungsbericht
- [AJO⁺18] ABIODUN, O. I. ; JANTAN, A. ; OMOLARA, A. E. ; DADA, K. V. ; MOHAMED, N. A. ; ARSHAD, H.: State-of-the-art in artificial neural network applications: A survey. In: *Heliyon* 4 (2018), Nr. 11, S. e00938
- [ALE] *EU-Projekt ALEAR: Artificial Language Evolution on Autonomous Robots*. <https://portal.ikw.uni-osnabrueck.de/~neurokybernetik/projects/alear.html>, Abruf: 27.07.2021
- [Ana18] ANALOG DEVICES INC.: *Low Power Chip Scale 10-Bit SD/HD Video Encoder ADV7390/ADV7391/ADV7392/ADV7393*. 2018. – Datasheet, Rev J
- [AWB88] ALOIMONOS, J. ; WEISS, I. ; BANDYOPADHYAY, A.: Active vision. In: *International journal of computer vision* 1 (1988), Nr. 4, S. 333–356
- [AWK⁺19] ASFOUR, T. ; WAECHTER, M. ; KAUL, L. ; RADER, S. ; WEINER, P. ; OTTENHAUS, S. ; GRIMM, R. ; ZHOU, Y. ; GROTZ, M. ; PAUS, F.: ARMAR-6: A high-performance humanoid for human-robot collaboration in real-world scenarios. In: *IEEE Robotics & Automation Magazine* 26 (2019), Nr. 4, S. 108–121
- [BBV00] BRUCE, J. ; BALCH, T. ; VELOSO, M.: Fast and inexpensive color image segmentation for interactive robots. In: *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)* Bd. 3 IEEE, 2000, S. 2061–2066

- [BH77] BROMME, R. ; HÖMBERG, E.: *Psychologie und Heuristik: Probleme der systematischen Effektivierung von Erkenntnisprozessen*. Bd. 2. Darmstadt: Steinkopff, 1977
- [Bha19] BHANDARI, M. B. and L. B. and Lee: Haptic identification of objects using tactile sensing and computer vision. In: *Advances in Mechanical Engineering* 11 (2019), Nr. 4, S. 1687814019840468
- [BHS⁺17] BOHG, J. ; HAUSMAN, K. ; SANKARAN, B. ; BROCK, O. ; KRAGIC, D. ; SCHAAL, St. ; SUKHATME, G. S.: Interactive perception: Leveraging action in perception and perception in action. In: *IEEE Transactions on Robotics* 33 (2017), Nr. 6, S. 1273–1291
- [BHW21] BURHAN HAFEZ, M. ; WERMTER, S.: Behavior Self-Organization Supports Task Inference for Continual Robot Learning. In: *arXiv e-prints* (2021), S. arXiv–2107
- [BK19] BILLARD, A. ; KRAGIC, D.: Trends and challenges in robot manipulation. In: *Science* 364 (2019), Nr. 6446
- [BMAK14] BOHG, J. ; MORALES, A. ; ASFOUR, T. ; KRAGIC, D.: Data-driven grasp synthesis—a survey. In: *IEEE Transactions on Robotics* 30 (2014), Nr. 2, S. 289–309
- [BSDT21] BOERDIJK, W. ; SUNDERMEYER, M. ; DURNER, M. ; TRIEBEL, R.: ”What’s This?-Learning to Segment Unknown Objects from Manipulation Sequences. In: *arXiv preprint arXiv:2011.03279v2* (2021)
- [Bun90] BUNDESEN, C.: A theory of visual attention. In: *Psychological review* 97 (1990), Nr. 4, S. 523
- [BVA⁺12] BÖGE, Ch. ; VAHRENKAMP, N. ; ASFOUR, T. ; DILLMANN, R. u. a.: Visual Servoing für ein- und zweiarmige Manipulationsaufgaben bei humanoiden Robotern. In: *Automatisierungstechnik* 60 (2012), Nr. 5, S. 309–317
- [BVPN20] BOTTAREL, F. ; VEZZANI, G. ; PATTACINI, U. ; NATALE, L.: GRASPA 1.0: GRASPA is a robot arm grasping performance benchmark. In: *IEEE Robotics and Automation Letters* 5 (2020), Nr. 2, S. 836–843

- [CG96] CORKE, P. I. ; GOOD, M. C.: Dynamic effects in visual closed-loop systems. In: *IEEE Transactions on Robotics and Automation* 12 (1996), Nr. 5, S. 671–683
- [CR08] CASTELHANO, M. S. ; RAYNER, K. E.: Eye movements during reading, visual search, and scene perception: An overview. (2008)
- [CVB⁺21] CUNHA, G. ; VICENTE, P. ; BERNARDINO, A. ; RIBEIRO, R. ; MORENO, P.: Online Body Schema Adaptation through Cost-Sensitive Active Learning. In: *arXiv preprint arXiv:2101.10892* (2021)
- [CZ98] CROSS, G. ; ZISSERMAN, A.: Quadric Reconstruction from Dual-Space Geometry. In: *ICCV: Proceedings of the Sixth International Conference on Computer Vision*. Washington, DC, USA : IEEE Computer Society, 1998. – ISBN 81–7319–221–9, S. 25
- [Dö01] DÖRNER, D.: *Bauplan für eine Seele*. 2. Auflage. Rowohlt Verlag, 2001
- [Daw79] DAWES, R. M.: The robust beauty of improper linear models in decision making. In: *American Psychologist* Bd. 34, 1979, S. 571—582
- [DMY⁺21] DENG, Sh. ; MEI, F. ; YANG, L. ; LIANG, Ch. ; JIANG, Y. ; YU, G. ; CHEN, Y.: Research on the Hand-eye calibration Method Based on Monocular Robot. In: *Journal of Physics: Conference Series* Bd. 1820 IOP Publishing, 2021, S. 012007
- [DN98] DAI, M. ; NEWMAN, T. S.: Hyperbolic and Parabolic Quadric Surface Fitting Algorithms – Comparison Between the Least Squares Approach and the Parameter Optimization Approach / Computer Science Department, University of Alabama, Huntsville. 1998 (TR-UAH-CS-1998-02). – Forschungsbericht
- [Dör02] DÖRNER, D.: *Die Mechanik des Seelenwagens: eine neuronale Theorie der Handlungsregulation*. (2002)
- [Dud15] DUDEN: *Das Fremdwörterbuch: Band 5*. Bibliographisches Institut GmbH, 2015 (Deutsche Sprache in 12 Bänden). – ISBN 9783411911301

- [EDAR⁺15] EPPNER, C. ; DEIMEL, R. ; ALVAREZ-RUIZ, J. ; MAERTENS, M. ; BROCK, O.: Exploitation of environmental constraints in human and robotic grasping. In: *The International Journal of Robotics Research* 34 (2015), Nr. 7, S. 1021–1038
- [Epp19] EPPNER, C.: *Robot grasping by exploiting compliance and environmental constraints*. Technische Universitaet Berlin (Germany), 2019
- [FBE⁺09] FABER, F. ; BENNEWITZ, M. ; EPPNER, C. ; GOROG, A. ; GONSIOR, Ch. ; JOHO, D. ; SCHREIBER, M. ; BEHNKE, S.: The humanoid museum tour guide Robotinho. In: *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication* IEEE, 2009, S. 891–896
- [FDCBA14] FERNÁNDEZ-DELGADO, M. ; CERNADAS, E. ; BARRO, S. ; AMORIM, D.: Do we need hundreds of classifiers to solve real world classification problems? In: *The journal of machine learning research* 15 (2014), Nr. 1, S. 3133–3181
- [Fit54] FITTS, P. M.: The information capacity of the human motor system in controlling the amplitude of movement. In: *Journal of experimental psychology* 47 (1954), Nr. 6, S. 381
- [Fora] *Forth Inc. - Webseite*. <https://www.forth.com/>, Abruf: 27.07.2021
- [Forb] *Forth Standard*. <https://forth-standard.org/standard/words>, Abruf: 27.07.2021
- [For16] FORSA. POLITIK- UND SOZIALFORSCHUNG GMBH: Service-Robotik: Mensch-Technik-Interaktion im Alltag. (2016). <https://www.bmbf.de/de/service-roboter-statt-pflegeheim-2727.html>
- [GG06] GIGERENZER, G. ; GAISSMAIER, W.: Denken und Urteilen unter Unsicherheit : Kognitive Heuristiken. In: FUNKE, Joachim (Hrsg.): *Denken und Problemlösen*. Göttingen : Hogrefe, 2006 (Enzyklopädie der Psychologie, Themenbereich C, Serie 2, 8), S. 330–374. – ISBN 978-3-8017-0527-5
- [GH97] GARLAND, M. ; HECKBERT, P. S.: Surface simplification using quadric error metrics. In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, S. 209–216

- [GHP11] GIGERENZER, G. ; HERTWIG, R. ; PACHUR, T.: *Heuristics: The Foundations of Adaptive Behavior*. OUP USA, 2011
- [Gig08] GIGERENZER, G.: *Bauchentscheidungen: Die Intelligenz des Unbewussten und die Macht der Intuition*. 6. Auflage. Wilhelm Goldmann Verlag, 2008
- [GWB02] GEGENFURTNER, K. R. ; WALTER, S. ; BRAUN, D. I.: *Visuelle Informationsverarbeitung im Gehirn*. Kopaed Verlag, 2002
- [Hay14] HAYKIN, S.: *Adaptive filter theory*. Pearson Education India, 2014
- [HF98] HALIR, R. ; FLUSSER, J.: *Numerically Stable Direct Least Squares Fitting of Ellipses*. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.7559>. Version:1998
- [HKH11] HILD, M. ; KUBISCH, M. ; HÖFER, S.: Using quadric-representing neurons (QRENS) for real-time learning of an implicit body model. In: *Proceedings of the 11th Conference on Mobile Robot and Competitions*, 2011
- [HLLR19] HOSSEINZADEH, M. ; LI, K. ; LATIF, Y. ; REID, I.: Real-time monocular object-model aware sparse SLAM. In: *2019 International Conference on Robotics and Automation (ICRA) IEEE*, 2019, S. 7123–7129
- [HM16] HESSE, St. ; MALISA, V.: *Taschenbuch Robotik-Montage-Handhabung*. Carl Hanser Verlag GmbH Co KG, 2016
- [HS88] HARRIS, Ch. G. ; STEPHENS, M.: A combined corner and edge detector. In: *Alvey vision conference* Bd. 15 Citeseer, 1988, S. 147–151
- [HSB⁺11] HILD, M. ; SIEDEL, T. ; BENCKENDORFF, C. ; KUBISCH, M. ; THIELE, C.: Myon: Concepts and Design of a Modular Humanoid Robot Which Can Be Reassembled During Runtime. In: *Proceedings of the 14th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 2011
- [IFR20] IFR - INTERNATIONAL FEDERATION OF ROBOTICS: Service-Roboter-Boom: Verkaufszahlen steigen weltweit um 32 %. In: *Pressemittteilung* (2020), Oktober

- [ITU07] ITU - INTERNATIONAL TELECOMMUNICATION UNION: *Interface for digital component video signals in 525-line and 625-line television systems operating at the 4:2:2 level of Recommendation ITU-R BT.601*. 12/2007
- [KBH11] KUBISCH, M. ; BENCKENDORFF, Ch. ; HILD, M.: Balance Recovery of a Humanoid Robot Using Cognitive Sensorimotor Loops (CSLs). In: *Proceedings of the 14th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 2011
- [KC⁺02] KRAGIC, D. ; CHRISTENSEN, H. I. u. a.: Survey on visual servoing for manipulation. In: *Computational Vision and Active Perception Laboratory, Fiskartorpsv* 15 (2002), S. 2002
- [KSM⁺17] KERZEL, M. ; STRAHL, E. ; MAGG, S. ; NAVARRO-GUERRERO, N. ; HEINRICH, St. ; WERMTER, St.: NICO—Neuro-inspired companion: A developmental humanoid robot platform for multimodal interaction. In: *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* IEEE, 2017, S. 113–120
- [LH91] LEE, S. ; HAHN, H.: An optimal sensing strategy for recognition and localization of 3D natural quadric objects. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 13 (1991), Nr. 10, S. 1018–1037
- [LW08] LENARCIC, J. (Hrsg.) ; WENGER, P. (Hrsg.): *Advances in Robot Kinematics: Analysis and Design*. Springer, Netherlands, 2008
- [LWQZ20] LIAO, Z. ; WANG, W. ; QI, X. ; ZHANG, X.: RGB-D object SLAM using quadrics for indoor environments. In: *Sensors* 20 (2020), Nr. 18, S. 5150
- [LZ19] LI, G. ; ZHU, R.: A multisensory tactile system for robotic hands to recognize objects. In: *Advanced Materials Technologies* 4 (2019), Nr. 11, S. 1900602
- [MAH⁺21] MUTHUSAMY, R. ; AYYAD, A. ; HALWANI, M. ; SWART, D. ; GAN, D. ; SENEVIRATNE, L. ; ZWEIRI, Y.: Neuromorphic eye-in-hand visual servoing. In: *IEEE Access* 9 (2021), S. 55853–55870

- [MBC14] MUSIĆ, J. ; BONKOVIĆ, M. ; CECIĆ, M.: Comparison of uncalibrated model-free visual servoing methods for small-amplitude movements: A simulation study. In: *International Journal of Advanced Robotic Systems* 11 (2014), Nr. 7, S. 108
- [MBNRKM13] MICHAELIS, R. ; BERGER, R. ; NENNSTIEL-RATZEL, U. ; KRÄGELOH-MANN, I.: Validierte und teilvalidierte Grenzsteine der Entwicklung. In: *Monatsschrift Kinderheilkunde* 161 (2013), Nr. 10, S. 898–910
- [MCB99] MALIS, E. ; CHAUMETTE, F. ; BOUDET, S.: 2 1/2 D visual servoing. In: *IEEE Transactions on Robotics and Automation* 15 (1999), Nr. 2, S. 238–250
- [MCTDK16] MAC, T. T. ; COPOT, C. ; TRAN, D. T. ; DE KEYSER, R.: Heuristic approaches in robot path planning: A survey. In: *Robotics and Autonomous Systems* 86 (2016), S. 13–28
- [MD96] MCLEOD, P. ; DIENES, Z.: Do fielders know where to go to catch the ball or only how to get there? In: *Journal of experimental psychology: human perception and performance* 22 (1996), Nr. 3, S. 531
- [MF00] MICHALEWICZ, Z. ; FOGEL, D. B.: *How to Solve It: Modern Heuristics*. Berlin: Springer, 2000. – ISBN 978-3-662-04133-8
- [Mil13] MILFORD, M.: Vision-based place recognition: how low can you go? In: *The International Journal of Robotics Research (ijrr)* (32-7 (2013)), S. 766—789
- [MM15] MIGUEL MORALES, P. d.: *VHDL-Based System Design of a Cognitive Sensorimotor Loop (CSL) for Haptic Human-Machine Interaction (HMI)*. Beuth-Hochschule für Technik Berlin, Bachelorarbeit, 2015
- [MN07] MICHEL, Ch. ; NOVAK, F.: *Kleines Psychologisches Wörterbuch: das Standardwerk*. Verlag Herder, 2007
- [Mor80] MORAVEC, H.: *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover / Carnegie Mellon University*. Pittsburgh, PA, September 1980. – Forschungsbericht
- [MP43] MCCULLOCH, W. S. ; PITTS, W.: A logical calculus of the ideas immanent in nervous activity. In: *The bulletin of mathematical biophysics* 5 (1943), Nr. 4, S. 115–133

- [MW06] MERZIGER, G. ; WIRTH, Th.: *Repetitorium der höheren Mathematik*. Binomi, 2006
- [NC21] NATALE, L. ; CANNATA, G.: Tactile sensing. In: *arXiv preprint arXiv:2105.05089* (2021)
- [NMS18] NICHOLSON, L. ; MILFORD, M. ; SÜNDERHAUF, N.: Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. In: *IEEE Robotics and Automation Letters* 4 (2018), Nr. 1, S. 1–8
- [Noë04] NOË, A.: *Action in perception*. MIT press, 2004
- [OLC19] OLIVER, G. ; LANILLOS, P. ; CHENG, G.: Active inference body perception and action for humanoid robots. In: *arXiv preprint arXiv:1906.03022* (2019)
- [OSG20] ODEHNAL, B. ; STACHEL, H. ; GLAESER, G.: *The Universe of Quadrics*. Springer Nature, 2020
- [PBJ93] PAYNE, J. W. ; BETTMAN, J. R. ; JOHNSON, E. J.: *The adaptive decision maker*. Cambridge, UK: Cambridge University Press, 1993
- [Pet02] PETITJEAN, S.: A Survey of Methods for Recovering Quadrics in Triangle Meshes. In: *ACM Computing Surveys* 34 (2002), June, Nr. 2, S. 211–262
- [PH18] PANRECK, B. ; HILD, M.: Fully Printed Strain Gauges: A Comparison of Aerosoljet-Printing and Micropipette-Dispensing. In: *International Journal of Electronics and Communication Engineering* 12 (2018), Nr. 9, S. 678–684
- [PM13] PETERMANN, F. ; MACHA, Th.: *Entwicklungstest für Kinder von sechs Monaten bis sechs Jahren-Revision (ET 6-6-R)*. Pearson Assessment & Information GmbH, 2013
- [Pri93] PRIEDITIS, A. E.: Machine discovery of effective admissible heuristics. In: *Machine learning* 12 (1993), Nr. 1-3, S. 117–141
- [RK10] RAHAYEM, M. R. ; KJELLANDER, J. A. P.: Quadric segmentation and fitting of data captured by a laser profile scanner mounted on an industrial robot. In: *The International Journal of Advanced Manufacturing Technology* (2010)

- [RO03] RIESKAMP, J. ; OTTO, P. E.: How people learn to select strategies. (2003)
- [RSGD19] ROSELL, J. ; SUÁREZ, R. ; GARCÍA, N. ; DIN, M. U.: Planning grasping motions for humanoid robots. In: *International Journal of Humanoid Robotics* 16 (2019), Nr. 06, S. 1950041
- [Sch17] SCHRÖDER, D.: *Intelligente Verfahren*. Springer, 2017
- [Sel05] SELIG, J. M.: *Geometric Fundamentals of Robotics*. Springer, New York, 2005
- [SH12] STEELS, L. ; HILD, M.: *Language Grounding in Robots*. New York : Springer, 2012
- [Shi94] SHI, J.: Good features to track. In: *1994 Proceedings of IEEE conference on computer vision and pattern recognition* IEEE, 1994, S. 593–600
- [Sie15] SIEDEL, T.: *Hybride Steuerung parallel gekoppelter Aktoren am Beispiel des humanoiden Roboters Myon*, Institut für Informatik, Humboldt-Universität zu Berlin, Diss., 2015
- [SKIH07] SHIOMI, M. ; KANDA, T. ; ISHIGURO, I. ; HAGITA, N.: Interactive Humanoid Robots for a Science Museum. In: *IEEE Intelligent Systems* 22 (2007), Nr. 2, S. 25–32. <http://dx.doi.org/10.1109/MIS.2007.37>. – DOI 10.1109/MIS.2007.37
- [SMB00] SCHMID, C. ; MOHR, R. ; BAUCKHAGE, Ch.: Evaluation of interest point detectors. In: *International Journal of computer vision* 37 (2000), Nr. 2, S. 151–172
- [SNAC13] SANTAMARIA-NAVARRO, A. ; ANDRADE-CETTO, J.: Uncalibrated image-based visual servoing. In: *2013 IEEE International Conference on Robotics and Automation* IEEE, 2013, S. 5247–5252
- [SSVO10] SICILIANO, B. ; SCIavicco, L. ; VILLANI, L. ; ORIOLO, G.: *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010
- [STR] *ST Robotics - Herstellerwebseite zu Roboforth*. <http://www.strobotics.com/roboforth.htm>, Abruf: 27.07.2021

- [SVS20] STEWART, E. ; VALSECCHI, M. ; SCHÜTZ, A.: A review of interactions between peripheral and foveal vision. In: *Journal of vision* 20 (2020), Nr. 12, S. 2–2
- [TAK20] TOKUDA, F. ; ARAI, Sh. ; KOSUGE, K.: Neural Network based Visual Servoing for Eye-to-Hand Manipulator. (2020)
- [Tel75] TELLER, E.: *Wegweiser durch die drei Reiche der Natur für Lehrende und Lernende*. Bd. 1. Spamer, 1875
- [Thi14] THIELE, C.: *Design der verteilten Echtzeit-Systemarchitektur DISTAL und Implementierung am Beispiel des humanoiden Roboters Myon*. Humboldt-Universität zu Berlin, Institut für Informatik, Diplomarbeit, 2014
- [TK20] TRETTNER, Ph. ; KOBELT, L.: Fast and Robust QEF Minimization using Probabilistic Quadrics. In: *Computer Graphics Forum* Bd. 39 Wiley Online Library, 2020, S. 325–334
- [TT10] THIBAUT, J.-P. ; TOUSSAINT, L.: Developing motor planning over ages. In: *Journal of experimental child psychology* 105 (2010), Nr. 1-2, S. 116–129
- [TWJ⁺14] TORTA, E. ; WERNER, F. ; JOHNSON, D. O. ; JUOLA, J. F. ; CUIJPERS, R. H. ; BAZZANI, M. ; OBERZAUCHER, J. ; LEMBERGER, J. ; LEWY, H. ; BREGMAN, J.: Evaluation of a small socially-assistive humanoid robot in intelligent homes for the care of the elderly. In: *Journal of Intelligent & Robotic Systems* 76 (2014), Nr. 1, S. 57–71
- [Vid A] VIDEOLOGY IMAGING SOLUTIONS INC.: *3.6mm M-12 Lens 32S3620N*. 04/16/15 PDS-32S3620N Rev A
- [Vid B] VIDEOLOGY IMAGING SOLUTIONS INC.: *1/4CCD High Resolution Color Board Camera With 8-Bit DIGITAL YUV Output*. 07/24/13 PDS-20K45XDIG Rev B
- [Vid C] VIDEOLOGY IMAGING SOLUTIONS INC.: *20K45XDIG/21K45XDIG (Color) 20K35XDIG/21K35XDIG (B&W)*. 07/25/2013 APN 20/21K45XDIG Rev C
- [VJ01] VIOLA, P. ; JONES, M.: Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE com-*

- puter society conference on computer vision and pattern recognition. CVPR 2001* Bd. 1 Ieee, 2001, S. I-I
- [VPPN18] VEZZANI, G. ; PATTACINI, U. ; PASQUALE, G. ; NATALE, L.: Improving Superquadric Modeling and Grasping with Prior on Object Shapes. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2018, S. 6875–6882
- [VWA⁺08] VAHRENKAMP, N. ; WIELAND, St. ; AZAD, P. ; GONZALEZ, D. ; ASFOUR, T. ; DILLMANN, R.: Visual servoing for humanoid grasping and manipulation tasks. In: *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots* IEEE, 2008, S. 406–412
- [WADL⁺20] WANG, H. ; ABU-DAKKA, F. J. ; LE, T. N. ; KYRKI, V. ; XU, H.: A Novel Design of Soft Robotic Hand with a Human-inspired Soft Palm for Dexterous Grasping. In: *arXiv preprint arXiv:2009.00979* (2020)
- [Wer13] WERNER, B.: *Entwicklung eines adaptiven sensomotorischen Algorithmus zur dynamischen Bewegungssteuerung autonomer Roboter*. Humboldt-Universität zu Berlin, Institut für Informatik, Diplomarbeit, 2013
- [WHB96] WILSON, W. J. ; HULLS, C. W. ; BELL, G.: Relative end-effector control using cartesian position based visual servoing. In: *IEEE Transactions on Robotics and Automation* 12 (1996), Nr. 5, S. 684–696
- [WSN87] WEISS, L. ; SANDERSON, A. ; NEUMAN, C.: Dynamic sensor-based control of robots with visual feedback. In: *IEEE Journal on Robotics and Automation* 3 (1987), Nr. 5, S. 404–417
- [Wur16] WURST, J.: *Entwicklung und Implementierung visuomotorischer Verfahren zur Bewegungsverfolgung, Entfernungsschätzung und Objekterkennung*. Beuth Hochschule für Technik Berlin, Bachelorarbeit, 2016
- [ZJZ21] ZHANG, S. ; JI, H. ; ZHANG, H.: Adaptive IBVS and Force Control for Uncertain Robotic System with Unknown Dead-zone Inputs. In: *International Journal of Control, Automation and Systems* (2021), S. 1–10

- [ZM18] ZAFAR, M. N. ; MOHANTA, J. C.: Methodology for path planning and optimization of mobile robots: A review. In: *Procedia computer science* 133 (2018), S. 141–152

Danksagung

Ich bedanke mich ganz herzlich bei all denen, die mich bei meiner wissenschaftlichen Arbeit und bei der Entstehung dieser Dissertationsschrift unterstützt haben. Mein ganz besonderer Dank gilt Prof. Dr. M. Hild. Als wissenschaftlicher Mitarbeiter in seiner Forschungsgruppe ließ er mir große Freiheiten zur selbstbestimmten Arbeit, stand jedoch bei Fragen jederzeit unterstützend zur Verfügung.

Die Tätigkeit im Forschungslabor Neurorobotik an der Berliner Hochschule für Technik ermöglichte es mir, am humanoiden Roboter Myon zu forschen, mir einen weitreichenden Einblick in umliegende Forschungsfelder zu verschaffen und umfangreiche Lehrerfahrungen in diversen technischen Fächern aufzubauen. Insbesondere der intensive Austausch mit den Teamkollegen, allen voran S. Untergasser, sowie mit den anderen Professoren und Forschungsgruppen hat mich regelmäßig zu neuen Ansätzen inspiriert, wofür ich sehr dankbar bin.

Abschließend gilt mein Dank meiner Familie, die mich unermüdlich unterstützt und motiviert hat.

Selbstständigkeitserklärung

Name: Panreck
Vorname: Benjamin

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.

Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

Datum: _____ Unterschrift: _____