universitätbonn ais
Rheinische Friedrich-Wilhelms-Universität Bonn
Insitiut für Informatik VI
Autonome Intelligente Systeme

Diplomarbeit

Lernen von Greifbewegungen aus Imitation und eigener Erfahrung

Vorgelegt von Kathrin Gräve

am 15. November 2009

Betreuer	**Jörg Stückler**
Erstgutachter	**Prof. Dr. Sven Behnke**
Zweitgutachter	**Prof. Dr. Stefan Wrobel**

Kurzfassung

Menschen müssen fast alle ihre Fähigkeiten erlernen. Nur sehr wenige Fähigkeiten sind angeboren, so dass Lernverfahren eine zentrale Rolle in der menschlichen Entwicklung spielen. Dabei nutzen Menschen verschiedene Arten des Lernens, wobei das Lernen durch Imitation und das Lernen aus Erfahrung zu den Wichtigsten gehören. Oftmals wird nicht nur ein Lernverfahren eingesetzt, sondern eine Kombination aus mehreren Verfahren, um einen schnelleren Lernerfolg zu erreichen.

Genau diesen Vorteil möchte sich auch die Robotik zunutze machen, um zeitaufwendige Programmierung zu ersetzen und die Interaktion zwischen Mensch und Roboter zu vereinfachen. Daher wird im Rahmen dieser Diplomarbeit ein interaktives Lernverfahren entwickelt, dass die Vorteile von Imitations- und verstärkendem Lernen kombiniert, um anthropomorphe Bewegungen zu lernen. Exemplarisch wird dafür die Aufgabe, ein Objekt an einer beliebigen Position auf einem Tisch zu greifen, untersucht. Um die Bewegungen zu generieren, wird ein Regler entworfen, dessen Parameter von dem vorgestellten Verfahren gelernt werden. Zur Repräsentation und Generalisierung gelernter Parametersätze wird Regression mit Gauß-Prozessen eingesetzt. Dies ermöglicht es dem Roboter, in jeder Situation zu entscheiden, ob er über genügend Wissen verfügt, um eine gute Bewegung zu generieren. Ist dies der Fall, versucht der Roboter, die gelernte Bewegung durch verstärkendes Lernen weiter zu verbessern. Andernfalls wird die Bewegung durch das Imitieren einer menschlichen Bewegung gelernt.

Mit diesen integrierten Ansatz werden die Vorteile beider Verfahren kombiniert. Durch Imitationslernen erworbene Informationen fokussieren die Suche des verstärkenden Lernens. Das verstärkende Lernen hilft, die Anzahl benötigter Demonstrationen klein zu halten.

Inhaltsverzeichnis

1 Einleitung

Kapitel

Schon seit vielen Jahren sind Roboter Gegenstand der Forschung. Zu Beginn standen vor allem Industrieroboter im Vordergrund, die dem Menschen schwere, schmutzige und gefährliche Arbeit abnehmen sollten. In den letzten Jahren verlagerte sich der Schwerpunkt der Forschung in Richtung autonomer Roboter, deren Einsatzgebiete sehr unterschiedlich sein können. Dazu gehören Roboter, die in für Menschen gefährlichen Situationen, wie der Feuerbekämpfung, der Bombenentschärfung oder auch beim Retten verschütteter Menschen, eingesetzt werden. Auch bei der Erkundung des Weltraums oder unter Wasser können Roboter in Bereichen arbeiten, die für Menschen nicht zugänglich sind. Ein weiteres wichtiges Einsatzgebiet für die Robotik ist der Servicebereich, in dem Roboter Aufgaben im Haushalt übernehmen oder bei der Pflege älterer oder kranker Menschen behilflich sind.

Menschen stellen sich schon seit vielen Jahren diese Art von Helfern vor, die ihnen lästige und oft zeitraubende Arbeiten abnehmen oder ihnen ermöglichen, auch im Alter länger selbstständig zu leben. Der erste Serviceroboter tauchte schon in dem Theaterstück „Rossum's Universal Robots“ von Karel Capek im Jahre 1920 auf. Bis heute gibt es eine Reihe weiterer Filme, in denen Roboter ebensogut Aufgaben im Haushalt erfüllen wie der Mensch. So nimmt zum Beispiel in dem Film „Der 200 Jahre Mann“ der Roboter Andrew seinen Besitzern sämtliche Arbeiten im Haushalt ab und serviert Mahlzeiten, die er zuvor selbstständig zubereitet hat.

Bis zur Realisierung dieser Visionen ist es jedoch noch ein weiter Weg. Schon heute existieren Roboter, die Aufgaben im Haushalt übernehmen können. Bis 2007 ist die Anzahl der Serviceroboter in Privathaushalten auf 3 Millionen gestiegen [1]. Momentan erledigen diese Roboter allerdings nur sehr einfache kleine Aufgaben, auf die sie spezialisiert wurden. So gibt es z.B. eine Reihe von Robotern, die dem Menschen Aufgaben wie das Staubsaugen oder das Rasenmähen abnehmen. Länder wie Korea haben die Zukunftsvision, dass bis zum Jahre 2013 in jedem Haushalt ein Roboter zu finden sein wird, der komplexe Aufgaben im Haushalt oder in der Pflege von älteren oder kranken Menschen übernimmt.

Arbeiteten die Roboter in der Industrie alleine und oftmals vom Menschen abgeschirmt, so müssen Haushaltsroboter mit Menschen umgehen können und oftmals sogar mit ihnen zusammenarbeiten. Während die wichtigsten Fähigkeiten der Roboter in der Industrie Stärke, Geschwindigkeit und Genauigkeit sind, ist es im Haushalt der Umgang mit Menschen und das Zurechtfinden in immer neuen Situationen und Umgebungen.

Zu solchen gehören menschliche Wohnumgebungen. Diese hat der Mensch seiner eigenen Gestalt und seinen Bewegungsmöglichkeiten entsprechend angepasst. So ist z.B. die Höhe von Tischen auf die übliche Arbeitshöhe des Menschen ausgerichtet, ebenso wie Türgriffe oder Gegenstände in Regalen bequem im Stehen zu erreichen sind. Die Umgebung soll nicht an einen Roboter angepasst werden müssen, sondern der Roboter an die Welt des Menschen. Somit ist es sinnvoll, die Gestalt und Bewegungsfähigkeit des Roboters ähnlich denen des Menschen zu gestalten. Dies verleiht dem Roboter einen ähnlichen Aktionsradius wie dem Menschen und erlaubt ihm, ähnliche Aufgaben wie dieser zu verrichten. Des Weiteren bieten ein menschenähnliches Verhalten und menschenähnliche Bewegungen den Vorteil, dass den Robotern von Menschen eine höhere Akzeptanz entgegengebracht wird. Sie erleichtern dem Menschen den Umgang mit dem Roboter. Menschen sind an menschliche Bewegungen und Verhaltenweisen gewöhnt und wissen mit diesen umzugehen. Wenn der Mensch z.B. einen vom Roboter entgegengereichten Becher annehmen möchte, so fällt es ihm leichter, wenn der Roboter dies in einer menschlichen Art und Weise tut.

Der Roboter muss in der Lage sein, sich an ständig ändernde Umgebungen und Situationen anzupassen und seine Fähigkeiten erweitern zu können. Da es sehr aufwendig und teilweise schwierig wäre, diese zu programmieren und gegebenenfalls an neue Situationen anzupassen, ist Lernen ein zentrales Thema der Robotikforschung. Dabei ist das Ziel, Lernverfahren zu entwickeln, die keinerlei Vorkenntnisse auf dem Gebiet der Robotik oder der Programmierung seitens des Lehrers voraussetzen. Jeder sollte in der Lage sein, den Roboter zu trainieren. Somit könnte jeder Eigentümer eines Roboters diesem Fähigkeiten passend zu den eigenen Bedürfnissen beibringen und es würde kein Experte benötigt. Besonders einfach fällt das Trainieren eines Roboters, wenn das Lernverfahren dem menschlichen Lernen nachempfunden ist. Somit könnten dem Roboter Fähigkeiten ähnlich wie einem Menschen beigebracht werden. Der Mensch muss somit nicht lernen, auf welche Weise er den Roboter trainieren kann, sondern der Roboter passt sich der menschlichen Lernart an.

Eine wesentliche Grundfähigkeit, damit der Roboter in der menschlichen Umwelt interagieren kann, ist das Greifen von Gegenständen. Diese wird sowohl für einfache Aufgaben, wie das Öffnen einer Tür oder das Holen eines Getränkes, als auch für komplexere Aufgaben, wie z.B. das Einräumen einer Spülmaschine, benötigt.

1.1 Zielsetzung

Im Rahmen dieser Diplomarbeit soll ein Lernverfahren entwickelt werden, mit dem auf einfache und intuitive Weise anthropomorphe Greifbewegungen gelernt werden können. Ein Roboterarm soll nach dem Training einen Becher an einer beliebigen Position und mit beliebiger Orientierung auf einem Tisch greifen können. Beim Training wird ein interaktiver Ansatz verfolgt, so dass keine Vorkenntnisse beim Umgang mit Robotern erforderlich sind. Dazu wird ein kombiniertes Verfahren aus Imitations- und verstärkendem Lernen benutzt. Beides sind Lernverfahren, die auch in der menschlichen Lernpsychologie verwendet werden und für jeden Menschen leicht verständlich und anwendbar sind. Zur Beschreibung der menschlichen Bewegung soll ein Regler entwickelt werden, dessen Parameter den genauen Verlauf der Trajektorie charakterisieren. Das Lernen durch

Imitation liefert hier den zusätzlichen Vorteil, dass die Feinheiten dieser Bewegungen direkt aus der Bewegung eines Menschen extrahiert werden können, und die Schwierigkeit anthropomorphe Bewegungen zu beschreiben entfällt. Um den Aufwand des Vorführens gering zu halten, werden die durch die Imitation bestimmten Trainingsbeispiele als Ausgangspunkt benutzt und selbstständig durch den Roboter mit verstärkendem Lernen verbessert und generalisiert.

1.2 Aufbau der Arbeit

Die wichtigsten Grundlagen, auf denen das entwickelte Lernverfahren aufbaut, werden in Kapitel 2 vermittelt. Dabei wird in Abschnitt 2.1 zunächst auf die Regression mit Gauß-Prozessen eingegangen, bevor in Abschnitt 2.2 verschiedene Ansätze zur Optimierung von Funktionen erläutert werden. Zusätzlich wird in Abschnitt 2.3 der in dieser Diplomarbeit verwendete Roboter beschrieben, dessen anthropomorphe Bauweise eine Voraussetzung für das entwickelte Lernverfahren darstellt. In Kapitel 3 wird zunächst ein Überblick über Arbeiten gegeben, die mit dem im Rahmen dieser Diplomarbeit entwickelten Lernverfahren verwandt sind und den aktuellen Stand der Forschung auf dem Gebiet des Imitations- und des verstärkenden Lernens widerspiegeln. Anschließend wird in Kapitel 4 der ebenfalls in dieser Diplomarbeit entworfene Regler vorgestellt, der anhand gelernter Parameter Greifbewegungen generiert. Das Verfahren, mit dem diese Parameter gelernt werden, wird in Kapitel 5 im Detail erläutert. Dabei wird sowohl auf das kombinierte Lernverfahren, als auch auf seine beiden Einzelkomponenten, das Imitationslernen und das verstärkende Lernen, eingegangen. Ebenfalls werden die Vorteile der Teilkomponenten beschrieben, die sich im kombinierten Verfahren ergänzen. Evaluiert wird das vorgestellte Lernverfahren durch Experimente sowohl in der Simulation als auch auf dem realen Roboter in Kapitel 6. In Kapitel 7 erfolgt eine Zusammenfassung der Diplomarbeit. Außerdem wird der wissenschaftliche Beitrag der Arbeit erläutert und ein Ausblick auf mögliche Verbesserungen des Verfahrens und weitere Anwendungsmöglichkeiten gegeben. Im Anhang befinden sich Informationen zu den genutzten Bewegungserfassungssystemen und Softwarepaketen.

2 Grundlagen

Kapitel

In diesem Kapitel werden die Grundlagen erläutert, auf denen das in Kapitel 5 vorgestellte Lernverfahren basiert. Zunächst wird die Regression mit Gauß-Prozessen beschrieben, die es erlaubt, Kostenwerte von gegebenen Punkten auf den gesamten Eingaberaum zu generalisieren. Anschließend werden verschiedene Optimierungsverfahren vorgestellt, mit denen Extrempunkte von Kostenfunktionen ermittelt werden können. Je nach Art der Kostenfunktion bietet sich dabei ein anderes der in Abschnitt 2.2 beschriebenen Verfahren an. Schließlich wird der anthropomorphe Roboter Dynamaid vorgestellt, mit dem das Lernverfahren entwickelt wurde. Zwar ist der Lernalgorithmus nicht von einem speziellen Roboter abhängig, die anthropomorphe Gestalt des Roboters stellt jedoch eine Grundvoraussetzung für das Verfahren dar, weshalb er hier beschrieben wird.

2.1 Regression mit Gauß-Prozessen

Ziel der Regressionsanalyse ist die Bestimmung eines funktionalen Zusammenhangs zwischen gegebenen Trainingsein- und -ausgaben und die Vorhersage unbeobachteter Funktionswerte. Formal bestehen die Trainingsbeispiele aus Paaren $(\vec{x_i}, y_i)$. Die $\vec{x_i}$ sind dabei Werte einer unabhängigen Variablen, die ein- oder mehrdimensional sein kann. Die y_i sind die dazugehörigen Funktionswerte. Sind diese reellwertig, spricht man von einem *Regressionsproblem*, nehmen die y_i dagegen nur eine endliche Menge von diskreten Werten an, von einem *Klassifikationsproblem* [2].

Bei der Regressionsanalyse geht man davon aus, dass zwischen den Ein- und Ausgabewerten ein funktionaler Zusammenhang f besteht, und versucht, diesen anhand der Daten zu bestimmen. Dies ist jedoch nur möglich, wenn die Menge der zulässigen Funktionen eingeschränkt wird, indem Annahmen über die gesuchte Funktion formuliert werden. Die Schwierigkeit bei der Formulierung dieser Annahmen besteht darin, einen Kompromiss zwischen guter Approximation der Trainingsdaten und hoher Generalisierungsfähigkeit zu finden. Diese beiden gegensätzlichen Anforderungen sind in Abbildung 2.1 dargestellt. Die Formulierung sinnvoller Annahmen setzt Vorwissen über die Daten voraus.

Grundsätzlich gibt es zwei Klassen von Ansätzen, um f zu bestimmen. Deterministische Verfahren liefern ausschließlich eine Schätzung der gesuchten Funktion, während stochastische Verfahren zusätzlich Aussagen über die Unsicherheit der Schätzung erlauben. Im Folgenden wird die Regression mit Bayesscher Inferenz beschrieben, die zur

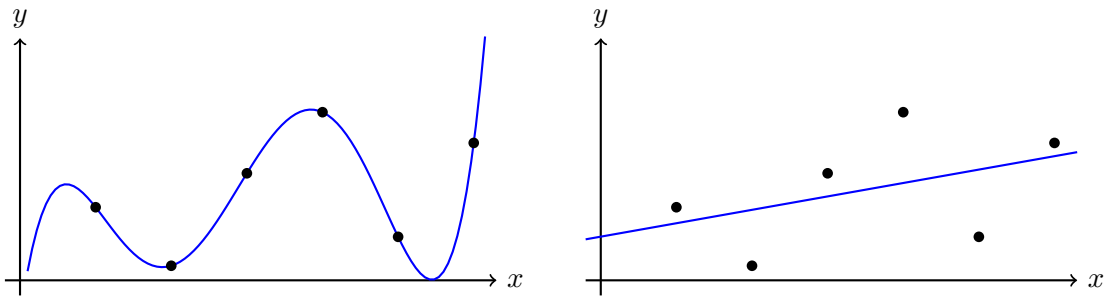


Abbildung 2.1: *Problem der Funktionswahl. Die Punkte stellen gegebene Trainingsdaten dar und die beiden Linien mögliche approximierende Funktionen. Das Polynom auf der linken Seite interpoliert die Trainingsdaten perfekt. Allerdings ist keine gute Generalisierungsfähigkeit auf neue Punkte zu erwarten. Die Gerade auf der rechten Seite verläuft durch keinen der Trainingspunkte. Dafür ist ihre Generalisierungsfähigkeit größer. Gewünscht ist ein Kompromiss aus diesen beiden Extremen.*

zweiten Kategorie gehört. Dafür werden zunächst parametrische Modelle für f betrachtet. Darauf aufbauend wird anschließend die in dieser Diplomarbeit verwendete Regression mit Gauß-Prozess-Modellen erläutert.

2.1.1 Bayessche Regression mit parametrischen Modellen

Eine Möglichkeit, Annahmen über f zu formulieren, ist ausschließlich Funktionen einer bestimmten parametrischen Form zu betrachten. Beispiele solcher *parametrischer Modelle* sind Geraden, Polynome n-ten Grades oder Linearkombinationen radialer Basisfunktionen. Die allgemeine Form eines solchen Modells ist

$$y_i = f(\vec{x_i}; \vec{w}), \tag{2.1}$$

wobei der Vektor $\vec{w}$ die Parameter des Modells enthält. Die Regressionsaufgabe reduziert sich dann auf eine Suche im Parameterraum nach dem optimalen Vektor $\vec{w}$.

Neben diesem deterministischen Zusammenhang zwischen Trainingsein- und -ausgaben, wird bei der stochastischen Inferenz eine weitere, zufällige Komponente berücksichtigt. Dieses sogenannte *Beobachtungsmodell* beschreibt Unsicherheiten in den Beobachtungen, etwa durch Rauschen:

$$y_i = f(\vec{x_i}; \vec{w}) + \varepsilon_i = f(\vec{x_i}; \vec{w}, \vec{\theta}) \tag{2.2}$$

Die Parameter des Beobachtungsmodells werden in dem Vektor $\vec{\theta}$ zusammengefasst. Im Fall eines additiven Rauschens enthält $\vec{\theta}$ dessen Verteilungsparameter. Häufig wird dabei eine Normalverteilung um den Mittelwert 0 angenommen.

Der Bayessche Ansatz geht davon aus, dass auch die gesuchten Parameter $\vec{w}$ mit einer Unsicherheit behaftet und somit stochastische Größen sind. Das erlaubt es, ihre Verteilung zu betrachten. Bevor Trainigsdaten vorliegen, gibt $p(\vec{w})$ die *a priori Wahrscheinlichkeitsverteilung* der Parameter an. Die Trainigsdaten liefern zusätzliche Informationen über die Parameter in Form der *Likelihood* $p(\vec{y} \mid X, \vec{w}, \vec{\theta})$. Beide Verteilungen werden durch

den *Satz von Bayes* zu einer *a posteriori* Verteilung kombiniert, die den Einfluss der Beobachtungen auf die a priori Annahmen beschreibt.

$$p(\vec{w} \mid \vec{y}, X, \vec{\theta}) = \frac{p(\vec{y} \mid X, \vec{w}, \vec{\theta}) \cdot p(\vec{w})}{p(\vec{y} \mid X, \vec{\theta})} \tag{2.3}$$

Dieser Schritt wird auch als *Bayessche Inferenz* bezeichnet. Der Einfachheit halber wurden die Ein- und Ausgaben der Trainigsbeispiele in der Matrix $X = (\vec{x_1}, \ldots, \vec{x_N})^T \in \mathbb{R}^{N \times M}$ und dem Vektor $\vec{y} = (y_1, \ldots, y_N)^T \in \mathbb{R}^N$ zusammengefasst. Im Zähler steht die sogenannte *Randwahrscheinlichkeitsverteilung*, die nach dem *Satz der totalen Wahrscheinlichkeit* durch

$$p(\vec{y} \mid X, \vec{\theta}) = \int p(\vec{y} \mid X, \vec{w}, \vec{\theta}) \cdot p(\vec{w}) \, \mathrm{d}\vec{w} \tag{2.4}$$

gegeben ist. Die a posteriori Verteilung der Parameter erlaubt es, eine Punktschätzung für den gesuchten Vektor $\vec{w}$ zu berechnen, und Aussagen über die Unsicherheit dieser Schätzung zu treffen.

Um Vorhersagen über den Funktionswert an einer unbeobachteten Stelle x^* zu erhalten, betrachtet man die *a posteriori Vorhersagewahrscheinlichkeit* des gesuchten Funktionswertes y^*.

$$p(y^* \mid \vec{x^*}, \vec{y}, X, \vec{\theta}) = \int p(y^* \mid \vec{x^*}, \vec{w}, \vec{\theta}) \cdot p(\vec{w} \mid \vec{y}, X, \vec{\theta}) \, \mathrm{d}\vec{w} \tag{2.5}$$

Auf diese Weise werden die Informationen *aller* möglichen Parametervektoren für die Vorhersage berücksichtigt, nicht nur die eines Einzelnen. Die Parametervektoren werden dabei mit ihrer a posteriori Wahrscheinlichkeit gewichtet. Genau wie bei der Schätzung der Parameter liefert die Verteilung Informationen über die stochastischen Eigenschaften der Vorhersage.

2.1.2 Gauß-Prozess-Regression

Parametrische Modelle haben den Nachteil, dass es häufig schwierig ist, ein sinnvolles Modell zu wählen. Diese Wahl ist entscheidend für die maximal erreichbare Qualität von Schätzungen. Das Ergebnis der Schätzung kann nur so gut sein, wie es das gewählte Modell zulässt. Nicht-parametrische Ansätze versuchen daher, Informationen über die Struktur des Modells ebenfalls aus den Daten abzuleiten.

Zu dieser Art von Verfahren gehört die Regression mit Gauß-Prozessen. Sie vermeidet die explizite Wahl eines parametrischen Modells, indem jeder Funktionswert als Zufallsvariable modelliert wird. Die gemeinsame Verteilung aller Funktionswerte entspricht einer Wahrscheinlichkeitsverteilung über Funktionen. Indem eine bestimmte Form dieser Verteilung angenommen wird, kann der Raum der zulässigen Funktionen eingeschränkt werden, ohne von vorne herein eine bestimmte Klasse von Funktionen kategorisch auszuschließen. Den Funktionen werden lediglich unterschiedliche Wahrscheinlichkeiten zugeordnet. Bei der Gauß-Prozess-Regression wird ein Gauß-Prozess verwendet, um diese Verteilung über Funktionen zu modellieren.

Ein Gauß-Prozess ist ein stochastischer Prozess, d.h. eine indizierte Menge von Zufallsvariablen, mit der Eigenschaft, dass jede endliche Teilmenge normalverteilt ist [3].

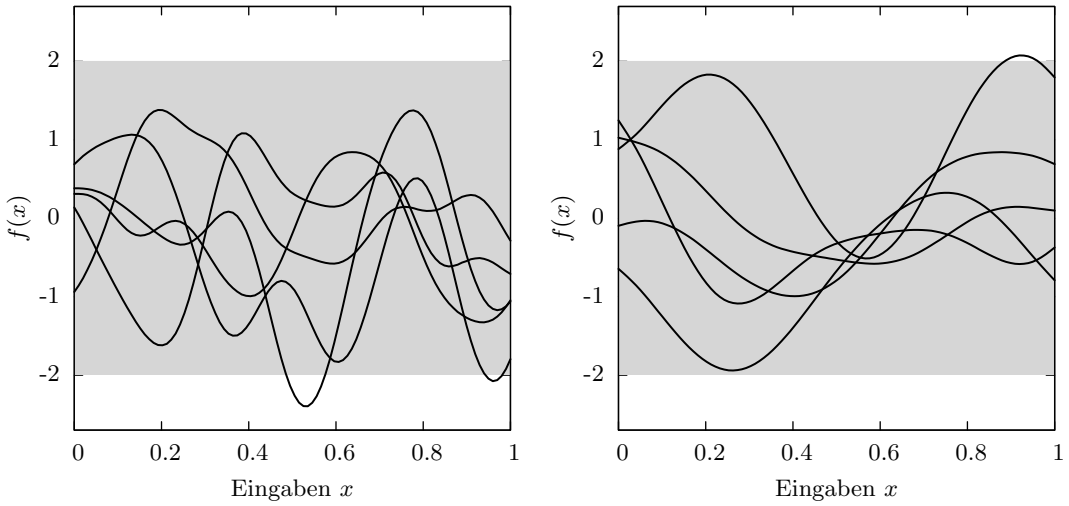


Abbildung 2.2: *Einfluss der Kernelbreite auf den Gauß-Prozess. Die dargestellten Funktionen wurden zufällig aus einem Gauß-Prozess mit der a priori Mittelwertfunktion $\mu(\vec{x}) = 0$ gezogen. Die schattierte Fläche stellt einen Bereich von $\pm 2\sigma$ der einzelnen Punkte dar. A priori ist diese für alle Punkte gleich. Als Kovarianzfunktion wurde eine RBF-Funktion gewählt. Dabei wurde für die rechte Abbildung die Kernelbreite verdoppelt, so dass die Stichproben dort deutlich glatter verlaufen.*

Dies bedeutet insbesondere, dass jeder einzelne Funktionswert und beliebige Paare von Funktionswerten normalverteilt sind. Betrachtet man statt Teilmengen von Funktionswerten die gesamte Funktion, so lassen sich Gauß-Prozesse als eine Verallgemeinerung der mehrdimensionalen Normalverteilung von endlich-dimensionalen Vektorräumen auf unendlich-dimensionale Räume von Funktionen auffassen. Jede Realisierung der Zufallsvariablen des Prozesses liefert anschaulich eine Folge von Funktionswerten.

Eine wichtige Eigenschaft von Gauß-Prozessen ist, dass sie sich vollständig durch eine Mittelwertfunktion $\mu(\vec{x})$ und eine Kovarianzfunktion $k(\vec{x}, \vec{x}')$ beschreiben lassen, genau wie sich eine mehrdimensionale Normalverteilung durch eine Kovarianzmatrix und einen Mittelwert-Vektor charakterisieren lässt. Die Kovarianzfunktion ordnet zwei Funktionswerten an zwei beliebigen Stellen eine Kovarianz zu. Sie kann auf verschiedene Arten gewählt werden. Häufig wird eine radiale Basisfunktion (RBF-Funktion) als Kovarianzfunktion verwendet. Funktionswerte, die im Eingaberaum nah beieinander liegen, bekommen dadurch höhere Kovarianzen zugewiesen, als solche, die weit auseinanderliegen. Die Parameter der Kovarianzfunktion werden als *Hyperparameter* bezeichnet. Durch ihre Wahl lässt sich die Form der Verteilung und damit das Aussehen der Funktionen bestimmen, die der Gauß-Prozess beschreibt. Im Fall der RBF-Funktion ist zum Beispiel die Breite der Glocke ein Hyperparameter. Mit ihm lässt sich die Glattheit der Funktionen bestimmen. Je größer er gewähhlt ist, desto größer ist der Radius im Eingaberaum, in dem die Funkionswerte miteinander korelliert sind, und desto wahrscheinlicher werden glatte Funktionen. Dieser Einfluss der Hyperparameter ist in Abbildung 2.2 veranschaulicht.

Grundlage für die Bayessche Regression mit diesem Modell ist wie in Abschnitt 2.1.1

auf Seite 12 eine a priori Verteilung. Im Unterschied zu 2.1.1 gibt diese die Verteilung der einzelnen Funktionswerte an, und nicht die der Parameter eines zugrundeliegenden parametrischen Modells. Da die gesuchte Funktion durch einen Gauß-Prozess beschrieben wird, sind alle endlichen Teilmengen von Funktionswerten f dieser Funktion normalverteilt. Für die a priori Verteilung der Trainingsbeispiele gilt daher:

$$p(\vec{f} \mid X, \vec{\Psi}) = \mathcal{N}(0, K) \tag{2.6}$$

Dabei enthält der Vektor $\vec{\Psi}$ die Hyperparameter der Kovariabzfunktion des Gauß-Prozesses. Vereinfachend wurde angenommen, dass der Mittelwert 0 ist. Falls entsprechendes Vorwissen vorhanden ist, können ohne Weiteres auch andere Mittelwerte verwendet werden. Die Elemente der Kovarianzmatrix K ergeben sich aus der Kovarianzfunktion:

$$K_{ij} = k(\vec{x}_i, \vec{x}_j)$$

Mit dem Beobachtungsmodell (2.2) ist die Likelihood der Trainingsbeispiele ebenfalls normalverteilt:

$$p(\vec{y} \mid \vec{f}, \vec{\theta}) = \mathcal{N}(\vec{f}, \sigma^2 \cdot I) \tag{2.7}$$

Dabei wird angenommen, dass das additive Rauschen der Beobachtungen unabhängig und normalverteilt mit dem Mittelwert 0 und der Varianz σ^2 ist. Da jeder beobachtete Wert y_i nur von dem Funktionswert an der Stelle $\vec{x_i}$ abhängt, ist es ausreichend, die Funktion an den Beispielen zu betrachten. Aus a priori Verteilung und der Likelihood liefert der Satz von Bayes die a posteriori Verteilung der gesuchten Funktionswerte.

$$p(\vec{f} \mid X, \vec{y}, \vec{\theta}, \vec{\Psi}) = \frac{p(\vec{y} \mid \vec{f}, \vec{\theta}) \cdot p(\vec{f} \mid X, \vec{\Psi})}{\int p(\vec{y} \mid \vec{f}, \vec{\theta}) \cdot p(\vec{f} \mid X, \vec{\Psi}) \, \mathrm{d}f} \tag{2.8}$$

Hier zeigt sich ein deutlicher Unterschied zu dem parametrischen Ansatz. Das Ergebnis der bayesschen Inferenz beim parametrischen Ansatz ist ein Satz von Parametern des Modells, der es erlaubt die Funktion an beliebigen Stellen auszuwerten. Die bayessche Inferenz mit Gauß-Prozessen liefert hingegen Verteilungen der Funktionswerte der Trainingsbeispiele. Um die Funktion auch an anderen Stellen auszuwerten, ist ein Vorhersageschritt notwendig. Das entsprechende Integral (2.5) lässt sich für Gauß-Prozesse allerdings analytisch auswerten, was im parametrischen Fall in der Regel nicht möglich ist. Dies ist möglich, wenn die Beobachtungen $\vec{y}$ und die unbekannten Funktionswerte $\vec{f^*}$ gemeinsam normalverteilt sind:

$$p\left(\begin{bmatrix}\vec{y} \\ \vec{f_*}\end{bmatrix}\right) = \mathcal{N}\left(\vec{0}, \begin{bmatrix}K + \sigma^2 I & K_* \\ K_*^T & K_{**}\end{bmatrix}\right) \tag{2.9}$$

Die bedingte Verteilung $\vec{f^*} \mid \vec{y}$ lautet dann nach [3]:

$$p(\vec{f_*} \mid \vec{y}) = \mathcal{N}\left(K_*^T C^{-1} \vec{y}, K_{**} - K_*^T C^{-1} K_*\right) \tag{2.10}$$

mit $C = K + \sigma^2 I$. Die Matrizen K_{**} bzw. K_* enthalten dabei die Kovarianzen zwischen den vorherzusagenden Punkten $\vec{x}_*$ bzw. zwischen $\vec{x}_*$ und $\vec{x}$. Diese Lösung ist äquivalent zu (2.5). Die a posteriori-Verteilung der Funktionswerte ist in Abbildung 2.3 dargestellt.

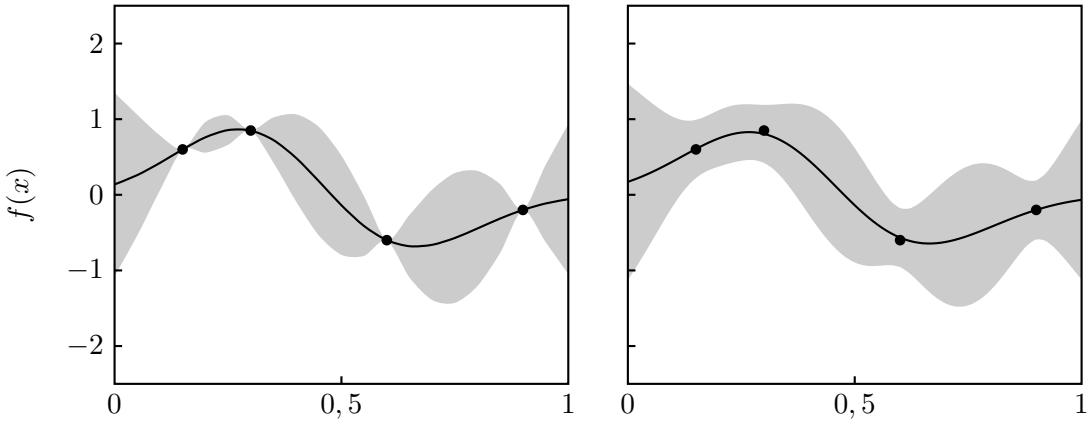


Abbildung 2.3: *A posteriori-Verteilung, der durch den Gauß-Prozess repräsentierten Funktionswerte. Die Beobachtungen sind durch Punkte markiert. Im linken Bild wurde angenommen, dass die Beispiele fehlerfrei beobachtet werden konnten. Die schattiert dargestellte a posteriori Unsicherheit ist an diesen Stellen Null. In der rechten Darstellung wurde angenommen, dass die Beobachtungen mit Rauschen behaftet sind. Die Beispiele werden lediglich approximiert und nach der Inferenz verbleibt eine Rest-Unsicherheit an diesen Stellen. Die Beobachtung von Beispielen reduziert auch die Unsicherheit der Prädiktionen in deren Umgebung.*

Der Rechenaufwand liegt durch die benötigte Matrix-Invertierung C^{-1} in $\mathcal{O}(n^3)$, wobei n die Anzahl der Trainingsbeispiele ist. Der benötigte Speicherplatz des Verfahrens wächst quadratisch in der Anzahl der Trainingsbeispiele, da jedes Trainingsbeispiel die Matrix C vergrößert.

2.1.3 Spärliche, inkrementelle Gauß-Prozess-Regression

Die oben beschriebene Formulierung der Gauß-Prozess-Regression hat für praktische Anwendungen zwei entscheidende Nachteile. Die benötigte Laufzeit und der benötigte Speicherplatz hängen kubisch bzw. quadratisch von der Anzahl der Trainingsbeispiele ab. Desweiteren muss das Inverse der Matrix C jedes Mal neu berechnet werden, wenn ein Trainingsbeispiel hinzukommt. Informationen aus früheren Berechnungen können dafür nicht verwendet werden.

In dieser Diplomarbeit wird daher ein von Csató und Opper [4, 5] vorgestellter Ansatz für spärliche, inkrementelle Gauß-Prozess-Regression verwendet. Das Verfahren ermöglicht die inkrementelle Aktualisierung der Matrix C^{-1} und die spärliche Approximation von Gauß-Prozessen. Dabei wird nur eine Teilmenge der Trainingsbeispiele gespeichert. Diese sogenannten *Basisvektoren* werden beim Training so angepasst, dass die a posteriori Verteilung der Basisvektoren möglichst genau mit der a posteriori Verteilung aller Trainingsbeispiele übereinstimmt. Gegebenenfalls wird die Menge der Basisvektoren dafür erweitert. Allerdings ist es möglich, eine obere grenze für die Anzahl Basisvektoren festzulegen. In diesem Fall können Basisvektoren automatisch durch andere ersetzt werden, um eine optimale Approximation zu gewährleisten. Auf diese Weise lässt sich die

Komplexität der Gauß-Prozess-Regression auf einen Speicherplatzbedarf von $\mathcal{O}(p^2)$ und eine Laufzeit von $\mathcal{O}(np^2)$ reduzieren, wobei n die Anzahl der Trainingsbeispiele und p die der Basisvektoren ist.

2.2 Optimierungverfahren

Als *Optimierungsproblem* bezeichnet man die Suche nach einem Eingabevektor $\vec{x}$, für den eine Funktion f ihr Maximum oder Minimum annimmt. Die Funktion f wird dabei bezeichnet als *Zielfunktion* bzw. *Kosten-* oder *Nutzenfunktion*, je nachdem, ob nach einem Minimum oder Maximum gesucht wird. Nur in wenigen Fällen lässt sich ein solches Extremum analytisch bestimmen. Für die meisten Arten von Problemen gibt es nur iterative Lösungsverfahren, deren Effizienz maßgeblich von der Komplexität der Funktion f abhängt [6]. Ist f linear, spricht man von einem *linearen Optimierungsproblem*, das z.B. durch *Innere-Punkte-Verfahren* effizient lösbar ist [7]. Auch für einige *nicht-lineare Optimierungsprobleme* ist eine effiziente Berechnung möglich, falls zusätzliche Eigenschaften von f bekannt sind, beispielsweise für konvexe Funktionen [8]. Im Allgemeinen lässt sich ein *globales* Extremum einer nichtlinearen Funktion jedoch nur durch eine vollständige Suche im Eingaberaum finden. Global bedeutet dabei, dass für $f : X \to Y$ gilt, $\forall x \in X : f(x) > f(x_{min})$ bzw. $f(x) < f(x_{max})$. Eine solche Suche ist in der Regel nicht effizient durchführbar, denn bereits für eine eindimensionale Funktion von $\mathbb{R}$ nach $\mathbb{R}$ ist der Eingaberaum unendlich groß. Effiziente Verfahren für nichtlineare Optimierung suchen daher nur nach einem *lokalen* Extremum als Approximation des Globalen. Für ein lokales Extremum x_{min} gilt für eine gewisse Umgebung, dass $f(x) > f(x_{min})$. Ein solcher Punkt lässt sich mit *lokalen* Informationen, wie etwa dem Gradienten der Funktion, finden.

Im Folgenden werden die in dieser Diplomarbeit verwendeten nichtlinearen Optimierungsverfahren vorgestellt: das *Downhill-Simplex-Verfahren*, das *Rprop-Verfahren* und die *Maximierung der erwarteten Verbesserung*.

2.2.1 Das Downhill-Simplex-Verfahren

Beim Downhill-Simplex-Verfahren [9] handelt es sich um ein Optimierungsverfahren für nichtlineare Funktionen nach J. Nelder und R. Mead aus dem Jahr 1965. Vorteile bietet dieses Verfahren im Gegensatz zu vielen anderen Optimierungsverfahren dadurch, dass kein Gradient der Kostenfunktion benötigt wird. Da somit keine Stetigkeit für die Funktionen verlangt wird, ist das Verfahren auf eine größere Klasse von Funktionen anwendbar als gradientenbasierte Verfahren. Nachteilig ist, dass diese Methode unter Umständen langsamer konvergiert als andere Verfahren. Die Probleme mit lokalen Nebenminima bleiben ebenfalls bestehen. Im Folgenden wird die Funktionsweise des Downhill-Simplex-Verfahrens am Beispiel eines Minimierungsproblems erläutert.

Um eine Funktion mit N Parametern zu minimieren, wird zunächst ein Simplex aus $N + 1$ Punkten bestimmt. Bei einem Simplex handelt es sich um die Verallgemeinerung eines Polygons auf N Dimensionen, wobei die Anzahl der Ecken stets $N + 1$ ist. Jeder Punkt entspricht einem Parametersatz. Zu jedem dieser Punkte wird der dazugehörige

Funktionswert y_i berechnet und der schlechteste und beste Funktionswert bestimmt.

$$y_s = \max_i(y_i) \qquad \text{schlechtester Punkt}$$
$$y_b = \min_i(y_i) \qquad \text{bester Punkt}$$

Das Ziel jeder Iteration ist es, den schlechtesten Parametersatz durch einen Besseren zu ersetzen. Da oftmals die Auswertung der Kostenfunktion aufwendig ist, muss mit einer geschickten Strategie nach neuen Punkten gesucht werden. Das Downhill-Simplex-Verfahren verwendet dazu vier Strategien:

- *Reflexion*
- *Kontraktion*
- *Expansion*
- *Kompression*

Zunächst wird versucht, durch *Reflexion* einen besseren Punkt zu finden. Dazu wird der schlechteste Punkt P_s am Mittelpunkt $\overline{P}$ aller anderen Simplexpunkte reflektiert.

$$P^* = (1 - \alpha) \cdot \overline{P} - \alpha \cdot P_s$$

Die positive Konstante α wird dabei als Reflektionskoeffizient bezeichnet. Liegt der Funktionswert des neuen Punktes zwischen dem schlechtesten und besten Funktionswert dieses Durchlaufes ($y_s > y^* > y_b$), wird P_s durch den neuen Punkt P^* ersetzt.

Erreicht der neue Punkt P^* einen Funktionswert, der besser ist als der vorherige beste Punkt ($y^* < y_b$), so wird versucht, noch etwas weiter in diese Richtung zu suchen, um einen noch besseren Wert zu erreichen. Dieser Schritt wird *Expansion* genannt.

$$P^{**} = \gamma \cdot P^* + (1 - \gamma) \cdot \overline{P}$$

Der Expansionskoeffizient γ nimmt dabei Werte größer 1 an. Ist auch P^{**} besser als der vorher Beste ($y^{**} < y_b$), wird P_s durch P^{**} ersetzt. Schlägt die Expansion fehl ($y^{**} > y_b$), so wird der schlechteste Punkt durch den durch die Reflexion entstandenen Punkt P^* ersetzt.

Falls der reflektierte Punkt P^* schlechter als der Punkt P_s mit den schlechtestem Funktionswert ist, so wird versucht einen besseren Punkt durch *Kontraktion* zu erhalten. Das bedeutet, dass der schlechteste Punkt P_s um einen gewissen Faktor näher an den Mittelwert $\overline{P}$ geschoben wird.

$$P^{**} = \beta \cdot P_s + (1 - \beta) \cdot \overline{P}$$

Die im Wertebereich zwischen 0 und 1 liegende Konstante β wird als Kontraktionskoeffizient bezeichnet. Ist der Wert P^{**} besser als das Minimum von y^* und y_s, so wird der schlechteste Parametersatz P_s durch den Neuen P^{**} ersetzt. Bringt auch dies keine Verbesserung, so wird das Simplex komprimiert, indem alle Punkte P_i durch $(P_i + P_b)/2$ ersetzt werden. Dieser Schritt wird als Kompression bezeichnet.

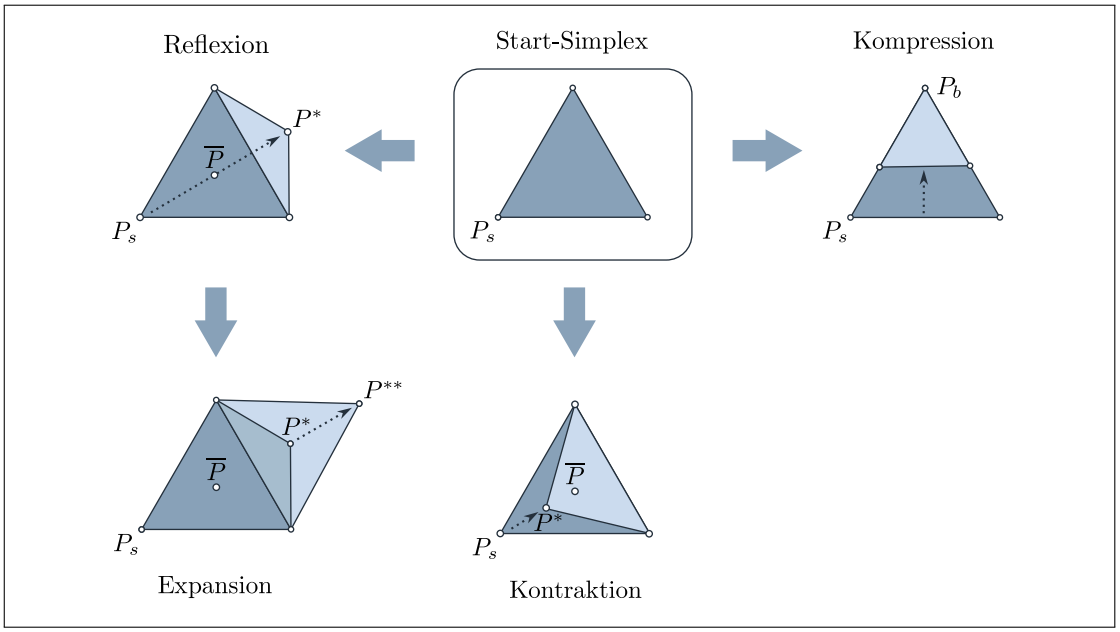


Abbildung 2.4: *Schritte des Downhill-Simplex-Verfahrens nach [10]*

Immer wenn ein neuer Punkt gefunden worden ist, wird der Algorithmus von vorne gestartet. Der Simplex nähert sich somit dem Optimum und zieht sich darum zusammen. Beendet wird der Algorithmus, wenn die Variationen der Funktionswerte im Simplex unter einen bestimmten Schwellwert sinken. Als Maß wird der quadratische Fehler $\sqrt{\sum(y_i - \overline{y})^2/n}$ verwendet.

2.2.2 Rprop

Rprop steht für „Resilient Propagation“ und wurde 1992 von M. Riedmiller und H. Braun vorgestellt [11, 12]. Im Gegensatz zum Downhill-Simplex-Verfahren handelt es sich bei Rprop um ein gradientenbasiertes Optimierungsverfahren. Ursprünglich wurde dieses in Bereich der *Neuronalen Netze* entwickelt, um die Wahl der Schrittweite des *Backpropagation of Error Algorithmus* [13] zu verbessern. Die Idee lässt sich jedoch auf allgemeine Gradientenabstiegsverfahren übertragen, da sie dieselben Schwachstellen haben. Dazu gehören das Überspringen von Extrema und der extrem langsame Lernfortschritt in flachen Bereichen der Fehlerfunktion.

Die Ursache beider Probleme liegt in der Wahl der Schrittweite dieser Verfahren, die typischerweise von dem Betrag des Gradienten der Fehlerfunktion abhängig ist. Dieser beschleunigt das Verfahren in steilen Bereichen der Fehlerfunktion, wohingegen die Bewegung in flachen Bereichen stark verlangsamt wird. Zwar gibt der Gradient durch sein Vorzeichen die Richtung des nächsten Extremums an, auf dessen Entfernung lässt sich von ihm jedoch nicht schließen. Die Koppelung der Lerngeschwindigkeit an den Gradientenbetrag ist daher problematisch.

Der Ansatz von Rprop besteht darin, die Schrittweite unabhängig vom *Betrag* des Gradienten der Fehlerfunktion zu wählen. Stattdessen wird sie in jeder Iteration anhand

des zeitlichen Verlaufs der *Vorzeichen* des Gradienten angepasst. Desweiteren wird das Vorzeichen des aktuellen Gradienten verwendet, um die Richtung der Aktualisierung zu bestimmen. Die Änderung der Parameter ergibt sich gemäß:

$$\Delta w_i^{(t)} = \begin{cases} -\Delta_i^{(t)} & \text{falls } \frac{\partial E^{(t)}}{\partial w_i} > 0 \\ +\Delta_i^{(t)} & \text{falls } \frac{\partial E^{(t)}}{\partial w_i} < 0 \\ 0 & \text{sonst} \end{cases} \tag{2.11}$$

Dabei wird für jede Parameteränderung $\Delta w_i^{(t)}$ eine individuelle Schrittweite $\Delta_i^{(t)}$ verwendet. Der Index t gibt dabei an, dass die Schrittweiten nicht konstant gewählt werden, sondern von der Zeit abhängen.

Zur Bestimmung der Schrittweite $\Delta_i^{(t)}$ werden in jeder Iteration der aktuelle Gradient und der der vorherigen Iteration betrachtet. Haben beide dasselbe Vorzeichen, so wird die Schrittweite vergrößert. Sind sie unterschiedlich, wird sie verringert.

$$\Delta_i^{(t)} = \begin{cases} \eta^+ \cdot \Delta_i^{(t-1)} & \text{falls } \frac{\partial E^{(t-1)}}{\partial w_i} \cdot \frac{\partial E^{(t)}}{\partial w_i} > 0 \\ \eta^- \cdot \Delta_i^{(t-1)} & \text{falls } \frac{\partial E^{(t-1)}}{\partial w_i} \cdot \frac{\partial E^{(t)}}{\partial w_i} < 0 \\ \Delta_i^{(t-1)} & \text{sonst} \end{cases} \tag{2.12}$$

Die Parameter η^+ und η^- steuern die Vergrößerung bzw. Verringerung der Schrittweite. Sie sind so gewählt, dass $0 < \eta^- < 1 < \eta^+$.

Haben die Ableitungen von E zu den Zeitpunkten $t-1$ und t unterschiedliche Vorzeichen, so bedeutet dies, dass ein Extremum übersprungen wurde. Somit war die vorangegangene Aktualisierung zu groß, daher wird die Schrittweite durch Multiplikation mit η^- verringert. Um Oszillationen zu vermeiden, wird zusätzlich die vorhergehende Aktualisierung rückgängig gemacht.

$$\Delta w_i^{(t)} = -\Delta w_i^{(t-1)}, \text{ wenn } \frac{\partial E^{(t)}}{\partial w_i} \cdot \frac{\partial E^{(t-1)}}{\partial w_i} < 0 \tag{2.13}$$

Durch diesen Rückschritt ändert sich im darauffolgenden Schritt erneut das Vorzeichen. Um die Schrittweite bei einem Vorzeichenwechsel nicht doppelt zu verringern, wird sie in dieser Iteration nicht angepasst.

Bleibt das Vorzeichen der partiellen Ableitung vom Zeitpunkt $t-1$ bis t erhalten, so wird die Schrittweite durch Multiplikation mit η^+ vergrößert. Dies vermeidet das Stagnieren des Lernverfahrens in flachen Bereichen der Fehlerfunktion, in denen die Gradienten kleine Werte annehmen.

In Experimenten haben Riedmiller et al. gezeigt, dass die genauen Werte von η^+ und η^- wenig Einfluss auf die Qualität der Ergebnisse oder die Konvergenzgeschwindigkeit des Verfahrens haben. In der Regel werden sie daher bei der Implementierung des Verfahrens problemunabhängig gewählt. Riedmiller et al. schlagen vor, $\eta^+ = 1,2$ zu wählen. Dieser Faktor ist groß genug, um für eine Beschleunigung in flachen Regionen zu sorgen, aber auch klein genug, um keine Oszillationen zu erzeugen. Der Faktor η^- wird verwendet, wenn ein Extremum übersprungen wurde. Da der Gradient aber keinerlei Informationen darüber liefert, wie weit das Extremum übersprungen wurde, ist es im Durchschnitt eine

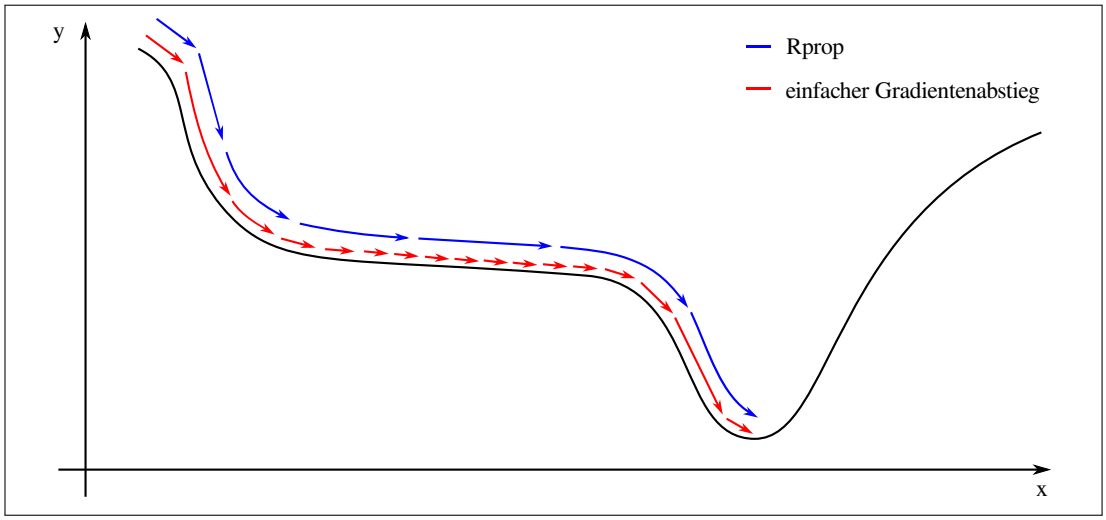


Abbildung 2.5: *Strategie von Rprop im Vergleich zum einfachen Gradientenabstieg*

gute Wahl, die Schrittweite zu halbieren. Der Faktor η^- sollte somit auf $0,5$ gesetzt werden, was einer binären Suche entspricht.

Bei einer Implementierung des Verfahrens ist es sinnvoll, die maximale und minimale Schrittweite zu begrenzen. Die minimale Schrittweite soll dabei lediglich eine gewisse minimale Lerngeschwindigkeit garantieren, ohne die Oszillationsgefahr zu erhöhen. Sie kann somit ebenfalls problemunabhängig gewählt werden. Die maximale Schrittweite soll garantieren, dass die Parameter nicht unkontrolliert wachsen können und hängt somit von der konkreten Problemstellung ab.

Ebenso problemabhängig ist die anfängliche Schrittweite Δ_0. Sie sollte unter Berücksichtigung der Größenordnung der Startparameter gewählt werden. Da die Schrittweite iterativ im Lernprozess angepasst wird, ist die Wahl der Startwerte nicht ausschlaggebend für den Lernerfolg.

Die geringe Zahl der Parameter ist einer der größten Vorteile von Rprop. Die meisten von ihnen sind fester Bestandteil des Verfahrens und müssen nicht von Hand eingestellt werden. Für die von Hand einstellbaren Parameter genügt eine grobe Abschätzung, da sie keinen Einfluss auf die Qualität des Ergebnisses haben, sondern vor allem die Geschwindigkeit des Verfahrens verbessern. Die explizite Wahl einer Lernrate ist im Gegensatz zu vielen anderen gradientenbasierten Verfahren nicht erforderlich.

Der zweite große Vorteil von Rprop ist die im Vergleich zu anderen Verfahren deutlich höhere Konvergenzgeschwindigkeit. Diese resultiert aus der Unabhängigkeit der Schrittweite vom Betrag des Gradienten der Fehlerfunktion. Die Topologie der Fehleroberfläche wird dabei in Form des Vorzeichens des Gradienten berücksichtigt. Auf diese Weise wird gleichzeitig die numerische Stabilität des Verfahrens erhöht, denn die Schrittweite kann sich nicht sprunghaft vergrößern oder verkleinern.

Für praktische Anwendungen ist hervorzuheben, dass das Verfahren durch die einfache Struktur der Aktualisierungsgleichungen (2.11) und (2.12) leicht zu implementieren ist. Der Rechenaufwand ist dabei im Vergleich zum normalen Gradientenabstieg nur unbedeutend größer.

2.2.3 Erwartete Verbesserung

Bei nichtlinearen Optimierungsproblemen kommt oftmals zu den Problemen bei der Suche nach lokalen Minima erschwerend hinzu, dass die Auswertung der Zielfunktion aufwendig ist. In solchen Fällen werden spezielle Algorithmen benötigt, die nach möglichst wenigen Schritten eine gute Lösung liefern, indem sie auf geschickte Weise den jeweils nächsten auszuwertenden Punkt wählen.

Ein solcher Algorithmus ist die *Maximierung der erwarteten Verbesserung*[1] [14, 15]. Er kann angewendet werden, wenn zusätzlich zu den Werten der Zielfunktion auch deren Unsicherheiten bekannt sind. Dies ist zum Beispiel der Fall, wenn die Zielfunktion durch Regression mit stochastischen Modellen, wie in Abschnitt 2.1.2 erklärt, ermittelt wurde.

Für eine gute Suchstrategie ist es wichtig, ein Gleichgewicht zwischen Erkundung von unbekannten Regionen und Ausnutzung von vielversprechenden bekannten Bereichen zu finden. Genau dies wird durch die Wahl mit Hilfe der erwarteten Verbesserung erreicht. Diese gibt an, welche Verbesserung an einem neuen Testpunkt gegenüber dem aktuell besten Punkt f_{best} zu erwarten ist.

Bei einem Minimierungsproblem ist die vorhergesagte Verbesserung wie folgt definiert:

$$\begin{aligned} I(\vec{x}) = I &= \begin{cases} f_{best} - \mu(\vec{x}) & \text{falls } \mu(\vec{x}) < f_{best} \\ 0 & \text{sonst} \end{cases} \\ &= \max(f_{best} - \mu(\vec{x}), 0) \end{aligned} \tag{2.14}$$

Dabei ist die Vorhersage $\hat{\mu}(\vec{x})$ an der Stelle $\vec{x}$ normalverteilt: $\hat{\mu}(\vec{x}) \sim \mathcal{N}(\mu(\vec{x}), \sigma^2(\vec{x}))$. Die erwartete Verbesserung ist dann der Erwartungswert der vorhergesagten Verbesserung, der als das Integral über die Dichtefunktion definiert ist:

$$\begin{aligned} E[I(x)] &= E[max(f_{best} - \mu(\vec{x}), 0)] \\ &= \int_{-\infty}^{\infty} I \cdot \phi(I) \mathrm{d}I \\ &= \int_{-\infty}^{\infty} I \cdot \frac{1}{\sqrt{2\pi} \cdot \sigma(\vec{x})} \exp\left(-\frac{(f_{best} - I - \hat{\mu}(\vec{x}))^2}{2 \cdot \sigma^2(\vec{x})} \right) \mathrm{d}I \\ &= \sigma(\vec{x}) \cdot [u \cdot \Phi(u) + \phi(u)] \end{aligned} \tag{2.15}$$

Dabei sind $\Phi(u)$ und $\phi(u)$ die Verteilungs- bzw. Dichtefunktion der Standardnormalverteilung

$$\Phi(u) = \frac{1}{2} \cdot \mathrm{erf}\left(\frac{u}{\sqrt{2}} \right) + \frac{1}{2} \qquad \phi(u) = \frac{1}{\sqrt{2\pi}} \cdot \exp\left(-\frac{u^2}{2} \right) \tag{2.16}$$

und $u = (f_{best} - \hat{\mu}(\vec{x}))/\sigma(\vec{x})$. Der erste Term der erwarteten Verbesserung $E[I(x)]$ gibt die vorhergesagte Differenz zwischen dem aktuell bestem Wert und dem Mittelwert an der Stelle $\vec{x}$, gewichtet mit der Wahrscheinlichkeit der Verbesserung, an. Der zweite Term der Funktion wird groß, wenn der Mittelwert von $\vec{x}$ nah an dem aktuell besten Wert liegt und eine hohe Unsicherheit aufweist.

[1] engl.: *expected improvement*

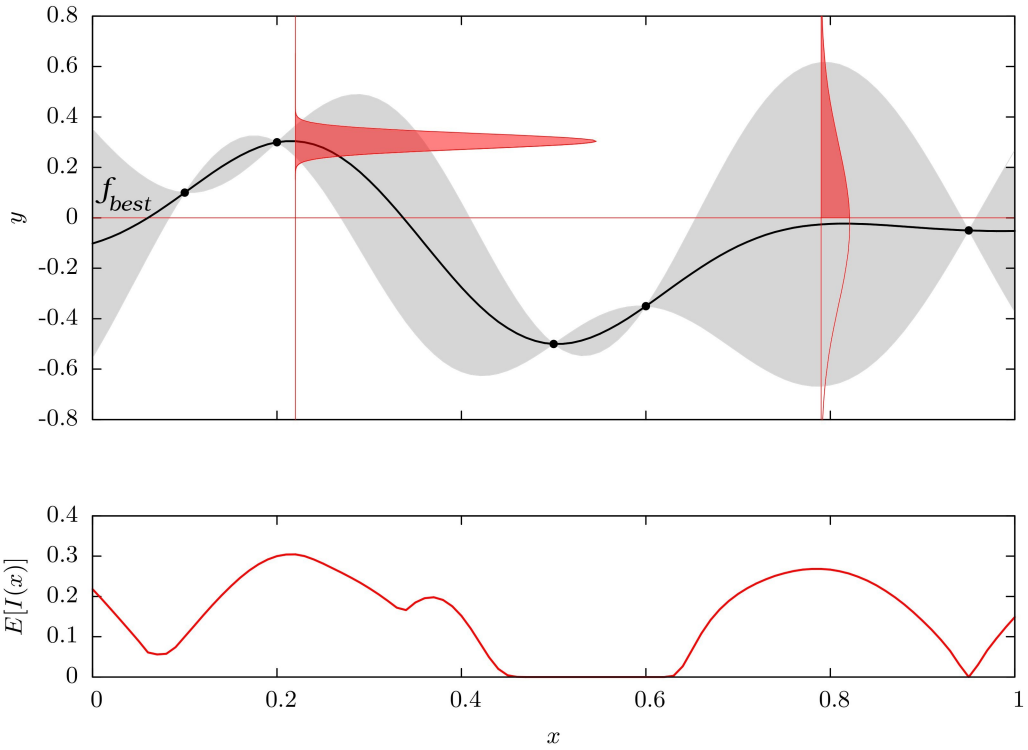


Abbildung 2.6: *Illustration der erwarteten Verbesserung. Die schwarze Linie stellt die a posteriori Mittelwertfunktion eines Gauß-Prozesses dar, der mit den als schwarze Punkte dargestellten Traingsbeispielen trainiert wurde. Die Standardabweichung der einzelnen Funktionswerte ist als grau schattierte Fläche eingezeichnet. Exemplarisch sind die Dichtefunktionen von zwei Funktionswerten in Rot dargestellt. Der rot schattierte Bereich überdeckt den Bereich ihrer Verteilung, in dem der Funktionswert über dem bisher besten Wert f_{best} liegt. In der Abbildung ist f_{best} durch die rote Linie bei $y = 0$ gekennzeichnet. Die erwartete Verbesserung entspricht dem Abstand des Schwerpunktes der schattierten Fläche von f_{best}. In er unteren Grafik sind die erwarteten Verbesserungen aller Funktionswerte dargestellt. Es ist zu erkennen, dass die erwartete Verbesserung für die beiden Beispielpunkte etwa gleich groß ist, obwohl der rechte Punkt einen deutlich niedrigeren Mittelwert aufweist. Der Grund hierfür liegt in der größeren Varianz des rechten Punktes.*

Die erwartete Verbesserung hat ihr Maximum somit an der Stelle x, an der der Mittelwert besser ist als der bisherige beste Wert und die Unsicherheit sehr hoch ist. Ebenso nimmt sie große Werte an, wenn die Vorhersage klein ist und die Unsicherheit hoch. Dagegen geht das Ergebnis der erwarteten Verbesserung bei bekannten Punkten und bei Punkten, an denen keine Verbesserung erwartet wird, gegen 0. Insgesamt sind die Werte der erwartete Verbesserung nie negativ.

Für eine Suche nach einem Maximum ändert sich nur die Definitionen von $I(x)$ und somit die von u.

$$I(x) = I = \mu(\vec{x}) - f_{best} \tag{2.17}$$

$$u = \frac{\hat{\mu}(\vec{x}) - f_{best}}{\sigma(\vec{x})} \tag{2.18}$$

Gradient der erwarteten Verbesserung

Um die Stelle $\vec{x}$ mit dem Maximum der erwarteten Verbesserung zu bestimmen, kann der Gradient der Funktion ausgenutzt werden [16]. Im Gegensatz zur ursprünglichen Kostenfunktion ist der Gradient der erwarteten Verbesserung berechenbar und häufig deutlich einfacher auszuwerten. Mehrmaliges Anwenden der Produktregel auf (2.15) liefert mit

$$\frac{\partial \Phi(u)}{\partial \vec{x}} = \phi(u) \tag{2.19}$$

den Gradienten

$$\frac{\partial E[I_{max}(x)]}{\partial (x)} = \frac{\partial \sigma(\vec{x})}{\partial \vec{x}} \cdot [u \cdot \Phi(u) + \phi(u)] + \sigma(x) \cdot \left[\frac{\partial u}{\partial \vec{x}} \cdot \Phi(u)\right] \tag{2.20}$$

Die Ableitung von $\sigma(x) = \sqrt{K_{**} - K_*^T C^{-1} K_*}$ lässt sich mit der Kettenregel bestimmen:

$$\frac{\partial \sigma(\vec{x})}{\partial \vec{x}} = -\frac{\left(\frac{\partial K_*^T}{\partial \vec{x}} \cdot C^{-1} \cdot K_*\right)}{\sigma(\vec{x})} \tag{2.21}$$

Der Unterschied zwischen Minimierungs- und Maximierungsproblem findet sich im Gradienten von u wieder.

$$\begin{aligned} \frac{\partial u_{min}}{\partial \vec{x}} &= \frac{\left(-\frac{\partial \hat{\mu}(\vec{x})}{\partial \vec{x}} - u \cdot \frac{\partial \sigma(x)}{\partial \vec{x}}\right)}{\sigma(x)} \\ \frac{\partial u_{max}}{\partial \vec{x}} &= \frac{\left(\frac{\partial \hat{\mu}(\vec{x})}{\partial \vec{x}} - u \cdot \frac{\partial \sigma(x)}{\partial \vec{x}}\right)}{\sigma(x)} \end{aligned} \tag{2.22}$$

Darin ist der Gradient von $\hat{\mu}(\vec{x}) = K_* \cdot C^{-1} \cdot y$ gegeben durch

$$\frac{\partial \hat{\mu}(\vec{x})}{\partial \vec{x}} = \frac{\partial K_*^T(\vec{x})}{\partial \vec{x}} \cdot C^{-1} \cdot \vec{y} \tag{2.23}$$

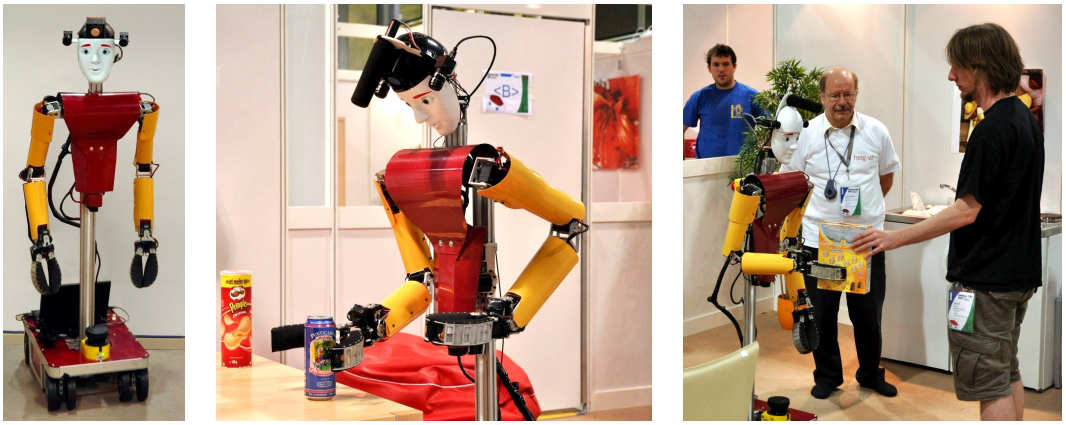


Abbildung 2.7: *Der Roboter Dynamaid der Abteilung AIS der Universität Bonn (Quelle: [19])*

Der letzte Gradient, um den der erwarteten Verbesserung berechnen zu können, ist der der Jacobi-Matrix $\frac{\partial K_*^T}{\partial \vec{x}}$.

$$\frac{\partial K_*^T}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial k(\vec{x},\vec{x_1})}{\partial x_1} & \cdots & \frac{\partial k(\vec{x},\vec{x_N})}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial k(\vec{x},\vec{x_1})}{\partial x_D} & \cdots & \frac{\partial k(\vec{x},\vec{x_N})}{\partial x_D} \end{bmatrix} \tag{2.24}$$

Dabei handelt es sich bei $x_1, \ldots, x_D$ um die Parameter der Dimension D. Die $\vec{x_1}, \ldots, \vec{x_N}$ sind die vorhandenen Trainingsbeispiele. Die genauen Gradienten der einzelnen Matrizeneinträge sind abhängig von der gewählten Kovarianzfunktion.

2.3 Roboter

Das in dieser Diplomarbeit entwickelte Lernverfahren wurde auf dem Roboter Dynamaid evaluiert. Dynamaid wurde als autonomer Haushaltsroboter in der Abteilung AIS der Universität Bonn entwickelt [17, 18]. Dabei wurden die besonderen Anforderungen berücksichtigt, die an Haushaltsroboter gestellt werden. Diese unterscheiden sich grundlegend von den Anforderungen an Industrieroboter.

In Industrieanwendungen zählen vor allem Genauigkeit, Stärke und Schnelligkeit. Die Roboter stehen in der Regel in einer speziell für sie angepassten Arbeitsumgebung und führen einen festen, sich wiederholenden Bewegungsablauf aus. Auf Interaktion mit dem Menschen sind diese Roboter nicht ausgelegt. Haushaltsroboter müssen sich dagegen in menschlichen Umgebungen zurechtfinden und direkt mit Menschen interagieren. Dabei zählt vor allem die Sicherheit des Menschen, autonomes Verhalten in sich ständig ändernden Umgebungen und intuitiver Umgang mit den Menschen. Im Gegensatz zur Industrie soll nicht die Umgebung dem Roboter angepasst werden, sondern der Roboter soll mit den gegebenen Umwelten zurechtkommen. Genau für diese Bedürfnisse wurde Dynamaid, die in Abbildung 2.7 zu sehen ist, entwickelt. Sie besitzt mit ca. 20 kg nur ein geringes Gewicht und stellt somit keine Gefahr für einen Menschen dar. Anhand

ihrer Sensoren kann sie Gegenstände und Menschen erkennen und ihnen ausweichen. Ihre schmale Gestalt erlaubt es ihr, sich auch gut in engeren Räumen oder schmalen Türdurchgängen zu bewegen. Unterstützt wird dies durch den Antrieb des Roboters. Durch vier unabhängig voneinander steuerbare Räder kann er sich omnidirektional bewegen und sich auf der Stelle drehen. Um eine intuitive Zusammenarbeit mit Menschen zu ermöglichen ist die Gestalt des Roboters der des Menschen nachempfunden. Seine Größe erlaubt Unterhaltungen auf Augenhöhe. Außerdem erleichtert sie das Greifen in menschlichen Umgebungen, in denen die Objekte oft auf Tischen oder in Regalen stehen. In dieser Höhe sind sie bequem für stehende Menschen erreichbar und somit auch für einen ähnlich großen Roboter. Um auch Objekte erreichen zu können, die besonders hoch oder niedrig stehen, ist Dynamaids Oberkörper höhenverstellbar angebracht. Auf diese Weise kann der Roboter seine Höhe je nach Aufgabe anpassen. Die beiden Arme von Dynamaid sind ebenfalls dem Menschen nachempfunden. Sowohl die Anordnung der Gelenke, als auch die Proportionen des Arms stimmen mit denen des Menschen überein. Dies erlaubt es dem Roboter, anthropomorphe Greifbewegungen auszuführen, wodurch Imitationslernen erst möglich wird.

Roboterarm

Der anthropomorphe Arm besitzt 7 Freiheitsgrade, die in Abbildung 2.8 dargestellt sind. Drei davon befinden sich in der Schulter, zwei im Ellbogen und die restlichen Zwei im Handgelenk. Die Freiheitsgrade des Arms entsprechen somit denen eines menschlichen Arms. Alle Gelenke werden durch Dynamixel Servomotoren angetrieben, die in Tabelle 2.1 aufgelistet sind.

Gelenk		Typ	max. Haltekraft [Nm]	max. Rotations-geschwindigkeit [rad/s]
Schultergelenk	Nickwinkel	2× EX-106	20,0	2,3
	Rollwinkel	EX-106	10,4	2,3
	Gierwinkel	EX-106	10,4	2,3
Ellbogengelenk	Nickwinkel	RX-64	6,4	2,6
	Gierwinkel	RX-28	3,8	2,6
Handgelenk	Nickwinkel	RX-64	6,4	2,6
	Rollwinkel	RX-64	6,4	2,6

Tabelle 2.1: *Darstellung der Gelenke des Roboters mit den dazugehörigen Servomotoren und deren Eigenschaften*

Die Servomotoren werden über einen seriellen RS-485 Bus von einem Atmel ATmega128 Mikrocontroller angesteuert. Dieser sendet unter anderem die gewünschten Gelenkwinkel, eine maximale Drehkraft und eine gewünschte Geschwindigkeit an die Servos, die daraufhin den gemessenen Gelenkwinkel an den Mikrocontroller zurückliefern.

Am Ende des Arms ist ein Greifer angebracht. Dieser besteht aus zwei Hälften, die einzeln geöffnet und geschlossen werden können. Jede Hälfte verfügt dafür über einen Dynamixel RX-28 Aktuator. Insgesamt sind Hand und Arm stark genug, um alltägliche

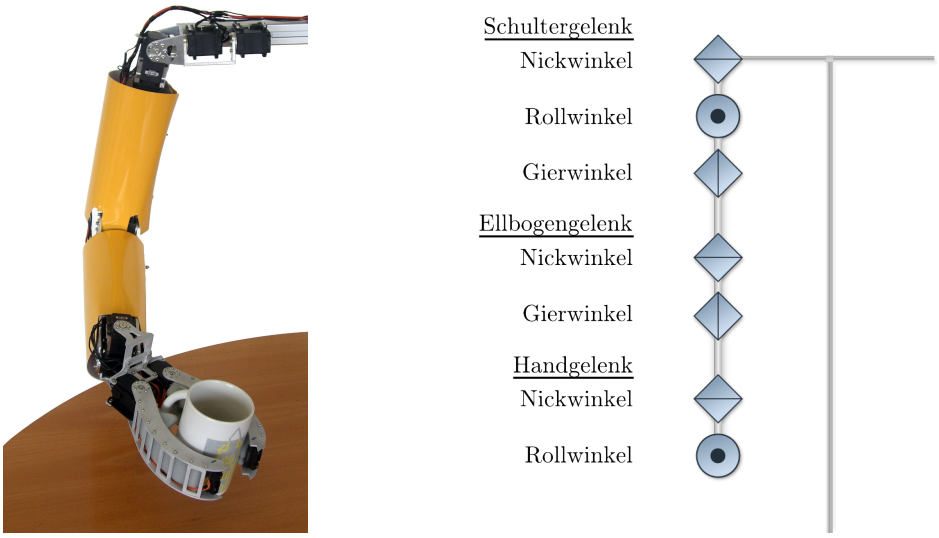


Abbildung 2.8: *Abbildung des in dieser Diplomarbeit benutzten Roboterarmes (links) und Schemazeichnung zur Anordnung der Gelenke (rechts)*

Objekte zu heben und zu tragen. Maximal darf die Gesamtlast 1kg betragen.

Sensorik zum Erkennen von Objekten

Zum Erkennen von Objekten auf Tischhöhe werden zwei Arten von Sensoren verwendet. Zum Lokalisieren von Objekten wird ein Hokuyo URG-04LX Laserentfernungsmesser verwendet. Der Nahbereich wird durch vier Infrarotsensoren der Firma Sharp vom Typ GP2D120XJ00F abgedeckt.

Der Laserentfernungsmesser befindet sich am beweglichen Oberkörper des Roboters und kann Objekte in einem Bereich von 240° und in bis zu 4m Entfernung erfassen. Durch einen Dynamixel RS-48 Servomotor ist er um die Roll-Achse drehbar. So kann er sowohl vertikal für die Erkennung von Tischen, als auch horizontal für die Erkennung von Objekten auf dem Tisch verwendet werden.

Für eine erhöhte Genauigkeit beim Greifen sind am Greifer zusätzlich vier Infrarotsensoren angebracht. Einer der Sensoren befindet sich unter dem Handgelenk und ist nach unten ausgerichtet. Mit ihm kann bestimmt werden, wie hoch sich die Hand über dem Tisch befindet. Ein weiterer Sensor befindet sich zwischen den beiden Hälften des Greifers. Die anderen Zwei sind an den Spitzen der beiden Greiferhälften angebracht. Mit diesen drei Sensoren kann erkannt werden, wo sich das Objekt relativ zur Handposition befindet und ob das Objekt gegriffen wurde. Um die Infrarotsensoren nutzen zu können, muss das Objekt zunächst mit dem Laserentfernungsmesser lokalisiert werden, da die maximale Reichweite der Sensoren 30cm beträgt.

3 Kapitel Verwandte Arbeiten

In diesem Kapitel werden Arbeiten vorgestellt, die mit dieser Arbeit thematisch verwandt sind. Dabei wird auf drei verschiedene Arten des Lernens eingegangen. In Abschnitt 3.1 wird ein kurzer Überblick über verwandte Arbeiten im Bereich des verstärkenden Lernens gegeben. Anschließend wird in Abschnitt 3.2 auf die Entwicklung des Imitationslernens eingegangen. In Abschnitt 3.3 werden ausgewählte Arbeiten vorgestellt, in denen die beiden zuvor beschriebenen Verfahren kombiniert werden, ähnlich dem in dieser Diplomarbeit verfolgten Ansatz.

3.1 Verstärkendes Lernen

Klassische Ansätze für verstärkendes Lernen basieren auf der Annahme, dass ein diskreter Zustandsraum vorliegt. Dies ist in der Praxis jedoch nur bei sehr einfachen Aufgabenstellungen der Fall. Im Folgenden werden Ansätze vorgestellt, wie verstärkendes Lernen in kontinuierlichen Zustandsräumen angewandt werden kann.

Kohl und Stone [20] benutzen eine Policy Gradient Methode, um die Laufgeschwindigkeit von Sony AIBO Robotern zu optimieren. Dabei werden zufällige Variationen der aktuellen Parameter gebildet, indem Parameter entweder um einen konstanten Wert vergrößert, verkleinert oder nicht verändert werden. Anhand der erreichten Laufgeschwindigkeiten der veränderten Parametersätze wird entschieden, wie die ursprünglichen Parameter für den nächsten Schritt angepasst werden. Problematisch an diesem Ansatz ist die hohe Zahl der auszuführenden Bewegungen für eine Aktualisierung der Parameter. Die dabei gewonnenen Informationen werden anschließend verworfen.

Lizotte et al. [21] verwenden einen probabilistischen Ansatz mit Gauß-Prozessen, der Entscheidungen auf Basis aller gesammelten Informationen trifft. Mit diesem Ansatz wird ebenfalls die Laufbewegung eines AIBO Roboters optimiert. Der Gauß-Prozess wird dazu verwendet, die Abbildung von Parametern auf die Laufgeschwindigkeit und die Glattheit der Bewegung zu approximieren. Als nächster zu testender Parametersatz wird derjenige ausgewählt, der die größte Wahrscheinlichkeit einer Verbesserung der Bewegung besitzt. Dabei wird die Möglichkeit einer Verschlechterung der Bewegung nicht in Betracht gezogen.

Der in dieser Diplomarbeit vorgestellte Ansatz des verstärkenden Lernens basiert ebenfalls auf Gauß-Prozessen und ermöglicht so eine gerichtete Suche unter Nutzung

aller zur Verfügung stehenden Informationen. Gleichzeitig wird durch Verwendung einer Kombination von *erwarteter Verbesserung* und *erwarteter Verschlechterung* vermieden, dass Punkte ausgewählt werden, die dem Roboter schaden könnten.

3.2 Imitationslernen

Das Lernen durch Imitation bei Menschen und Tieren wird bereits seit Ende des 19. Jahrhunderts von Verhaltensforschern und Kognitionswissenschaftlern untersucht. Seit ca. 30 Jahren interessiert sich auch die Robotik für diesen Ansatz [22]. Anwendungen fanden sich zu Beginn vor allem in der Programmierung von Industrierobotern. Statt Bewegungsfolgen von Hand zu programmieren, wurden diese durch Teleoperation erzeugt und dabei aufgezeichnet. Die Aufzeichnung wurde anschließend in Teilsequenzen gegliedert, so dass die gestellte Aufgabe als eine Folge von Zustandsübergängen repräsentiert werden konnte [23]. Dieses Verfahren wurde als Programmierung durch *Führung* bezeichnet [24]. Mit der Entwicklung autonomer mobiler Roboter erweiterte sich das Spektrum der Bereiche, in denen Roboter eingesetzt werden können, und es entstanden vielfältige Ansätze für das Lernen durch Imitation.

Eine Reihe von Ansätzen untersucht, wie komplexe Verhaltensweisen durch Imitation auf den Roboter übertragen werden können. Diese Ansätze gehen davon aus, dass der Roboter bereits ein Repertoire an Basisfähigkeiten besitzt, die zur Lösung komplexerer Aufgaben kombiniert werden können [25].

Die Frage, wie Roboter die benötigten Basisfähigkeiten erlernen können, stellt einen weiteren Forschungsschwerpunkt dar. Die meisten Ansätze gehen dabei davon aus, dass der Vorgang in 3 Phasen unterteilt werden kann. In der *Wahrnehmungsphase* wird die zu imitierende Bewegung erfasst. Diese Informationen werden anschließend in der *Erkennungsphase* in eine geeignete interne Repräsentation überführt. In der *Reproduktionsphase* wird die Bewegung ausgeführt und dabei gegebenenfalls an die aktuelle Situation angepasst [26, 27]. Die konkrete Implementierung dieser Phasen unterscheidet sich zwischen den verschiedenen Ansätzen.

Calinon et al. [28] verwendeten Lernen aus Imitation, um dem humanoiden Roboter HOAP-2 einfache Manipulationsaufgaben wie das Verschieben einer Schachfigur oder das Greifen eines Eimers beizubringen. Der Trainer führt die Bewegung vor, indem er die Arme des Roboters entsprechend der gewünschten Bewegung bewegt und der Roboter die Bewegung in seinem eigenen Koordinatensystem und mit seinen eigenen Sensoren aufzeichnet. Die Vorteile dieses Verfahrens sind, dass das Korrespondenzproblem zwischen der menschlichen Kinematik und der des Roboters umgangen werden kann und der Trainer keine Sensoren tragen muß. Die aufgezeichneten Bewegungen wirken allerdings nicht wie menschliche natürliche Bewegungen. Mittels Hauptkomponentenanalyse werden die Gelenkwinkeltrajektorien des Roboters, die binäre Öffnung der Hand und das Verhältnis der Hand zu Objekten in der Umgebung zunächst in einen niedrigdimensionalen Raum projiziert. Die niedrigdimensionalen Trajektorien werden anschließend durch Gauss-Mischverteilungen [1] repräsentiert, aus denen durch Regression eine generalisierte Trajektorie bestimmt wird. Um die Aufgabe zu imitieren, wird eine

[1] engl.: gaussian mixture model (GMM)

spezielle Kostenfunktion bezüglich dieser generalisierten Trajektorie und den im Modell enthaltenen Korrelationen zwischen den Eingabedaten optimiert.

In der ursprünglichen Form wurden mit diesem Verfahren zum Lernen sämtliche Trainingsbeispiele benötigt. In [29] erweitern die Autoren das Verfahren um die Fähigkeit des inkrementellen Lernens, und trainieren die Handzeichen von Basketball-Schiedsrichtern mit dem Roboter HOAP-3.

Pastor et al. [30] nutzen nichtlineare Differentialgleichungen, um Bewegungen zu repräsentieren. Die verwendete Form der Differentialgleichung entspricht einem durch externe Kräfte beeinflussten Federsystem. Sie enthält die Zielposition der Bewegung als expliziten Parameter, so dass durch Anpassung dieses Parameters Trajektorien zu beliebigen Positionen erzeugt werden können. Die Imitation einer gegebenen Trajektorie kann durch Anpassung der externen Kräfte gelernt werden. Die Vorteile eines solchen Ansatzes sind seine Flexibilität bei der Wahl des Endpunktes und seine Toleranz gegenüber Störungen. Um dies zu demonstrieren, trainierten die Autoren das Aufnehmen und Absetzen von Objekten und das Eingießen von Wasser in einen Becher an verschiedenen Positionen.

Im Gegensatz zu den oben beschriebenen Ansätzen, erlauben Aleotti und Caselli [31] das gleichzeitige Lernen von verschiedenen Lösungen derselben Aufgabe. Dazu werden die demonstrierten Bewegungen zunächst mit einem abstandsbasierten Verfahren in verschiedene Klassen eingeteilt. Für jede Klasse wird anschließend ein Hidden Markov-Modell der Trajektorien trainiert. Basierend auf der Likelihood der Daten, gegeben das Modell der Klasse, werden die konsistentesten Trajektorien jeder Klasse herausgefiltert. Diese werden schließlich durch je eine nicht-uniforme rationale B-Spline für jede Klasse approximiert, die einer generalisierten Form der durch die Klasse repräsentierten Lösung entspricht. Die Autoren schlagen ein interaktives System vor, das die durch die Splines repräsentierten Trajektorien in einem Simulator visualisiert und dem Benutzer zur Auswahl anbietet.

Die ersten beiden genannten Verfahren verwenden mehrere Demonstrationen einer Bewegung, um daraus eine generalisierte Repräsentation abzuleiten. Diese Bewegung kann anschließend auch in ähnlichen Situationen ausgeführt werden. Dabei erlauben die Ansätze jedoch nicht, für unterschiedliche Situationen, unterschiedliche Arten von Bewegungen zu parametrisieren. Der Ansatz von Aleotti erlaubt zwar, unterschiedliche Bewegungen zu repräsentieren, ist jedoch auf eine feste Situation beschränkt. Der in dieser Diplomarbeit vorgestellte Ansatz benötigt keine wiederholten Demonstrationen und generalisiert die in verschiedenen Situationen gelernten Bewegungen auch auf unbekannte Situationen.

3.3 Kombination von verstärkendem Lernen und Imitationslernen

Einer der ersten, der verstärkendes Lernen mit Imitationslernen kombinierte um den Suchraum des verstärkenden Lernens einzuschränken, war Schaal [32]. Dabei wählte er eine klassische Lernaufgabe, das Balancieren eines Stabes, welche das System nach einer Demonstration von 30 Sekunden in nur einem Versuch lernen konnte.

Smart und Kaelbling [33] betrachten ein klassisches Verfahren des verstärkenden Lernens, das Q-Lernen. Dieses ist in seiner ursprünglichen Formulierung nur auf diskrete Zustandsräume anwendbar. Die Autoren schlagen daher vor, die Werte-Funktion durch eine Approximation zu ersetzen, um das Verfahren auch auf kontinuierlichen Räumen anwenden zu können. Um eine schnelle Suche zu ermöglichen, schlagen die Autoren ein zweistufiges Lernverfahren vor, in dem ein Roboter zunächst durch ein Programm oder einen Menschen gesteuert wird und dabei Informationen über besuchte Zustände und erhaltene Belohnungen sammelt. Diese Informationen dienen dazu, die Wertefunktion zu initialisieren. In der zweiten Phase wird der Roboter durch das verstärkende Lernen gesteuert.

Guenter und Billard [34] erweiterten einen früheren Ansatz [35], in dem mit Hilfe von Imitationslernen Trajektorien einer Hand gelernt wurden. In dem früheren Ansatz wurden Bewegungen durch dynamische Systeme beschrieben. Um die Robustheit des Verfahrens zu erhöhen und Hindernissen ausweichen zu können, wurde verstärkendes Lernen eingesetzt, um die Parameter des dynamischen Systems zu modulieren. In Experimenten benötigte das Verfahren mindestens 750 Iterationen, um einen Pfad um ein Hindernis herum zu generieren.

Peters und Schaal [36] trainieren das Treffen eines ruhenden Baseballs mit einem Schläger auf einem Roboterarm mit 7 Freiheitsgraden. Die Aufgabe wird durch Bewegen des Roboterarms vorgeführt und der Roboter zunächst mit einem überwachten Lernverfahren trainiert. Dieses ist allerdings nicht in der Lage, die Bewegung ausreichend gut zu imitieren, um den Ball zu treffen. Die Autoren setzen daher verstärkendes Lernen ein, welches die Bewegung nach 200-300 Iterationen soweit verbessern kann, dass der Ball an der vorgegebenen Position getroffen wird.

Die vorgestellten Ansätze kombinieren Imitations- und verstärkendes Lernen in zweistufigen Ansätzen, in denen eine aus Demonstrationen gelernte Bewegung anschließend durch verstärkendes Lernen verbessert oder an eine neue Situation angepasst wird. In dieser Diplomarbeit wurde ein integriertes Lernverfahren entwickelt, welches anhand des vorhandenen Wissens in jeder einzelnen Situation entscheidet, welches Teilverfahren angewandt wird.

4 Kapitel Generierung von Greifbewegungen

Die Fähigkeit, Objekte zu greifen, ist eine wesentliche Voraussetzung, um mit der Umgebung zu interagieren und notwendig für die meisten alltäglichen Aufgaben. Menschen erlernen sie innerhalb des ersten Lebensjahres. Während die Koordination der Hände anfangs noch trainiert und Bewegungen bewusst ausgeführt werden müssen, entwickelt sie sich mit der Zeit zu einer unterbewussten Fähigkeit. Dabei entwickelt der Mensch auch ein intuitives Gefühl dafür, welche Bewegungen menschlich wirken. Andererseits fällt es ihm schwer, explizit zu beschreiben, was diese 'Menschlichkeit' ausmacht, und diese Merkmale auf einen Roboter zu übertragen. Einfache mathematische Modelle für Bewegungen reichen oft nicht aus, so dass Bewegungen von Robotern häufig unnatürlich, hölzern und kantig wirken. Diese typische Einschätzung zeigt, dass Glattheit eine wichtige Eigenschaft menschlicher Bewegungen ist. Durch die Analyse von Greifbewegungen ist außerdem erkennbar, dass diese in der Regel nicht geradlinig verlaufen, sondern einen Bogen beschreiben, und dass die Geschwindigkeit der Bewegungen nicht konstant sind.

In diesem Kapitel wird ein Regler beschrieben, der solche anthropomorphen Greifbewegungen erzeugen kann. Da diese Bewegungen sehr unterschiedlich aussehen können, muss sich der Regler flexibel an die entsprechende Trajektorie anpassen können. Dazu erhält er als Eingabe 29 Parameter, die die zu generierende Bewegung charakterisieren. Diese Parametrisierung bietet zwei Vorteile. Zum einen ist die Dimension des Parametervektors klein und dieser somit besser handhabbar als die vollständige Repräsentation einer Trajektorie. Zum anderen sind die Parameter durch die Form der Bewegung motiviert und somit für den Menschen leicht interpretierbar. Die Ausgabe des Reglers besteht aus einer Zieltrajektorie von Positionen und Orientierungen des Endeffektors über die Zeit und der Öffnung der Hand zu jedem Zeitpunkt. Dies hat den Vorteil, dass die Werte anschaulich und gut visualisierbar sind und dass der Regler auf beliebigen Roboterarmen unabhängig von ihrer Bauweise genutzt werden kann.

Inverse Kinematik

Der im Rahmen dieser Diplomarbeit benutzte Roboterarm wird, wie die meisten anderen Roboter auch, mit Hilfe von Gelenkwinkeln angesteuert. Daher müssen die errechneten Zielposen der Hand durch inverse Kinematik in einen Winkel pro Robotergelenk umgewandelt werden. Der Regler setzt dabei mit einer Rate von $50Hz$ neue Posen für die Hand. Durch die inverse Kinematik werden Grenzen sowohl in den Endeffektor- als auch

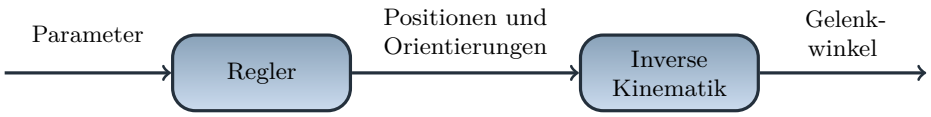


Abbildung 4.1: *Bestimmung der für den Roboter benötigten Gelenkwinkel*

in den Gelenkwinkelgeschwindigkeiten des Roboters eingehalten. Zusätzlich wird, falls es keine eindeutige Lösung gibt, um eine Position zu erreichen, dafür gesorgt, dass sich die Gelenkwinkel in der Mitte des zulässigen Bereichs bewegen.

Bestimmung der Positionen der Trajektorie

Um die Ausgabetrajektorie zu generieren, wird die Bewegung in zwei Teilsequenzen gegliedert. Dazu wird zusätzlich zum Start- und Zielpunkt ein *Viapunkt* definiert, der den höchsten Punkt der Trajektorie darstellt. Dieser Punkt zählt, genau wie der Start- und Zielpunkt, zu den Eingabeparametern des Reglers. Als erste Teilsequenz wird die Bewegung vom Start- zum Viapunkt betrachtet. Hat man diesen erreicht, wird in die zweite Teilsequenz vom Via- zum Zielpunkt eingeschwenkt.

Die beiden Teilsequenzen werden anschließend in viele kleine Zwischenziele unterteilt, die der Reihe nach angesteuert werden. Auf diese Weise entsteht eine glatte Bewegung. Zusätzlich vereinfacht der geringe Abstand der Zwischenziele die Ansteuerung, z.B. durch einen inversen Kinematik-Algorithmus.

Die Zwischenziele beider Teilsequenzen werden auf die gleiche Weise bestimmt. Jede Sequenz wird durch einen Anfangspunkt P_A und einen Endpunkt P_E begrenzt. In der ersten Hälfte sind dies Start- und Viapunkt, in der Zweiten Via- und Zielpunkt.

Zur Bestimmung der Richtung, in der das nächste Zwischenziel liegt, wird je eine Tangente durch den Anfangs- und den Endpunkt einer Teilsequenz gelegt. Die Tangenten geben die gewünschte Richtung der Trajektorie an diesen Punkten an und sind für alle Zwischenziele einer Teilbewegung gleich. Die Richtungen der Tangenten gehören ebenfalls zu den Eingabeparametern des Reglers. Die Richtung zum nächsten Zwischenziel entspricht zu Beginn einer Teilsequenz in etwa der Richtung der Tangente am Anfangspunkt, zum Ende der Sequenz entspricht sie der Richtung der Tangente am Endpunkt. Zwischen diesen beiden Punkten werden die Zwischenziele so bestimmt, dass sich ein glatter Übergang zwischen Start- und Endrichtung ergibt. Dazu wird für jedes Zwischenziel zunächst je ein Hilfspunkt auf jeder der beiden Tangenten gewählt.

Die Lage dieser Punkte ist abhängig von der Distanz zwischen der aktuellen Position P und dem Ursprung der jeweiligen Tangente. Je weiter die aktuelle Position vom Tangentenursprung entfernt liegt, desto weiter liegt auch der Hilfspunkt von ihm entfernt. Die genaue Entfernung wird durch Skalierungsfaktoren α_A und α_E bestimmt:

$$\begin{aligned} H_A &= P_A + \alpha_A \cdot \|P_A - P\| \cdot d_A \\ H_E &= P_E - \alpha_E \cdot \|P_E - P\| \cdot d_E \end{aligned} \tag{4.1}$$

Dabei stehen d_A und d_E für die Richtungen der Tangenten am Anfangs- und Endpunkt. Die Skalierungsfaktoren sind sowohl für den Anfangs- und den Endpunkt eines Segmentes,

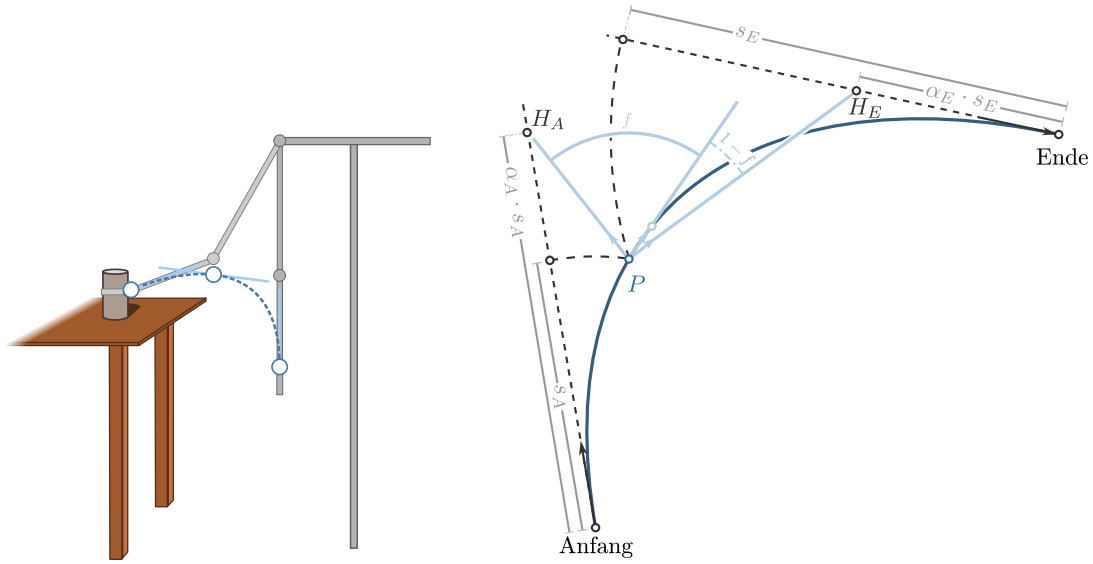


Abbildung 4.2: *Im linken Bild ist eine mögliche Greiftrajektorie dargestellt. Die Kreise kennzeichnen dabei den Start-, Via- und Zielpunkt. Duch die blauen Linien sind die Richtungen an diesen Punkten kenntlich gemacht. Diese Bewegung wird am Viapunkt in zwei Teilsequenzen gegliedert. Im rechten Bild ist die Bestimmung der Richtung zum nächsten Zwischenziel für eine Teilsequenz dargestellt. Durch den Anfangs- und Endpunkt verlaufen die dunkelgrau gestrichelt dargestellten Tangenten. Auf diesen werden die Hilfspunkte H_A und H_E bestimmt, indem der Abstand des aktuellen Punktes P auf die Tangenten projiziert und anschließend mit den Faktoren α_A bzw. α_E gestreckt wird. Anschließend werden die hellblau dargestellten Richtungen von P zu den Hilfspunkten mit dem Faktor f gewichtet, um die endgültige Richtung des nächsten Zwischenzieles zu bestimmen. Auf diese Weise entsteht die dunkelblau dargestellte, runde Trajektorie vom Anfangs- zum Endpunkt.*

als auch für jede Teilsequenz unterschiedlich und stellen weitere Eingabeparameter des Reglers dar. Durch ihre Wahl lässt sich steuern, wie gebogen bzw. wie gerade die Trajektorie verläuft. Somit sind weit ausholende Greifbewegungen genauso möglich wie geradlinige.

Die Richtung d des nächsten Zielpunktes wird durch Interpolation der Richtungen von der aktuellen Position zu den beiden Hilfspunkten bestimmt:

$$d = f \cdot \frac{H_A - P}{\|H_A - P\|} + (1 - f) \cdot \frac{H_E - P}{\|H_E - P\|} \tag{4.2}$$

Der Einfluss der Richtung zum Punkt auf der Anfangstangente nimmt mit der Nähe zum Endpunkt ab. Ab einer gewissen Distanz, die durch einen weiteren Eingabeparameter θ bestimmt wird, hat der Punkt auf der Anfangstangente sogar überhaupt keinen Einfluss mehr auf die Richtungsbestimmung. Die Richtung zum nächsten Zielpunkt hängt dann

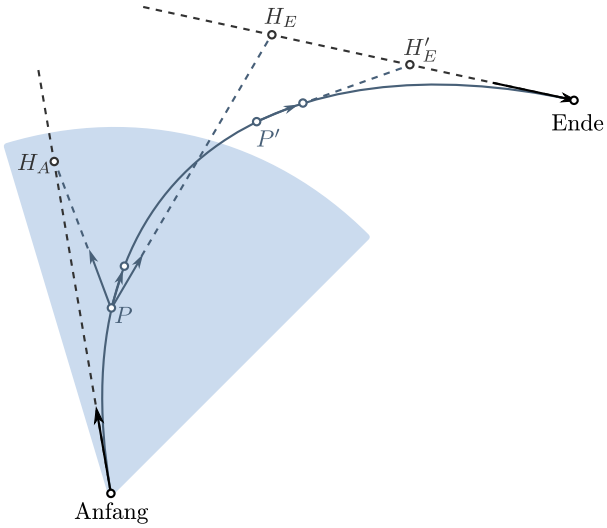


Abbildung 4.3: *Einfluss der Richtung am Anfangspunkt der Teilsequenz: Der Einflussbereich der Richtung am Anfangspunkt ist schattiert. Die Richtung des nächsten Zwischenzieles am Punkt P, der in diesem Bereich liegt, hängt von den Richtungen zu den Hilfspunkten H_A und H_E ab. Für den Punkt P', der außerhalb des schattierten Bereichs liegt, hängt die Richtung zum nächsten Zwischenziel nur noch vom Hilfspunkt H'_E ab. Innerhalb des schattierten Bereichs nimmt der Einfluss der Richtung am Anfangspunkt mit sinkender Entfernung zum Endpunkt ab.*

nur noch von der Richtung zum zweiten Hilfspunkt ab.

$$f = 1 - \frac{d_A}{\theta \cdot \|P_E - P_A\|} \qquad 0 \leq f \leq 1 \tag{4.3}$$

Durch die Richtungen der Zielpunkte wird die Form der Trajektorie kontrolliert. Daneben ermöglicht der Regler auch die Steuerung der Geschwindigkeit der Bewegung. Diese beträgt am Anfangspunkt immer 0 und steigt in Richtung Viapunkt. Am Viapunkt nimmt sie ihren maximalen Wert an, der durch einen weiteren Eingabeparameter festgelegt werden kann. Von dort an sinkt die Geschwindigkeit in Richtung Zielpunkt. Am Zielpunkt erreicht sie einen kleinen, konstanten Wert, der hilft, Stagnationen vor dem Ziel zu vermeiden. Die Geschwindigkeit an den einzelnen Punkten wird durch lineare Interpolation bestimmt. Dabei gehört sie nicht direkt zur Ausgabe des Reglers, sondern wird implizit durch den Abstand zweier aufeinanderfolgender Posen deutlich. Da der Zeitabstand zwischen den ausgegebenen Posen immer gleich ist, entspricht ein großer Abstand einer hohen Geschwindigkeit.

Bevor die generierten Zwischenziele an die inverse Kinematik weitergegeben werden, wird überprüft, ob sich die Geschwindigkeit, die Beschleunigung und die komponentenweisen Beschleunigungen der drei Richtungen nicht zu stark ändern. Ist die Änderung zu groß, so werden die entsprechenden Werte verringert. Somit können keine zu extremen Änderungen entstehen, die dem Roboter schaden oder zu Problemen mit der inversen Kinematik führen könnten.

Zur Bestimmung des nächsten Zwischenzieles P_* wird der aktuelle Endeffektorpunkt um einen Zeitschritt mit der aktuellen Geschwindigkeit v in die zuvor bestimmte Richtung d verschoben.

$$P_* = P + v \cdot \Delta t \cdot d \tag{4.4}$$

Um einen glatten Übergang zwischen den Teilsequenzen zu erhalten, stimmen der Endpunkt und dessen Tangente des ersten Segmentes mit dem Anfangspunkt und der dazugehörigen Tangente der zweiten Sequenz überein.

Bestimmung der Ausrichtung der Hand

Neben der Position ist auch die Ausrichtung der Hand im Laufe einer Bewegung charakteristisch für anthropomorphe Bewegungen. Bei den von diesem Regler generierten Trajektorien wird die Hand - wie auch beim Menschen - umso stärker auf das zu greifende Objekt ausgerichtet, je mehr sie sich ihm nähert. In der ersten Teilsequenz einer Greifbewegung wird die Hand möglichst bequem gehalten. Für menschliche Bewegungen bedeutet dies, dass die Hand eine Verlängerung des Unterarms bildet. In der zweiten Teilsequenz findet eine lineare Interpolation von der bequemen Handorientierung in Richtung Ziel statt.

Bestimmung der Öffnung der Hand

Zusätzlich zu der Pose bestimmt der Regler auch den Öffnungsgrad der Hand anhand drei weiterer Eingabeparameter. Zu Beginn der Bewegung ist die Hand geschlossen und wird bis zum Viapunkt auf einen Wert geöffnet, der durch einen Eingabeparameter bestimmt wird.. Ein weiterer Parameter sagt aus, wie weit die Hand maximal geöffnet wird, und in welcher Distanz zum Zielpunkt dies der Fall ist. Von diesem Punkt an wird die Hand bis zum Zielpunkt wieder geschlossen. Dies spiegelt das menschliche Greifverhalten wieder. Ab einer gewissen Entfernung zum Greifobjekt schließt der Mensch seine Hand um das Objekt zu greifen.

Die Idee, Bewegungen in Teilsequenzen aufzuteilen und fließende Bewegungen durch Einschwenken in vorgegebene Übergangsrichtungen zu erzeugen, erlaubt nicht nur das Erzeugen beliebiger Greifbewegungen. Durch Hinzufügen weiterer Viapunkte und damit Teilsequenzen, lassen sich beliebige Trajektorien erzeugen.

Parameter	Dimension
Startpunkt	3
Viapunkt	3
Zielpunkt	3
Richtung am Startpunkt	3
Richtung am Viapunkt	3
Richtung am Zielpunkt	3
Öffnung der Hand am Viapunkt	1
Maximale Öffnung der Hand	1
Abstand zum Ziel bei max. Öffnung der Hand	1
Beschleunigung der Bewegung vom Via- zum Zielpunkt	1
Beschleunigung der Bewegung vom Start- zum Viapunkt	1
Begrenzung des Einflusses der Richtung am Startpunkt	1
Begrenzung des Einflusses der Richtung am Viapunkt	1
Skalierungsfaktor auf Tangente am Startpunkt	1
Skalierungsfaktor auf Tangente zum Viapunkt	1
Skalierungsfaktor auf Tangente vom Viapunkt	1
Skalierungsfaktor auf Tangente zum Zielpunkt	1

Tabelle 4.1: *Die 29 Parameter des Reglers*

5 Kapitel Lernverfahren

5.1 Überblick

Lernen ist eine der wichtigsten Fähigkeiten, sowohl von Menschen, als auch von Tieren. Sie dient zur Aneignung von Fähigkeiten und Kenntnissen. Spätestens ab der Geburt an, so sind sich Wissenschaftler sicher, fangen Kinder an zu lernen. Nur wenige menschliche Fähigkeiten sind angeboren, die Meisten werden im Laufe des Lebens erlernt. Dies erlaubt es dem Menschen, sich flexibel an eine sich verändernde Umwelt anzupassen. Ohne Lernen wären Menschen somit nicht überlebensfähig.

Menschen lernen auf verschiedene Weisen. Zwei der meist benutzten Lernarten sind das Lernen am Modell und das Lernen aus eigener Erfahrung. Beim Lernen am Modell eignet sich der Mensch Fähigkeiten durch Beobachtung und Nachahmung an. Lernt er aus eigener Erfahrung, so entwickelt und erprobt der Mensch eigene Lösungsstrategien und erkennt anhand der Reaktion der Umwelt, ob diese gut sind oder nicht. Menschen nutzen oft nicht nur eine Art des Lernens alleine, um sich Fähigkeiten anzueignen. Das Lernen am Modell und das Lernen aus eigener Erfahrung bieten sich besonders zur Kombination an. Das Nachahmen einer Fähigkeit liefert oft nicht genau dasselbe Ergebnis wie die Demonstration. Viele Fähigkeiten verlangen Training, jedoch liefert Imitationslernen dem Menschen eine Idee, wie das Problem gelöst werden kann. Ausgehend davon probiert ein Mensch ähnliche Lösungen aus und evaluiert die Ergebnisse, um dann seine Leistung zu verbessern. Dieses Lernen aus Erfahrung wird auch als *verstärkendes Lernen* bezeichnet. Durch das Zusammenspiel der beiden Lernarten kann der Mensch sich wesentlich schneller Fähigkeiten aneignen als durch verstärkendes Lernen allein.

Genau diesen Vorteil macht sich auch die Robotik zunutze. Da viele Fähigkeiten nicht programmiert werden können oder der Aufwand zu hoch wäre, sollen Roboter diese erlernen. In der Robotik ist verstärkendes Lernen schon lange bekannt und wird zur Lösung vielfältiger Probleme genutzt. Allerdings stellt sich hierbei das Problem, dass Roboter nicht in der Lage sind, intuitiv neue Lösungen zu finden, die schnell zum Ziel führen. Auf der anderen Seite ist das Durchsuchen des oftmals hochdimensionalen Lösungsraumes sehr aufwendig. Dieser Aufwand kann durch Imitationslernen verringert werden, indem auch der Roboter eine Imitation als Ausgangspunkt für seine Suche verwendet.

Das im Rahmen dieser Diplomarbeit entwickelte Lernverfahren kombiniert diese beiden

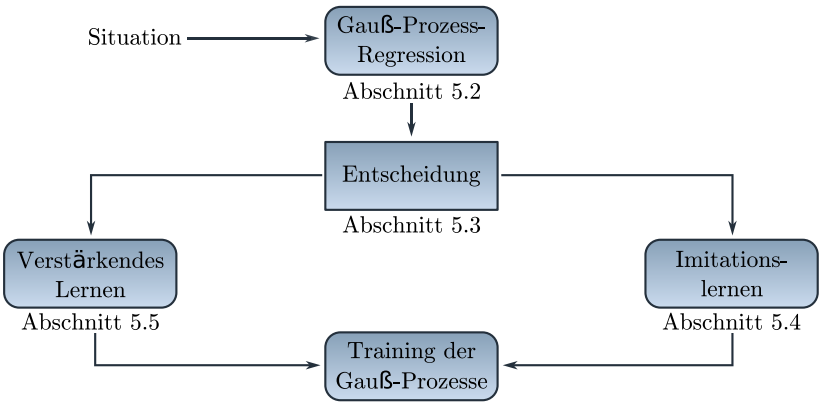


Abbildung 5.1: *Übersicht über das Lernverfahren*

Arten des Lernens. Es ist somit biologisch motiviert und stark an menschliches Lernen angelehnt.

Mit diesem kombinierten Verfahren soll das Greifen einer Tasse trainiert werden. Dazu wird zu Beginn jeder Iteration eine Tasse auf eine beliebige Position auf einen Tisch gestellt. Je nach Situation kommt anschließend das Lernen durch Imitation oder verstärkendes Lernen zum Einsatz. Die Entscheidung zwischen diesen beiden Komponenten wird mit Hilfe der in Abschnitt 2.1.2 vorgestellten Gauß-Prozesse gefällt. Diese repräsentieren den funktionalen Zusammenhang zwischen den möglichen Greifbewegungen und einer Bewertung. Dafür wird der Gauß-Prozess mit den bisher ausgeführten Bewegungen trainiert, die durch die Parameter des in Abschnitt 4 vorgestellten Reglers parametrisiert werden. Die Bewertung wird durch eine Gütefunktion ausgedrückt. Neben dem geschätzten Wert dieser Gütefunktion liefert der Gauß-Prozess auch eine Unsicherheit dieser Schätzung. Anhand dieser beiden Größen kann der Roboter bestimmen, ob er die Tasse an einer ähnlichen Position schon einmal beobachtet hat, und ob eine gute Lösung für eine Greifbewegung an dieser Position vorliegt.

Ist dies der Fall, wird ausgehend von dieser Lösung versucht, ähnliche Parameter zu finden, von denen erwartet wird, dass sie die Güte weiter verbessern. Ist die Situation unbekannt oder gibt es nur Bewegungen, die bei der Ausführung einen schlechten Gütewert erhalten haben, so wird der Mensch aufgefordert, die zu der Position der Tasse passende Bewegung vorzuführen. Aus dieser erhält der Roboter die Informationen, um die Bewegung nachzuahmen und auszuführen. So wird verhindert, dass der Roboter gänzlich unbekannte Parameter testet, die ihm zum einen schaden könnten und zum anderen auch nicht eine menschlich wirkende Bewegung garantieren würden.

Unabhängig von der Wahl des Lernverfahrens findet nach der Ausführung der Bewegung eine automatische Bewertung statt. Dies geschieht in zwei Stufen. Zunächst wird die Bewegung in der Simulation getestet, um zu verhindern, dass der Roboter Schaden nimmt. Dies könnte beispielsweise geschehen, wenn er Bewegungen ausführt, die zu Kollisionen mit dem Tisch führen könnten oder solche, die Probleme mit der inversen Kinematik verursachen. Solche Bewegungen erhalten direkt eine negative Gütebewertung von -1 und werden nicht mehr für die zweite Ausführungsstufe zugelassen.

In der zweiten Stufe wird die Bewegung auf dem realen Roboter ausgeführt. Schafft der Roboter es nicht die Tasse zu greifen und hochzuheben, so erhält er eine Bewertung von 0. Falls er die Tasse greifen kann, so sind die Gütewerte abhängig vom Versatz v der Tasse. Dazu wird die Position der Tasse vor und nach dem Greifen bestimmt. Auf diese Weise kann erkannt werden, wie weit der Roboter die Tasse verschoben hat.

$$f_{Güte} = \begin{cases} 1 - v & \text{falls die Tasse gegriffen wurde} \\ 0 & \text{falls die Tasse verfehlt wurde} \\ -1 & \text{falls Kollisionen oder IK Probleme auftraten} \end{cases} \tag{5.1}$$

Die getesteten Parameter und die Bewertung der Ausführung werden als neues Beispiel zur Menge der Trainingsbeispiele des Gauß-Prozesses hinzugefügt. Wird die Tasse zu einem späteren Zeitpunkt an eine ähnliche Position gestellt, hat der Roboter mehr Wissen über das Problem. Er ist dann nicht mehr auf Demonstrationen angewiesen, sondern kann die entsprechende Greifbewegung durch Lernen aus eigener Erfahrung verbessern. Je mehr Wissen der Roboter ansammelt, desto gezielter können die Parameter der Bewegung verbessert werden. Die Struktur dieses Lernverfahrens ist in Abbildung 5.1 zusammengefasst.

Im Folgenden wird zunächst in Abschnitt 5.2 die Rolle der Gauß-Prozesse im Lernverfahren erläutert. Basierend darauf wird im folgenden Abschnitt 5.3 beschrieben, wie mit Hilfe von Vorhersagen der Gauß-Prozesse entschieden wird, welches Teilverfahren in einer gegebenen Situation zum Einsatz kommen soll. Die beiden Teilverfahren Imitationslernen und verstärkendes Lernen und ihre Rolle im Rahmen des kombinierten Verfahrens werden anschließend in den Abschnitten 5.4 und 5.5 vorgestellt.

5.2 Gauß-Prozesse im Lernverfahren

Lernen bedeutet, nicht nur Informationen über bekannte Beispiele auswendig zu lernen, sondern auch auf andere Situationen generalisieren zu können.

Das im Rahmen dieser Diplomarbeit entwickelte Verfahren zum Lernen von Greifbewegungen soll nicht nur in der Lage sein, demonstrierte Bewegungen zu imitieren, sondern auch bei ähnlichen Positionen der Tasse eine geeignete Greifbewegung finden können. Dies ist nötig, da es in der Regel nicht möglich ist, für jede mögliche Position der Tasse eine eigene Beispielbewegung aufzunehmen. Ein generalisierungsfähiges Lernverfahren erlaubt es, aus wenigen vorgeführten Bewegungen das Greifen auf dem gesamten Tisch zu lernen.

Das hier beschriebene Lernverfahren macht sich die Generalisierungsfähigkeit von Gauß-Prozessen zunutze. Die Funktionsweise von Gauß-Prozessen ist in Abschnitt 2.1 erläutert. Ihre Trainingsdaten bestehen aus den Parametern, die die bisher ausgeführten Bewegungen repräsentieren, und den zugehörigen Gütewerten der Bewegungen. Gauß-Prozesse sind für das in dieser Diplomarbeit vorgestellte Verfahren besonders geeignet, da sie zum einen die getesteten Parameter zusammen mit ihrer Güte speichern und zudem diese Werte auch generalisieren können. Auf diese Weise kann auch unter nicht ausgeführten Bewegungen nach guten Parametern gesucht werden. Zum anderen liefern Gauß-Prozesse nicht nur zu jedem angefragten Parametersatz den Mittelwert der zu

erwartetenden Güte, sondern auch die Unsicherheit über diese Schätzung. Gerade in der Robotik ist dieses Wissen wichtig, da Parameter die sehr unsicher sind und somit keine große Ähnlichkeit mit den vorhandenen Beispielen haben, bei der Ausführung zu Schäden am Roboter führen könnten. Aufgrund dieser Informationen können zielgerichtete Entscheidungen getroffen werden, um für das verstärkende Lernen vielversprechende, neue zu testende Parameter zu finden.

Diesem Ansatz liegt die Annahme zugrunde, dass die zu approximierende Funktion ein Gauß-Prozess ist. Das bedeutet insbesondere, dass die einzelnen Funktionswerte normalverteilt sind. Die Gütefunktion für die Bewertung der Greifbewegungen erfüllt diese Bedingung jedoch nicht, so dass die gesamte Gütefunktion nicht durch einen einzelnen Gauß-Prozess repräsentiert werden kann. Daher werden in dieser Diplomarbeit drei verschiedene Gauß-Prozesse verwendet, um die drei möglichen Fälle (siehe Gleichung (5.1)) zu repräsentieren. Ein Prozess speichert die Gütewerte abhängig vom Versatz der Tasse, nachdem diese gegriffen wurde. Zwei weitere speichern die Bewertung der Greifbewegung, falls die Tasse verfehlt wurde oder Probleme mit Kollisionen oder der inversen Kinematik auftraten. Da in dieser Diplomarbeit Gauß-Prozesse mit einem a priori Mittelwert von 0 verwendet werden, wird bei Problemen ein Gütewert von 0 statt -1 in dem entsprechenden Gauß-Prozess gespeichert.

Jeder Gauß-Prozess liefert im Vorhersageschritt einen Mittelwert und eine Unsicherheit für einen angefragten Punkt im Parameterraum zurück. Um zu bestimmen, wie gut und sicher ein Parametersatz ist, müssen alle Beispiele aus allen drei Gauß-Prozessen in die Bewertung einfließen. Dazu wird ein kombinierter Mittelwert und eine zugehörige Unsicherheit benötigt. Für eine bessere Übersichtlichkeit werden die drei Kostenfälle aus Gleichung (5.1) durch die Indizes +, - und 0 dargestellt:

$$\mu = \frac{\kappa_-}{\kappa} \cdot (\mu_- - 1.0) + \frac{\kappa_0}{\kappa} \cdot \mu_0 + \frac{\kappa_+}{\kappa} \cdot \mu_+ \tag{5.2}$$

mit

$$\kappa = \kappa_- + \kappa_0 + \kappa_+$$

Dabei handelt es sich bei κ um die Konfidenz des Parametersatzes. Sie gibt die Sicherheit der angefragten Parameter an. Sind diese unbekannt so geht der Konfidenzwert gegen 0, sind sie bekannt so ist der Wert 1.

$$\kappa_i = 1 - \frac{\sigma}{\sqrt{k_{**_i} + s_{0_i}^2}} \qquad i \in \{+, 0, -\} \tag{5.3}$$

Der kombinierte Mittelwert ist somit die Summe der Mittelwerte aller drei Prozesse, gewichtet mit ihrem Anteil an der Summe der Konfidenz. Der Gauß-Prozess, dessen Schätzung die höchste Sicherheit aufweist, hat den größten Einfluss auf den Mittelwert μ. Da für den Gütewert -1 bei Kollisionen mit dem Tisch oder Problemen mit der inversen Kinematik der Wert 0 als Güte gespeichert wurde, muss dieser Versatz auch in Gleichung 5.2 berücksichtigt werden. Passend zum Mittelwert μ wird, ebenfalls durch Gewichtung mit der Konfidenz, ein kombinierter Wert für σ bestimmt.

$$\sigma^2 = \left(\frac{\kappa_-}{\kappa}\right)^2 \cdot \sigma_-^2 + \left(\frac{\kappa_0}{\kappa}\right)^2 \cdot \sigma_0^2 + \left(\frac{\kappa_+}{\kappa}\right)^2 \cdot \sigma_+^2 \tag{5.4}$$

5.3 Entscheidung für ein Lernverfahren

Die Entscheidung, welches Teilverfahren gewählt wird um eine Tasse an einer bestimmten Position zu greifen hängt davon ab, wie gut die Tasse dort vermutlich gegriffen werden kann. Dies lässt sich anhand der Mittelwerte und Unsicherheiten der Gütewerte ermitteln, die die Gauß-Prozesse für diese Position vorhersagen. Dabei ist eine Prädiktion immer nur für vollständige Parametersätze möglich. Zu der Position der Tasse müssen also zunächst geeignete Werte für die restlichen der in Tabelle 4.1 aufgelisteten Parameter gefunden werden. Dazu wird, ausgehend von den Trainingsbeispielen, mit einem Optimierungsverfahren gearbeitet. Bei den meisten Optimierungsverfahren ist es wichtig, dass geeignete Initalisierungswerte bestimmt werden, so auch bei dem in dieser Diplomarbeit benutzten Gradientenabstiegsverfahren Rprop.

Geeignete Startparameter werden unter denjenigen Parametersätzen gesucht, mit denen bisher erfolgreich gegriffen wurde. Im Gegensatz zu einer zufälligen Wahl ist bei den bereits gelernten Parametersätzen sichergestellt, dass sie anthropomorphe Greifbewegungen erzeugen. Aus diesen Parametersätzen wird derjenige ausgewählt, der den besten Mittelwert aufweist und dessen Position der Tasse (x_P, y_P) besonders nah an der aktuellen Situation (x_A, y_A) liegt. Somit muss folgende Gleichung minimiert werden:

$$f(\vec{x}) = (1.0 - \mu(\vec{x})) + \sqrt{(x_A - x_P)^2 + (y_A - y_P)^2} \tag{5.5}$$

Die Parameter des besten Beispieles werden bis auf die Zielposition als Initalisierungswerte übernommen. Die Zielposition wird aus der aktuellen Position der Tasse bestimmt.

Falls noch keine erfolgreiche Greifbewegung durchgeführt wurde, und somit noch kein positives Beispiel vorhanden ist, fällt die Entscheidung direkt auf das Lernen durch Imitation.

Andernfalls wird ein Optimierungsverfahren benutzt, um die aus den Beispielen ausgewählten Initialisierungswerte an die aktuelle Tassenposition anzupassen und sie zu verbessern. Dabei werden von den 31 Eingabewerten lediglich 4 Parameter optimiert, die maßgeblich über das Gelingen der Greifbewegung entscheiden. Zu diesen gehören die Offsetwerte der x- und y-Koordinaten der Zielposition. Diese werden benötigt, um Fehler in der Lasermessung oder der Motorik des Roboters auszugleichen. Desweiteren wird die Öffnung der Hand optimiert, die durch zwei weitere Parameter beschrieben wird. Da diese Parameter auf die Position der Tasse des besten Beispiels ausgerichtet sind, müssen diese für die aktuelle Position angepasst werden.

Die anderen Parameter werden festgehalten. Sie sind vor allem ausschlaggebend für menschlich wirkende Bewegungen, haben aber einen vergleichsweise geringen Einfluss auf das Gelingen der Greifbewegung. Somit hat auch das Optimierungskriterium keinen Einfluss auf diese Parameter. Würden auch sie optimiert werden, so könnte nicht garantiert werden, dass die resultierenden Bewegungen noch menschlich wirken.

Bei dem verwendeten Optimierungsverfahren Rprop handelt es sich um eine erweiterte Gradientenabstiegsmethode. Hierbei wird im Gegensatz zu anderen Gradientenabstiegsverfahren die Schrittweite zur Änderung der Parameter unabhängig vom Betrag des Gradienten bestimmt. Stattdessen wird diese über den zeitlichen Verlauf des Gradienten errechnet. Die Arbeitsweise von Rprop ist in Abschnitt 2.2.2 detailliert beschrieben.

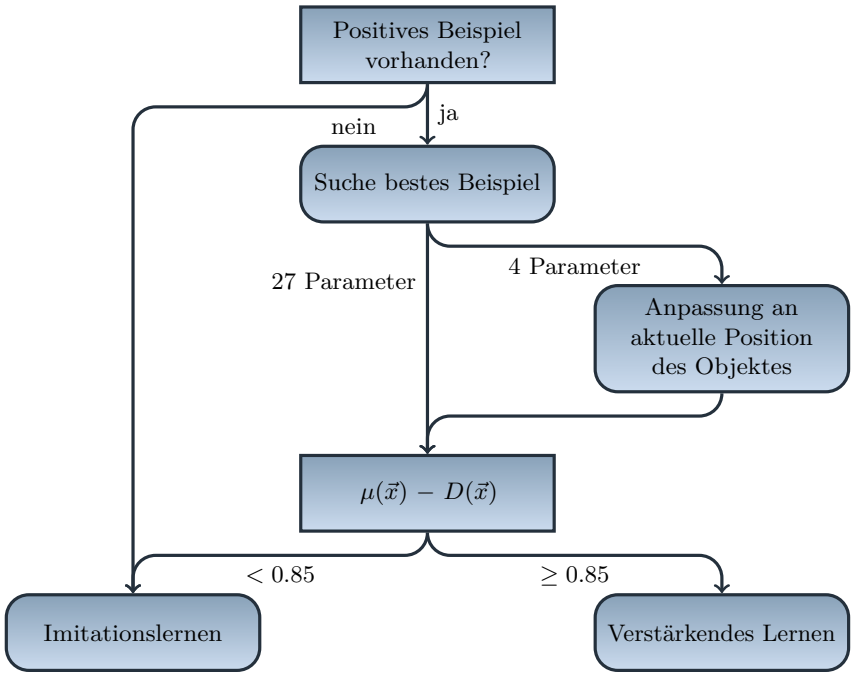


Abbildung 5.2: *Entscheidung für Imitations- oder verstärkendes Lernen*

Ausgehend von der zuvor gefundenen Initialisierung wird mit Rprop nach Parametern gesucht, deren Gütewerte hoch sind und die gleichzeitig eine geringe Standardabweichung aufweisen. Eine geringe Standardabweichung gibt an, dass es ein Trainingsbeispiel mit ähnlichen Parametern gibt. Sind auch die Werte der Güte an dieser Stelle hoch, ist das Trainingsbeispiel gut gewesen. In diesem Fall ist zu erwarten, dass der Roboter auch mit den gefundenen, ähnlichen Parametern die Tasse gut greifen kann, ohne dabei Schaden zu nehmen. Diese Parameter werden bestimmt, indem folgende Funktion maximiert wird:

$$f(\vec{x}) = \mu(\vec{x}) - \sigma(\vec{x}) \tag{5.6}$$

Das Verfahren Rprop benötigt dafür die partiellen Ableitungen der Funktion f.

$$\frac{\partial f(\vec{x})}{\partial \vec{x}} = \frac{\partial \mu(\vec{x})}{\partial \vec{x}} - \frac{\partial \sigma(\vec{x})}{\partial \vec{x}} \tag{5.7}$$

Da sich μ und σ gemäß (5.2) und (5.4) aus drei Komponenten zusammensetzen, gilt dies auch für ihre partiellen Ableitungen.

$$\begin{aligned}\frac{\partial \mu}{\partial \vec{x}} = &\left(\frac{\partial \kappa_-}{\partial \vec{x}} \cdot \kappa - \frac{\partial \kappa}{\partial \vec{x}} \cdot \kappa_-\right) \cdot \kappa^{-2} \cdot (\mu_- - 1.0) + \frac{\kappa_-}{\kappa} \cdot \frac{\partial \mu_-}{\partial \vec{x}} \\ &+ \left(\frac{\partial \kappa_0}{\partial \vec{x}} \cdot \kappa - \frac{\partial \kappa}{\partial \vec{x}} \cdot \kappa_0\right) \cdot \kappa^{-2} \cdot \mu_0 + \frac{\kappa_0}{\kappa} \cdot \frac{\partial \mu_0}{\partial \vec{x}} \\ &+ \left(\frac{\partial \kappa_+}{\partial \vec{x}} \cdot \kappa - \frac{\partial \kappa}{\partial \vec{x}} \cdot \kappa_+\right) \cdot \kappa^{-2} \cdot \mu_+ + \frac{\kappa_+}{\kappa} \cdot \frac{\partial \mu_+}{\partial \vec{x}}\end{aligned} \tag{5.8}$$

$$\begin{aligned}\frac{\partial\sigma}{\partial x} = \frac{1}{2}\cdot\sigma\cdot\Bigg[&2\cdot\frac{\kappa_-}{\kappa}\cdot\left(\frac{\partial\kappa_-}{\partial\vec{x}}\cdot\kappa-\frac{\partial\kappa}{\partial\vec{x}}\cdot\kappa_-\right)\cdot\kappa^{-2}\cdot\sigma_-^2+\left(\frac{\kappa_-}{\kappa}\right)^2\cdot\frac{\partial\sigma_-^2}{\partial\vec{x}}\\&+2\cdot\frac{\kappa_0}{\kappa}\cdot\left(\frac{\partial\kappa_0}{\partial\vec{x}}\cdot\kappa-\frac{\partial\kappa}{\partial\vec{x}}\cdot\kappa_0\right)\cdot\kappa^{-2}\cdot\sigma_0^2+\left(\frac{\kappa_0}{\kappa}\right)^2\cdot\frac{\partial\sigma_0^2}{\partial\vec{x}}\\&+2\cdot\frac{\kappa_+}{\kappa}\cdot\left(\frac{\partial\kappa_+}{\partial\vec{x}}\cdot\kappa-\frac{\partial\kappa}{\partial\vec{x}}\cdot\kappa_+\right)\cdot\kappa^{-2}\cdot\sigma_+^2+\left(\frac{\kappa_+}{\kappa}\right)^2\cdot\frac{\partial\sigma_+^2}{\partial\vec{x}}\Bigg]\end{aligned} \tag{5.9}$$

Die darin enthaltenen Ableitungen sind gegeben durch

$$\frac{\partial\mu_i}{\partial\vec{x}} = \frac{\partial K_{*_i}^T}{\partial\vec{x}}\cdot C_i^{-1}\cdot y_i \tag{5.10}$$

$$\frac{\partial\sigma_i^2}{\partial\vec{x}} = 2\cdot\sigma_i\cdot-\left(\frac{\partial K_{*_i}^T}{\partial\vec{x}}\cdot C_i^{-1}\cdot K_{*_i}\right)/\sigma_i \tag{5.11}$$

$$\frac{\partial\kappa_i}{\partial\vec{x}} = -\frac{1}{K_{**_i}+s_{i0}^2}\cdot\frac{\partial\sigma_i}{\partial\vec{x}} \tag{5.12}$$

$$\frac{\partial\kappa}{\partial\vec{x}} = \sum_{i=1}^{3}\left(-\frac{1}{K_{**_i}+s_{i0}^2}\cdot\frac{\partial\sigma_i}{\partial\vec{x}}\right) \tag{5.13}$$

$$\frac{\partial K_*^T}{\partial\vec{x}} = \begin{bmatrix}\frac{\partial k(\vec{x},\vec{x_1})}{\partial x_1} & \cdots & \frac{\partial k(\vec{x},\vec{x_N})}{\partial x_1}\\ \vdots & \ddots & \vdots\\ \frac{\partial k(\vec{x},\vec{x_1})}{\partial x_D} & \cdots & \frac{\partial k(\vec{x},\vec{x_N})}{\partial x_D}\end{bmatrix} \tag{5.14}$$

Dabei steht i für die verschiedenen Fälle +, - und 0. Die Abhängigkeiten von $\vec{x}$ wurden der Übersichtlichkeit halber nicht explizit dargestellt. Die Funktion $k(x,x')$ ist, wie in Kapitel 2.1, die Kovarianzfunktion des Gauß-Prozesses. Im Rahmen dieser Diplomarbeit wird ein RBF-Kernel verwendet, dessen Ableitung wie folgt gegeben ist:

$$\frac{\partial k(\vec{x},\vec{x}')}{\partial x_j} = \frac{\partial}{\partial x_j}\prod_i e^{\frac{(x_i-x_i')^2}{2\sigma^2}} \tag{5.15}$$

$$= \frac{(x_j-x_j')^2}{\sigma^2}\cdot\prod_i e^{\frac{(x_i-x_i')^2}{2\sigma^2}} \tag{5.16}$$

Nachdem Rprop die besten lokalen Parameter dem Gütekriterium entsprechend gefunden hat, wird anhand dieser entschieden, welches Teilverfahren genutzt wird. Diese Entscheidung hängt von der erwarteten Güte des gefundenen Parametersatzes und der erwarteten Verschlechterung für diesen ab. Die erwartete Verschlechterung $D(\vec{x})$ gibt an, um wieviel sich der Parametersatz aufgrund seiner Standardabweichung erwartungsweise von seinem Mittelwert verschlechtern könnte (siehe Gleichung (5.19)). Falls für den bestimmten Parametersatz der schlechteste zu erwartende Gütewert $\mu(\vec{x})-D(\vec{x})$ unter einer empirisch bestimmten Grenze von 0,85 liegt, so deutet dies darauf hin, dass der Roboter keine gute Lösung kennt. Daher wird in diesem Fall Imitationslernen als Verfahren gewählt. Wird eine gute Lösung erwartet, die keine große Unsicherheit aufweist, so darf

der Roboter seine eigenen Erfahrungen machen, indem er Parameter wählt, die für gut gehalten werden, und diese ausführt. Dieses Lernen aus Erfahrung ist als verstärkendes Lernen implementiert.

5.4 Imitationslernen

Menschen lernen viele ihrer Fähigkeiten durch Imitation. In der Psychologie wird dies auch als Lernen am Modell oder Beobachtungslernen bezeichnet. Neben klassischer und operanter Konditionierung ist Imitationslernen die dritte klassische Form des menschlichen Lernens. Darunter versteht man den Erwerb von Fähigkeiten durch Beobachtungen und Nachahmung. Durch Imitationslernen ist der Mensch in der Lage, sich komplexe Verhaltensweisen anzueignen. Da diese Art des Lernens besonders schnell und effizient ist, ist sie bei allen Menschen jeden Alters die meist genutzte.

Die Erforschung des Lernens ist nicht nur Gegenstand der Psychologie. Auch in der Robotik spielen Lernverfahren eine zentrale Rolle. Aufgrund der vielen Vorteile des Imitationslernens hat die Bedeutung dieses Verfahrens für die Robotik in den letzten Jahren stark zugenommen. Bewegungen müssen dadurch nicht mehr aufwendig programmiert werden, sondern können durch Imitation trainiert werden.

Besonders schwer ist es, menschlich wirkende Bewegungen zu erschaffen, da diese viele Freiheitsgrade aufweisen, die teilweise redundant sind. Zusätzlich gibt es eine große Vielfalt an Bewegungen. Durch Nachahmen der menschlichen Bewegungen können solche komplexen Fähigkeiten auf eine einfache Weise auf einen Roboter übertragen werden. Imitation stellt somit einen vielversprechenden Weg dar, um anthropomorphe Bewegungen zu generieren. Ein weiterer Vorteil ist, dass sich der Trainer im Umgang mit dem Roboter genauso verhalten kann, wie er es im Umgang mit einem Menschen tun würde. Dies erleichtert die Arbeit mit dem Roboter und ermöglicht auch Nicht-Experten, auf eine einfache und für sie natürliche Art und Weise einen Roboter zu trainieren.

Ein weiterer Vorteil von Imitationslernen ist, dass es auch in Kombination mit anderen Verfahren eingesetzt werden kann. Beispielsweise kann durch Übernahme menschlichen Wissens der Suchraum von verstärkendem Lernen eingeschränkt und somit die Geschwindigkeit des Lernverfahrens erhöht werden.

Im Rahmen dieser Diplomarbeit soll ein Roboter durch Imitation menschliche Greifbewegungen erlernen. Dafür werden aus den vom Menschen vorgeführten Bewegungen Parameter eines Reglers gelernt, die es diesem erlauben, die vorgeführte Bewegung zu generieren.

5.4.1 Vorverarbeitung der Motion Capture-Daten

Um die demonstrierten Bewegungen des Menschen erfassen zu können, werden in dieser Diplomarbeit eine Motion Capture-Anlage und ein Datenhandschuh verwendet. Bei beiden handelt es sich um Bewegungserfassungssysteme. Die Motion Capture-Anlage kann Marker mit Hilfe von Infrarotkameras innerhalb eines eingeschränkten räumlichen Bereichs erkennen. Um die Bewegungen eines Menschen aufzunehmen, muss dieser einen schwarzen Anzug tragen, der an bestimmten Stellen mit Markern ausgestattet ist. Die Positionen der Marker werden anschließend durch die Software der Motion Capture-

Anlage Positionen auf einem vordefinierten menschlichen Skelett zugeordnet. Auf diese Weise können Bewegungen des gesamten menschlichen Körpers erfasst werden. Das vorgefertigte Skelett der in dieser Arbeit verwendeten Motion Capture-Anlage enthält allerdings keine Finger. Somit kann durch die Anlage zwar die Pose der Hand bestimmt werden, jedoch nicht ob diese geöffnet oder geschlossen ist. Um die Beugung der Finger zu erfassen, müssten zusätzlich auch an ihnen Marker angebracht werden. Da diese dann sehr nah aneinander liegen würden und alle identisch sind, wäre es für die Motion-Capture-Anlage sehr schwierig, die einzelnen Marker zu erkennen und auseinanderzuhalten. Um dieses Problem zu vermeiden, werden diese Daten separat mit einem Datenhandschuh aufgenommen. Dieser arbeitet mit je einem Dehnungsmessstreifen pro Finger, der die Beugung des jeweiligen Fingers misst. Eine detaillierte Beschreibung der Motion Capture-Anlage und des Datenhandschuhs befindet sich im Anhang A.

Datenerfassung

Über eine in dieser Diplomarbeit entwickelte Client-Server Architektur können die Bewegungsdaten der Motion Capture-Anlage und des Datenhandschuhs in Echtzeit abgefragt werden. Die Rolle des Servers, der die Daten der Motion Capture-Anlage verschickt, übernimmt die dazugehörige Software *Arena*, die im Abschnitt B.1 beschrieben ist. Die Daten des Handschuhs werden durch einen eigens implementierten Server zur Verfügung gestellt. Die Software Arena verwendet zum Versenden der Skelettinformationen ein binäres Protokoll, dessen Grundzüge in einer frei verfügbaren Entwicklerdokumentation beschrieben sind [37]. Diese Informationen enthalten unter anderem:

- Anzahl und Namen der aufgenommenen Skelette
- Anzahl und Positionen der zum Skelett gehörenen Marker
- Anzahl und Position zusätzlich erkannte Marker
- Anzahl, Ids, Positionen und Orientierungen der einzelnen Körperglieder der Skelette

Die Daten der Server werden durch zwei in dieser Diplomarbeit entwickelte Client-Anwendungen entgegengenommen und weiterverarbeitet. Nach Zuordnung der Identifikationsnummern aus den Datenpaketen zu den einzelnen Körpergliedern, werden nur diejenigen Daten weiter bearbeitet, die zur Hand, zur Hüfte und zur Tasse gehören. Die Hüfte wird dabei als Mittelpunkt des menschlichen Körpers angenommen. Zusätzlich wird die Empfangszeit gespeichert, da keine Zeitangaben in den Daten von der Motion Capture-Anlage enthalten sind.

Um die aufgenommenen Daten für die Simulation oder den Roboter benutzen zu können, müssen sie zunächst, wie in Abbildung 5.3 dargestellt, vom Koordinatensystem der Motion Capture-Anlage in das Roboter-Koordinatensystem überführt werden. Dazu muss das Koordinatensystem sowohl gedreht und gespiegelt, als auch sein Ursprung verschoben werden. Dabei wird für die Simulation und den realen Roboter dasselbe System verwendet. Der Nullpunkt dieser beiden Systeme befindet sich im Schultergelenk des Roboterarms. Der Nullpunkt der Motion Capture-Anlage liegt allerdings an einem im Motion Capture-Raum festgelegten Punkt auf dem Boden. Um beide Nullpunkte in einem zu vereinigen, wird der Nullpunkt der Motion Capture-Anlage zunächst in den

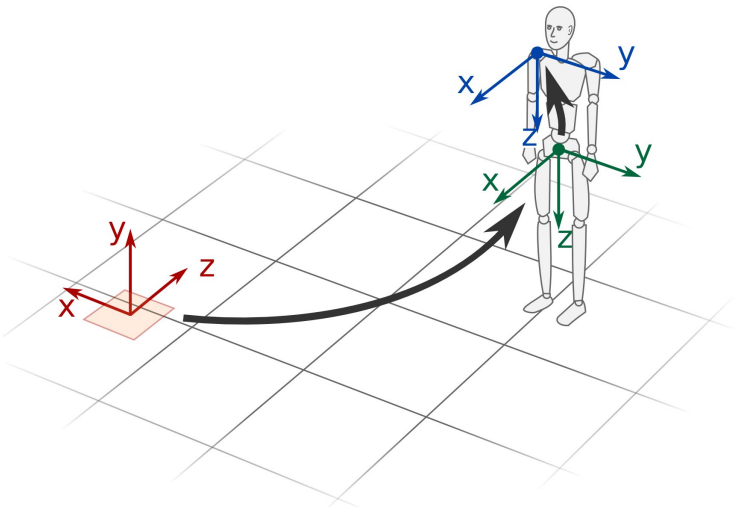


Abbildung 5.3: *Transformation vom Koordinatensystem der Motion Capture-Anlage in die Schulter des Menschen bzw. des Roboters*

Körpermittelpunkt mit Hilfe der Hüftposition in x- und y-Richtung verschoben. Von diesem Punkt ist die Verschiebung in das Schultergelenk bekannt. Durch dieses Verfahren kann der Punkt im Raum, an dem der Demonstrator die Aufnahme macht, jedes Mal frei gewählt werden und unterliegt keinen Einschränkungen. Die Maßeinheit wird einheitlich auf Meter festgelegt und gegebenenfalls umgerechnet.

Der Datenhandschuh übermittelt den Grad der Handöffnung mit einer Auflösung von 6 Bit pro Finger. Ist ein Finger geknickt, so liefert er die Zahl 63, ist ein Finger gestreckt eine 0. Simulation und Roboter benötigten Fließkommazahlen zwischen 0 und 1, die angeben, wie weit der Greifer geöffnet ist. Um die Werte der 5 Finger in eine Öffnung der Hand umzurechnen, werden sie aufsummiert und auf den Wertebereich von 0 bis 1 normiert. Dazu wurde zuvor bestimmt, bei welchen Werten die Hand des Menschen geöffnet bzw. geschlossen ist, was für den Roboter den Werten 0 bzw. 1 entspricht. Zwischen den beiden Werten wurde linear interpoliert.

Bewegungssegmentierung

Bei der Aufnahme von Motion Capture-Daten wird zur groben Begrenzung der Bewegungsaufzeichnung jeweils am Anfang und am Ende der Aufnahme ein Knopf gedrückt. Direkt nach Ende der Aufnahme hat der Benutzer die Möglichkeit, eine fehlerhafte Trajektorie zu verwerfen. Zusätzlich werden die aufgenommenen Bewegungen automatisch überprüft und verworfen, falls die Zahl der Punkte in der Trajektorie nicht ausreichend ist, oder die Trajektorie große Lücken aufweist.

Diese grobe Segmentierung der Bewegung ist nicht ausreichend. Da zu Beginn und am Ende einer Aufnahme die Hand häufig über längere Zeit an einer Stelle gehalten wird, entstehen an diesen Stellen oft verrauschte Daten, die nicht zu der Trajektorie gehören. Dem Lernalgorithmus ist daher ein Verfahren vorgeschaltet, das diese verrauschten Daten automatisch entfernt. Dies ist für die überflüssigen Punkte am Ende ohne großen Aufwand

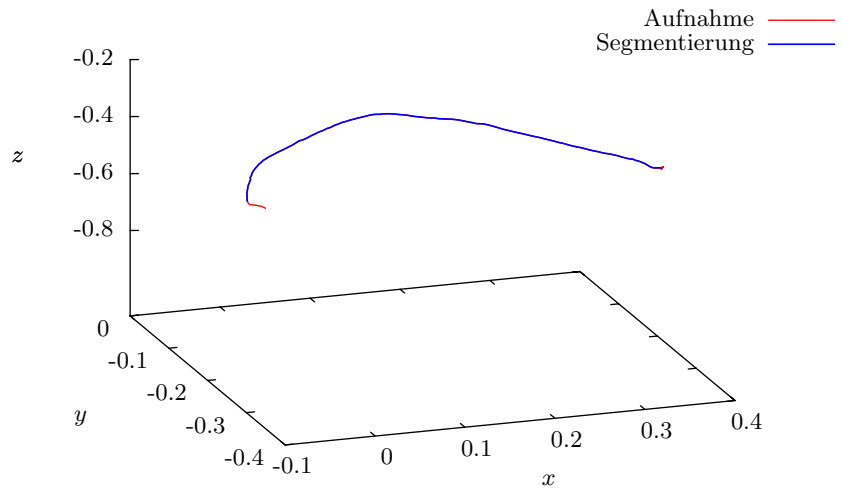


Abbildung 5.4: *Segmentierung der Bewegung, um nicht zur Bewegung gehörende Punkte zu entfernen*

möglich. Dazu wird, beginnend von dem letzten Punkt der Trajektorie, der Schwerpunkt aller Punkte, die sich innerhalb eines gewissen Fensters befinden bestimmt. Ausgehend von diesem Punkt wird erneut ein Fenster gewählt und nach derselben Methode ein neuer Schwerpunkt berechnet. Dabei werden die zuvor verwendeten Punkte nicht mehr berücksichtigt. Dies wird so lange wiederholt, bis die dadurch entstehenden Schwerpunkte konvergieren. Die Fensterbreite beträgt 1cm in jede Richtung.

Der Anfang der eigentlichen Trajektorie zeichnet sich dadurch aus, dass die z-Koordinate wächst und die Geschwindigkeit größer als ein bestimmter Schwellwert wird. Um diesen Anfang zu finden, wird zunächst der erste Punkt mit einer entsprechend hohen Geschwindigkeit gesucht. Dabei werden die Geschwindigkeiten der Punkte in einer gewissen Umgebung geglättet, um Rauschen zu unterdrücken. Ist dieser Punkt gefunden, wird von ihm aus rückwärts der Beginn der Bewegung gesucht. Das bedeutet, dass in Richtung des Anfangs der Trajektorie die Geschwindigkeiten aller Punkte mit einem niedrigen Schwellwert verglichen werden, bis ein Punkt gefunden wird, dessen Geschwindigkeit unter dem Schwellwert liegt. Dieser Punkt wird als vorläufiger Anfang der Trajektorie angenommen. Alle früheren Punkte werden verworfen. In einem zweiten Schritt wird überprüft, ob das Höhenkriterium an dem gefundenen Anfangspunkt erfüllt ist. Ist dies nicht der Fall, wird der erste Punkt gesucht, der das Kriterium erfüllt und dieser als Anfang der Trajektorie festgelegt. Das Ergebnis dieser automatischen Segmentierung ist die Trajektorie der Greifbewegung, aus der die Parameter des Reglers bestimmt werden. Ein Beispiel einer solchen Trajektorie ist in Abbildung 5.4 abgebildet.

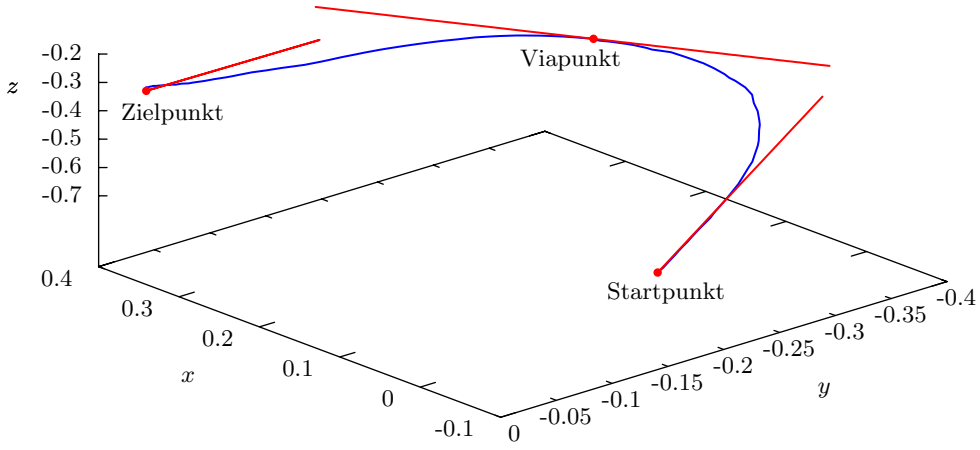


Abbildung 5.5: *Lage von Start-, Via- und Zielpunkt und zugehörige Richtungen der Bewegung*

5.4.2 Parameterextraktion

Um die 29 für den Regler benötigten Parameter aus dieser Trajektorie zu extrahieren, werden zwei unterschiedliche Verfahren angewandt. Einige Parameter werden direkt aus den Koordinaten der Trajektorie berechnet, die anderen sind nur durch iterative Optimierung bestimmbar.

Direkt berechenbare Parameter

Bei den direkt bestimmbaren Parametern handelt es sich hauptsächlich um kinematische Parameter, deren Einfluss auf den Verlauf der Trajektorie für den Menschen offensichtlich ist. Diese Parameter können durch geschlossene Formeln berechnet werden, in die ausschließlich die aufgenommenen Punkte der Trajektorie eingehen. Direkt bestimmbar sind:

- der Start-, Via- und Zielpunkt
- die Bewegungesrichtungen an diesen drei Punkten
- die Öffnung der Hand am Viapunkt
- die maximale Öffnung der Hand über die gesamte Trajektorie
- die Distanz zum Ziel, wenn die Hand maximal geöffnet ist

Bestimmung des Start-, Via- und Zielpunktes

Durch die Vorverarbeitung der Trajektorie, bei der die überflüssigen Punkte am Anfang und am Ende entfernt worden sind, sind Start-und Zielpunkt der Trajektorie nun einfach zu bestimmen. Es handelt sich dabei um die Position der Hand im ersten bzw. letzten

Frame. Der Viapunkt ist ein Zwischenziel, welches die Trajektorie auf dem Weg vom Start- zum Zielpunkt erreichen soll. Daher ist er nicht eindeutig festgelegt und es gibt verschiedene Kriterien, wie er gewählt werden kann. Die Parameter, die später als Eingabe für den Regler dienen, sollten so gewählt sein, dass sie die demonstrierte Trajektorie besonders gut charakterisieren. Solche Eigenschaften könnten für den Viapunkt die Höhe, die Geschwindigkeit oder die Krümmung an diesem Punkt sein. Bei empirischen Untersuchungen hat sich der höchste Punkt der Trajektorie als bestes Kriterium zur Generierung von Greifbewegungen herausgestellt.

Bestimmung der Richtungen an Start-, Via- und Zielpunkt

Zur Bestimmung der Richtungen werden die Geschwindigkeiten aller Punkte der Trajektorie in einem bestimmten Fenster um den Punkt, dessen Richtung bestimmt werden soll, berechnet. Die Punkte fließen abhängig von ihrer jeweiligen Distanz zu diesem Punkt in die Geschwindigkeit ein. Befindet sich kein Punkt in diesem Fensterbereich, so werden die beiden nächsten Punkte zur Geschwindigkeitsberechnung benutzt. Die Richtung ergibt sich aus der Geschwindigkeit durch anschließendes Normieren.

Bestimmung der Parameter zur Öffnung der Hand

Die Öffnung der Hand am Viapunkt lässt sich direkt aus den Daten ablesen. Da das Öffnen der Hand bei der menschlichen Greifbewegung erst zum größten Teil nach dem Viapunkt stattfindet, reicht es aus, die maximale Öffnung der Hand erst nach diesem Punkt zu suchen. Für die Interpolation der Öffnung der Hand ist es nicht nur wichtig zu wissen, wie weit sich die Hand maximal öffnet, sondern auch, wo sie die maximale Öffnung annimmt. Dabei müssen jedoch nicht alle Koordinaten des Punktes gespeichert werden, sondern es genügt, die Entfernung zum Ziel festzuhalten.

Durch die Konstruktion des in dieser Diplomarbeit verwendeten Roboters werden zusätzlich zu den 29 Parametern des Reglers zwei Offsetwerte benötigt. Diese geben einen Versatz der Zielposition in x- und y-Richtung an. Die in der Motion Capture-Anlage ausgeführte Bewegung ist eine natürliche Greifbewegungen eines Menschen und ohne Berücksichtigung der Anatomie des Roboters entstanden. Aus dieser Bewegung werden die Parameter generiert, die im Regler ausgeführt die gleiche Bewegung auf dem Roboter erzeugen sollen. Wäre die Kinematik des Roboters exakt, besäße der Roboter eine menschenähnliche Hand und würde der Laser die Position der Tasse genau messen, könnte der Roboter mit diesen Parametern die Tasse ebenso gut greifen wie es der Mensch vorgemacht hat. In der Praxis ist dies jedoch nicht der Fall, so dass ein Offsetvektor gelernt werden muss. Im Regler wird dieser zur Zielposition dazuaddiert. Er ist abhängig von dem systematischen Fehler des Lasers und der Ungenauigkeit der Roboter-Kinematik, die je nach Position des zu greifenden Objekts unterschiedlich sein kann. Zusätzlich hängt der Versatz der Tasse von der Öffnung der Hand ab, die nicht mit der des Menschen übereinstimmt. Die Zielposition des Roboters ergibt sich somit durch Verschiebung der gemessenen Position durch die Offsetwerte. Diese werden durch das Imitationslernen zunächst mit 0 initialisiert und durch das verstärkende Lernen optimiert.

Iterativ bestimmbare Parameter

Von den 29 Parametern des Reglers können 8 nicht direkt berechnet werden, da es für diese Parameter keine geschlossene Formel gibt. Dies hängt unter anderem damit zusammen, dass Begrenzungen in Geschwindigkeit und Beschleunigung eingehalten werden müssen. In dieser Diplomarbeit werden sie daher iterativ durch ein Optimierungsverfahren bestimmt. Dazu wird eine Kostenfunktion definiert, die eine Ähnlichkeit der generierten Trajektorie zu der demonstrierten Trajektorie misst. Um die Kosten für einen Parametersatz bestimmen zu können, werden zunächst die zuvor direkt berechneten Parameter an den Regler gegeben. Zusammen mit den 8 zu testenden Parametern, bilden sie die vollständige Eingabe für den Regler. Der Regler bestimmt aus den Parametern eine Trajektorie, wobei davon ausgegangen wird, dass die gewünschten Zwischenziele auch erreicht werden. Die Ähnlichkeit zwischen dieser generierten Trajektorie und der demonstrierten Trajektorie wird aus zwei Teilen berechnet:

- der Dauer der Bewegungen
- der mittleren quadratischen Abweichung

Daraus ergibt sich die Kostenfunktion

$$f(T_D, T_R) = \left(1 - \frac{t_D}{t_R}\right)^2 \cdot \lambda + dist(T_D, T_R) \tag{5.17}$$

mit den demonstrierten und duch den Regler generierten Trajektoiren T_D und T_R, ihren Längen t_D und t_R und dem Abstandsmaß $dist(\cdot, \cdot)$. Für den Gewichtungsfaktor λ zwischen Bestrafung der Dauer und der räumlichen Abweichung wurde in dieser Diplomarbeit der Wert 10^{-4} gewählt.

Da sowohl die zeitliche Dauer der generierten Trajektorie, als auch die der demonstrierten Trajektorie bekannt ist, kann der Zeitunterschied zwischen diesen Beiden leicht bestimmt werden. Für die mittlere quadratische Abweichung wird eine Punkt-zu-Geraden-Metrik bestimmt, die in Abbildung 5.6 veranschaulicht ist. Dazu wird für jeden Punkt q der generierten Trajektorie der Punkt p auf der demonstrierten Bewegung gesucht, der diesem am nächsten ist. Durch diesen Punkt und seine beiden Nachbarn wird je eine Gerade gelegt. Der Punkt q der generierten Trajektorie wird sowohl auf die Gerade zwischen den Punkten p und $p-1$, als auch auf die zwischen p und $p+1$ projiziert. Die Projektion mit der kleineren Distanz gibt die Abweichung an diesem Punkt q an. Über alle Punkte normiert, entsteht die mittlere quadratische Abweichung. Um eine möglichst große Ähnlichkeit zwischen den beiden Trajektorien zu erreichen, wird die Summe der beiden Komponenten minimiert. Der Versuch eine zeitliche Zuordnung für die Bestimmung dieser Abweichung zu verwenden, lieferte keine guten Ergebnisse, da die Zeit durch diesem Ansatz zu stark gewichtet wurde.

Diese Kostenfunktion ist sehr komplex, da für jede Evaluierung eine neue Trajektorie durch den Regler generiert werden muss. Dadurch kann auch der Gradient der Funktion nicht bestimmt werden. Dies muss bei der Wahl des Optimierungsverfahrens berücksichtigt werden. In dieser Diplomarbeit wird das Downhill-Simplex-Vefahren verwendet, das in Abschnitt 2.2.1 beschrieben ist. Dabei werden die Startwerte der Parameter von Hand auf die richtige Größenordnung eingestellt und gleichzeitig das Startsimplex groß gewählt, um Probleme durch eine Initialisierung in einem Nebenminimum zu vermeiden.

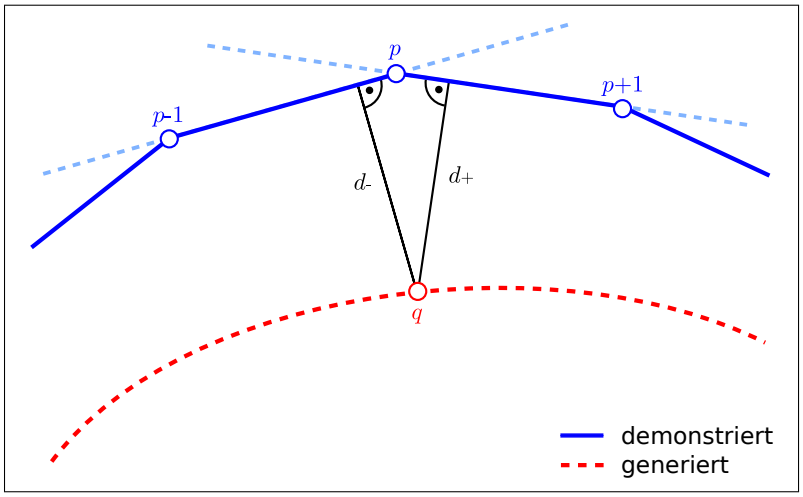


Abbildung 5.6: *Berechnung des Abstandes eines Punktes auf der generierten Trajektorie des Reglers zu der demonstrierten Trajektorie. Dargestellt sind die Lotfußpunkte von q auf den Geraden $\overline{p-1,p}$ und $\overline{p,p+1}$ und die Abstände von q zu diesen Punkten. Der Abstand von q zur aufgenommenen Trajektorie wird als das Minimum von d_- und d_+ definiert.*

5.5 Verbesserung durch verstärkendes Lernen

Ebenso wie Imitationslernen, ist auch das Lernen aus Erfahrungen ein für den Menschen typisches Lernverfahren. Hierbei lernt der Mensch durch die Reaktion seiner Umwelt auf seine Aktionen. Er erfährt dabei nicht, wie er eine Aufgabe richtig gelöst hätte, sondern nur, ob die Aufgabe gut oder schlecht gelöst wurde. Dies kann entweder durch eine außenstehende Person geschehen, die das Ergebnis der Aktionen bewertet, oder durch Selbstreflektion, indem der Mensch selbst das Ergebins seiner Aktion wahrnimmt und bewertet.

Dieses Prinzip des Lernens aus Belohnung und Bestrafung steht im Gegensatz zum Lernen durch Imitation, bei dem die richtige Lösung bekannt ist. Bei verstärkendem Lernen handelt es sich somit um einen Mittelweg zwischen überwachtem Lernen und unüberwachtem Lernen.

Auch in der Robotik wird verstärkendes Lernen schon seit Mitte der 50er Jahre erforscht [38]. Im Gegensatz zu überwachten Lernverfahren bietet es den Vorteil, dass keine Lösungen für das zu lernende Problem vorliegen müssen. Stattdessen genügt es, wenn der Roboter eine Bewertung seiner Lösungen erhält. Diese kann durch einen Menschen gegeben oder ohne menschliches Zutun durch den Roboter selbst ermittelt werden, z.B. indem dieser die Folgen seiner Aktionen auf die Umwelt mit seinen Sensoren misst. Dies ist in dieser Diplomarbeit der Fall. Hier ermittelt der Roboter eine Bewertung seiner Aktionen indem der Versatz des zu greifenden Objektes nach der Bewegung mit dem Laser-Entfernungsmesser des Roboters bestimmt wird. Darüberhinaus werden Kollisionen des Roboterarms mit dem Tisch und Probleme mit der inversen Kinematik erkannt.

In dieser Diplomarbeit wird verstärkendes Lernen zur Verbesserung der durch Imita-

tionslernen erhaltenden Parameter einer Bewegung eingesetzt, um den Versatz des zu greifenden Objektes bei der Bewegung weiter zu verringern. Die durch Imitation gelernten Parameter dienen als Initialisierung für das verstärkende Lernen. Sie repräsentieren eine menschlich wirkende Bewegung. Somit kann das verstärkende Lernen schon mit einer guten Bewegung starten und das zeitaufwendige Durchsuchen des gesamten, hochdimensionalen Parameterraumes entfällt. Auf diese Weise fokussiert das Imitationslernen die Suche im Rahmen des verstärkenden Lernens, wodurch das Verfahren erheblich schneller wird.

Auf der anderen Seite kann durch verstärkendes Lernen die Anzahl der benötigten Demonstrationen klein gehalten werden. Dies ist ein wichtiges Kriterium für die Praxistauglichkeit des Verfahrens, da das Demonstrieren von vielen Bewegungen mit einem großen Zeitaufwand für den Menschen verbunden ist. Desweiteren ist das verstärkende Lernen notwendig, da Imitationslernen allein oftmals nicht ausreicht, um Bewegungen perfekt zu lernen.

Dies hängt zum Einen damit zusammen, dass die Sensoren des Roboters oftmals systematische Fehler aufweisen, so dass die Position des zu greifenden Objektes nicht genau bestimmt werden kann. Zum anderen ist die Kinematik des Roboters zwar sehr ähnlich zu der des Menschen, stimmt allerdings nicht genau mit dieser überein. Dies trifft bei dem in dieser Diplomarbeit verwendeten Roboter insbesondere auf die Hand zu. Schließlich treten bei der Ausführung von Bewegungen mechanische Ungenauigkeiten auf. Der durch diese Faktoren entstehende Versatz eines Objektes bei einer Greifbewegung kann mit Hilfe von verstärkendem Lernen verbessert werden.

5.5.1 Verstärkendes Lernen mit Hilfe der erwarteten Verbesserung

Bei verstärkendem Lernen ist es wichtig, dass die neuen zu testenden Punkte geschickt gewählt werden, um schnell zu einem guten Ergebnis zu kommen. Menschen entscheiden sich intuitiv durch ihr schon erlangtes Wissen über ein Problem für neue, vielversprechende Lösungsstrategien. Da Roboter keine derartige Intuition besitzen, müssen geschickte Suchstrategien durch mathematische Verfahren gefunden werden.

Die in dieser Diplomarbeit verwendete Strategie basiert auf der in Abschnitt 2.2.3 vorgestellten *erwarteten Verbesserung* eines Punktes. Diese gibt an, welche Verbesserung von einer Lösung gegenüber der besten bekannten Lösung zu erwarten ist. Über den Raum aller möglichen Lösungen kann eine Oberfläche der erwarteten Verbesserung definiert werden. Die Strategie zur Wahl neuer zu testender Beispiele sucht auf dieser Oberfläche ein Maximum. Bei seiner Suchstrategie verfolgt das Verfahren einen Kompromiss zwischen Punkten mit einem gutem Mittelwert und solchen mit großer Unsicherheit

5.5.2 Kombination von erwarteter Verbesserung und Verschlechterung

Die erwartete Verbesserung lässt sich sowohl für die Suche nach einem Minimum, als auch für die Suche nach einem Maximum der Gütefunktion verwenden. Die meisten Verfahren suchen nur nach einem Minimum oder einem Maximum. Das in dieser Arbeit vorgestellte Verfahren nutzt beides. Da mit einem realen Roboter gearbeitet wird, ist es nicht nur wichtig, besonders schnell eine möglichst optimale Lösung zu erhalten, sondern auch, dass möglichst wenige für den Roboter gefährliche Situationen entstehen. Die optimale

Konstante	Wert
α	$5,0$
β_1	10^{-4}
β_2	$0,025$
γ_1	-1600
γ_2	$0,0175$
δ	$0,02$

Tabelle 5.1: *Werte der Konstanten in den Gleichungen* (5.18) *und* (5.19)

Lösung entspricht dem Maximum der Gütefunktion aus Gleichung (5.1). Ein Minimum der Gütefunktion entspricht einer möglicherweise gefährlichen Situation für den Roboter. Die erwartete Verbesserung bei der Suche nach einem solchen Minimum wird daher in dieser Diplomarbeit als *erwarteten Verschlechterung* bezeichnet.

Die in dieser Diplomarbeit verfolgte Strategie, um geschickt neue zu testende Punkte zu bestimmen, setzt sich aus der erwarteten Verbesserung und Verschlechterung der Gütefunktion zusammen. Die verwendete Funktion ist wie folgt definiert:

$$f_r(\vec{x}) = \alpha \cdot E[I(x)]_{max} - f(E[I(x)]_{min}) \tag{5.18}$$

Der Einfluss der erwarteten Verschlechterung wird, wie in Abbildung 5.7 dargestellt, durch eine abschnittsweise definierte Funktion aus einer Hyperbel und einer Geraden beschrieben

$$f(E[I(x)]_{min}) = \begin{cases} \beta_1 \cdot (\Delta - g + \beta_2)^{-2} & \text{falls } \Delta \geq (g - \delta) \\ \gamma_1 \cdot (\Delta - g + \gamma_2) & \text{sonst} \end{cases} \tag{5.19}$$

mit $\Delta = f_{best} - E[I(x)]_{min}$. Dabei sind die in dieser Diplomarbeit verwendeten Werte der Konstanten in Tabelle 5.1 aufgeführt. Bei α und β_1 handelt es sich um empirisch bestimmte Konstanten, die die Gewichtung der erwarteten Verbesserung gegenüber der erwarteten Verschlechterung steuern. Die Konstante β_2 verschiebt die quadratische Funktion, so dass diese erst in der Nähe der Grenze eine große Steigung aufweist. Der Wert ist ebenfalls empirisch bestimmt. Der Schwellwert δ gibt den Punkt in Abhängigkeit von der Grenze g an, ab dem die Hyperbel linear fortgesetzt wird. Die Konstanten γ_1 und γ_2 sind so gewählt, dass sie die Gerade beschreiben, die durch den Punkt $(g - \delta, f(g - \delta))$ verläuft und an diesem dieselbe Steigung wie die Hyperbel hat.

Das Verhältnis der erwarteten Verbesserung zu der erwarteten Verschlechterung in der Gütefunktion (5.18) hängt durch 5.19 von der Differenz Δ zwischen dem bisher besten Mittelwert f_{best} und der erwarteten Verschlechterung ab. Solange diese Differenz nicht unter eine gewisse Grenze g sinkt, werden keine gefährlichen Situationen für den Roboter erwartet. Die Verschlechterung hat dann so gut wie keinen Einfluss auf den Wert der Gütefunktion und es wird anhand der erwarteten Verbesserung entschieden, welche Lösung einen besonders hohen Nutzen verspricht. Nähert sich die Differenz Δ der Grenzet g, so wird auch der Einfluss der Verschlechterung auf die Gütefunktion größer. Überschreitet Δ die Grenze, werden die Werte der Gütefunktion stark negativ, um eine Wahl dieser Parameter zu verhindern. In diesem Fall nimmt die erwartete Verbesserung aufgrund der möglichen Gefährdung für den Roboter keinerlei Einfluss mehr auf die Güte.

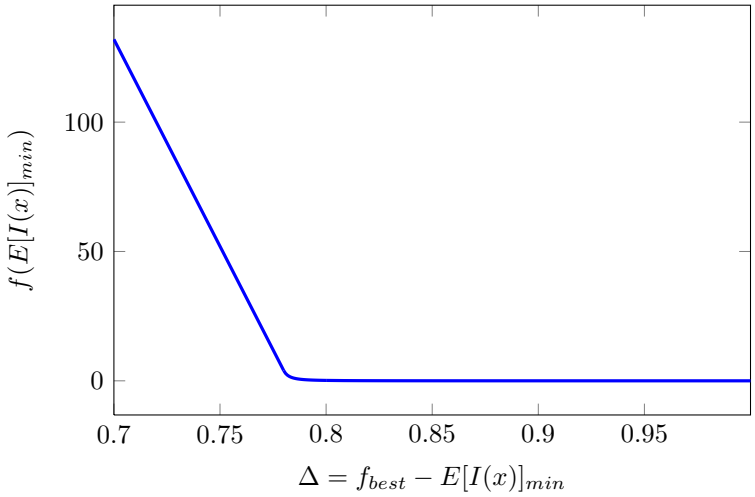


Abbildung 5.7: *Einfluss der erwarteten Verschlechterung*

Der Schutz des Roboters hat Vorrang vor der möglichen erwarteten Verbesserung an dieser Stelle.

Die Grenze g wird variabel angepasst, abhängig vom Mittelwert der Güte der bisher besten Lösung f_{best}:

$$g = \begin{cases} 0.8 & \text{falls } 1 - ((1 - f_{best}) \cdot \lambda) < 0.8 \\ 0.95 & \text{falls } 1 - ((1 - f_{best}) \cdot \lambda) > 0.95 \\ 1 - ((1 - f_{best}) \cdot \lambda) & \text{sonst} \end{cases} \tag{5.20}$$

Die Konstante λ steuert die zulässige Verschlechterung in Abhängigkeit von der noch erreichbaren Verbesserung und wurde in dieser Diplomarbeit mit $\lambda = 3.0$ gewählt. Eine variable Grenze bringt den Vorteil, dass nicht nur der Roboter vor Schäden bewahrt wird, sondern die Suche zusätzlich, gerade bei guten Werten für f_{best} zielgerichteter verläuft. Ist f_{best} schon auf sehr gute Werte gestiegen, so hätte die Suche bei Verwendung einer festen Grenze einen sehr großen Spielraum nach unten. Somit könnten auch Werte mit großer Unsicherheit getestet werden, die viel schlechter ausfallen könnten als ihr Mittelwert vorhersagt. Durch eine variable Grenze wird der Suchbereich nach unten beschränkt. Um zu verhindern, dass gefährliche Lösungen für den Roboter ausgewählt werden, ist in Gleichung 5.20 der mögliche Bereich für g nach unten begrenzt. Um einem Stagnieren der Suche entgegenzuwirken, wird der Bereich nach oben ebenfalls begrenzt.

5.5.3 Optimierung durch Gradientenabstieg

In Abschnitt 5.3 wurde beschrieben, wie Parametersätze bekannter Bewegungen auf einen neuen Punkt generalisiert werden können. Die Generalisierung lieferte dabei einen Mittelwert und eine Unsicherheit der zu erwartenden Güte. Im Rahmen des verstärkenden Lernens werden die generalisierten Parameter als Initialisierung für eine Suche auf der Oberfläche der Gütefunktion verwendet. Der Mittelwert der zu erwartenden Güte wird als Initialisierung für f_{best} verwendet.

Um von dieser Startlösung, ein Optimum auf der Gütefunktion (5.18) zu bestimmen, wird ein gradientenbasiertes Verfahren benutzt. Da ein normales Gradientenverfahren sich als zu langsam erwies, wird in dieser Diplomarbeit Rprop verwendet. Genauere Informationen zu Rprop befinden sich in Abschnitt 2.2.2. Der Gradient der Fitnessfunktion ist wie folgt definiert:

$$\frac{\partial f_r(\vec{x})}{\partial \vec{x}} = \alpha \cdot \frac{\partial E[I(x)]_{max}}{\partial \vec{x}} - \frac{\partial f(E[I(x)]_{min})}{\partial \vec{x}} \tag{5.21}$$

$$\frac{\partial f(E[I(x)]_{min})}{\partial \vec{x}} = \begin{cases} \beta \cdot 2 \cdot (\Delta - g + \beta_2)^{-3} \cdot \frac{\partial E[I(x)]_{min}}{\partial \vec{x}} & \text{falls } \Delta \geq (g - \delta) \\ -\gamma_1 \cdot \frac{\partial E[I(x)]_{min}}{\partial \vec{x}} & \text{sonst} \end{cases} \tag{5.22}$$

Die hierin enthaltenden Formeln sind aus Abschnitt 2.2.3 und den Gleichungen (5.8)-(5.14) bekannt. Dabei werden nur die vier in Abschnitt 5.3 aufgeführten Parameter durch verstärkendes Lernen verbessert.

5.5.4 Erweiterung des Suchraums durch zufällige Suche

Um das Steckenbleiben in einem Nebenminimum zu vermeiden, wird, falls sich die Gütewerte während der Suche nach einem geeigneten zu testenden Punkt nicht verbessern, zu den gefundenen Parametern $\vec{p}$ ein Rauschterm hinzugefügt. Dieser ist normalverteilt um den Mittelwert 0. Seine Varianz $\sigma^2_{Rauschen}$ ist abhängig vom Mittelwert der zu erwartenden Güte der Parameter $\vec{p}$. Da die erreichbare Güte nach oben begrenzt ist, ist eine solche Anpassung sinnvoll.

$$\sigma^2_{Rauschen} = \begin{cases} 0.002 & \text{falls } 1.0 - \mu(\vec{p}) < 0.04 \\ 0.04 & \text{falls } 1.0 - \mu(\vec{p}) > 0.04 \\ 1 - \mu(\vec{p}) & \text{sonst} \end{cases} \tag{5.23}$$

Die obere Grenze der Varianz sorgt dafür, dass die Parameter sich nicht zu extrem ändern dürfen, was wieder zu gefährlichen Situationen für den Roboter führen würde. Die unter Grenze verhindert das Steckenbleiben aufgrund einer zu kleinen Standardabweichung.

Von den verrauschten Parametern aus wird nun ebenfalls mittels Rprop auf der Gütefunktion nach einem Maximum gesucht. Um die Parameter zu bestimmen, die tatsächlich auf dem Roboter ausgeführt werden, werden die durch diese zweite Suche erreichten Gütewerte mit denen der Parameter $\vec{p}$ und der Initialisierungsparameter der Suche verglichen. Die Parameter mit dem besten Fitnesswert werden ausgeführt und bewertet.

6 Kapitel Experimente

In diesem Kapitel wird das im Rahmen dieser Diplomarbeit entwickelte Lernverfahren experimentell am Beispiel des Lernens von Greifbewegungen evaluiert. Dazu werden Experimente sowohl in einer simulierten Umgebung, als auch auf dem realen Roboter Dynamaid durchgeführt. Die Simulation erlaubt die Durchführung vieler Experimente, so dass die Ergebnisse statistisch aussagekräftig sind. Dabei können die meisten Experimente unüberwacht ausgeführt werden, da keine menschlichen Eingaben nötig sind. Beim realen Roboter ist dies aus Sicherheitsgründen nicht möglich, so dass Experimente mit einem hohen Zeitaufwand verbunden sind. Somit können nur vergleichsweise wenige Experimente auf dem Roboter durchgeführt werden, die zwar statistisch nicht repräsentativ sind, jedoch die Praxistauglichkeit des Lernverfahrens auf einem realen System demonstrieren.

Im Folgenden wird zunächst in Abschnitt 6.1 der Aufbau der Versuche sowohl in der simulierten Umgebung, als auch in einem realen Szenario beschrieben. Anschließend werden in Abschnitt 6.2 und 6.3 die beiden Komponenten, das Imitations- und das verstärkende Lernen unabhängig voneinander untersucht. Zum Abschluss wird in Abschnitt 6.4 das kombinierte Lernverfahren erprobt und anhand der Ergebnisse gezeigt, dass durch Ausnutzung der Vorteile beider Verfahren der Lernerfolg verbessert werden kann.

6.1 Versuchsaufbau

Die Aufgabe des Roboters in den in diesem Kapitel beschriebenen Experimenten ist das Greifen einer Tasse, die auf einem Tisch vor ihm steht. Dabei soll der Roboter ausgehend von einer Ruheposition des Armes eine Greifbewegung ausführen, die Tasse anheben und wieder absetzen. Schließlich soll er wieder in die Ruheposition zurückfahren. Die zu lernende Bewegung beinhaltet dabei nur die Greifbewegung zur Tasse.

Um die Ergebnisse vergleichbar zu machen und Rückschlüsse von der Simulation auf das reale Verhalten zu ermöglichen, wurde in der simulierten Umgebung das tatsächliche Szenario nachgebildet. In beiden Fällen hat der Tisch eine Höhe von 75cm, wie sie in menschlichen Umgebungen üblich ist.

Für die Aufnahme von demonstrierten Greifbewegungen für das Imitationslernen wurden in dieser Diplomarbeit eine Motion Capture-Anlage und ein Datenhandschuh verwendet, die um einen weiteren Tisch herum aufgebaut sind. Da der Roboter zwar

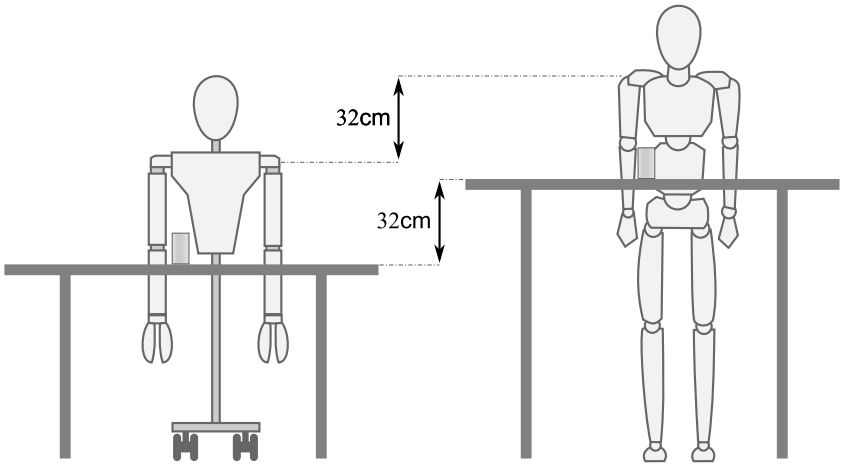


Abbildung 6.1: *Versuchaufbau mit den unterschiedlichen Tischhöhen, um den gleichen Abstand von der Schulter des Menschen zur Tasse wie beim Roboter zu erhalten*

einen Oberkörper mit anthropomorphen Proportionen besitzt, jedoch nicht die Größe eines Erwachsenen erreicht, ist dieser Tisch etwas höher. Auf diese Weise wird der Größenunterschied zwischen dem Roboter und dem demonstrierenden Menschen ausgeglichen, so dass sich die Tassen auf den beiden Tischen jeweils von der Schulter des Menschen und des Roboters aus gesehen an derselben Position befinden. Dies ist für eine Übertragung der aufgenommenen Bewegung auf den Roboter nötig. In dieser Diplomarbeit hatte der zweite Tisch eine Höhe von 107cm. Da der Roboter lernen soll, auf einem typischen Tisch einer menschlichen Umgebung zu greifen, wird die Höhe des Tisches des Trainers verändert und nicht die des Roboters.

Neben der relativen Höhe des Tisches muss auch sichergestellt werden, dass sich Mensch und Roboter im selben Abstand vor dem Tisch befinden. In den hier beschriebenen Experimenten betrug der Abstand vom Tisch immer 25cm. Um die genaue Positionierung der Tasse für den Menschen anhand gegebener Koordinaten auf dem Tisch zu vereinfachen, wurde auf diesem ein Messraster angebracht.

6.1.1 Simulation

Für die Experimente in der simulierten Umgebung und zur Erprobung von Bewegungen, bevor diese auf dem realen Roboter ausgeführt werden dürfen, wurde der physikalische Simulator *Gazebo* benutzt. Details zu dessen Funktionsweise sind in Abschnitt B.3 beschrieben.

Um die Güte einer Greifbewegung zu ermitteln, wird zunächst bestimmt, ob die Tasse gegriffen werden konnte oder verfehlt wurde. Da der simulierte Roboter im Gegensatz zu dem Realen über keine Sensoren in der Hand verfügt, ist er nicht in der Lage, zu

messen, ob sich die Tasse in der Hand befindet. Um dies festzustellen wird daher die vertikale Koordinate der Tasse während des Anhebens betrachtet. Ändert sich diese, hält der Roboter die Tasse in der Hand. Bleibt sie konstant wurde die Tasse verfehlt. Wurde die Tasse erfolgreich gegriffen, hängt die Güte der Greifbewegung von der Strecke ab, um die die Tasse verschoben wurde.

Während der gesamten Bewegung wird ein Kollisionstest zwischen der Hand des Roboters und der Tischplatte durchgeführt. Dazu werden anhand der Position, der Ausrichtung und der Öffnung der Hand die Positionen der Endpunkte des Greifers bestimmt und mit der bekannten Position der Tischplatte verglichen. Als weitere Schutzmaßnahme für den Roboter werden Probleme mit der inversen Kinematik abgefangen. Dabei kann es sich zum Beispiel um eine Überstreckung des Arms oder Singularitäten handeln. Wird eine Kollision oder ein Problem mit der inversen Kinematik festgestellt, wird der Bewegung ein entsprechend schlechter Gütewert zugeordnet.

6.1.2 Reales Szenario

Bevor der Roboter im realen Szenario die Tasse greifen darf, wird die Bewegung in der Simulation überprüft. Entstehen Kollisionen oder Probleme mit der inversen Kinematik, so wird verhindert, dass diese Bewegung auf dem realen Roboter ausgeführt wird. Um die Güte einer Bewegung zu bestimmen, kann im realen Szenario durch Infrarot-Sensoren in der Hand des Roboters ermittelt werden, ob die Tasse gegriffen wurde. Die Position der Tasse wird mit Hilfe des Laserentfernungsmessers des Roboters bestimmt. Um die Kosten zu bestimmen, falls die Tasse erfolgreich gegriffen wurde, wird die Position der Tasse vor und nach der Greifbewegung gemessen und der Versatz bestimmt. Da sich während der Greifbewegung der Arm des Roboters zeitweise im Sichtfeld des Entfernungsmessers befindet und teilweise die Tasse verdeckt, kann die zweite Messung erst erfolgen, sobald sich der Arm wieder in seiner Ausgangsposition befindet.

6.2 Imitationslernen

Um den in Kapitel 5.4 beschriebenen Ansatz des Imitationslernens zu validieren, wurde der Algorithmus unabhängig vom übrigen Verfahren mit einer Reihe von Greifbewegungen getestet. Dabei wurde untersucht, wie gut das Ziel der Imitation, die Übereinstimmung der Trajektorien des Roboters und der des Menschen, erreicht wurde.

Für die Experimente wurde eine Tasse an unterschiedlichen Positionen auf dem Tisch platziert und die entsprechenden Greifbewegungen mit Hilfe der Motion Capture-Anlage und des Datenhandschuhs aufgezeichnet. Um sicherzustellen, dass der Roboter und der Mensch in allen Experimenten die gleiche Position im Bezug auf die Tasse haben, wurde eine Position für den Menschen auf dem Boden markiert.

Aus den aufgezeichneten Bewegungen wurden anschließend, wie in Abschnitt 5.4.2 beschrieben, Parameter für den Regler extrahiert. Dieser generiert daraus eine Trajektorie, die mit der menschlichen Bewegung verglichen werden kann. Die Punkte dieser Trajektorie stellen Zielpunkte für den Roboter dar, die dieser durch inverse Kinematik anzusteuern versucht. Bei der Bestimmung der Parameter wird davon ausgegangen, dass der Roboter diese Ziele erreicht. Daher wird bei den folgenden Untersuchungen nicht

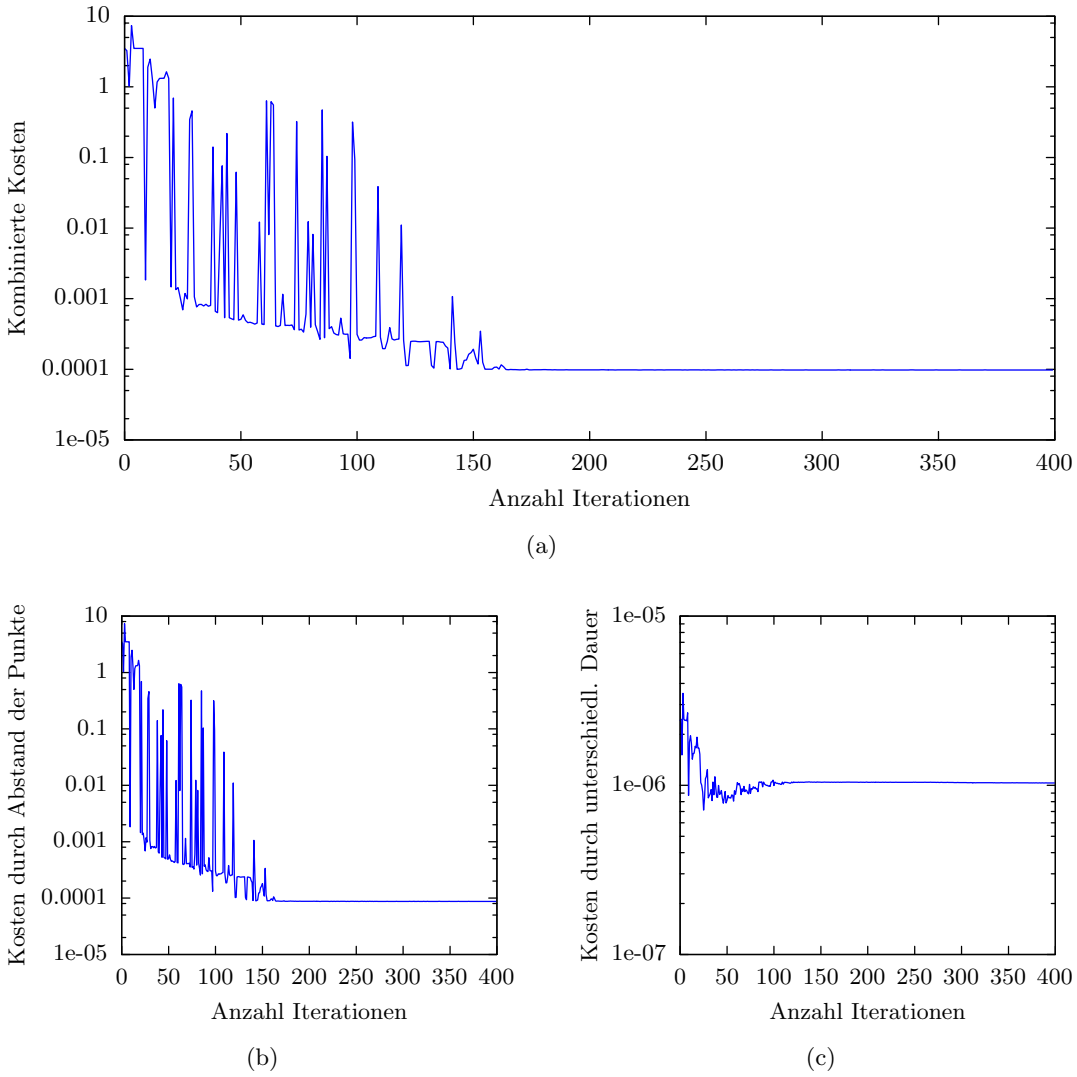


Abbildung 6.2: *Darstellung von gemittelten Lernkurven des Downhill-Simplex-Verfahrens. Gemittelt wurde über 30 Trajektorien von Bewegungen, mit denen Objekte an unterschiedlichen Positionen auf einem Tisch gegriffen wurden. In Abbildung (a) ist die kombinierte Kostenfunktion zu sehen, deren einzelne Teilkomponenten, der gemittelte räumliche Abstand pro Punkt und der Unterschied in der Dauer der Ausführung, in Abbildung (b) und (c) dargestellt sind. In allen drei Abbildungen ist die y-Achse mit einer logarithmischen Skala versehen.*

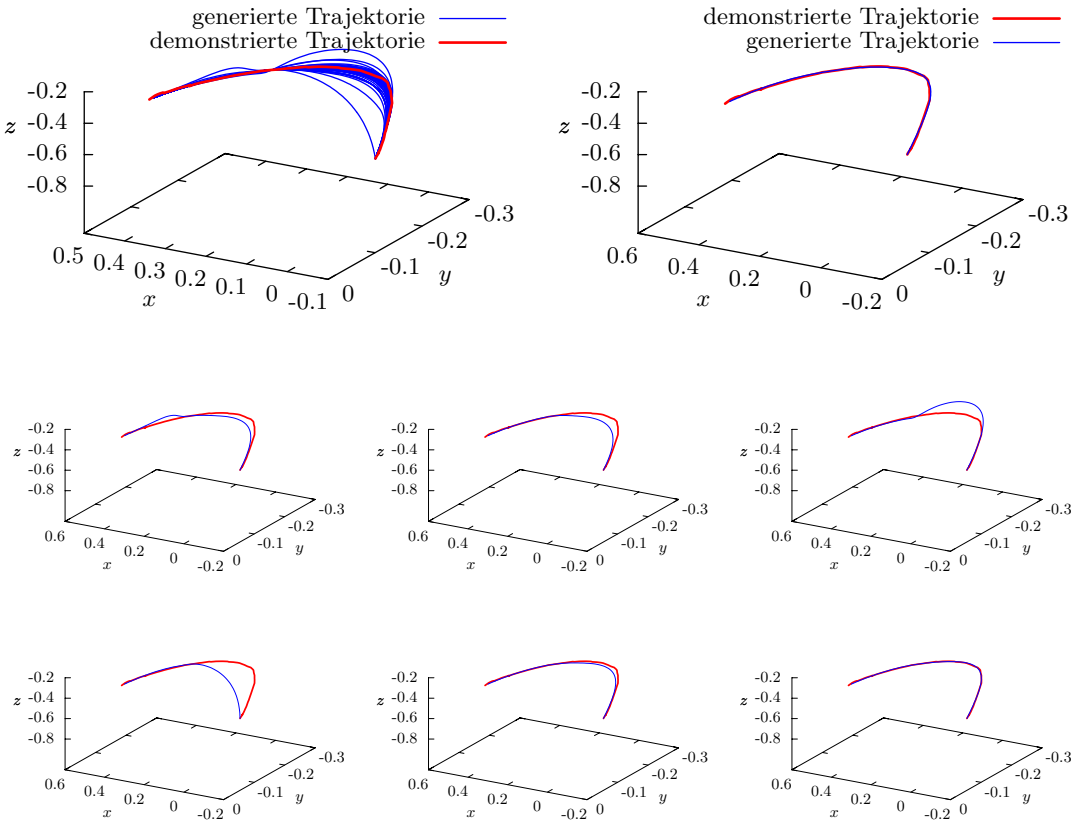


Abbildung 6.3: *Trajektorien im Verlauf des Downhill-Simplex-Verfahrens am Beispiel einer aufgenommenen Bewegung. Im linken oberen Bild sind in blau alle in den ersten 40 Iterationen durch das Downhill-Simplex-Verfahren erzeugten Trajektorien dargestellt. Im weiteren Verlauf entstehen keine sichtbaren Veränderungen mehr. Die rote Trajektorie beschreibt die demonstrierte Bewegung nach der Segmentierung. Im rechten oberen Bild ist das Endergebnis des Verfahrens zu sehen. In den unteren Bildern sind exemplarisch einige Zwischenergebnisse des Downhill-Simplex-Verfahrens dargestellt.*

zwischen Greifbewegungen in der Simulation und auf dem realen Roboter unterschieden.

Die Parameter des Reglers werden auf zwei unterschiedliche Arten aus den aufgenommenen Bewegungen bestimmt. Von den 29 Parametern werden 21 direkt aus den Daten berechnet. Die übrigen 8 Parameter werden durch iterative Optimierung der Trajektorie bestimmt. Dabei werden die direkt bestimmten Parameterwerte zwar zur Generierung der Trajektorien verwendet, selbst jedoch nicht verändert.

In den nachfolgenden Experimenten wird untersucht, wie gut das Downhill-Simplex-Verfahren und der hier vorgestellte Regler die menschliche Trajektorie nachbilden können. Dazu wurden 30 durch einen Menschen vorgeführte Trajektorien betrachtet, für die eine Tasse an verschiedenen Stellen auf dem Tisch positioniert wurde.

Abbildung 6.2(a) zeigt den Verlauf der Kosten als Mittelwerte über alle 30 Trajektorien.

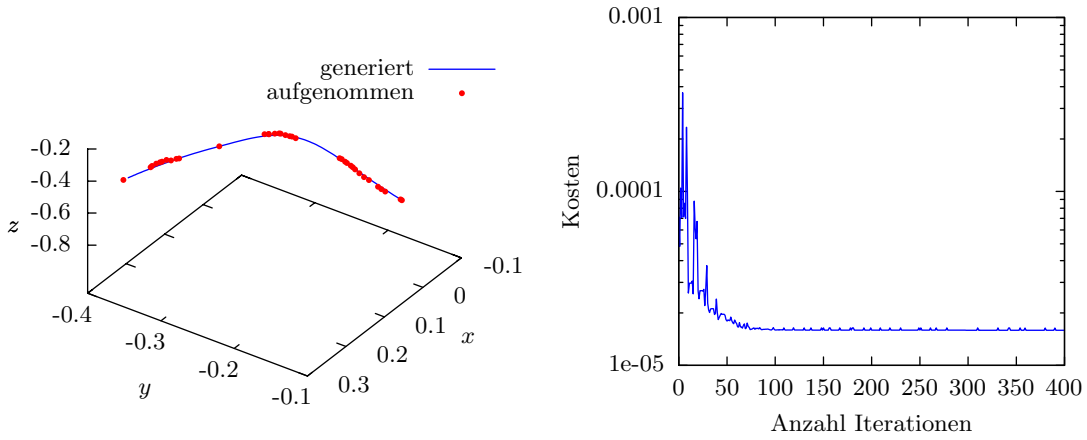


Abbildung 6.4: *Endergebnis (links) und Lernkurve (rechts) der Optimierung durch das Downhill-Simplex-Verfahren bei einer demonstrierten Bewegung, die große Lücken aufweist*

Diese Kosten setzen sich, wie in Abschnitt 5.4.2 beschrieben, aus zwei Teilkomponenten zusammen. Diese bestimmen den räumlichen Abstand bzw. den Unterschied der Ausführungsdauer der Trajektorien. Die gemittelten Werte dieser Komponenten sind in Abbildung 6.2(b) und 6.2(c) dargestellt. Es ist zu erkennen, dass die mittleren Kosten nach ca. 110 Schritten bei weniger als 10^{-4} konvergieren. Dabei liegen die Kosten für einen unterschiedlichen Abstand der Trajektorien zu Beginn der Optimierung um eine Größenordnung von 10^6 über den Kosten für eine unterschiedliche Dauer und dominieren während der gesamten Optimierung die kombinierten Kosten. Dies wurde so gewählt, um eine starke Präferenz für einen räumlich ähnlichen Verlauf der Bewegungen auszudrücken. Die ähnliche Dauer der Bewegungen spielt für eine gute Imitation nur eine untergeordnetere Rolle, soll aber dennoch berücksichtigt werden.

In Abbildung 6.3 sind exemplarisch für den Verlauf der Optimierung die Trajektorien zu einigen ausgewählten Zeitpunkten dargestellt. Dabei handelt es sich bei allen Darstellungen um dieselbe demonstrierte Bewegung. Die Grafik zeigt, dass alle generierten Trajektorien durch die vorgegebenen Start-, End- und Viapunkte verlaufen. Dazwischen nähert sich der Verlauf im Laufe der Optimierung der Zieltrajektorie an. Dabei generiert der Regler ausschließlich glatte Bewegungen.

Auch Motion Capture-Daten, die große Lücken oder Ausreißer aufweisen, können mit dem hier vorgestellten Verfahren gut approximiert werden. Ein Beispiel einer solchen Aufnahme ist in Abbildung 6.4 zu sehen.

Abbildung 6.5 zeigt Ausschnitte einer Videosequenz, die während des Imitationslernens aufgenommen wurde. Auf den Bildern ist die Ähnlichkeit der Bewegungsabläufe des Menschen und des Roboters nach Optimierung der Parameter zu erkennen.

Neben der seitlichen Greifbewegung wurde auch das Greifen eines Objekts von oben trainiert, um zu demonstrieren, dass unterschiedliche Bewegungen imitiert werden können. Aufgrund der Grenzen der Bewegungsfreiheit des Roboters wurde für diesen Versuch die Tischhöhe 25cm verringert, so dass sowohl der Mensch, als auch der Roboter die

Bewegung bequem durchführen konnten.

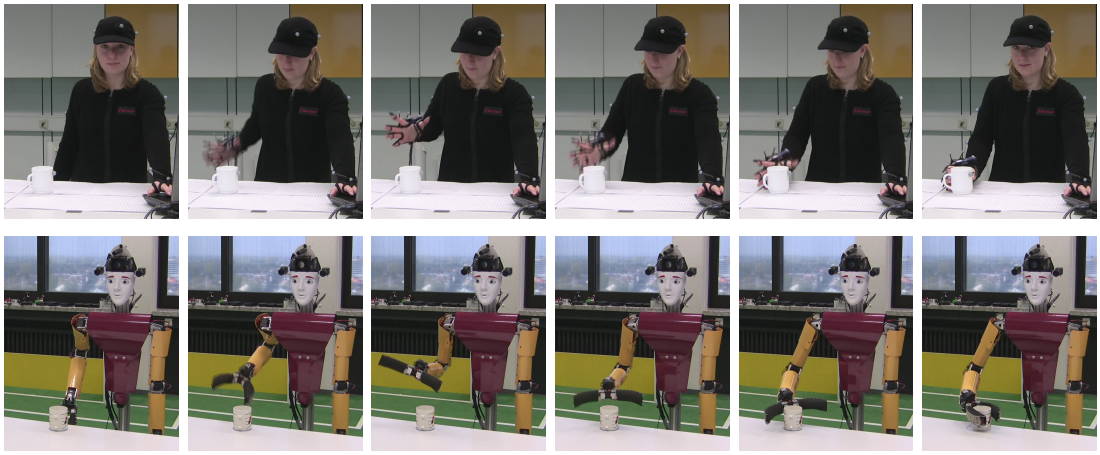


Abbildung 6.5: *Ausschnitte einer Videoseqenz des Imitationslernens beim Greifen von der Seite. Der Mensch führt eine Bewegung vor, die durch das Tragen eines Motion Capture-Anzugs und eines Datenhandschuhs aufgezeichnet werden kann. Diese wird vom Roboter imitiert.*

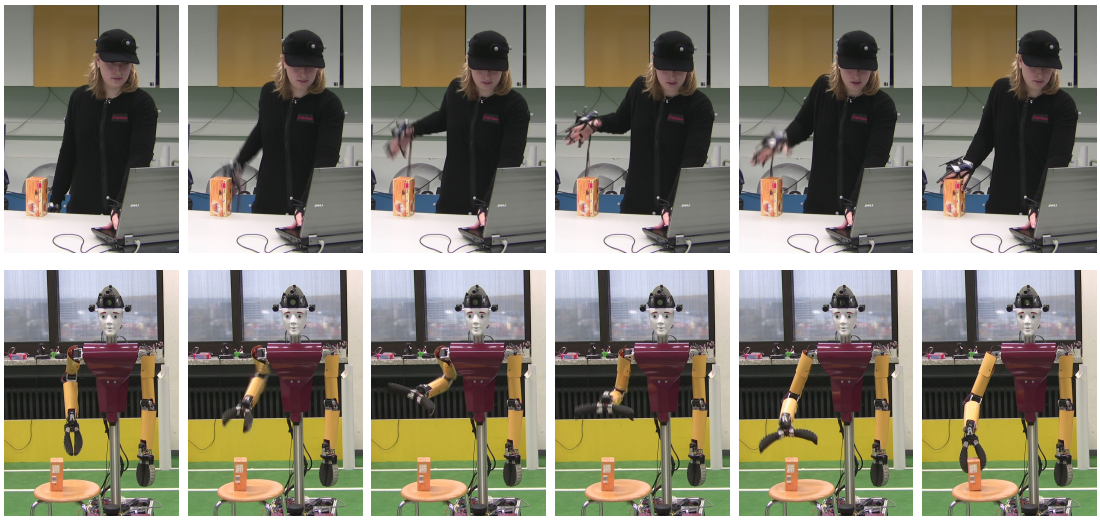


Abbildung 6.6: *Ausschnitte einer Videosequenz des Imitationslernens beim Greifen von oben.*

6.3 Verstärkendes Lernen

In diesem Abschnitt werden eine Reihe von Experimenten beschrieben, um die zweite Komponente des Lernverfahrens, das verstärkende Lernen, zu evaluieren. Die Aufgabe des verstärkenden Lernens ist es, die durch Imitation gelernten Bewegungen weiter zu verbessern. Dabei werden, wie in Abschnitt 5.5 beschrieben, lediglich 4 der insgesamt 31 Parameter des Reglers angepasst. Für die übrigen 27 Parameter werden daher geeignete Initialisierungswerte benötigt. Für die hier beschriebenen Experimente wurde das Verfahren mit einer vorgeführten Bewegung initialisiert, um das Lernproblem ähnlich zu der Aufgabe des verstärkenden Lernens im kombinierten Lernverfahren zu halten. Praktisch wurde dazu die Entscheidung für ein Teilverfahren so modifiziert, dass immer das verstärkende Lernen gewählt wurde.

Wie auch bei der Evaluation des Imitationslernens, wurden Experimente sowohl auf dem realen Roboter, als auch in einer simulierten Umgebung durchgeführt. Um die Ungenauigkeit des Laserentfernungsmessers und der Motorik des realen Roboters, welche in der Simulation nicht vorhanden sind, zu berücksichtigen, wurde für die Experimente in der simulierten Umgebung der Offset-Wert in Blickrichtung des Roboters auf 15cm gesetzt. Somit lag die Zielposition, die der simulierte Roboter anfuhr, zu Beginn des Lernverfahrens 15cm hinter der wahren Position der Tasse.

Parameter	Kernelbreite	
	Erfolg	Misserfolg
Zielpunkt in x- und y-Richtung	0,2	0,02
Offset in x- und y-Richtung	0,02	0,002
Maximale Öffnung der Hand	0,02	0,002
Abstand vom Ziel bei max. Öffnung der Hand	0,02	0,002

Tabelle 6.1: *Kernelbreiten der 3 benutzten Gauß-Prozesse*

Die für den Verlauf des Lernprozesses wichtigen Kernelbreiten sind in Tabelle 6.1 beschrieben. Dabei wurde für den Gauß-Prozess, der erfolgreiche Greifbewegungen repräsentiert, eine andere Kernelbreite gewählt, als für die Gauß-Prozesse, die fehlgeschlagene Versuche repräsentieren. Auf diese Weise konnte der Einfluss fehlgeschlagener Greifversuche lokal beschränkt werden, ohne dabei die Generalisierung positiver Beispiele zu beeinflussen. In der Tabelle sind lediglich die Parameter aufgefürt, die im verstärkenden Lernen verbessert werden. Die einzelnen Werte wurden empirisch bestimmt.

6.3.1 Lernen von Offsetwerten

Um die Lernstrategie des Verfahrens zu untersuchen, werden in diesem Abschnitt zunächst Experimente betrachtet, bei denen nur die Offsetwerte in x- und y-Richtung durch verstärkendes Lernen verbessert wurden. Alle anderen Parameter wurden von der demonstrierten Bewegung übernommen. Diese Einschränkung auf einen zweidimensionalen Parameterraum erlaubt die Visualisierung der Strategie.

Abbildung 6.7 zeigt die Entwicklung der Kosten und der Parameterwerte während des Lernvorgangs. Durch die künstliche Änderung des Offsetwertes in x-Richtung ist die Güte

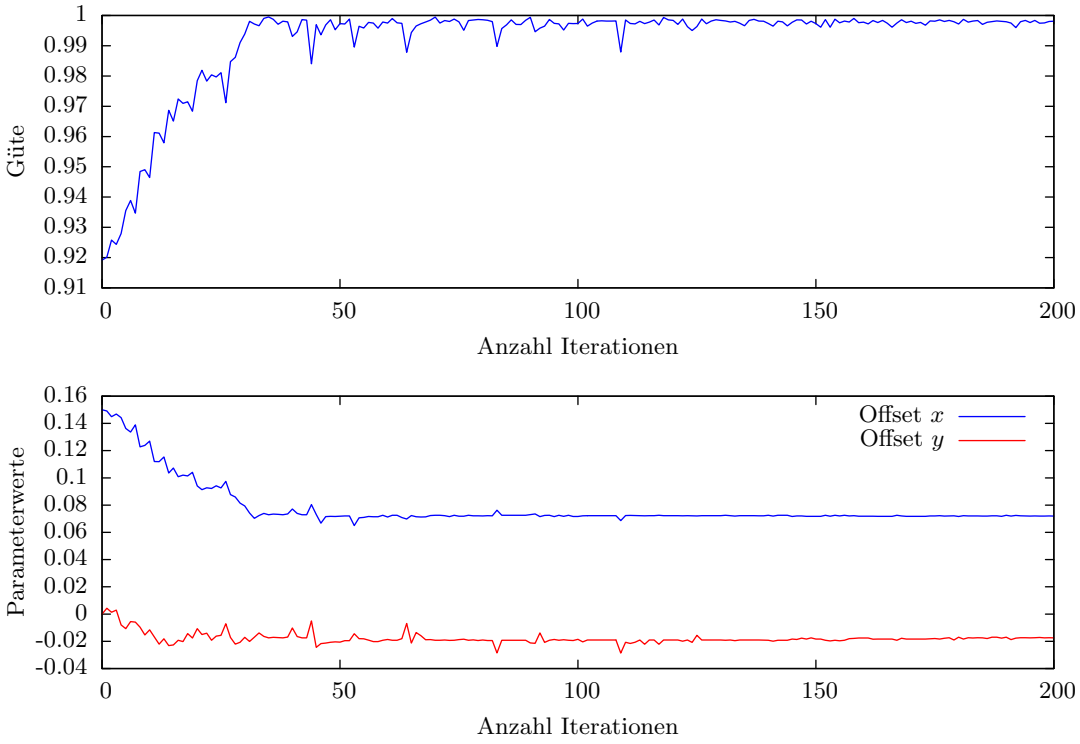


Abbildung 6.7: *Verlauf der Güte und Parameter beim verstärkenden Lernen mit 2 Parametern*

für die durch Imitation gelernte Bewegung lediglich 0,92. Bereits nach ca. 30 Iterationen liegt der Versatz der Tasse unter 1cm und die Güte konvergiert bei einem Versatz von unter 3mm. Bis zur 130. Iteration weist die Kurve jedoch immer wieder Ausreißer nach unten auf. Das Verfahren probiert dort Parametersätze aus, die eine vergleichsweise große Unsicherheit bzgl. der zu erwartenden Güte aufweisen.

Im unteren Teil von Abbildung 6.7 ist zu erkennen, dass auch die Werte der beiden Offsetparameter $0,07$m in x-Richtung bzw. $-0,02$m in der y-Richtung konvergieren. Die Tatsache, dass die Offsetwerte nicht gegen 0 konvergieren zeigt, dass aufgrund der unterschiedlichen Beschaffenheit der menschlichen Hand und der des Roboters, die Position der Tasse alleine nicht ausreicht, um eine optimale Greifbewegung zu erzeugen. Die Offsetparameter erlauben es, diesen Unterschied auszugleichen.

In Abbildung 6.8 sind die Werte dargestellt, die im verstärkenden Lernen Einfluss auf die Wahl der Parameter haben. Da in diesem Experiment nur die beiden Offsetparameter betrachtet werden, können diese in der x- und y-Koordinatenachse aufgetragen werden. Die Farben in den Darstellung drücken die Werte der jeweiligen Funktion aus. Dabei ist die Farbskala pro Zeile gleich. Bei den dargestellten Funktionen handelt es sich in Zeile 5 um die Gleichung 5.18, durch die Parametersätze bewertet werden. Diese setzt sich wiederum aus der erwarteten Verbesserung und Verschlechterung zusammen, die in Zeile 3 und 4 abgebildet sind. Um diese zu bestimmen werden der Mittelwert (Zeile 1) und die Standardabweichung (Zeile 2) benötigt. In den Spalten sind unterschiedliche

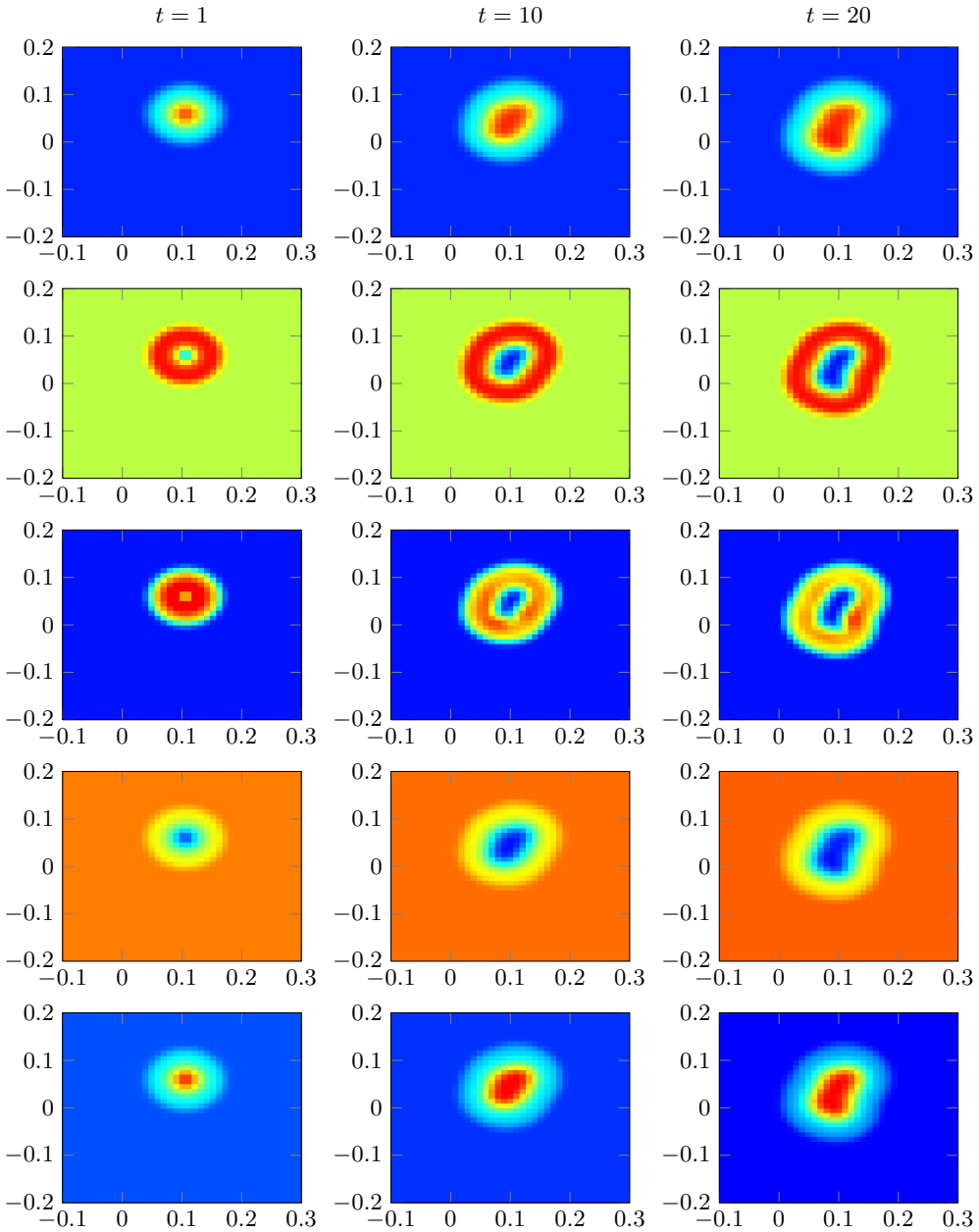
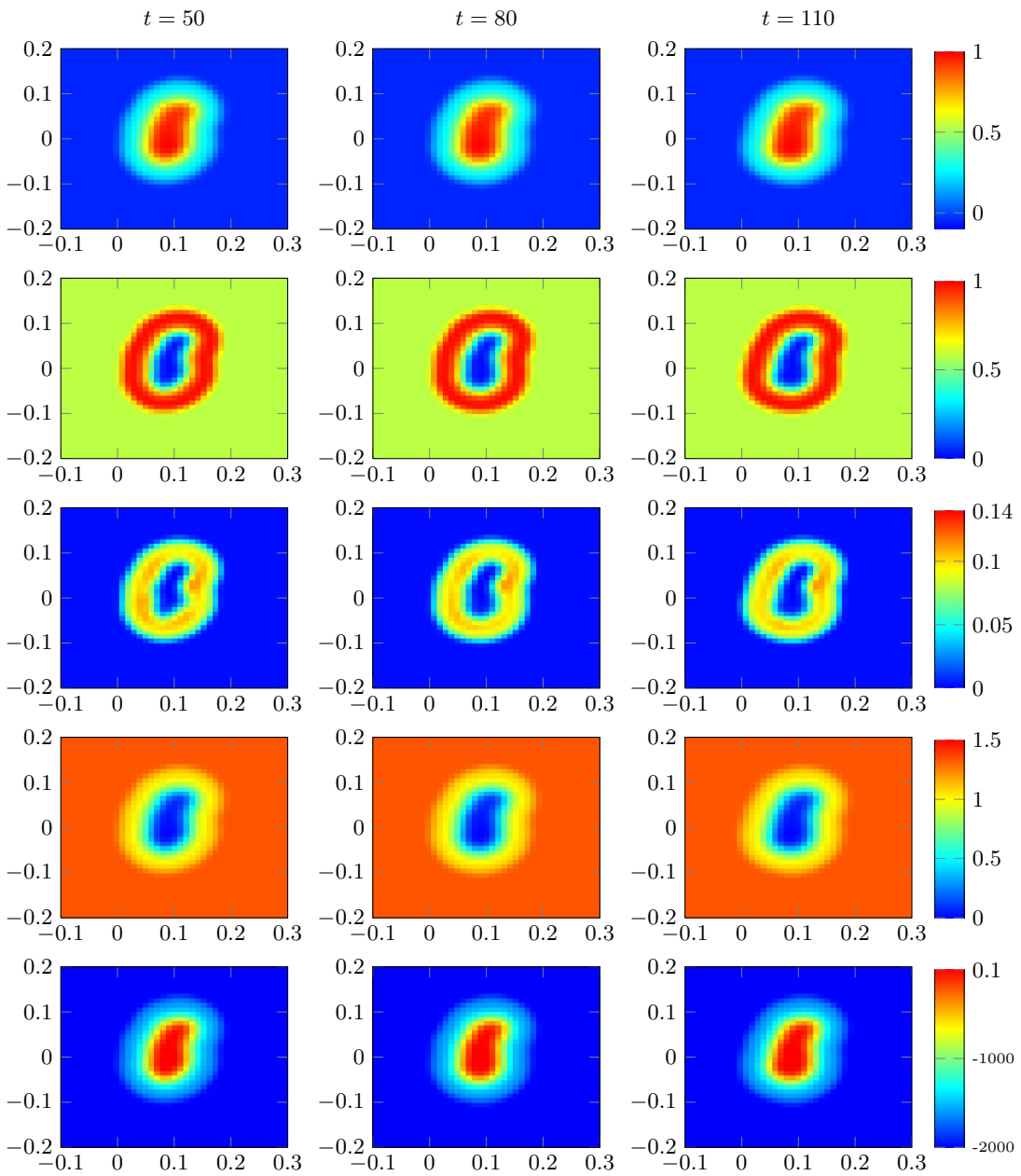


Abbildung 6.8: *In der Darstellung sind untereinander der Mittelwert, die Standardabweichung, die erwartete Verbesserung, die erwartete Verschlechterung und die kombinierte Gütefunktion abgebildet. Die Werte werden durch die Farbe ausgedrückt. In den Spalten ist der Verlauf der Werte zu verschiedenen Zeitpunkten zu sehen. Auf der x- und y-Achse sind die Offsetwerte in x- und y-Richtung dargestellt.*


$t = 50$
$t = 80$
$t = 110$
0.2
0.1
0
−0.1
−0.2
−0.1
0
0.1
0.2
0.3
1
0.5
0
0.14
0.05
1.5
-1000
-2000

Iterationen dargestellt. Beim Mittelwert stellt die rot gefärbte Fläche die Werte der Parametersätze dar, die gute Gütewerte beim Greifprozess erhalten haben. Die gleiche Fläche ist in den Bildern der Standardabweichung blau gefärbt, was auf eine kleine Unsicherheit hindeutet. Dieser Bereich vergrößert sich schlauchförmig in Richtung des in Abbildung 6.7 dargestellten Endergebnisses für die Parameter.

Es ist zu erkennen, dass die erwartete Verbesserung gute Werte kreisförmig um die schon getesteten Parametersätze vermutet. Die kleinste erwartete Verschlechterung ist an den schon bekannten Punkten und deren Umgebung zu erwarten, da sie eine kleine Standardabweichung und gute Mittelwerte aufweisen. In einem großen Teil, der von der erwarteten Verbesserung mit hohen Werten versehenen Bereichen, nehmen die Werte der erwarteten Verschlechterung hohe negative Werte an, was auf gefährliche Situationen für den Roboter hinweist. Diese werden in der kombinierten Funktion durch negative Werte bestraft. Die Verschlechterung hat somit einen großen Einfluss auf die Gesamtfunktion. In der Darstellung der kombinierten Funktion ist zu erkennen, dass besonders die Werte vielversprechend sind, die sehr nah an den schon bekannten Werten liegen.

Die Suchstrategie auf der in Abbildung 6.8 dargestellten Oberfläche wird in Abbildung 6.9 veranschaulicht. Deutlich ist zu erkennen, wie sich die Suche zielstrebig in Richtung Optimum der Güte entwickelt. Zwischendurch macht die Suche kleinere Rückschritte. Dies lässt sich dadurch erklären, dass sich an jedem ausprobierten Parametersatz sowohl an diesem Punkt selbst, als auch in seiner Umgebung die Unsicherheit verringert. Punkte in der Richtung der bereits ausprobierten Parametersätze versprechen oftmals einen guten Mittelwert und haben zudem eine ausreichend große Unsicherheit, um Verbesserungen zu ermöglichen. Des Weiteren ist zu sehen, dass zu Beginn fast nur blau markierte Punkte gewählt wurden, was bedeutet, dass diese mit Hilfe der erwarteten Verbesserung und Verschlechterung deterministisch bestimmt wurden. Je besser die Werte der Gütefunktion werden, umso geringere Verbesserungen werden in einer Iteration erreicht. Dies ist sowohl aus der Lernkurve in Abbildung 6.7 ersichtlich, als auch an der Steigung der Pfeile in Abbildung 6.9, die immer kleiner wird. Je kleiner die Steigung, desto häufiger werden rote Punkte in den Iterationsschritten gewählt. Diese Punkte verhindern das Steckenbleiben in einem Nebenminimum durch eine zufällige Komponente, die in Abschnitt 5.5.4 beschrieben ist. Ausgehend von dieser wird abermals mit Rprop nach einem Optimum gesucht. Gegen Ende der Suche findet die deterministische Strategie keine interessanten neuen Punkte mehr, so dass neue Punkte nur noch durch die zufällige Komponente entstehen. Dies zeigt sich an der hohen Anzahl roter Punkte in der Nähe des Maximums. Dabei entsteht eine sternförmige Suche um dieses, wie im unteren Teil von Abbildung 6.9 zu sehen ist.

6.3.2 Lernen aller Parameter mit fester Position der Tasse

Nachdem im vorherigen Abschnitt zur Illustration der Suchstragie nur zwei Parameter betrachtet wurden, wird in diesem Abschnitt das verstärkende Lernen mit vier Parametern untersucht. Um dieses zunächst an einem einfachen Problem zu evaluieren, wird wie im vorherigen Abschnitt die Position der Tasse festgehalten. Somit muss keine Generalisierung über verschiedene Positionen auf dem Tisch stattfinden.

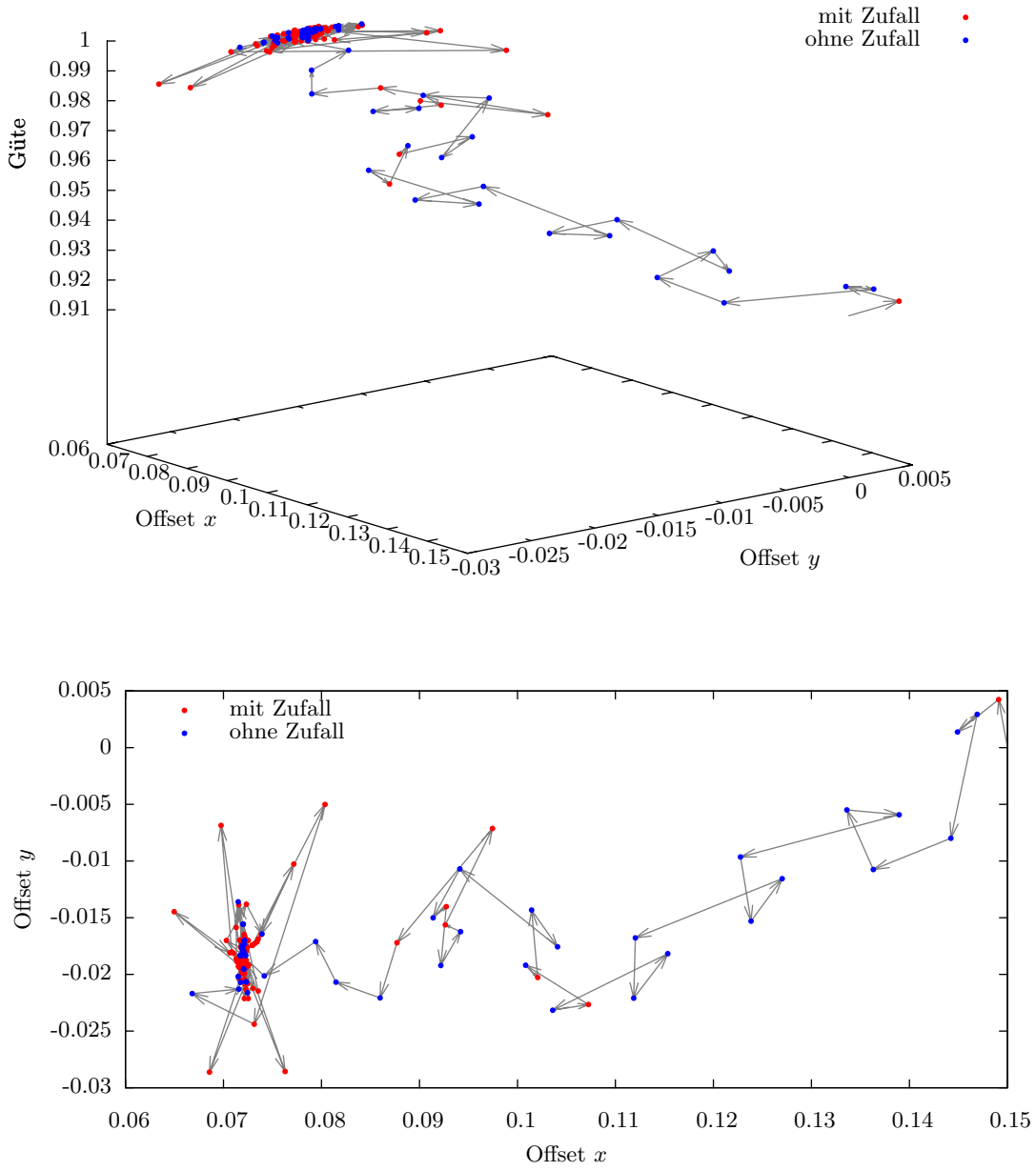


Abbildung 6.9: *Suchstrategie im Parameterraum. Die während der Suche ausgewählten Parametersätze sind durch Pfeile verbunden. Blaue Punkte wurden aufgrund der erwarteten Verbesserung und Verschlechterung, wie in Abschnitt 5.5 beschrieben, gewählt. Um nicht in Nebenminima steckenzubleiben, wird, falls im Optimierungsprozess keine Änderungen auftreten, eine Zufallskomponente hinzugefügt. Ausgehend von diesem neuen Parametersatz wird abermals eine Optimierung vorgenommen. Der beste der beiden Parametersätze wird ausgeführt. Rote Punkte stellen Parametersätze mit der Zufallskomponente dar, die aufgrund besserer Gütewerte ausgewählt wurden. Die genauen Details des Verfahrens werden in Abschnitt 5.5 beschrieben. In der unteren Darstellung ist der Parameterverlauf von oben zu sehen.*

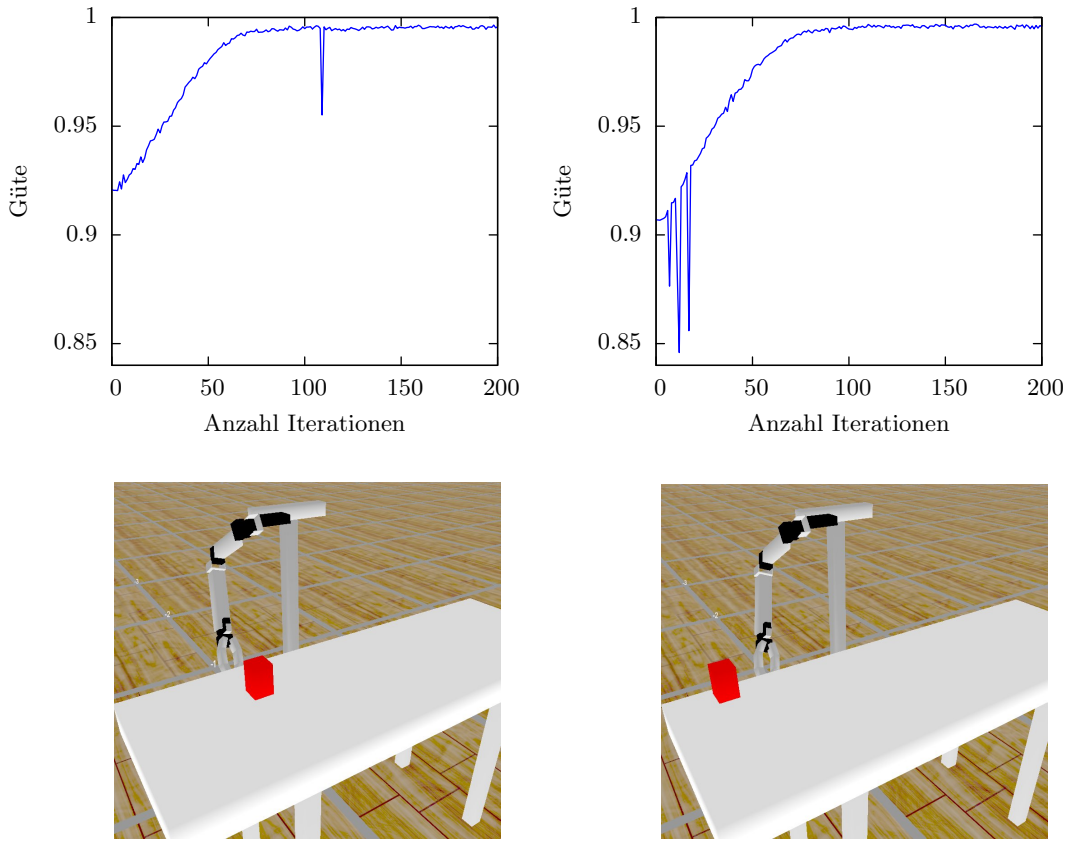


Abbildung 6.10: *Über 25 Versuche gemittelte Lernkurven für zwei verschiedene feste Positionen der Tasse in der Simulation und darunter die dazugehörigen dreidimensionalen Darstellungen der Tassenpositionen in Gazebo.*

Experimente in der simulierten Umgebung

Die folgenden Versuche wurden für zwei unterschiedliche Positionen der Tasse durchgeführt. Abbildung 6.10 zeigt die beiden Positionen der Tasse und die dazugehörigen über 25 Versuche gemittelten Lernkurven. Es ist zu erkennen, dass beide Lernkurven einen ähnlichen Verlauf aufweisen und in weniger als 100 Iterationen bei einem Versatz von weniger als 1cm konvergieren. Die in den Abbildungen auffallenden Ausreißer nach unten entstehen, wenn in dieser Iteration einer oder mehrere der 25 Versuche eine Güte von 0 aufweisen. Dieses bedeutet, dass die Tasse nicht gegriffen werden konnte. Dies war bei der rechten Tassenposition häufiger der Fall als bei der vorderen, allerdings lieferte auch die Initialisierung der Parameter in diesem Fall eine schlechtere Güte als beim Greifen nach vorne. Um das Anfangsniveau der Güte für das Greifen an der vorderern Position zu erreichen, benötigte das verstärkende Lernen ca. 20 Iterationen. In diesen Iterationen treten auch die Ausreißer auf. Dies ist darauf zurückzuführen, dass bei diesen relativ schlechten Parametern, mit denen der Roboter die Tasse nur mit einem großen Versatz greifen kann, schon eine leichte Veränderung der Parameter dazu führen kann, dass er sie gar nicht mehr greifen kann.

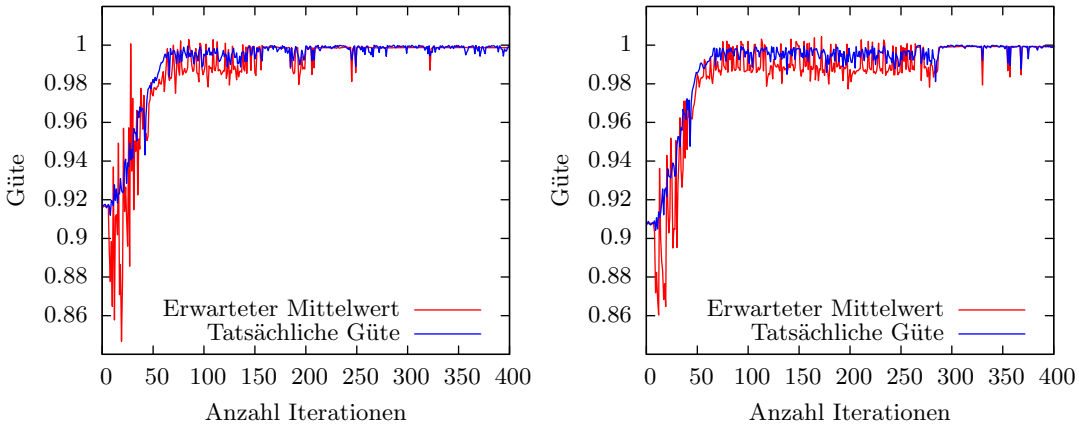


Abbildung 6.11: *Exemplarischer Vergleich der tatsächlichen Güte mit der zuvor für diese Parametersatz erwarteten Güte anhand eines Versuches in der Simulation. Die beiden Grafiken zeigen dieses Verhalten für 2 verschiedene Tassenpositionen.*

In Abbildung 6.11 sind exemplarisch für je einen Versuch an der vorderen und rechten Position der Tasse die erwartete und tatsächliche Güte der gewählten Parametersätze aufgetragen. Während am Anfang die Vorhersage des Gauß-Prozesses noch deutlich von der tatsächlich Güte abweicht, stimmen sie am Ende des Versuches überein, d.h. der Gauß-Prozess hat die richtige Abbildung der Parameter auf die Gütewerte gelernt. Bei den abweichenden Gütewerten zu Beginn unterschätzen die erwarteten Gütewerte zumeist die Tatsächlichen. Dies hängt mit der Extrapolation der Mittelwerte der Gauß-Prozesse zusammen, die für unbekannte Parametersätze gegen 0 gehen. Ist der komplette Parameterraum, der keine Gefahr für den Roboter darstellt, durchsucht, findet keine weitere Exploration statt und der erwartete Gütewert stimmt mit dem Tatsächlichen überein.

In den bisherigen Untersuchungen wurde die maximal zulässige Differenz zwischen dem erwarteten Mittelwert und der erwarteten Verschlechterung variabel gewählt, wie in Gleichung (5.22) beschrieben. Im Gegensatz dazu ist in Abbildung 6.12 eine Lernkurve dargestellt, bei der diese Grenze auf den konstanten Wert 0,8 festgelegt wurde. Dies führt dazu, dass auch bei sehr guten Greifbewegungen im nächsten Schritt Parametersätze gewählt werden dürfen, die möglicherweise zu einer großen Verschlechterung führen können. Diese erwarten durch ihre große Unsicherheit eine Güte über 1, die in der Realität nicht erreichbar ist. Als Folge davon werden diese Punkte bevorzugt gewählt, führen jedoch oftmals nicht wie erwartet zu einer Verbesserung, sondern zu einer Verschlechterung. Dies führt zu der in Abbildung 6.12 dargestellten Lernkurve, die ab Iterationsschritt 70 nicht weiter steigt, sondern wieder fällt. Eine Wahl der Grenze in Abhängigkeit von der Entfernung zum erreichbaren Maximum wie in Gleichung (5.22) ist daher sinnvoll.

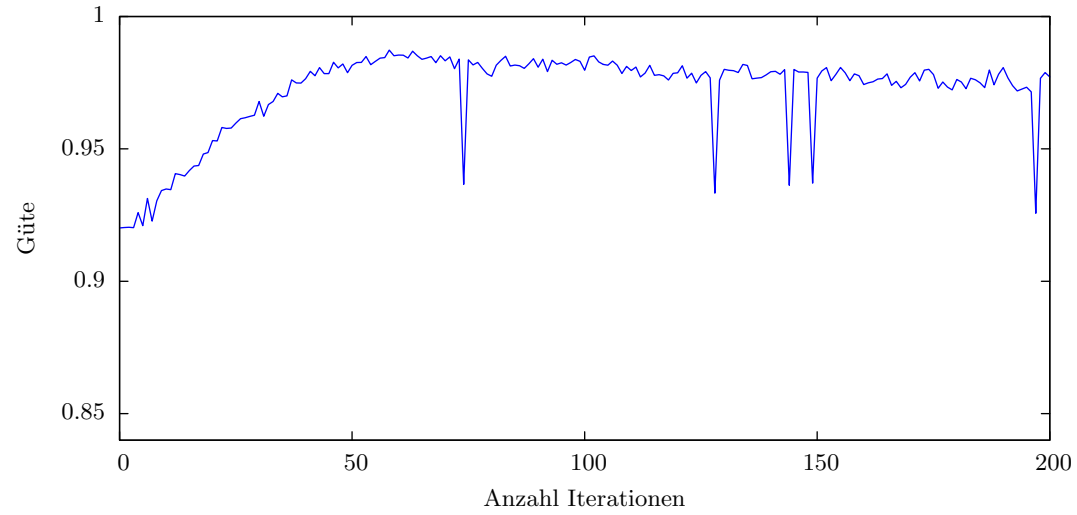


Abbildung 6.12: *Über 25 Versuche gemittelte Lernkurve mit einer festen Grenze für das verstärkende Lernen von 0,8 in der Simulation.*

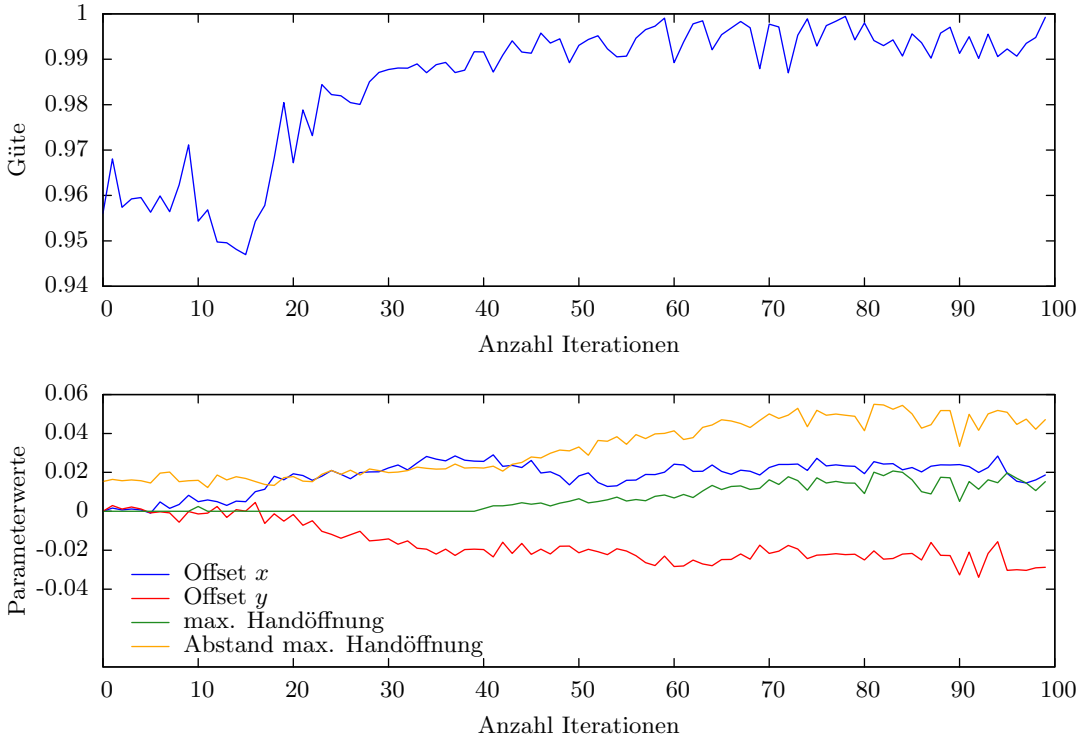


Abbildung 6.13: *Lernkurve und Parameterverlauf eines Versuches auf dem realen Roboter für eine feste Position der Tasse. Diese befand sich 40cm vor dem Mittelpunkt des Roboters und um 20cm von diesem nach rechts verschoben.*

Experimente auf dem realen Roboter

Neben der simulierten Umgebung wurde das Lernen mit fester Position der Tasse auch auf dem realen Roboter trainiert. Dabei kommen die Messunsicherheit des Laserentfernungsmessers und mechanische Probleme erschwerend hinzu, die es in der Simulation nicht gibt. Eine künstliche Verschlechterung durch Veränderung der Offsetwerte wird nicht vorgenommen. Die Offsetwerte starten bei 0.

Die Lernkurve und der Verlauf der Parameter während dieses Experimentes sind in Abbildung 6.13 dargestellt. Bereits zu Beginn wird durch die Imitation der menschlichen Bewegung eine Güte von 0,96 erreicht. Nach 55 Iterationen wird eine Güte von ca. 0,995 erreicht, d.h. die Tasse wird beim Greifen um durchschnittlich 5mm verschoben. In den folgenden Iterationen verbessert sich dieser Wert im Mittel nicht weiter, bedingt durch die motorische Ungenauigkeit des Roboters.

Im Verlauf der Parameter ist nach den 100 durchgeführten Greifbewegungen zwar eine deutliche Tendenz zu erkennen, die Werte sind jedoch nicht konvergiert. Dies kommt zustande, da beim Ausführen eines Parametersatzes aufgrund von Ungenauigkeiten beim Absetzen der Tasse unterschiedliche Gütewerte entstehen können. Eine Konvergenz, wie sie in der simulierten Umgebung beobachtet wurde, ist daher auf dem realen Roboter nicht zu erwarten.

Ein direkter Vergleich der Ergebnisse der Simulation und des realen Roboters ist in Abbildung 6.14 dargestellt. Dabei wurde die Tasse mittig vor dem Roboter plaziert, wie auf dem Foto zu sehen ist. Die Greifbewegung an dieser Position ist im Vergleich zu den oben beschriebenen für den Roboter komplizierter, da größere Gelenkbewegungen erforderlich sind und die Tasse von der Seite statt direkt gegriffen werden muss.

In der unteren Lernkurve sind die Ergebnisse des Trainings auf dem Roboter dargestellt. Innerhalb der 160 durchgeführten Iterationen konnte der Versatz der Tasse von anfänglich 6cm auf ca. 2cm verringert werden.

In der oberen Lernkurve sind zum Vergleich die Ergebnisse des Trainings in der Simulation dargestellt. In beiden Fällen wurde das Training mit derselben imitierten Bewegung begonnen. Trotzdem ist bereits die erreichte Güte der mit diesen Parametern durchgeführten Bewegung in der Simulation deutlich höher. Dies verdeutlicht den Einfluss der Ungenauigkeiten des realen Roboters auf die Ergebnisse. Im weiteren Verlauf des Trainings konvergieren die Gütewerte in der Simulation auf 0,995, was deutlich über dem Ergebnis des realen Roboters liegt. Auch die Varianz der Werte in der Simulation ist geringer als auf dem realen Roboter.

Besonders das letzte Experiment lässt vermuten, dass die erreichbare Genauigkeit beim Training von Greifbewegungen mit dem in dieser Diplomarbeit vorgestellten Ansatz des verstärkenden Lernens durch die Genauigkeit des Roboters begrenzt ist.

6.3.3 Lernen aller Parameter mit variabler Position der Tasse

Im vorherigen Abschnitt wurde die Fähigkeit des verstärkenden Lernverfahrens, die optimale Greifbewegung in einer gegebenen Situation zu finden, demonstriert. Aufbauend auf diesen Ergebnissen wird in diesem Abschnitt untersucht, in wie weit das Verfahren in der Lage ist, auch für ähnliche aber zuvor nicht demonstrierte Situationen optimale Greifbewegungen zu generieren. Dazu wurde um die Zielposition einer demonstrierten

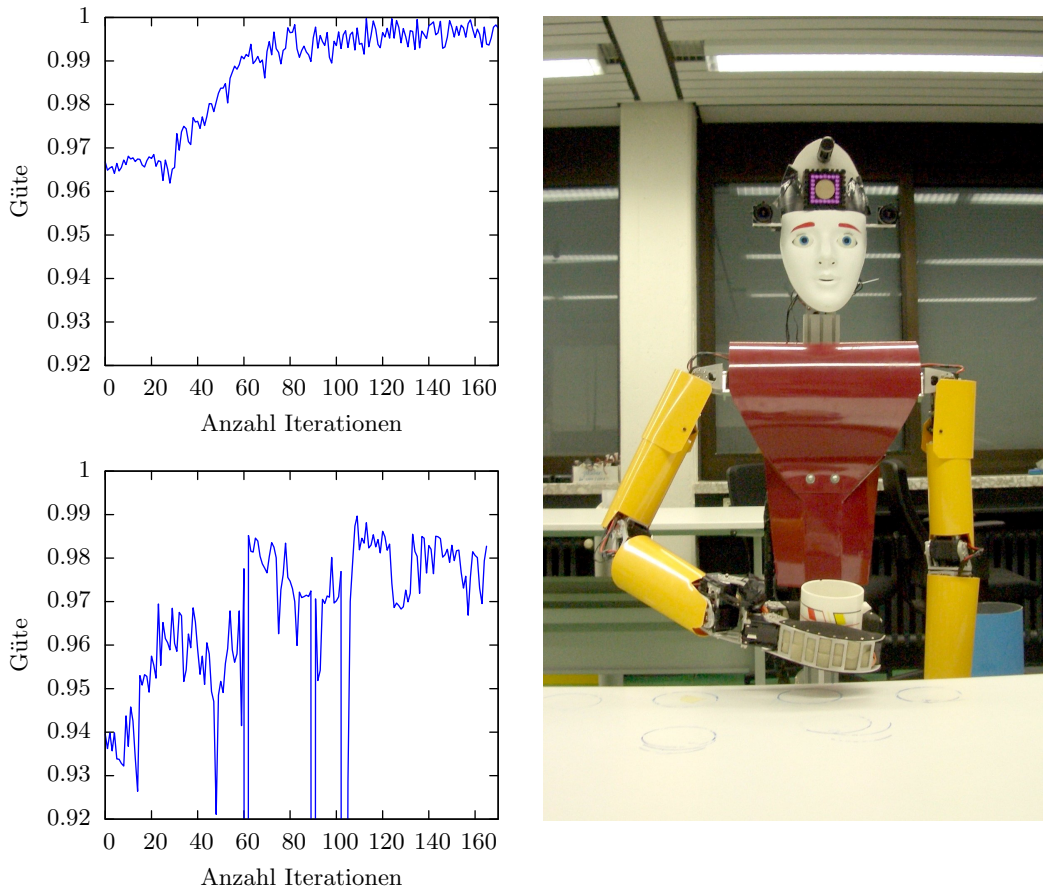


Abbildung 6.14: *Lernkurven für eine feste Tassenposition, die, wie auf dem Bild zu sehen ist, mittig vor dem Roboter lag. Die obere Lernkurve ist das Ergebnis des Versuches in der Simulation, während die untere auf dem realen Roboter entstanden ist. An den Stellen, an denen die Kurve nach unten aus der abgebildeten Skala hinausragt, konnte die Tasse nicht gegriffen werden und der Gütewert lag bei 0.*

Greifbewegung herum ein $50cm^2$ großer Bereich festgelegt, in dem die Tasse zufällig platziert wurde.

In Abbildung 6.15 ist die gemittelte Lernkurve über 25 Versuche dargestellt. Im Vergleich zu Abbildung 6.10, in der des Greifen einer Tasse an einer festen Position gelernt wurde, benötigt das Verfahren ca. 70 Iterationen mehr, um bei weniger als 1cm Versatz zu konvergieren. Zwar wurde die Tasse in jeder Iteration an einer anderen Stelle platziert, es ist jedoch zu sehen, dass die gelernten Informationen über eine Tassenposition auch auf alle anderen Positionen der Tasse Auswirkungen haben. Dadurch steigt die Kurve kontinuierlich und das Lernverfahren macht keine großen Rückschritte.

An dem exemplarisch ausgewählten Parameterverlauf eines Versuches in Abbildung 6.16 ist zu sehen, dass die Offsetwerte konvergieren und für alle Positionen in dem trainierten Bereich zu guten Gütewerten führen. Die beiden Parameter der Hand hingegen schwanken

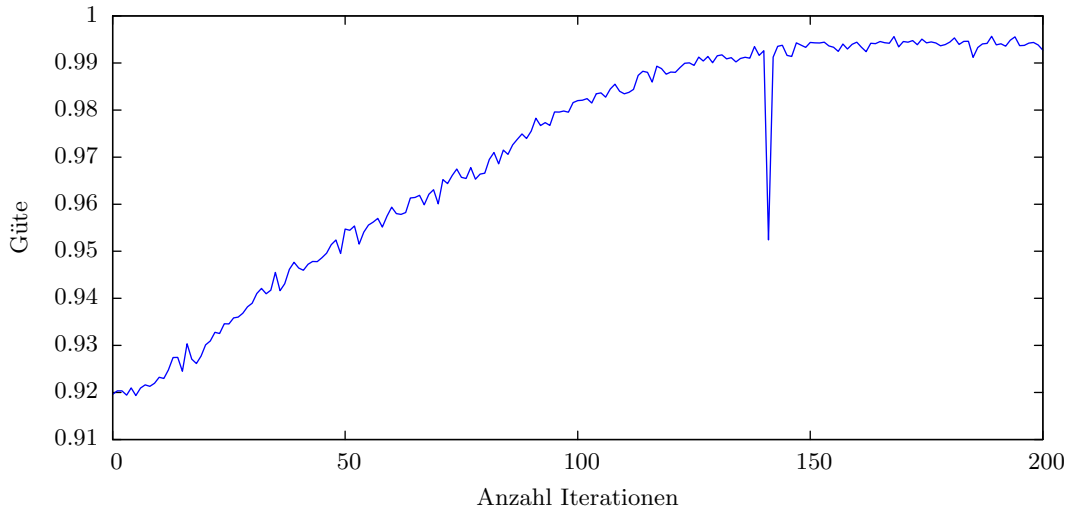


Abbildung 6.15: *Über 25 Versuche gemittelte Lernkurve für eine variable Tassenposition innerhalb eines* $50cm^2$*-Bereichs in der Simulation.*

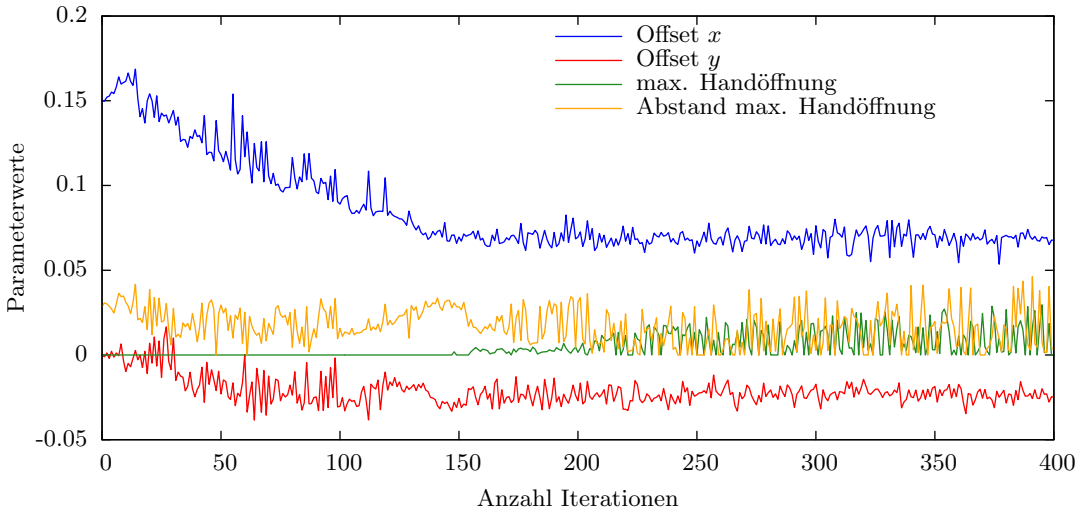


Abbildung 6.16: *Parameterverlauf mit vier Parametern für einen exemplarisch ausgewählten Versuch. Dabei wurde ein Bereich von* $50cm^2$ *in der Simulation gelernt.*

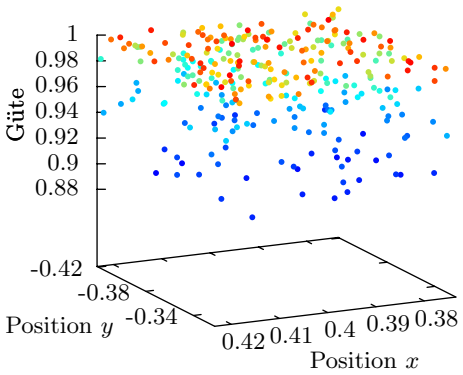


Abbildung 6.17: *Darstellung des Verlaufs der Gütewerte für eine variable Position der Tasse in einem Bereich von* $50cm^2$ *in der Simulation. Auf der x- und y-Koordinatenachse ist die Position der Tasse aufgetragen. Die z-Achse stellt die Güte der getesteten Werte dar. Um den zeitlichen Verlauf darzustellen, wurde die Anzahl der Iterationen in den Farben von blau nach rot kodiert.*

um bis zu 5cm, ohne dass dies Einfluss auf die Gütewerte hat. Dies lässt darauf schließen, dass es hier nicht nur einen guten Wert gibt, sondern ein ganzer Parameterbereich zu einem guten Ergebnis führt.

In Abbildung 6.17 ist auf der x- und y-Achse die Position der Tasse eines Parametersatzes dargestellt. In der z-Richtung ist die Güte der entsprechenden Parameter zu sehen. Die einzelnen Iterationsschritte sind durch die Farbe der Punkte von blau nach rot kodiert.

In der Darstellung ist zu sehen, dass die Farben schichtartig verteilt sind. Dies bedeutet, dass die Parameter für die verschiedenen Positionen der Tasse nicht etwa nacheinander, sondern parallel verbessert werden. Es finden dabei keinerlei Fehlversuche beim Greifen der Tasse statt.

Im Vergleich zu einer zufälligen Suchstrategie zeigt sich, dass die vorgestellte Strategie auf Basis der erwarteten Verbesserung und -Verschlechterung in kürzerer Zeit bessere Parametersätze findet. In Abbildung 6.18 ist oben links eine zufällige Strategie mit kleiner Varianz dargestellt. Es ist zu erkennen, dass diese Strategie zwar beinahe so gute Ergebnisse wie vorgestellte Strategie, die unten rechts dargestellt ist liefert, dabei aber ca. doppelt so viele Iterationen benötigt. Oben rechts ist zu erkennen, dass eine größere Varianz bei der zufälligen Suche schneller zu guten Werten führt. Um ähnlich gute Ergebnisse wie das vorgestellte Verfahren zu erreichen, wird jedoch fast dieselbe Zahl an Iterationen benötigt. Die größere Varianz führt allerdings auch dazu, dass häufig Parameter ausprobiert werden, mit denen die Tasse nicht gegriffen werden kann, und die ein Risiko für den Roboter darstellen. In den Experimenten führten diese Parametersätze in einigen Fällen sogar zu einer Kollision mit dem Tisch. Unten links ist eine Kombination der beiden anderen zufälligen Strategien dargestellt, bei der die Varianz zunächst groß gewählt wurde, und verringert wurde, je besser die gefundenen

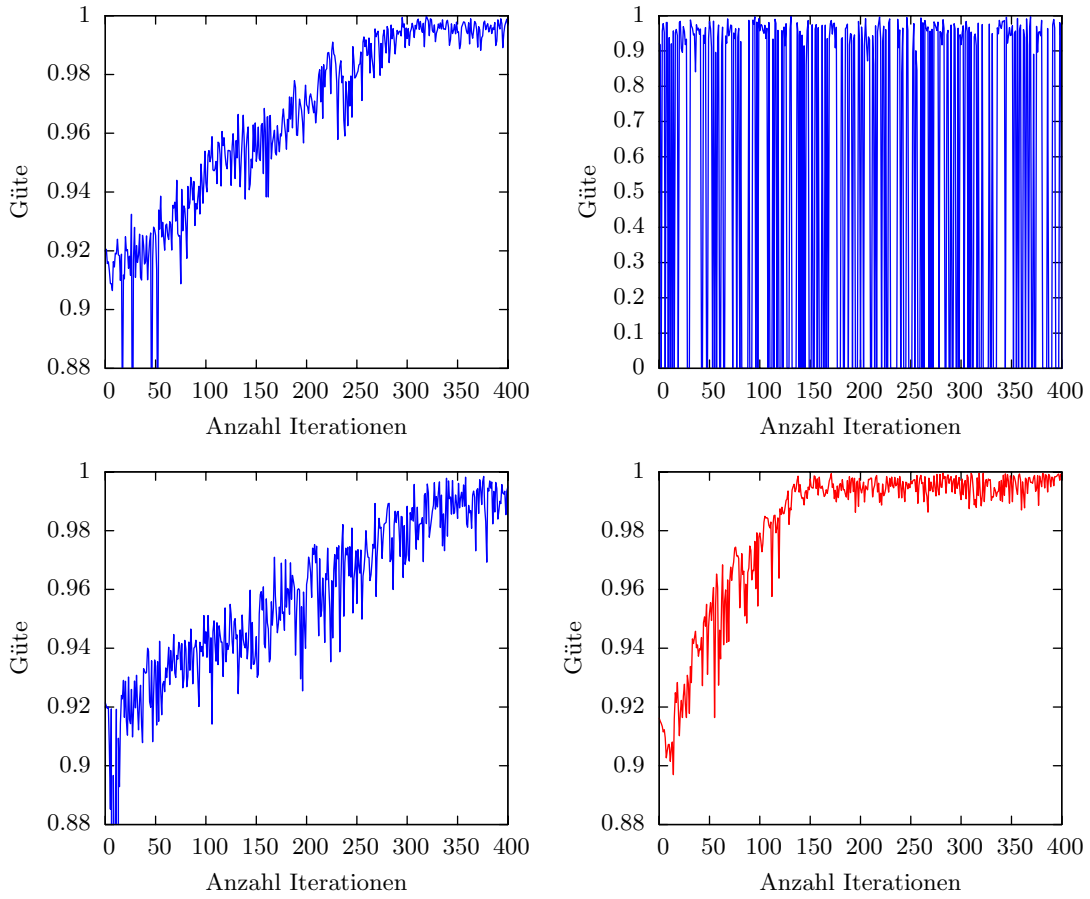


Abbildung 6.18: *Vergleich der in rot dargestellten Suchstrategie des entwickelten Verfahrens mit verschiedenen in blau dargestellten zufälligen Strategien. Für die zufälligen Strategien wurden die Parameter durch ein normalverteiltes, additives Rauschen mit Mittelwert 0 verändert. Oben links wurde eine Varianz von 0,005 verwendet, oben rechts eine Varianz von 0,04. Unten links wurde die Varianz verringert, je näher die erreichte Güte dem theoretischen Maximum von 1 kam. Unten rechts ist das Ergebnis der in dieser Diplomarbeit vorgestellten Strategie auf Basis der erwarteten Verbesserung und -Verschlechterung zu sehen. Werte in den beiden linken Darstellungen, die unterhalb des dargestellten Bereichs liegen, stellen Parametersätze dar, mit denen die Tasse nicht gegriffen werden konnte.*

Parametersätze wurden. Um häufige Fehlschläge, wie sie bei der großen varianz beobachtet wurden, zu vermeiden, wurde die Varianz außerdem verringert, falls die kombinierte Kostenfunktion eine Verschlechterung der Güte erwarten ließ. Es zeigt sich, dass diese Strategie nicht, wie erwartet, zu einer Verbesserung gegenüber einer konstanten Varianz führt, sondern der Lernfortschritt sogar noch langsamer steigt als in den beiden anderen zufälligen Strategien. Dies ist dadurch zu erklären, dass eine große Varianz häufig zu sehr schlechter zu erwartender Güte führt und somit selten angewendet wird.

6.4 Kombiniertes Lernverfahren

In diesem Abschnitt wird das kombinierte Lernverfahren mit Imitations- und verstärkendem Lernen untersucht. Zunächst wird ein Experiment beschrieben, um die Generalisierungsfähigkeit des Verfahrens zu erproben. Diese ist notwendig, um die Anzahl der benötigten Imitationen gering zu halten. Anschließend wird dieses Experiment erweitert, und das Greifen einer Tasse an einer beliebigen Position auf dem gesamten Tisch trainiert.

6.4.1 Generalisierungsfähigkeit des Verfahrens

Um die Generalisierungsfähigkeit des Verfahrens zu evaluieren, wurde das Verhalten in zwei unterschiedlichen Experimenten verglichen. Zunächst wurden Greifbewegungen an zwei unterschiedlichen Stellen auf dem Tisch aufgenommen. Ausgehend von diesen Demonstrationen sollte das Verfahren das Greifen einer Tasse, die mittig zwischen den aufgenommenen Positionen stand, lernen. Die beiden bekannten Positionen befanden sich in ca. 20cm Entfernung voneinander. Um nur mit verstärkendem Lernen zwischen diesen beiden Positionen lernen zu können, wurde die Tasse zunächst auf 4 Zwischenpositionen gestellt, um die mittlere Position anzunähern. Im ersten Experiment bestanden die Traingingsbeispiele ausschließlich aus den Parametersätzen der beiden demonstrierten Bewegungen. Für das zweite Experiment wurden diese Parametersätze zunächst durch verstärkendes Lernen optimiert. Anschließend wurde untersucht, wie sich die dabei gewonnenen Informationen auf das Lernen an der mittleren Position auswirken.

Experimente in der simulierten Umgebung

In Abbildung 6.19 sind die Lernkurven beider Experimente nebeneinander dargestellt. Die rechte Grafik zeigt, dass durch das verstärkende Lernen bereits zu Beginn ein gemittelter Gütewert von 0,987 erreicht wurde, was deutlich über dem Wert des ersten Experiments liegt, der im Mittel nur knapp 0,8 betrug. Außerdem konnte im ersten Experiment die Tasse häufig nicht gegriffen werden, wohingegen im zweiten Experiment keine Fehlgriffe auftraten. Beide Lernkurven konvergieren gegen denselben Wert. Dabei wurden im ersten Experiment 100 Iterationen benötigt, wohingegen die optimale Greifbewegung im zweiten Experiment durch Ausnutzung der vorhandenen Informationen bereits nach 20 Iterationen erreicht wurde.

Ebenso ist in dem in Abbildung 6.20 dargestellten Verlauf der Parameter zu erkennen, dass der Lernvorgang im zweiten Experiment bei guten Parameterwerten startet. Wie schon in Abbildung 6.16, zeigt sich auch hier, dass die Offsetwerte nur für bestimmte

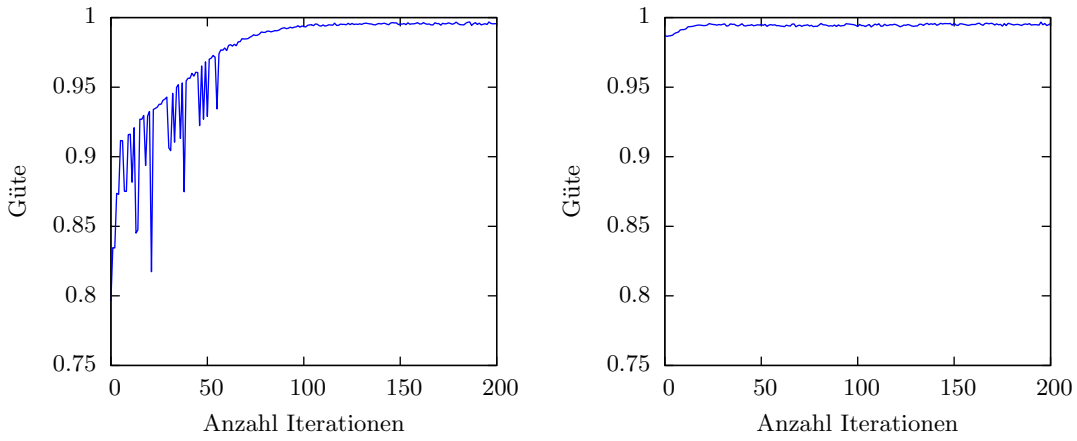


Abbildung 6.19: *Die Darstellung zeigt gemittelte Lernkurven über 25 Versuche in der Simulation zum Lernen mit Vorwissen. Dabei ist auf der linken Seite das erste Experiment zu sehen, bei dem das Vorwissen nur aus zwei Imitationsbeispielen bestand. In der rechten Abbildung ist das zweite Experiment dargestellt, das zwei gelernte, unterschiedliche Positionen der Tasse als Vorinformationen erhielt.*

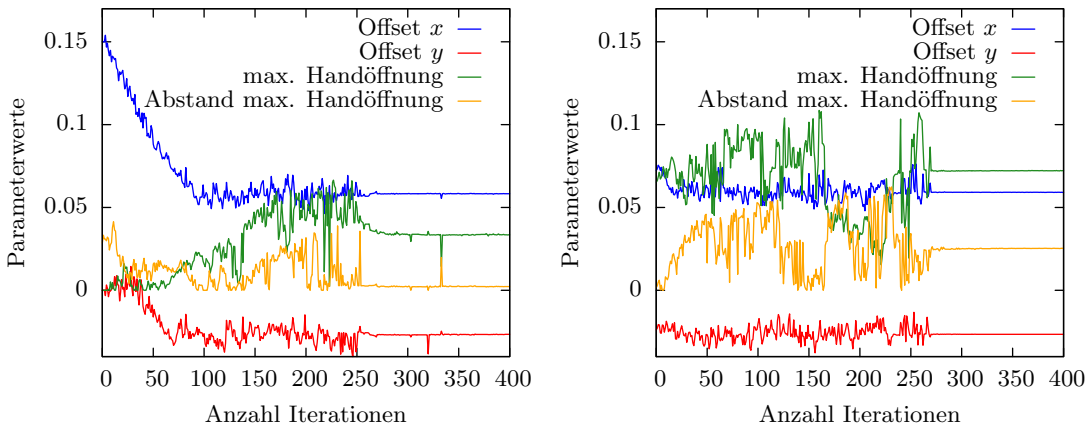


Abbildung 6.20: *Bei den beiden dargestellten Parameterverläufen handelt es sich um exemplarisch ausgewählte Versuche für das Lernen mit Vorwissen. Links ist dabei der Verlauf mit nur wenig Vorwissen in Form von zwei Imitationsbeispielen dargestellt. Auf der rechten Seite ist der Verlauf nach dem Lernen dieser beiden Imitationsbeispiele zu sehen. Beide Versuche fanden in der Simulation statt.*

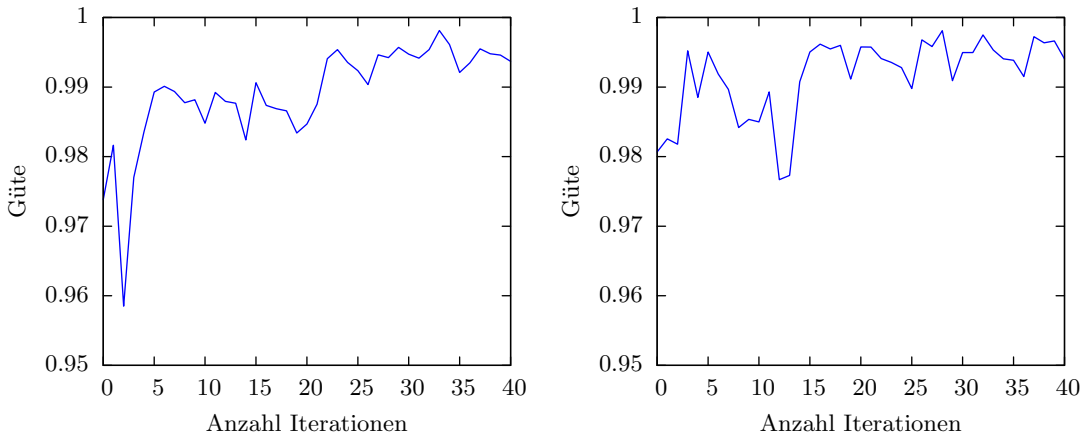


Abbildung 6.21: *Vergleich der Lernkurven mit viel bzw. wenig Vorwissen auf dem realen Roboter. In der linken Darstellung ist die zum ersten Experiment gehörende Lernkurve zu sehen, bei dem nur zwei Imitationsbeispiele als Trainingsdaten vorlagen. Im zweiten Experiment, das in der rechten Abbildung dargestellt ist, lagen zwei gelernte Imitationsbeispiele vor.*

Werte gute Ergebnisse liefern, während die Parameter zur Handöffnung über ein ganzes Intervall gute Greifbewegungen erzeugen.

Experimente auf dem realen Roboter

Das oben beschriebene Experiment wurde ebenfalls auf dem realen Roboter durchgeführt. Auch hier zeigt die rechte Grafik in Abbildung 6.21, dass die zusätzlichen Informationen aus dem verstärkenden Lernen auf ähnliche Tassenpositionen generalisiert werden können und den Lernprozess für die mittlere Position beschleunigen. Das erste Experiment in der linken Grafik erzielt ebenfalls einen guten Lernerfolg. Dieser ist auf die gute Initialisierung durch das Imitationslernen zurückzuführen. Während im ersten Experiment ca. 25 Iterationen benötigt werden, um einen Gütewert von 0,995 zu erreichen, nutzt das Lernverfahren im zweiten Experiment die gegebenen Informationen und erreicht den Wert 0,995 schon nach 4 Schritten. Das zeitweise Abfallen der Kurve ist durch das Testen neuer Parametersätze zu erklären.

6.4.2 Lernen auf dem gesamten Tisch

In diesem Experiment wurden Greifbewegungen an beliebigen Positionen im Aktionsradius des Roboters auf dem Tisch mit dem integrierten Lernverfahren trainiert. Die Positionen der Tasse wurden dabei zufällig bestimmt. Wie in Abschnitt 5.3 beschrieben, wurde die Entscheidung, ob Imitations- oder verstärkendes Lernen eingesetzt wurde durch das Lernverfahren vorgenommen. In Abbildung 6.22 sind die Positionen der Tasse, für die das Lernverfahren eine Demonstration verlangt hat, gekennzeichnet. Insgesamt wurden für einen Tisch der Größe $600cm^2$ 15 Imitationen benötigt, von denen 13 in den ersten 24 Iterationen und 2 weitere bis zur 42. Iteration angefragt wurden.

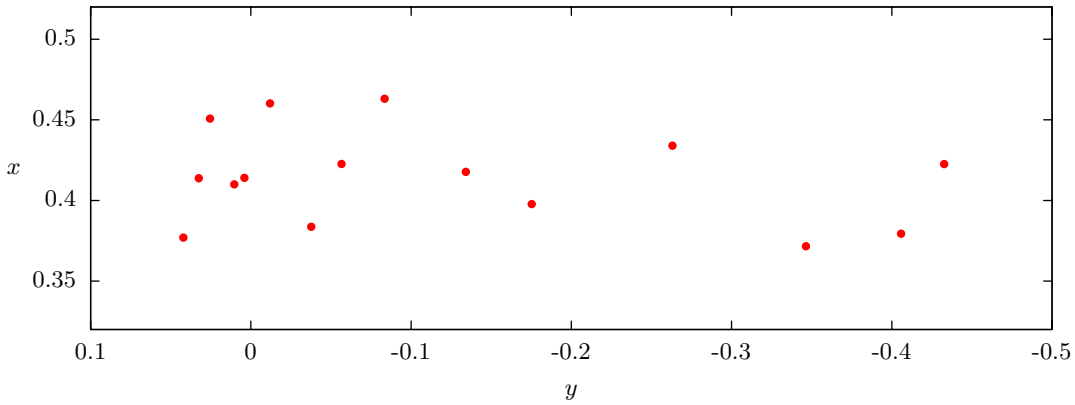


Abbildung 6.22: *Darstellung der Positionen für die in der Simulation eine Demonstration gefordert wurde. Dabei ist der durch das Koordinatensystem eingefasste Bereich derjenige, in dem gelernt wurde.*

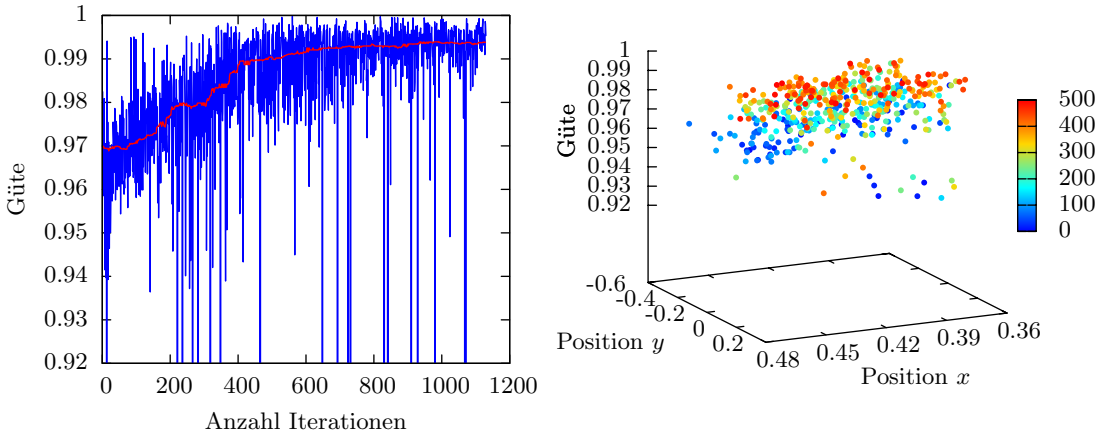


Abbildung 6.23: *In der linken Darstellung ist die Lernkurve zu sehen, die während des Lernprozesses für eine Fläche von 600cm^2 in der Simulation entstanden ist. An den Stellen, an denen die Kurve aus der abgebildeten Skala hinausragt, konnte die Tasse nicht gegriffen werden und der Gütewert lag bei 0. Die blaue Kurve stellt dabei die wahren Gütewerte dar, während die rote Kurve mit Hilfe des Medians über 100 Iterationen geglättet wurde. Im rechten Bild ist der Verlauf der Gütewerte über den gesamten Bereich der möglichen Tassenpositionen dargestellt. In der x- und y-Koordinate ist dabei die Position der Tasse aufgetragen. Die z-Achse gibt den dazugehörigen Gütewert an. Der aktuelle Iterationsschritt ist in der Farbe kodiert. Um die Übersichtiglichkeit zu erhalten, wurden nur die ersten 500 Iterationen abgebildet und die Beispiele, bei denen die Tassen nicht gegriffen werden konnte, ausgelassen.*

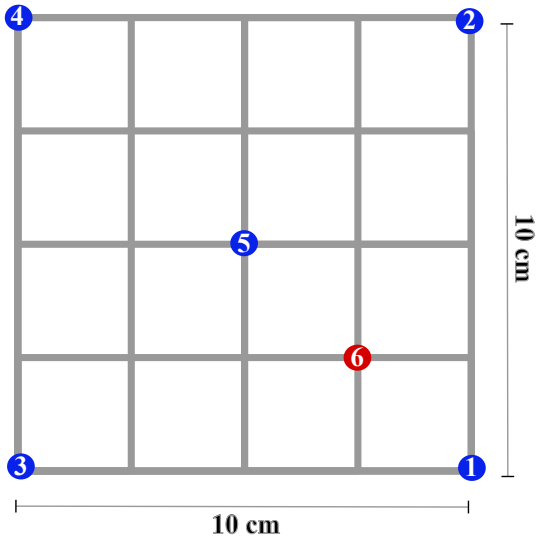


Abbildung 6.24: *Raster, um festzustellen, wann verstärkendes Lernen in Abhängigkeit von der Nähe zum nächsten Trainingsbeispiel gewählt wird. Dabei stellen die Zahlen die Reihenfolge der angefragten Positionen dar. Bei blauen Punkten fiel die Entscheidung auf Imitationslernen, bei Roten auf verstärkendens Lernen. Das Experiment wurde in der Simulation druchgeführt.*

In Abbildung 6.23 ist die Lernkurve für dieses Experiment dargestellt, welches mit Hilfe der Simulation durchgeführt wurde. Im Vergleich zu den Lernkurven anderer Experimente wurden hier deutlich mehr Iterationen benötigt, bis das Verfahren konvergierte. Dies liegt an der großen Zahl unterschiedlicher Positionen, für die Greifbewegungen gelernt werden sollten.

Rechts neben der Lernkurve sind die in den einzelnen Iterationen durchgeführten Greifbewegungen zusammen mit der erreichten Güte dargestellt. Die x- und y-Koordinaten entsprechen dabei der Position auf dem Tisch. Der Verlauf des Lernverfahrens ist durch die Farbe der Punkte kodiert, die im Verlauf des Verfahrens von blau über grün und gelb nach rot wechselt. Der Übersichtlichkeit halber sind dabei nur die ersten 500 Iterationen dargestellt und Punkte, an denen die Tasse nicht gegriffen werden konnte, sind nicht dargestellt. Die Grafik spiegelt die zufällige Verteilung der Positionen der Tasse zu jedem Zeitpunkt und die Verbesserung der Gütewerte im Verlauf des Verfahrens wieder.

Die Größenordnung, in der es ein gutes Beispiel geben muss, damit verstärkendes Lernen möglich ist, ist abhängig von den Werten der Parameter des Gauß-Prozesses, besonders der Kernelbreiten. Um diese Entfernung experimentell für die in dieser Diplomarbeit verwendeten Parameter zu bestimmen, wurde in einem weiteren Experiment das Verfahren für eine Reihe von Tassenpositionen nach einem vorgegebenen Schema angewandt. Dieses Schema ist in Abbildung 6.24 dargestellt. Die Reihenfolge der gewählten Positionen ist in der Abbildung durch Zahlen angegeben. Die Positionen auf dem Tisch, an denen das Verfahren zu wenig Vorwissen hatte und daher das Imitationslernen gewählt wurde, sind durch blaue Kreise gekennzeichnet. Positionen mit einer Entscheidung für das verstärkende Lernen sind durch rote Kreise gekennzeichnet. Zunächst wurden Greifbewegungen für

die Eckpunkte eines 10×10cm großen Bereichs trainiert. In allen Fällen verlangte das Lernverfahren dafür eine Demonstration. Anschließend wurden die Seiten des Bereichs schrittweise halbiert und die Tasse an den so entstandenen Eckpunkten plaziert. Für eine Seitenlänge von 5cm verlange das Verfahren noch eine Demonstration, wohingegen es bei einer Seitenlänge von 2,5cm das verstärkende Lernen wählte. Dies entspicht einem Abstand von 3.5cm zu dem nächstgelegenen Trainingsbeispiel. Die Gütewerte der imitierten Bewegungen lagen bei ca. 0,98.

Diese Experimente zeigen, dass die Anzahl der benötigten Imitationen von der Reihenfolge der gewählten Positionen der Tasse und den Gütewerten der Trainingsbeispiele in deren Nähe abhängt.

6.5 Zusammenfassung der Experimente

In diesem Kapitel wurde anhand von Experimenten die Lernfähigkeit des vorgestellten Ansatzes demonstriert. Zunächst wurden dazu die beiden Teilverfahren, das Lernen durch Imitation und das verstärkende Lernen, getrennt voneinander evaluiert. Dabei wurde gezeigt, dass das Imitationslernen die menschliche Trajektorie mit einer mittleren quadratischen Abweichung von 10^{-4}m pro Punkt und einem Zeitunterschied von 10^{-6}s auf die Trajektorie des Roboters abbilden konnte und der Roboter somit eine fast identische Trajektorie zu der des Menschen ausführt.

Beim verstärkenden Lernen wurde sowohl die Strategie des Verfahrens, als auch die dadurch entstehende Verbesserung untersucht. Unter Verwendung der vorgestellten Suchstrategie wies der Roboter ein zielstrebiges Verhalten auf, so dass bei einer festen Tassenposition und vier zu lernenden Parametern nach ca. 70 Iterationen das Maximum erreicht wurde. Der Roboter konnte dabei die Tasse mit einem Versatz von unter 1cm greifen. In der Evaluation auf dem realen Roboter konnte gezeigt werden, dass das Verfahren mit verrauschten Beobachtungen umgehen kann und der Versatz der Tasse auf einen Wert unter 1cm reduziert werden kann.

Ebenso wurde gezeigt, dass gelernte Bewegungen auf unterschiedliche Positionen der Tasse generalisiert werden können. Um Greifbewegungen in einem $50cm^2$ großen Bereich zu lernen, wurden ca. 140 Iterationen benötigt, bis für alle Positionen die optimale Bewegung generiert werden konnte. Dabei ist zu erkennen, dass Trainingsbeispiele auch Einfluss auf benachbarte Positionen haben.

Um das kombinierte Verfahren zu validieren, wurde gezeigt, dass dieses anhand von nur 15 vorgeführten Bewegungen die Tasse an beliebigen Positionen auf den Tisch greifen konnte. Um diese allerdings alle optimal greifen zu können, musste der Roboter in der Simulation ca. 1000 Mal greifen, was auf dem realen Roboter nicht durchführbar ist. Da der Versatz der Tasse allerdings schon durch die Imitationen bei nur ca. 4cm liegt, wäre es möglich, dass der reale Roboter online während seines normalen Einsatzes weiterlernt.

7 Zusammenfassung und Ausblick

Kapitel

7.1 Zusammenfassung

Im Rahmen dieser Diplomarbeit wurde ein interaktives Lernverfahren entwickelt, welches die Vorteile von Imitations- und verstärkendem Lernen kombiniert. Evaluiert wurde das Lernverfahren anhand der exemplarischen Aufgabe, einem humanoiden Roboter das anthropomorphe Greifen einer Tasse beizubringen. Die Tasse konnte sich an jeder beliebigen Position auf einem Tisch befinden. Dazu wurde ein parametrisierter Regler entwickelt, mit dem anthropomorphe Bewegungen generiert werden können.

Die 29 Eingabeparameter des Reglers wurden im Rahmen des Trainings durch das Lernverfahren bestimmt. Sie beschreiben charakteristische Merkmale menschlicher Bewegungen. Anhand dieser wird eine glatte Trajektorie aus Positionen, Orientierungen und Werten für die Öffnung der Hand generiert.

Die Qualität einer Greifbewegung wird mit Hilfe einer Gütefunktion ermittelt. Diese verwendet als Kriterium $(1 - v)$, wobei v der Versatz der Tasse nach dem Greifen ist. Konnte die Tasse nicht gegriffen werden, fällt die Bewertung neutral aus. Wenn es allerdings Kollisionen mit dem Tisch oder Probleme mit der inversen Kinematik gab, so ist diese negativ.

Die Bewertung einer Bewegung stellt zusammen mit den zugehörigen Parametern des Reglers ein Trainingsbeispiel für das Lernverfahren dar. Zur Generalisierung der Trainingsbeispiele auf veränderte Situationen werden innerhalb des Verfahrens Gauß-Prozesse verwendet. Diese liefern zu jedem Parametersatz den Mittelwert und die Standardabweichung der zu erwartenden Güte. Da in dieser Diplomarbeit eine Gütefunktion verwendet wird, deren Werte nicht normalverteilt sind und somit nicht den Annahmen von Gauß-Prozessen entsprechen, wird für jeden der drei Gütefälle ein eigener Gauß-Prozess verwendet. Aus den Verteilungsparametern dieser Gauß-Prozesse werden durch Gewichtung mit ihrer Konfidenz ein gemeinsamer Mittelwert und eine Varianz der zu erwartenden Güte berechnet.

Um das Greifen an einer gegebenen Position zu lernen, entscheidet das Verfahren zunächst, ob dies durch Imitations- oder verstärkendes Lernen geschehen soll. Diese Entscheidung wird auf Grundlage des bisher erhaltenen Wissens getroffen, welches durch die Ausgaben der Gauß-Pozesse repräsentiert wird. Um eine Prädiktion von den Gauß-Prozessen zu erhalten, muss ein vollständiger Parametersatz vorliegen. Für die gegebene

Tassenposition müssen daher zunächst passende Parameter gefunden werden. Ausgehend von einem geeigneten Trainingsbeispiel wird dazu mittels des Lernverfahrens Rprop nach Parametern gesucht, die sowohl eine hohe Güte, als auch eine geringe Unsicherheit aufweisen. Für den auf diese Weise bestimmten Parametersatz wird anschließend anhand der Differenz zwischen dem Mittelwert und der erwarteten Verschlechterung bestimmt, ob genügend Informationen vorliegen, um auf eine Imitation verzichten zu können. In diesem Fall werden die Parameter der Greifbewegung nur mit Hilfe von verstärkendem Lernen optimiert.

Beim Imitationslernen soll die Trajektorie des Roboters möglichst genau an die des Menschen angepasst werden. Da die Bewegung durch den hier entwickelten Regler aus Parametern generiert wird, müssen geeignete Parameter bestimmt werden. Die menschliche Bewegung wird mithilfe einer Motion Capture-Anlage aufgezeichnet. Zusätzlich wird ein Datenhandschuh benutzt, um die Öffnung der Hand über die Zeit zu bestimmen. Ein Teil der gesuchten Parameter ist aus diesen Aufnahmen direkt berechenbar. Die anderen werden mit Hilfe des Downhill-Simplex-Verfahrens iterativ bestimmt, da für die verwendete Kostenfunktion keine geschlossene Formel vorliegt und somit auch kein Gradient bestimmt werden kann.

Falls die Entscheidung auf verstärkendes Lernen gefallen ist, so wird mit Hilfe von Rprop nach einem optimalen Parametersatz im kontinuierlichen Parameterraum gesucht. Dabei ist die Gütefunktion eine Kombination aus der erwarteten Verbesserung und Verschlechterung. Die erwartete Verbesserung favorisiert Parameter, die eine große Verbesserung der Güte versprechen, unabhängig von der möglichen Verschlechterung. Die erwartete Verschlechterung verhindert die Wahl von Parametern mit potentiell großer Verschlechterung der Güte, die zu Schäden am Roboter führen könnten.

7.2 Beitrag der Arbeit

Das in dieser Diplomarbeit entwickelte Lernverfahren erlaubt es humanoiden Robotern, anthropomorphe Greifbewegungen zu erlernen. Dabei wurde ein integrierter Ansatz verfolgt, der Imitations- und verstärkendes Lernen kombiniert. Es konnte gezeigt werden, dass auf diese Weise die Vorteile beider Verfahren ausgenutzt werden können.

Das Imitationslernen erlaubt es dem Roboter, in unbekannten Situationen menschliches Vorwissen mit einzubeziehen. Aufbauend auf diesen Erfahrungen kann er anschließend durch verstärkendes Lernen seine Strategie optimieren. Dabei wird die Suche durch die Beispiele des Imitationslernens fokussiert und so die Lerngeschwindigkeit des Verfahrens erhöht. Gleichzeitig kann durch den Einsatz von verstärkendem Lernen die Anzahl der vorzuführenden Bewegungen deutlich reduziert werden.

Das Verfahren ist so entworfen worden, dass es von Laien bedienbar ist und keinerlei Vorkenntnisse bedarf, da der Roboter auf eine für den Menschen sehr intuitive Art und Weise lernt.

Ein weiterer Beitrag dieser Arbeit ist die Entwicklung einer probabilistischen Strategie zur Wahl der nächsten Aktion des verstärkenden Lernens. Sie bezieht sowohl die erwartete Verbesserung, als auch die erwartete Verschlechterung eines Parameters in die Entscheidung ein. Auf diese Weise ist eine gerichtete Suche möglich, bei der gleichzeitig die Bewegung des Roboters optimiert und Bewegungen, die den Roboter beschädigen

könnten, vermieden werden.

Zur niedrigdimensionalen Repräsentation von Bewegungen wurde in dieser Diplomarbeit ein parametrisierter Regler entwickelt, der ein breites Spektrum anthropomorpher Bewegungen erzeugen kann. Die Parameter des Reglers beschreiben dabei intuitive Merkmale der Bewegungen. Sie lassen sich teilweise direkt, teilweise iterativ aus Motion Capture-Aufnahmen bestimmen, um auf diese Weise anthropomorphe Bewegungen zu erhalten.

7.3 Ausblick

Mit dem in dieser Diplomarbeit entwickelten Lernverfahren können anthropomorphe Greifbewegungen eines humanoiden Roboters gelernt werden. Im Folgenden werden Ansatzpunkte für eine zukünftige Weiterentwicklung beschrieben. Diese haben vor allem eine Verbesserung der Lerngeschwindigkeit und eine Erweiterung der Anwendungsmöglichkeiten des Verfahrens zum Ziel.

Bestimmung der Parameter
In dem beschriebenen Lernverfahren werden Gauß-Prozesse zur Generalisierung der Trainingsbeispiele und für die Suchstrategie des verstärkenden Lernens verwendet. Die Einstellung der Hyperparameter der Gauß-Prozesse wurde in dieser Diplomarbeit von Hand vorgenommen und die Werte empirisch bestimmt. Eine mögliche Weiterentwicklung besteht darin, diese Parameter automatisch aus vorhandenen Daten zu ermitteln.

Verbesserung der Lerngeschwindigkeit
Die Lerngeschwindigkeit des Verfahrens kann auf zwei Arten verbessert werden. Eine Möglichkeit ist, die Anzahl der für einen guten Lernerfolg benötigten Iterationen zu reduzieren. Die zweite Möglichkeit besteht in der Verringerung der Laufzeit einer einzelnen Iteration des Verfahrens.

Die Anzahl der benötigten Iterationen hängt von der Suchstrategie des verstärkenden Lernens ab. Das in dieser Arbeit vorgestellte Verfahren sucht mit dem Gradientenabstiegsverfahren Rprop eine optimale Kombination aus erwarteter Verbesserung und Verschlechterung. Eine mögliche Verbesserung dieser Strategie wäre ein hierarchischer Ansatz, bei dem die Kostenfunktion auf mehreren Glättungsstufen betrachtet wird. Die Suche nach geeigneten Parametern würde auf der obersten Schicht mit der stärksten Glättung beginnen. Wurde hier ein Maximum auf der Kostenfunktion gefunden, so wird die Suche an dieser Stelle in der darunterliegenden Schicht fortgesetzt. Durch ein solches Verfahren ließe sich der Suchraum über die lokale Nachbarschaft hinaus erweitern und Probleme mit lokalen Nebenmaxima würden verringert. Hierfür müsste untersucht werden, inwieweit sich eine solche Hierarchie durch Gauß-Prozesse mit unterschiedlichen Kernelbreiten repräsentieren lässt.

Um die einzelnen Iterationen des Verfahrens zu beschleunigen, könnte die Anzahl der Trainingsbeispiele begrenzt werden. Dies könnte durch so genannte *spärliche* Ansätze zur Gauß-Prozess-Regression erreicht werden. Diese haben jedoch den Nachteil, dass die benötigte Anzahl Trainingsbeispiele für eine hinreichend gute Approximation im Voraus schwer abzuschätzen ist. Ein alternativer Ansatz besteht darin, zwar alle Trainingsbeispiele zu speichern, für den Prädiktionsschritt aber immer nur eine Teilmenge zu verwenden

und so den Berechnungsaufwand zu reduzieren. Da bei dem in dieser Diplomarbeit vorgestellten Lernverfahren immer nur Prädiktionen für einen einzigen Punkt bestimmt werden, könnte die Menge der zu verwendenden Trainingsbeispiele basierend auf dem Abstand zu diesem Punkt gewählt werden. Dieser Parameter wäre deutlich intuitiver als eine Grenze bezüglich der Anzahl der Trainingsbeispiele. Die die Wahl der Punkte könnte durch eine geeignete Datenstruktur effizient gestaltet werden. Eine alternative Möglichkeit besteht darin, den Parameterraum in eine feste Anzahl Bereiche aufzuteilen, und nur die Beispiele eines Bereiches zu berücksichtigen. Dies würde eine inkrementelle Schätzung der Matrix C^{-1} für die einzelnen Bereiche erlauben, wodurch der Rechenaufwand erheblich verringert werden könnte.

Weitere Einsatzmöglichkeiten des Verfahrens
Um den vorgestellten Ansatz zu untersuchen, wurde in dieser Diplomarbeit das Greifen einer Tasse auf einem Tisch trainiert. Dabei wurde zur Vereinfachung eine konstante Höhe des Tisches angenommen. Indem weitere Parameter, wie z.B. die Höhe in der sich ein Objekt befindet und dessen Größe, betrachtet werden, könnten unterschiedliche Greifbewegungen gelernt werden. So könnte der Roboter entscheiden, ob er ein Objekt von oben oder horizontal greifen muss. Desweiteren könnte er flache Objekte durch Drehen der Hand vom Tisch heben oder sogar bei großen Objekten beide Hände einsetzen. Um diese Erweiterungen zu implementieren, müsste zusätzlich zu den bisherigen Parametern die Orientierung aus der menschlichen Bewegung extrahiert werden und der Regler dementsprechend angepasst werden.

Neben dem Greifen von Objekten kann das entwickelte Lernverfahren auch für viele andere Aufgaben eingesetzt werden. Dazu gehören besonders Aufgaben, die der Mensch zwar vorführen kann, bei denen er die Lösung allerdings nicht beschreiben kann. Ein Beispiel für eine solche Aufgabe ist das Zeigen auf ein Objekt. Die Imitationsbeispiele könnten dabei aus vom Menschen demonstrierten Zeigebewegungen bestehen, deren Posen über die Zeit gespeichert werden. Zusätzlich müsste der angezeigte Punkt erfasst werden. Die Bewertung der Bewegung des Roboters könnte durch einen Menschen erfolgen, der mit Hilfe eines Motion Capture-Markers oder eines Laserpointers anzeigt, an welche Position der Roboter seiner Meinung nach gezeigt hat.

Das in dieser Diplomarbeit vorgestellte verstärkende Lernverfahren kann z.B. auch zur Verbesserung der Laufbewegungen humanoider Roboter eingesetzt werden. Wie gut ein Roboter läuft, wie sehr er dabei schwankt und wie oft er umfällt ist ebenfalls abhängig von Parametern. Als Initialisierung des Lernverfahrens könnten dabei die vorhandenen Parameter dienen.

Ein weiterer möglicher Anwendungsbereich des hier beschriebenen Lernansatzes sind komplexere Lernaufgaben, die nicht nur aus dem Lernen von Bewegungsprimitiven bestehen. Beispielsweise könnte das Verfahren in verschiedenen Schichten eines hierarchischn Systems eingesetzt werden. Auf dessen unterster Ebene würden, wie in dieser Diplomarbeit vorgestellt, Bewegungsprimitive gelernt, die auf höheren Ebenen zu komplexen Bewegungsabläufen zusammengesetzt würden.

Technische Verbesserungen
Schließlich könnte das Lernverfahren technisch so weiterentwickelt werden, dass es an beliebigen Orten anwendbar ist. In dieser Diplomarbeit wurde eine Motion Capture-Anlage

zur Erfassung von Bewegungen genutzt. Dies hatte den Vorteil, dass die Bewegungen direkt als dreidimensionale Trajektorien vorlagen und keine zusätzlichen Verfahren notwendig waren, um diese zu bestimmen. Allerdings war das Verfahren dadurch auf den Aufnahmebereich der Motion Capture-Anlage beschränkt und das Training war nur in einem speziellen Motion Capture-Anzug möglich. Um das System praxistauglich zu machen, müsste die Aufnahme der Bewegungen direkt durch den Roboter erfolgen. Dies könnte zum einen über Kameras geschehen oder über Beschleunigungs- und Lagesensoren, die in der Hand gehalten werden könnten.

A Verwendete Hardware

Anhang

A.1 Motion Capture-Anlage

Bei einer Motion Capture-Anlage handelt es sich um ein digitales Bewegungserfassungssystem. Damit ist das Aufzeichnen von Bewegungen in Echtzeit und das Umwandeln in ein digitales Format zur Weiterverarbeitung durch einen Computer möglich. Es können sowohl Bewegungen von einzelnen oder mehreren Personen, als auch von Tieren oder Objekten aufgenommen werden.

Anwendung finden Motion Capture-Anlagen in vielen Bereichen. Dazu gehören sowohl die Animation von Charakteren in der Film- und Unterhaltungsindustrie, als auch Anwendungen in der Medizin zur Analyse von Bewegungen. Auch in der Robotik können Motion Capture-Anlagen in der Forschung eingesetzt werden. Sie werden meist für Aufgaben benötigt, bei denen die Körperbewegung besonders natürlich erscheinen soll. Zum einen ist es kaum möglich, die komplexen Einzelheiten und die dynamischen Zusammenhänge menschlicher Bewegungen von Hand zu erstellen, zum anderen ist die Erfassung der Bewegung mit Hilfe dieser Anlagen deutlich weniger zeitaufwendig.

Es existieren sehr viele unterschiedliche Arten von Motion Capture-Systemen. Hauptsächlich unterscheiden sie sich in ihren Sensoren. Im Rahmen dieser Diplomarbeit wird ein optisches System benutzt. Dieses ist die meistgenutzte Art der Motion Capture-Systeme. Dabei wird die aufzunehmende Person an jedem Körperglied mit speziellen infrarot-reflektierenden Markern ausgestattet, die einfallendes Licht in die Richtung reflektieren, aus der es gekommen ist. Die Person führt ihre Bewegungen in einem bestimmten Bereich aus, der von Infrarotkameras erfasst wird. Diese sind mit Infrarot-Scheinwerfern ausgestattet, die die Marker beleuchten, so dass diese im Kamerabild gut erkennbar sind. Da die Marker von mehreren Kameras gleichzeitig erfasst werden, können mit Hilfe von Schnittpunktberechnungen ihre 3d-Positionen im Raum bestimmt werden. Dazu müssen die Kameras synchronisiert sein und mehrmals pro Sekunde Bilder liefern. Die Verfolgung der Marker über die Zeit spiegelt die vorgeführte Bewegung wieder, die von einer virtuellen Person auf dem Bildschirm ausgeführt wird.

Die im Rahmen dieser Diplomarbeit verwendete optische Motion Capture-Anlage stammt von der Firma *OptiTrack* [39]. Zu dem von OptiTrack angebotenen System gehören 12 Kameras vom Typ *Flex:V100*. Mit diesen können die 2,5cm großen Marker, die ebenfalls bei OptiTrack erhältlich sind, bis in eine Entfernung von ca. 6m erkannt

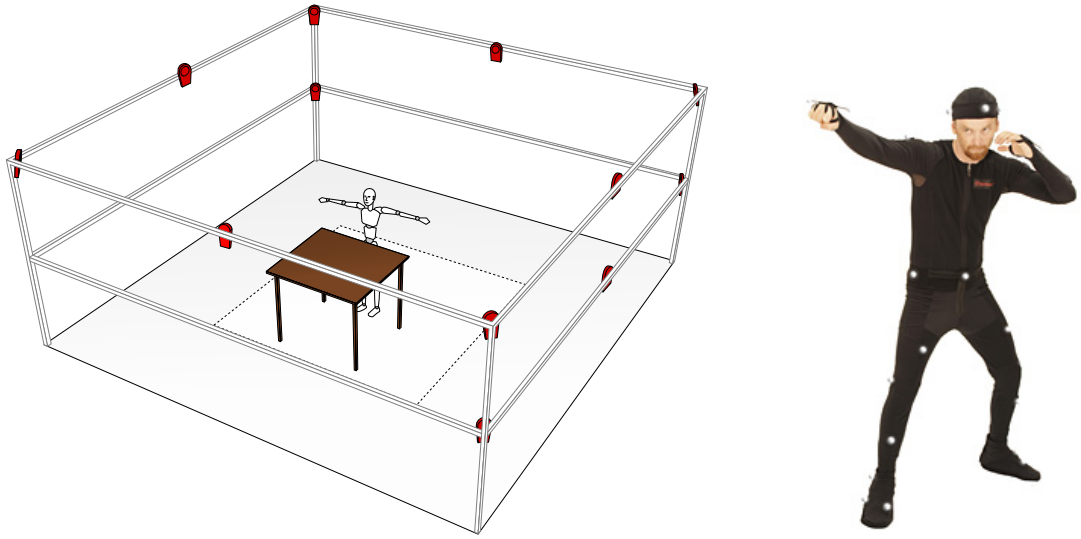


Abbildung A.1: *Kameraanordnung der Motion Capture-Anlage in der Abteilung AIS der Universität Bonn und der verwendete Motion Capture-Anzug (Quelle: [39])*

werden. Die Kameras liefern dabei Bilder mit einer Auflösung von 640×480 Pixeln und einer Bildwiederholfrequenz von 100 Hz. Angeschlossen werden sie über ein USB 2.0 Kabel, welches auch die Stromversorgung der Kameras sicherstellt. Eine zusätzliche externe Stromversorgung ist nicht notwendig.

Zusätzlich zu den Kameras gehört zu dem Motion Capture-System ein spezieller schwarzer Anzug, auf dem die Marker angebracht werden können, und Zubehör zur Kalibrierung der Anlage.

Je nach Anwendungszweck bietet OptiTrack verschiedene Programme für die Motion Capture-Anlage an. In dieser Diplomarbeit wird die Software *Arena* verwendet, die das Aufzeichnen von Bewegungen des ganzen Körpers ermöglicht. Diese ist im Anhang B.1 genauer erläutert.

Das System kann mit jedem zeitgemäß ausgestatteten Computer betrieben werden.

A.2 Datenhandschuh

Ein Datenhandschuh ermöglicht die Erfassung von räumlichen Bewegungen der Hand und ihres Öffnungsgrades. Dazu sind Sensoren in den Handschuh integriert, die die Beugung der Finger erfassen. Weitere Sensoren ermöglichen die Bestimmung der Position und die Orientierung der Hand im Raum.

In dieser Diplomarbeit wurde das Modell P5 der Firma Essential Reality verwendet, die sich heute Alliance Distributors Holding nennt [40, 41]. Der Datenhandschuh ist seit 2002 auf dem Markt und wurde für Computer- und Konsolenspiele entwickelt. Das System besteht aus einem Datenhandschuh und einem Empfangsgerät, die in Abbildung A.2 dargestellt sind.

Der Handschuh selbst ist offen und wird durch verstellbare Ringe am Fingerende und durch ein Gummiband an der Hand gehalten. Auf diese Weise passt er sich flexibel an

Abbildung A.2: *Datenhandschuh P5 mit Empfangseinheit (Quelle: [42, 43])*

verschiedene Handgrößen an. Allerdings kann der Handschuh nur an der rechten Hand getragen werden. Ein Modell für die linke Hand ist nicht erhältlich.

Die Beugung der Finger wird durch Dehnungsmessstreifen ermittelt, die an der Oberseite der Finger angebracht sind. Wird ein Finger geknickt, wird der zugehörige Messstreifen gedehnt und verändert dabei seinen elektrischen Widerstand. Der Widerstand liefert somit den Grad der Krümmung des Fingers. Allerdings sagt er nichts darüber aus, welches Glied des Fingers bewegt wurde. Die Auflösung der Messung beträgt 6 Bit.

Zur Bestimmung der Position und Orientierung der Hand im Raum sind auf der Oberseite des Handschuhs 8 Infrarot-Dioden angebracht. Deren Position wird 60 Mal pro Sekunde von mehreren Photodioden im Empfangsgerät aufgezeichnet. Auf diese Weise lässt sich die Position des Handschuhs im Raum auf ca. 3cm und seine Orientierung auf ca. 1° genau bestimmen, sofern sich der Handschuh nicht weiter als 90cm von der Empfangsstation entfernt befindet [40]. Ab einer Entfernung von 1,20m kann der Handschuh gar nicht mehr benutzt werden, da das Licht der Dioden dann für eine Ortung nicht mehr ausreicht. Dies wird durch eine rot blinkende Infrarot-Diode angezeigt. Zusätzlich eingeschränkt wird die Bewegungsfreiheit durch ein Kabel, dass den Datenhandschuh mit dem Empfangsgerät verbindet.

Die Empfangsstation überträgt die Position und Orientierung des Handschuhs über eine USB1.1-Schnittstelle an einen angeschlossenen Computer oder eine Spielekonsole. Über dieses Kabel wird auch der benötigte Strom bezogen. Eine zusätzliche Stromversorgung ist somit nicht nötig.

B Verwendete Software

Anhang

B.1 Arena

Arena ist die zur Motion Capture Anlage OptiTrack zugehörige Software zur Aufnahme von Ganzkörperbewegungen [44]. Die Software führt dabei durch alle nötigen Arbeitsschritte, von der Kalibrierung der Kameras, über die Aufnahme von Bewegungsdaten bis hin zur Nachbearbeitung und Fehlerkorrektur der Daten. Da die Aufnahme von Bewegungsdaten zeitverzögert gestartet werden kann, kann eine einzige Person die Software bedienen und sich selber aufnehmen. Während der Aufnahme der Daten kann am Bildschirm ein virtueller Charakter verfolgt werden, der live die vorgeführte Bewegung imitiert. Die Software, deren Oberfläche in Abbildung B.1 zu erkennen ist, kann bis zu zwei Personen gleichzeitig aufnehmen.

Die entstandenen Daten können gespeichert, in die Formate *bvh* oder *c3d* exportiert oder in Echtzeit über ein lokales Netz an andere Anwendungen versandt werden. Während Arena nur unter Windows läuft, können Programme zur Verarbeitung der Daten so auf anderen Computern unter anderen Betriebssystemen laufen. Zusätzlich wird auf diese Weise der aufnehmende Computer entlastet. NaturalPoint gibt eine mögliche Wiederholfrequenz von 100 Bildern/s für die Netzwerkübertragung an [45]. Da die Übertragung auf UDP und Multicast basiert, können mehrere Anwendungen gleichzeitig

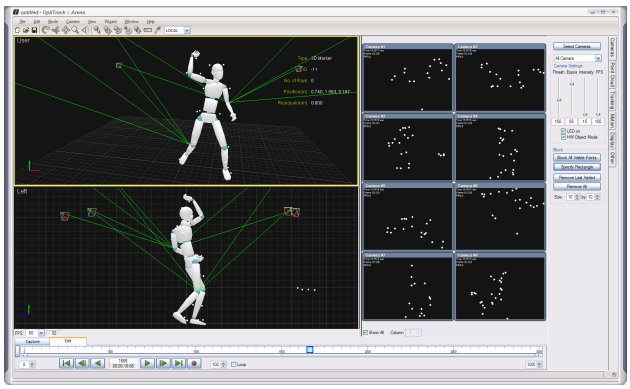


Abbildung B.1: *Grafische Oberfläche der Software Arena (Quelle: [44])*

mit den Daten arbeiten.

Um Fehler in den aufgenommenen Daten zu korrigieren, stellt Arena eine Reihe von Werkzeugen zur Nachbearbeitung bereit. Damit können unter anderem in der Trajektorie vorhandene Lücken durch spezielle Verfahren gefüllt werden.

Eine Voraussetzung zur Nutzung der Software sind mindestens 6 angeschlossene Kameras. Die Höchstzahl erlaubter Kameras beträgt 24.

B.2 Software Datenhandschuh

Die offizielle Download-Seite von Essential Realities ist nicht mehr erreichbar. Die Software für den P5 Datenhandschuh ist allerdings weiter von verschiedenen anderen Seiten im Netz erhältlich. Sie besteht aus einem Treiberpaket und einem Software Development Kit (SDK), die separat heruntergeladen werden können.

Treiber

Das Treiberpaket enthält neben dem Treiber für den Datenhandschuh das *P5 Control Panel*, mit dem die Einstellungen des Handschuhs angepasst werden können. Dieses ist über die Gruppe *P5 Glove*, die im Startmenü erstellt wird, oder über einen Eintrag in der Systemsteuerung erreichbar.

Zu den Konfigurationsmöglichkeiten gehört die Kalibrierung der Beugungssensoren der einzelnen Finger, Einstellung zur Nutzung des Datenhandschuhs als Mausersatz und die Möglichkeit, die Knöpfe des Datenhanschuhs zu überprüfen. Für das Positionierungsverhalten des Handschuhs im Raum sind keine Einstellungen vorgesehen. Im Gegensatz zur Motion Capture Anlage muss die optische Einheit in der Empfangsstation auch nicht kalibriert werden, um den Handschuh lokalisieren zu können.

Software Development Kit

Nach der Installation des Treibers kann der P5 Datenhandschuh bereits als Ersatz des Mauszeigers und in einer Reihe von Spielen verwendet werden. Um allerdings eigene Anwendungen entwickeln zu können, die mehr als eine 2D-Position auf dem Bildschirm verwenden, ist der direkte Zugriff auf die Sensorwerte des Handschuhs nötig. Dies wird durch das P5 Software Development Kit ermöglicht.

Das 40MB umfassende Paket enthält Beispielprogramme mit Quellcode, eine Bibliothek und eine C++ Header-Datei, um eigene Anwendungen daran zu binden, und eine Windows Hilfe-Datei in der die API für den Zugriff auf den Handschuh beschrieben ist.

Die dokumentierte API besteht aus der Klasse `CP5DLL`, die den Zugriff auf einen angeschlossenen Handschuh und die im P5 Control Panel vorhandenen Einstellungen ermöglicht, und den Modulen *P5 SDK Bend* und *P5 SDK Motion*. Diese beiden Module enthalten Routinen, die den Zugriff auf die Sensordaten des Handschuhs ermöglichen, d.h. auf die Werte der Beugungssensoren und die dreidimensionale Position und Orientierung des Handschuhs. Dabei können die Sensorwerte gefiltert oder ungefiltert abgefragt werden.

B.3 Player/Gazebo

Als Softwareplattform wurde in diese Diplomarbeit *Player* [46] verwendet. Die Software abstrahiert von den Eigenschaften verschiedener Roboter, deren Sensoren und Aktuatoren, und stellt eine einheitliche Schnittstelle für Robotikanwendungen zur Verfügung.

Dazu verwendet sie ein Client-Server-Modell. Der Player-Server ist durch Treiber in der Lage mit der Hardware zu kommunizieren, das heißt Aktuatoren anzusteuern und Sensoren auszulesen. Auf der anderen Seite ermöglicht er es Client-Anwendungen, über einen Satz vordefinierter Schnittstellen, auf diese Informationen zuzugreifen. Die Kommunikation erfolgt mittels TCP/IP. Auf diese Weise sind Client-Anwendungen sowohl unabhängig von der verwendeten Plattform und Programmiersprache, als auch von der konkret verwendeten Hardware, solange diese der erwarteten Schnittstelle entspricht.

Beispielsweise würde ein Algorithmus, der einen Roboter in einer zweidimensionalen Karte steuert, mit Player nicht mehr für einen konkreten Robotertyp implementiert werden. Stattdessen würde er seine Befehle an die `position2d`-Schnittstelle von Player schicken. Die Übersetzung dieser Befehle in Kommandos für einen speziellen Roboter übernimmt der Player-Treiber des entsprechenden Roboters. Der Algorithmus wäre so mit verschiedensten Robotertypen lauffähig.

Durch den modularen Aufbau von Player ist es ohne Weiteres möglich, das Programm um weitere Treiber für neue Roboter- oder Sensorarten zu erweitern. Da das Player-Projekt seine Software unter der GPL veröffentlicht, lässt sich auch die Software selber nach Belieben an die eigenen Bedürfnisse anpassen. So ist es zum Beispiel möglich, Player um weitere Schnittstellen zu ergänzen.

Ein Treiber ist dabei jedoch nicht auf die Kommunikation mit Hardware beschränkt. Im Rahmen des Player-Projekts wird unter anderem der physikalische Simulator *Gazebo* [47] entwickelt. Er stellt ebenfalls einen Player-Treiber bereit, der den Zugriff auf simulierte Objekte ermöglicht. Für Client-Anwendungen besteht somit kein Unterschied zwischen simulierten Objekten und realer Hardware. Dies erlaubt es, Algorithmen zunächst in einer Simulationsumgebung zu testen, ehe sie auf reale Hardware übertragen werden.

Gazebo

Gazebo ist eine 3D-Simulation für Robotikanwendungen, deren grafische Oberfläche für den in diese Diplomarbeit benutzen Roboter in Abbildung B.2 zu sehen ist. Sie erlaubt die Simulation von mehreren Robotern, Sensoren und Objekten in einer gemeinsamen Umgebung. Dabei werden Sensormessungen realistisch wiedergegeben, d.h. sie sind fehlerbehaftet. Auch die Interaktion zwischen Objekten wird basierend auf physikalischen Gesetzen realitätsgetreu nachgebildet. Dafür verwendet Gazebo die *Open Dynamics Engine*, eine freie Bibliothek, die Routinen zur Simulation der Dynamik von starren Körpern beinhaltet. Die 3D-Darstellung basiert auf *OGRE*, einer ebenfalls freien 3D Engine.

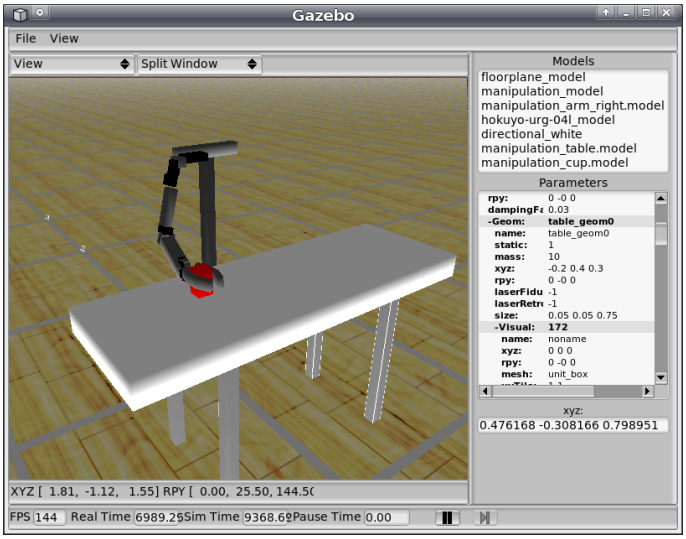


Abbildung B.2: *Grafische Benutzeroberfläche des Roboter-Simulators* Gazebo

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Servicerobotik: Definition und Potential. `http://www.wimi-care.de/pdfs/WiMi-Care-WB5-Servicerobotik-DefinitionundPotential.pdf`. Online, zuletzt besucht am 15.11.2009.

[2] Malte Kuß. *Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning*. PhD thesis, Technische Universität Darmstadt, March 2006.

[3] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[4] M. Opper and O. Winther. A Bayesian approach to on-line learning. 1999.

[5] Lehel Csató. *Gaussian Processes - Iterative Sparse Approximations*. PhD thesis, Aston University, March 2002.

[6] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, third edition, 2008.

[7] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999.

[8] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[9] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, January 1965.

[10] Satoru Hiwa, Tomoyuki Hiroyasu, and Mitsunori Miki. ISDL Report No. 20060806006. `http://mikilab.doshisha.ac.jp/dia/research/report/2006/0806/006/report20060806006.html`, 2006. Online, zuletzt besucht am 15.11.2009.

[11] Martin Riedmiller and Heinrich Braun. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP algorithm. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 586–591, San Francisco, CA, 1993.

[12] Martin Riedmiller. Rprop-description and implementation details. Technical report, University of Karlsruhe, 1994.

[13] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[14] Donald R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21(4):345–383, December 2001.

[15] Hansjörg Schmidt. Parallelisierung Ersatzmodell-gestützter Optimierungsverfahren. Master's thesis, Technische Universität Chemnitz, February 2009.

[16] Phillip Boyle. *Gaussian Processes for Regression and Optimisation.* PhD thesis, Victoria University of Wellington, 2007.

[17] Jörg Stückler, Kathrin Gräve, Jochen Kläß, Sebastian Muszynski, Michael Schreiber, Oliver Tischler, Ralf Waldukat, and Sven Behnke. Dynamaid: Towards a Personal Robot that Helps with Household Chores. In *RSS 2009 Workshop on Mobile Manipulation in Human Environments*, 2009.

[18] Sven Behnke, Jörg Stückler, and Michael Schreiber. NimbRo @Home 2009 Team Description, 2009.

[19] Project NimbRo - Learning Humanoid Robots. `http://www.ais.uni-bonn.de/nimbro/`. Online, zuletzt besucht am 15.11.2009.

[20] Nate Kohl and Peter Stone. Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2619–2624, May 2004.

[21] Daniel J. Lizotte, Tao Wang, Michael H. Bowling, and Dale Schuurmans. Automatic Gait Optimization with Gaussian Process Regression. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 944–949, 2007.

[22] Aude Billard, Sylvain Calinon, Rüdiger Dillmann, and Stefan Schaal. Robot Programming by Demonstration. In Bruno Siciliano and Oussama Khatib, editors, *Handbook of Robotics*, pages 1371–1394. Apringer, 2008.

[23] Bruno Dufay and Jean-Claude Latombe. An Approach to Automatic Robot Programming Based on Inductive Learning. *The International Journal of Robotics Research*, 3(4):3–20, 1984.

[24] Tomás Lozano-Pérez. Robot Programming. *Proceedings of the IEEE*, 71(7):821–841, July 1983.

[25] Michael Pardowitz, Steffen Knoop, Rüdiger Dillmann, and Raoul D. Zöllner. Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(2):322–332, 2007.

[26] Paul Bakker and Yasuo Kuniyoshi. Robot See, Robot Do : An Overview of Robot Imitation. In *AISB96 Workshop on Learning in Robots and Animals*, pages 3–11, 1996.

[27] Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, 1994.

[28] Sylvain Calinon, Florent Guenter, and Aude Billard. On Learning, Representing and Generalizing a Task in a Humanoid Robot. *IEEE transactions on systems, man and cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, 37(2):286–298, 2007.

[29] Sylvain Calinon and Aude Billard. Inremental Learning of Gestures by Imitation in a Humanoid Robot. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 255–262, New York, NY, USA, 2007. ACM.

[30] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and Generalization of Motor Skills by Learning from Demonstration. In *International Conference on Robotics and Automation (ICRA2009)*, 2009.

[31] Jacopo Aleotti and Stefano Caselli. Trajectory clustering and stochastic approximation for robot programming by demonstration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1029–1034, 2005.

[32] Stefan Schaal. Learning From Demonstration. *Advances in neural information processing systems*, 9:1040–1046, 1997.

[33] William D. Smart and Leslie Pack Kaelbing. Effective Reinforcement Learning for Mobile Robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, 4:3404–3410, 2002.

[34] Florent Guenter and Aude G. Billard. Using Reinforcement Learning to Adapt an Imitation Task. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1022–1027, 2007.

[35] Micha Hersch, Florent Guenter, Sylvain Calinon, , and Aude Billard. Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations. *IEEE Trans. on Robotics*, 24(6):1463–1467, 2008.

[36] Jan Peters and Stefan Schaal. Reinforcement Learning for Parameterized Motor Primitives. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 73–80. IEEE, 2006.

[37] NaturalPoint NatNet SDK. `http://media.naturalpoint.com/software/OptiTrack_files/NatNetSDK1.4.zip`. Version 1.4 vom 23.04.2008.

[38] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[39] OptiTrack Optical Motion Capture Solutions. `http://www.naturalpoint.com/optitrack/products/motion-capture/`. Online, zuletzt besucht am 15.11.2009.

[40] Dominik Heim and Volker Pilz. 3D-Navigation mit dem P5 Data Glove. Seminar Computeranimation & Visualisierung, Wintersemester 2004/2005, Fachhochschule Kaiserslautern, Fachbereich Informatik und Mikrosystemtechnik, Dezember 2004.

[41] Andrew Davison. *Pro Java 6 3D Game Development: Java 3D, JOGL, JInput and JOAL APIs*, chapter 14. Apress, April 2007.

[42] Examining the P5 Glove by Essential Reality. `http://www.mts.net/~kbagnall/p5/p5dissassembly.html`. Online, zuletzt besucht am 15.11.2009.

[43] P5 Datenhandschuh. `http://www.hwp.ru/articles/Virtualnaya_perchatka_P5_Glove_-_poshchupay_drugoy_mir/`. Online, zuletzt besucht am 15.11.2009.

[44] ARENA Motion Capture Software. `http://www.naturalpoint.com/optitrack/products/full-body-mocap.html`. Online, zuletzt besucht am 15.11.2009.

[45] NatualPoint Motion Capture FAQ. `http://www.naturalpoint.com/optitrack/support/motion-capture-faq.html`. Online, zuletzt besucht am 29.10.2009.

[46] Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, pages 317–323, 2003.

[47] Nathan Koenig and Andrew Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 3, pages 2149–2154, 2004.

Selbstständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe angefertigt habe und dabei keine anderen Hilfsmittel als die in der Arbeit angegebenen benutzt habe. Zitate, die wörtlich oder sinngemäß aus anderen Arbeiten übernommen wurden, habe ich als solche kenntlich gemacht.

Bonn, den 15.11.2009

_____

Kathrin Gräve