

Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelor-Forschungsprojekt

Peg-in-hole mit flexiblen Objekten

Nils Boike, Eugen Mozikov,
Moritz Staub, Matthias Tompert

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. rer. nat. habil. Miriam Mehl
Betreuer/in:	M.Sc. Benjamin Maier
Beginn am:	14. November 2018
Beendet am:	14. Mai 2019

Peg-in-hole mit flexiblen Objekten

Nils Boike, Eugen Mozikov, Moritz Staub, Matthias Tompert

Abstract— In dieser wissenschaftlichen Arbeit implementieren wir, aufbauend auf dem Werk von Bodenhagen et al.[1], eine Lernmethode für Peg-In-Hole-Aktionen mit flexiblen Objekten. Dabei wird die Bewegung eines Roboter-Greifarmes durch eine quadratische Bézier-Kurve beschrieben; das zu untersuchende flexible Objekt wird lediglich durch seine Deformationseigenschaften in waagrecht Lage spezifiziert. Wurde eine erfolgreiche Aktion gelernt, so lässt sich dieses Wissen auf ähnliche, bisher nicht untersuchte Objekte, übertragen. Dazu wird ein zwei-Phasen-Ansatz gewählt: Zunächst findet eine Lernphase offline statt, in welcher potentielle Peg-In-Hole-Aktionen zu bestimmten Peg-In-Hole-Problemen simuliert und die erfolgreichen Aktionen gespeichert werden. In der zweiten Phase wird online zu einem neuen Peg-in-Hole-Problem eine Lösung gesucht, indem die gespeicherten, erfolgreich gelösten Probleme mit dem aktuellen Problem verglichen werden. Gespeichert werden neben dem Feature-Vektor des flexiblen Objekts die Kontrollpunkte der Bézier-Kurve, die in der Offline-Simulation keine Kollision mit dem Spalt ergaben. Anstelle eines naiven Ansatzes zur Auswahl eines erfolgversprechenden Kontrollpunkts (i.e. einer Iteration über den kompletten Datensatz mit anschließender Auswertung mithilfe eines Kerneldichteschätzers) verwenden wir einen effizienteren Ansatz auf Basis von B-Spline-basierten Dünngittern zur Optimierung. Auch wurde der Vorschlag von Bodenhagen et al., die Startposition des Greifarmes zu verbessern, von uns aufgegriffen. Für die Optimierung greifen wir auf die General Sparse Grid Toolbox SG++ [5] zurück.

Index Terms—Peg-In-Hole, Flexible Objects, Computational Geometry, Action Learning, Robotics, Geometric Model.

1 EINLEITUNG

Objekte durch schmale Löcher zu bewegen, fällt Menschen nicht sonderlich schwer. Es gibt zahlreiche Steckwürfel, Puzzles und Kordelspielzeuge zum Stecken, Stapeln und Schnüren für Vorschulkin-der von 3 bis 5 Jahren, mit Hilfe derer früh die nötigen motorischen Fähigkeiten erlernt werden können.

Diese für den Menschen scheinbar einfache Aufgabe, ist für Maschinen aber auch heute noch deutlich anspruchsvoller und, trotz des stetigen Fortschrittes in den Bereichen Machine Learning und Artificial Intelligence, immer noch ein aktuelles Problem, welches es zu lösen gilt. Noch schwieriger wird das Problem, wenn die Objekte nicht starr, sondern biegsam und in ihrer Form, Konsistenz und Länge variabel sind.

Die praktische Lösung eines solchen Problems ist besonders für die heutige Industrie interessant: der Einsatz von menschlicher Handarbeit kostet sehr viel Zeit und damit auch Geld - beides Ressourcen, die heutzutage in einer globalisierten Wertschöpfungskette bereits äußerst knapp bemessen sind. Eine gleichbleibend hohe Qualität ist ebenfalls schwer einzuhalten, nicht umsonst wird oft vom "menschlichen Faktor" gesprochen.

Da sich nicht jeder komplizierte Prozess, von einem Roboter automatisiert durchführen lässt, ist menschliche Handarbeit derzeit noch an vielen Stellen unvermeidlich. Dies kann einen ganzen Industriezweig wie die Kabelstrangzulieferer (beispielsweise der Automobilindustrie, welche in jedem Fahrzeug kilometerlange Kabelbäume verlegen muss), vor gewaltige Herausforderungen stellen.

Diese Schwierigkeit lässt sich aber auch in vermeintlich banalen, alltäglichen Situationen wiedererkennen: So ist beispielsweise ein USB-Type-C-Kabelsatz zwischen 4x und 10x teurer als ein USB 2.0 Kabelsatz, was hauptsächlich auf die großen Zeitkosten menschlicher Handarbeit zurückzuführen ist [3]. Dieser Kostenfaktor trägt massiv dazu bei, die breite Einführung des neuen Standards in der Konsumentenelektronik zu bremsen.

Gerade deshalb bildet das automatisierte Lösen von Peg-In-Hole-Problemen mit biegsamen Objekten eine große Problemstellung, auf welche noch viel zu selten das Augenmerk gelegt worden ist - erst in einem späteren Werk von den Originalautoren Bodenhagen, Fugl et al.[2] wird von einer "Nähe zu industriellen Bedingungen"

gesprochen.

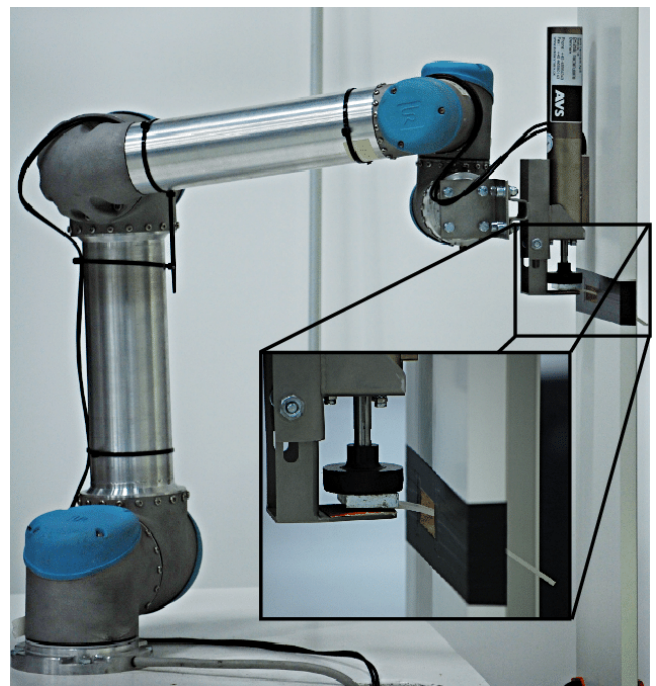


Abbildung 1: Ausführung einer Peg-In-Hole-Aktion von Bodenhagen et al. 2012

In dieser Arbeit untersuchen wir näher, wie ein Roboterarm effizient und in Echtzeit, einen Bewegungspfad seines Greifers anhand der Deformationsdaten eines bisher unbekannten flexiblen Objektes, bestimmen kann, um diesen durch eine definierte Öffnung kollisionsfrei hindurchzuführen. Dazu modellieren wir Objekte verschiedener Länge, Breite, Dicke und Dichte, welche wir zur Deformationsberechnung als Kragträger-Balken (engl: cantilever beam) betrachten. Daraufhin simulieren wir eine Peg-In-Hole-Aktion mit diesem Objekt und einer möglichen Bewegungsbahn des Robotergreifers als rationale Bézier-Kurve. Erfolgreich verlaufene Simulationen werden in einer Datenbank gesammelt, um später auf dieses gewonnene

Nils Boike	Eugen Mozikov
st151897@stud.uni-stuttgart.de	st114858@stud.uni-stuttgart.de
Matthias Tompert	Moritz Staub
st149346@stud.uni-stuttgart.de	st148535@stud.uni-stuttgart.de

Wissen zurückgreifen zu können. Eine Peg-in-Hole-Aktion wird als erfolgreich angesehen, falls während der Durchführung das flexible Objekt die Ränder der Öffnung nicht berührt.

Dank der Abstraktion der Objekte auf ihre Deformationseigenschaften ist es uns möglich, Parameter erfolgreicher Aktionen auf bisher nicht betrachtete, aber ähnliche Objekte effizient zu übertragen, ohne große Vorkenntnisse über das zu betrachtende Objekt zu benötigen. Durch diesen Ansatz wird ein Großteil des Lernaufwandes offline, i.e. nicht in Echtzeit, abgewickelt, was für ein schnell agierendes System während des tatsächlichen Einsatzes sorgt.

Einen tieferen Einblick in die Modellierung der Objekte, der Greifarmbewegung, sowie unserer Methoden, gewährt Kapitel 2. Unsere Resultate und ihre Auswertung finden sich dann anschließend im 3. Kapitel.

2 SETUP UND METHODIK

Im Folgenden beschreiben wir im Detail die Modellierung der untersuchten Objekte (Kapitel 2.1), ihrer Deformation (Kapitel 2.2) sowie die Erzeugung einer möglichen Trajektorie des Roboter-Greifarms (Kapitel 2.3). Modellierung der Kollision und Kollisionserkennung folgen anschließend in Kapitel 2.4.

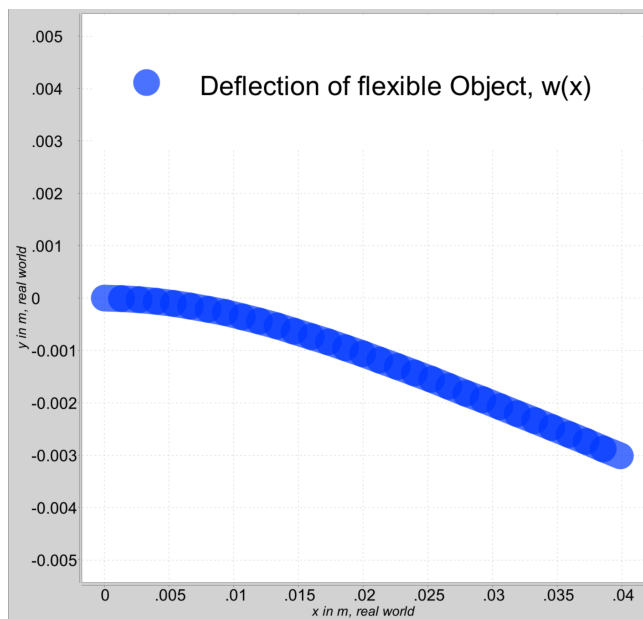


Abbildung 2: Deformation eines flexiblen Objektes als waagrecht Bernoulli-Euler-Kragträger.

2.1 Objektbeschreibung

Damit ein reales Objekt effizient numerisch untersucht werden kann, gilt es zunächst, es auf seine relevanten Eigenschaften zu reduzieren, um ein geeignetes Modell dafür zu finden. Im Gegensatz zu dem gewöhnlichen Peg-In-Hole-Problem ist für biegsame Objekte eine viel größere Palette an Eigenschaften zu berücksichtigen. So sind nicht nur die Form und Maße des Objektes für den Ausgang einer Peg-In-Hole-Aktion von Bedeutung - Material, Gewicht und Elastizität spielen eine zunehmend zentrale Rolle.

Eine wichtige Maßeinheit ist das sog. Elastizitätsmoment (engl. Young's Modulus) – eine numerische Konstante, welche den Widerstand eines Feststoffes gegenüber Verformung beschreibt, wenn sich dieser Stoff in einem eindimensionalen Belastungsszenario (i.e. Dehnung / Streckung) befindet. Es wird angenommen, dass das zu untersuchende Objekt aus einem einheitlichen Material besteht und deshalb das Elastizitätsmoment an jedem Punkt des Objektes gleich ist.

	Länge (mm)	Breite (mm)	Dicke (mm)	Dichte (g/cm ³)
Objekt 1	40	15	2	1.15
Objekt 2	60	15	2	1.15
Objekt 3	80	15	2	1.15

Tabelle 1: Untersuchte Objekte

Eine weitere relevante geometrische Größe ist das Flächenträgheitsmoment, welches den Biege- und Widerstand eines Querschnitts beschreibt und in der Festigkeitslehre zur Berechnung von (mechanischer) Spannung sowie Verformung herangezogen wird. Für die von uns untersuchten Objekte nehmen wir eine gleichbleibende Objektdicke sowie Objektbreite an, was in einem gleichmäßigen, rechteckigen Querschnitt über das gesamte Objekt resultiert. In diesem vereinfachten Fall lässt sich das Flächenträgheitsmoment durch

$$I = \frac{b \cdot h^3}{12} = A \cdot \frac{h^2}{12}$$

bestimmen, wobei b und h die Breite und Höhe/Dicke des Objektes beschreiben und A die Querschnittsfläche des Objektes ist.

Weiterhin wird unser Objekt durch seine Abmessungen (Länge, Breite, Höhe) sowie seine Dichte charakterisiert. Dichte sowie Elastizitätsmoment sind als dem Werkstoff inhärente Kenngrößen nicht direkt messbar, sondern müssen anderweitig bestimmt werden. Es wird vorausgesetzt, dass diese Daten für die zu untersuchenden Objekte zur Verfügung stehen.

Es wird der Einfachheit halber ein rechteckiger Querschnitt für das betrachtete Objekt angenommen. Ein abweichender (e.g. rund) oder asymmetrischer Querschnitt kann ebenfalls modelliert werden, indem ein geeigneter rechteckiger Hüllkörper gewählt wird, solange die Steifigkeit und die Querschnittsfläche des Objektes auf seiner kompletten Länge konstant bleibt.

Auf die Modellierung des Greifmechanismus wurde verzichtet, da dieser keinen Einfluss auf die relevanten Parameter einer Peg-In-Hole-Aktion ausübt. Wird ein Objekt gut am Greifarm befestigt, so ist nur der frei hängende Teil des Objektes für uns von Bedeutung.

Während das durch den Roboterarm gegriffene Objekt sich dynamisch verformen kann, wird das Loch als unbeweglich, rechteckig und die Lochränder als unendlich dünn angenommen, denn für eine Peg-in-Hole-Aktion ist lediglich der Abstand beider Lochränder sowie der Minimalabstand zwischen den beiden Lochrändern und dem Objekt interessant.

2.2 Deformation

Bei der Modellierung der Deformation des Objektes wird größtenteils auf Erkenntnissen von Bodenhagen et al.[1] aufgebaut: So wird für die Berechnung der Durchbiegung das Objekt als Kragträger¹ betrachtet; für die Verformung unter Last wird die Bernoulli-Euler-Balkentheorie herangezogen. Dabei werden folgende Größen für die Berechnung der Durchbiegung benötigt:

- das *Elastizitätsmoment* E
- das *Flächenträgheitsmoment* I , welches die Verformung bzw. Biegung eines Objektes beschreibt
- die *Gleichstreckenlast* q , welche gleichmäßig auf die gesamte Objektfläche wirkt. In unserem Fall ist es primär die Erdanziehungskraft und diese wird für die gesamte Länge, sowie alle Rotationen, des Objektes als konstant angenommen
- L als die Länge des Objektes

¹ Kragträger: ein im Sinne der techn. Mechanik einseitig fest eingespannter Balken, auf dessen gesamter Länge eine quer wirkende Kraft ausgeübt wird

- x beschreibt einen Punkt auf dem Objekt

Im Allgemeinen beschreibt die Euler-Bernoulli-Gleichung den Zusammenhang zwischen der Durchbiegung eines Trägers und dessen Belastung:

$$\frac{d^2}{dx^2} \left(EI \frac{d^2 w}{dx^2} \right) = q$$

wobei $w(x)$ die Ablenkung des Trägers in z -Richtung an einem Punkt x darstellt.

Im konkreten Fall eines gleichmäßig belasteten Kragträgers gilt statische Bestimmtheit, d.h. die Gleichlast lässt sich durch

$$(EI(x) \cdot w''(x))'' = q(x)$$

bestimmen, was eine gewöhnliche Differentialgleichung vierten Grades darstellt. Um eine diskrete Lösung in Abhängigkeit von der Objektlänge für die Durchbiegung $w(x)$ zu bestimmen, benötigen wir vier Randbedingungen:

- $w'''(x=L) = 0$, d.h. es wirkt kein Biegemoment auf das frei schwebende Ende, sowie
- $w''(x=L) = 0$, d.h. auf das frei bewegliche Ende wirkt keine Scherkraft
- für das im Greifer befindliche Ende gilt: $w(x=0) = 0$, d.h. dass das Objekt sich an dem festgehaltenen Ende nicht durchbiegt

Wie von Bodenhausen et al.[1] vorgeschlagen, korrigieren wir die Startposition der Peg-In-Hole-Aktion und verfeinern an dieser Stelle bisher das verwendete Modell: als vierte Randbedingung fordern wir:

$$w'(x) = \tan(\alpha)$$

womit die Steigung an dem im Greifer befindlichen Ende nicht - wie ursprünglich angenommen - null, sondern $\tan(\alpha)$ beträgt, wobei α den Winkel des Greifarmes in Initialstellung darstellt (ein Winkel von 0° ist dabei als waagrecht zu betrachten). Dies berücksichtigt, dass das Objekt zu Beginn der Peg-In-Hole-Aktion sich nicht wie ein klassischer Kragträger verhält (d.h. starr, waagrecht vor dem Loch eingespannt ist), sondern sich bereits vor dem Beginn des Experimentes durch die einwirkende Gravitation durchbiegt. Herleitung und die resultierende analytische Lösung finden sich in Abbildung 3.

Während das neue Modell eine notwendige Verbesserung gegenüber der von Bodenhausen et al. verwendeten klassischen Gleichung der Bernoulli-Euler-Balkentheorie darstellt, findet die Letztere bei uns trotzdem Verwendung: um möglichst viele verschiedene Objekte effizient zu beschreiben, verwenden wir Ablenkungswerte einer längennormalisierte Durchbiegung eines waagrecht eingespannten Objektes (vgl. Abbildung 2), welche an bestimmten Punkten equidistant gemessen wurden. Die dadurch gewonnenen Messwerte werden in einem sogenannten *Feature-Vektor* gespeichert, dessen Dimensionalität von der Auflösung der Durchbiegungsmessung abhängt.

Mit diesem Feature-Vektor ist es uns möglich, ein Objekt von seinen Maßen zu abstrahieren und lediglich anhand seiner Durchbiegungseigenschaften zu identifizieren. Dank diesem Verfahren können Objekte verschiedener Längen, welche sich jedoch unter Einfluss der Erdanziehungskraft gleich verhalten, als ähnliche Objekte angesehen werden. Damit lassen sich bekannte Greifartrajektorien ähnlicher Objekte auf das neu untersuchte Objekt übertragen.

Während Bodenhausen et al.[1] NURBS²-Flächen zur Beschreibung der Objekte und zur Berechnung der Feature-Vektoren heranzieht, beschränken wir uns auf eine zweidimensionale Welt Darstellung und verwenden deshalb lediglich die Durchbiegung eines

²NURBS, Kurzform für *Nicht-Uniforme Rationale B-Splines*, ist ein mathematisches Modell, welches häufig in Computergrafik oder Computer-Aided Design (CAD) zur Repräsentation von Kurven oder Oberflächen verwendet wird. Das Modell zeichnet sich durch eine effiziente Verarbeitung durch Computerprogramme bei gleichzeitig leichter Verständlichkeit und Interaktionsmöglichkeiten für Menschen aus.

$$w''''(x) = \frac{q}{EI}$$

(a) Ablenkung des Objektes

$$W''''(x) = \frac{q}{EI} + c_1$$

$$w'''(L) = 0 \rightarrow \frac{q}{EI} + c_1 = 0$$

$$\Rightarrow c_1 = -\frac{qL}{EI}$$

$$\hookrightarrow w''''(x) = \frac{qx}{EI} - \frac{qL}{EI}$$

(b) Nach erster Randbedingung

$$W''(x) = \frac{qx^2}{2EI} - \frac{qLx}{EI} + c_2$$

$$w''(x) = 0 \rightarrow \frac{qx^2}{2EI} - \frac{qLx}{EI} + c_2 = -\frac{qL^2}{2EI} + c_2$$

$$\Rightarrow c_2 = \frac{qL^2}{2EI}$$

$$\hookrightarrow w''(x) = \frac{qx^2}{2EI} - \frac{qLx}{EI} + \frac{qL^2}{2EI}$$

(c) Nach zweiter Randbedingung

$$W'(x) = \frac{qx^3}{6EI} - \frac{qLx^2}{2EI} + \frac{qL^2x}{2EI} + c_3$$

$$w'(x) = \tan(\alpha) \rightarrow \frac{q0^3}{6EI} - \frac{qL0^2}{2EI} + \frac{qL^20}{2EI} + c_3$$

$$\Rightarrow c_3 = \tan(\alpha)$$

$$\hookrightarrow w'(x) = \frac{qx^3}{6EI} - \frac{qLx^2}{2EI} + \frac{qL^2x}{2EI} + \tan(\alpha)$$

(d) Nach dritter Randbedingung

$$W(x) = \frac{qx^4}{24EI} - \frac{qLx^3}{6EI} + \frac{qL^2x^2}{2 * 2EI} + \tan(\alpha)x + c_4$$

$$w(0) = 0 \rightarrow c_4 = 0$$

$$\begin{aligned} \hookrightarrow w(x) &= \frac{qx^4}{24EI} - \frac{qLx^3}{6EI} + \frac{qL^2x^2}{4EI} + \tan(\alpha)x \\ &= \frac{qx^4 - 4qLx^3 + 6qL^2x^2}{24EI} + \tan(\alpha)x \\ &= \frac{qx^2 * (x^2 - 4Lx + 6L^2)}{24EI} + \tan(\alpha)x \end{aligned}$$

(e) Diskrete Lösung nach letzter Randbedingung

Abbildung 3: Durchbiegung des flexiblen Objektes in Abhängigkeit von Gleichlast, Flächenträgheits- sowie Elastizitätsmoment und Länge. Die Gleichlast wird hierbei als konstant angenommen.

waagrecht fixierten Objektes zur Berechnung des Feature-Vektors. Dies geht einher mit reduzierter Komplexität für Modellierung und Deformationsberechnung, ohne Einfluss auf das Ergebnis einer Peg-In-Hole-Aktion zu nehmen, denn auch wenn Bodenhausen et al. die Objekte als NURBS-Flächen modelliert haben, war eine Lateralbewegung des Robotergreifers nicht Teil des Modells. Triviale Fälle, in welchen das Objekt breiter ist als das Loch und offensichtlich kein kollisionsfreies Einführen möglich ist, liegen nicht im Fokus dieser Arbeit und wurden daher nicht berücksichtigt.

2.3 Greifertrajektorie

Um die Komplexität der Peg-in-Hole-Aktion gering zu halten, verzichten wir, wie bereits erwähnt, auf die Modellierung des Objektes als dreidimensionale Fläche und beschränken uns auf einen zweidimensionalen Raum. Die Bewegung des Roboter-Greifarms wird dabei durch einen dreidimensionalen Pfad approximiert, wobei die ersten zwei Dimensionen die absolute Position des Greifarms im einem kartesischen Koordinatensystem angeben. Die dritte Dimension entspricht der Rotation des Greifarms relativ zur Waagerechten. Die Kontrollpunkte befinden sich damit in $\mathbb{R}^2 \times SO(2)$.³ Dies entspricht dem Modell, welches von Bodenhausen et al.[1] verwendet wurde.

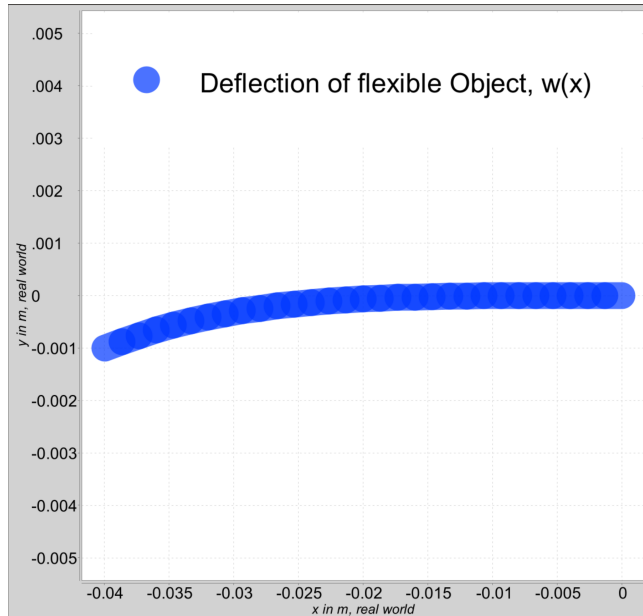


Abbildung 4: Deformation eines flexiblen Objektes in Initialstellung vor Experimentdurchführung. Man beachte die waagrechte Ausrichtung des frei schwebenden Endes und die damit verbundene Korrektur für die bereits in Initialstellung vorhandene Durchbiegung.

Der Pfad ist als eine rationale Bézier-Kurve zweiten Grades modelliert (vgl. Parabel) und wird somit durch drei Kontrollpunkte definiert, wobei zwei dieser drei Kontrollpunkte fest verankert sind:

- Der Endpunkt $P(1)$ dieser Bézier-Kurve ist genau in der Mitte des Spaltes positioniert. Der Greifer hat in diesem Punkt eine Rotation von 0° .
- Der Startpunkt $P(0)$ wird online mithilfe der Durchbiegung (siehe Abbildung 3) errechnet; die durch die Durchbiegung resultierende Stauchung des Objektes muss dabei kompensiert werden. Der Winkel des Greifers wird in diesem Punkt dabei so gewählt, dass das frei bewegliche Ende des Objektes horizontal, exakt vor der Spaltmitte, positioniert ist.

Damit liegt der Freiheitsgrad für die Wahl einer zu einem gegebenen flexiblen Objekt passenden Greifertrajektorie im dritten Kontrollpunkt cp , welcher sich zwischen den Punkten P_0 und P_1 befindet. Mit diesen drei Kontrollpunkten lässt sich eine rationale Bézier-Kurve zweiten Grades darstellen durch:

$$P(t) = P_0 + B(t)(P_1 - P_0)$$

für $t \in [0; 1]$, wobei

³ $SO(2)$ bezeichnet die *spezielle orthogonale Gruppe* der 2×2 -Matrizen, d.h. echte Drehungen in zwei Dimensionen, innerhalb einer Ebene.

$$B(t) = \frac{\sum_{i=0}^2 b_{i,2}(t)P_i w_i}{\sum_{i=0}^2 b_{i,2}(t)w_i}$$

Dabei ist $b_{i,2}$ das Bernsteinpolynom zweiten Grades und P_i der i -te Kontrollpunkt der Kurve:

$$P_i \in \{0, cp, 1\}$$

Als Gewichte w_i werden dabei $w = [1, 2, 1]$ gewählt. Dies sorgt für einen verstärkten Einfluss des zwischen P_0 und P_1 liegenden Kontrollpunktes cp auf die resultierende Trajektorie. Bodenhausen et al. weisen an dieser Stelle darauf hin, dass die Kontrollpunkte unabhängig von der Bewegungsgeschwindigkeit des Greifers und insbesondere unabhängig von dem flexiblen Objekt sind. Ein Kontrollpunkt wird somit immer zu einer, mehr oder minder sinnvollen, Trajektorie führen - ein erfolgreicher Verlauf der Peg-in-Hole-Aktion wird dadurch aber nicht garantiert.[1]

Tabelle 2: Simulationsdaten zu Objekt 1:

Spaltgröße [mm]:	6.00	5.00	4.00	3.00
Max. Abstand [mm]:	1.89	1.39	0.89	0.39
Mitt. Abstand [mm]:	1.06	0.81	0.42	0.02
Simulierte CPs	8000	8000	8000	8000
Davon Erfolge	652	496	411	138
Davon Koll.	7348	7504	7589	7862

Auswertung der Erfolgreichen Simulationen:

Optimierung				
berech. Abst. [mm]:	1.61	1.08	0.59	0.36
best simulated Clearance in%	85%	78%	66%	91%

Um möglichst sinnvolle Ergebnisse zu erhalten, den Aufwand zu begrenzen sowie die Trefferrate in den späteren Simulationsschritten zu erhöhen, haben wir die Domäne gültiger Kontrollpunkte cp eingeschränkt:

- der Kontrollpunkt sollte nicht weiter von dem Schlitz entfernt sein als P_0 - dies führt zwar nicht notwendigerweise zu einer Kollision, verschlechtert aber in vielen Fällen das Ergebnis, denn P_0 wird bereits so gewählt, dass das freie Ende des Objektes sich in der optimalen Position befindet
- der Kontrollpunkt sollte nicht über P_1 hinausgehen - dies führt unweigerlich zu einer Kollision
- der Kontrollpunkt sollte über der Durchbiegung im Initialzustand liegen - ein Kontrollpunkt unter dieser Kurve würde ein Absacken des Greifarms während der Bewegung führen. Dies ist zwar nicht notwendigerweise mit einer Kollision verbunden, verringert jedoch die Chancen eines erfolgreichen Ausganges der Peg-in-Hole-Aktion.
- der Winkel sollte nicht größer als der Startwinkel des Greifers in Initialstellung und nicht kleiner als 0° sein. Ein Winkel größer als der Startwinkel würde zwangsläufig eine Positionierung des Objektes im Schlitz zur Folge haben, bei welcher der für die Kollision relevante Abschnitt des Objektes nicht länger horizontal im Schlitz ausgerichtet ist. Dies würde einen negativen Einfluss auf den resultierenden Mindestabstand mit sich bringen und ist damit nicht gewünscht. Dies ist gleichzeitig bedingt durch den vorhergehenden Punkt, denn ein Winkel

größer als in Initialstellung benötigt einen Kontrollpunkt unterhalb der Durchbiegungskurve, sonst führt er unweigerlich zu einer Kollision mit dem oberen Teil des Schlitzes. Ein Winkel kleiner null würde eine ähnliche Problematik mit dem unteren Teil des Schlitzes nach sich ziehen.

2.4 Kollisionserkennung und Abstand

Zur effizienten Erkennung von Kollisionen und der Berechnung des Minimalabstandes zwischen Objekt und Lochrand machen wir uns Methoden aus der Computergrafik zunutze, denn Kollisionen von Objekten ist ein wichtiger Bestandteil moderner dreidimensionaler Grafikframeworks⁴ Dazu werden sowohl die Lochränder als auch das flexible Objekt als Hüllkörper (engl. bounding volume, vgl. bounding box) modelliert. Dies erlaubt uns von bereits etablierter und performanter Kollisionserkennung Gebrauch zu machen.

Einen ähnlichen Ansatz verfolgen Bodenhausen et al., indem sie mithilfe von RobWork⁵[4] das flexible Objekt als ein “rigid device“, bestehend aus mehreren, aneinander hängenden, rechteckigen Boxen mit definierten Gelenken sowie zugehörigen Winkeln, darstellen.

Während die Hüllkörper der Lochränder in jedem Schritt der Simulation beibehalten werden, verzichten wir, abweichend von Bodenhausen et al., auf die Modellierung des kompletten flexiblen Objektes durch mehrere Boxen. Stattdessen wird zu jedem Simulationszeitpunkt nur der für die Kollisionserkennung relevante Teil des flexiblen Objektes durch *einen* Hüllkörper approximiert. Dies sorgt dafür, dass zu jedem Zeitpunkt während der Simulation höchstens drei Hüllkörper auf Kollision und Abstand hin untersucht werden, was die Komplexität der Simulation gering hält und die Durchführungsgeschwindigkeit der einzelnen Simulationsdurchläufe erheblich beschleunigt.

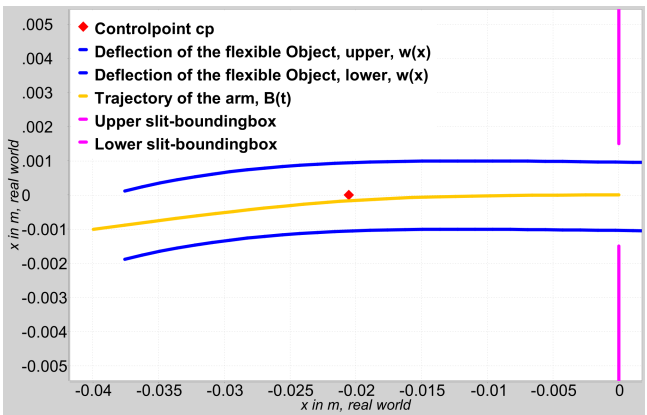


Abbildung 5: Visualisierung eines Simulationsdurchlaufes einer Peg-in-Hole-Aktion. Zu sehen sind die obere und untere Kante des flexiblen Objektes (blau), die durch den Kontrollpunkt *cp* (rot) sowie *P0* (linkes Ende der Kurve) und *P1* (rechtes Ende der Kurve) festgelegte Trajektorie des Roboter-Greifarmes (orange). Der Schlitz wird durch zwei dünne Hüllkörper (magenta) dargestellt.

2.5 Dünne Gitter

2.5.1 Offline Simulation

Da das System kein Vorwissen über mögliche Peg-in-Hole-Aktionen mit sich bringt, wenden wir ein Zwei-Schritt-Verfahren an: Im ersten Schritt wird ein Datensatz potentieller Kontrollpunkte generiert, indem die Domäne der möglichen Kontrollpunkte *cp* equidistant abgetastet wird. Der Suchbereich wird, wie eingangs in Kapitel 2.3

⁴Framework: deutsch: Rahmenstruktur; beschreibt in der Softwaretechnik ein Programmiergerüst, welches einen Rahmen sowie Entwurfsmuster zur Anwendungserstellung zur Verfügung stellt.

⁵ein open-source Robotik-Framework speziell für den Einsatz in Industrie und Bildung; bietet unter Anderem effiziente Kollisionserkennung und Mindestabstandsberechnung an.

beschrieben, eingeschränkt. Für jeden so erhaltenen Kontrollpunkt wird mithilfe eines bestimmten Feature-Vektors ein Simulationsdurchlauf einer Peg-in-Hole-Aktion durchgeführt. Positiv verlaufene Simulationen, d.h. Aktionen, die zu keiner Kollision geführt und einen Mindestabstand größer null produziert haben, tragen zusammen mit dem zugehörigen Kontrollpunkt sowie dem jeweiligen Feature-Vektor zu einer Dichte bei, welche mithilfe des Dünngitter-Dichteschätzers der SGPP-Bibliothek ausgewertet wird.

Die Verwendung einer Simulation als Vorschritt hilft, die Wahl eines geeigneten Parameters für eine Peg-in-Hole-Aktion von der Echtzeitanforderung zu entkoppeln, indem eine viel größere Anzahl an Aktionen im Voraus simuliert werden kann, als wenn beide Mechanismen eng verknüpft wären.

2.5.2 Online Parameterwahl

In dem letzten Schritt des Zwei-Schritt-Lernverfahrens wird auf Basis des Dünngitter-Dichteschätzers ein B-Spline-basiertes dünnes Gitter für den Optimierungsschritt generiert. Die resultierende Dünngitterfunktion wird daraufhin mithilfe des Gradientenverfahrens minimiert. Dabei haben wir die Zielfunktion angepasst, sodass das Optimum der Zielfunktion gleichzusetzen ist mit dem Maximalabstand der Peg-in-Hole-Aktion. Als Startpunkt des Gradientenverfahrens wählen wir einen Punkt, der sich nahe an der Mitte des Objektes befindet - dies hat in vielen Simulationsdurchläufen die Präzision der Peg-in-Hole-Aktion an kritischen (i.e. kollisionsnahen) Stellen verbessert.

Tabelle 3: Simulationsdaten zu Objekt 2:

Spaltgröße [mm]:	9.00	8.00	7.00	6.00
Max. Abstand [mm]:	3.08	2.58	2.09	1.60
Mitt. Abstand [mm]:	1.29	1.05	0.85	0.66
Simulierte CPs	8000	8000	8000	8000
Davon Erfolge	426	330	227	133
Davon Koll.	7574	7670	7773	7867
Auswertung der Erfolgreichen Simulationen:				
Optimierung				
berech. Abst. [mm]:	2.51	2.43	2.26	1.77
best simulated Clearance in %	82%	94%	108%	110%

3 AUSWERTUNG UND RESULTATE

3.1 Erkenntnisse

Wie in den Tabellen in Kapiteln 2.3 und 2.5 zusehen ist, haben wir für die 2 Simulationsobjekte Erfolge erzielen können. Im Gegensatz zu dem Aufbau von Bodenhausen [1] haben wir verschiedene Spaltgrößen verwendet, um deren Auswirkung auf die CPs der jeweiligen Objekte abgleichen zu können. Für das Objekt 1, mit 8000 simulierten CPs, kommt man, je nach Spaltgröße (siehe Tabelle zwischen 3mm und 6mm) auf 138 bis 652 erfolgreich gewählte, und damit gespeicherte, CPs. Bei Objekt 2 sind es hingegen deutlich weniger erfolgreich simulierte CPs (bei gleicher Anzahl an simulierten CPs wie bei Objekt 1), obwohl Objekt 2 nur 20mm länger ist als Objekt 1 und es größere Spalten gibt (von 5mm bis 8mm). Die Anzahl der erfolgreichen CPs reicht von 133 (bei 6mm Spaltgröße) zu 426 (bei 9mm). Das Objekt 2 hat bei 50% mehr Länge und gleicher Spaltgröße nur 20,4% (bei 6mm Spaltgröße) der Anzahl erfolgreicher CPs von Objekt 1. Nach Auswertung dieser erfolgreichen Simulationen gibt es ein Optimierungspotenzial des Abstandes zwischen Objekt und Spaltwand, jedoch nur bei Objekt 2 mit 8% bei 7mm und 10% bei 6mm Spaltgröße. Die anderen simulierten CPs scheinen bereits optimal zu sein.

Damit decken sich unsere Beobachtungen mit den Ergebnissen von Bodenhagen et al.[1]. Darin schlussfolgern sie, dass mit zunehmender Länge der Objekte auch die Schwierigkeit steigt, diese kollisionsfrei einzuführen: wir konnten in unseren Experimenten sowohl die höhere Erfolgswahrscheinlichkeit als auch den höheren Mindestabstand bei kürzeren Objekten reproduzieren.

3.2 Schwierigkeiten

Um Zukünftige arbeiten zu vereinfachen müsste die Einbindung der General Sparse Grid Toolbox SG++ in Java verbessert werden, da diese nur in C++ funktioniert. Auch ist der Werteraum zu überarbeiten da sich sämtliche Punkte nur zwischen $[0, 1]$ befinden und somit sämtliche Werte in den positiven Raum skaliert werden müssen.

4 ZUKÜNFTIGE ARBEIT

- Durch eine bessere Wahl des Modells zur Bestimmung der Durchbiegung flexibler Objekte kann die Genauigkeit der Durchbiegung und damit der Experimente verbessert werden, denn Bernoulli-Euler-Balkentheorie gilt nur für extrem kleine Belastungen und geringe Rotationen.
- Verwendung von höherdimensionalen Bézier-Kurven zur Beschreibung des Greiferpfades, um Präzision an kritischen Stellen zu erhöhen
- Parametrierung der verwendeten Dünngitter-Bibliothek SG++ verbessern - beispielsweise wäre die Verwendung von B-Spline-basierten Gittern innerhalb des Dünngitter-Dichteschätzers wünschenswert.

5 DISKUSSION

In dieser Arbeit haben wir das von Bodenhagen et al. vorgeschlagene System zur Durchführung von Peg-in-Hole-Aktionen im Kontext flexibler Objekte implementiert, welches die Objekte auf ihre physikalischen Elastizitätseigenschaften abstrahiert. Durch die Abstraktion von Objekten auf ihre Feature-Vektoren wird dem System ermöglicht, auf Basis bereits bekannter Objekte, Lern- sowie Optimierungsmethoden zur Auswahl passender Parameter für bisher nicht näher betrachtete Objekte anzubieten. Wir haben das System an mehreren Stellen verfeinert, um einerseits die Güte der Modellierung zu erhöhen und andererseits die Echtzeitanforderungen an das System, mit der Verwendung von Dünngittern, zu verbessern.

ACKNOWLEDGMENTS

Diese Arbeit wurde von Benjamin Maier betreut.

LITERATUR

- [1] L. Bodenhagen, A. R. Fugl, M. Willatzen, H. G. Petersen, and N. Krüger. Learning peg-in-hole actions with flexible objects. In *ICAART (I)*, pages 624–631. Citeseer, 2012.
- [2] L. Bodenhagen, A. R. Fugl, A. Jordt, M. Willatzen, K. A. Andersen, M. M. Olsen, R. Koch, H. Petersen, and N. Krüger. An adaptable robot vision system performing manipulation actions with flexible objects. *Automation Science and Engineering, IEEE Transactions on*, 11:749–765, 07 2014.
- [3] S. Burke. Why USB 3.1 Type-C isn't on more cases & cable factory tour in Dongguan, China. <https://www.gamersnexus.net/guides/3453-why-usb-type-c-isnt-on-more-cases/>, 25 März 2019. [Online, Zugriff am 29-April-2019].
- [4] L. Ellekilde and J. A. Jorgensen. Robwork: A flexible toolbox for robotics research and education. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 1–7, June 2010.
- [5] D. Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Verlag Dr. Hut, München, Aug. 2010.