

# Automatisches finden von Fügepositionen State of the Art

Özgür Gümüşlü

Sommersemester 2021

## Contents

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Peg in Hole Strategien</b>	<b>2</b>
2.1	Passive Nachgiebigkeitsregler . . . . .	2
2.2	Aktive Nachgiebigkeitsregler . . . . .	2
2.2.1	Impedanz-Regler . . . . .	2
2.2.2	Hybrider Kraft/Lage-Regler . . . . .	3
2.3	Regelungsmethoden mit Hilfsttechnologien . . . . .	3
2.4	Optimierung der Regelungsmethoden . . . . .	3
2.5	Entscheidung . . . . .	3
<b>3</b>	<b>Suchstrategie</b>	<b>4</b>
3.1	Stochastisches Suchverfahren . . . . .	4
3.2	Spiralsuche . . . . .	5
3.3	Rastersuche . . . . .	6
3.4	Winkelbasiertes Suchverfahren . . . . .	9
<b>4</b>	<b>Literaturliste</b>	<b>10</b>

# 1 Einleitung

In diesem Dokument geht es darum einen State of the Art für das sogenannte *"Peg in Hole strategies"* zu erstellen. Grundsätzlich werden die Methoden wie folgt kategorisiert:

- Passive Nachgibiegsregler
- Aktive Nachgibiegsregler
- Regelungsmethoden mit Hilfstechologien
- Optimierung der Regelmethode

im weiteren wird eine kurze Übersicht der einzelnen Methoden vorgestellt.

## 2 Peg in Hole Strategien

### 2.1 Passive Nachgibiegsregler

Hier gibt der Endeffektor bei einem Kontakt mit der Umwelt nach. Durch die passive Nachgibiegskeit des Endeffektors ist der Roboter nicht in der Lage sich der Umwelt zu adaptieren. Mit Methoden aus der künstlichen Intelligenz kann dieses Verfahren verbessert werden. Mit Hilfe der KI ist der Roboter nun in der Lage sich seiner Umgebung anzupassen.

### 2.2 Aktive Nachgibiegsregler

Dieses hat ein sehr großes Einsatzspektrum. Hier werden Kräfte und Drehmoment in den Achsen am Endeffektor gemessen und an die Regelung weitergeleitet um die gewünschte Trajektorie des Endeffektors zu erzeugen. Diese lassen sich wie folgt unterteilen.

#### 2.2.1 Impedanz-Regler

Hier wird die Strecke zwischen dem TCP und Sollposition als Feder-Dämpfer-Masse-System simuliert  $F = C * \Delta x$  hiermit kann die Kraft für den Pressvorgang mittels gegebener Federsteifigkeit  $C$  ermittelt werden. Für genauere Informationen siehe Grobkonzept.

### **2.2.2 Hybrider Kraft/Lage-Regler**

Bei dieser Methode werden anhand der Randbedingungen von der Umwelt Selektionsmatrizen erstellt. Mit Hilfe dieser Selektionsmatrizen können nun Lage und Kraft parallel geregelt werden. Für genauere Informationen siehe Grobkonzept.

## **2.3 Regelungsmethoden mit Hilfstechnologien**

Für erhöhung der Flexibilität der Montageaufgaben können Hilfstechnologien integriert werden. Mit dem Einsatz der Hilfstechnologien können die Anwendungsbereiche der oben erwähnten Steuerungsmethoden ausgedehnt werden. Zu den Hilfsmethoden können folgende Systeme herangezogen werden:

- Haptische Systeme
- Vision Systeme
- Traktorienplaner
- Kinetische Führung

In dieser Arbeit kommen keine Hilfstechnologien zum Einsatz und werden nur zur Vollständigkeitshalber kurz erwähnt.

## **2.4 Optimierung der Regelungsmethoden**

Hier werden die Fähigkeiten des Roboters anhand von Machine Learning erweitert werden. Hier werden zwischen supervised und unsupervised learning unterscheiden. Learning by Demonstration(LbD) ist die am meisten typische Form des supervised learnings. LbD wird mittlerweile auch für Industrielle Roboter verwendet. Der Roboter sind mit Hilfe von Machine Learning Algorithmen in der Lage anhand von Demonstrationen zu lernen und müssen nicht mehr für jede einzelne Aktion programmiert werden.

## **2.5 Entscheidung**

Wie schon im Grobkonzept erwähnt, wurde die Entscheidung für den Impedanz-Regler gefällt. Einerseits weil dieser Regler schon im Roboter integriert ist, zusätzlich braucht es für hochpräzise Aufgaben keine genauen Koordinaten

sondern kann mittels Suchstrategie zum Erfolg führen und ist dadurch bei Abweichung relativ zuverlässig.

### 3 Suchstrategie

In diesem Teil wird ein größeres Augenmerk den Prozessen des Einfügens vom Werkstück gelegt. Hier gibt es mehrere Verfahren: Stochastische-Suche, Spiral-Suche, Raster-Suche, Kipp-Verfahren, Verzahnung etc. In dieser Arbeit werden auf die folgenden Verfahren näher eingegangen.

- Stochastisches Suchverfahren
- Spiralsuche
- Rastersuche
- Winkelbasiertes Suchverfahren

#### 3.1 Stochastisches Suchverfahren

Dieses entspricht der primitivsten Suchmethode, hier wird das Füge teil zufällig auf der x/y -Ebene bewegt bis der Einfü gepunkt gefunden wird. Algorithmisch kann dies folgendermaßen beschrieben, werden sollte die Füge position unbekannt sein:

$$x_{\Delta,c} = x_{\Delta,c-1} + (k_x * rand())$$

$$y_{\Delta,c} = y_{\Delta,c-1} + (k_y * rand())$$

*rand()* liefert zufällige Werte zwischen [-1,1] und  $k_x$  und  $k_y$  entsprechen Suchkoeffizienten für die x-/y Achsen.

Sollte die Einfüge position mit einer gewissen Abweichung bekannt sein reduziert sich kann der Offset weggelassen werden und der Algorithmus reduziert sich somit auf:

$$x_{\Delta,c} = k_x * rand()$$

$$y_{\Delta,c} = k_y * rand()$$

Sollte die Initialposition nahe der Füge position sein ist eine Gauß-Verteilung für die *rand()* Funktion vorzuziehen. Diese Suchmethode ist nicht deterministisch und gibst somit keine Garantie für einen Erfolg.

### 3.2 Spiralsuche

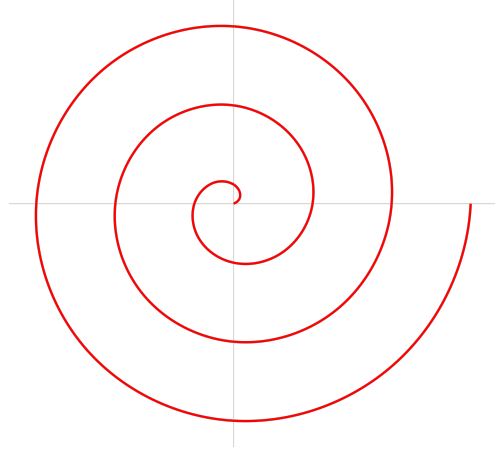


Figure 1: Archimedische Spiralsuche[2]

Dieses Suchverfahren startet parallel zur xy-Ebene im Ursprung (ungefähre Fügeposition) und führt eine Kreisbewegung und erhöht währenddessen den Radius des Kreises. Hierzu gibt es folgende Parameter:

$p_0$  entspricht dem Startpunkt. Also hier dem Ursprung der Spirale.  
 $r$  entspricht dem Radius des Suchmusters welches erhöht wird  $t$  die Turns Anzahl der Umdrehungen. Schnittpunkt der Spirale mit der x-Achse.  
 $s$  entspricht der Suchgeschwindigkeit,  $c$  dem Iterationszähler und  $f$  der Aktualisierungsfrequenz.  $\theta_{mx}$  die totale Winkel Suchlänge ist als

$$\theta_{mx} = t * 360[grad]$$

definiert. Die maximale Änderung des Radiuses pro Umdrehung  $\Delta r$  ist mit

$$\Delta r = \frac{r}{\theta_{mx}}$$

definiert. die Änderung des Winkels in [Grad]

$$\Delta\theta = \frac{\min(s, 360[grad])}{f}$$

um Aussagen treffen zu können wo sich das Füge teil gerade befindet wird eine Laufvariable hinzugezogen  $c_{local}$  und wird mit

$$c_{local} = c \bmod \left( \frac{\theta_{mx}}{\Delta\theta} \right)$$

berechnet. Mit dem  $c_{local}$  lässt sich nun die Position innerhalb der Spirale anhand des Änderung des Winkels.

$$\theta' = c_{local} * \Delta\theta$$

Mit diesem kann zurück auf den aktuellen Radius zurückgeschlossen werden.

$$r_{cur} = \theta' * \Delta r$$

Mit Hilfe der von  $\theta'$  und  $r_{cur}$  lässt sich nun der Offset des Werkstücks von der X-/Y Achse bestimmen:

$$x_{\Delta,c} = r_{cur} * \cos \theta'$$

$$y_{\Delta,c} = r_{cur} * \sin \theta'$$

Anmerkung: Bei zu großen Abständen der einzelnen Spiralen kann es vorkommen, dass die Fügeposition nicht bestimmt werden kann und somit würde das zum Fehlschlagen des Fügeprozesses führen.

### 3.3 Rastersuche

[2] Diese Strategie eignet sich für Suchen bei denen die Fügeposition unbekannt ist.

Wie in Figure 2 abgebildet beginnt die Suche bei  $p_0$  und der Greifer bewegt sich mit dem Fügeteil die Strecke  $w$  entlang der y-Achse an dem Punkt angekommen fährt der Greifer nun  $l$  entlang der x-Achse. Sollte die Fügeposition noch nicht gefunden sein fährt der Greifer wieder die Strecke  $w$  entlang der negativen y-Achse. Diesen Vorgehen wird mit  $s$  der Suchgeschwindigkeit  $n$  mal für die Anzahl der Rasterscans wiederholt. Gegeben sind wieder  $p_0$  die Suchlaufvariable  $c$  und die Frequenzaktualisierung  $f$ . Somit lässt sich  $l_{step}$  folgendermaßen bestimmen:

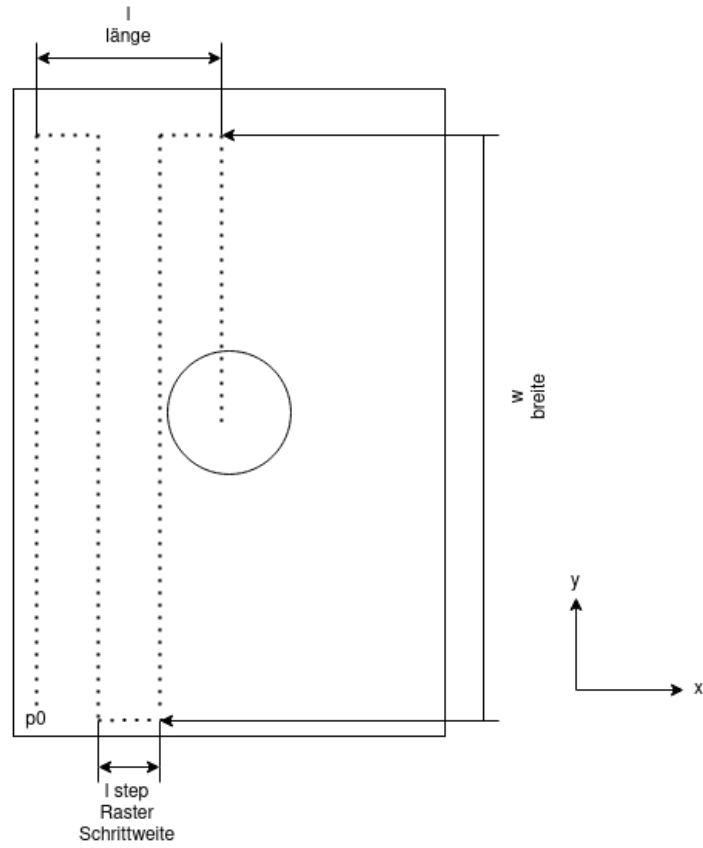


Figure 2: Rastersuche[2]

- Für gerade  $n$  gilt  $l_{step} = \frac{1}{n-1}$
- anderenfalls gilt  $l_{step} = \frac{1}{n}$

die totale Länge  $l_{total}$  lässt sich folgendermaßen bestimmen:

- Für gerade  $n$  gilt  $l_{total} = n * w + l$
- anderenfalls gilt  $l_{total} = (n + 1) * w + l$

der zurückgelegte Weg per iteration  $\Delta l$  beträgt

$$\Delta l = \frac{s}{f}$$

Der prozentuale Anteil jeder Breiten-Schritt Kombination welche aus dem Breitenkomponente,  $q_{raster}$  besteht wird berechnet als

$$q_{raster} = \frac{w}{w + l_{step}}$$

Um vorhersehbares Verhalten gewährleisten zu können wird ein lokaler Suchiterator zur Hilfe gezogen um bestimmen zu können wo auf dem Rastermuster das Werkstück liegen soll. Dieser  $c_{local}$  wird folgend berechnet:

- Für  $(c \bmod \frac{2l_{total}}{\Delta l}) > (\frac{2l_{total}}{\Delta l})$  gilt  $c_{local} = \frac{2l_{total}}{\Delta l} - (c \bmod \frac{2l_{total}}{\Delta l})$
- anderenfalls  $c_{total} = c \bmod \frac{2l_{total}}{\Delta l}$

$c_{local}$  wird verwendet um die Position auf Raster basierend auf  $p0$  zu bestimmen. Um festzustellen ob sich das Werkstück auf den Kanten der X-Achse des Rechtecks befindet oder sich entlang der Y-Achse befindet, wird das Verhältnis des Suchweges zum Suchweg eines Breiten- und Längenschrittes eines Rasters berechnet

$$q_{new} = \frac{c_{local} * \Delta l}{w + l_{step}}$$

Mit diesem Wert lässt sich nun die Richtung evaluieren

$$h = \lfloor q_{new} \rfloor$$

und

$$d = \lceil q_{new} \rceil$$

Diese Werte bestimmen dann die X- und Y-Achsenversätze des Werkstückes gemäß

- Wenn  $d > q_{raster}$  gilt  $x_{\Delta,c} = h + \frac{d - q_{raster}}{1 - q_{raster}} * l_{step}$
- anderenfalls  $x_{\Delta,c} = c \bmod \frac{2l_{total}}{\Delta l}$
- Wenn  $(c \bmod \frac{2l_{total}}{\Delta l}) > (\frac{2l_{total}}{\Delta l})$  gilt  $y_{\Delta,c} = \frac{2l_{total}}{\Delta l} - (c \bmod \frac{2l_{total}}{\Delta l})$
- anderenfalls  $y_{\Delta,c} = c \bmod \frac{2l_{total}}{\Delta l}$

Anmerkung: Auch hier ähnlich wie bei der Rastersuche kann wenn man die Rasterschrittweite zu groß wählt der Fügeprozess fehlschlagen. Sollte aber die Rasterschrittweite gut gewählt werden verspricht auch diese Methode ähnlich wie die Spiralsuche relativ viel Erfolg.



### 3.4 Winkelbasiertes Suchverfahren

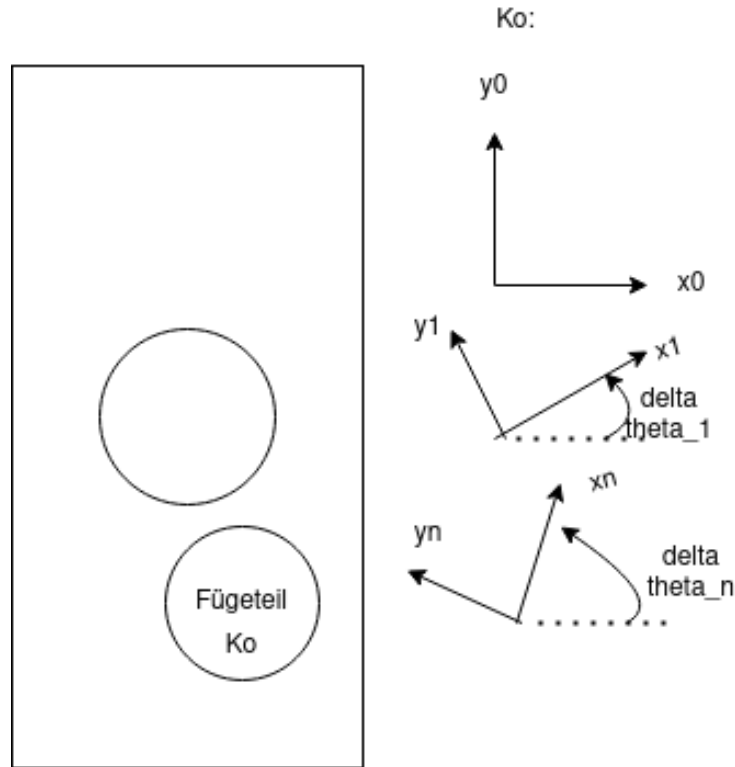


Figure 3: Winkelbasiertes Suchverfahren

Hier ist die Idee das Fügeteil auf der x/y-Ebene nur in +/- y-Richtung um  $\Delta y$  fahren zu lassen sollte dies nicht zum Ziel führen, wird der Winkel um  $\Delta\theta$  verändert und der Vorgang wiederholt. Dies wird solange gemacht bis  $\Delta\theta$  einen Winkel von  $\pi$  erreicht hat. Diese Variante kann viel Zeit in Anspruch nehmen je nach Auflösung von  $\Delta\theta$ . Jedoch hängt der Erfolg von diesem Winkel ab. Grundsätzlich gilt je feiner der Winkel desto höher die Wahrscheinlichkeit, dass diese Methode zum Erfolg führt.

## 4 Literaturliste

- Xianmin Zhang <sup>1,2</sup> , Yanglong Zheng <sup>1,2</sup> , Jun Ota <sup>3</sup> and Yanjiang Huang, Peg-in-Hole Assembly Based on Two-Phase Scheme and F/T Sensor for Dual-Arm Robot, 2017
- Xiaoqing Li, Rui Li, Hong Qiao, Chao Ma, and Liang li, Human-Inspired Compliant Strategy for Peg-in-Hole Assembly Using Environmental Constraint and Coarse Force Information, (o.D)
- JEREMY A. MARVEL, ROGER BOSTELMAN, and JOE FALCO, Multi-Robot Assembly Strategies and Metrics, 2018 [2]
- Jingang Jiang , Zhiyuan Huang , Zhuming Bi , Xuefeng Ma , Guang Yu, State-of-the-Art control strategies for robotic PiH assembly, 2020 [3]