Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Research Report
RR-09-01

Verfahren zur Vermeidung von Kollisionen gesteuert beweglicher Teile einer Anlage

Udo Frese und Holger Täubig

April 2009

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-67608 Kaiserslautern
Tel.: + 49 (631) 205 75-0
Fax: + 49 (631) 205 75-503

Stuhlsatzenhausweg 3
D-66123 Saarbrücken
Tel.: + 49 (681) 302-5151
Fax: + 49 (681) 302-5341

Robert-Hooke-Str. 5
D-28359 Bremen, Germany
Tel.: +49 (421) 218-64 100
Fax: +49 (421) 218-64 150

E-Mail: info@dfki.de WWW: http://www.dfki.de

Deutsches Forschungszentrum für Künstliche Intelligenz

DFKI GmbH

German Research Center for Artificial Intelligence

Founded in 1988, DFKI today is one of the largest nonprofit contract research institutes in the field of innovative software technology based on Artificial Intelligence (AI) methods. DFKI is focusing on the complete cycle of innovation — from world-class basic research and technology development through leading-edge demonstrators and prototypes to product functions and commercialization.

Based in Kaiserslautern, Saarbrücken and Bremen, the German Research Center for Artificial Intelligence ranks among the important “Centers of Excellence” worldwide.

An important element of DFKI’s mission is to move innovathons as quickly as possible from the lab into the marketplace. Only by maintaining research projects at the forefront of science can DFKI have the strength to meet its technology transfer goals.

The key directors of DFKI are Prof. Wolfgang Wahlster (CEO) and Dr. Walter Olthoff (CFO).

DFKI’s research departments are directed by internationally recognized research scientists:

- Image Understanding and Pattern Recognition (Director: Prof. T. Breuel)
- Knowledge Management (Director: Prof. A. Dengel)
- Deduction and Multiagent Systems (Director: Prof. J. Siekmann)
- Language Technology (Director: Prof. H. Uszkoreit)
- Intelligent User Interfaces (Prof. Dr. Dr. h.c. mult. W. Wahlster)
- Institute for Information Systems at DFKI (Prof. Dr. P. Loos)
- Robotics (Prof. F. Kirchner)
- Safe and Secure Cognitive Systems (Prof. B. Krieg-Brückner)

and the associated Center for Human Machine Interaction (Prof. Dr.-Ing. Detlef Zühlke)

In this series, DFKI publishes research reports, technical memos, documents (eg. workshop proceedings), and final project reports. The aim is to make new results, ideas, and software available as quickly as possible.

Prof. Wolfgang Wahlster
Director

Verfahren zur Vermeidung von Kollisionen gesteuert beweglicher Teile einer Anlage

Udo Frese und Holger Täubig

DFKI-RR-09-01

Diese Arbeit wurde durch das Bundesministerium für Bildung und Forschung unter Förderkennzeichen 01 IM F02 A (Projekt SAMS) gefördert.

ISSN 0946-008X

Verfahren zur Vermeidung von Kollisionen gesteuert beweglicher Teile einer Anlage

Udo Frese und Holger Täubig
Deutsches Forschungszentrum für Künstliche Intelligenz
Sichere Kognitive Systeme
28359 Bremen, Enrique-Schmidt-Straße 5
+49/421/218/64207
Udo.Frese@dfki.de
http://www.dfki.de/sks

2. April 2009

Zusammenfassung

Dieser technische Bericht beschreibt das *Verfahren zur Vermeidung von Kollisionen gesteuert beweglicher Teile einer Anlage*, wie es durch das Deutsche Forschungszentrum für künstliche Intelligenz beim Deutschen Patentamt unter 102009006256.4-32 zum Patent [1] angemeldet wurde.

Das Verfahren dient der zentralen Übewachung bewegter Anlagenteile (u.a. Roboterarme und Fahrzeuge) zum Schutz vor Kollisionen mit einander oder mit der Umwelt. Es ist entsprechend mit und ohne die zusätzliche sensorielle Erfassung von Hindernissen einsetzbar.

Wesentlicher Bestandteil des Algorithmus ist die Darstellung des von einem bewegten Anlagenteil während seiner Bewegung überstrichenen Volumens in Form der konvexen Hülle einer endlichen Punktmenge zzgl. eines Pufferradius. Diese Darstellung ist robust und in Echtzeit ohne numerische Probleme berechenbar. Die Darstellung wird für alle involvierten Körper berechnet und anschließend dazu verwendet, alle relevanten Paare von Körpern oder die sensoriell erfassten Bereiche der Umwelt auf Kollisionen zu testen. Dazu wird ein neuer und echtzeitfähiger Algorithmus angegeben.

Alle Berechnungen werden mittels konservativer Abschätzungen durchgeführt, so dass ein mathematisch beweisbar sicheres Gesamtsystem entsteht. Es werden einige solcher konservativer Abschätzungen für unterschiedliche Teile einer bewegten Anlage angegeben, die es dem Anwender des Verfahrens ermöglichen, bei der Beschreibung der Anlagengeometrie und -kinematik zwischen einer möglichst exakten Modellierung und einer einfacher und schneller berechnenbaren zu variieren. Für Fahrzeugtrajektorien (z.B. Bremsbewegungen) wird außerdem ein Konfigurationsraum verwendet, der es erlaubt Kurven- und Kreisfahrten einheitlich und ohne Singularitäten zu modellieren.

1 Zielsetzung

Das Verfahren dient dazu, gesteuerte bewegte Teile einer Anlage, zu überwachen und rechtzeitig einen Halt auszulösen, bevor diese miteinander oder mit der unbeweglichen Umgebung kollidieren. Bewegliche Teile in diesem Sinne sind insbesonders, aber nicht ausschließlich, a) Roboterarme, Achsentische, Portalkräne, oder allgemein jede zyklenfreie Hintereinanderschaltung von rotatorischen oder linearen Gelenken und ausserdem b) Fahrzeuge und Fahrzeuge auf denen solche Hintereinanderschaltung von Gelenken montiert sind. Ein Beispiel wir in Abb. 1 gegeben.
Das Verfahren setzt voraus, dass die Geometrie der Anlage, also der beweglichen Teile, Werkstücke und Umgebung vorab bekannt ist, alle beweglichen Teile mit Positionsgebern (bzw. Winkelgeber) versehen sind und für die Bremswege dieser beweglichen Teile Formeln bezogen auf die Geschwindigkeit vorhanden sind.
Das Hauptverfahren (erfinderischer Beitrag I-III) arbeitet ausschliesslich auf der vorkonfigurierten Geometrie und den gemessenen Positionen und verwendet keine sensorielle Erfassung von Hindernissen. Eine Erweiterung des Verfahrens dient zusätzlich der sensoriellen Absicherung von Personen vor Kollision mit Teilen der Anlage, insbesondere mit Roboterarmen und Fahrzeugen (erfinderischer Beitrag IV).

2 Vorgehensweise Grundverfahren (Erfinderischer Beitrag I)

Dieser Abschnitt beschreibt zuerst den Hauptteil des Verfahrens ohne Berücksichtigung von Fahrzeugen und ohne Berücksichtigung sensoriell erfasster Personen als Hindernisse.

2.1 Überblick

- Vor Inbetriebnahme des Systems wird die Geometrie der Anlage einkonfiguriert. Sie wird dargestellt, als eine Menge starrer, konvexer Körper, die ggf. durch bewegliche Gelenke verbunden sind.

Schritte a)-d) werden in jedem Takt der Robotersteuerung durchgeführt.

a) Aus den Messwerten der Gelenkpositionsgeber und deren Ableitung wird der Bremsweg im jeweiligen Gelenk als Interval von Gelenkpositionen bestimmt (das sogenannte Bremsintervall, für Drehgelenke ein Winkelinterval, für lineare Gelenke ein Streckeninterval). Unsicherheiten in der Gelenksensorik können hier aufgeschlagen werden.

b) Aus den Bremsintervallen aller Gelenke werden die Bremszonen, d.h. die von den verschiedenen Körpern beim Bremsen überstrichenen 3D Volumina berechnet ("swept volume"). Dazu wird jeder Körper schrittweise von dem mitbewegten Koordinatensystem an dem er angebracht ist, über die verschiedenen Gelenke ins Weltkoordinatensystem übertragen. Bei jedem dieser Schritte wird das Bremsintervall des jeweiligen Gelenkes mit einberechnet.

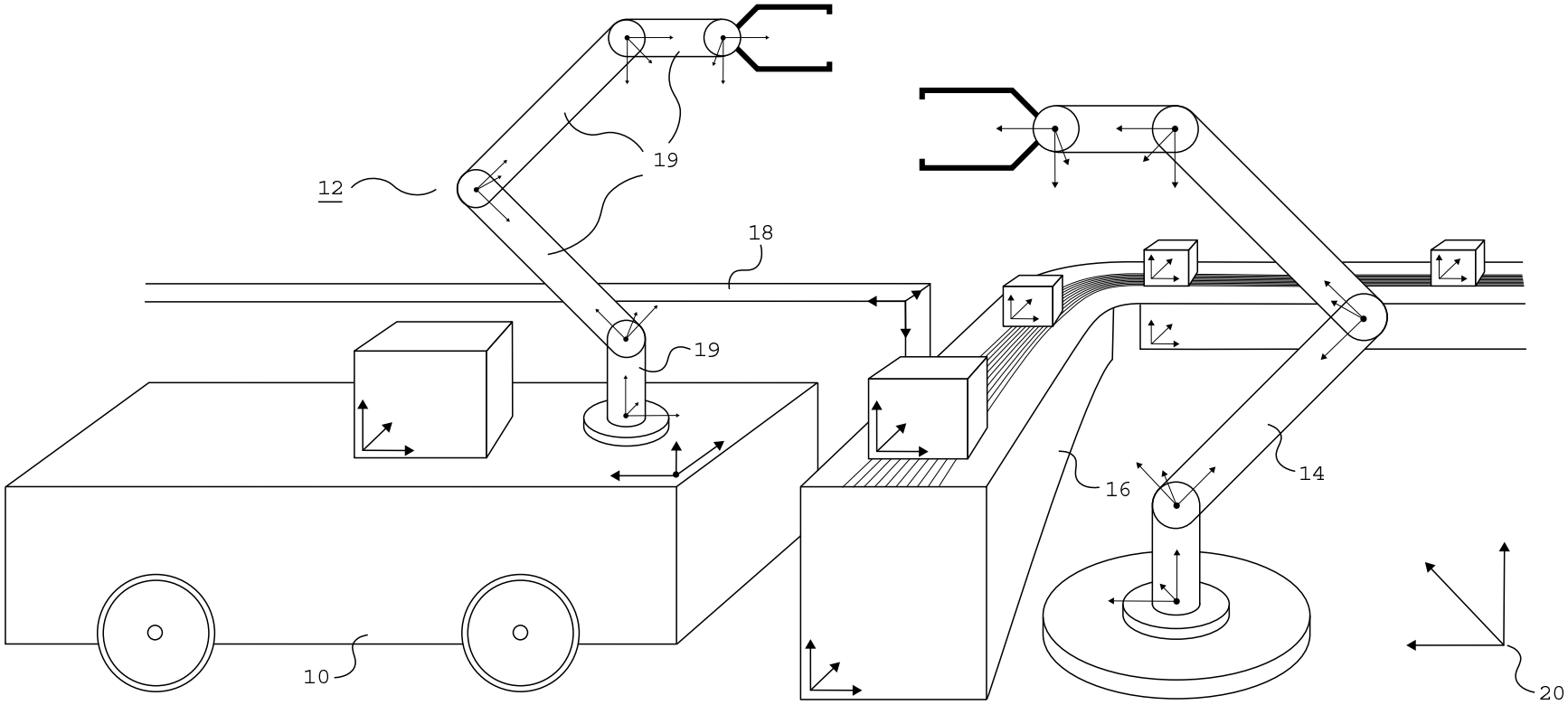


Abbildung 1: Anwendungsbeispiel für das diskutierte Verfahren. Gezeigt ist eine Anlage, bestehend aus den Anlagenteilen Fahrzeug (10), Roboterarm auf dem Fahrzeug(12), stationärer Roboterarm (14), Förderband (16) und einer Trennwand (18). Die Anlage besteht aus starren Körpern (von denen lediglich einige beispielhaft mit den Bezugszahlen 19; 22 gekennzeichnet sind) an denen jeweils ein 3D-Koordinatensystem festgelegt ist (Dreiergruppen von Pfeilen in der Abbildung). Die Körper sind jeweils durch Mechanismen zur gesteuerten Bewegung, (in der Abbildung jeweils in den Ursprüngen der Koordinatensysteme) verbunden. In diesem Beispiel sind solche Mechanismen, die Drehgelenke der Roboterarme, das Fahrzeug und das Förderband. Für den unbeweglichen Teil der Anlage ist ein Weltkoordinatensystem (20) definiert.

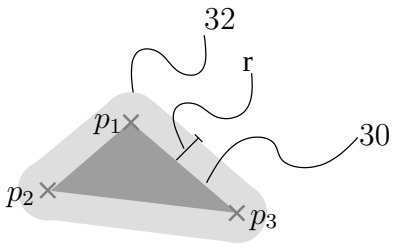


Abbildung 2: Ein Beispiel für die Darstellung einer Menge als $V(r, (p_i)_{i=1}^n)$ nach (1). $V(r, (p_i)_{i=1}^n)$ ergibt sich als die konvexe Hülle (30) von $n = 3$ Punkten p_1, p_2, p_3 die in alle Richtungen um den Radius r vergrößert wird (32).

c) Für alle (relevanten) Paare von Körpern wird die Distanz zwischen ihren Bremszonen bestimmt (bekannter GJK Algorithmus [2, 3]). Massgeblich für die Distanz zweier Körper ist der erste gemeinsame Knoten im kinematischen Baum auf dem Weg zur Wurzel.

d) Ist eine der bestimmten Distanzen 0, wird die Anlage gestoppt.

Erfinderischer Beitrag I ist dabei die schnelle Berechnung der Bremszonen in b) in einer Darstellung, die hinterher in c) noch schnelle Abstandsberechnung ermöglicht.

2.2 Sicherheitsabstände, Messunsicherheiten und Reaktionszeiten

In einer Anwendung sind meistens bestimmte Sicherheitsabstände vorgeschrieben und Messunsicherheiten, sowie Reaktionszeiten müssen berücksichtigt werden, damit eine Kollision vermieden werden kann. Diese werden in den Algorithmus wie folgt eingebracht: Die Hälfte des globalen Sicherheitsabstandes wird zu dem Pufferradius (s.u.) aller geometrischen Objekte der Szene addiert. Messunsicherheiten werden auf die Bremsintervalle aufgeschlagen. Reaktionszeiten werden werden bei den Bremsintervallen mit einbezogen, d.h. mit der Geschwindigkeit malgenommen addiert.
Im Endeffekt heisst dies, dass der Algorithmus den Roboter/die Anlage noch rechtzeitig stoppt, wenn er in dem ersten Zyklus eine Bremsung einleitet, in dem sich zwei Bremszonen berühren (mathematisch schneiden). Es geht also im Algorithmus darum zu bestimmen, *ob* zwei Bremszonen kollidieren.

2.3 Darstellung von 3D Volumina, insbesondere Bremszonen

Schlüsselidee ist die Darstellung aller im Algorithmus vorkommenden 3D Volumina (Körper, Bremszonen, Zwischenergebnisse) als konvexe Hülle einer endlichen Menge von Punkten zzgl. eines Pufferradius.

$$V\left(r; (p_i)_{i=1}^n\right) = \left\{ p_r + \sum_{i=1}^{n} \lambda_i p_i \middle| \lambda_i \geq 0 \forall i, \quad \sum_{i=1}^{n} \lambda_i = 1, \quad |p_r| \leq r \right\} \tag{1}$$

$$r > 0, \quad p_i \in \mathbb{R}^3 \forall i \tag{2}$$

Das heisst, im Rechner wird ein 3D Volumen repräsentiert als ein array von 3D Punkten mit einem Pufferradius. Das dargestellte Volumen ist durch (1) gegeben (Abb.2). Ist $r = 0$, so beschreibt $V\,(0; (p_i)_{i=1}^n)$ einen konvexen Polyeder, mit p_i als potentiellen Ecken. Im Unterschied zur üblichen Darstellung in der Computational Geometry wird auf eine explizite Verknüpfung der Ecken über Kanten und Flächen verzichtet, es können sogar einzelne der p_i im Inneren des Polyeders liegen. Dadurch wird die algorithmische Behandlung viel einfacher und ist frei von Fehlerquellen durch numerische Sonderfälle, wie z.B. (fast) doppelte Ecken oder Kanten.
Durch den zusätzlichen Pufferradius können runde Geometrien mit wenig Punkten eng umschlossen werden. Bespielsweise beschreibt $V(r; (p))$ eine Kugel, $V(r; (p_1, p_2))$ einen Zylinder mit Kugelkappen und $V(r; (p_1, p_2, p_3))$ eine abgerundete Dreiecksplatte. Der Pufferradius erlaubt ausserdem bei Rechnungen Approximationsfehler aufzuschlagen, so dass die präsentierten Algorithmen konservative Approximationen berechnen, d.h. das berechnete Volumen enthält garantiert das exakte Volumen und ist aber ggf. etwas größer.

2.4 Struktur der folgenden Abschnitte

Die nächsten Abschnitte beschreiben die Operationen auf 3D Volumina, die der Gesamtalgorithmus verwendet.
In jedem Abschnitt wird zuerst abstrakt in Mengennotation definiert, was für ein beliebiges Volumen, also eine beliebige Teilmenge von $\mathbb{R}^3$, das gesuchte Ergebnisvolumen ist. Als Beispiel sei Gleichung (5), die Translation um ein Intervall von Distanzen, betrachtet. Hier ist das Ergebnis definiert, als die Menge aller Punkt die man erhält, indem man einen Punkt aus der Eingabemenge um eine Distanz verschiebt, die innerhalb des gegebenen Intervalls liegt. Der eigentliche Beitrag der Erfindung findet sich in den dann folgenden Formeln (z.B. (7)) der Form

$$\text{Operation}\,(V\,(r; (p_i)_{i=1}^n)\,, \text{Grenzen}) = V\big(\ldots\big). \tag{3}$$

Eine solche Gleichung sagt aus, wie im Algorithmus Operation(...) auf ein im Rechner repräsentiertes Volumen angewendet wird. Dieses Eingabevolumen ist abstrakt $V\,(r; (p_i)_{i=1}^n, \ldots)$, im Rechner konkret gegeben durch den Radius r und die Punkte p_i. Die Rechnung steckt in der Klammer von $V(\ldots)$ auf der rechten Seite, die konkret angibt, was mit dem Radius r und den Punkten p_i geschieht, damit Ergebnisradius und Ergebnispunkte das Ergebnisvolumen darstellen.
Insgesamt sagt so eine Gleichung dann aus, dass die konkrete Rechnung auf den konkreten Daten die abstrakte Operation auf den, durch die konkreten Daten beschriebenen abstrakten Volumina ausführt.
In den meisten Fällen steht in der Gleichung kein $=$, sondern ein $\subset$. Dies bedeutet, dass die angegebenen Rechnung nicht exakt sondern näherungsweise ist, die Näherung aber so erfolgt, dass das konkrete Ergebnis nur größer, nie kleiner als das abstrakte Ergebnis ist. Darin liegt ein wesentlicher Beitrag der Erfindung, da diese Eigenschaft garantiert, das die Näherungen *sicher* sind.

2.5 Koordinatentransformation

K1 Ein 3D Volumen $V\left(r; p_1, \ldots, p_n\right)$ kann einfach in ein anderes Koordinatensystem transformiert werden, indem alle Punkte p_i transformiert werden. Für eine Koordinatentransformation $R \in \mathbb{R}^{3\times3}$ und Translation $t \in \mathbb{R}^3$ gilt also

$$R \cdot V\left(r; (p_i)_{i=1}^n\right) + t = V\left(r; (R \cdot p_i + t)_{i=1}^n\right). \tag{4}$$

Dies setzt voraus, dass R Längen erhält (Starrkörpertransformation, orthonormal ist), anderfalls muss r mit einer Schranke für $|Rv|/|v|$ multipliziert werden.

2.6 Berechnung der Bremszone eines Lineargelenks

Gegeben ist ein Volumen $V\left(r; p_1, \ldots, p_n\right)$ in Koordinaten am bewegten Ende eines Lineargelenkes. Es soll das Volumen berechnet werden, dass dieses überstreicht, wenn sich das Lineargelenk in einem Interval $[t_0 \ldots t_1]$ entlang der Translationsrichtung a bewegt. D.h. der Effekt des Bremsens dieses Lineargelenkes soll in das Volumen mit eingerechnet werden. Das zu beschreibende Volumen ist also

$$\mathrm{Trans}(V, a, t_0, t_1) = \left\{ v + \lambda a \middle| v \in V, \lambda \in [t_0, \ldots, t_1] \right\}, \tag{5}$$

$$V \subset \mathbb{R}^3, a \in \mathbb{R}^3, t_0, t_1 \in \mathbb{R} \tag{6}$$

T0 Die erste Möglichkeit zur Approximation ist in (5) $\lambda = \frac{t_0+t_1}{2}$ fest zu setzen und den entstehenden Fehler $\frac{t_1-t_0}{2}|a|$ auf r aufzuschlagen.

$$\mathrm{Trans}(V\left(r; (p_i)_{i=1}^n\right), a, t_0, t_1) \subset \tag{7}$$

$$V\left(r + \frac{t_1 - t_0}{2}|a|; \left(p_i + \frac{t_0 + t_1}{2}a\right)_{i=1}^n\right) \tag{8}$$

Diese Approximation ist 0. Ordnung, d.h. der Fehler wächst mit $O(|t_0 - t_1|)$.

T1 Die zweite Möglichkeit ist exakt, verdoppelt aber dafür die Zahl der Punkte in der Repräsentation. Sie erzeugt für jeden ursprünglichen Punkt p_i einen Punkt $p_i + t_0 a$ und einen Punkt $p_i + t_1 a$. Alle weiteren Punkte $p_i + \lambda a$, $\lambda \in [t_0 \ldots t_1]$ sind dann durch die Definition der Repräsentation als konvexen Hülle automatisch eingeschlossen.

$$\mathrm{Trans}(V\left(r; (p_i)_{i=1}^n\right), a, t_0, t_1) = \tag{9}$$

$$V\left(r; (p_i + t_0 a, p_i + t_1 a)_{i=1}^n\right) \tag{10}$$

Diese Technik wurde 2004 von Ericson vorgestellt [3].

2.7 Berechnung der Bremszone eines Drehgelenkes

Dieser Abschnitt beschreibt die analoge (und etwas schwierigere) Aufgabe für ein Drehgelenk. Gegeben ist ein Volumen $V\left(r; p_1, \ldots, p_n\right)$ in Koordinaten am bewegten Ende eines Drehgelenkes. Es soll das Volumen berechnet werden, dass dieses überstreicht, wenn sich das Drehgelenk in einem Interval $[\theta_0 \ldots \theta_1]$ um den Nullpunkt um Achse a dreht. D.h. der

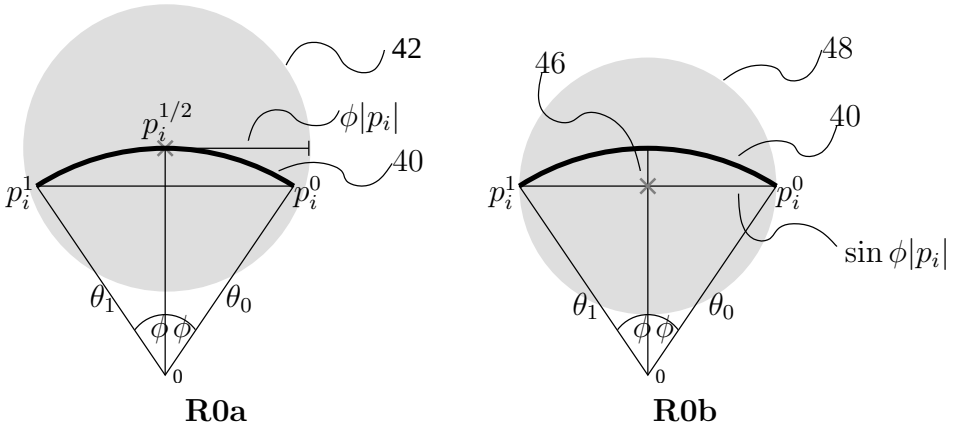


Abbildung 3: Ein Punkt p_i bewegt sich kreisförmig, zwischen Winkel θ_0 (p_i^0) und θ_1 (p_i^1). Der dabei überstrichene Kreisbogen (40) wird durch eine Kreisfläche (42,48) überdeckt. **R0a**: In der einfachen Formel liegt der Mittelpunkt ($p_i^{1/2}$) in der Mitte des Kreisbogens und der Radius entspricht der Länge des Bogensegmentes ($\phi|p_i|$) **R0b**: Bei der komplizierteren Formel liegt der Mittelpunkt bei $\cos\phi p_i^{1/2}$ (46), der Mitte zwischen den Enden des Bogens. Der Radius ist $\sin\phi|p_i|$.

Effekt des Bremsens dieses Drehgelenkes soll in das Volumen mit eingerechnet werden. Das zu beschreibende Volumen ist also

$$\mathrm{Rot}(V, a, \theta_0, \theta_1) = \left\{ \mathrm{Rot}(a, \lambda) \cdot v \middle| v \in V, \lambda \in [\theta_0, \ldots, \theta_1] \right\}, \tag{11}$$

$$V \subset \mathbb{R}^3, a \in \mathbb{R}^3, \theta_0, \theta_1 \in \mathbb{R}. \tag{12}$$

Dabei ist $\mathrm{Rot}(a, \alpha)$ die Rotationsmatrix um den Nullpunkt, Achse a und Winkel α, entsprechend der Rodriguez - Formel [4, (A.3)].
Analog zum Lineargelenk gibt es Approximationen 0. Ordnung, eine einfache und eine engere. Aber es gibt keine exakte Lösung, sondern mehrere Approximationen 1. Ordnung. Sei im folgenden für ein durch p_i beschriebenens Volumen,

$$p_i^\lambda = \mathrm{Rot}(a, (1-\lambda)\theta_0 + \lambda\theta_1) p_i \tag{13}$$

die rotierten zwischen Anfang ($\lambda = 0$) und Ende ($\lambda = 1$) des Kreisbogens.

R0a Die einfachste Approximation 0. Ordnung setzt analog zu **T0** $\lambda = \frac{t_0+t_1}{2}$ und schlägt den daraus resultierenden Fehler von $\leq \frac{\theta_1-\theta_0}{2} \max_i |p_i|$ auf den Radius r auf (Abb. 3, links).

$$\mathrm{Rot}(V\left(r; (p_i)_{i=1}^n\right), a, \theta_0, \theta_1) \subset \tag{14}$$

$$V\left(r + \frac{\theta_1 - \theta_0}{2} \max_i |p_i|; \left(p_i^{\frac{1}{2}}\right)_{i=1}^n\right) \tag{15}$$

Der Zuwachs in r ist $O(|\theta_1 - \theta_0|)$, daher die Benennung als 0. Ordnung.

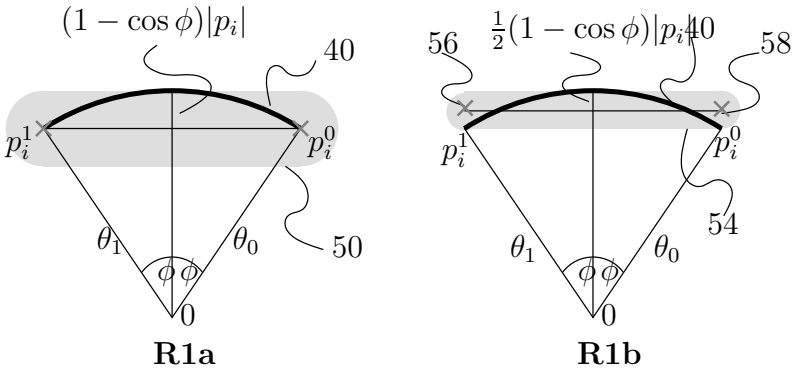


Abbildung 4: **R1a**: Der Bogen (40) wird überdeckt (50) durch die konvexe Hülle seiner Endpunkte (p_i^0, p_i^1) zzgl. eines Radius von $(1-\cos\phi)|p_i|$. **R1b**: Ein nur halb so grosser Radius ist für die überdeckende Menge (54) nötig, wenn die beiden Punkte (56, 58) auf der Höhe der Bogenmitte liegen.

R0b Die beste Approximation 0. Ordnung, d.h. die mit dem geringsten Zuwachs an r erhält man nach dem Prinzip in (Abb. 3, rechts) durch die Formel

$$\mathrm{Rot}(V\left(r;(p_i)_{i=1}^n\right), a, \theta_0, \theta_1) \subset \tag{16}$$

$$V\left(r+\sin\phi\max_i|p_i|;\left(\cos\phi p_i^{\frac{1}{2}}\right)_{i=1}^n\right), \tag{17}$$

$$\text{mit } \phi=\min\left\{\frac{\theta_1-\theta_0}{2},\frac{\pi}{2}\right\}. \tag{18}$$

Der Zuwachs in r ist kleiner als bei R0a, aber trotzdem $O(|\theta_1-\theta_0|)$.

R1a Eine Approximation 1. Ordnung erhält man, indem für jeden ursprünglichen Punkt p_i eine um $\alpha=\theta_0$ und eine um $\alpha=\theta_1$ gedrehte Kopie erzeugt wird. Durch die Definition der Repräsentation als konvexe Hülle ist damit automatisch die Strecke zwischen beiden Punkten enthalten. Es verbleibt als Fehler der Abstand zwischen dieser und dem exakten Kreisbogen, der entsteht wenn man p_i um $\alpha\in[\theta_0..\theta_1]$ dreht (Abb. 4, links). Dieser Abstand ist $\leq(1-\cos(\frac{\theta_1-\theta_0}{2}))|p_i|$ und wächst damit $O(|\theta_1-\theta_0|^2)$.

$$\mathrm{Rot}(V\left(r;(p_i)_{i=1}^n\right), a, \theta_0, \theta_1) \subset \tag{19}$$

$$V\left(r+d\max_i|p_i|;\left(p_i^0,p_i^1\right)_{i=1}^n\right), \tag{20}$$

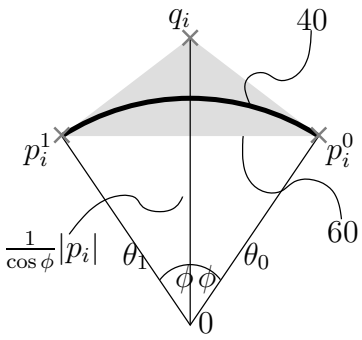
$$\text{mit } d=1-\cos\phi \tag{21}$$

Der Preis für die genauere Approximation ist eine Verdopplung der Punktzahl.

R1b Eine engere Umschreibung wird dadurch erreicht, dass die Strecke zwischen den beiden eingefügten Punkten in die Mitte des Kreisbogens gelegt wird (Abb. 4, rechts).

$$\mathrm{Rot}(V\left(r;(p_i)_{i=1}^n\right), a, \theta_0, \theta_1) \subset \tag{22}$$

$$V\left(r+\frac{d}{2}\max_i|p_i|,\left(p_i^0+\frac{d}{2}p_i^{\frac{1}{2}},p_i^1+\frac{d}{2}p_i^{\frac{1}{2}}\right)_{i=1}^n\right) \tag{23}$$



R1c

Abbildung 5: **R1c**: Wird ein Kreisbogen (40) ohne Pufferradius überdeckt (60), werden neben Anfangs- (p_i^0) und Endpunkt (p_i^1) noch der Schnittpunkt (q) der Tagenten in Anfangs- und Endpunkt benötigt. Die konvexe Hülle dieser drei Punkte (60) überdeckt den Kreisbogen (40).

Diese Approximation erhöht r um den geringstmöglichen Betrag, steht aber an den Enden um $\frac{d}{2}$ über den Kreisbogen über.

R1c Für denkbare Vereinfachungen des Algorithmus mag es von Vorteil sein, auf den Pufferradius zu verzichten. Es ist auch möglich, eine konservative Approximation zu berechnen, die den Pufferradius nicht erhöht, dafür aber aus jedem Punkt 3 Punkte macht um den Kreisbogen in einem Dreieck einzuschliessen (Abb. 5). Diese ist allerdings merklich weniger eng, selbst als die 2 Punkte Approximation aus R1b.

$$\mathrm{Rot}(V\left(r;(p_i)_{i=1}^n\right), a, \theta_0, \theta_1) \subset \tag{24}$$

$$V\left(r, \left(p_i^0,\ p_i^1,\ q_i\right)_{i=1}^n\right) \tag{25}$$

$$\text{mit } q_i = \frac{1}{\cos\phi} p_i^{\frac{1}{2}} \tag{26}$$

Asymptotisch ist der Fehler $1 - \frac{1}{\cos\phi} = O(|\theta_1 - \theta_0|^2)$

R1d Die Approximationen R1a-c lassen sich um den Preis einer erhöhten Punktzahl verbessern, indem der Bogen in Teile geteilt wird, jeder Teil des Bogen getrennt approximiert und die resultierenden Punkte zusammengefasst. Der Fehler an der Aussenseite des Bogens ist damit $O\left(\frac{|\theta_1-\theta_0|^2}{m^2}\right)$ und läßt sich so beliebig reduzieren. Der Fehler an der Innenseite ergibt sich aber allein schon durch die Beschränkung auf konvexe Volumina, wodurch unvermeidlich immer die konvexe Hülle des Bogens überdeckt wird.
Die Anzahl der einfügten Punkte multipliziert sich mit jedem Gelenk entlang einer kinematischen Kette. Daher ist diese Technik eher für den Fall eines Fahrzeugs (Sektion 4) als für Roboterarme interessant.

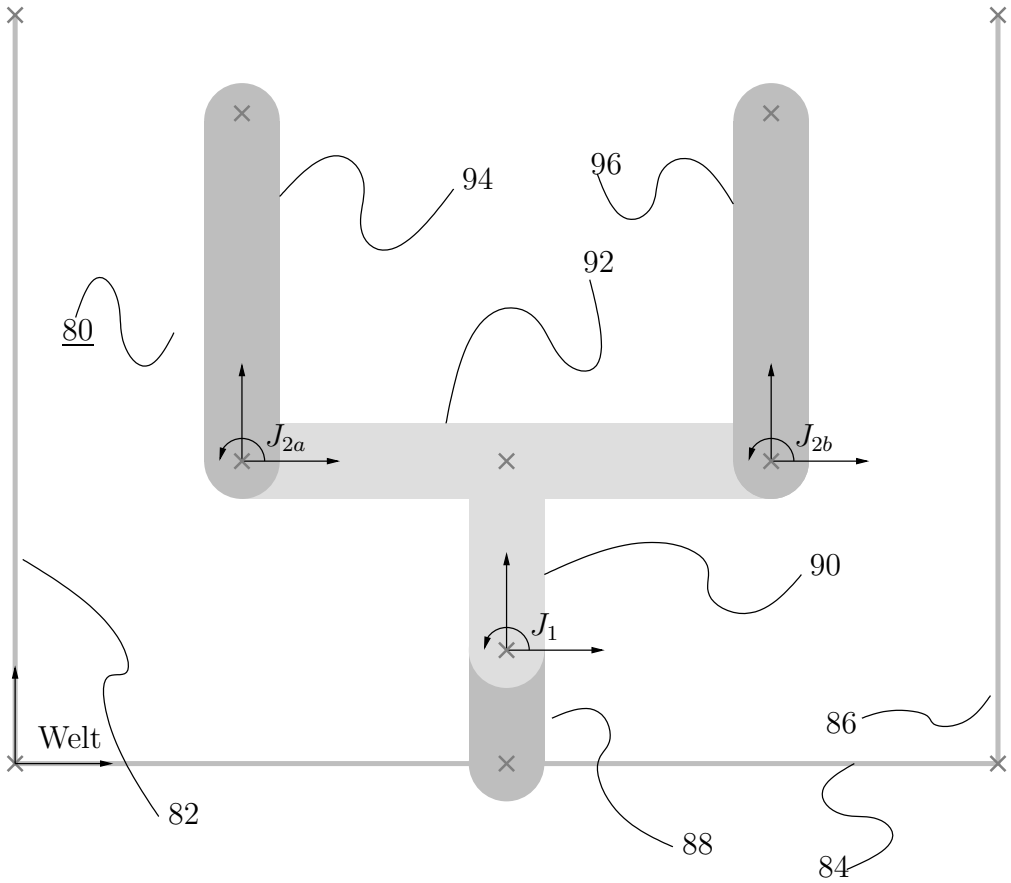


Abbildung 6: Ausgearbeitetes Beispiel: Der oben abgebildete Roboter (80) besteht aus einem Drehgelenk J_1, gefolgt von einer Gabel (90, 92) und zwei parallelen weiteren Drehgelenken J_{2a}, J_{2b}. Die Umgebung ist fest im Weltkoordinatensystem definiert aus 4 Körpern: 3 Strecken, dargestellt als konvexe Hülle zweier Punkte ohne Pufferradius (82, 84, 86) und einem Oval, dargestellt als konvexe Hülle zweier Punkte mit Pufferradius (88). Die Gabel ist fest im Koordinatensystem J_1, das sich mit dem 1. Gelenk dreht und besteht aus 2 Ovalen (90, 92). Teile 2a (94) und 2b (96) sind fest in den Koordinatensystemen J_{2a} bzw. J_{2b} die sich mit den Gelenken J_1 und J_{2a} bzw. J_{2b} mitdrehen und bestehen je aus je einem Oval. Die Punkte, die zur Definition der Körper in dieser Darstellung als $V\left(r; (p_i)_{i=1}^n\right)$ dienen, sind jeweils als graue Kreuze markiert. In diesem Beispiel sei für die Rechnungen in Abb. 7- 9 angenommen, dass alle Gelenke mit gewisser Geschwindigkeit nach links drehen.

(a) (b)

Abbildung 7: Bremszonen in den Koordinatensystemen J_{2a} und J_{2b} im Beispiel aus Abb. 6. (a) Die Bremszone von Teil 2a in J_{2a} ist der ursprüngliche Körper, da dieser fest in J_{2a} ist. Andere Körper haben keine Bremszonen in diesem Koordinatensystem, da Bremszonen eines Körpers vom Körperkoordinatensystem bis zur Weltkoordinatensystem berechnet werden. (b) Analoges gilt für Teil 2b in J_{2b}.

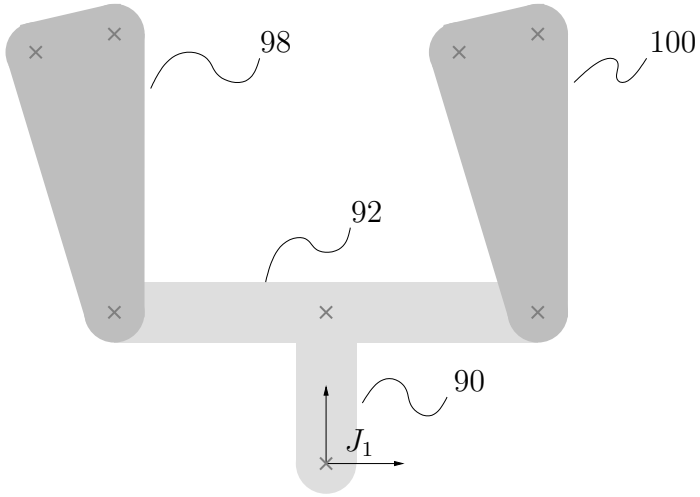


Abbildung 8: Bremszonen im Koordinatensystem J_1 im Beispiel aus Abb. 6. Das Beispiel nimmt an, dass sich alle Gelenke mit gewisser Geschwindigkeit nach links drehen. Teil 1 (90, 92) ist fest in J_1, so dass die Bremszone aus den ursprünglichen Körper besteht. Auf die Körper von Teil 2a/b (90, 92 in Abb. 6) wird das Winkelintervall der Gelenke 2a/b nach R1b aufgeschlagen (98, 100). Dadurch entsteht aus jedem der beiden Punkte eine gedrehten und eine ungedrehte Kopie. Das Ergebnis sind durch 4 Punkte definierte Volumina, bei denen in diesem Fall 2 Punkte identisch sind, weil sie auf der Drehachse liegen. Die Entfernung zwischen Teil 2a und 2b wird mit diesen Bremszonen berechnet, so dass Gelenk 1 keinen Einfluss auf sie hat.

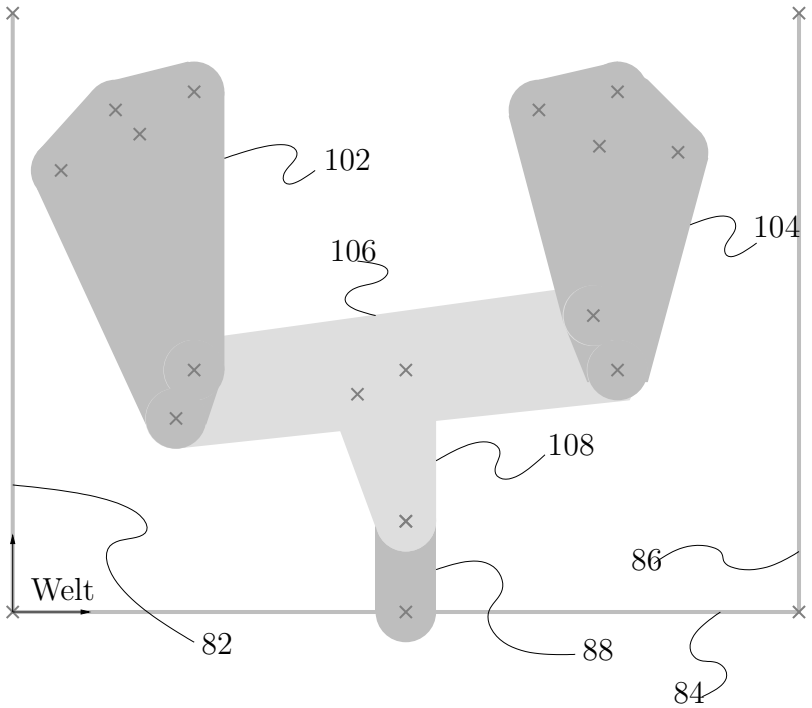


Abbildung 9: Bremszonen im Weltkoordinatensystem im Beispiel aus Abb. 6. Das Beispiel nimmt an, dass sich alle Gelenke mit gewisser Geschwindigkeit nach links drehen. Die Umgebung (82, 84, 86, 88) ist fest in der Welt, so dass die ursprünglichen Körper gleichzeitig Bremszonen im Weltsystem sind. Auf Teil 1 (90, 92 in Abb. 8), sowie die Bremszonen von Teil 2a/b in J_1 (98, 100 in Abb. 8) wird das Winkelintervall von Gelenk 1 nach R1b aufgeschlagen (102, 104, 106, 108). Dadurch verdoppeln sich die Punkte in den vier involvierten Volumina. Die Entfernungen zwischen allen Roboterteilen und der Welt wird mit diesen Bremszonen berechnet.

2.8 Berechnung der Bremszonen eines kinematischen Mechanismus

Dieser Abschnitt beschreibt, wie die Berechnungen der Bremszonen für einzelne Gelenke (Sektion 2.6, 2.7) hintereinander ausgeführt werden um die Bremszonen für einen komplexeren kinematischen Mechanismus, z.B. einen Roboter zu berechnen.
Die vorkonfigurierte Szene besteht aus Körpern, von denen jeder als $V\left(r;(p_i)_{i=1}^n\right)$ im jeweiligen Körperkoordinatensystem abgespeichert ist. Die Koordinatensysteme sind durch einen kinematischen Baum verbunden. Das heisst, jedes Koordinatensystem ist durch eine Kette von Transformationen zu dem Weltkoordinatensystem, der Wurzel des Baumes in Beziehung gesetzt. Die Transformationen haben einen konstanten vorkonfigurierten Anteil, und bei beweglichen Teilen einen Anteil der von der jeweiligen Gelenkposition als Rotation oder Verschiebung abhängt. Dies entspricht der üblichen Definition einer Vorwärtskinematik [4, chapter 3].
Im Schritt b) des Algorithmus werden für jeden Körper sukzessive die Bremszonen in allen Koordinatensystemen, vom Körperkoordinatensystem bis zur Wurzel berechnet. Dabei wird mit der vorkonfigurierten Beschreibung des Körpers im Körperkoordinatensystemen gestartet und die Transformationen sukzessive angewendet. Für feste Transformationen wird **K1** benutzt, bevorzugt **T0** oder **T1** für Lineargelenke, und **R0b**, **R1b** für Rotationsgelenke.
Dabei ist die Wahl zwischen 0. Ordnung und 1. Ordnung ein vorkonfigurierter Kompromiss zwischen Genauigkeit und der sich jeweils verdoppelnden Anzahl an Punkten in den Volumina. Bei langen kinematischen Ketten, z.B. Robotern haben meist die unteren Achsen den größten Einfluss auf den Bremsweg und sollten deshalb in 1. Ordnung approximiert werden, die oberen Achsen haben einen geringeren Einfluss und können 0. Ordnung approximiert werden. Da die Anzahl der Punkte in den vorderen Teilen eines Roboters sich mit jeder Achse 1. Ordnung verdoppelt, ist es für diese Teile viel wichtiger, als für die statische Umgebung, dass sie mit wenigen Punkten approximiert wird.
Ein größeres Beispiel für das Vorgehen des Algorithmus ist in Abbildungen 6 bis 9 beschrieben.
Diese Vorgehensweise an sich ist nicht neu, sie wurde z.B. von Fuhrmann [5, Spalte 3], [6, 0008] beschrieben und ist nach seinen Angaben in gängigen CAD Programmen als “swept volume” Funktion implementiert. Diese Rechnungen sind genau, aber auch sehr rechenaufwändig, so dass sie sich nicht für Echtzeitanwendungen eignen. Im Gegensatz dazu sind die in Sektionen 2.5-2.7 beschriebenen Operationen weniger exakt, aber konservativ und dafür schnell genug für die Berechnung in Echtzeit.

2.9 Distanzberechnung zwischen den Bremszonen

Für jedes Paar von Körpern wird die Distanz zwischen den Bremszonen beider Körper mit dem bekannten GJK Algorithmus bestimmt [2]. Der GJK Algorithmus benötigt eine von vornherein nicht bekannte Anzahl von Iterationen um die Distanz zwischen den beiden konvexen Volumina zu bestimmen. Er liefert allerdings nach jeder Iteration eine untere und eine obere Schranke für die Distanz, so dass häufig vorzeitig abgebrochen werden kann, wenn klar ist dass die Distanz gross genug ist. In jeder Iteration betrachtet der Algorithmus einen Simplex von ein bis drei Punktpaaren als Kandidat für die kürzeste

Verbindung, bestimmt jeweils die kürzesten Distanz auf diesem Simplex und sucht, ob es Punktpaare gibt, die näher dran liegen. Diese Information wird über die verschiedenen Takte der Robotersteuerung hinweg aufbewahrt. Dadurch startet der GJK Algorithmus jeweils mit dem Simplex, der im letzten Takt die kürzeste Distanz beinhaltete. Obwohl sich die Volumina natürlich ein bischen bewegt und verformt haben, ist dies ein gute Kandidat. Dadurch wird die nötige Zahl der Iterationen erheblich verringert.
Massgeblich für die Distanzberechnung sind die beiden Bremszonen im ersten gemeinsamen Koordinatensystem im kinematischen Baum auf dem Weg zur Wurzel. Das bedeutet z.B. dass bzgl. der Kollision zweier Roboterarme auf einem Fahrzeug, die Bremszonen im Fahrzeugkoordinatensystem auf Distanz geprüft werden. Der Bremsweg des Fahrzeugs selbst betrifft beide gleichermassen, ihre Relativlage daher nicht und wird dementsprechend auch nicht mit eingerechnet.
Nicht gegeneinander getestet werden ausserdem Paare von Körpern, zwischen denen kein Gelenk liegt. Ausserdem können Paare von Körpern als nicht zu testen vorkonfiguriert werden. Dies ist nötig, weil z.B. die beiden Seiten eines Gelenkes geometrisch immer kollidieren.

3 Beschränkung des Rechenaufwandes (Erfinderischer Beitrag II)

Das soweit beschriebene Verfahren ist verhältnismäßig schnell, hat aber zwei Rechenzeitprobleme. Zum einen werden alle Paare von Körpern getestet, dadurch wächst die Rechenzeit quadratisch mit der Anzahl Körper. Viele etablierte Verfahren verwenden eine Hierarchie sogenannter Boundingvolumina um dieses Problem zu lösen. Haben zwei Obervolumina eine gewisse Distanz, so ist die Distanz für alle Paare von Untervolumina mindestens genauso gross. Haben also zwei Boundingvolumina eine genügend hohe Distanz, braucht die Distanz aller Unterkörper nicht berechnet werden.
Diese Vorgehensweise ist im Kontext dieses Algorithmus auch möglich, aber relativ kompliziert. Daher soll sie vermieden werden um den Algorithmus einfach zu halten, insbesondere für einen sicherheitsgerichteten Einsatz.
Ein zweites Problem ist, dass die Distanzberechnung mit dem GJK Algorithmus eine von vornherein nicht klare Anzahl an Iterationen benötigt. In einer Anlagen- oder Robotersteuerung laufen aber alle Prozesse in einem festen Takt, dadurch müsste ein Vielfaches der mittleren Rechenzeit für den Algorithmus reserviert werden, damit auch unter ungünstigen Bedingungen die Zykluszeit eingehalten werden kann.
Um diesen beiden Problemen zu begegnen, wird das Grundverfahren noch einmal verfeinert (erfinderischer Beitrag II). Dadurch wird die Rechenzeit sehr viel kleiner und auf einen festen Wert beschränkt.

3.1 Überblick

3a) Es für jede Bremszone i ihre Änderung seit dem letzten Zyklus als pauschaler Änderungsradius δr_i abgeschätzt. Daraus wird eine Schranke für die Distanz aller Paare von Bremszonen hergeleitet, indem von der Distanz der Bremszonen i

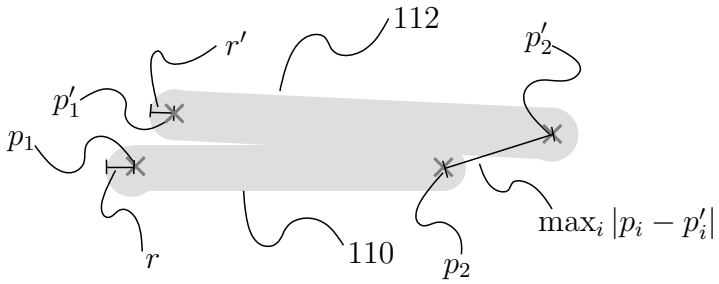


Abbildung 10: Berechnung des Änderungsradius: Die Abbildung zeigt eine Bremszone in einem Taktzyklus (110) dargestellt als $V(r, (p_i)_{i=1}^n$ mit $n = 2$ und die korrespondierende Bremszone im nächsten Taktzyklus (112), dargestellt als $V(r', (p'_i)_{i=1}^n$ Der Ausdruck $\max_i |p_i - p'_i| + r' - r$ ist eine obere Schranke für die Distanz eines Punktes der neuen Bremszone zur alten Bremszone. Werden Distanzschranken für die alte Bremszone um diesen Betrag verringert, so sind sie auch gültige Schranken für die neue Bremszone.

und j aus dem letzten Zyklus, die Änderungsradien $\delta r_i + \delta r_j$ abgezogen werden. Dadurch ergibt sich eine konservative Abschätzung für die neue Distanz.

3b) Es wird eine feste Anzahl an Iterationen des GJK Algorithmus durchgeführt. Zuerst für die Paare, bei denen obige Rechnung Distanz 0 ergeben hat, danach reihum. Ist danach immer noch ein Paar 0, wird gestoppt.

Erfinderischer Beitrag II ist dabei das Schema zur konservativen Fortschreibung und zyklischen Neuberechnung der paarweisen Distanzen um die Rechenzeit auf einen festen Wert zu beschränken und trotzdem stets sicher zu bleiben.

3.2 Berechnung des Änderungsradius

Anschaulich gesprochen ist der Änderungsradius δr_j die maximale Distanz, die sich ein Punkt der Bremszone j seit dem letzten Zyklus bewegt hat. Für jeden Punkt der neuen Bremszone ist also die Distanz zur alten Bremszone $\leq \delta r_j$. Formal ist dies ein Radius, so dass, wenn man die alte Zone um diesen Radius in alle Richtungen vergrössert, sie die neue Bremszone einschliesst.
In dem speziellen Fall dieses Algorithmus lässt sich der Änderungsradius einfach durch die maximale Änderung in einem der Punkte p_i von Bremszone j plus der Differenz der Pufferradien berechnen (Abb. 10).

$$\text{Buffer}(V, r) := \left\{ v + p | v \in V, p \in \mathbb{R}^3, |p| \leq \max(r, 0) \right\} \tag{27}$$

$$V(r'; (p'_i)_{i=1}^n) \subset \text{Buffer}(V(r; (p_i)_{i=1}^n), \delta r), \text{ mit } \delta r_j = \max_i |p_i - p'_i| + r' - r \tag{28}$$

Dabei ist $V(r'; (p'_i)_{i=1}^n)$ die Darstellung der neuen Bremszone j und $V(r; (p_i)_{i=1}^n)$ die Darstellung der alten Bremszone j aus dem letzten Zyklus der Steuerung. Ist für ein Paar von Bremszonen i, j, D_{ij} ihre Distanz im letzten Zyklus, so ist die Distanz im neuen Zyklus

$$D'_{ij} \geq D_{ij} - \delta r_i - \delta r_j. \tag{29}$$

Die Begründung ist: Wäre D'_{ij} kleiner, so gäbe es einen Punkt in Bremszone i und einen in Bremszone j, die eine kleinere Distanz, als $D_{ij} - \delta r_i - \delta r_j$ hätten. Der Punkt in Bremszone

i hat sich höchstens δr_i seit dem letzten Zyklus bewegt, analog der Punkt in Bremszone j. Demnach hätten die Punkte im letzten Zyklus eine Distanz $< D_{ij}$ gehabt. Das ist ein Widerspruch dazu, dass die Distanz im letzten Zyklus $\geq D_{ij}$ war.
Das Auswerten von (29) für alle Paare i, j von Bremszonen hat zwar rechnerisch immer noch quadratischen Aufwand, ist aber vernachlässigbar, da es sich nur um 2 Subtraktionen pro Paar handelt.

3.3 Kontrollalgorithmus zur Neuberechnung von Distanzen

Die Grundidee des Kontrollalgorithmus ist folgende: Der Algorithmus hält für jedes Paar i, j von Bremszonen eine untere Schranke für die Distanz und aktualisiert diese in jedem Zyklus durch Abziehen von δr_i und δr_j. Solange alle Distanzschranken > 0 sind, müssen alle echten Distanzen > 0 sein und eine Kollision kann ausgeschlossen werden. Ist für ein Paar die Distanzschranke ≤ 0, so könnte es sein, dass sie wirklich kollidieren, es könnte aber auch sein, dass einfach die Schranke übervorsichtig ist. Daher versucht der Algorithmus durch genaueres Nachrechnen auf den Bremszonen zu etablieren, dass die Distanz in Wirklichkeit größer ist. Schafft er das nicht, z.B. weil die Zonen wirklich kollidieren, muss der Roboter/die Anlage gestoppt werden.
Dieses Etablieren einer größeren Distanzschranke geschieht durch eine Iteration des GJK Algorithmus. Der GJK Algorithmus benötigt nämlich eine vorher unbekannte Anzahl Iterationen um die wirkliche Distanz zu berechnen, liefert aber mit jeder Iteration eine – immer besser werdende – untere Schranke.
Als Beispiel sei ein Paar von Bremszonen betrachtet, dass relativ weit, z.B. $1m$ voneinander entfernt ist und von dem sich eine Bremszone pro Zyklus $1cm$ hin und her bewegt. Eine hohe Entfernung ist durchaus typisch für die meisten Paare von Bremszonen. Durch das Fortschreiben nach Sektion 3.2 sinkt die Distanzschranke jeden Zyklus um $1cm$, so dass nach 100 Zyklen der Algorithmus eine Distanzschranke von 0 hat und nachrechnen muss, wie gross die Distanz wirklich ist. Effektiv muss in diesem Beispiel nur jeden 100ten Zyklus die GJK Distanzrechnung durchgeführt werden. 99 von 100 Zyklen kommen mit den zwei Subtraktion der Änderungsradien aus.
Der Kontrollalgorithmus darf eine feste Anzahl an GJK-Iterationen auf beliebigen Paaren von Bremszonen durchführen. Dadurch ist die Rechenzeit fest und nicht mehr variabel. Mit diesen Iterationen muss er versuchen, alle Distanzschranken > 0 zu halten und somit nachzuweisen, dass keine Kollision vorliegt. Schafft er das nicht, muss er anhalten. Solange also ein Paar von Bremszonen Distanzschranke ≤ 0 hat, führt der Kontrollalgorithmus GJK-Iterationen auf diesem Paar durch.
Bleiben danach noch GJK-Iterationen im Zeitbudget über, führt der Kontrollalgorithmus GJK-Iterationen auf allen Paaren von Bremszonen reihum aus. Das bewirkt, dass sich die Distanzschranken dieser Paare erhöhen und mehr Zeit bleibt, bis auf dem Paar eine erneute GJK-Iteration notwendig wird. Dadurch arbeit der Algorithmus sozusagen vor.
Diese Strategie hat sehr hilfreiche Eigenschaften: Sie ist adaptiv in Bezug auf Geschwindigkeit und Entfernungen und überwindet die Problematik quadratischer Rechenzeit. Die δr_i entsprechen grob der im Zyklus zurückgelegten Strecke. D.h., bewegt sich die Anlage langsamer, stehen mehr Zyklen, also mehr Iterationen zur Verfügung um die notwendige Anzahl GJK-Iterationen durchzuführen. Für viele Paare sind die Distanzen gross, z.B. weil die Anlage sich gerade weit von der Umgebung entfernt, oder für solche Paare, die

zu weit entfernten Anlagenteilen gehören. Dann muss dieses Paar selten neu berechnet werden und es reicht fast immer eine einzelne GJK Iteration. Ausserdem bewegt sich natürlich kein Körper nahe an allen Teilen der Umgebung.
Insgesamt sieht man, dass der Algorithmus immer sicher ist, wenige Iterationen braucht und nur in absoluten Ausnahmefällen einen unnötigen Halt einleitet.

4 Integration von Fahrzeugen (Erfinderischer Beitrag III)

Dieser Abschnitt beschreibt, wie Fahrzeuge in den Algorithmus integriert werden können, d.h. drohende Kollisionen zwischen ihnen und der vorkonfigurierten Anlage rechtzeitig detektiert werden können. Die sensorielle Absicherung gegen Personen, ist Gegenstand des nächsten Abschnittes. Zentrales Problem ist, dass Position / Orientierung eines Fahrzeugs in Bezug auf die Welt variabel sind, einen Bremsweg haben, aber nicht durch eine Abfolge eindimensionaler Achsen, wie bei einer festen Anlage, beschrieben sind.
Das Fahrzeug muss dazu seine Position und Orientierung (Pose) in der Ebene (x, y, θ) und deren Ableitung, also seine Geschwindigkeit messen. Der Vektor (x, y) beschreibt dabei die Position eines fest gewählten Referenzpunktes am Fahrzeug in der Welt und zusammen mit θ eine Transformation von Fahrzeugkoordinaten zu Weltkoordinaten.

4.1 Definition des Bremsweges eines Fahrzeugs analog zum Bremsinterval

In Schritt a) wird der Bremsweg des Fahrzeugs ausgerechnet. Dieser ist, anders als bei einem eindimensionalen Gelenk, kein Intervall mehr sondern wird als konvexe Hülle von Konfigurationen im Konfigurationsraum dargestellt.

$$K\left((k_j)_{j=1}^m\right) = \left\{ \sum_{j=1}^m \lambda_j k_j \Big| \lambda_j \geq 0 \forall j, \quad \sum_{j=1}^m \lambda_j = 1 \right\}, \quad k_j \in \mathbb{R}^3 \tag{30}$$

Dabei besteht eine Konfiguration k_j aus drei Komponenenten

$$k_j = \begin{pmatrix} k_{jx} \\ k_{jy} \\ k_{j\theta} \end{pmatrix} \tag{31}$$

von denen (k_{jx}, k_{jy}) die Position eines gewählten Referenzpunkt des Fahrzeuges und $k_{j\theta}$ die Orientierung des Fahrzeugs ist. Die Angaben beziehen sich zweckmässigerweise auf das Koordinatensystem des Fahrzeugs am Anfang der Bremsung (d.h. die Konfiguration $(0, 0, 0)$ ist eine der k_j). Dadurch wird durch ein k aus dem Konfigurationsraum eine Transformation zwischen körperfesten Koordinaten des Fahrzeugs in Konfiguration k und weltfesten Koordinaten definiert:

$$T(k) = \begin{pmatrix} \cos k_\theta & -\sin k_\theta & 0 & k_x \\ \sin k_\theta & \cos k_\theta & 0 & k_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad k = \begin{pmatrix} k_x \\ k_y \\ k_\theta \end{pmatrix} \in \mathbb{R}^3 \tag{32}$$

Dies ist eine homogene Transformationsmatrix, die Rotation und Translation kombiniert. Zur Vereinfachung der Notation sei daher, wie üblich vereinbart, das Ortsvektoren, besonders die p_i mit einer impliziten 1 als vierte Komponente versehen sind.

4.2 Berechnung der Bremszone eines Körpers am Fahrzeug

Gesucht ist die Menge aller Punkte in der Welt, die von einem am Fahrzeug festen Volumen überstrichen werden, wenn das Fahrzeug alle in $K\left((k_i)_{i=1}^n\right)$ definierten Bremskonfigurationen durchläuft.

$$\text{Brake}(V, K) = \left\{ T(k) \cdot v \middle| v \in V, k \in K \right\} \tag{33}$$

F1a Die Vorgehensweise ist analog zur 1. Ordnung Approximation der Rotation. Für ein als konvexe Hülle definiertes $V\left(r; (p_i)_{i=1}^n\right)$ und $K\left((k_j)_{j=1}^m\right)$ wird für jedes Paar p_i, k_j das Ergebnis berechnet und alle in einer konvexen Hülle zusammengefasst. Wäre die Drehung linear in θ wären alle Zwischenwerte durch die Definition als konvexe Hülle schon automatisch enthalten. So wird die Nichtlinearität der Drehung abgeschätzt und ihr Effekt auf r addiert.

$$\text{Brake}\left(V\left(r; (p_i)_{i=1}^n\right), K\left((k_j)_{j=1}^m\right)\right) \subset \tag{34}$$

$$V\left(r + (1 - \cos\phi) \max_i |p_i|; (T(k_j) \cdot p_i)_{i,j=1}^{n,m}\right), \tag{35}$$

$$\phi = \min\left(\frac{\max_j k_{j\theta} - \min_j k_{j\theta}}{2}, \frac{\pi}{2}\right) \tag{36}$$

4.3 Parametrisierung der Bremsbewegung

Nimmt man an, dass das Fahrzeug beim Bremsen einer exakten Kreisbahn folgt ergibt sich eine einfachere Lösung durch Anwendung von einer der Approximationen **R0a-R1d**. Fahrzeuge können kontinuierlich von einer Kurve in Geradeausfahrt übergehen. Dabei wandert das Drehzentrum ins Unendliche. Deshalb wird, anders als beim Roboterarm, eine Version der Formeln benötigt, bei der Geradeausfahrt keinen Sonderfall darstellt und insbesondere der Nullpunkt des Koordinatensystems nicht im Drehzentrum liegen muss. Dazu wird die relative Bremskonfiguration (s, α) bzgl. der Fahrzeugkonfiguration zu Bremsbeginn verwendet. s beschreibt dabei die auf der Kreisbahn zurückgelegte Bremsweglänge (vorzeichenbehaftet) und α die Änderung der Orientierung (bzgl. Θ). Eine Geradeausfahrt entsteht für $\alpha = 0$ und $s \neq 0$.

Koordinatentransformation Die Darstellung der Bremszone des Fahrzeugs erfolgt im Fahrzeugkoordinatensystem beim Bremsstart (Abb. 11). Dieses System hat seinen Ursprung im Referenzpunkt $(x, y)^T$ des Fahrzeuges zu Bremsbeginn, seine z-Achse zeigt nach oben und seine x-Achse entlang der Orientierung Θ in Fahrtrichtung. Die Koordinatentransformation wird durch die folgende Transformationsmatrix $T(s, \alpha)$ beschrieben.

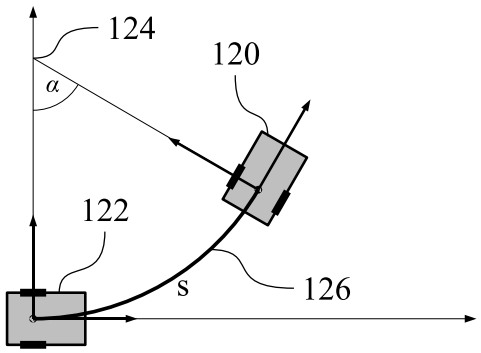


Abbildung 11: $\boldsymbol{T(s,\alpha)}$: Koordinatentransformation von Koordinaten des Fahrzeugs das sich (s,α) entlang eines Kreisbogens (126) bewegt hat (120) in Koordinaten des Fahrzeugs am Anfang der Bewegung (122).

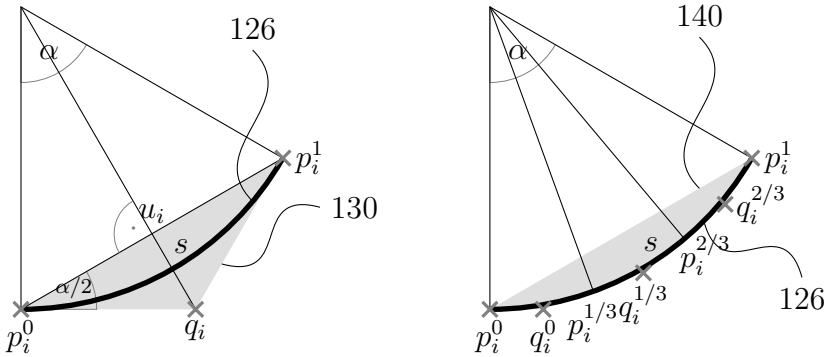


Abbildung 12: **F1c**: p_i^0,p_i^1 und q_i definieren eine konvexe Obermenge (130) des Kreisbogens (126). p_i^1 ist das Bild von p_i^0 nach Transformation entsprechend der relative Bremskonfiguration (s,α). Der Punkt q_i berechnet sich in zwei Schritten: a) $u_i = \frac{p_i^0+p_i^1}{2}$ b) $q_i = u_i + \tan\frac{\alpha}{2}\left(\begin{smallmatrix}0&1\\-1&0\end{smallmatrix}\right)(u_i - p_i^0)$. In Gleichung (40) wird eine zusammengefasste Formel für die Schritte a) und b) verwendet. **F1d**: Die konvexe Hülle des Kreisbogens (126) kann beliebig genau approximiert werden. Hier für $h = 3$ eine Approximation durch ein Fünfeck (140).

$$T(s,\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & s\,\mathrm{sinc}\,\frac{\alpha}{2}\cos\frac{\alpha}{2} \\ \sin\alpha & \cos\alpha & 0 & s\,\mathrm{sinc}\,\frac{\alpha}{2}\sin\frac{\alpha}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{37}$$

Bremsvolumen Das vom Fahrzeug während des Bremsvorganges mit Bremsweg s und Orientierungsänderung α überstrichene Volumen ist dann

$$\mathrm{Rot}(V,s,\alpha) = \left\{ T(\lambda s, \lambda\alpha)\cdot v \middle| \lambda \in [0\ldots 1], v \in V \right\} \tag{38}$$

4.4 Vereinfachte Berechnung mit exakter Bremskurve

Dieser Abschnitt beschreibt zwei einfache Approximationen von $\mathrm{Rot}(V\,(r;(p_i)_{i=1}^n)\,,s,\alpha)$, die ohne einen Aufschlag auf den Pufferradius r auskommen. Zunächst wird eine **R1c** entsprechende Lösung angegeben, die den Kreisbogen in ein Dreieck einschließt und so aus jedem Punkt 3 Punkte macht. Die dann folgende Erweiterung der Lösung implementiert die Idee aus **R1d** und schließt den Kreisbogen letztlich in ein $(h+2)$-Eck ein. Grundsätzlich ist sie in der Lage die konvexe Hülle des Bogens um den Preis einer erhöhten Anzahl von Punkten beliebig genau zu approximieren.

F1c Der Kreisbogen wird wie in Abbildung (Abb. 12, links) beschrieben durch ein Dreieck umschlossen. Dazu wird der Schnittpunkt der beiden in Anfangs- und Endpunkt angelegten Tangenten ermittelt. Diese Methode ist unabhängig vom Koordinatensystem in dem P und S gegeben sind und benötigt keine Sonderbehandlung im Falle einer Geradeausfahrt.

$$\mathrm{Rot}(V\,(r;(p_i)_{i=1}^n)\,,s,\alpha) \subset \tag{39}$$

$$V\left(r,\left(p_i^0,p_i^1,q_i\right)_{i=1}^n\right) \tag{40}$$

$$\text{mit } p_i^\lambda = T(\lambda s,\lambda\alpha)\cdot p_i \tag{41}$$

$$\text{und } q_i = p_i^0 + Q(\alpha)\cdot\frac{1}{2}\left(p_i^1 - p_i^0\right) \tag{42}$$

$$\text{und } Q(\alpha) = \begin{pmatrix} 1 & \tan\frac{\alpha}{2} & 0 & 0 \\ -\tan\frac{\alpha}{2} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{43}$$

F1d Die Erweiterung teilt den Kreisbogen in h gleiche Teile, approximiert jeden Teil getrennt und fasst dann alle Punkte zusammen. Dazu werden die Trennpunkte $p_i^{k/h} = T(\frac{k}{h}s,\frac{k}{h}\alpha)p_i^0$ für $k=0,\ldots,h$ berechnet und dann für je zwei benachbarte Trennpunkte entsprechend **F1c** der fehlende Dreieckspunkt berechnet. Dabei ist die Teilbogenlänge jeweils $\frac{s}{h}$ und die Winkeländerung $\frac{\alpha}{h}$. Die Trennpunkte $p_i^{k/h}$ werden, mit Ausnahmen von Startpunkt $(k=0)$ und Endpunkt $(k=h)$ des gesamten Kreisbogens, nicht für die Approximation benötigt, da sie in der konvexen Hülle der restlichen Punkte bereits

vorhanden sind. Es entstehen damit $h+2$ anstatt 3 Punkte aus jedem ursprünglichen Punkt p_i.

$$\mathrm{Rot}(V\left(r;(p_i)_{i=1}^{n}\right),s,\alpha)\subset \tag{44}$$

$$V\left(r,\left(p_i^0,p_i^1,\ \left(q_i^{\frac{k}{h}}\right)_{k=0}^{h-1}\right)_{i=1}^{n}\right) \tag{45}$$

$$\text{mit } q_i^\lambda = p_i^\lambda + Q(\frac{\alpha}{h})\cdot\frac{1}{2}\left(p_i^{\lambda+\frac{1}{h}}-p_i^\lambda\right) \tag{46}$$

Zum gleichen Ergebnis gelangt man, indem zunächst q_i^0 und daraus alle weiteren $q_i^{k/h}$ wie folgt berechnet werden

$$q_i^\lambda = T(\lambda s,\lambda\alpha)\cdot q_i^0 \tag{47}$$

5 Schutzfeldberechnung für Laserscanner (Erfinderischer Beitrag IV)

Das bisherige Verfahren betrachtet nur Umgebungen die geometrisch vorkonfiguriert sind, bei denen bewegliche Anlagenteile mit Positions- bzw. Winkelgebern versehen sind, aber bei denen weitere Objekte oder Personen in der Anlage nicht sensoriell erfasst werden. Dieser Abschnitt erweitert sie auf von einem Laserscanner sensoriell erfasste Hindernisse, besonders Personen. Dies geschieht, indem aus den Bremszonen Schutzfelder für den Laserscanner berechnet werden, die dieser dann überwacht. Laserscanner tasten die Umgebung mit einem Laserstrahl ab und messen dadurch in der Ebene in jeder Richtung die Entfernung zum nächsten Hindernis. Ein Schutzfeld gibt für jeden Strahl des Scanners, also jeden Winkel in der Ebene an, bis zu welcher Entfernung vom Scanner die Umgebung frei von Hindernissen sein muss. Befindet sich in diesem Bereich ein Hindernis, vornehmlich eine Person, stoppt der Laserscanner die Anlage.
Es gibt zwei praktisch besonders interessante Einsatzfälle: a) Ein Roboterarm mit festem Laserscanner, b) ein Fahrzeug mit am Fahrzeug befindlichen Laserscanner.
Letzterer Fall hat eine weitere Besonderheit: Es muss die Pose des Fahrzeugs nicht gemessen werden, da das Schutzfeld relativ zum Fahrzeug definiert ist. Trotzdem geht die Geschwindigkeit in die Berechnung des Bremsweges ein. Diese Sondersituation ergibt sich daraus, das der Laserscanner am Fahrzeug befestigt ist, aber Hindernisse wahrnimmt, die als fest in der Welt angenommen werden.

5.1 Berechnung der Schutzfelder

Ein 3D Volumen $V\left(r;(p_i)_{i=1}^{n}\right)$ wird einfach durch weglassen der Z Koordinate in die Ebene transformiert. Auf den resultierenden Punkten wird ein konvexe Hüllen Algorithmus angewandt, der die Punkte, die Ecken von $V(\ldots)$ bilden entgegen den Uhrzeigersinn zu einem Polygon durchnummeriert. Das Polygon wird durch Kreise von r um die Ecken und Parallelen zu den Kanten um r Erweitert.
Alle in Frage kommenden Strahlen des Laserscanners werden mit allen Kanten und Kreisen geschnitten. Für jeden Strahl ist das Schutzfeld die größte Entfernung eines solchen Schnittes.

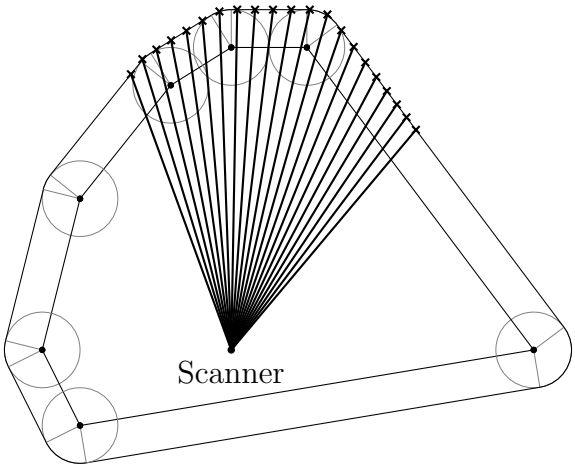


Abbildung 13: Das Polygon ist das Ergebnis des konvexe Hüllen Algorithmus, der auf die in die Ebene transformierten Punkte angewandt wurde. Das Polygon wird um den Radius r erweitert und dann das Schutzfeld als Ergebnis der Schnitte zwischen den Laserstrahlen und dem erweiterten Polygon ermittelt.

6 Detailfragen

a) Welches technische Problem wird durch die Erfindung gelöst? (Erfinderischer Beitrag I, II) Die Bewegung von einem oder mehreren Roboterarme zu überwachen und rechtzeitig einen Halt auszulösen bevor die Arme miteinander oder mit der Umgebung kollidieren. Dies vermeidet Unfälle in Situationen, wo der Roboter nicht immer exakt die selbe Bahn abfährt, z.B. beim teach-in oder bei sensorgeführten Arbeiten. Das Verfahren dient zur Kollisionsvermeidung mit geometrisch vorab bekannten Hindernissen, also in der Grundversion nicht zum Schutz von Personen.
(Erfinderischer Beitrag III) Erweiterung der selben Aufgabe auf gesteuerte Fahrzeuge, auf denen sich auch Roboterarme befinden können.
(Erfinderischer Beitrag IV) Erweiterung des Verfahrens auf sensoriell erfasste Hindernisse. Das so erweiterte Erfahren dient auch zur Kollisionsvermeidung mit Personen.

b) In welcher Weise wurde das Problem bisher gelöst? Es gibt sehr ausgefeilte Lösungen für a-priori Bahnplanung, wo die Bahn im Rechner auf Kollision getestet und dann in den Roboter geladen wird. Dazu bedarf es aber einer festen Bahn, die Roboterbewegung kann nicht manuell oder durch Sensoren gesteuert werden.
Stand der Technik ist ebenfalls das Testen der aktuellen Stellung des Roboters auf Distanz zu Hindernissen, so dass in einem festen Sicherheitsabstand zum Hindernis gebremst werden kann. Dies ist aber nur für kleine Geschwindigkeiten praktikabel. Unterschiedliche Teile des Roboters bewegen sich nämlich sehr unterschiedlich schnell und brauchen sehr unterschiedliche Bremswege. Ausserdem hat der Bremsweg eine Richtung, nämlich die Bewegungsrichtung, während ein Sicherheitsabstand in allen Richtungen und an allen Punkten des Roboters gleich wirkt. Dadurch wird bei höheren Geschwindigkeiten der Sicherheitsabstand so konservativ, das das System oft stoppt, obwohl dies unnötig wäre. Ein Patent [7] schlägt vor, statt der aktuellen Stellung die Stellung auf Kollision zu testen

bei der der Roboter anhalten wird. Das ist besser, weil es keinen Sicherheitsabstand in alle Richtungen aufschlägt, sondern die Bewegungsrichtung berücksichtigt. Es ist aber nicht wirklich sicher, weil es nicht ausschliesst, dass während des Bremsen es zu einer Kollisions kommt, die am nominellen Haltepunkt dann schon wieder vorbei wäre. Das ist praktisch unerwünscht, vor allen Dingen aber entsteht dadurch ein (rechtliches) Risiko, das man als Hersteller nicht eingehen möchte.
Viele Ansätze gehen das Problem zu hoher Rechenzeit an. Werden Roboter und Umgebung in n Teile zerlegt, müssen im Prinzip n^2 Paare auf Kollision getestet werden. Mehrere Ansätze [8]bauen eine Hierarchie von Volumina, so dass, wenn ein paar von Obervolumina eine bestimmte Distanz hat, alle ihre Paare von Untervolumina mindestens die selbe Distanz haben. Diese Strategie funktioniert gut, ist aber nicht ganz einfach zu implementieren. Ausserdem hängt die Rechenzeit von der Stellung des Roboters ab. Das ist für eine Robotersteuerung, die in einem festen Zeittakt laufen muss problematisch. Erfinderischer Beitrag II liefert eine einfachere Alternative, die die Rechenzeit konstant hält. Es ist inherent, dass manche Stellungen des Roboters kompliziertere Rechnungen erfordern, als andere. Dadurch, dass der Algorithmus Ergebnisse von einem Zyklus zum nächsten ubertragen kann, kann er "vorarbeiten". Das heisst, er fängt schon an zu rechnen bevor der Roboter so eine problematische Stellung erreicht und kann somit die Rechenlast pro Schritt beschränken.
Zur Personensicherung vor automatischen Fahrzeugen werden sogenannte Laserscanner verwendet, die die Umgebung auf Hindernisse abtasten und das Fahrzeug anhalten, sobald sich ein Hindernis im sogenannten Schutzfeld befindet. Gegenwärtig wird eine feste Anzahl Schutzfelder vorkonfiguriert und das passende je nach Geschwindigkeit ausgewählt. Dadurch sind die Schutzfelder unnötig gross, weil ein Schutzfeld ja für alle Geschwindigkeiten und Lenkwinkel bei denen es aktiv ist gross genug sein muss. Die Erfindung, besonders erfinderischer Beitrag III+IV ermöglicht, zur Laufzeit aufgrund der aktuellen Geschwindigkeit Schutzfelder auszurechnen.

c) Welche Nachteile besitzen die bekannten Lösungen? siehe b)

d) Welche Aufgabe liegt der Erfindung zugrunde? Ein oder mehrere Roboterarme oder Fahrzeuge rechtzeitig zu stoppen, bevor sie mit der Umgebung oder miteinander kollidieren. Im erfinderischen Beitrag IV ausserdem, rechtzeitig vor einer Kollision mit einer Person zu stoppen.

e) Wie wird diese Aufgabe durch die Erfindung gelöst? siehe Beschreibung am Anfang

f) Worin ist das wesentlich Neue der erfindungsgemäßen Lösung zu sehen?
Erfinderischer Beitrag I: Die Berechnung der Bremsvolumina, also der vom Roboter beim Bremsen überstrichenen Region im Raum in Echtzeit, sicher (d.h. verwendete Approximationen vergrößern die Region höchstens, verkleinern sie nie), und in einer Darstellung die Distanzberechnung in Echtzeit erlaubt.
Erfinderischer Beitrag II: Die Beschleunigung der Berechnung der Distanzen auf eine *feste* Rechenzeit erlaubt das Verfahren auf einem kleinen Rechner, in der Regel dem

bestehenden Rechner der Robotersteuerung, mitlaufen zu lassen. Insbesondere wird die Rechenzeit auf einen festen Wert beschränkt, was nötig ist, da Robotersteuerungen in einem festen Taktzyklus laufen müssen.
Erfinderischer Beitrag III: Erweiterung von I auf Fahrzeuge. Insbesondere funktioniert die dort gegebene Formel ohne Sonderfälle sowohl für Kurven, also auch Geradeausfahrt.
Erfinderischer Beitrag IV: Umrechnung eines Bremsvolumens in ein Schutzfeld eines Laserscanners in Echtzeit. Dadurch können die Beiträge I-III nicht nur benutzt werden, um Roboter vor Kollision mit Robotern zu schützen, sondern auch um Personen vor der Kollision mit Robotern zu schützen.

g) Welche Vorteile werden durch die Erfinderung erzielt? Die Erfindung ermöglicht, Kollisionen von Roboterarmen zu vermeiden. Erinderischer Beitrag I berechnet Im Vergleich zu bekannten Lösungen dabei den Bremsweg des Roboterarmes präzise mit ein. Dadurch kann mit mittleren und hohen Geschwindigkeiten gefahren werden, das System wird sicher vor Kollisionen geschützt und es werden andererseits nicht so konservative Sicherheitszuschläge angewendet, das das System immer abschaltet.
Erfinderischer Beitrag II sorgt dafür, dass die Rechnung schnell durchgeführt werden kann. Die benötigte Rechenzeit lässt sich fest vorgeben. Schafft der Algorithmus es nicht, in dieser Zeit nachzurechnen, dass keine Kollision entsteht, so hält er an. Dabei skaliert der Algorithmus mit der Geschwindigkeit des Roboters, also unnötiges Anhalten kann flexibel durch langsameres Fahren oder mehr Rechenzeit umgangen werden. Der größte Vorteil durch die fixe Rechenzeit ist, dass das Verfahren auf einem kleinen Rechner in Echtzeit laufen kann und man sicher ist, dass es in Echtzeit läuft.
Erfinderischer Beitrag III und IV erlauben gegenüber einem Laserscanner mit festen Schutzfelder geringere und trotzdem noch sichere Sicherheitsabstände, so dass schneller und effektiver gefahren werden kann.

h) Welche Vorteile at die Erfindung in wirtschaftlicher Hinsicht? Der Endanwender der Roberarme vermeidet Kollisionsunfälle, die den Roboter beschädigen und dadurch Kosten und Ausfälle verursachen. Der Hersteller einer Robotersteuerung (typischerweise der Roboterhersteller) kann dadurch den Kundennutzen seiner Roboter erhöhen. Der erste erfinderische Beitrag erlaubt dies auch bei schnellen Bewegungen sicher zu tun. Der zweite erfinderische Beitrag verringert die benötigte Rechenzeit und ermöglicht so, das System in eine bestehende Steuerung zu integrieren, ohne das ein schnellerer Rechner nötig würde. Erfinderischer Beitrag III und IV ermöglichen Fahrzeugen schnelleres Fahren und verringern den Konfigurationsaufwand eines Laserscanner Schutzsystems im Vergleich zum üblichen manuellen Konfigurieren von Schutzfeldern.

Literatur

[1] U. Frese and H. Täubig, "Verfahren zur Vermeidung von Kollisionen gesteuert beweglicher Teile einer Anlage," Patentanmeldung beim Deutschen Patentamt unter 102009006256.4-32, 2009, (eingereicht).

[2] E. Gilbert, D. Johnson, and S. Keerthi, "A fast procedure for computing the distance between complex objects in 3d space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, 1988.

[3] C. Ericson, "The Gilbert-Johnson-Keerthi (GJK) algorithm," in *SICGRAPH Conference Plenary Talk*, 2004.

[4] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction into Robotic Manipulation*. CRC; 1 edition (March 22, 1994), 2006.

[5] A. Fuhrmann and E. Schömer, "A general method for computing the reachable space of mechanisms," in *Proceedings of the ASME Design Engineering Technical Conference 2001*, 2001.

[6] A. Fuhrmann *et al.*, "Verfahren zur kollisionsprüfung mechanischer konstruktionen," DE10131241B4, 2003.

[7] W. El-Houssaine, "Method and control device for avoiding collisions between cooperating robots," US#6678582, 2004.

[8] M. Nakano *et al.*, "Apparatus for detecting the collision of moving objects," U.S. Patent #5056031, 1991.

Verfahren zur Vermeidung von Kollisionen gesteuert beweglicher Teile einer Anlage

Udo Frese und Holger Täubig

RR-09-01
Research Report