

RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN  
INSTITUT FÜR INFORMATIK I



---

Natalia Tanatarova

Kritische Orientierungen in  
der Bewegungsplanung  
konvexer Roboter

16. September 2005

---

Diplomarbeit

Betreuer: Prof. Dr. Rolf Klein



# Inhaltsverzeichnis

<b>1</b>	<b>Problembeschreibung</b>	<b>1</b>
1.1	Einleitung . . . . .	1
1.2	Problembeschreibung . . . . .	2
<b>2</b>	<b>Funktionen</b>	<b>7</b>
2.1	Fälle und Schritte . . . . .	7
2.2	Fall 1: Zwei Hinderniskanten . . . . .	8
2.2.1	Halbebenentest . . . . .	8
2.2.2	Winkeltest . . . . .	8
2.2.3	Kantentest . . . . .	11
2.3	Fall 2: Zwei Hindernisecken . . . . .	12
2.3.1	Halbebenentest und Abstandstest . . . . .	12
2.3.2	Winkeltest . . . . .	13
2.3.3	Kantentest . . . . .	15
2.4	Fall 3: Verschiedene Kontaktpaartypen . . . . .	16
2.4.1	Halbebenentest und Abstandstest . . . . .	16
2.4.2	Winkeltest . . . . .	16
2.4.3	Kantentest ( $h > r$ ) . . . . .	18
2.4.4	Kantentest mit $h < r$ . . . . .	23
<b>3</b>	<b>Eckpunkte der unteren Kontur</b>	<b>31</b>
3.1	Zusammenführung von $A_L(O_i)$ und $A_R(O_i)$ . . . . .	31
3.2	A-Funktion und B-Funktion . . . . .	32
3.3	A-Funktion . . . . .	32
3.4	B-Funktion . . . . .	36
3.4.1	Ableitung und Umkehrfunktion . . . . .	36
3.4.2	Schnittpunkte $F_{B1}(x) \cap F_{B2}(x)$ und $F_B(x) \cap F_A(x)$ . . . . .	37
<b>4</b>	<b>Zyklische Tripel</b>	<b>41</b>
4.1	Algorithmus von Kedem und Sharir . . . . .	41
4.2	Plazierungen von einem Zyklus . . . . .	42
4.2.1	Drei Roboterecken . . . . .	43
4.2.2	Drei Roboterkannten . . . . .	46

4.2.3	Verschiedene Kontaktpaartypen . . . . .	47
<b>5</b>	<b>Sonderfälle</b>	<b>51</b>
5.1	Parallele Kanten . . . . .	51
5.2	Ecke/Ecke-Kontaktpaare . . . . .	52
5.3	Nicht kritische Plazierungen . . . . .	54
<b>6</b>	<b>Das Java-Applet</b>	<b>55</b>
6.1	Das Benutzer-Interface . . . . .	55
6.2	Die Programmstruktur . . . . .	57
<b>7</b>	<b>Komplexität und Schlußbemerkung</b>	<b>59</b>
7.1	Komplexität . . . . .	59
7.2	Schlußbemerkung . . . . .	59

# Abbildungsverzeichnis

1.1	Bewegungsplanung . . . . .	2
1.2	Plazierung $(x,y,\theta)$ . . . . .	2
1.3	Kontaktpaar $O=(W,S)$ . . . . .	3
1.4	Kritische Plazierungen . . . . .	3
1.5	$O_2/O_1$ . . . . .	5
1.6	$f_{O_1O_2}$ . . . . .	5
2.1	Testschritte . . . . .	7
2.2	Fall 1. Halbebenentest . . . . .	8
2.3	Fall 1. $S, \alpha, d, \beta_1, \beta_2$ . . . . .	9
2.4	Fall 1. Winkel $\phi$ . . . . .	9
2.5	Fall 1. Einschränkungen für $\beta_2$ . . . . .	10
2.6	Fall 1. Winkelbedingung für $O_2/O_1$ . . . . .	10
2.7	Fall 2. Halbebenentest . . . . .	12
2.8	Fall 2. $S, \alpha, d, x_i, u_i, v_i$ und $\beta_i$ . . . . .	13
2.9	Fall 2. $h_{ij}$ . . . . .	13
2.10	Fall 2. Einschränkungen für $\alpha$ und $\beta_i$ . . . . .	14
2.11	Fall 2. Winkelbedingung für $O_2/O_1$ . . . . .	15
2.12	Fall 3. Halbebenentest und Abstandstest . . . . .	16
2.13	Fall 3. $\alpha_L, \alpha_R$ und $\alpha'_R$ . . . . .	16
2.14	Fall 3. Einschränkungen aus den Roboterwinkeln . . . . .	17
2.15	Fall 3. Winkel zwischen Hindernissen . . . . .	17
2.16	Fall 3. Winkelbedingung für $O_2/O_1$ . . . . .	18
2.17	Fall 3. $h, r, M_1$ und $M_2$ . . . . .	19
2.18	Fall 3. $h > r.  x_1M_1  \rightarrow \alpha$ . . . . .	20
2.19	Fall 3. $h > r. p_{11}, p_{12} \rightarrow \alpha_L, \alpha_R$ . . . . .	24
2.20	Fall 3. $h > r.  x_2M_2  \rightarrow \alpha$ . . . . .	25
2.21	Fall 3. $h > r. p_{21}, p_{22} \rightarrow \alpha_L, \alpha_R$ . . . . .	26
2.22	Fall 3. $h > r.$ Minimum für $ M_2p_{21} ,  M_2p_{22} $ . . . . .	27
2.23	Fall 3. $h > r. d_{M_i}(p)$ . . . . .	27
2.24	Fall 3. $h < r.  M_1x_1  \rightarrow \alpha$ . . . . .	28
2.25	Fall 3. $h < r.$ Minimum für $ M_1p_{11} ,  M_1p_{12} $ . . . . .	28
2.26	Fall 3. $h < r. p_{11}, p_{12} \rightarrow \alpha_L, \alpha_R$ . . . . .	29

2.27	Fall 3. $h < r$ . $p_{21}, p_{22} \rightarrow \alpha_L, \alpha_R$ . . . . .	30
3.1	A-Funktion . . . . .	33
3.2	B-Funktion . . . . .	34
4.1	Suche nach einem Zyklus . . . . .	42
4.2	Verschiedene Fälle bei einem Zyklus . . . . .	43
4.3	Drei Roboterecken . . . . .	43
4.4	Fall 1.1 . . . . .	44
4.5	Fall 1.2 . . . . .	46
4.6	Drei Roboterkanten . . . . .	47
4.7	Zwei Roboterecken und eine Roboterkante . . . . .	48
4.8	Zwei Roboterkanten und eine Roboterecke . . . . .	49
5.1	Sonderfälle . . . . .	51
5.2	Fall 1. $S_2$ ist eine Roboterecke . . . . .	52
5.3	Fall 1. Abstandstest . . . . .	53
5.4	Fall 2. Abstandstest . . . . .	53
5.5	Nicht kritische Plazierungen . . . . .	54
6.1	Applet . . . . .	56
6.2	Die Programmstruktur . . . . .	58

# Kapitel 1

## Problembeschreibung

### 1.1 Einleitung

Das allgemeine Problem der Bewegungsplanung für Roboter ist sehr komplex. Für spezielle Probleme aus diesem Bereich gibt es viele bekannte Algorithmen. Man betrachtet z.B. einen punkt- oder kreisförmigen Roboter in der Ebene mit polygonalen Hindernissen, oder löst das Problem der Bewegungsplanung für ein Liniensegment [10]. Für einen konvexen Roboter mit reinen Translationsbewegungen gibt es das Roadmap-Verfahren [7]. Für einen beliebigen Roboter kann man in  $O(mn\alpha(mn)\log^2(mn))$  Zeit eine kollisionsfreie Translation von  $s$  nach  $t$  bestimmen oder feststellen, dass keine solche existiert [11][4]. Je allgemeiner das Problem wird, desto komplexere Algorithmen werden dafür entwickelt.

Kedem und Sharir [5][6] beschreiben einen Bewegungsplanungsalgorithmus für einen konvexen Roboter in der Ebene mit polygonalen Hindernissen, wobei Rotation- und Translationsbewegungen erlaubt sind (siehe auch [4]). Bei diesem Algorithmus handelt es sich um eine Highlevel-Beschreibung. Für eine konkrete Implementierung werden viele spezielle Fragen nur angedeutet. Es wird z.B. behauptet, dass bestimmte Funktionen nicht mehr als vier Nullstellen haben können, aber es ist überhaupt nicht klar, wie man diese Funktionen (und eine geeignete Parametrisierung) findet. Die Lösung solcher Probleme stellt sich als technisch schwierig heraus. In dieser Arbeit wird ein Teil dieses Algorithmus praktisch umgesetzt und die Lösung in einem Java-Applet animiert dargestellt.

In diesem Kapitel wird das Problem genau beschrieben. In Kapitel 2 werden bestimmte Funktionen aufgestellt, die eine große Rolle bei der Berechnung der kritischen Plazierungen spielen. In weiteren drei Kapiteln (3 bis 5) werden verschiedene Typen von kritischen Plazierungen behandelt. Danach wird kurz das Java-Applet beschrieben, sein Benutzer-Interface und die Programmstruktur (Kapitel 6). Und im letzten Kapitel befinden sich die Komplexitätsangaben und die Schlußbemerkung.

## 1.2 Problembeschreibung

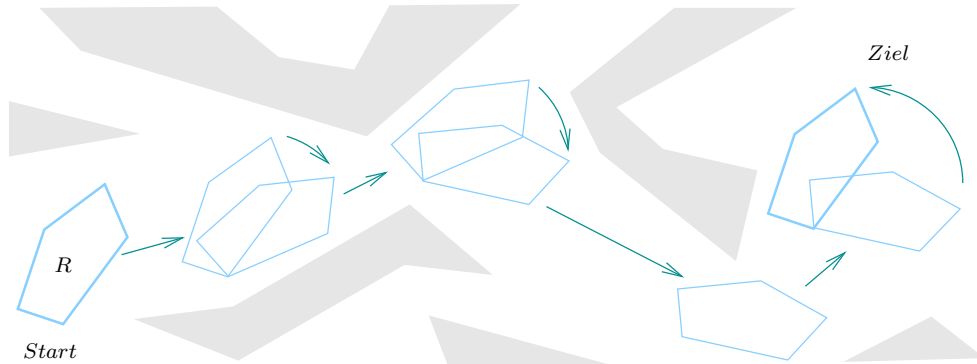


Abbildung 1.1: Bewegungsplanung

Sei Roboter  $R$  ein konvexes Polygon mit  $m$  Ecken, und seine Umgebung eine Ebene mit (nicht unbedingt konvexen) Polygonen  $P_i$ , die insgesamt  $n$  Ecken haben.

Gesucht wird ein Bewegungsplan aus Translation- und Rotationsbewegungen, die den Roboter aus der Startposition in die Zielposition überführen, so dass er auf dem Weg keine Hindernisse schneidet (siehe Abbildung 1.1).

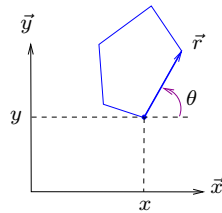


Abbildung 1.2: Plazierung  $(x,y,\theta)$

### Definition 1

Eine *Plazierung*  $(x,y,\theta)$  von  $R$  ist definiert durch die Koordinaten  $x,y$  von seinem Referenzpunkt und die *Orientierung* - den Winkel  $\theta$  zwischen der  $x$ -Achse und dem Referenzvektor  $\vec{r}$  (siehe Abbildung 1.2).

### Definition 2

Eine *halbfreie Plazierung* ist eine Plazierung, bei der Roboter die Hindernisse berühren, aber nicht schneiden darf.

Es ist viel einfacher einen Weg zu planen, wenn der Roboter sich nicht drehen kann. Für eine feste Orientierung  $\theta$  besteht dann die Menge aller



halbfreien Plazierungen aus Polygonen, in denen man den Weg nur für den Referenzpunkt berechnet.

Deswegen versucht man bei der allgemeineren Bewegungsplanung eine endliche Anzahl der Orientierungen zu betrachten. Wenn man aber falsche Winkel nimmt, findet man möglicherweise nicht alle Wege: es kann sein, dass der Roboter nur mit einer bestimmten Orientierung an einer engen Stelle zwischen zwei Hindernissen vorbeigehen kann. Wenn man diese Orientierung nicht beachtet, findet man diesen Weg nicht.

Der Algorithmus von Kedem und Sharir [5] verwendet für die Bewegungsplanung nur Orientierungen mit bestimmten Eigenschaften. Ein paar Definitionen dazu:

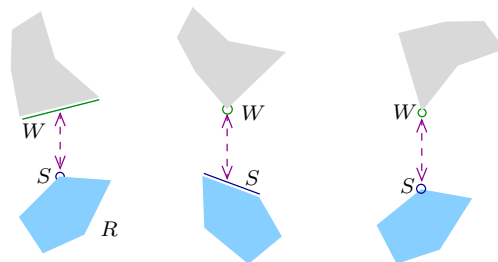


Abbildung 1.3: Kontakt paar  $O=(W,S)$

### Definition 3

Ein *Kontakt paar*  $O = (W,S)$  besteht entweder

- aus einer Roboterecke (S) und einer Hinderniskante (W) oder
- aus einer Roboterseite (S) und einer Hindernisecke (W) oder
- aus einer Roboterecke (S) und einer Hindernisecke (W)

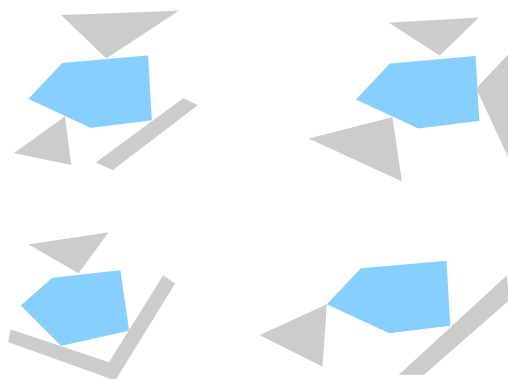


Abbildung 1.4: Kritische Plazierungen

**Definition 4**

Eine halbfreie Platzierung heißt *kritische Platzierung*, wenn entweder

- drei verschiedene Ecke/Kante-Kontaktpaare oder
- ein Ecke/Ecke-Kontaktpaar und ein Ecke/Kante-Kontaktpaar

beteiligt sind.

Um diese Platzierungen geht es weiter in dieser Arbeit. Kedem und Sharir betrachten in [5] noch weitere spezielle Orientierungen, die hier nicht beschrieben werden.

Kritische Platzierungen mit einem Ecke/Ecke-Kontaktpaar werden in Kapitel 5 behandelt. In den Kapiteln 1 (ab jetzt) bis 4 werden nur Ecke/Kante-Kontaktpaare betrachtet und einfach “Kontaktpaar” genannt.

Folgende Definitionen braucht man für die Berechnung dieser speziellen Platzierungen.

**Definition 5**

Eine *Tangente*  $T$  eines Kontaktpaars ist die Gerade, auf der die beteiligte Kante liegt.

**Definition 6**

Seien  $O_1 = (S_1, W_1)$  und  $O_2 = (S_2, W_2)$  zwei Kontaktpaare,  $T_1$  und  $T_2$  ihre Tangenten, die sich im Punkt  $Z_{12}$  schneiden.  $O_2$  *beschränkt*  $O_1$  bei Winkel  $\theta$  (Bezeichnung  $O_2/O_1$ ), wenn

- eine Platzierung  $Z=(x,y,\theta)$  von  $R$  mit  $O_1$  und  $O_2$  existiert und
- die Konvexe Hülle von  $S_1 \cup S_2$  stets  $W_2$  schneidet, wenn  $R$  bei fester Orientierung  $\theta$  entlang  $T_1$  in Richtung  $Z_{12}$  verschoben wird, bis zu der letzten Platzierung, bei der  $S_1$  und  $W_1$  sich berühren, siehe Abbildung 1.5.

Aus der Konvexität des Roboters folgt, dass bei einer Platzierung mit zwei Kontaktpaaren  $O_1$  und  $O_2$  bei Winkel  $\theta$  gilt: entweder  $O_1/O_2$  oder  $O_2/O_1$  oder beides (oder  $T_1||T_2$ ). Also kann es zwei Typen von kritischen Platzierungen mit drei Kontaktpaaren (mit nicht parallelen Tangenten) geben:

- Zwei beteiligte Kontaktpaare beschränken das dritte.
- Die drei Kontaktpaare beschränken sich in einem Zyklus (z.B.  $O_1/O_2, O_2/O_3, O_3/O_1$ ).

Beide Typen von kritischen Platzierungen lassen sich mit Hilfe der folgenden speziellen Funktionen finden.

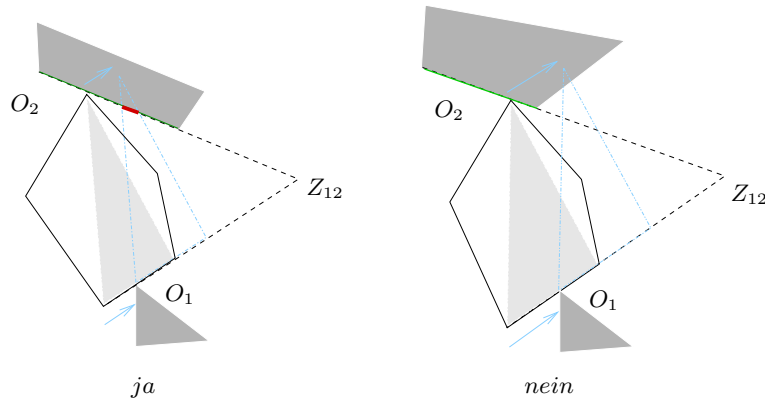


Abbildung 1.5:  $O_2/O_1$

**Definition 7**

Seien  $O_1 = (S_1, W_1)$  und  $O_2 = (S_2, W_2)$  zwei Kontaktpaare und sei  $\prod_{O_1O_2} = \{\theta \mid O_2/O_1 \text{ bei } \theta\}$  nicht leer.

Dann wird Funktion  $f_{O_1O_2} : \prod_{O_1O_2} \rightarrow R^+$  folgendermaßen definiert:

$$f_{O_1O_2}(\theta) = \begin{cases} |x_1u_1| & , \text{ falls } S_1 \text{ Roboterseite} \\ |v_1x_1| & , \text{ falls } S_1 \text{ Roboterecke,} \end{cases}$$

wobei  $x_1$  der Berührungspunkt von  $S_1$  und  $W_1$ ,  
 $u_1$  der Endpunkt der  $O_1$ -Kante in Richtung  $Z_{12}$  und  
 $v_1$  der andere Endpunkt der  $O_1$ -Kante ist (siehe Abbildung 1.6).

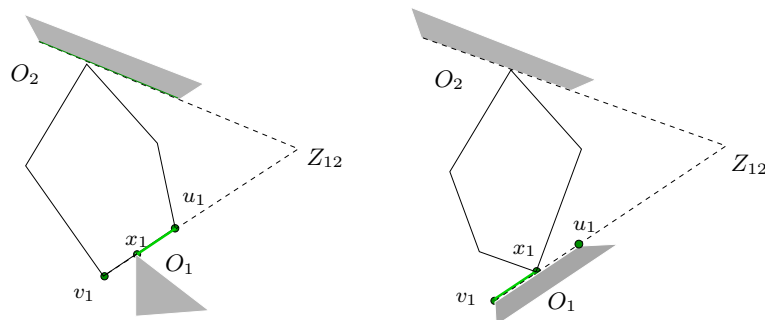


Abbildung 1.6:  $f_{O_1O_2}$

Der Funktionswert gibt also die Distanz an, um die der Roboter bei konstantem  $\theta$  und unter Aufrechterhaltung des Kontaktes  $O_1$  geschoben werden kann, ohne dass ein verbotener Schnitt entsteht. Mit einer größeren Distanz bei gleichem  $\theta$  gibt es keine halbfreie Platzierung. Deshalb sind alle halbfreien Platzierungen mit Kontaktpaar  $O_1$  auf der unteren Kontur der entsprechenden Menge von Funktionen zu finden. Aber umgekehrt gilt nicht: nicht alle

Plazierungen, die auf der unteren Kontur zu finden sind, sind wirklich halbfrei. Es kann sein, dass ein anderes Hindernis, das keine Funktion in dem Arrangement hat, einen verbotenen Schnitt verursacht. Wir berechnen also zuerst aus den unteren Konturen die “Kandidaten” und danach werden die nicht halbfreie Plazierungen aussortiert.

Die Plazierungen, bei denen die Tangenten von Kontaktpaaren parallel sind, werden in Kapitel 5 behandelt.

**Definition 8**

Für ein Kontaktpaar  $O_i$  ist  $A(O_i)$  das Arrangement der Funktionsgraphen  $f_{O_i O_k}$  mit  $O_k/O_i$ .

Für ein Kontaktpaar  $O_i$  kann es verschiedene andere Kontaktpaare geben, die es beschränken. Dabei muss nicht immer der gleiche Endpunkt der  $O_1$ -Kante beteiligt sein (der Schnittpunkt der Tangenten kann links oder rechts liegen). Deshalb berechnet man zwei Arrangements  $-A_L(O_i)$  und  $A_R(O_i)$ , eins für jede Seite. Die untere Kontur wird zuerst getrennt in  $A_L(O_i)$  und  $A_R(O_i)$  bestimmt, und dann werden beide Konturen zusammengeführt.

Insgesamt sind für die Berechnung der kritischen Plazierungen also folgende Schritte notwendig:

- finde, welche Kontaktpaare sich beschränken,
- speichere die Funktionen  $f_{O_i O_j}$ , bilde die Arrangements  $A_L(O_i), A_R(O_i)$ ,
- berechne die untere Konturen in  $A_L(O_i), A_R(O_i)$   
→ kritische Plazierungen von Typ 1,
- finde kritische Plazierungen von Typ 2 (mit einem Zyklus),
- finde die Plazierungen mit einem Ecke/Ecke-Kontaktpaar,
- finde die Plazierungen mit parallelen Tangenten,
- sortiere die nicht halbfreien Plazierungen aus.

Schon die ersten zwei Schritte erweisen sich als nicht einfach. Darum geht es im nächsten Kapitel.

# Kapitel 2

## Funktionen

### 2.1 Fälle und Schritte

Um die Arrangements  $A(O_i)$  für jedes Kontaktpaar  $O_i$  zu bekommen muss man für jedes Paar von Kontaktpaaren  $O_1$  und  $O_2$  überprüfen, ob sie einander beschränken, und falls ja, die entsprechenden Funktionen aufstellen.

Die erste Voraussetzung für  $O_2/O_1$  ist die Existenz einer Platzierung mit  $O_1$  und  $O_2$ . Das kann man in folgenden Schritten überprüfen:

- *Halbebenentest*. Ist die Hinderniskante/Ecke von der anderen Hinderniskante/Ecke aus sichtbar?
- *Abstandstest*. Sind die Hindernisse zu weit auseinander?
- *Winkeltest*. Hier werden Winkel zwischen den Kanten berechnet. Es kann z.B. sein, dass ein Roboterwinkel nicht in den Hinderniswinkel passt.
- *Kantentest*. Der letzte Test. Hier wird überprüft wie die Kantenendpunkte liegen.

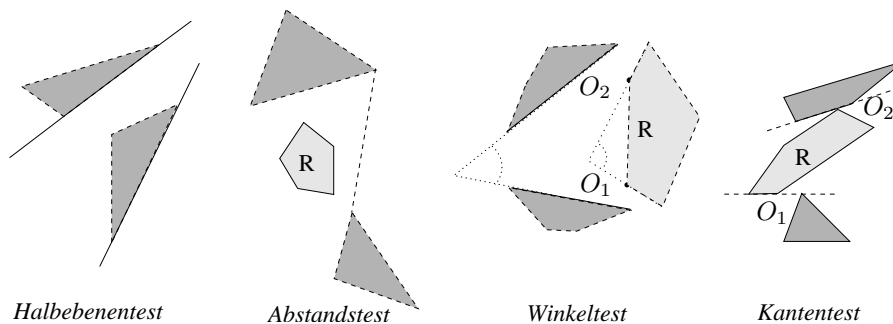


Abbildung 2.1: Testschritte

Wenn es eine Plazierung mit  $O_1$  und  $O_2$  gibt, also die erste Voraussetzung erfüllt ist, muss man nur noch eine Winkelbedingung überprüfen, um zu wissen, ob  $O_1$   $O_2$  beschränkt oder nicht. Weil es zwei Typen von Kontaktpaaren gibt, muss man sich mit mehreren Fällen beschäftigen. Sei  $O_1 = (S_1, W_1)$  und  $O_2 = (S_2, W_2)$ . Dann gibt es diese 3 Fälle:

1.  $W_1$  und  $W_2$  sind Hinderniskanten ( $S_1$  und  $S_2$  sind Roboterecken)
2.  $W_1$  und  $W_2$  sind Hindernisecken ( $S_1$  und  $S_2$  sind Roboteranten)
3.  $W_1$  ist eine Hinderniskante und  $W_2$  ist eine Hindernisecke oder umgekehrt

## 2.2 Fall 1: Zwei Hinderniskanten

### 2.2.1 Halbebenentest

Seien  $O_1$  und  $O_2$  zwei verschiedene Kontaktpaare mit Roboterecken ( $S_1$  und  $S_2$ ) und Hinderniskanten ( $W_1$  und  $W_2$ ). Mindestens ein Endpunkt der einen Hinderniskante muss von der anderen Hinderniskante aus sichtbar sein, d.h., wenn eine Kante komplett in der falschen Halbebene liegt, kann es keine Plazierung mit  $O_1$  und  $O_2$  geben.

Wenn ein Endpunkt in der falschen Halbebene liegt, kann man ihn für die weiteren Tests verschieben, ohne Plazierungen zu verlieren. Beides folgt aus der Konvexität des Roboters. (siehe Abbildung 2.2)

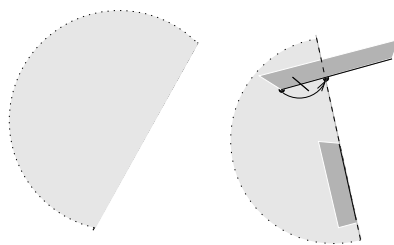


Abbildung 2.2: Fall 1. Halbebenentest

### 2.2.2 Winkeltest

Sei  $S$  der Schnittpunkt von  $T_1$  und  $T_2$ ,  $\alpha = \angle(T_1, T_2)$  der Winkel zwischen  $T_1$  und  $T_2$ ,  $x_1$  und  $x_2$  die Berührungspunkte bei einer Plazierung mit  $O_1$  und  $O_2$ , wenn sie existiert,  $\beta_1 = \angle(\overrightarrow{x_1 x_2}, \overrightarrow{x_1 S})$  und  $\beta_2 = \angle(\overrightarrow{x_2 x_1}, \overrightarrow{x_2 S})$  (siehe Abbildung 2.3).

Aus den Einschränkungen für  $\beta_1$  oder  $\beta_2$  kann man später Einschränkungen für  $\theta$  bekommen.

Seien  $u_1, u_2$  die Kantenendpunkte in Richtung S und  $v_1, v_2$  die Kantenendpunkte in die andere Richtung. Die Funktion  $f_{O_1O_2}$  ist in diesem Fall gleich  $|x_1v_1|$ , weil  $S_1$  eine Roboterecke ist. Also ist

$$f_{O_1O_2} = |Sv_1| - |Sx_1| = |Sv_1| - d \cdot \sin(\beta_2)/\sin(\alpha),$$

wobei  $d$  der Abstand zwischen  $x_1$  und  $x_2$  ist. Weil  $f_{O_1O_2}$  von  $\beta_2$  abhängt, interessieren uns die Intervalle von  $\beta_2$ .

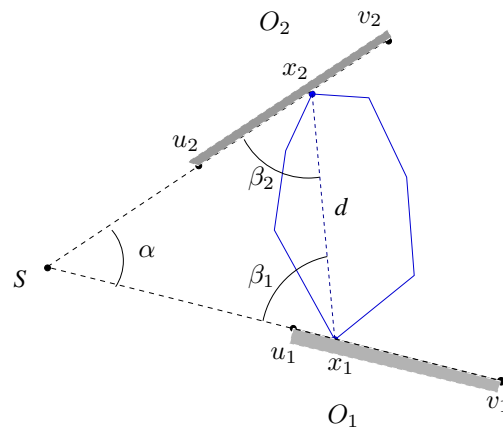


Abbildung 2.3: Fall 1. S,  $\alpha$ , d,  $\beta_1, \beta_2$

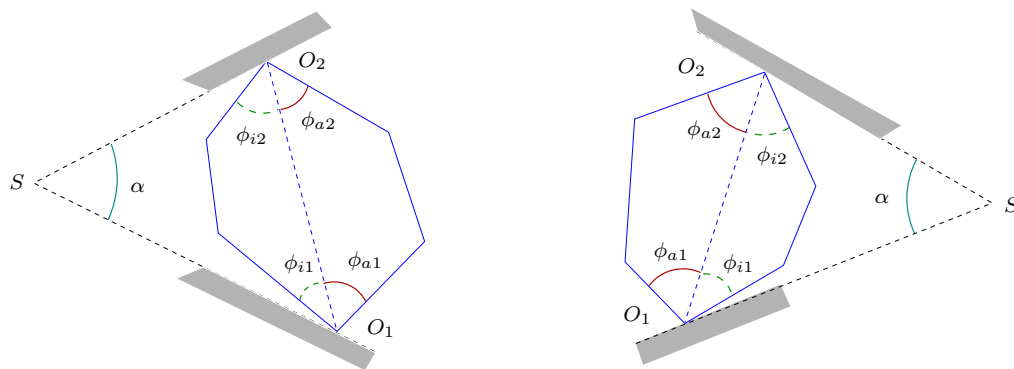
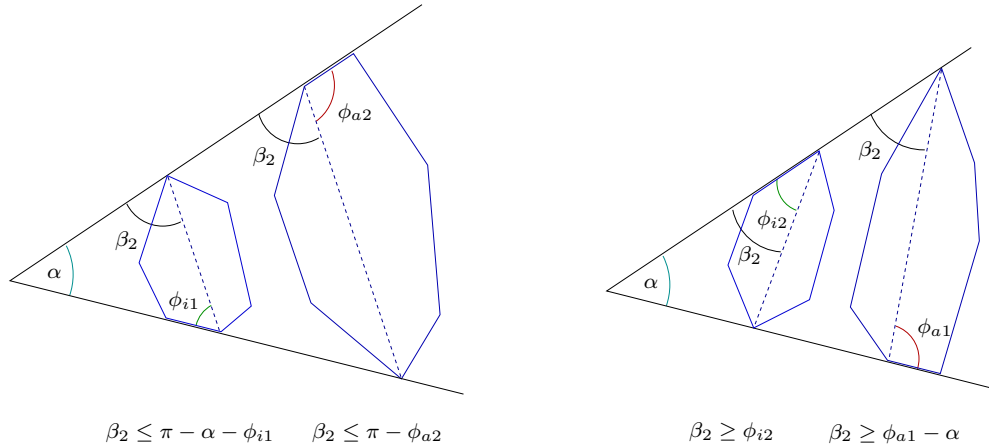
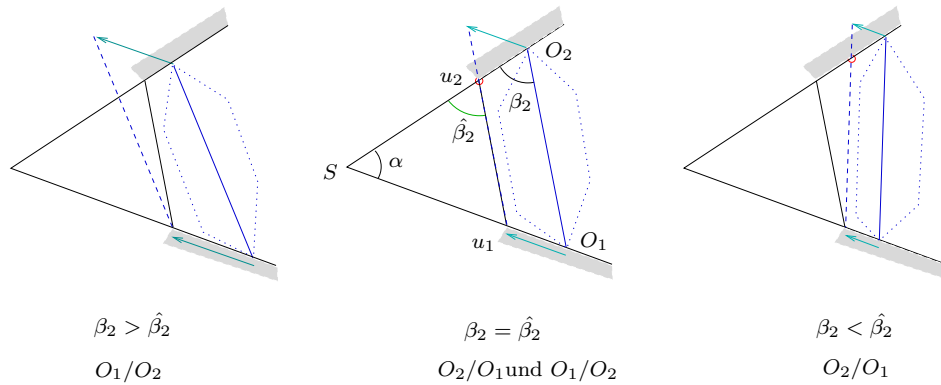


Abbildung 2.4: Fall 1. Winkel  $\phi$

Welche Seite von R wird bei einer Platzierung mit  $O_1$  und  $O_2$  aus S sichtbar? Um einfacher über die Orientierungen zu sprechen habe ich  $O_1$  immer unten gezeichnet und  $O_2$  immer oben. Nachdem man herausgefunden hat, ob S links oder rechts liegt ( $v_1 \rightarrow S \rightarrow v_2$  in oder gegen den Uhrzeigersinn),

Abbildung 2.5: Fall 1. Einschränkungen für  $\beta_2$ Abbildung 2.6: Fall 1. Winkelbedingung für  $O_2/O_1$ 

kann man innere und äußere Winkel  $\phi_{i1}$ ,  $\phi_{i2}$ ,  $\phi_{a1}$ ,  $\phi_{a2}$  im Roboter berechnen (siehe Abbildung 2.4). Bei einer halbfreien Plazierung muss gelten:

$$\phi_{i1} + \phi_{i2} + \alpha \leq \pi, \quad \phi_{a1} + \phi_{a2} - \alpha \leq \pi.$$

Außerdem gilt für  $\beta_2$  (siehe Abbildung 2.5):

$$\max(\phi_{i2}, \phi_{a1} - \alpha) \leq \beta_2 \leq \min(\pi - \phi_{a2}, \pi - \alpha - \phi_{i1}) \quad (2.1)$$

Sei  $\hat{\beta}_2$  der Winkel zwischen  $u_2S$  und  $u_2u_1$  (siehe Abbildung 2.6). Dann gilt:

$$O_2/O_1 \Leftrightarrow \beta_2 \leq \hat{\beta}_2 \quad (2.2)$$

Sei  $B_w$  das Intervall für  $\beta_2$  aus den Bedingungen 2.1 und 2.2. Wenn  $B_w$  nicht leer ist, wird es bei den weiteren Tests verwendet und eventuell



verkleinert, so dass am Ende eine Intervallmenge entsteht, die den gültigen Plazierungen mit  $O_1$  und  $O_2$  entspricht, bei denen  $O_2/O_1$  gilt.

### 2.2.3 Kantentest

Wir haben noch nicht berücksichtigt, dass  $x_1$  zwischen  $u_1$  und  $v_1$  liegen muss und  $x_2$  zwischen  $u_2$  und  $v_2$ . Sei  $k = d/\sin(\alpha)$ . Dann gilt:

$$\begin{aligned} |u_1S| &\leq |x_1S| \leq |v_1S| \\ &\Downarrow \\ |u_1S| &\leq k \cdot \sin \beta_2 \leq |v_1S| \\ &\Downarrow \\ \frac{|u_1S|}{k} &\leq \sin \beta_2 \leq \frac{|v_1S|}{k} \end{aligned}$$

Genauso erhält man die Bedingung  $\frac{|u_2S|}{k} \leq \sin \beta_1 \leq \frac{|v_2S|}{k}$ . Wenn  $|u_1S| > k$  oder  $|u_2S| > k$  kann es keine Platzierung geben. Sonst gibt es entweder ein oder zwei Intervalle für  $\beta_2$ , abhängig davon ob  $|v_iS| > k$  oder nicht. Sei  $\beta_{u_1} = \arcsin \frac{|u_1S|}{k}$  und entsprechend  $\beta_{u_2}$ ,  $\beta_{v_1}$  und  $\beta_{v_2}$ . Dann gilt:

$$\beta_2 \in B_{x_1} := \begin{cases} [\beta_{u_1}, \pi - \beta_{u_1}], & \text{falls } |v_1S| \geq k \\ [\beta_{u_1}, \beta_{v_1}] \cup [\pi - \beta_{v_1}, \pi - \beta_{u_1}], & \text{falls } |v_1S| < k \end{cases} \quad (2.3)$$

und

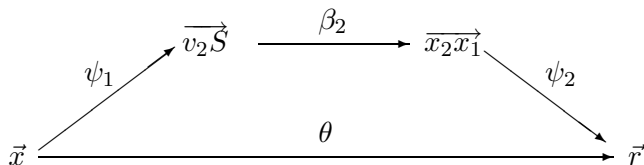
$$\beta_1 \in \begin{cases} [\beta_{u_2}, \pi - \beta_{u_2}], & \text{falls } |v_2S| \geq k \\ [\beta_{u_2}, \beta_{v_2}] \cup [\pi - \beta_{v_2}, \pi - \beta_{u_2}], & \text{falls } |v_2S| < k \end{cases}$$

$\Downarrow$

$$\beta_2 \in B_{x_2} := \begin{cases} [\beta_{u_2} - \alpha, \pi - \alpha - \beta_{u_2}], & \text{falls } |v_2S| \geq k \\ [\beta_{u_2} - \alpha, \beta_{v_2} - \alpha] \cup [\pi - \alpha - \beta_{v_2}, \pi - \alpha - \beta_{u_2}], & \text{falls } |v_2S| < k \end{cases} \quad (2.4)$$

Wenn  $B := B_w \cap B_{x_1} \cap B_{x_2} \neq \emptyset$ , gibt es Platzierung(en) mit  $O_2/O_1$  und  $B$  enthält Intervalle von  $\beta_2$ , auf denen  $f_{O_1O_2}$  definiert ist. Daraus muss man noch Intervalle für  $\theta$  herleiten. Dazu definieren wir zwei Hilfswinkel  $\psi_1 = \angle(\vec{x}, \overrightarrow{v_2S})$  und  $\psi_2 = \angle(\overrightarrow{x_2x_1}, \vec{r})$ , wobei  $\vec{x}$  die x-Achse ist,  $\vec{r}$  die Referenzkante von  $R$ ,  $\overrightarrow{x_2x_1}$  der Vektor in  $R$  zwischen den beiden beteiligten Ecken, und alle Winkel gegen den Uhrzeigersinn gerichtet sind.

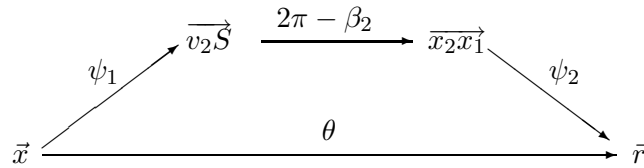
Wenn  $S$  links liegt, ist  $\beta_2 = \angle(\overrightarrow{v_2S}, \overrightarrow{x_2x_1}) \Rightarrow \theta = \psi_1 + \beta_2 + \psi_2$ :



Also müssen wir zu den Endpunkten von  $\beta_2$ -Intervallen  $\psi_1 + \psi_2$  addieren, um  $\theta$ -Intervalle zu erhalten (eventuell muss man noch dafür sorgen, dass Intervalle innerhalb von  $[0, 2\pi]$  liegen). Die Funktion sieht dann so aus:

$$f_{O_1 O_2}(\theta) = |v_1 S| - k \cdot \sin(\theta - \psi_1 - \psi_2)$$

Wenn aber S rechts liegt, ist  $\beta_2 = \angle(\overrightarrow{x_2 x_1}, \overrightarrow{v_2 S}) \Rightarrow 2\pi - \beta_2 = \angle(\overrightarrow{v_2 S}, \overrightarrow{x_2 x_1})$ :



↓

$$\theta = \psi_1 + 2\pi - \beta_2 + \psi_2$$

Also werden Intervalle nach dem Schema  $[a, b] \rightarrow [-b + \psi_1 + \psi_2, -a + \psi_1 + \psi_2]$  umgewandelt und die Funktion sieht so aus:

$$f_{O_1 O_2}(\theta) = |v_1 S| - k \cdot \sin(-\theta + \psi_1 + \psi_2)$$

## 2.3 Fall 2: Zwei Hindernisecken

### 2.3.1 Halbebenentest und Abstandstest

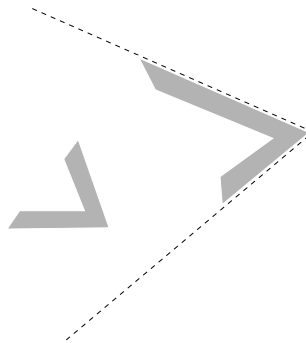


Abbildung 2.7: Fall 2. Halbebenentest

Seien  $O_1$  und  $O_2$  zwei Kontaktpaare mit Roboterkannten und konvexen Hindernisecken. Die Hindernisecken müssen füreinander sichtbar sein. D.h.,

Fälle wie in Abbildung 2.7 werden ausgeschlossen. Außerdem kann man einen Abstandstest machen: wenn der Abstand zwischen den beiden Hindernisecken größer ist als der Roboterdurchmesser (größter Eckenabstand), kann es auch keine Plazierung geben.

### 2.3.2 Winkeltest

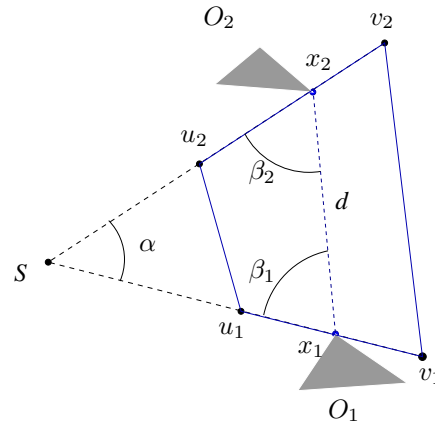


Abbildung 2.8: Fall 2.  $S$ ,  $\alpha$ ,  $d$ ,  $x_i$ ,  $u_i$ ,  $v_i$  und  $\beta_i$

Wir definieren  $S$ ,  $\alpha$ ,  $d$ ,  $x_i$ ,  $u_i$ ,  $v_i$  und  $\beta_i$  wie im Fall 1, obwohl die Kanten jetzt dem Roboter gehören und Ecken den Hindernissen.  $f_{O_1O_2}$  ist in diesem Fall gleich  $|u_1x_1|$ , weil  $S_1$  eine Roboterkante ist, aber sonst sieht sie fast genauso aus wie im Fall 1 und hängt wieder von  $\beta_2$  ab:

$$f_{O_1O_2} = |u_1x_1| = |Sx_1| - |Su_1| = -|Su_1| + d \cdot \sin(\beta_2) / \sin(\alpha)$$

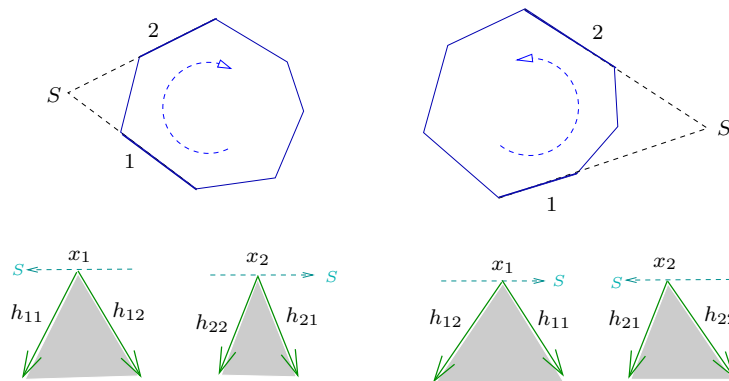


Abbildung 2.9: Fall 2.  $h_{ij}$

Hier kann  $S$  auch links oder rechts liegen. Dementsprechend definieren wir  $\vec{h}_{11}$  als den Vektor, der aus  $x_1$  in die links oder rechts liegende Nachbar-

ecke (auf dem gleichen Hindernis) zeigt,  $\overrightarrow{h_{12}}$  soll in die andere Nachbarecke zeigen, und genauso  $\overrightarrow{h_{21}}$  und  $\overrightarrow{h_{22}}$  aus  $x_2$  (siehe Abbildung 2.9).

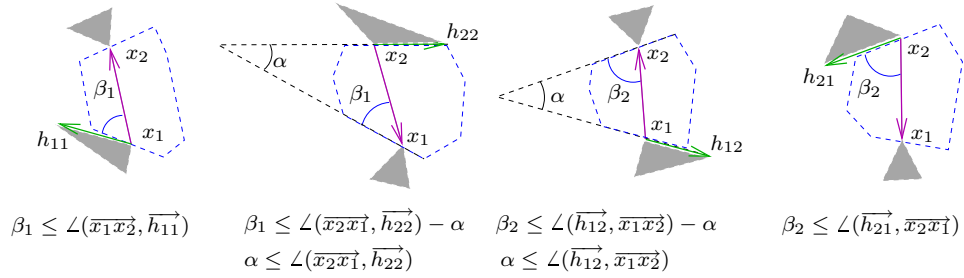


Abbildung 2.10: Fall 2. Einschränkungen für  $\alpha$  und  $\beta_i$

Die Winkel in folgenden Ungleichungen sind für den Fall mit S links. Wenn S rechts ist, müssen sie entweder in andere Richtung (im Uhrzeigersinn) berechnet werden oder die Argumente müssen vertauscht werden. Bei einer Platzierung mit  $O_1$  und  $O_2$  gilt:

$$\alpha \leq \angle(\overrightarrow{x_2x_1}, \overrightarrow{h_{22}})$$

$$\alpha \leq \angle(\overrightarrow{h_{12}}, \overrightarrow{x_1x_2})$$

Außerdem gibt es folgende Einschränkungen für  $\beta_1$  und  $\beta_2$  (siehe Abbildung 2.10):

$$\beta_1 \leq \beta_{1max} := \min(\angle(\overrightarrow{x_1x_2}, \overrightarrow{h_{11}}), \angle(\overrightarrow{x_2x_1}, \overrightarrow{h_{22}}) - \alpha)$$

$$\beta_2 \leq \beta_{2max} := \min(\angle(\overrightarrow{h_{21}}, \overrightarrow{x_2x_1}), \angle(\overrightarrow{h_{12}}, \overrightarrow{x_1x_2}) - \alpha)$$

↓

$$\pi - \alpha - \beta_{1max} \leq \beta_2 \leq \beta_{2max} \quad (2.5)$$

und

$$\beta_{1max} + \beta_{2max} + \alpha \geq \pi$$

Die Winkelbedingung für  $O_2/O_1$  ist ähnlich wie im Fall 1:

$$O_2/O_1 \Leftrightarrow \beta_2 \geq \hat{\beta}_2, \quad (2.6)$$

wobei  $\hat{\beta}_2 = \angle(\overrightarrow{v_2S}, \overrightarrow{v_2v_1})$  (siehe Abbildung 2.11).

Sei  $B_w$  das Intervall für  $\beta_2$  aus den Bedingungen 2.5 und 2.6. Wenn  $B_w \neq \emptyset$ , wird es beim Kantentest berücksichtigt.

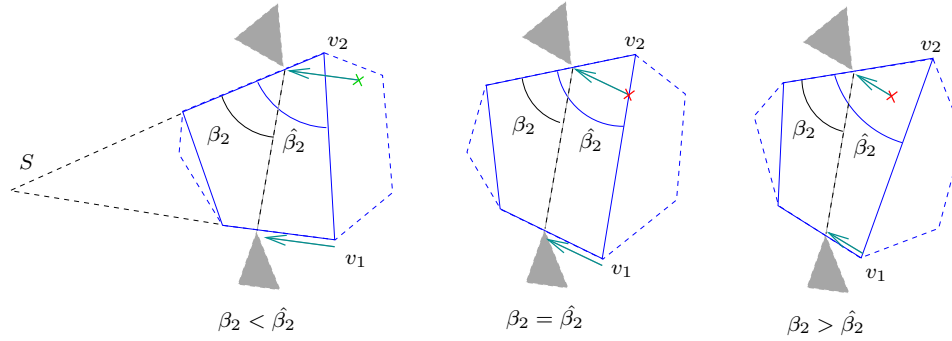


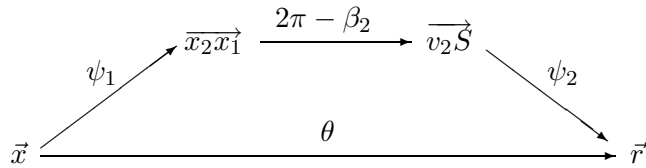
Abbildung 2.11: Fall 2. Winkelbedingung für  $O_2/O_1$

### 2.3.3 Kantentest

Die Bedingung, dass  $x_1$  zwischen  $u_1$  und  $v_1$  liegt und  $x_2$  zwischen  $u_2$  und  $v_2$  kann man genauso beschreiben wie im Fall 1. Es gibt keine Platzierung mit  $O_1$  und  $O_2$ , wenn  $|u_1S| > k := d/\sin \alpha$  oder  $|u_2S| > k$ . Und genauso findet man Winkelintervalle  $B_{x_1}$  und  $B_{x_2}$  für  $\beta_2$  (2.3 und 2.4). Wenn  $B := B_w \cap B_{x_1} \cap B_{x_2} \neq \emptyset$ , gibt es Platzierung(en) mit  $O_2/O_1$  und B enthält Intervalle von  $\beta_2$ , auf denen  $f_{O_1O_2}$  definiert ist.

Jetzt müssen wir wieder aus B Intervalle für  $\theta$  herleiten. Sei  $\psi_1 = \angle(\vec{x}, \overrightarrow{x_2x_1})$  und  $\psi_2 = \angle(\overrightarrow{v_2S}, \vec{r})$ .

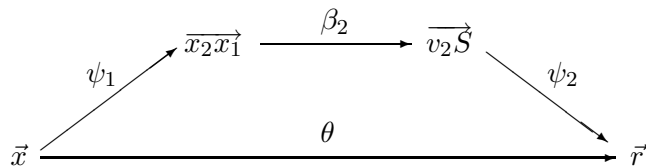
Wenn S links liegt, ist  $\beta_2 = \angle(\overrightarrow{v_2S}, \overrightarrow{x_2x_1}) \Rightarrow \theta = \psi_1 + 2\pi - \beta_2 + \psi_2$ :



Also werden Intervalle nach dem Schema  $[a,b] \rightarrow [-b + \psi_1 + \psi_2, -a + \psi_1 + \psi_2]$  umgewandelt und die Funktion sieht so aus:

$$f_{O_1O_2}(\theta) = -|Su_1| + k \cdot \sin(-\theta + \psi_1 + \psi_2)$$

Wenn S rechts liegt, ist  $\beta_2 = \angle(\overrightarrow{x_2x_1}, \overrightarrow{v_2S}) \Rightarrow \theta = \psi_1 + \beta_2 + \psi_2$ :



Also muss man zu den Endpunkten von  $\beta_2$ -Intervallen  $\psi_1 + \psi_2$  addieren,

und die Funktion sieht so aus:

$$f_{O_1O_2}(\theta) = -|Su_1| + k \cdot \sin(\theta - \psi_1 - \psi_2)$$

## 2.4 Fall 3: Verschiedene Kontaktpaararten

### 2.4.1 Halbebenentest und Abstandstest

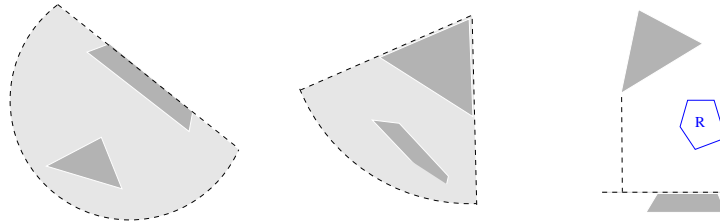


Abbildung 2.12: Fall 3. Halbebenentest und Abstandstest

Sei  $O_1$  ein Kontaktpaar mit Hinderniskante  $e_1$  und  $O_2$  ein Kontaktpaar mit Hindernisecke  $p_2$ . Erstens, muss  $p_2$  in der richtigen Halbebene von  $e_1$  liegen, und zweitens, mindestens ein Endpunkt von  $e_1$  muss aus  $p_2$  sichtbar sein. Außerdem kann man testen, ob der Abstand zwischen  $p_2$  und der Geraden, die  $e_1$  enthält, nicht größer ist als der Roboterdurchmesser (siehe Abbildung 2.12).

### 2.4.2 Winkeltest

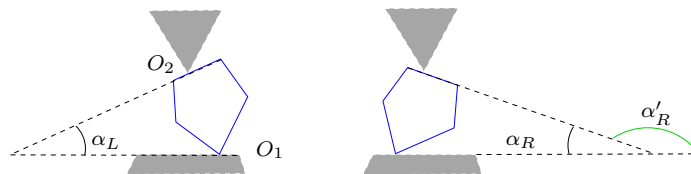


Abbildung 2.13: Fall 3.  $\alpha_L$ ,  $\alpha_R$  und  $\alpha'_R$

Hier ist der Winkel zwischen  $T_1$  und  $T_2$  abhängig von der Orientierung  $\theta$  und nicht mehr fest wie in den ersten zwei Fällen. Schnittpunkt  $S$  kann bei einer Orientierung links und bei einer anderen rechts liegen. Dementsprechend werden für Winkelbedingungen  $\alpha_L$  und  $\alpha_R$  benutzt. Aber für die Funktionen  $f_{O_1O_2}$  verwende ich statt  $\alpha_R$  den Winkel  $\alpha'_R = \pi - \alpha_R$  (siehe Abbildung 2.13).

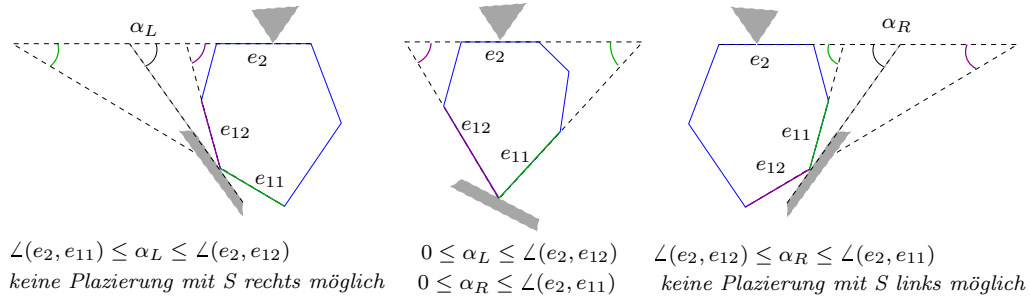


Abbildung 2.14: Fall 3. Einschränkungen aus den Roboterwinkeln

Zuerst betrachten wir die Einschränkungen für  $\alpha_L$  und  $\alpha_R$ , die aus den Roboterwinkeln folgen. Sei  $e_2$  die  $O_2$ -Kante von R und  $e_{11}$  und  $e_{12}$  zwei andere Kanten von R, die adjazent zur  $O_1$ -Ecke sind und seien  $l_2$ ,  $l_{11}$  und  $l_{12}$  die Geraden, auf denen diese Kanten liegen. Abhängig davon, ob ein Schnittpunkt von  $l_2$  und  $l_{1i}$  links oder rechts von  $e_2$  liegt, gibt es verschiedene Fälle (siehe Abbildung 2.14). Es kann auch sein, dass Kanten parallel sind. Wenn  $e_2 \parallel e_{11}$  bzw.  $e_2 \parallel e_{12}$ , dann ist keine Platzierung mit S rechts bzw. links möglich und die untere Schranke für  $\alpha_L$  bzw.  $\alpha_R$  ist 0.

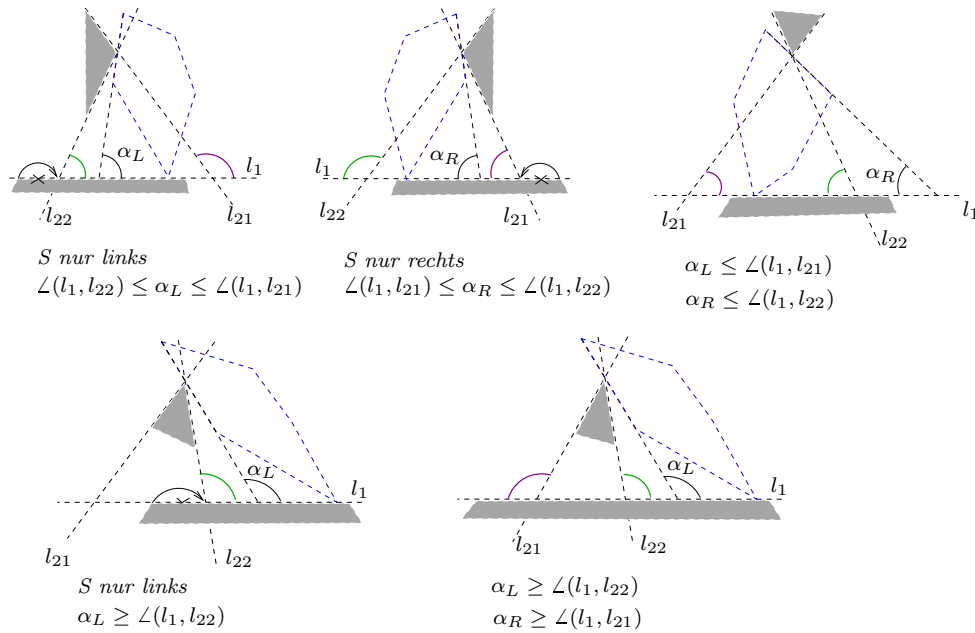


Abbildung 2.15: Fall 3. Winkel zwischen Hindernissen

Hindernisse können  $\alpha_L$  und  $\alpha_R$  auch stark einschränken. Hier gibt es

auch verschiedene Fälle, je nachdem, wie die Schnittpunkte der Geraden  $l_1$   $l_{21}$  und  $l_2$   $l_{21}$  liegen (siehe Abbildung 2.15). Wenn  $l_1 \parallel l_{2i}$ , dann ist die untere Schranke gleich 0, oder die obere gleich  $\pi$  oder es sind keine Plazierungen mit S links bzw. rechts möglich. An dieser Stelle wird auch ein Endpunkt von  $e_1$  verschoben, falls er aus  $x_2$  nicht sichtbar ist.

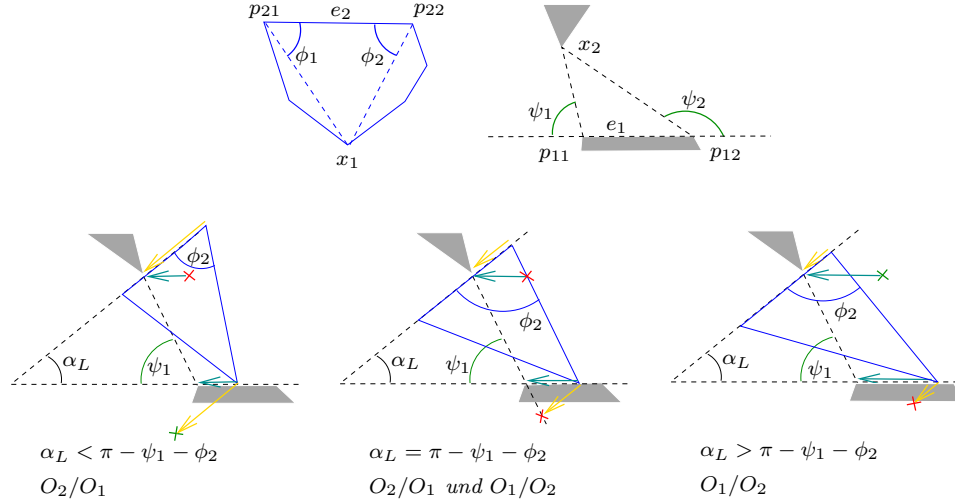


Abbildung 2.16: Fall 3. Winkelbedingung für  $O_2/O_1$

Als letztes überprüfe ich, ob  $O_2/O_1$  oder umgekehrt ( $O_1/O_2$ ). Sei  $\phi_1 = \angle(\overrightarrow{p_{21}x_1}, \overrightarrow{e_2})$ ,  $\phi_2 = \angle(-\overrightarrow{e_2}, \overrightarrow{p_{22}x_1})$ ,  $\psi_1 = \angle(\overrightarrow{p_{11}x_2}, -\overrightarrow{e_1})$  und  $\psi_2 = \angle(\overrightarrow{e_1}, \overrightarrow{p_{12}x_2})$ , wobei  $e_i = (p_{i1}, p_{i2})$  die Kante von  $O_i$  ist. (siehe Abbildung 2.16). Dann gilt:

$$O_2/O_1 \Leftrightarrow \begin{cases} \alpha_L \leq \pi - \psi_1 - \phi_2 & , \text{ falls S links} \\ \alpha_R \leq \pi - \psi_2 - \phi_1 & , \text{ falls S rechts} \end{cases} \quad (2.7)$$

und

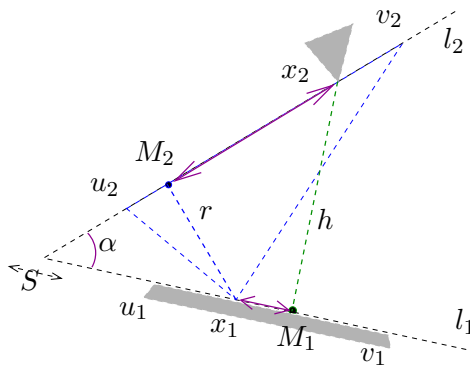
$$O_1/O_2 \Leftrightarrow \begin{cases} \alpha_L \geq \pi - \psi_1 - \phi_2 & , \text{ falls S links} \\ \alpha_R \geq \pi - \psi_2 - \phi_1 & , \text{ falls S rechts} \end{cases} \quad (2.8)$$

Aus allen Winkelbedingungen zusammen bekommt man Intervalle für  $\alpha_L$  und  $\alpha_R$ . Wenn sie nicht leer sind (mindestens eins) muss man noch den Kantentest machen.

### 2.4.3 Kantentest ( $h > r$ )

Sei  $h$  der Abstand zwischen  $x_2$  und  $l_1$  (Hindernisse),  $M_1$  der Punkt auf  $l_1$  mit  $|x_2M_1| = h$ ,  $r$  der Abstand zwischen  $x_1$  und  $l_2$  (im Roboter) und  $M_2$  der Punkt auf  $l_2$  mit  $|x_1M_2| = r$  (siehe Abbildung 2.17). Für die Bedingung, dass  $x_1$  zwischen  $u_1$  und  $v_1$  liegt, braucht man einen Referenzpunkt auf  $l_1$ . Weil S sich in diesem Fall abhängig von der Orientierung verschiebt, benutze ich  $M_1$  auf  $l_1$  und  $M_2$  auf  $l_2$ . Für die Funktionen  $f_{O_1O_2}$  und  $f_{O_2O_1}$



Abbildung 2.17: Fall 3.  $h$ ,  $r$ ,  $M_1$  und  $M_2$ 

muss man den Zusammenhang zwischen der Orientierung ( $\alpha$ ) und  $|M_i x_i|$  in beide Richtungen herausfinden:

- $(\theta \rightarrow) \quad \alpha \rightarrow |M_1 x_1| \quad (\rightarrow |x_1 v_1|)$  für  $f_{O_1 O_2}(\theta)$ ,
- $(\theta \rightarrow) \quad \alpha \rightarrow |M_2 x_2| \quad (\rightarrow |x_1 u_1|)$  für  $f_{O_2 O_1}(\theta)$ ,
- $|M_1 x_1| \rightarrow \alpha$  für Definitionsbereich von  $f_{O_1 O_2}$  und  $f_{O_2 O_1}$   
(  $x_1$  zwischen  $u_1$  und  $v_1 \Rightarrow \alpha \in ? \quad (\Rightarrow \theta \in ?)$  ),
- $|M_2 x_2| \rightarrow \alpha$  für Definitionsbereich von  $f_{O_1 O_2}$  und  $f_{O_2 O_1}$   
(  $x_2$  zwischen  $u_2$  und  $v_2 \Rightarrow \alpha \in ? \quad (\Rightarrow \theta \in ?)$  ).

Betrachten wir zuerst die Zusammenhänge  $|M_i x_i| \rightarrow \alpha$  und die Einschränkungen für  $\alpha_L$  und  $\alpha_R$ , die aus "  $x_i$  zwischen  $v_i$  und  $u_i$  " folgen ( Kantentest ). Sei zuerst  $h > r$ . Für einen Wert  $x = |x_1 M_1|$  gibt es vier mögliche  $\alpha(x)$ , je nachdem, ob  $x_1$  links oder rechts von  $M_1$  liegt und ob  $S$  links oder rechts von  $x_1$  liegt (siehe Abbildung 2.18). Ich nenne sie  $\alpha_1^1, \alpha_2^1, \alpha_3^1$  und  $\alpha_4^1$ . Sei  $\beta_1 = \angle(x_2 S, x_2 x_1)$  und  $\beta_2 = \angle(x_2 x_1, x_2 M_1)$ . Dann ist

$$\cos(\alpha_i) = \pm \sin(\pm \beta_1 \pm \beta_2) = \pm \sin \beta_1 \cos \beta_2 \pm \cos \beta_1 \sin \beta_2$$

Sei

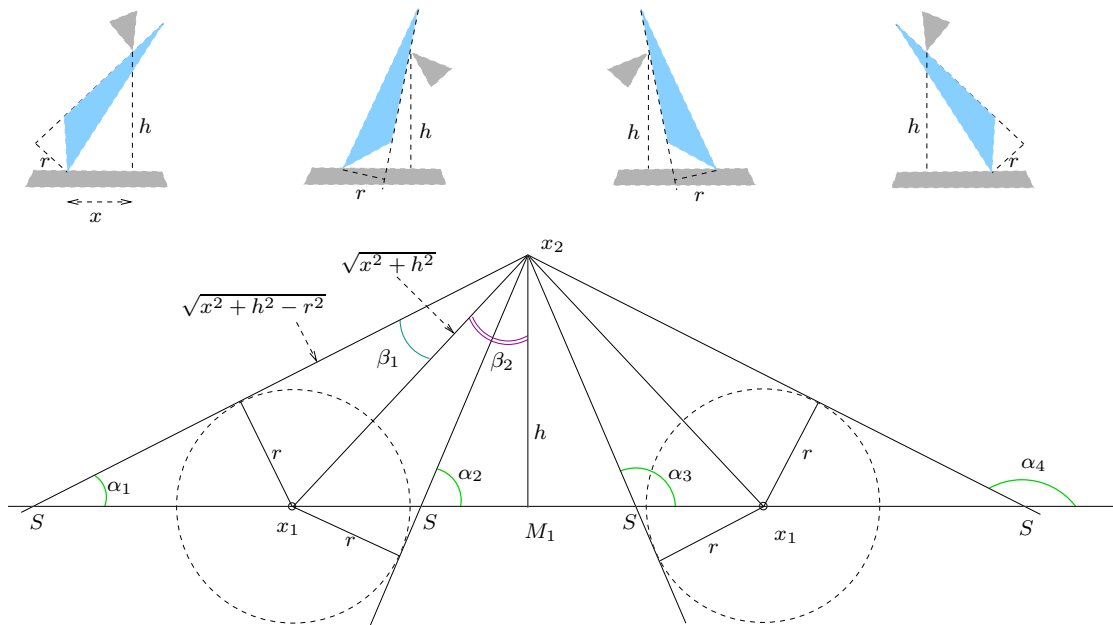
$$f(h, r, x) = \cos \beta_1 \sin \beta_2 = \frac{x \sqrt{x^2 + h^2 - r^2}}{x^2 + h^2}$$

und

$$g(h, r, x) = \sin \beta_1 \cos \beta_2 = \frac{hr}{x^2 + h^2}.$$

Dann gilt:

- $\alpha_1^1(x) = \arccos(\sin(\beta_1 + \beta_2)) = \arccos(f(h, r, x) + g(h, r, x))$
- $\alpha_2^1(x) = \arccos(\sin(-\beta_1 + \beta_2)) = \arccos(f(h, r, x) - g(h, r, x))$

Abbildung 2.18: Fall 3.  $h > r$ .  $|x_1M_1| \rightarrow \alpha$ 

- $\alpha_3^1(x) = \arccos(-\sin(-\beta_1 + \beta_2)) = \arccos(-f(h, r, x) + g(h, r, x))$
- $\alpha_4^1(x) = \arccos(-\sin(\beta_1 + \beta_2)) = \arccos(-f(h, r, x) - g(h, r, x))$

Daraus kann man jetzt die Bedingung “ $x_1$  zwischen  $v_1$  und  $u_1$ ” konstruieren.  $u_1$  ist nach der Definition der Endpunkt von der  $O_1$ -Kante in Richtung S (und  $v_1$  der andere Endpunkt). Weil hier diese Richtung von der Orientierung abhängt, werden für den Kantentest  $p_{11}$  (der linke Endpunkt) und  $p_{12}$  (der rechte Endpunkt) verwendet. Sei  $d_1 = |M_1p_{11}|$  und  $d_2 = |M_1p_{12}|$ . Wenn  $p_{11}$  links von  $M_1$  liegt, muss  $\alpha_L$  größer sein als  $\alpha_1^1(d_1)$ , weil sonst der Kontakt von  $O_1$  verloren geht. Wenn  $p_{11}$  rechts von  $M_1$  liegt, kommt in die Bedingung  $\alpha_3^1$  und so weiter (siehe Abbildung 2.19).

Für ein  $x = |x_2M_2|$  gibt es auch verschiedene mögliche  $\alpha(x)$  (siehe Abbildung 2.20). Weil hier  $x$  eine andere Größe bezeichnet, tauschen  $r$  und  $h$  die Plätze in den Formeln, die sonst gleich sind:

$$\cos \beta_1 \sin \beta_2 = \frac{x\sqrt{x^2 + r^2 - h^2}}{x^2 + r^2} = f(r, h, x)$$

$$\sin \beta_1 \cos \beta_2 = \frac{hr}{x^2 + r^2} = g(r, h, x)$$

↓

- $\alpha_1^2(x) = \arccos(\sin(\beta_1 + \beta_2)) = \arccos(f(r, h, x) + g(r, h, x))$
- $\alpha_2^2(x) = \arccos(\sin(-\beta_1 + \beta_2)) = \arccos(f(r, h, x) - g(r, h, x))$
- $\alpha_3^2(x) = \arccos(-\sin(-\beta_1 + \beta_2)) = \arccos(-f(r, h, x) + g(r, h, x))$
- $\alpha_4^2(x) = \arccos(-\sin(\beta_1 + \beta_2)) = \arccos(-f(r, h, x) - g(r, h, x))$

Daraus erhalten wir die Bedingung “ $x_2$  zwischen  $v_2$  und  $u_2$ ”. Seien  $p_{21}$  und  $p_{22}$  die Endpunkte von  $e_2$ , und sei  $d_1 = |M_2 p_{21}|$  und  $d_2 = |M_2 p_{22}|$ , dann sind  $\alpha_L$  und  $\alpha_R$  so eingeschränkt, wie in Abbildung 2.21 steht.

Noch eine Einschränkung ergibt sich aus der Tatsache, dass  $f(r, h, x)$  für  $x < \sqrt{h^2 - r^2} =: x_{min}$  nicht definiert ist. Für diese  $x$  gibt es keine Plazierungen mit  $O_1$  und  $O_2$ , weil der Abstand zwischen den beiden Hindernissen größer ist als der Abstand zwischen den beiden Kontaktpunkten auf  $R$ . Also muss man zuerst prüfen, ob  $d_1 \geq x_{min}$  und  $d_2 \geq x_{min}$  gilt. Wenn ja, können die Winkel  $\alpha_i^2(x)$  berechnet werden. Wenn nicht, sind Plazierungen nur von einer Seite (S nur links oder nur rechts) möglich oder man verwendet die Winkel  $\alpha_i^2(x_{min})$  - je nachdem, ob  $p_{21}$  und  $p_{22}$  links oder rechts von  $M_2$  liegen (siehe Abbildung 2.22).

Wenn  $\alpha_L$  oder  $\alpha_R$  alle diese Tests bestanden haben, d.h., wenn die Definitionsbereiche von  $f_{O_1 O_2}$  und  $f_{O_2 O_1}$  nicht leer sind, gibt es Platzierung(en) mit  $O_1$  und  $O_2$ . Also kann man die erhaltenen  $\alpha$ -Winkelintervalle in  $\theta$ -Intervalle transformieren und die Funktionen speichern. Weil  $S$ ,  $M_1$ ,  $x_1$ ,  $p_{11}$  und  $p_{12}$  in fast beliebiger Reihenfolge auf  $l_1$  liegen können und in jedem Fall  $|x_1 v_1|$  anders aus  $|M_1 p_{11}|$ ,  $|M_1 p_{12}|$   $|M_1 x_1|$  zusammengesetzt wird, führe ich  $d_{M_1}(p)$  ein - die Position von  $p$  auf  $l_1$  relativ zu  $M_1$ :

$$d_{M_1}(p) = \begin{cases} |M_1 p|, & \text{falls } p \text{ rechts von } \overrightarrow{M_1 x_2} \text{ liegt} \\ -|M_1 p|, & \text{sonst} \end{cases}$$

und sei

$$d_{M_2}(p) = \begin{cases} |M_2 p|, & \text{falls } p \text{ rechts von } \overrightarrow{x_1 M_2} \text{ liegt} \\ -|M_2 p|, & \text{sonst} \end{cases}$$

Dann ist z.B. wenn  $S$  links liegt,  $|x_1 v_1| = d_{M_1}(p_{12}) - d_{M_1}(x_1)$ , unabhängig davon, wo  $M_1$  liegt (siehe Abbildung 2.23). Die Funktionen sehen dann so aus:

$$f_{O_1 O_2} = |x_1 v_1| = \begin{cases} d_{M_1}(p_{12}) - d_{M_1}(x_1), & \text{falls } S \text{ links von } \overrightarrow{x_1 x_2} \\ -d_{M_1}(p_{11}) + d_{M_1}(x_1), & \text{sonst} \end{cases}$$

$$f_{O_2 O_1} = |x_2 u_2| = \begin{cases} -d_{M_2}(p_{21}) + d_{M_2}(x_2), & \text{falls } S \text{ links von } \overrightarrow{x_1 x_2} \\ d_{M_2}(p_{22}) - d_{M_2}(x_2), & \text{sonst} \end{cases}$$

Dabei sind  $d_{M_1}(p_{12})$  und  $d_{M_2}(p_{22})$  fest, aber  $d_{M_1}(x_1)$  und  $d_{M_2}(x_2)$  hängen von der Orientierung ab. Jetzt kommen wir zu den Zusammenhängen  $\alpha \rightarrow$

$|M_i x_i|$ . Wir betrachten wieder die gleichen Fälle wie in Abbildung 2.18 und noch zwei Fälle, wo  $M_1$  zwischen  $x_1$  und S liegt. Hier ist es besser,  $\alpha'_R$  statt  $\alpha_R$  zu betrachten, weil dann die Funktionsgleichungen einfacher sind.  $d_{M_1}(x_1)$  kann man als Summe oder Differenz von  $|SM_1|$  und  $|Sx_1|$  berechnen:

$$f_{\sin}(r, \alpha) := \frac{r}{\sin \alpha} = |Sx_1|, \quad f_{\tan}(h, \alpha) := \frac{h}{\tan \alpha} = \pm |SM_1|$$

↓

$$d_{M_1}(x_1) = \begin{cases} f_{\sin}(r, \alpha) - f_{\tan}(h, \alpha), & \text{falls S links von } \overline{x_1 x_2} \\ -f_{\sin}(r, \alpha) - f_{\tan}(h, \alpha), & \text{sonst} \end{cases}$$

↓

$$f_{O_1 O_2} = \begin{cases} d_{M_1}(p_{12}) - f_{\sin}(r, \alpha) + f_{\tan}(h, \alpha), & \text{falls S links von } \overline{x_1 x_2} \\ -d_{M_1}(p_{11}) - f_{\sin}(r, \alpha) - f_{\tan}(h, \alpha), & \text{sonst} \end{cases}$$

$d_{M_2}(x_2)$  kann man als Summe oder Differenz von  $|SM_2|$  und  $|Sx_2|$  berechnen:

$$f_{\sin}(h, \alpha) = \frac{h}{\sin \alpha} = |Sx_2|, \quad f_{\tan}(r, \alpha) = \frac{r}{\tan \alpha} = \pm |SM_2|$$

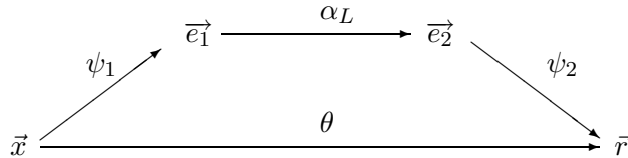
↓

$$d_{M_2}(x_2) = \begin{cases} f_{\sin}(h, \alpha) - f_{\tan}(r, \alpha), & \text{falls S links von } \overline{x_1 x_2} \\ -f_{\sin}(h, \alpha) - f_{\tan}(r, \alpha), & \text{sonst} \end{cases}$$

↓

$$f_{O_2 O_1} = \begin{cases} -d_{M_2}(p_{21}) + f_{\sin}(h, \alpha) - f_{\tan}(r, \alpha), & \text{falls S links von } \overline{x_1 x_2} \\ d_{M_2}(p_{22}) + f_{\sin}(h, \alpha) + f_{\tan}(r, \alpha), & \text{sonst} \end{cases}$$

Für den Fall  $h > r$  bleibt noch die Umwandlung von  $\alpha$ -Intervallen in die  $\theta$ -Intervalle. Sei  $\psi_1$  der Winkel (gegen den Uhrzeigersinn) zwischen der x-Achse und der Hinderniskante  $e_1$ , und  $\psi_2$  der Winkel zwischen der Roboterkante  $e_2$  und der Roboter-Referenzkante  $\vec{r}$ , dann ist



↓

$$\theta = \psi_1 + \alpha_L + \psi_2$$

$$\theta = \psi_1 + 2\pi - \alpha_R + \psi_2$$

$$\theta = \psi_1 + \pi + \alpha'_R + \psi_2$$

D.h., die Intervalle von  $\alpha_L$  werden um  $\psi_1 + \psi_2$  nach rechts verschoben und die Intervalle von  $\alpha_R$  gespiegelt und nach links verschoben. Sei  $\psi = \psi_1 + \psi_2$ . Dann sehen die Funktionen so aus:

$$f_{O_1O_2}(\theta) = \begin{cases} d_{M_1}(p_{12}) - \frac{r}{\sin(\theta-\psi)} + \frac{h}{\tan(\theta-\psi)}, & \text{falls S links von } \overrightarrow{x_1x_2} \\ -d_{M_1}(p_{11}) - \frac{r}{\sin(\theta-\psi-\pi)} - \frac{h}{\tan(\theta-\psi-\pi)}, & \text{sonst} \end{cases}$$

$$f_{O_2O_1}(\theta) = \begin{cases} -d_{M_2}(p_{21}) + \frac{h}{\sin(\theta-\psi)} - \frac{r}{\tan(\theta-\psi)}, & \text{falls S links von } \overrightarrow{x_1x_2} \\ d_{M_2}(p_{22}) + \frac{h}{\sin(\theta-\psi-\pi)} + \frac{r}{\tan(\theta-\psi-\pi)}, & \text{sonst} \end{cases}$$

Sie haben also die Form  $a + \frac{b}{\sin(\theta+d)} + \frac{c}{\tan(\theta+d)}$ .

#### 2.4.4 Kantentest mit $h < r$

Sei jetzt  $h < r$ . Für ein  $x = |M_1x_1|$  gibt es auch vier mögliche  $\alpha$ . Sie sind gleich den  $\alpha_1^1$  bis  $\alpha_4^1$ , aber die dazugehörige Plazierungen sehen anders aus. Zwei davon sind auf Abbildung 2.24 zu sehen, die anderen zwei sind symmetrisch (mit  $x_1$  rechts von  $M_1$ ).

Dabei muss man wieder überprüfen, ob  $M_1x_1$  nicht ausserhalb des Definitionsbereichs von  $f(h, r, x)$  liegt. Wenn  $x < \sqrt{r^2 - h^2} =: x_{min}$ , dann ist der Abstand zwischen den Hindernissen kleiner als der Abstand zwischen den beiden Kontaktpunkten auf R. Wenn also  $M_1p_{11} < x_{min}$  oder  $M_1p_{12} < x_{min}$ , sind Plazierungen nur von einer Seite (S nur links oder nur rechts) möglich oder man verwendet die Winkel  $\alpha_i^2(x_{min})$  (siehe Abbildung 2.25).

Die Bedingungen “ $x_i$  zwischen  $u_i$  und  $v_i$ ” bestehen aus ähnlichen Einschränkungen für  $\alpha_L$  und  $\alpha_R$  wie im Fall  $h > r$ , siehe Abbildung 2.26 und Abbildung 2.27.

Nach diesen Tests stehen die Intervalle für  $\alpha_L$  und/oder  $\alpha_R$  fest, und wenn sie nicht leer sind, muss man die entsprechenden Funktionen speichern.  $f_{O_2O_1}$  und  $f_{O_1O_2}$  werden genauso berechnet wie im Fall  $h > r$ .

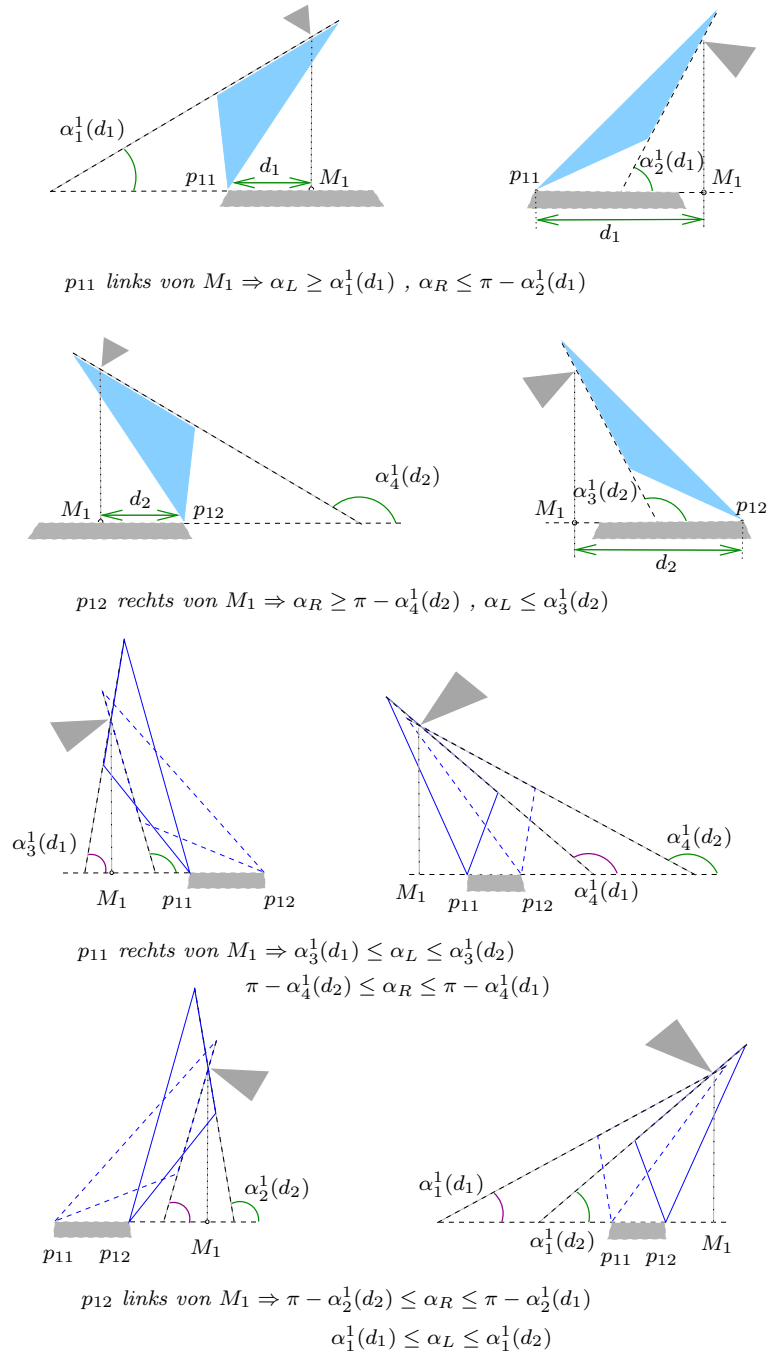


Abbildung 2.19: Fall 3.  $h > r$ .  $p_{11}, p_{12} \rightarrow \alpha_L, \alpha_R$

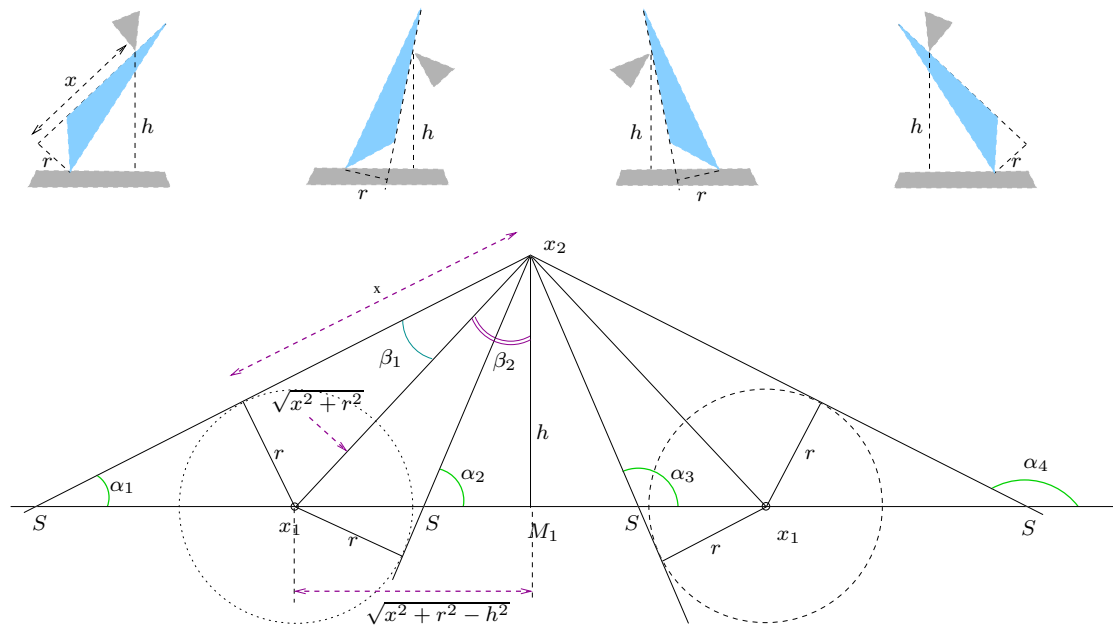


Abbildung 2.20: Fall 3.  $h > r$ .  $|x_2 M_2| \rightarrow \alpha$

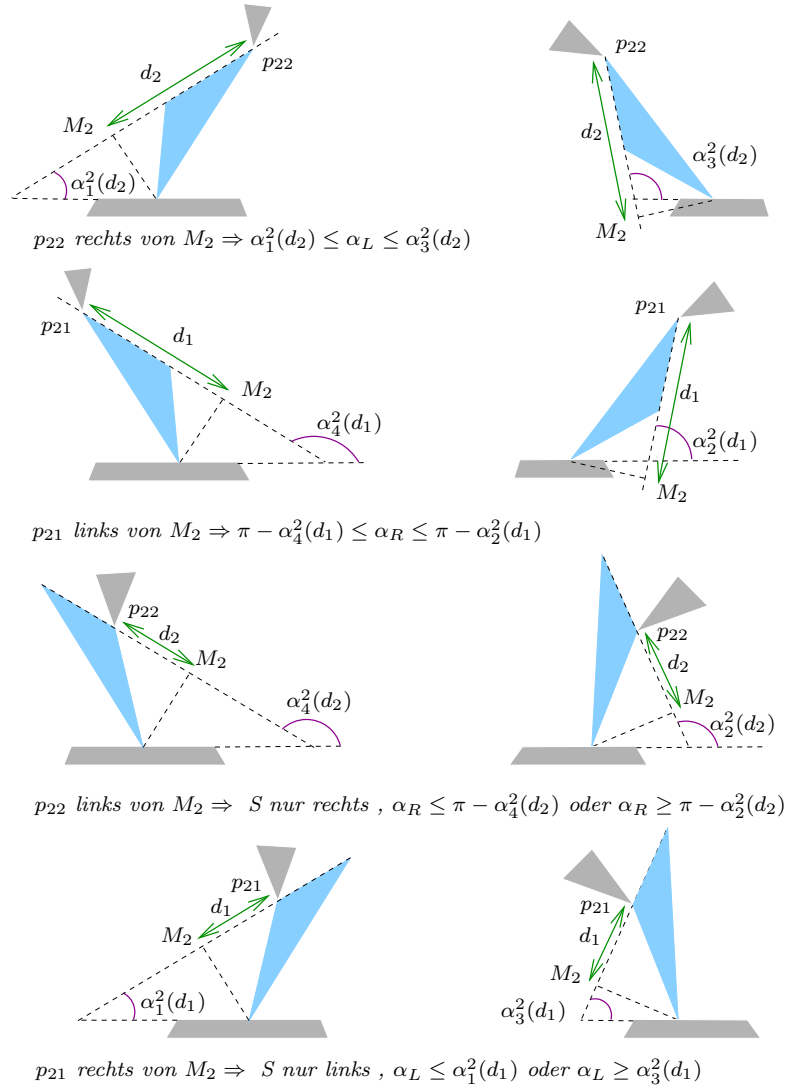


Abbildung 2.21: Fall 3.  $h > r$ .  $p_{21}, p_{22} \rightarrow \alpha_L, \alpha_R$



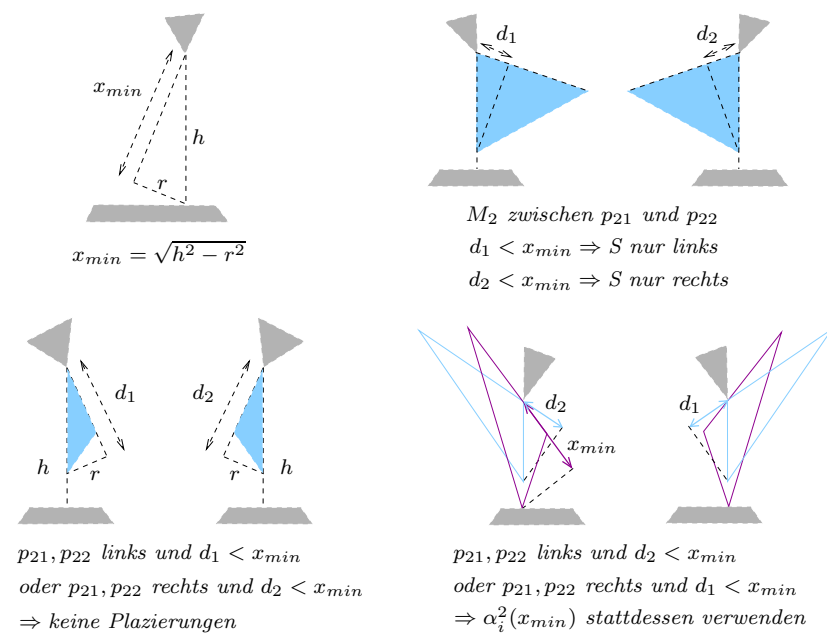


Abbildung 2.22: Fall 3.  $h > r$ . Minimum für  $|M_2 p_{21}|, |M_2 p_{22}|$

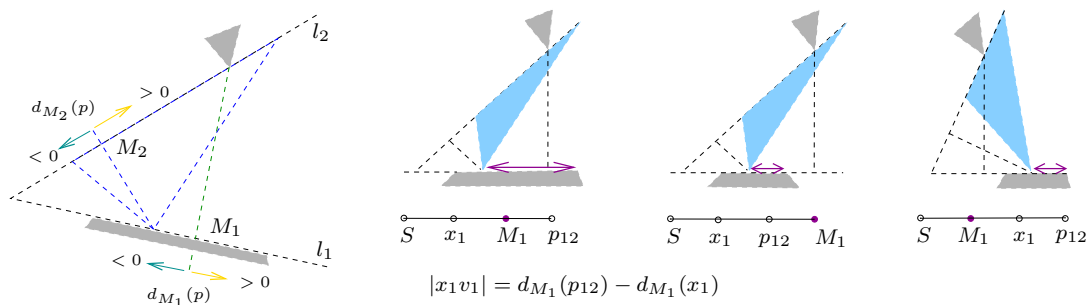


Abbildung 2.23: Fall 3.  $h > r$ .  $d_{M_i}(p)$

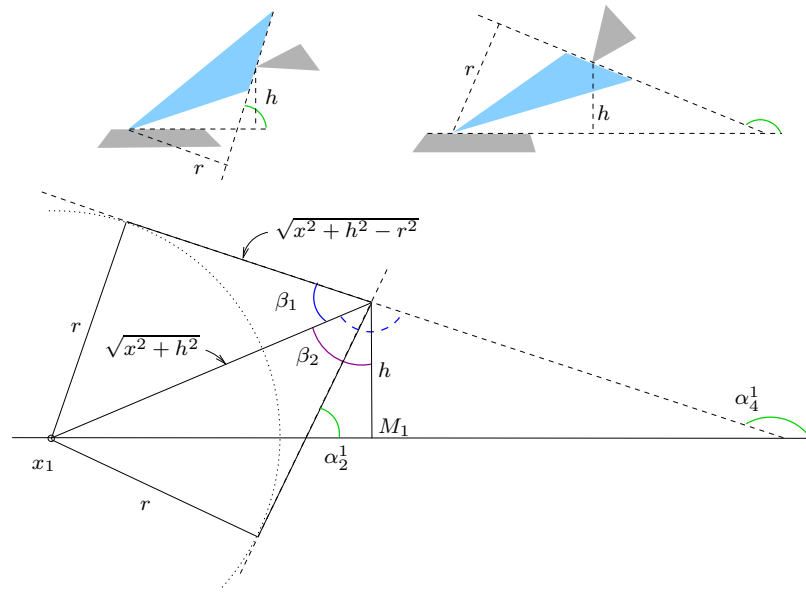


Abbildung 2.24: Fall 3.  $h < r$ .  $|M_1 x_1| \rightarrow \alpha$

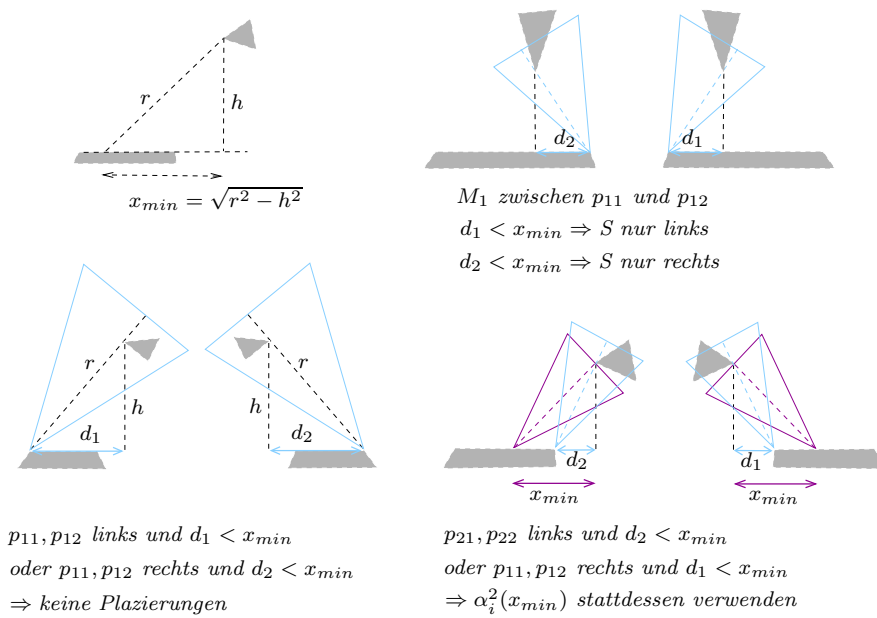


Abbildung 2.25: Fall 3.  $h < r$ . Minimum für  $|M_1 p_{11}|, |M_1 p_{12}|$

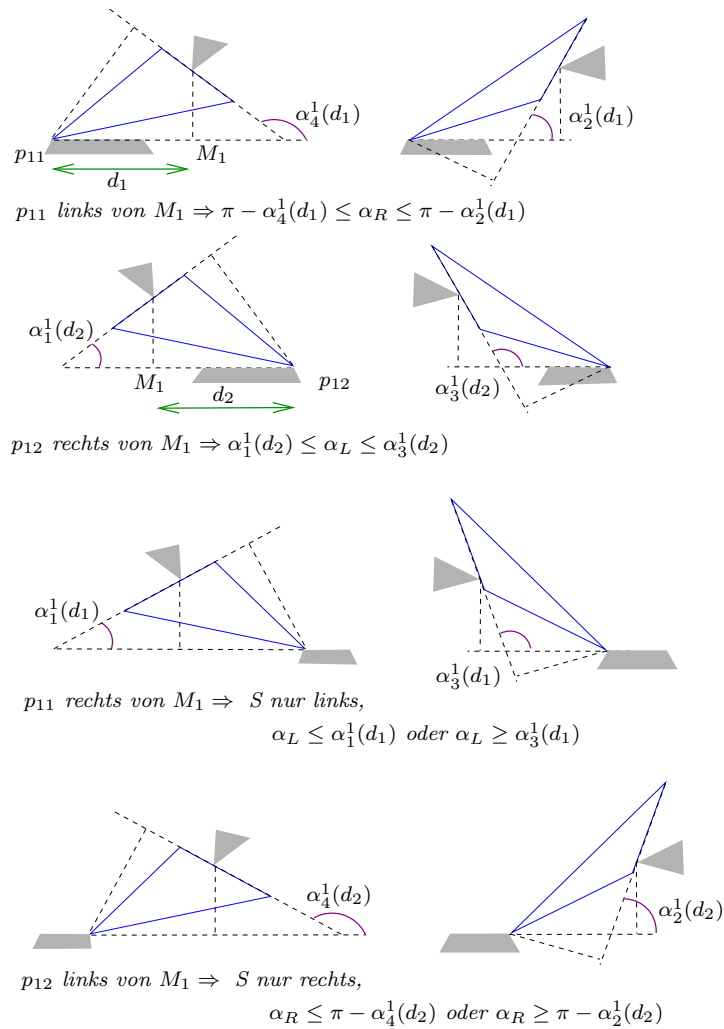


Abbildung 2.26: Fall 3.  $h < r$ .  $p_{11}, p_{12} \rightarrow \alpha_L, \alpha_R$

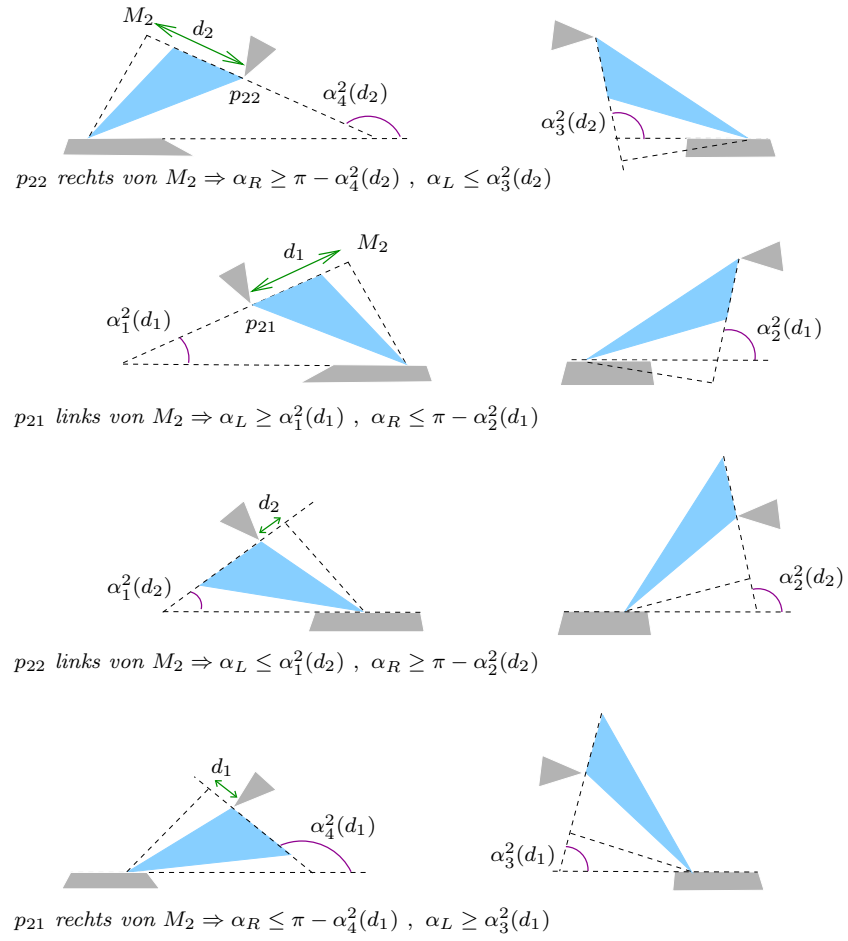


Abbildung 2.27: Fall 3.  $h < r$ .  $p_{21}, p_{22} \rightarrow \alpha_L, \alpha_R$

# Kapitel 3

## Eckpunkte der unteren Kontur

### 3.1 Zusammenführung von $A_L(O_i)$ und $A_R(O_i)$

Nachdem wir alle Funktionen  $f_{O_i O_j}$  berechnet und in die Arrangements  $A_L(O_i)$ ,  $A_R(O_i)$  eingeteilt haben, müssen wir in jedem Arrangement (mit mehr als einer Funktion) die untere Kontur bilden und aus ihren Eckpunkten (den Schnittpunkten der Funktionen, die auf der unteren Kontur liegen) die erste Gruppe von kritischen Plazierungen bekommen.

Das sind alle kritischen Plazierungen, bei denen zwei Kontaktpaare  $O_2, O_3$  das dritte Kontaktpaar  $O_1$  beschränken, und zwar in die gleiche Richtung - zu demselben Endpunkt der  $O_1$ -Kante. Beide Beschränkungsfunktionen liegen entweder in  $A_L(O_1)$  oder in  $A_R(O_1)$ . Es kann aber auch sein, dass die Funktion  $f_{O_1 O_2}$  in  $A_L(O_1)$  ist, und  $f_{O_1 O_3}$  in  $A_R(O_1)$ . D.h., beide  $O_2$  und  $O_3$  beschränken  $O_1$ , aber in verschiedene Richtungen, so dass in den Funktionen der Abstand zu verschiedenen Kantenendpunkten gemessen wird. Solche kritischen Plazierungen findet man wie folgt. Für ein Kontaktpaar  $O_i$  seien  $\Psi_L(O_i)$  und  $\Psi_R(O_i)$  die untere Konturen von  $A_L(O_i)$  und  $A_R(O_i)$ .

- Für jedes Kontaktpaar  $O_i$  mische alle Eckpunkte (nicht nur die Schnittpunkte) von  $\Psi_L(O_i)$  und  $\Psi_R(O_i)$  in eine sortierte (Intervall-) Liste L.
- Gehe die Liste L durch. Über jedem Intervall  $I \in L$  sind  $\Psi_L(O_i)$  und  $\Psi_R(O_i)$  jeweils durch eine Funktion  $f_{O_i O_j}$  und  $f_{O_i O_k}$  vertreten.
  - Transformiere  $f_{O_i O_k}$ , so dass sie den Abstand vom gleichen Kantenendpunkt aus wie  $f_{O_i O_j}$  misst:  $f_{O_i O_k}^*(\phi) = d_i - f_{O_i O_k}(\phi)$ , wobei  $d_i$  die Länge der Kante von  $O_i$  ist.
  - Wenn sich  $f_{O_i O_k}^*$  und  $f_{O_i O_j}$  über dem Intervall I schneiden, speichere den Schnittpunkt als eine kritische Platzierung.

So beschreiben es Kedem und Sharir. Wir können aber auch zuerst ein Arrangement  $A_R^*(O_i)$  mit den transformierten Funktionen aus  $A_R(O_i)$  aufstellen, dann in  $A_R^*(O_i)$  die obere Kontur  $\Psi_R^*(O_i)$  bestimmen und die Schnittpunkte  $\Psi_L(O_i) \cap \Psi_R^*(O_i)$  berechnen.

## 3.2 A-Funktion und B-Funktion

Für die Berechnung der unteren Kontur muss man wissen, wie die Funktionen aussehen, und ihre Ableitungen, Umkehrfunktionen und die Schnittpunkte bestimmen können (unter Umkehrfunktion versteht man hier die Ermittlung aller (mehrerer)  $x_1, \dots, x_n$ , die für ein gegebenes  $y$  die Gleichung  $f(x) = y$  erfüllen). Um diese spezielle Berechnungen geht es weiter in diesem Kapitel. Die untere Kontur selbst wird dann z.B. mit einem Divide & Conquer-Algorithmus gefunden.

In Kapitel 2 haben wir zwei Arten von Funktionen gefunden:

$$F_A(x) := a + b \sin(c + dx), \quad a, b, c \in \mathbb{R}, b \neq 0, d \in \{-1, 1\}$$

$$\begin{aligned} F_B(x) &:= a + \frac{b}{\sin(d+x)} + \frac{c}{\tan(d+x)} = \\ &= a + \frac{b + c \cos(x+d)}{\sin(x+d)}, \quad a, b, c, d \in \mathbb{R}, b \neq 0, c \neq 0 \end{aligned}$$

Die A-Funktion ( $F_A$ ) sieht wie Sinuskurve aus, nur verschoben und mit einer Konstante multipliziert. Die B-Funktion ( $F_B$ ) ist etwas komplizierter: wenn  $|b| < |c|$ , sieht sie ähnlich wie eine Tangensfunktion aus, aber wenn  $|b| > |c|$ , hat sie lokale Maxima und Minima und teilweise andere Grenzwerte in  $x = k\pi + d$ ,  $k = \dots -1, 0, 1, 2, \dots$  (siehe Abbildung 3.1, Abbildung 3.2). Aber die Funktionen  $f_{O_i O_j}$  sind Teilfunktionen der A- oder B-Funktionen, und diese Teilfunktionen sind beschränkt (nach der Definition ist der Funktionswert ein Abstand und ist kleiner als die Länge einer bestimmten endlichen Kante). D.h., auf diesen Intervallen sind die Funktionen auch stetig und wir müssen uns nicht um die Stellen kümmern, in denen sie nicht definiert sind. Außerdem liegen alle Definitionsbereiche von den Teilfunktionen innerhalb von  $[0, 2\pi]$  und wir haben immer eine endliche Lösung für die Umkehrfunktion und für die Berechnung der Schnittpunkte.

## 3.3 A-Funktion

$$F_A(x) = a + b \sin(c + dx), \quad a, b, c \in \mathbb{R}, b \neq 0, d \in \{-1, 1\}$$

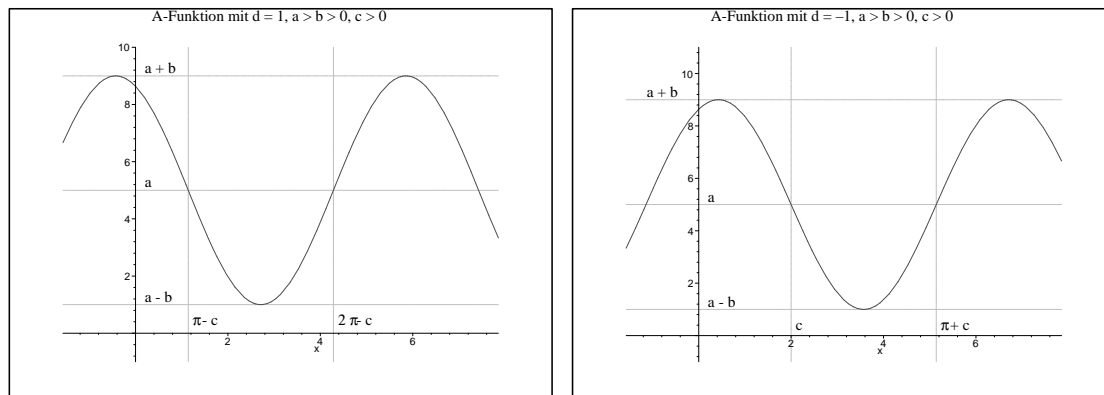


Abbildung 3.1: A-Funktion

Die Ableitung von der A-Funktion ist  $bd \cos(c + dx)$ . Lokale Extrema sind  $\{\pi/2 - c + k\pi \mid k = \dots, -1, 0, 1, \dots\}$ , wenn  $d=1$  und  $\{\pi/2 + c + k\pi \mid k = \dots, -1, 0, 1, \dots\}$ , wenn  $d=-1$ . Der Wertebereich liegt zwischen  $y_{min} = \min(a + b, a - b)$  und  $y_{max} = \max(a + b, a - b)$ .

Die Umkehrfunktion für  $F_A(x)$  mit  $d=1$  (mit  $d=-1$  analog) wird so berechnet:

- Wenn  $y > y_{max}$  oder  $y < y_{min}$ , dann gibt es keine Lösung.
- $y = y_{max} \Rightarrow x = \pi/2 - c$  (einzige Lösung). Wenn  $x$  ausserhalb von  $[0, 2\pi]$  liegt, muss es entsprechend um  $2\pi$  verschoben werden. Das gilt auch für alle folgende Berechnungen von Umkehrfunktionen und Schnittpunkten.
- $y = y_{min} \Rightarrow x = 3\pi/2 - c$ .
- Sonst ist  $x_1 = \arcsin(\frac{y-a}{b}) - c$  und

$$x_2 = \begin{cases} \pi - \arcsin(\frac{y-a}{b}) - c, & x_1 + c \geq 0 \\ -\pi - \arcsin(\frac{y-a}{b}) - c, & \text{sonst} \end{cases}$$

(zwei Lösungen).

Die Schnittpunkte von zwei verschiedenen A-Funktionen kann man als Nullstellen der Differenzfunktion finden. Sei  $f_1(x) = a_1 + b_1 \sin(c_1 + d_1 x)$  und  $f_2(x) = a_2 + b_2 \sin(c_2 + d_2 x)$ . Dann ist

$$g(x) = f_1(x) - f_2(x) = a_1 + b_1 \sin c_1 \cos x + d_1 b_1 \cos c_1 \sin x -$$

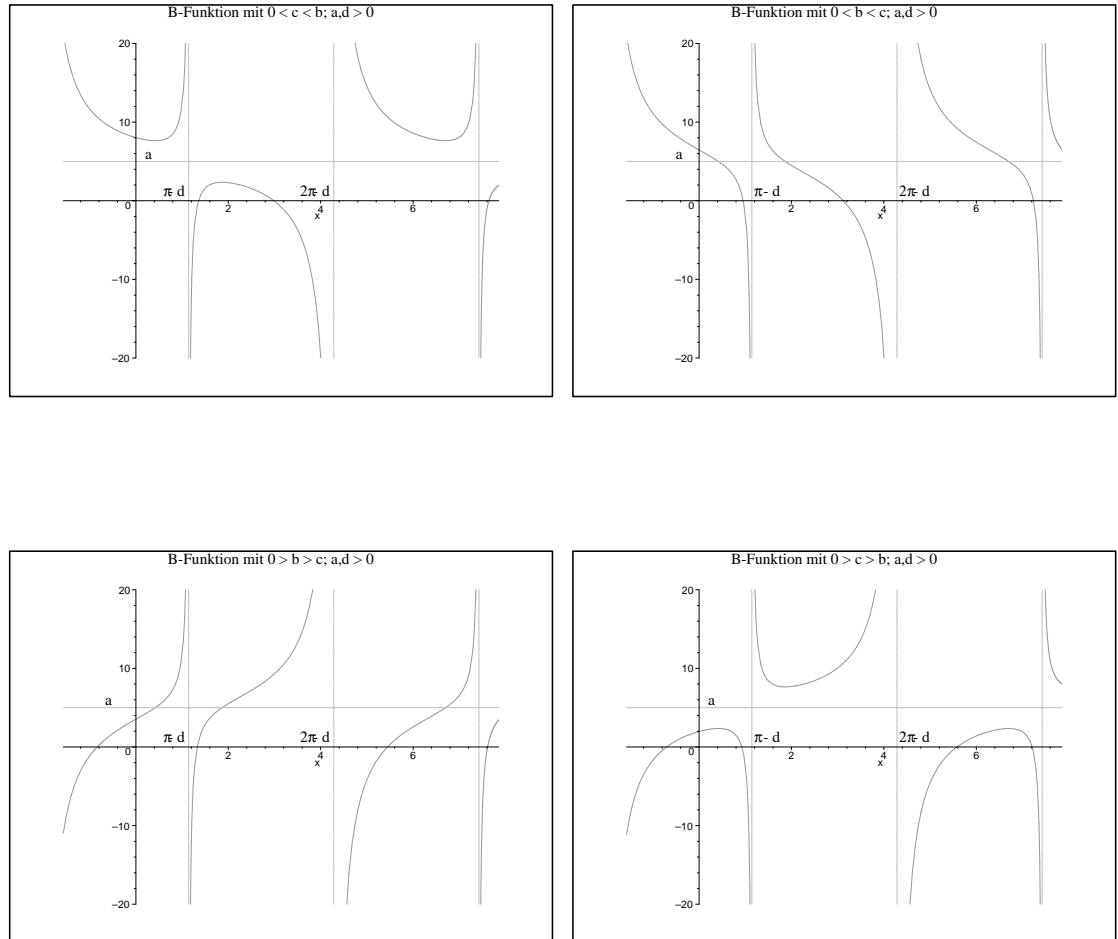


Abbildung 3.2: B-Funktion



$$\begin{aligned}
& -a_2 - b_2 \sin c_2 \cos x - d_2 b_2 \cos c_2 \sin x = \\
& = a_1 - a_2 + (b_1 \sin c_1 - b_2 \sin c_2) \cos x + (d_1 b_1 \cos c_1 - d_2 b_2 \cos c_2) \sin x.
\end{aligned}$$

Wir suchen also die Nullstellen von einer Funktion

$$F_C(x) := a_3 + b_3 \sin x + c_3 \cos x$$

mit  $a_3 = a_1 - a_2$ ,  $b_3 = d_1 b_1 \cos c_1 - d_2 b_2 \cos c_2$  und  $c_3 = b_1 \sin c_1 - b_2 \sin c_2$ . Dabei können  $b_3$  und  $c_3$  nicht gleichzeitig  $=0$  sein: sonst ist  $g(x)$  konstant, hat also keine Nullstellen wegen  $f_1(x) \not\equiv f_2(x) \Rightarrow g(x) \neq 0$ .

Sei  $\mathcal{N} = \{x \mid g(x) = 0\}$  die gesuchte Menge der Nullstellen.

Wir müssen mehrere Fälle betrachten:

1. Zuerst prüfen wir, ob ein  $x$  mit  $\cos x = 0$  zufällig eine Nullstelle ist (damit wir später  $\cos x \neq 0$  annehmen können):

$$a_3 + b_3 = 0 \Rightarrow \pi/2 \in \mathcal{N}, \text{ und}$$

$$a_3 - b_3 = 0 \Rightarrow 3\pi/2 \in \mathcal{N}.$$

2.  $a_3 = 0$

$$\Rightarrow b_3 \sin x + c_3 \cos x = 0$$

$$\Rightarrow \tan x = -c_3/b_3 \quad (\cos x \neq 0, |b| + |c| \neq 0 \Rightarrow b_3 \neq 0)$$

$$\Rightarrow \mathcal{N} \supseteq \{\arctan(-c_3/b_3), \pi + \arctan(-c_3/b_3)\}.$$

3.  $a_3, b_3 \neq 0, c_3 = 0$

$$\Rightarrow b_3 \sin x = -a_3$$

$$(a) \quad |a_3| > |b_3| \Rightarrow \mathcal{N} = \emptyset$$

$$(b) \quad |a_3| \leq |b_3| \Rightarrow \mathcal{N} = \{\arcsin(-a_3/b_3), \pi - \arcsin(-a_3/b_3)\}$$

4.  $a_3, c_3 \neq 0, b_3 = 0$

$$\Rightarrow c_3 \cos x = -a_3$$

$$(a) \quad |a_3| > |c_3| \Rightarrow \mathcal{N} = \emptyset$$

$$(b) \quad |a_3| \leq |c_3| \Rightarrow \mathcal{N} = \{\arccos(-a_3/c_3), 2\pi - \arccos(-a_3/c_3)\}$$

5.  $a_3, b_3, c_3 \neq 0$

$$\Rightarrow b_3 \sin x + c_3 \cos x = -a_3$$

$$\Rightarrow (b_3 \sin x + c_3 \cos x)^2 = a_3^2$$

$$\Rightarrow b_3^2 \sin^2 x + c_3^2 \cos^2 x + 2b_3 c_3 \sin x \cos x - a_3^2 = 0$$

$$\Rightarrow b_3^2 \tan^2 x + c_3^2 + 2b_3 c_3 \tan x - a_3^2(1 + \tan^2 x) = 0$$

$$t := \tan x$$

$$\Rightarrow (b_3^2 - a_3^2)t^2 + 2b_3 c_3 t + c_3^2 - a_3^2 = 0$$

- (a)  $a_3^2 - b_3^2 = 0$   
 $\Rightarrow t = (a_3^2 - c_3^2)/(2b_3c_3)$   
 $\Rightarrow \mathcal{N} = \{\arctan(t), \pi + \arctan(t)\}$
- (b)  $a_3^2 - b_3^2 \neq 0$
- i.  $c_3^2 + b_3^2 - a_3^2 < 0 \Rightarrow \mathcal{N} = \emptyset$
  - ii.  $c_3^2 + b_3^2 - a_3^2 \geq 0$   
 $\Rightarrow t_{1,2} = (b_3c_3 \pm a_3\sqrt{c_3^2 + b_3^2 - a_3^2})/(a_3^2 - b_3^2)$   
 $\Rightarrow \mathcal{N} = \{\arctan(t_1), \pi + \arctan(t_1), \arctan(t_2), \pi + \arctan(t_2)\}$

### 3.4 B-Funktion

$$F_B(x) = a + \frac{b + c \cos(x + d)}{\sin(x + d)}, \quad a, b, c, d \in \mathbb{R}, \quad b \neq 0, \quad c \neq 0$$

#### 3.4.1 Ableitung und Umkehrfunktion

Die Ableitung von der B-Funktion ist  $-(c + b \cos(d + x))/\sin^2(d + x)$ . Wenn  $|c| \leq |b|$ , dann hat sie Nullstellen - lokale Extrema von  $F_B$  - in  $\arccos(-c/b) - d$  und  $2\pi - \arccos(-c/b) - d$ . Sonst ist die B-Funktion monoton.

Die Umkehrfunktion wird mit Hilfe der Funktion  $F_C(x)$  berechnet:

$$a + \frac{b + c \cos(x + d)}{\sin(x + d)} = y$$

$$(y - a) \sin(x + d) - c \cos(x + d) - b = 0$$

$$(y - a) \sin x \cos d + (y - a) \cos x \sin d - c \cos x \cos d + c \sin x \sin d - b = 0$$

$$((y - a) \cos d + c \sin d) \sin x + ((y - a) \sin d - c \cos d) \cos x - b = 0$$

$\Downarrow$

$$F_C(x) = a_3 + b_3 \sin x + c_3 \cos x = 0,$$

mit  $a_3 = -b$ ,  $b_3 = (y - a) \cos d + c \sin d$  und  $c_3 = (y - a) \sin d - c \cos d$ .

Die Nullstellen von  $F_C(x)$  werden wie oben berechnet.

### 3.4.2 Schnittpunkte $F_{B_1}(x) \cap F_{B_2}(x)$ und $F_B(x) \cap F_A(x)$

Wir müssen auch zwischen zwei B-Funktionen oder zwischen  $F_A(x)$  und  $F_B(x)$  Schnittpunkte berechnen können. Wie Kedem und Sharir gezeigt haben, kann es maximal 4 Schnittpunkte geben. Sie entsprechen den Nullstellen der Differenzfunktion  $g(x) = f_1(x) - f_2(x)$ , aber diese Funktion sieht nicht so aus, als ob man die Nullstellen einfach herleiten könne. Die Nullstellen von  $g'(x)$ , mögliche lokale Extrema von  $g(x)$ , sind auch nicht leicht zu finden. Deswegen bestimmen wir die Schnittpunkte numerisch, und zwar wie folgt:

Zuerst wird der gemeinsame Definitionsbereich von  $f_1$  und  $f_2$  berechnet (wenn leer: fertig). Er wird gleichzeitig in Intervalle geteilt, auf denen jede der beiden Funktionen monoton ist. Jedes dieser Intervalle  $[a, b]$  wird auf folgende Weise behandelt:

1. Falls das Intervall aus einem Punkt besteht: Funktionswerte vergleichen (und weiter zum nächsten Intervall).
2. Falls die Wertebereiche von  $f_1$  und  $f_2$  sich auf  $[a, b]$  nicht überschneiden: kein Schnitt.
3. Falls  $g(a) < \epsilon$ : Schnittpunkt gefunden ( $\epsilon$  entspricht der gewünschten Genauigkeit). Wenn das schon der vierte Schnittpunkt ist: fertig. Sonst muss man mit  $a := a + \delta$  weiter suchen. Die Wahl von  $\delta$  ist nicht ganz einfach, wenn man keine Nullstellen verlieren oder zwei mal zählen will. Dazu kommen wir später.
4. Falls  $g(b) < \epsilon$ : Schnittpunkt gefunden (analog wie oben).
5. Wenn  $g(x) = f_1(x) - f_2(x)$  in  $a$  und  $b$  verschiedene Vorzeichen hat, muss es auf  $[a, b]$  mindestens ein Schnittpunkt geben. Diesen kann man durch die Intervallhalbierung finden (das Newton-Verfahren ist dafür nicht so gut geeignet, weil die Ableitung aus dem Intervall führen kann).
6. Wenn  $g(x)$  gleiches Vorzeichen in  $a$  und  $b$  hat, gibt es auf  $[a, b]$  möglicherweise Schnittpunkte. Hier kann man auch mit einer Intervallunterteilung weiter suchen, aber mit einer zusätzlichen Terminierungsabfrage für den Fall, dass es keine Schnittpunkte gibt. Dazu beweisen wir eine Lemma.

**Lemma 9** Sei  $f(x)$  stetig und differenzierbar in  $[a, b]$ , und sei  $|f'(x)|$  beschränkt:  $|f'(x)| \leq C, \forall x \in [a, b]$ . Dann gilt:

$$f(a) > \frac{C}{2}(b-a) \text{ und } f(b) > \frac{C}{2}(b-a) \Rightarrow \nexists x \in [a, b] : f(x) = 0$$

**Proof.** Nehmen wir an,  $\exists x \in [a, b] : f(x) = 0$ . Dann gibt es zwei Fälle:

1.  $x \leq (a+b)/2 \Rightarrow$  Nach dem Mittelwertsatz von Lagrange  
 $\exists \xi \in [a, x] : f'(\xi) = (f(x) - f(a))/(x - a)$   
 $\Rightarrow |f'(\xi)| = |f(a)|/(x - a) > \frac{C(b-a)}{2(x-a)} \geq C$  - Widerspruch.
2.  $x > (a+b)/2 \Rightarrow \exists \xi \in [x, b] : f'(\xi) = (f(b) - f(x))/(b - x)$   
 $\Rightarrow |f'(\xi)| = |f(b)|/(b - x) > \frac{C(b-a)}{2(b-x)} > C$  - Widerspruch.

□

Wenn also bei der Intervallunterteilung  $(b-a) \rightarrow 0$ , dann verkleinert sich auch die obere Schranke für  $f(a)$  und  $f(b)$ , und wenn es keine Nullstellen in  $[a, b]$  gibt, werden  $f(a)$  und  $f(b)$  irgendwann größer als  $\frac{C}{2}(b-a)$ . Das kann als Terminierungsabfrage verwendet werden.

Als Konstante  $C$  für die Differenzfunktion  $g(x)$  kann man die Summe der Schranken  $C_1$  und  $C_2$  der einzelnen Funktionen  $f_1(x)$  und  $f_2(x)$  nehmen. Die Ableitung von einer A-Funktion ist beschränkt durch  $|b|$ , und für eine B-Funktion kann man  $\max(|F'_B(a)|, |F'_B(b)|)$  nehmen.

Ungeklärt ist noch die Wahl von  $\delta$ . Nehmen wir an,  $x_1$  wurde als Nullstelle akzeptiert, weil  $g(x_1) < \epsilon$  gilt. Wenn rechts (oder links) von  $x_1$  noch eine Nullstelle ist, muss  $\delta$  klein genug sein, so dass diese Nullstelle nicht verloren geht. Andererseits wird man eine Nullstelle zwei mal zählen, wenn  $\delta$  zu klein ist und  $g(x_1 + \delta)$  deshalb immer noch kleiner als  $\epsilon$  ist. Das kann man wie folgt lösen:

- Um keine Nullstellen zu verpassen wird der nächste Punkt mit  $g(x) > \epsilon$  gesucht. Dazu kann man die Ableitung  $g'(x)$  verwenden, solange sie nicht kleiner ist als ein  $\epsilon_2$ :  $x_{i+1} = y_i/g'(x_i)$ , wobei

$$y_i = \begin{cases} |g(x_i) - \epsilon| & , \text{ falls } g'(x_i)g(x_i) > 0 \\ 2\epsilon - |g(x_i) - \epsilon| & , \text{ sonst} \end{cases}$$

- Wenn die Ableitung aber sehr klein ist, kann man die Tatsache benutzen, dass  $g(x)$  gleichmäßig stetig ist: man kann ein  $\hat{\delta}$  finden, so dass  $|g(x + \hat{\delta}) - g(x)| < \hat{\epsilon} = \min(\epsilon + g(x_i), \epsilon - g(x_i))$ . D.h., mit  $x_{i+1} = x_i + \hat{\delta}$  kann man weitergehen bis die Ableitung wieder groß genug ist oder  $g(x_i) > \epsilon$ .

Als  $\hat{\delta}$  kann man  $\min(\delta_1, \delta_2)$  verwenden, wobei  $\delta_1$  und  $\delta_2$  das gleiche Problem für die einzelnen Funktionen  $f_1$  und  $f_2$  lösen, d.h.,  $|f_1(x + \delta_1) - f_1(x)| < \hat{\epsilon}/2$  und  $|f_2(x + \delta_2) - f_2(x)| < \hat{\epsilon}/2$ . Für eine A-Funktion  $f_1$  kann man  $\delta_1 = |2 \arcsin(\hat{\epsilon}/4b)|$  nehmen:

$$\begin{aligned} |f_1(x + \delta_1) - f_1(x)| &= |b(\sin(c \pm x) - \sin(c \pm x + \delta_1))| = \\ &= |2b \sin(\delta_1/2) \cos(c \pm x + \delta_1/2)| < \hat{\epsilon}/2 \end{aligned}$$

Für eine B-Funktion  $\delta_1 = \min(\arcsin(\hat{\epsilon}k^2/4c), 2 \arcsin(\hat{\epsilon}k^2/8b))$ , wobei  $k$  die untere Schranke für  $|\sin(x+d)|$  auf diesem Intervall ist:

$$\begin{aligned}
 |f_1(x + \delta_1) - f_1(x)| &= \left| \frac{b + c \cos(x + d)}{\sin(x + d)} - \frac{b + c \cos(x + d + \delta_1)}{\sin(x + d + \delta_1)} \right| = \\
 &= \left| \frac{c \sin(\delta_1) + b(\sin(x + d + \delta_1) - \sin(x + d))}{\sin(x + d) \sin(x + d + \delta_1)} \right| < \\
 &< \left| \frac{c \sin(\delta_1) + 2b \sin(\delta_1/2) \cos(x + d + \delta_1/2)}{k^2} \right| < \hat{\epsilon}/2
 \end{aligned}$$



# Kapitel 4

## Zyklische Tripel

### 4.1 Algorithmus von Kedem und Sharir

Jetzt müssen wir noch die kritischen Plazierungen mit einem Zyklus  $(O_1/O_3, O_3/O_2, O_2/O_1)$  finden.

Der Algorithmus von Kedem und Sharir verwendet dafür auch die Eckpunkte der unteren Konturen:

- Sortiere die Listen der Eckpunkte der unteren Konturen zu einer Liste der Intervalle  $L = I_0, I_1, \dots$ . Weil die einzelne Listen schon vorsortiert sind, brauchen wir in jedem Schritt nur das erste Element von jeder Liste betrachten.
- Beginne mit dem äußersten linken Intervall  $I_0 \in L$ .  
Für jedes  $O_1$  teste, ob über  $I_0$  ein Zyklus  $(O_1, O_2, O_3)$  existiert. Dabei muss man für jedes Kontaktpaar beide Arrangements beachten, d.h. man durchsucht einen binären Baum mit 8 Blättern (siehe Abbildung 4.1):
  - Für jedes  $O_1$  und jedes  $E \in \{L, R\}$  finde, welche Funktion über diesem Intervall auf der unteren Kontur von  $A_E(O_1)$  liegt,
  - für das entsprechende Kontaktpaar  $O_2$  und jedes  $E \in \{L, R\}$  finde, welche Funktion über diesem Intervall auf der unteren Kontur von  $A_E(O_2)$  liegt,
  - für das entsprechende Kontaktpaar  $O_3$  und jedes  $E \in \{L, R\}$  teste, ob die Funktion, die über diesem Intervall auf der unteren Kontur von  $A_E(O_3)$  liegt, dem Kontaktpaar  $O_1$  entspricht.

Wenn also ein Zyklus existiert,

- Berechne die Winkel  $\theta_{11}, \theta_{12}, \dots$  der maximal vier zugehörigen kritischen Plazierungen, an denen der Zyklus beteiligt ist. Wie findet man diese Winkel? Die Funktionen  $f_{O_i O_j}$  helfen hier wenig, weil

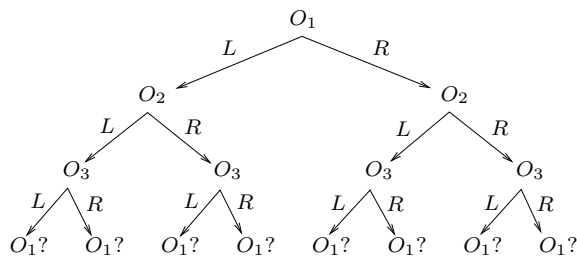


Abbildung 4.1: Suche nach einem Zyklus

keine zwei von ihnen sich auf die gleiche Kante beziehen. Das Problem betrachten wir im nächsten Abschnitt.

- Berichte sofort für die in  $I_0$  enthaltenen Orientierungen  $\theta_i$  die Plazierungen  $(\theta_i, (O_1, O_2, O_3))$ .
- Füge die übrigen Plazierungen  $(\theta_j, (O_1, O_2, O_3))$  in eine nach Winkeln sortierte Warteschlange  $W$  ein.
- Bearbeite von links nach rechts die übrigen Intervalle  $I_k \in L$ .
  - Beim Übergang von  $I_{k-1}$  nach  $I_k$  kann nur in einem Arrangement  $A(O_1)$  ein Wechsel der unteren Kontur stattfinden.  $f_{O_1 O_2}$  werde von  $f_{O_1 O_4}$  abgelöst.
  - Falls  $O_1$  über  $I_{k-1}$  einen Zyklus mit  $O_2$  hatte, entferne aus  $W$  die zu  $(O_1, O_2, O_3)$  gehörenden kritischen Plazierungen; diese sind nicht mehr aktuell.
  - Teste, ob  $O_1, O_4$  einen Zyklus  $(O_1, O_4, O_5)$  über  $I_k$  bilden. Wenn ja, berechne wie oben die kritischen Plazierungen.
  - Berichte alle in  $W$  gespeicherten Plazierungen  $(\theta, (O_i, O_k, O_l))$  mit  $\theta \in I_k$ , und entferne sie aus  $W$ .

## 4.2 Plazierungen von einem Zyklus

Wir haben also drei Kontaktpaare, die sich irgendwie in einem Zyklus beschränken, und zwar über einem bestimmten Winkelintervall für die Orientierung  $\theta$ . Wir müssen aber die (maximal vier) Winkel  $\theta_{11}, \theta_{12}, \dots$  aus diesem Intervall finden, die den kritischen Plazierungen mit diesen drei Kontaktpaaren entsprechen.

Hier gibt es vier Fälle (siehe Abbildung 4.2):

- drei Roboterecken (und drei Hinderniskanten),



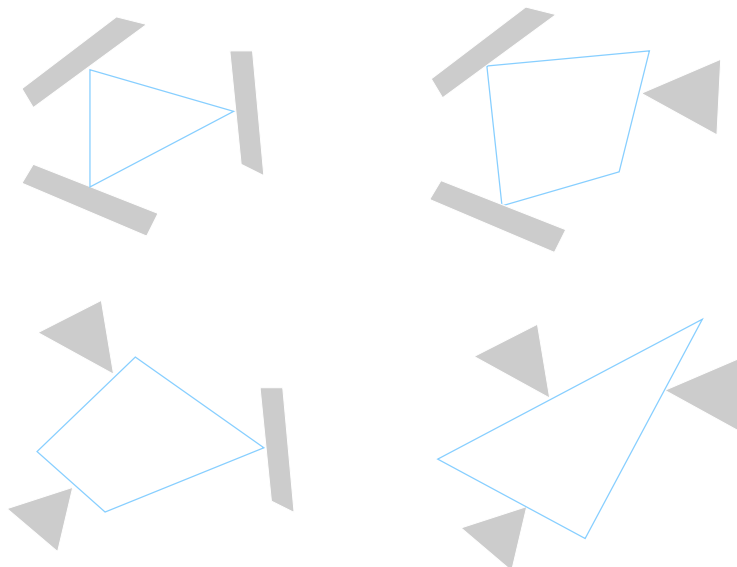


Abbildung 4.2: Verschiedene Fälle bei einem Zyklus

- zwei Roboterecken und eine Roboterkante,
- eine Roboterecke und zwei Roboterkanten,
- drei Roboterkanten.

### 4.2.1 Drei Roboterecken

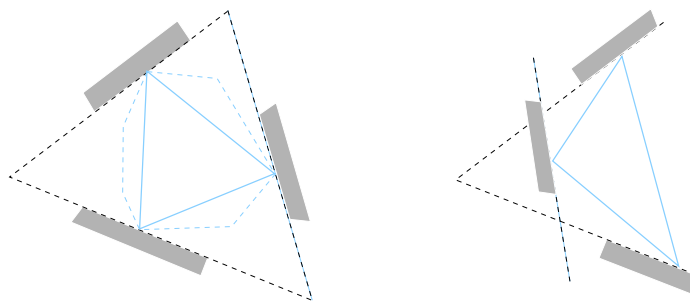


Abbildung 4.3: Drei Roboterecken

Seien  $O_1, O_2$  und  $O_3$  Kontaktpaare mit Roboterecken und Hinderniskanten, die sich zyklisch beschränken, und seien  $T_i$  ihre Tangenten. Weil jede zwei von ihnen sich beschränken, sind keine zwei Tangenten parallel (und alle Roboterecken sind verschieden). Jede Tangente entspricht einer Hinderniskante und teilt die Ebene lokal in eine “freie” und eine “verbotene”

Halbebene. Der Durchschnitt von allen drei freien Halbebenen ist entweder ein Dreieck (Fall 1.1, siehe Abbildung 4.4) oder ein offenes Gebiet (Fall 1.2, siehe Abbildung 4.5).

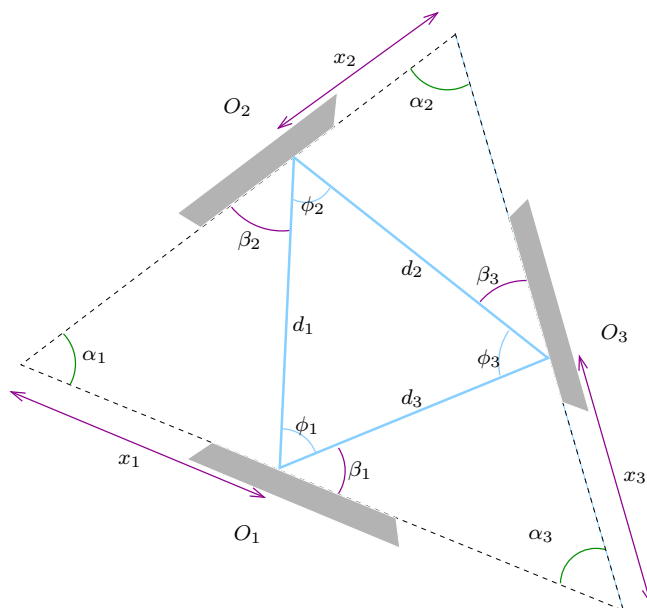


Abbildung 4.4: Fall 1.1

Im ersten Fall muss der Roboter im Hindernisdreieck plaziert werden. Die entsprechenden Orientierungen können wie folgt berechnet werden. Sei

- $d_1$  der Abstand zwischen den Roboterecken von  $O_1$  und  $O_2$ ,  $d_2$  und  $d_3$  analog,
- $\phi_1, \phi_2$  und  $\phi_3$  die Winkel des Dreiecks aus den beteiligten Roboterecken.
- $\alpha_1$  der Winkel zwischen  $T_1$  und  $T_2$ ,  $\alpha_2$  und  $\alpha_3$  analog,
- $x_1$  der Abstand zwischen dem Berührungspunkt von  $O_1$  und dem Schnittpunkt von  $T_1$  und  $T_2$ ,  $x_2$  und  $x_3$  analog.
- $\beta_1$  der Winkel zwischen  $T_1$  und der Geraden durch die Berührungspunkte von  $O_1$  und  $O_3$ ,  $\beta_2$  und  $\beta_3$  analog

(siehe Abbildung 4.3). Dabei sind  $d_i$ ,  $\phi_i$  und  $\alpha_i$  fest, und  $x_i$  und  $\beta_i$  ändern sich mit der Orientierung und nehmen bei verschiedenen möglichen Plazierungen verschiedene Werte an.

Zuerst schreiben wir drei Gleichungen für den Zusammenhang von  $x_i$  und  $\beta_i$  auf:

$$x_1 = \frac{d_1}{\sin \alpha_1} \sin \beta_2 \quad (4.1)$$

$$x_2 = \frac{d_2}{\sin \alpha_2} \sin \beta_3 = \frac{d_2}{\sin \alpha_2} \sin(\beta_2 + \phi_2 - \alpha_2) \quad (4.2)$$

$$x_3 = \frac{d_3}{\sin \alpha_3} \sin \beta_1 = \frac{d_3}{\sin \alpha_3} \sin(\beta_2 - \phi_1 + \alpha_1) \quad (4.3)$$

Die vierte Gleichung erhalten wir aus der Tatsache, das sich das große Dreieck aus vier kleinen zusammensetzt. Sei  $S_h$  die Fläche des Hindernisdreiecks (Ecken sind die Schnittpunkte von  $T_i$ ) und  $S_r$  die Fläche des Roboterdreiecks (Ecken sind die Roboterecken von  $O_i$ ). Dann gilt:

$$\begin{aligned} S_h - S_r &= \frac{1}{2}x_1d_1 \sin(\pi - \alpha_1 - \beta_2) + \frac{1}{2}x_2d_2 \sin(\pi - \phi_2 - \beta_2) + \\ &+ \frac{1}{2}x_3d_3 \sin\left(\pi - \alpha_3 - (\pi - \phi_1 - (\pi - \beta_2 - \alpha_1))\right) \\ &\quad \updownarrow \end{aligned}$$

$$2S_h - 2S_r = x_1d_1 \sin(\alpha_1 + \beta_2) + x_2d_2 \sin(\phi_2 + \beta_2) + x_3d_3 \sin(\alpha_2 + \phi_1 - \beta_2)$$

Wenn wir hier  $x_i$  durch die Ausdrücke aus 4.1, 4.2 und 4.3 ersetzen, bekommen wir eine Gleichung mit nur einer Unbekannten  $\beta_2$ :

$$\begin{aligned} 2S_h - 2S_r &= \frac{d_1^2}{\sin \alpha_1} \sin \beta_2 \sin(\alpha_1 + \beta_2) + \\ &+ \frac{d_2^2}{\sin \alpha_2} \sin(\beta_2 + \phi_2 - \alpha_2) \sin(\phi_2 + \beta_2) + \\ &+ \frac{d_3^2}{\sin \alpha_3} \sin(\beta_2 + \alpha_1 - \phi_1) \sin(\alpha_2 + \phi_1 - \beta_2) \end{aligned}$$

Nach der Anwendung der Formeln  $\sin x \sin y = \frac{1}{2}(\cos(x-y) - \cos(x+y))$  und  $\cos(x+y) = \cos x \cos y - \sin x \sin y$  erhalten wir eine Gleichung der Form  $A + B \sin(2\beta_2) + C \cos(2\beta_2) = 0$ , und diese können wir lösen wie in Kapitel 3. Aus jeder Lösung für  $\beta_2$  muss man noch den Winkel  $\theta$  bekommen. Das gleiche haben wir schon bei der Berechnung von Funktionen in Kapitel 2 gemacht, nur für ganze Intervalle. Danach muss man überprüfen, ob der Winkel in dem Intervall enthalten ist, auf dem der Zyklus existiert. Außerdem muss man die Werte von  $x_i$  ausrechnen (mit 4.1, 4.2 und 4.3) und überprüfen, ob die Berührungspunkte auf den entsprechenden Kanten liegen.

Im zweiten Fall setzen sich die Dreiecke anders zusammen. O.B.d.A. sei  $O_2$  das Kontaktpaar in der Mitte, zwischen dem Hindernisdreieck und dem Roboterdreieck. Wir definieren die Variablen genauso wie im Fall 1.1 (siehe Abbildung 4.5). Dann gelten 4.1, 4.2 und 4.3 auch, und die vierte Gleichung ist

$$\begin{aligned} S_h + S_r &= \frac{1}{2}x_3d_3 \sin\left(\pi - \alpha_3 - \phi_1 - (\pi - \beta_2 - \alpha_1)\right) - \\ &- \frac{1}{2}x_1d_1 \sin(\alpha_1 + \beta_2) - \frac{1}{2}x_2d_2 \sin(\phi_2 + \beta_2) \Leftrightarrow \end{aligned}$$

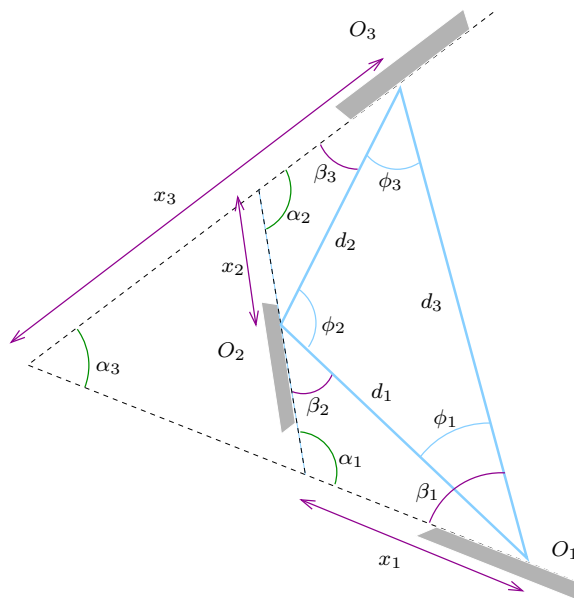


Abbildung 4.5: Fall 1.2

$$\Leftrightarrow 2S_h + 2S_r = -x_1 d_1 \sin(\alpha_1 + \beta_2) - x_2 d_2 \sin(\phi_2 + \beta_2) + x_3 d_3 \sin(\alpha_2 + \phi_1 - \beta_2)$$

Hier gibt es nur andere Vorzeichen, sonst ist alles wie im Fall 1.1.

#### 4.2.2 Drei Roboteranten

Wenn  $O_1, O_2$  und  $O_3$  alle aus Roboteranten und Hindernisecken bestehen, kann man die entsprechenden Plazierungen mit den Gleichungen aus dem Fall 1 finden. Dabei werden  $x_i, \alpha_i$  und  $\beta_i$  genauso definiert,  $d_i$  und  $\phi_i$  sind die Abstände und Winkel zwischen den Hindernisecken (siehe Abbildung 4.6).

Wie im Fall 1 gibt es zwei Möglichkeiten, es gelten 4.1, 4.2 und 4.3, und die vierte Gleichung ist entweder

$$2S_r - 2S_h = x_1 d_1 \sin(\alpha_1 + \beta_2) + x_2 d_2 \sin(\phi_2 + \beta_2) + x_3 d_3 \sin(\alpha_2 + \phi_1 - \beta_2)$$

oder

$$2S_r + 2S_h = -x_1 d_1 \sin(\alpha_1 + \beta_2) - x_2 d_2 \sin(\phi_2 + \beta_2) + x_3 d_3 \sin(\alpha_2 + \phi_1 - \beta_2),$$

wobei  $S_h$  die Fläche des Hindernisdreiecks ist und  $S_r$  die Fläche des Roboterdreiecks (Ecken sind die Schnittpunkte von  $T_i$ ).

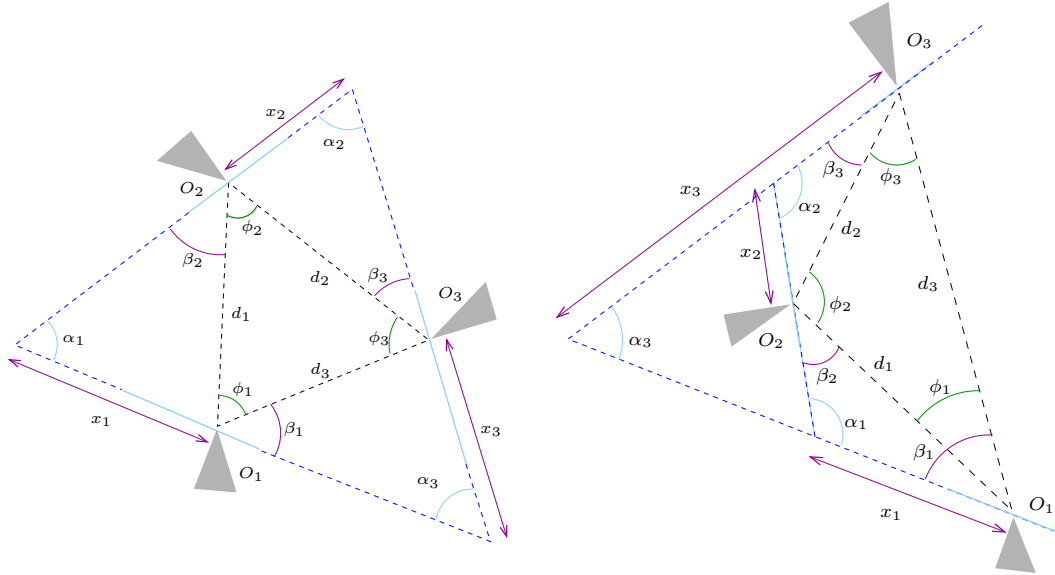


Abbildung 4.6: Drei Roboteranten

### 4.2.3 Verschiedene Kontaktpaartypen

Wenn die beteiligten Kontaktpaare nicht alle den gleichen Typ haben, muss man die Plazierungen anders finden. Seien zuerst  $O_1$  und  $O_2$  Kontaktpaare mit Roboterecken und  $O_3$  mit einer Roboterante (siehe Abbildung 4.7). Wenn man den Roboter unter Aufrechterhaltung der Kontakte  $O_1$  und  $O_2$  bewegt, ändern sich auch die Koordinaten der  $O_3$ -Kante. Die Winkel, bei denen der Kontakt  $O_3$  entsteht, ergeben sich aus folgenden Gleichungen:

Sei  $\vec{e}_1$  der normierte Vektor in Richtung der  $O_1$ -Kante und  $\vec{e}_2$  analog. Die Koordinaten der Endpunkte der  $O_3$ -Kante  $p_1$  und  $p_2$  lassen sich als Funktion von Winkel  $\beta$  beschreiben:

$$\vec{p}_1 = \vec{S} + \left( \frac{d_1}{\sin \alpha} \sin \beta \right) \vec{e}_1 + d_{13} \overline{v_1(\beta)},$$

wobei  $v_1(\beta)$  der Vektor  $e_1$  gedreht um den Winkel  $\beta + \alpha - \phi_1$  gegen den Uhrzeigersinn ist:

$$v_{1x}(\beta) = e_{1x} \cos(\beta + \alpha - \phi_1) - e_{1y} \sin(\beta + \alpha - \phi_1),$$

$$v_{1y}(\beta) = e_{1x} \sin(\beta + \alpha - \phi_1) + e_{1y} \cos(\beta + \alpha - \phi_1).$$

Mit der Drehung von  $\vec{e}_2$  um  $\pi - \beta - \phi_2$  im Uhrzeigersinn erhalten wir auch die Gleichungen für  $p_2$ . Weil  $p_3$  zwischen  $p_1$  und  $p_2$  liegt, gilt für ein  $\lambda \in [0, 1]$ :

$$\vec{p}_3 = \vec{p}_1 + \lambda(\vec{p}_2 - \vec{p}_1)$$

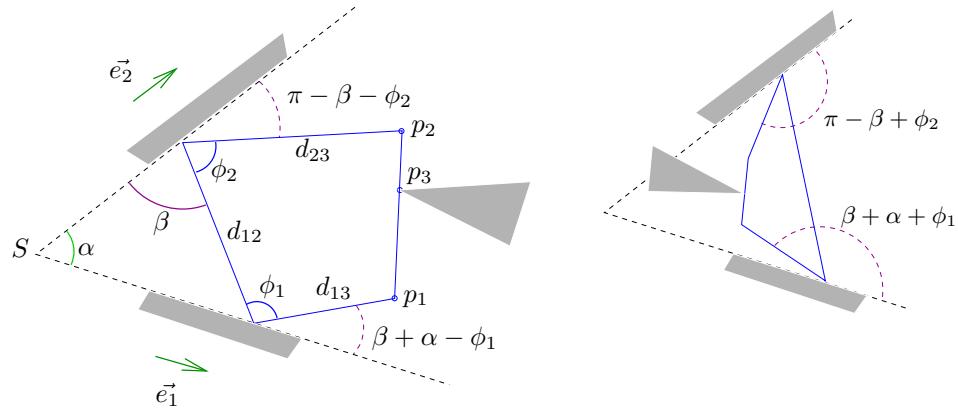


Abbildung 4.7: Zwei Roboterecken und eine Roboterkannte

Da die Koordinaten von  $p_3$  aber bekannt sind (die Ecke von  $O_3$ ), hat man zwei Gleichungen mit zwei Unbekannten ( $\beta$  und  $\lambda$ ):

$$p_{1x}(\beta) + \lambda \cdot (p_{2x}(\beta) - p_{1x}(\beta)) = p_{3x},$$

$$p_{1y}(\beta) + \lambda \cdot (p_{2y}(\beta) - p_{1y}(\beta)) = p_{3y}.$$

Aus jeder Lösung für  $\beta$  findet man eine Orientierung, bei der eine Platzierung mit diesen drei Kontaktpaaren existiert.

Wenn die Ecke von  $O_3$  auf der anderen Seite von  $d_1$  liegt, sind die Drehwinkel für  $\vec{v}_1$  und  $\vec{v}_2$  entsprechend  $\alpha + \beta + \phi_1$  und  $\pi - \beta + \phi_2$ , sonst ändert sich bei der Berechnung nichts.

Jetzt bleibt noch der Fall mit zwei Roboterkannten und einer Roboter-ecke; dieser ist aber symmetrisch: wenn man den Roboter festhält und die Hindernisse bewegt, so dass zwei Kontakte mit den Roboterkannten erhalten bleiben, bekommt man dieselben Gleichungen (siehe Abbildung 4.8). Daraus findet man den Winkel  $\beta$  und die dazugehörige Platzierung(en).

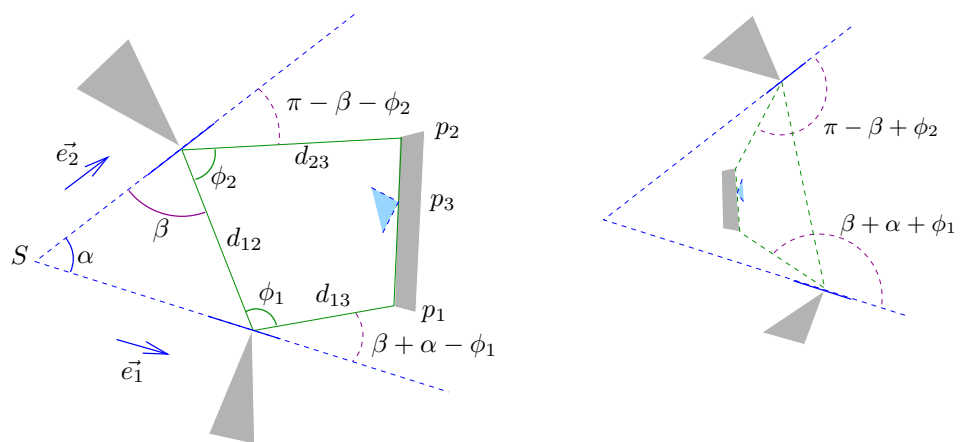


Abbildung 4.8: Zwei Roboteranten und eine Roboterecke





# Kapitel 5

## Sonderfälle

### 5.1 Parallele Kanten

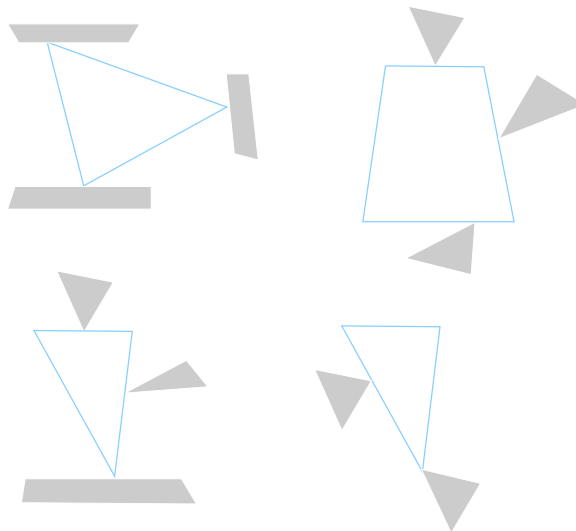


Abbildung 5.1: Sonderfälle

Es gibt zwei Gruppen von kritischen Plazierungen, die noch nicht betrachtet wurden: Plazierungen mit parallelen Tangenten und Plazierungen mit Ecke/Ecke-Kontaktpaaren. Für solche Kontaktpaare gibt es keine Beschränkungsfunktion und sie werden als Sonderfälle behandelt.

Zwei parallele Hinderniskanten kann man entweder als Degeneriertheit verbieten oder durch Perturbation minimal verändern, so dass sie nicht mehr parallel sind (siehe [2],[1]). Das gleiche gilt auch für zwei parallele Roboteranten.

Es kann aber Plazierungen geben, bei denen eine Roboterante und eine Hinderniskante parallel sind. Für diese  $O(mn)$  Orientierungen kann man mit

dem Algorithmus [8] für reine Translationsbewegungen den Konfigurationsraum in  $O(mn \log(mn))$  bestimmen und aus seinen Knoten die entsprechenden Plazierungen finden, insgesamt also in  $O(m^2 n^2 \log(mn))$  Zeit, was in die Gesamtlaufzeit passt.

## 5.2 Ecke/Ecke-Kontaktpaare

Um kritische Plazierungen mit Ecke/Ecke-Kontaktpaaren zu finden, kann man für jedes der  $O(nm)$  Ecke/Ecke-Kontaktpaare und jedes der  $O(nm)$  Ecke/Kante-Kontaktpaare testen, ob es eine entsprechende Plazierung gibt (insgesamt also  $O(n^2 m^2)$  Tests). Den Test kann man wie in Kapitel 2 in einen Halbebenentest, einen Abstandstest und einen Winkeltest unterteilen. Im Halbebenentest überprüft man, ob die Hindernisse füreinander sichtbar sind. Bei dem Abstandstest findet man eine oder zwei mögliche Orientierungen für eine Plazierung, wenn sie existiert. Beim Winkeltest muss man noch überprüfen, ob die Nachbarkanten der beteiligten Ecken bei diesen Orientierungen einen Schnitt verursachen.

Der Abstandstest sieht wie folgt aus:

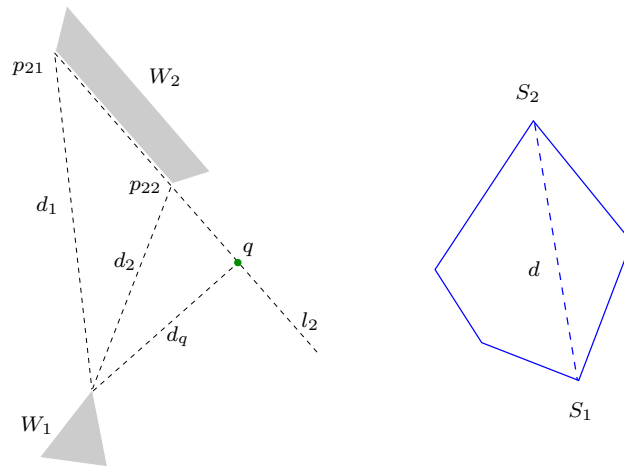


Abbildung 5.2: Fall 1.  $S_2$  ist eine Roboterecke

Sei  $O_1 = (S_1, W_1)$  ein Ecke/Ecke-Kontaktpaar und  $O_2 = (S_2\text{-Roboterecke}, W_2\text{-Hinderniskante})$ . Sei  $l_2$  die Gerade, die  $W_2$  enthält,  $p_{21}$ ,  $p_{22}$  die Endpunkte von der  $W_2$ -Kante,  $d = |S_1 S_2|$  der Abstand zwischen den Roboterecken,  $d_1 = |W_1 p_{21}|$ ,  $d_2 = |W_1 p_{22}|$ ,  $d_q = |W_1 l_2|$  und  $q$  der Punkt auf  $l_2$ , in dem  $d_q$  angenommen wird (siehe Abbildung 5.2). In folgenden Fällen gibt es keine Plazierung:

- $d < d_q$ ,
- $d < \min(d_1, d_2)$  und  $q \notin W_2$ ,
- $d > \max(d_1, d_2)$

(siehe Abbildung 5.3).

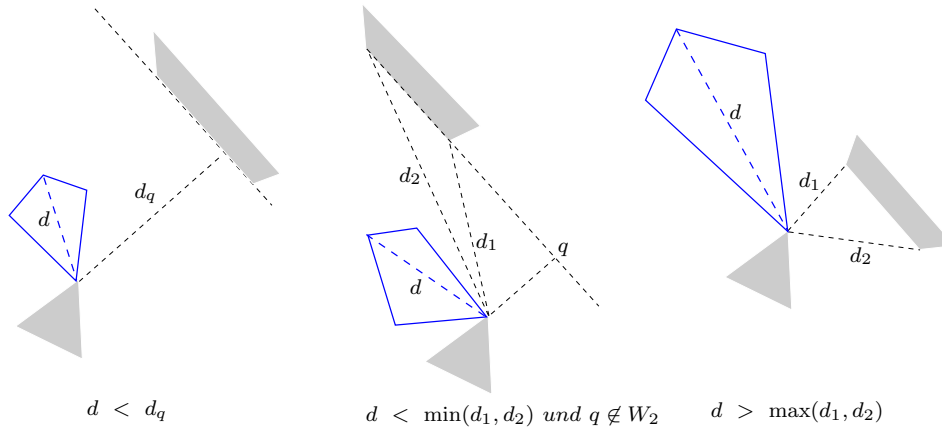


Abbildung 5.3: Fall 1. Abstandstest

Sonst gibt es eine oder zwei Orientierungen, bei denen die zwei Kontakte entstehen. Die entsprechenden Winkel findet man aus  $d$ ,  $d_1$  und  $\angle(\overrightarrow{W_1 p_{21}}, \overrightarrow{W_2})$ .

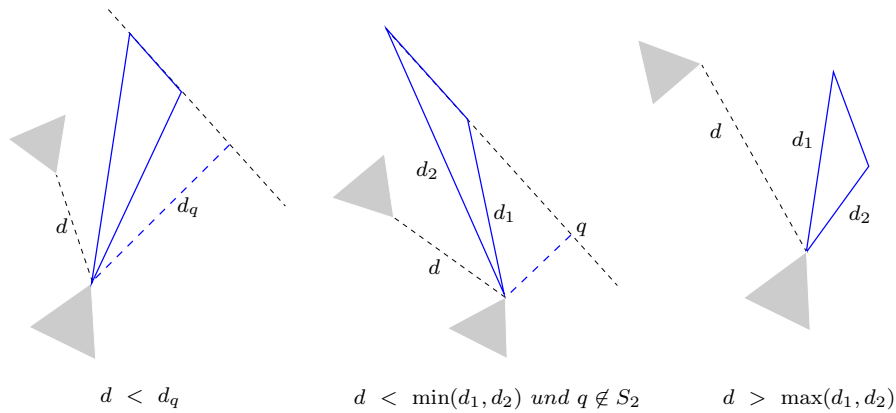


Abbildung 5.4: Fall 2. Abstandstest

Im zweiten Fall besteht  $O_2$  aus einer Roboterseite und einer Hindernisecke. Sei  $l_2$  die Gerade, die  $S_2$  enthält,  $p_{21}, p_{22}$  die Endpunkte von der  $S_2$ -Kante,  $d = |W_1 W_2|$ ,  $d_1 = |S_1 p_{21}|$ ,  $d_2 = |S_1 p_{22}|$ ,  $d_q = |S_1 l_2|$  und  $q$  der Punkt auf  $l_2$ , in dem  $d_q$  angenommen wird. Dann sieht der Abstandstest genauso aus wie oben (siehe Abbildung 5.4), und die entsprechenden Orientierungen (eine oder zwei) findet man auch analog.

### 5.3 Nicht kritische Plazierungen

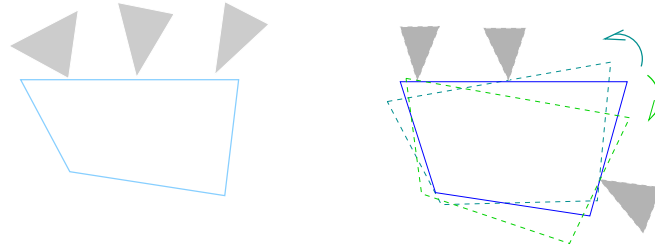


Abbildung 5.5: Nicht kritische Plazierungen

Die Plazierungen auf Abbildung 5.5 sind zwar der Definition nach kritisch, aber für die Bewegungsplanung nicht kritisch: bei diesen Orientierungen kann sich der Roboter immer noch bewegen und die Struktur des Konfigurationsraumes ändert sich nicht.

# Kapitel 6

## Das Java-Applet

### 6.1 Das Benutzer-Interface

Das Applet besteht aus zwei Editor-Fenstern - für Hindernisse und für den Roboter, einem Fenster für die Funktionsgraphen von  $f_{O_i O_j}$  und ein paar Buttons (siehe Abbildung 6.1).

In den Editor-Fenstern kann man Polygone mit der Maus zeichnen oder editieren, oder aus dem “Random Polygon Tool” importieren. Im Roboter-Editor kann man nur ein konvexes Polygon einfügen.

Wenn der Roboter und die Hindernisse gesetzt sind, kann man auf “calculate all” klicken, um die Plazierungen zu berechnen. Wenn sich irgendwelche Kontaktpaare beschränken, werden  $f_{O_i O_j}$  im Funktionsfenster angezeigt. Mit “next f” und “prev f” kann man die Liste aller Funktionen durchgehen. Dabei werden in den Textfeldern die Nummern  $i$  und  $j$  der Kontaktpaare ausgegeben, im Funktionsfenster wird die entsprechende Funktion farbig und dick gezeichnet (wenn der Definitionsbereich nicht zu klein ist) und in der Scene werden ein paar (drei) Plazierungen aus jedem Intervall im Definitionsbereich angezeigt. Kante und Ecke von  $O_1$  wird rot, und von  $O_2$  - grün markiert.

Mit den weiteren Buttons kann man die kritischen Plazierungen anzeigen lassen: entweder alle zusammen (“placings”) oder einzeln (“>” und “<”). Bei einer einzelnen Plazierung (von Typ 1) wird im Funktionsfenster das entsprechende Arrangement gezeichnet und der Schnittpunkt markiert.

Mit “show ref.edge” kann man sich die Referenzkante des Roboters anzeigen lassen (um den Wert von  $\theta$  bei einer Plazierung abzuschätzen).

Auf die Konsole wird unter anderem ausgegeben, welche Kontaktpaare welche beschränken (gekennzeichnet durch ihre Nummer), sortiert nach Typ. Diese Nummern kann man in die Textfelder für  $O_1$  und  $O_2$  eingeben, um diese bestimmte zwei Kontaktpaare (und die Funktionen) zu sehen. Wenn es Schnittpunkte auf einer unteren Kontur gibt, werden in der Konsole auch zusätzliche Information dazu sichtbar: welche Kontaktpaare und Funktionen

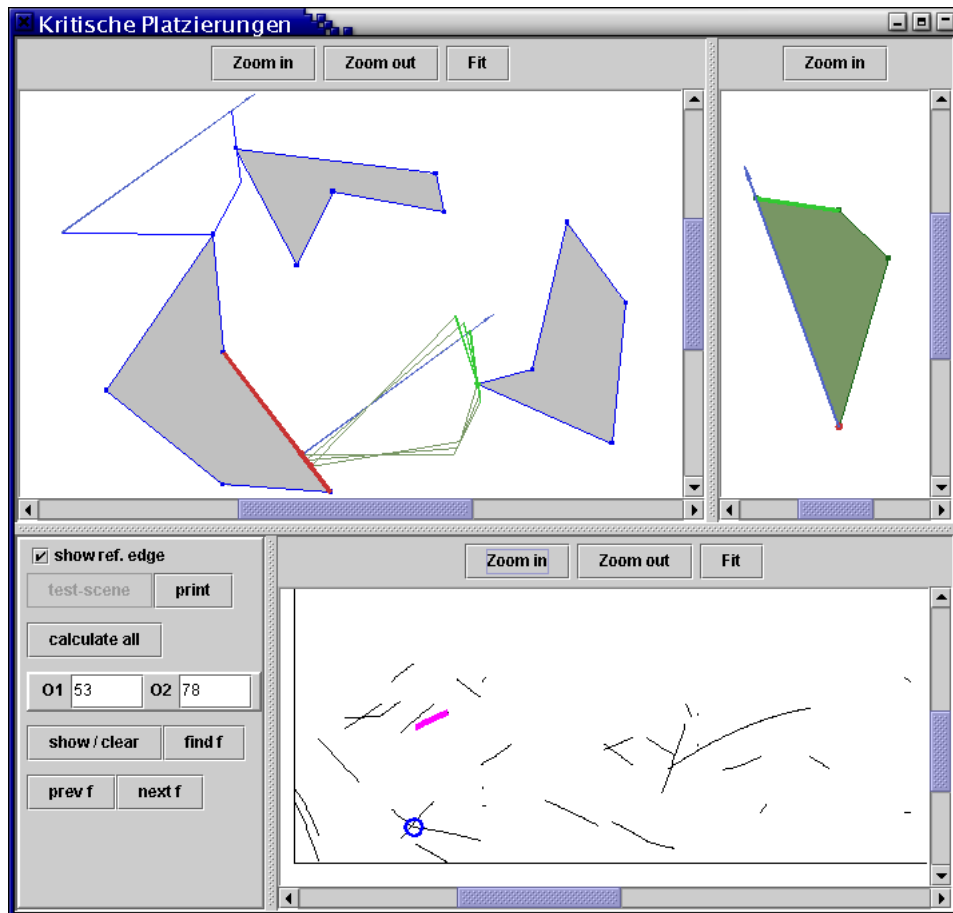


Abbildung 6.1: Applet

(A oder B) beteiligt sind.

## 6.2 Die Programmstruktur

Die Hauptklasse, die alles verwaltet und die Komponenten zusammenfügt, heißt `CrPlacings`. Sie ist von `JFrame` abgeleitet. Hier werden alle Fenster initialisiert, die Ereignisse abgefragt und ggf. weitergegeben und die Funktionen  $f_{O_i O_j}$  berechnet.

Für Editorenfenster sind `CPSceneEditor` und `CPRobotEditor` zuständig, sie sind von `Polygon2SceneEditor` abgeleitet. Hier gibt es ein paar Funktionen zum Zeichnen von Plazierungen und Kontaktpaaren. `CPRobotScene` ist von `Polygon2Scene` abgeleitet, um im `Robotereditor` nur ein konvexes Polygon zu erlauben.

Die Klasse `CPFunktions` ist von `Polygon2SceneWorker` abgeleitet. Sie verwaltet die Funktionen  $f_{O_i O_j}$ , nachdem sie in `CrPlacings` berechnet werden und zeichnet sie. Hier gibt es auch die speziellen Berechnungen für die untere Kontur: Ableitungen, Schnittpunkte, Umkehrfunktionen. Die Kontur selbst wird mit der Klasse `anja.analysis.RealEnvelope` erstellt.

Außerdem gibt es noch Klassen für eine Intervallmenge (Definitionsbereiche von Funktionen), für ein Kontaktpaar und für eine Platzierung.

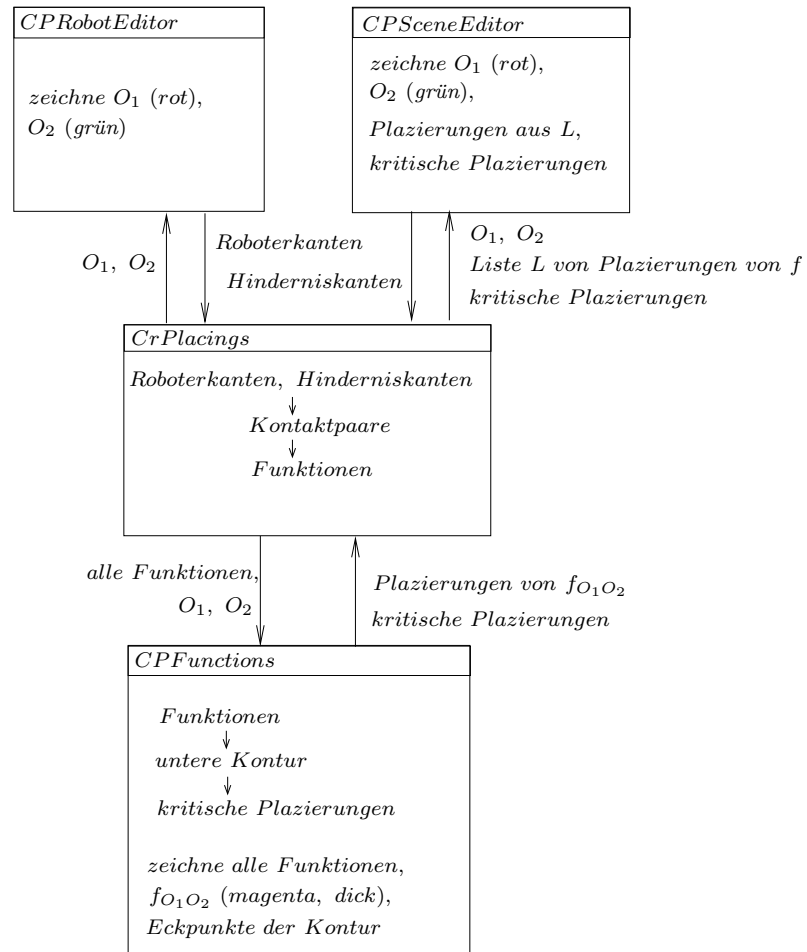


Abbildung 6.2: Die Programmstruktur



# Kapitel 7

## Komplexität und Schlußbemerkung

### 7.1 Komplexität

Insgesamt braucht man für die Berechnung der kritischen Plazierungen  $O(mn\lambda_6(mn)\log(mn))$  Zeit. Das setzt sich wie folgt zusammen:  $n$  ist die Anzahl der Hinderniskanten,  $m$  der Roboteranten, also gibt es  $O(nm)$  Kontaktpaare.

- Für je zwei Kontaktpaare wird getestet, ob sie sich beschränken und wenn ja, werden die entsprechenden Funktionen in die Arrangements  $A_L(O_i)$ ,  $A_R(O_i)$  gespeichert. Also  $O(n^2m^2)$  Tests, jeder in  $O(1) \Rightarrow O(n^2m^2)$ .
- Für jedes der  $O(nm)$  vielen Arrangements wird die untere Kontur bestimmt (mit Divide & Conquer). Weil sich zwei Funktionen maximal vier mal schneiden ergibt sich  $O(\lambda_6(mn)\log(mn))$  pro Arrangement (siehe [11]), also  $O(mn\lambda_6(mn)\log(mn))$  insgesamt.
- Sortierung der  $O(nm)$  Listen der Eckpunkte in eine Liste (für die zyklischen Tripel).  
Wegen  $\log(\lambda_6(mn)) \leq \log(m^2n^2) = 2\log(mn)$ :  
 $m n \lambda_6(mn) \log(m n \lambda_6(mn)) \in O(m n \lambda_6(mn) \log(mn))$ .
- Die Behandlung der Sonderfälle (Parallele Kanten und Ecke/Ecke-Kontaktpaare) geht in  $O(m^2n^2 \log(nm))$ .

### 7.2 Schlußbemerkung

Die Teilaufgaben des Algorithmus von Kedem und Sharir sind nicht einfach zu lösen. Besonders bei den Tests, ob es eine Platzierung mit zwei Kontaktpaaren gibt, und bei der Berechnung der Funktionen muss man viele

verschiedene Fälle berücksichtigen. Nicht trivial sind auch die Bestimmung der Schnittpunkte für die untere Kontur, die Ermittlung der Orientierungen, die zu einem Zyklus gehören, und die Behandlung der Sonderfälle. Andererseits kann man auch nicht erwarten, dass in einem Algorithmus jedes Detail und alle Sonderfälle beschrieben werden, wenn er an sich schon kompliziert genug ist.

# Literaturverzeichnis

- [1] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9(1):66–104, 1990.
- [2] I. Emiris and J. Canny. An efficient approach to removing geometric degeneracies. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 74–82, 1992.
- [3] R. Jesse. *Das Einsteigerseminar Java-Swing*. Verlag moderne industrie Buch, Landsberg, 2002.
- [4] T. Kamphans and R. Klein. *Bewegungsplanung für Roboter*. Vorlesungsskript, Universität Bonn, Institut für Informatik, 2001.
- [5] K. Kedem and M. Sharir. An efficient motion planning algorithm for a convex rigid polygonal object in 2-dimensional polygonal space. *Discrete Comput. Geom.*, 5:43–75, 1990.
- [6] K. Kedem, M. Sharir, and S. Toledo. On critical orientations in the Kedem-Sharir motion planning algorithm for a convex polygon in the plane. *Discrete Comput. Geom.*, 17:227–240, 1997.
- [7] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.
- [8] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Trans. Comput.*, C-32:108–120, 1983.
- [9] L. Rade and B. Westergren. *Springers Mathematische Formeln*. Springer-Verlag, Berlin, 2000.
- [10] J. T. Schwartz and M. Sharir. On the “piano movers” problem I: The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Commun. Pure Appl. Math.*, 36:345–398, 1983.
- [11] M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, New York, 1995.

**Erklärung**

Hiermit erkläre ich, dass ich diese Arbeit selbständig durchgeführt habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.