# Robotik II: Screw Theorie, Hände und Greifen

R. Haschke, J. Steil

letzte Änderung 31. März 2004

All rights reserved.

# Einleitung

Das vorliegende Skript ist die Zusammenfassung der Vorlesung Robotik II, die ich im Wintersemester 2003/04 an der Technischen Fakultät der Universität Bielefeld gehalten habe. Das Skript erhebt weder Anspruch auf Vollständigkeit noch auf absolute Korrektheit. Korrekturen und andere Verbesserungsvorschläge nehme ich gerne entgegen (rhaschke@techfak.uni-bielefeld.de).

Bielefeld, 31. März 2004

Robert Haschke

# Inhaltsverzeichnis

1		ew-The sionen	eorie: Exponentialdarstellung von homogenen Transfor-	5	
	1.1		age eines festen Körpers	5	
	1.2		sentation von Rotationen	6	
		1.2.1	Exponentielle Koordinaten		
		1.2.2	Euler ZYZ-Winkel	8	
		1.2.3	Euler ZYX-Winkel		
		1.2.4	Quaternionen	9	
	1.3		neine Bewegungen (Rotation und Translation)		
	1.0	1.3.1	,	12	
		1.3.1		12 12	
			Ŭ		
	1 4	1.3.3		13	
	1.4			14	
	1.5			15	
	1.6			17	
	1.7	Jacobi	-Matrix	20	
	1.8	hes: Kräfte und Drehmomente	22		
		1.8.1	screw-Koordinaten von wrenches	23	
		1.8.2	Dualität von twists und wrenches	23	
	1.9	Transp	ponierte Jacobi-Matrix	24	
<b>2</b>	Mel	hrfingr	iges Greifen	25	
	2.1			25	
		2.1.1		 25	
		2.1.2		 27	
	2.2			28	
	2.2	2.2.1		28	
		2.2.1		20 30	
	0.0	2.2.3	·	32	
	2.3	3 Die Greif-Bedingung			

	2.4	Manipulierbarkeit	36			
	2.5	Strukturelle Kräfte	37			
	2.6	Zusammenfassung	39			
	2.7	Optimierung von Kontaktkräften $\vec{f_c}$	39			
	2.8	Anwendungsorientierung und Virtuelle Kontakte	43			
		2.8.1 Task-Ellipsoid	43			
		2.8.2 Virtuelle Kontakte	44			
3	$\operatorname{Gef}$	ügiges Greifen	45			
4	Inve	inverse Hand-Kinematik und Griffplanung 45				
5	Ant	Antriebskonzepte				
6	Lernen zu Greifen					
7 Reinforcement Learning (Verstärkungslernen)						
	7.1	Markovsche Entscheidungsprozesse (MDP)	47			
	7.2	Definitionen des kumulierten Rewards	48			
	7.3	Policy Iteration	50			
	7.4	Value Iteration	50			
	7.5	Q-Lernen	51			
	7.6	Exploration vs. Exploitation	52			
	7.7	Temporal Difference Learning: $TD(\lambda)$	53			
	7.8	Generalisierung	54			
	7.9	Self-Organizing Map (SOM) / Neural Gas	55			
	7.10	Verbleibende Probleme	56			
8	Imitation Learning					
	8.1	Was soll nachgeahmt werden?	57			
	8.2	Wie soll eine Handlungssequenz in sinnvolle Teilstücke (primitives) zerlegt werden?	58			
	8.3		59			

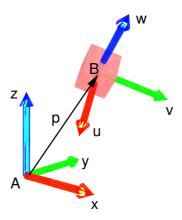


Abbildung 1: Körper mit körperfesten KS B, relativ zum festen Welt-KS A

# 1 Screw-Theorie: Exponentialdarstellung von homogenen Transformationen

# 1.1 Die Lage eines festen Körpers

**Definition 1.1** Ein fester Körper (rigid body) ist eine Menge von Punkten, deren relative Lage sich nicht ändern kann. Die Lage all dieser Punkte ist daher relativ zu einem körperfesten Koordinatensystems (KS) zeitlich konstant.

Die Lage eines festen Körpers (und damit all seiner Punkte) im dreidimensionalen Raum ist durch den Ursprung und die Orientierung seines körperfesten KS  $(B_{p,u,v,w})$  relativ zu einem festen Welt-KS  $(A_{0,x,y,z})$  bestimmt (Abb. 1).

Die homogene Transformation, die das Welt-KS A in das körperfeste KS B überführt, ist gegeben durch:

$$T_{ab} = \begin{pmatrix} xu & xv & xw & p_x \\ yu & yv & yw & p_y \\ zu & zv & zw & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} R_{ab} & \vec{p} \\ 0 & 1 \end{pmatrix}$$

Gleichzeitig transformiert  $T_{ab}$  die homogenen Koordinaten eines Punktes Q bzgl. des körperfesten KS ( $\bar{q}_{u,v,w} = [q_u, q_v, q_w, 1]^t$ ) in Koordinaten desselben Punktes bzgl. des Welt-KS ( $\bar{q}_{x,y,z} = [q_x, q_y, q_z, 1]^t$ ):

$$\bar{q}_{x,y,z} = T_{ab} \, \bar{q}_{u,v,w}$$
 bzw.  
 $\vec{q}_a = R_{ab} \, \vec{q}_b + \vec{p}$ .

**Bemerkung:** Die homogenen Koordinaten von Richtungsvektoren werden anstatt mit einer Eins mit einer Null ergänzt, so dass bei der Transformation der translatorische Anteil  $\vec{p}$  ignoriert wird. Im Gegensatz zu Punktvektoren haben Richtungsvektoren keine Verankerung im Raum, sondern können beliebig verschoben werden.

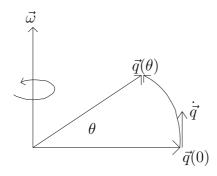


Abbildung 2: Drehung um die Drehachse  $\vec{\omega}$  und den Winkel  $\theta$ .

Die perspektivische Transformation und die Skalierung würden die Geometrie von festen Körpern verändern, so dass sie keine gültigen Transformationen für Festkörper darstellen.

# 1.2 Repräsentation von Rotationen

**Definition 1.2** Die Menge  $SO(3) = \{R \in \mathbb{R}^{n \times n} | RR^t = 1, \det R = 1\}$  heißt spezielle orthogonale Gruppe (bzgl. der Matrizenmultiplikation). Sie enthält die Menge der Rotationsmatrizen, die die Rechtshändigkeit von KS erhalten.

Im folgenden suchen wir eine geeignete Repräsentation für solche Matrizen. Offenbar genügen drei freie Parameter (Freiheitsgrade), um Rotationsmatrizen zu beschreiben.

### 1.2.1 Exponentielle Koordinaten

**Notation:** Das Kreuzprodukt  $\vec{a} \times \vec{b}$  kann auch als lineare Abbildung  $\hat{a}$  aufgefasst werden, die auf den Vektor  $\vec{b}$  angewandt wird:

$$\vec{a} \times \vec{b} = \hat{a} \cdot \vec{b}$$
 wobei  $\hat{a} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} = -\hat{a}^t$ .

Wir betrachten eine Rotation um eine beliebige Drehachse  $\vec{\omega}$  ( $\|\vec{\omega}\| = 1$ ) und den Winkel  $\theta$  (s. Abb. 2). Fasst man  $\vec{\omega}$  als Rotationsgeschwindigkeit auf, so lässt sich die Geschwindigkeit eines Punktes  $\vec{q}$  entlang der Kreisbahn schreiben als (differentielle Drehungen):

$$\vec{q}(t) = \vec{\omega} \times \vec{q}(t) = \hat{\omega} \, \vec{q}(t) \,.$$

Integration dieser linearen Differentialgleichung ergibt:

$$\vec{q}(\theta) = \int_0^\theta \hat{\omega} q(t)dt = e^{\hat{\omega}\theta} \vec{q}(0) = R(\hat{\omega}, \theta) \vec{q}(0)$$

wobei  $e^{\hat{\omega}\theta}$  die Exponentialfunktion einer Matrix ist:

$$e^{\hat{\omega}\theta} = \mathbf{1} + \hat{\omega}\theta + \frac{(\hat{\omega}\theta)^2}{2!} + \frac{(\hat{\omega}\theta)^3}{3!} + \dots$$
$$= \mathbf{1} + \hat{\omega}\sin\theta + \hat{\omega}^2(1 - \cos\theta)$$

**Satz 1.3** Die Matrix  $R(\hat{\omega}, \theta) = e^{\hat{\omega}\theta}$  ist eine Rotationsmatrix.

**Definition 1.4** Die Menge der antisymmetrischen Matrizen (skew-symmetric) bezeichnen wir mit  $so(3) = \{S \in \mathbb{R}^{3\times 3} \mid S^t = -S\}.$ 

Die Abbildung  $\vec{\omega} \mapsto \hat{\omega}$  bildet Vektoren  $\vec{\omega} \in \mathbb{R}^3$  bijektiv auf antisymmetrische Matrizen  $\hat{\omega} \in so(3)$ . Die Matrix-Exponentialfunktion exp :  $so(3) \to SO(3)$  bildet antisymmetrische Matrizen auf Rotationsmatrizen ab. Frage: Ist diese Abbildung auch surjektiv, d.h. gibt es zu jeder Rotationsmatrix  $R \in SO(3)$  eine antisymmetrische Matrix  $\hat{\omega}\theta$ , so dass  $e^{\hat{\omega}\theta} = R$ ?

Satz 1.5 (Euler) Jede Rotationsmatrix  $R \in SO(3)$  ist äquivalent zu einer Rotation um eine feste Achse  $\vec{\omega} \in \mathbb{R}^3$  und einen Drehwinkel  $\theta \in [0, 2\pi)$ .

**Beweis**: Mit den Bezeichnungen  $c_{\theta} = \cos \theta$ ,  $s_{\theta} = \sin \theta$ ,  $\eta_{\theta} = 1 - \cos \theta$  gilt:

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \stackrel{!}{=} e^{\hat{\omega}\theta} = \begin{pmatrix} \omega_1^2 \eta_{\theta} + c_{\theta} & \omega_1 \omega_2 \eta_{\theta} - \omega_3 s_{\theta} & \omega_1 \omega_3 \eta_{\theta} + \omega_2 s_{\theta} \\ \omega_1 \omega_2 \eta_{\theta} + \omega_3 s_{\theta} & \omega_2^2 \eta_{\theta} + c_{\theta} & \omega_2 \omega_3 \eta_{\theta} - \omega_1 s_{\theta} \\ \omega_1 \omega_3 \eta_{\theta} - \omega_2 s_{\theta} & \omega_2 \omega_3 \eta_{\theta} + \omega_1 s_{\theta} & \omega_3^2 \eta_{\theta} + c_{\theta} \end{pmatrix}$$

Für die Spur der beiden Matrizen gilt:

$$sp R = r_{11} + r_{22} + r_{33} \stackrel{!}{=} 3\cos\theta + (1 - \cos\theta) \sum \omega_i^2 = 1 + 2\cos\theta$$

$$\Rightarrow \theta = \cos^{-1}\left(\frac{sp R - 1}{2}\right) \in [0, \pi].$$

Diese Gleichung kann nach  $\theta$  aufgelöst werden, weil die Eigenwerte  $\lambda_i$  von R den Betrag Eins haben (Übung) und daher gilt:

$$-1 \le \operatorname{sp} R = \sum \lambda_i \le 3.$$

Um die Rotationsachse zu bestimmen, nutzen wir die restlichen Matrixeinträge:

$$\begin{vmatrix}
r_{32} - r_{23} &= 2\omega_1 s_{\theta} \\
r_{13} - r_{31} &= 2\omega_2 s_{\theta} \\
r_{21} - r_{12} &= 2\omega_3 s_{\theta}
\end{vmatrix} \qquad \stackrel{\theta \neq 0}{\Rightarrow} \qquad \vec{\omega} = \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix} \cdot \frac{1}{2s_{\theta}} \tag{1.1}$$

Falls R=1 ist, folgt sp R=3 und daher  $\theta=0$ . In diesem Fall ist  $\vec{\omega}$  beliebig, da  $e^{\hat{\omega}\theta}=e^0=1$ .  $\square$ 

**Bemerkung:** Der Winkel  $\theta$  ist nicht eindeutig bestimmt, da  $\theta \pm 2\pi n$  und  $-\theta \pm 2\pi n$  denselben cosinus besitzen. Wählt man  $-\theta + 2\pi \in [\pi, 2\pi]$ , muss die Drehachse  $\vec{\omega}$  umgedreht werden. D.h. zu  $R \neq \mathbf{1}$  gibt es zwei verschiedene Paare  $(\vec{\omega}, \theta)$  mit  $\theta \in$ 

 $[0,2\pi)$ , so dass  $R={\rm e}^{\hat\omega\theta}$ . Solche Singularitäten sind typisch für jede 3-parametrige Repräsentationen von Rotationen.

Die Singularität bei R=1 kann vermieden werden, wenn man die Trennung von normierter Drehachse  $\vec{\omega}$  und Drehwinkel  $\theta$  aufhebt und stattdessen beliebige  $\vec{\omega} \in \mathbb{R}^3$  betrachtet für deren Norm gilt:  $0 \le ||\vec{\omega}|| \le \pi$ . Allerdings bleibt die Singularität bei  $||\vec{\omega}|| = \pi$  (Drehung um 180°) erhalten.

#### 1.2.2 Euler ZYZ-Winkel

Eine beliebige Rotation  $R_{ab}$  zwischen den KS  $O_a$  und  $O_b$  kann auch durch Aneinanderreihung von Standard-Rotationen erreicht werden:

- 1. Rotiere um den Winkel  $\alpha$  um die z-Achse
- 2. Rotiere um den Winkel  $\beta$  um die neue y-Achse
- 3. Rotiere um den Winkel $\gamma$ um die neue z-Achse

$$R_{ab} = R_{z,\alpha} \cdot R_{y,\beta} \cdot R_{z,\gamma}$$

Es stellt sich wieder die Frage nach der Surjektivität und den Singularitäten dieser Abbildung.

- $(\alpha, \beta, \gamma) \mapsto R$  ist surjektiv.
- Singularität:  $\beta = 0, \ \alpha = -\gamma \ (R = 1)$

## 1.2.3 Euler ZYX-Winkel

- 1. Rotiere um den Winkel  $\psi$  um die x-Achse (yaw)
- 2. Rotiere um den Winkel  $\theta$  um die alte y-Achse (pitch)
- 3. Rotiere um den Winkel  $\phi$  um die alte z-Achse (roll)

$$R_{ab} = R_{z,\phi} \cdot R_{u,\theta} \cdot R_{x,\psi}$$

- $(\phi, \theta, \psi) \mapsto R$  ist surjektiv.
- Singularitäten:  $\theta = \pm \pi/2$ . (Übung)

#### 1.2.4 Quaternionen

Quaternionen sind 4-dimensionale Vektoren

$$Q = (q_0, \vec{q}) \qquad q_0 \in \mathbb{R}, \quad \vec{q} \in \mathbb{R}^3$$
$$= q_0 + q_1 \hat{i} + q_2 \hat{j} + q_3 \hat{k}, \qquad q_i \in \mathbb{R},$$

deren Addition komponentenweise erfolgt. Man kann sich 1,  $\hat{i}$ ,  $\hat{j}$  und  $\hat{k}$  als Einheitsvektoren dieses 4-dim. Raumes vorstellen. Die distributive und assoziative Multiplikation ist entsprechend folgender Regeln definiert:

• 
$$\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i} \cdot \hat{j} \cdot \hat{k} = -1$$

• 
$$\hat{i} \cdot \hat{j} = -\hat{j} \cdot \hat{i} = \hat{k}$$
,  $\hat{j} \cdot \hat{k} = -\hat{k} \cdot \hat{j} = \hat{i}$ ,  $\hat{k} \cdot \hat{i} = -\hat{i} \cdot \hat{k} = \hat{j}$  (zyklisch)

Weiterhin gilt:

- $Q^* = (q_0, -\vec{q})$  (konjugiertes Quaternion zu  $Q = (q_0, \vec{q})$ )
- $||Q||^2 = QQ^* = Q^*Q = \sum q_i^2$  (Betrag)
- $Q^{-1} = \frac{Q^*}{\|Q\|^2}$  (inverses Quaternion)
- $Q \cdot P = (q_0, \vec{q}) \cdot (p_0, \vec{p}) = (q_0 p_0 \vec{q} \cdot \vec{p}, q_0 \vec{p} + p_0 \vec{q} + \vec{q} \times \vec{p})$  (Multiplikation)
- $Q \cdot (0, \vec{p}) \cdot Q^* = (0, R(\vec{q}, q_0)\vec{p})$

Eine Rotation  $R_{ab} = R(\vec{\omega}, \theta) = e^{\hat{\omega}\theta}$  wird durch das Quaternion  $Q_{ab} = (\cos \frac{1}{2}\theta, \vec{\omega} \sin \frac{1}{2}\theta)$  repräsentiert. Dann gilt für  $R_{ab}R_{bc} = R_{ac}$  auch  $Q_{ab}Q_{bc} = Q_{ac}$ . Die Quaternion-Darstellung erlaubt eine besonders effiziente Berechnung von Verkettungen von Rotationen (Übung).

# 1.3 Allgemeine Bewegungen (Rotation und Translation)

Eine allgemeine Bewegung setzt sich aus einer Translation und einer Rotation zusammen. Wir wollen den für Rotationen eingeführten Formalismus nun auf homogene Transformationen verallgemeinern. Daher definieren wir:

**Definition 1.6** Die Menge  $SE(3) = \{(p, R) | p \in \mathbb{R}^3, R \in SO(3)\}$  heißt spezielle euklidische Gruppe.

In Korrespondenz zur Definition von so(3) definieren wir:

**Definition 1.7** Ein Element  $\hat{\xi}$  der Menge

$$se(3) = \left\{ \begin{bmatrix} \hat{\omega} & \vec{v} \\ 0 & 0 \end{bmatrix} \mid \vec{v} \in \mathbb{R}^3, \quad \hat{\omega} \in so(3) \right\}$$

heißt twist und die Darstellung von  $\hat{\xi}$  als 6-dimensionaler Vektor  $\vec{\xi} = (\vec{v}, \vec{\omega})$  nennen wir twist-Koordinaten.

Im folgenden zeigen wir, dass die Exponentialfunktion die Menge se(3) der twists surjektiv auf die Menge SE(3) der homogenen Transformationen abbildet – genauso wie antisymmetrische Matrizen auf Rotationsmatrizen abgebildet werden. Zunächst ist aber zu zeigen, dass  $e^{\hat{\xi}}$  überhaupt eine homogene Transformation erzeugt.

Satz 1.8 Für jedes  $\hat{\xi} \in se(3)$  liefert  $e^{\hat{\xi}}$  eine homogene Transformation aus SE(3).

# Beweis:

Fall 1)  $\vec{\omega} = 0$  (reine Translation)

$$\hat{\xi}^2 = 0 \qquad \Rightarrow \qquad \hat{\xi}^n = 0 \quad \text{für alle } n \ge 2$$
$$e^{\hat{\xi}\theta} = \mathbf{1}_{4\times 4} + \hat{\xi}\theta = \begin{pmatrix} \mathbf{1}_{3\times 3} & \theta\vec{v} \\ 0 & 1 \end{pmatrix}$$

Fall 2)  $\vec{\omega} \neq 0$ ,  $\|\vec{\omega}\| = 1$  (Rotation und Translation)

Wir transformieren  $\hat{\xi}$  zunächst mittels  $T = \begin{pmatrix} I & \vec{\omega} \times \vec{v} \\ 0 & 1 \end{pmatrix}$  in eine geeignete Form:

$$\hat{\xi}' = T^{-1}\hat{\xi}T = \begin{pmatrix} I & -\vec{\omega}\times\vec{v} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{\omega} & \vec{v} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} I & \vec{\omega}\times\vec{v} \\ 0 & 1 \end{pmatrix} \stackrel{(\|\vec{\omega}\|=1)}{=} \begin{pmatrix} \hat{\omega} & \vec{\omega}\vec{\omega}^t\vec{v} \\ 0 & 0 \end{pmatrix}$$

Nun gilt  $(\hat{\xi}')^n = \begin{pmatrix} \hat{\omega}^n & 0 \\ 0 & 0 \end{pmatrix}$  für  $n \geq 2$ , so dass:

$$e^{\hat{\xi}\theta} = Te^{\hat{\xi}'\theta}T^{-1} = T\begin{pmatrix} e^{\hat{\omega}\theta} & \vec{\omega}\vec{\omega}^t\vec{v}\theta\\ 0 & 1 \end{pmatrix}T^{-1}$$

$$= \begin{pmatrix} e^{\hat{\omega}\theta} & (I - e^{\hat{\omega}\theta})\vec{\omega} \times \vec{v} + \vec{\omega}\vec{\omega}^t\vec{v}\theta\\ 0 & 1 \end{pmatrix} \in SE(3)$$
(1.2)

Es bleibt zu zeigen, dass zu jeder homogenen Transformation ein erzeugender twist existiert (Surjektivität der Abbildung exp :  $se(3) \rightarrow SE(3)$ ).

**Satz 1.9** Zu jedem  $T \in SE(3)$  existiert ein twist  $\hat{\xi} \in se(3)$  und  $\theta \in \mathbb{R}$ , so dass  $e^{\hat{\xi}\theta} = T$  gilt.

### Beweis:

Fall 1) reine Translation 
$$(R = \mathbf{1})$$
: Setze  $\hat{\xi} = \begin{pmatrix} 0 & \vec{p}/\|\vec{p}\| \\ 0 & 0 \end{pmatrix}$  und  $\theta = \|\vec{p}\|$ .

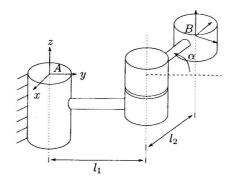


Abbildung 3: Rotation um eine feste Achse im Raum

Fall 2) allg. Bewegung  $(R \neq 1)$ 

Löse die Gleichung  $T=\mathrm{e}^{\hat{\xi}\theta}$ . Zunächst berechne  $\vec{\omega}$  wie in Theorem 1.5. Dann löse die Gleichung

$$p = (I - e^{\hat{\omega}\theta})\hat{\omega}\vec{v} + \theta\vec{\omega}\vec{\omega}^t\vec{v} = M\vec{v} \quad \text{mit} \quad M = (I - e^{\hat{\omega}\theta})\hat{\omega} + \theta\vec{\omega}\vec{\omega}^t.$$

Es lässt sich zeigen, dass die Matrix M invertierbar ist, falls  $\theta \in (0, 2\pi)$ . Also folgt  $\vec{v} = M^{-1}\vec{p}$ .  $\square$ 

**Bemerkung:** Die Abbildung ist – wie auch schon exp :  $so(3) \to SO(3)$  – nicht eindeutig. Ausserdem ist der Vektor  $\vec{v}$  im twist i.a. nicht als Translationsvektor interpretierbar; er beinhaltet sowohl Komponenten von  $\vec{p}$  als auch von R, wie das folgende Beispiel zeigt.

**Beispiel 1.10** Wir betrachten die homogene Transformation  $T_{ab}$ , die das feste KS A mittels folgender elementarer Transformationen in das KS B überführt (Abb. 3):

- Translation um  $l_1$  entlang der  $y_A$ -Achse
- ullet Rotation um  $\alpha$  um die neue z-Achse
- Translation um  $l_2$  entlang der neuen y-Achse

$$T_{ab}(\alpha) = T_{l_1,y} R_{\alpha,z} T_{l_2,y} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 & -l_2 \sin \alpha \\ \sin \alpha & \cos \alpha & 0 & l_1 + l_2 \cos \alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} R_{ab} & \vec{p}_{ab} \\ 0 & 1 \end{pmatrix}$$

Um den zugehörigen twist zu finden, berechnen wir zunächst den Rotationsteil  $e^{\hat{\omega}\theta} = R_{ab}$ . Die Rotation findet offenbar um die z-Achse statt, so dass folgt:  $\vec{\omega} = [0, 0, 1]^t$  bzw.  $\hat{\omega} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$  und  $\theta = \alpha$ . Um den linearen Anteil  $\vec{v}$  zu bestimmen, müssen wir die Gleichung

$$M\vec{v} = [(I - e^{\hat{\omega}\theta})\hat{\omega} + \vec{\omega}\vec{\omega}^t\theta]\vec{v} = \vec{p}_{ab} = \begin{pmatrix} -l_2\sin\alpha\\ l_1 + l_2\cos\alpha\\ 0 \end{pmatrix}$$

lösen. Für M gilt:

$$M = \begin{pmatrix} 1 - \cos \alpha & \sin \alpha \\ -\sin \alpha & 1 - \cos \alpha \\ & & 1 - 1 \end{pmatrix} \hat{\omega} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \alpha = \begin{pmatrix} \sin \alpha & \cos \alpha - 1 & 0 \\ 1 - \cos \alpha & \sin \alpha & 0 \\ 0 & 0 & \alpha \end{pmatrix}$$

so dass folgt:

$$\vec{v} = M^{-1} \vec{p}_{ab} = \begin{pmatrix} \frac{\sin \alpha}{2(1 - \cos \alpha)} & \frac{1}{2} & 0\\ -\frac{1}{2} & \frac{\sin \alpha}{2(1 - \cos \alpha)} & 0\\ 0 & 0 & \frac{1}{\alpha} \end{pmatrix} \begin{pmatrix} -l_2 \sin \alpha\\ l_1 + l_2 \cos \alpha\\ 0 \end{pmatrix} = \begin{pmatrix} \frac{\frac{1}{2}(l_1 - l_2)}{\frac{\sin \alpha}{2(1 - \cos \alpha)}}(l_1 + l_2)\\ 0 & 0 \end{pmatrix}$$

# 1.3.1 Interpretation von homogenen Transformationen

Homogene Transformationen können auf zwei verschiedene Arten interpretiert werden. Zum einen können wir eine homogene Transformation als Koordinaten-Transformation  $T_{ab}: B \to A$  von einem (z.B. körperfesten) Koordinatensystem B in ein anderes (z.B. weltfestes) KS A auffassen. Die homogenen Koordinaten eines Punktes  $\vec{q}$  bzgl. des KS B werden dann mittels  $T_{ab}$  in die zugehörigen Koordinaten desselben Punktes bzgl. des KS A transformiert:

$$\bar{q}_a = T_{ab}\bar{q}_b$$

Zum anderen kann dieselbe homogene Transformation  $T_{ab}: A \to A$  als Bewegung aufgefasst werden, die das KS A in das KS B überführt. In diesem Fall werden die Koordinaten des Punktes  $\vec{q}$  vor der Bewegung in die Koordinaten nach der Bewegung überführt, wobei beides mal auf das KS A Bezug genommen wird:

$$\bar{q}'_a = T_{ab}\bar{q}_a$$

### 1.3.2 Rotation um eine beliebige Achse im Raum

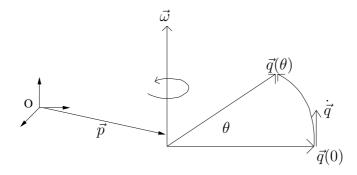


Abbildung 4: Drehung um die Drehachse  $l = \{\vec{p} + \lambda \vec{\omega} \mid \lambda \in \mathbb{R}\}$  im Raum (Drehwinkel  $\theta$ ).

Wir betrachten eine Rotation um eine beliebige Drehachse l im Raum, deren Richtung durch  $\vec{\omega}$  und deren Position durch einen Punkt  $\vec{p}$  auf der Achse bestimmt wird:

 $l = \{\vec{p} + \lambda \vec{\omega} \mid \lambda \in \mathbb{R}\}$  (Abb. 4). Die Koordinatentransformation  $T_{ab}(\theta) : B \to A$  hat dann folgende Form:

$$T_{ab}(\theta) = T_{\vec{p}} \cdot T_{R_{\vec{\omega},\theta}} = \begin{pmatrix} \mathbf{1} & \vec{p} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_{\vec{\omega},\theta} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_{\vec{\omega},\theta} & \vec{p} \\ 0 & 1 \end{pmatrix} \quad \text{und} \quad \bar{q}_a(\theta) = T_{ab}(\theta)\bar{q}_b.$$

Wir wollen nun die reine Bewegung  $R_{\vec{\omega}}(\theta): A \to A$  von der reinen Koordinatentransformation  $T_{ab}(0): B \to A$  trennen:

$$T_{ab}(\theta) = \underbrace{T_{\vec{p}} \cdot T_{R_{\vec{\omega},\theta}} \cdot T_{\vec{p}}^{-1}}_{= R_{\vec{\omega}}(\theta)} \cdot \underbrace{T_{\vec{p}} \cdot T_{R_{\vec{\omega},0}}}_{T_{ab}(0)}$$

Der twist, der die Bewegung  $R_{\vec{\omega}}(\theta)$  erzeugt, hat nun eine besonders einfache Form:

$$R_{\vec{\omega}}(\theta) = e^{\hat{\xi}_R \cdot \theta}$$
 mit  $\hat{\xi}_R = \begin{pmatrix} \hat{\omega} & -\vec{\omega} \times \vec{p} \\ 0 & 0 \end{pmatrix}$  bzw.  $\xi_R = (-\vec{\omega} \times \vec{p}, \vec{\omega})$ 

Um dies zu beweisen, setzen wir  $\hat{\xi}_R$  in (1.2) ein und nutzen  $R = R_{\vec{\omega},\theta} = e^{\hat{\omega}\theta}$ :

$$e^{\begin{pmatrix} \hat{\omega} & -\vec{\omega} \times \vec{p} \\ 0 & 0 \end{pmatrix} \theta} = \begin{pmatrix} R & -(\mathbf{1} - R)\hat{\omega}^2 \vec{p} - \vec{\omega} \vec{\omega}^t \hat{\omega} \vec{p} \theta \\ 0 & 1 \end{pmatrix}$$

Für den Positionsanteil  $\vec{q}$  gilt:

$$\begin{aligned} \vec{q} &= -(\mathbf{1} - R)\hat{\omega}^2 \vec{p} - (\hat{\omega}^2 + \mathbf{1})\hat{\omega}\vec{p}\theta \\ &= -(\mathbf{1} - R)(\vec{\omega}\vec{\omega}^t - \mathbf{1})\vec{p} - (\hat{\omega}^3 + \hat{\omega})\vec{p}\theta \\ &= -\vec{\omega}\vec{\omega}^t \vec{p} + \vec{p} + R\vec{\omega}\vec{\omega}^t \vec{p} - R\vec{p} \\ &= -\vec{\omega}\vec{\omega}^t \vec{p} + \vec{p} + \vec{\omega}\vec{\omega}^t \vec{p} - R\vec{p} = -R\vec{p} + \vec{p} \end{aligned}$$

Für  $R_{\vec{\omega}}(\theta) = T_{\vec{p}} \cdot T_{R_{\vec{\omega},\theta}} \cdot T_{\vec{p}}^{-1}$  gilt:

$$R_{\vec{\omega}}(\theta) = \begin{pmatrix} \mathbf{1} & \vec{p} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{1} & -\vec{p} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R & -R\vec{p} + \vec{p} \\ 0 & 1 \end{pmatrix}$$

Beide Seiten der Gleichung  $R_{\vec{\omega}}(\theta) = e^{\hat{\xi}_R \cdot \theta}$  sind also gleich.

**Beispiel 1.11** Der erzeugende twist für die Rotationsbewegung aus Beispiel 1.10 hat damit die einfache Form:

$$\xi = (\vec{v}, \vec{\omega}) = (l_1, 0, 0, 0, 0, 1)^t$$
.

# 1.3.3 Translation entlang einer beliebigen Achse

Wir hatten in Satz 1.9 bereits gesehen, dass die Translation entlang der Achse  $\vec{\omega}$  um den Betrag d die folgende Darstellung besitzt:

$$T_{\vec{\omega}}(d) = \begin{pmatrix} \mathbf{1} & d \cdot \vec{\omega} \\ 0 & 1 \end{pmatrix} = e^{\hat{\xi}_T \cdot d} \quad \text{mit} \quad \hat{\xi}_T = \begin{pmatrix} 0 & \vec{\omega} \\ 0 & 0 \end{pmatrix} \quad \text{bzw.} \quad \xi_T = (\vec{\omega}, 0)$$

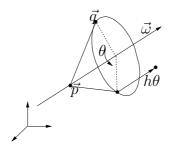


Abbildung 5: Bewegung eines Punktes  $\vec{q}$  bei Anwendung eines screw.

## 1.4 Screw-Koordinaten

Wir wollen nun die Rotation  $R_{\vec{\omega}}(\theta)$  und die Translation  $T_{\vec{\omega}}(h\theta)$  in einer Bewegung zusammenfassen. Da die Translation entlang der Rotationsachse erfolgt, kommutieren die beiden Bewegungen (gilt nicht im Allgemeinen!) und wir erhalten eine Schraubenbewegung  $S(\theta)$  entlang der Achse  $l = \{\vec{p} + \lambda \vec{\omega}\}$  (Abb. 5):

$$S(\theta) = R_{\vec{\omega}}(\theta) \cdot T_{\vec{\omega}}(h\theta) = T_{\vec{\omega}}(h\theta) \cdot R_{\vec{\omega}}(\theta) = \begin{pmatrix} R_{\vec{\omega},\theta} & -R_{\vec{\omega},\theta}\vec{p} + \vec{p} + h\theta\vec{\omega} \\ 0 & 1 \end{pmatrix} = e^{(\hat{\xi}_R + h\hat{\xi}_T)\theta}$$

Definition 1.12 Ein screw besteht aus

- einer Achse  $l = \vec{p} + \lambda \vec{w}, \lambda \in \mathbb{R}$ ,
- $\bullet$  dem pitch h
- und dem Betrag M.

Er beschreibt eine Rotation um die Achse l um den Winkel  $\theta=M$  bei gleichzeitiger Translation entlang dieser Achse um den Betrag  $d=h\theta$ . Falls  $h=\infty$  handelt es sich um eine reine Translation entlang der Achse l um den Betrag M.

Der zugehörige erzeugende twist hat die Form  $\xi_S = \xi_R + h\xi_T = (-\vec{\omega} \times \vec{p} + h\vec{\omega}, \vec{\omega})$ . Umgekehrt können aus twist-Koordinaten  $\theta\xi = (\theta\vec{v}, \theta\vec{\omega})$  sehr leicht die screw-Koordinaten berechnet werden:

1. Achse: 
$$l = \begin{cases} \{\frac{\vec{\omega} \times \vec{v}}{||\vec{\omega}||^2} + \lambda \vec{\omega} \mid \lambda \in \mathbb{R} \} & \text{falls } \vec{\omega} \neq 0 \\ \{0 + \lambda \vec{v} \mid \lambda \in \mathbb{R} \} & \text{falls } \vec{\omega} = 0 \end{cases}$$

2. pitch: 
$$h = \begin{cases} \frac{\vec{\omega}^t \vec{v}}{\|\vec{\omega}\|^2} & \text{falls } \vec{\omega} \neq 0 \\ \infty & \text{falls } \vec{\omega} = 0 \end{cases}$$

3. Betrag: 
$$M = \begin{cases} \|\theta\vec{\omega}\| & \text{falls } \vec{\omega} \neq 0 \\ \|\theta\vec{v}\| & \text{falls } \vec{\omega} = 0 \end{cases}$$

Wir unterscheiden folgende spezielle screws:

• zero-pitch screw: reine Rotation

• infinite-pitch screw: reine Translation

**Zusammenfassung:** Zu jeder homogenen Transformation  $T \in SE(3)$  existiert ein twist (bzw. ein screw), der diese Transformation erzeugt. Wie wir in Beispiel 1.10 gesehen haben, ist der twist i.d.R. abhängig von den Bewegungsparametern, im Beispiel dem Drehwinkel  $\alpha$  und den Verschiebungen  $l_1$ ,  $l_2$ . Im Falle von Schraubenbewegungen kann dieser Bewegungsparameter jedoch als Betrag  $M = \theta$  ausgeklammert werden:  $\xi = \theta \bar{\xi}$ , so dass  $\bar{\xi}$  für alle Werte von  $\theta$  dieselbe Form hat. Dieser Umstand macht die Schraubenbewegungen so interessant für die Robotik.

# 1.5 Vorwärts-Kinematik

Wir betrachten kinematische Ketten mit Drehgelenken (Winkel  $\theta_i \in Q_i \subseteq [0, 2\pi)$ ) und prismatischen Gelenken (Verschiebung  $\theta_i \in Q_i \subseteq \mathbb{R}$ ). Der Gelenkwinkelraum eines Manipulators mit k Gelenken ist dann definiert durch  $Q = Q_1 \times Q_2 \times \cdots \times Q_k$ . Wir definieren ein Basis-KS S und ein Tool-KS T, sowie lokale KS  $L_i$  für jedes Segment, so dass deren z - Achse mit der Bewegungsachse  $\vec{\omega}_i$  des i-ten Gelenks übereinstimmt. Die Vorwärtskinematik  $T_{st}: Q \to SE(3)$  können wir bestimmen

1. durch Angabe von Relativ-Transformationen (z.B. DH-Konvention)

$$T_{st}(\theta_1,\ldots,\theta_k) = T_{s1}(\theta_1) \cdot T_{12}(\theta_2) \cdot \ldots \cdot T_{k-1\,k}(\theta_k) \cdot T_{kt}$$

2. durch Angabe der twists, die die Bewegung der Gelenke erzeugen.

Die zweite Methode ist meist einfacher, da der erzeugende twist i.d.R. direkt abgelesen werden kann. Die DH-Konvention erfordert hingegen einige Anstrengungen, um die Parameter für die Relativ-Transformationen zu bestimmen.

Um die twist-Darstellung zu bestimmen, legen wir neben den KS S und T noch eine Referenzkonfiguration des Manipulators fest, in der alle Gelenkparameter  $\theta_i = 0$  sind (Nullstellung). Die zugehörige Koordinaten-Transformation ist  $T_{st}(0): T \to S$ . Beginnend beim letzten Gelenk  $\theta_k$  bestimmt man nun die twists der einzelnen Gelenkbewegungen, wobei immer nur ein Gelenk bewegt wird.

Bewege 
$$\theta_k$$
: 
$$T_{st}(\theta_k) = e^{\hat{\xi}_k \theta_k} T_{st}(0)$$
Bewege  $\theta_{k-1}$ : 
$$T_{st}(\theta_{k-1}, \theta_k) = e^{\hat{\xi}_{k-1} \theta_{k-1}} T_{st}(\theta_k) = e^{\hat{\xi}_{k-1} \theta_{k-1}} e^{\hat{\xi}_k \theta_k} T_{st}(0)$$
...
Bewege  $\theta_1$ : 
$$T_{st}(\theta_1, \dots, \theta_k) = e^{\hat{\xi}_1 \theta_1} T_{st}(\theta_2, \dots, \theta_k) = e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_2 \theta_2} \dots e^{\hat{\xi}_k \theta_k} T_{st}(0)$$

Da die Bewegung immer relativ zum Basis-KS S ausgeführt wird, müssen die neuen Transformation von links multipliziert werden (im Gegensatz zur DH-Konvention,

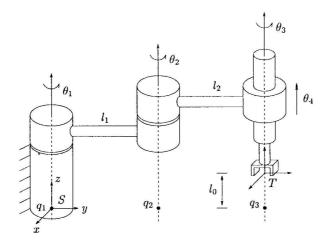


Abbildung 6: SCARA-Manipulator in einer einfachen Referenzkonfiguration

die die Gelenkbewegungen in umgekehrter Reihenfolge ausführt). Man kann zeigen, dass die Reihenfolge der Gelenkbewegungen keine Rolle spielt (Übung). Die Vorwärtskinematik ist also gegeben durch:

$$T_{st}(\theta_1, \dots, \theta_k) = e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_2 \theta_2} \dots e^{\hat{\xi}_k \theta_k} T_{st}(0)$$

Die twists  $\hat{\xi}_i$  hängen lediglich von der Wahl des Basis-KS S und der Referenzkonfiguration ab (das Tool-KS T ist dadurch eindeutig bestimmt). Für Drehbzw. Schiebegelenke ergibt sich:

$$\xi_i^R = (-\vec{\omega}_i \times \vec{p}_i, \vec{\omega}_i)$$
 bzw.  $\xi_i^T = (\vec{\omega}_i, 0)$ 

Beispiel 1.13 (SCARA) Als Beispiel betrachten wir einen SCARA-Manipulator (Selective Compliance Assembly Robot Arm) wie in Abb. 6. Die Achsen lassen sich leicht ablesen:

$$\vec{\omega}_i = (0, 0, 1)^t$$
  $\vec{q}_1 = (0, 0, 0)^t$   $\vec{q}_2 = (0, l_1, 0)^t$   $\vec{q}_3 = (0, l_1 + l_2, 0)^t$ 

Daraus bestimmen wir die twist-Koordinaten

$$\xi_1 = (-\vec{\omega}_1 \times \vec{q}_1, \vec{\omega}_1) = (0, 0, 0, 0, 0, 1)^t \quad \xi_2 = (-\vec{\omega}_2 \times \vec{q}_2, \vec{\omega}_2) = (l_1, 0, 0, 0, 0, 1)^t$$
  
$$\xi_3 = (-\vec{\omega}_3 \times \vec{q}_2, \vec{\omega}_3) = (l_1 + l_2, 0, 0, 0, 0, 1)^t \quad \xi_4 = (\vec{\omega}_4, 0) = (0, 0, 1, 0, 0, 0)^t$$

und schließlich die einzelnen Bewegungen:

$$T_{st}(0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_1 + l_2 \\ 0 & 0 & 1 & l_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad e^{\hat{\xi}_1 \theta_1} = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad e^{\hat{\xi}_2 \theta_2} = \begin{pmatrix} c_2 & -s_2 & 0 & l_1 s_2 \\ s_2 & c_2 & 0 & l_1 (1 - c_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$e^{\hat{\xi}_3 \theta_3} = \begin{pmatrix} c_3 & -s_3 & 0 & (l_1 + l_2) s_3 \\ s_3 & c_3 & 0 & (l_1 + l_2) (1 - c_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad e^{\hat{\xi}_4 \theta_4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \theta_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ausmultiplizieren ergibt:

$$T_{st}(\theta) = \begin{pmatrix} R(\theta) & \vec{p}(\theta) \\ 0 & 1 \end{pmatrix}$$

$$R(\theta) = \begin{pmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\vec{p}(\theta) = \begin{pmatrix} -l_1 \sin \theta_1 - l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ l_0 + \theta_4 \end{pmatrix}$$

# 1.6 Geschwindigkeiten

Wir wollen nun auch Geschwindigkeiten von festen Körpern in Form von twists ausdrücken. Bei der Herleitung der Exponentialdarstellung von Rotationen sind wir von der Gleichung

$$\dot{q}_a = \vec{\omega}^s \times \vec{q}_a = \hat{\omega}^s \vec{q}_a$$

ausgegangen, die die Geschwindigkeit eines Punktes  $\vec{q}$  bezüglich des ortsfesten KS A bei Anwendung einer Rotation  $\vec{\omega}^s$  angibt. Wir wollen nun  $\hat{\omega}^s$  durch  $R_{ab}$  ausdrücken:

$$\begin{aligned} \vec{q}_a(t) &= R_{ab}(t) \vec{q}_b \\ \text{Ableiten:} \qquad \dot{q}_a(t) &= \dot{R}_{ab}(t) \vec{q}_b \\ &= \dot{R}_{ab}(t) \cdot R_{ab}^{-1}(t) \cdot R_{ab}(t) \cdot \vec{q}_b \\ &= \dot{R}_{ab}(t) \cdot R_{ab}^{-1}(t) \cdot \vec{q}_a(t) \end{aligned}$$

Wir können also  $\hat{\omega}^s$  mit  $\dot{R}_{ab}(t) \cdot R_{ab}^{-1}(t)$  identifizieren, wobei beide Matrizen antisymmetrisch sind (Übung). Die Matrix  $\dot{R}$  selbst ist wenig aussagekräftig, da sie Ortskoordinaten – ausgedrückt im körperfesten KS  $(\vec{q}_b)$  – in Geschwindigkeiten – ausgedrückt im Welt-KS  $(\dot{q}_a)$  – überführt und daher auf zwei verschiedene KS Bezug nimmt. Ausserdem ist  $\dot{R} \notin SO(3)$  und  $\dot{R} \notin so(3)$ .

**Definition 1.14** Wir definieren:  $\hat{\omega}_{ab}^s := \dot{R}_{ab} \cdot R_{ab}^{-1}$  und  $\hat{\omega}_{ab}^b := R_{ab}^{-1} \cdot \dot{R}_{ab}$ . Während  $\vec{\omega}_{ab}^s$  die Rotationsgeschwindigkeit im *ortsfesten Welt-KS A* angibt, beschreibt  $\vec{\omega}_{ab}^b$  die Rotationsgeschwindigkeit aus Sicht des *sich bewegenden Körper-KS B*.

Lemma 1.15 Es gilt: (Übung)

$$\hat{\omega}_{ab}^s = R_{ab} \, \hat{\omega}_{ab}^b \, R_{ab}^{-1}$$
 bzw.  $\vec{\omega}_{ab}^s = R_{ab} \, \vec{\omega}_{ab}^b$ 

Die Geschwindigkeit eines Punktes  $\vec{q}$  kann also entweder bzgl. des Welt-KS A oder bzgl. des körperfesten KS B angegeben werden:

$$\vec{v}_a(t) = \hat{\omega}_{ab}^s R_{ab}(t) \, \vec{q}_b = \hat{\omega}_{ab}^s \, \vec{q}_a(t) = \vec{\omega}_{ab}^s \times \vec{q}_a(t)$$
$$\vec{v}_b(t) = R_{ab}^{-1}(t) \vec{v}_a(t) = R_{ab}^{-1}(t) \hat{\omega}_{ab}^s R_{ab}(t) \, \vec{q}_b = \hat{\omega}_{ab}^b \vec{q}_b = \vec{\omega}_{ab}^b \times \vec{q}_b$$

Allgemeine Transformation Für eine allgemeine homogene Transformation

$$T_{ab}(t) = \begin{pmatrix} R_{ab}(t) & p_{ab}(t) \\ 0 & 1 \end{pmatrix} \in SE(3)$$

betrachten wir analog:

$$\begin{split} \dot{T}_{ab}T_{ab}^{-1} &= \begin{pmatrix} \dot{R}_{ab} & \dot{\vec{p}}_{ab} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} R_{ab}^{-1} & -R_{ab}^{-1} \, \vec{p}_{ab} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \dot{R}_{ab} \, R_{ab}^{-1} & -\dot{R}_{ab} \, R_{ab}^{-1} \, \vec{p}_{ab} + \dot{p}_{ab} \\ 0 & 0 \end{pmatrix} \\ &=: \begin{pmatrix} \hat{\omega}_{ab}^s & \vec{v}_{ab}^s \\ 0 & 0 \end{pmatrix} = \hat{V}_{ab}^s \in se(3) \end{split}$$

und

$$T_{ab}^{-1}\dot{T}_{ab} = \begin{pmatrix} R_{ab}^{-1} & -R_{ab}^{-1} \vec{p}_{ab} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{R}_{ab} & \dot{\vec{p}}_{ab} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} R_{ab}^{-1} \dot{R}_{ab} & R_{ab}^{-1} \dot{p}_{ab} \\ 0 & 0 \end{pmatrix}$$
$$=: \begin{pmatrix} \hat{\omega}_{ab}^{b} & \vec{v}_{ab}^{b} \\ 0 & 0 \end{pmatrix} = \hat{V}_{ab}^{b} \in se(3)$$

Beide Größen können also als twist aufgefasst werden!

**Definition 1.16** Der twist  $\hat{V}_{ab}^s = \dot{T}_{ab} \, T_{ab}^{-1}$  heißt Welt-Geschwindigkeit und gibt die Geschwindigkeit des sich bewegenden KS B aus Sicht des Welt-KS A an. Der twist  $\hat{V}_{ab}^b = T_{ab}^{-1} \, \dot{T}_{ab}$  heißt Körper-Geschwindigkeit und gibt die Geschwindigkeit des KS B relativ zum KS A aus Sicht von B an. In twist-Koordinaten erhalten wir die Darstellung als 6-dim. Vektor:

$$\begin{split} V_{ab}^s &= \begin{bmatrix} \vec{v}_{ab}^s \\ \vec{\omega}_{ab}^s \end{bmatrix} = \begin{bmatrix} -\dot{R}_{ab} \, R_{ab}^{-1} \vec{p}_{ab} + \dot{p}_{ab} \\ (\dot{R}_{ab} \, R_{ab}^{-1})^\vee \end{bmatrix} \\ V_{ab}^b &= \begin{bmatrix} \vec{v}_{ab}^b \\ \vec{\omega}_{ab}^b \end{bmatrix} = \begin{bmatrix} R_{ab}^{-1} \dot{p}_{ab} \\ (R_{ab}^{-1} \, \dot{R}_{ab})^\vee \end{bmatrix} \end{split}$$

Für die Geschwindigkeit eines Punktes  $\vec{q}$  des Körpers gilt also in homogenen bzw. in 3D-Koordinaten:

$$\bar{v}_a(t) = \hat{V}_{ab}^s \, \bar{q}_a(t) \qquad \text{bzw.} \qquad \vec{v}_a(t) = \vec{\omega}_{ab}^s \times \vec{q}_a(t) + \vec{v}_{ab}^s \\
\bar{v}_b(t) = \hat{V}_{ab}^b \, \bar{q}_b \qquad \text{bzw.} \qquad \vec{v}_b(t) = \vec{\omega}_{ab}^b \times \vec{q}_b + \vec{v}_{ab}^b$$

**Bemerkung:** Der "lineare" Anteil  $\vec{v}_{ab}^s$  der Welt-Geschwindigkeit enthält wieder Rotationsanteile und kann daher nicht einfach als Geschwindigkeit des Ursprungs von KS B relativ zu KS A interpretiert werden! Nur die Komponenten  $\vec{\omega}_{ab}^b$  und  $\vec{v}_{ab}^b$  der Körper-Geschwindigkeit können sinnvoll geometrisch interpretiert werden:  $\vec{\omega}_{ab}^b$  ist die Rotationsgeschwindigkeit von KS B aus Sicht dieses KS und  $\vec{v}_{ab}^b$  ist die

Geschwindigkeit des Ursprungs von KS B relativ zum welt-festen KS A, ebenfalls aus Sicht von B.

Zwischen den beiden Darstellungen der Geschwindigkeits-twists gilt folgende Relation:

$$\hat{V}^s_{ab} = \dot{T}_{ab} \, T^{-1}_{ab} = T_{ab} \, T^{-1}_{ab} \, \dot{T}_{ab} \, T^{-1}_{ab} = T_{ab} \, \hat{V}^b_{ab} \, T^{-1}_{ab}$$

Um die Transformation direkt auf Basis der twist-Koordinaten zu machen, benötigen wir die folgende

**Definition 1.17** Zu einer gegebenen Transformation  $T = \begin{pmatrix} R & \vec{p} \\ 0 & 1 \end{pmatrix}$  definieren wir die Adjungierte  $Ad_T$ :

$$Ad_T = \begin{pmatrix} R & \hat{p} R \\ 0 & R \end{pmatrix} \in \mathbb{R}^{6 \times 6}.$$

Die Inverse hat folgende Form:

$$Ad_T^{-1} = \begin{pmatrix} R^t & -R^t \,\hat{p} \\ 0 & R^t \end{pmatrix} = Ad_{T^{-1}}.$$

Die Adjungierte transformiert nun twist-Koordinaten bzgl. verschiedener KS (hier von B nach A):

$$Ad_{T_{ab}}V_{ab}^{b} = \begin{pmatrix} R_{ab} & \hat{p}_{ab} R_{ab} \\ 0 & R_{ab} \end{pmatrix} \begin{pmatrix} \vec{v}_{ab}^{b} \\ \vec{\omega}_{ab}^{b} \end{pmatrix} = \begin{pmatrix} R_{ab} \vec{v}_{ab}^{b} + \hat{p}_{ab} R_{ab} \vec{\omega}_{ab}^{b} \\ R_{ab} \vec{\omega}_{ab}^{b} \end{pmatrix} \stackrel{Def}{=} \begin{pmatrix} \dot{\vec{p}}_{ab} + \vec{p}_{ab} \times \vec{\omega}_{ab}^{s} \\ \vec{\omega}_{ab}^{s} \end{pmatrix}$$

$$= V_{ab}^{s}.$$

Dies gilt auch für beliebige twists:

**Lemma 1.18** Sei  $\hat{\xi} \in se(3)$  ein twist und  $T \in SE(3)$  eine homogene Transformation. Dann ist  $\hat{\xi}' = T \hat{\xi} T^{-1}$  ebenfalls ein twist mit den Koordinaten  $\xi' = Ad_T \xi$ . (Übung)

Beispiel 1.19 Wir wollen die Welt-Geschwindigkeit  $V^s_{ab}$  der Transformation

$$T_{ab}(t) = \begin{pmatrix} \cos \theta(t) & -\sin \theta(t) & 0 & -l_2 \sin \theta(t) \\ \sin \theta(t) & \cos \theta(t) & 0 & l_1 + l_2 \cos \theta(t) \\ 0 & 0 & 1 & l_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

bestimmen. Wir lassen die Zeitabhängigkeit und die Indizes ab zunächst weg. Nach Definition gilt:  $\vec{v}_{ab}^s = -\dot{R}R^t\vec{p} + \dot{\vec{p}}$  und  $\hat{\omega}_{ab}^s = \dot{R}R^t$ .

$$\dot{R}R^{t} = \begin{pmatrix} -\dot{\theta}\sin\theta & -\dot{\theta}\cos\theta & 0\\ \dot{\theta}\cos\theta & -\dot{\theta}\sin\theta & 0\\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -\dot{\theta} & 0\\ \dot{\theta} & 0 & 0\\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -\vec{\omega}_{3}^{s} & 0\\ \vec{\omega}_{3}^{s} & 0 & 0\\ 0 & 0 & 0 \end{pmatrix}$$

Es folgt also:

$$\vec{\omega}_{ab}^s = \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}(t) \end{pmatrix} \quad \text{und} \quad \vec{v}_{ab}^s = \begin{pmatrix} 0 & \dot{\theta} & 0 \\ -\dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -l_2 \sin \theta \\ l_1 + l_2 \cos \theta \\ 0 \end{pmatrix} + \begin{pmatrix} -l_2 \dot{\theta} \cos \theta \\ -l_2 \dot{\theta} \sin \theta \\ 0 \end{pmatrix} = \begin{pmatrix} l_1 \dot{\theta}(t) \\ 0 \\ 0 \end{pmatrix}$$

Wir erhalten also den twist, der im vorigen Beispiel die eigentliche Bewegung  $e^{\hat{\xi}\theta(t)}$  erzeugte. Machen wir wieder die Aufteilung  $T(t) = e^{\hat{\xi}\theta(t)}T(0)$  folgt für die Ableitung:

$$\dot{T}(t) = \dot{\theta}\xi \cdot e^{\hat{\xi}\theta(t)} \cdot T(0) + e^{\hat{\xi}\theta(t)} \cdot 0$$
$$\hat{V}_{ab}^{s} = \dot{T}(t) \cdot T^{-1}(t) = \dot{\theta}\xi \cdot e^{\hat{\xi}\theta(t)} \cdot T(0) \cdot T^{-1}(0) \cdot e^{-\xi\theta(t)} = \dot{\theta}\xi$$

Die Geschwindigkeit  $V^s_{ab}$  stimmt demnach mit der Ableitung des erzeugenden twists  $\frac{d}{dt}\hat{\xi}\theta(t) = \hat{\xi}\dot{\theta}(t)$  überein.

**Lemma 1.20 (Übung)** Für drei sich gegeneinander bewegende KS (A, B und C) gelten folgende Relationen für die Welt- und Körpergeschwindigkeiten:

$$V_{ac}^{s} = V_{ab}^{s} + Ad_{T_{ab}}V_{bc}^{s}$$

$$V_{ac}^{b} = Ad_{T_{bc}^{-1}}V_{ab}^{b} + V_{bc}^{b}$$

$$V_{ab}^{b} = -V_{ba}^{s} = -Ad_{T_{ba}}V_{bc}^{b}$$

Warum drücken wir Geschwindigkeiten mit so vielen Indizes aus? Die Lage eines Körpers im Raum ist zunächst unabhängig von der Wahl eines KS – erst durch die Angabe von Koordinaten legt man sich auf ein KS fest, z.B.  $\vec{q}_a$  bzgl. A oder  $\vec{q}_b$  bzgl. B. Geschwindigkeiten hingegen beschreiben immer Relativ-Geschwindigkeiten zwischen zwei Systemen, weshalb man die beiden Bezugskoordinatensysteme als untere Indizes  $V_{ab}$  angibt. Der obere Index gibt nun an, aus Sicht welchen Koordinatensystems man die Geschwindigkeit betrachtet:  $V_{ab}^s$  gibt die Geschwindigkeit aus Sicht des KS A an und  $V_{ab}^b$  aus Sicht des KS B.

Wir haben gesehen, dass wir sowohl Geschwindigkeiten als auch homogene Transformationen durch twists ausdrücken können. Dabei werden die homogenen Transformationen durch Anwendung der Exponential-Funktion aus einem twist erzeugt, sind aber selbst kein twist. Geschwindigkeiten können hingegen direkt als twist interpretiert werden.

# 1.7 Jacobi-Matrix

Die Jacobi-Matrix bildet Gelenkgeschwindigkeiten  $\dot{\theta}_i(t)$  auf Tool-Geschwindigkeiten  $V_{st}^s$  ab. Mit der twist-Darstellung ergibt sich:

$$\hat{V}_{st}^{s} = \dot{T}_{st}(\theta) T_{st}^{-1}(\theta) = \sum_{i=1}^{k} \left( \frac{\partial T_{st}}{\partial \theta_{i}} \dot{\theta}_{i} \right) T_{st}^{-1}(\theta) = \sum_{i=1}^{k} \left( \frac{\partial T_{st}}{\partial \theta_{i}} T_{st}^{-1}(\theta) \right) \dot{\theta}_{i}$$

Für die einzelnen Differentiale erhalten wir:

$$\frac{\partial T_{st}}{\partial \theta_{i}} T_{st}^{-1}(\theta) = e^{\hat{\xi}_{1}\theta_{1}} \dots e^{\hat{\xi}_{i-1}\theta_{i-1}} \cdot \frac{\partial}{\partial \theta_{i}} \left( e^{\hat{\xi}_{i}\theta_{i}} \right) \cdot e^{\hat{\xi}_{i+1}\theta_{i+1}} \dots e^{\hat{\xi}_{k}\theta_{k}} \cdot T_{st}(0) \cdot T_{st}(\theta)^{-1} 
= e^{\hat{\xi}_{1}\theta_{1}} \dots e^{\hat{\xi}_{i-1}\theta_{i-1}} \cdot \left( \hat{\xi}_{i} \right) e^{\hat{\xi}_{i}\theta_{i}} \cdot e^{\hat{\xi}_{i+1}\theta_{i+1}} \dots e^{\hat{\xi}_{k}\theta_{k}} \cdot T_{st}(0) \cdot T_{st}(\theta)^{-1} 
= e^{\hat{\xi}_{1}\theta_{1}} \dots e^{\hat{\xi}_{i-1}\theta_{i-1}} \cdot \left( \hat{\xi}_{i} \right) e^{-\hat{\xi}_{i-1}\theta_{i-1}} \dots e^{-\hat{\xi}_{1}\theta_{1}} = \hat{\xi}'_{i}$$

Sie sind also wieder als twist interpretierbar:

$$\left(\frac{\partial T_{st}}{\partial \theta_i} T_{st}^{-1}(\theta)\right)^{\vee} = \xi_i' = Ad_{e^{\hat{\xi}_1 \theta_1} \dots e^{\hat{\xi}_{i-1} \theta_{i-1}}} \xi_i ,$$

und zwar als der in die aktuelle Konfiguration transformierte twist  $\xi_i$  der Referenz-konfiguration. Offenbar hängt diese Transformation nur von den vorangehenden Gelenken  $\theta_1, \ldots, \theta_{i-1}$  ab. Die Jacobi-Matrix bekommt also die einfache Form

$$J_{st}^s(\theta) = [\xi_1', \xi_2', \dots, \xi_k'] \quad \text{mit} \quad \xi_i' = Ad_{e^{\hat{\xi}_1 \theta_1} \dots e^{\hat{\xi}_{i-1} \theta_{i-1}}} \xi_i$$

und es gilt:

$$V_{st}^s = J_{st}^s(\theta) \cdot \dot{\theta}$$

Analog kann man auch die Jacobi-Matrix für Körper-Geschwindigkeiten herleiten (Übung). Außerdem gilt:

$$V_{st}^b = J_{st}^b(\theta) \cdot \dot{\theta} \quad \text{mit} \quad J_{st}^b(\theta) = Ad_{T_{st}(\theta)}^{-1} \cdot J_{st}^s(\theta)$$

Beispiel 1.21 (SCARA-Manipulator) Die Achsen des Manipulator bleiben fest, lediglich die Unterstützungsvektoren verschieben sich:

$$\vec{q}_1' = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad \vec{q}_2' = \begin{pmatrix} -l_1 \sin \theta_1 \\ l_1 \cos \theta_1 \\ 0 \end{pmatrix} \qquad \vec{q}_3' = \begin{pmatrix} -l_1 \sin \theta_1 - l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ 0 \end{pmatrix}$$

Wir rechnen die twist-Koordinaten mit der bekannten Formel  $\xi_i' = (-\vec{\omega}_i \times \vec{q}_i', \vec{\omega}_i)$  für Drehgelenke aus:

$$J_{st}^{s} = \begin{pmatrix} 0 & l_{1}\cos\theta_{1} & l_{1}\cos\theta_{1} + l_{2}\cos(\theta_{1} + \theta_{2}) & 0\\ 0 & l_{1}\sin\theta_{1} & l_{1}\sin\theta_{1} + l_{2}\sin(\theta_{1} + \theta_{2}) & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 1 & 1 & 1 & 0 \end{pmatrix}$$

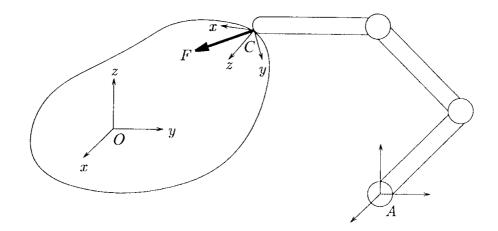


Abbildung 7: Transformation von wrenches zwischen verschiedenen KS (Kontakt-KS C nach Objekt-KS O)

# 1.8 Wrenches: Kräfte und Drehmomente

Eine verallgemeinerte Kraft umfasst eine lineare Kraft  $\vec{f} \in \mathbb{R}^3$  und ein Drehmoment  $\vec{\tau} \in \mathbb{R}^3$ . Ein solches Paar nennen wir *wrench* und schreiben:

$$F = (\vec{f}, \vec{\tau}) \in \mathbb{R}^6$$
.

Ein wrench  $F_o$  wird – genau wie Geschwindigkeiten und Positionsvektoren – immer bzgl. eines KS (hier Objekt-KS O) angegeben. Die Kraft und das Drehmoment sollen jeweils im Ursprung dieses KS angreifen. Um zu sehen, wie wrench-Koordinaten von einem KS in ein anderes transformiert werden, betrachten wir die instantane Arbeit (Leistung) die erbracht wird, wenn ein wrench  $F_c$  eine Körper-Geschwindigkeit  $V_c$  hervorruft:

$$\delta W = V_{ac}^b \cdot F_c = (V_{ac}^b)^t \cdot F_c = \left(Ad_{T_{oc}}^{-1} V_{ao}^b\right)^t \cdot F_c = (V_{ao}^b)^t \cdot \left(Ad_{T_{cc}}^t \cdot F_c\right) = V_{ao}^b \cdot F_o$$

Wir erhalten also:

$$F_{o} = Ad_{T_{oc}^{-1}}^{t} F_{c} = \begin{pmatrix} R_{oc}^{t} & -R_{oc}^{t} \hat{p}_{oc} \\ 0 & R_{oc}^{t} \end{pmatrix}^{t} F_{c} = \begin{pmatrix} R_{oc} & 0 \\ \hat{p}_{oc} R_{oc} & R_{oc} \end{pmatrix} \begin{pmatrix} \vec{f}_{c} \\ \vec{\tau}_{c} \end{pmatrix}$$
$$= \begin{pmatrix} R_{oc} \vec{f}_{c} \\ R_{oc} \vec{\tau}_{c} + \vec{p}_{oc} \times (R_{oc} \vec{f}_{c}) \end{pmatrix}$$

Im Vergleich dazu die Transformation von twists:

$$V_{oc}^{s} = Ad_{T_{oc}}V_{oc}^{b} = \begin{pmatrix} R_{oc} & \hat{p}_{oc}R_{oc} \\ 0 & R_{oc} \end{pmatrix} \begin{pmatrix} \vec{v}_{oc}^{b} \\ \vec{\omega}_{oc}^{b} \end{pmatrix} = \begin{pmatrix} R_{oc}\vec{v}_{oc}^{b} + \vec{p}_{oc} \times (R_{oc}\vec{\omega}_{oc}^{b}) \\ R_{oc}\vec{\omega}_{oc}^{b} \end{pmatrix}.$$

In beiden Fällen werden die Richtungsvektoren  $(\vec{v}, \vec{\omega}, \vec{f}, \vec{\tau})$  mittels der Rotationsmatrix  $R_{oc}$  im neuen KS ausgedrückt. Hinzu kommen eine zusätzliche Drehmoment-Komponente  $\vec{p}_{oc} \times \vec{f}_o$ , die durch die Kraft  $\vec{f}$  hervorgerufen wird, bzw. eine zusätzliche Geschwindigkeitskomponente  $\vec{p}_{oc} \times \vec{\omega}_{oc}^s$ , die durch die Rotationsgeschwindigkeit  $\vec{\omega}_{oc}$  hervorgerufen wird.

#### 1.8.1 screw-Koordinaten von wrenches

Wie schon twists, können wir auch wrenches als screw auffassen und umgekehrt. Der dem screw (l, h, M) (angegeben bzgl. einem KS A) zugeordnete wrench  $F_a$  entsteht durch Anwendung einer Kraft vom Betrag M entlang der Achse l und einem Drehmoment vom Betrag hM entlang dieser Achse:

$$F_a = M \begin{pmatrix} \vec{\omega} \\ -\vec{\omega} \times \vec{p} + h\vec{\omega} \end{pmatrix}$$
 falls  $h < \infty$  bzw.  $F_a = M \begin{pmatrix} 0 \\ \vec{\omega} \end{pmatrix}$  falls  $h = \infty$ 

(Beim zugehörigen twist wurde eine Rotation um den Betrag M und eine Translation um den Betrag hM durchgeführt – die Rollen von Rotations- und Translationsanteil sind also wieder genau vertauscht.) Umgekehrt können wir zu einem wrench  $F_a = (\vec{f}, \vec{\tau})$  den zugehörigen screw, durch Lösung obiger Gleichung finden. Es ergibt sich:

$$\begin{split} \vec{f} &= 0: \qquad M = \|\vec{\tau}\|, \ \vec{\omega} = \frac{\vec{\tau}}{M}, \ h = \infty, \ \vec{p} = 0 \\ \vec{f} &\neq 0: \qquad M = \|\vec{f}\|, \ \vec{\omega} = \frac{\vec{f}}{M}, \ h = \frac{\vec{f}^t \vec{\tau}}{\|\vec{f}\|^2}, \ \vec{p} = \frac{\vec{f} \times \vec{\tau}}{\|\vec{f}\|^2} \end{split}$$

#### 1.8.2 Dualität von twists und wrenches

Auch diese Gleichungen lassen die Dualität von twists und wrenches erkennen. Twists und wrenches stellen Vektoren in 6-dim. Raum dar, die dual zueinander sind.

**Definition 1.22** Ein twist V und ein wrench F heißen reziprok zueinander, wenn ihr Skalarprodukt  $\delta W = F \cdot V$  Null ist, die beiden Vektoren also orthogonal zueinander stehen. In diesem Falle werden auch die zugeordneten screws als reziprok bezeichnet. Eine Menge von twists  $V_1, \ldots, V_n$  bzw. eine Menge von wrenches  $F_1, \ldots, F_n$  spannen einen linearen Unterraum der Dimension  $m \leq n$  des  $\mathbb{R}^6$  auf. Der dazu orthogonale Unterraum mit Dimension o enthält dazu orthogonale wrenches bzw. twists. Es gilt:

$$m + o = 6$$

- Die zu  $\{V_1, \ldots, V_n\}$  orthogonalen wrenches stellen Kräfte dar, die keine Bewegung im Unterraum der twists erzeugen können.
- Die zu  $\{F_1, \ldots, F_n\}$  orthogonalen twists stellen Bewegungen dar, denen die wrenches nicht widerstehen können.

# 1.9 Transponierte Jacobi-Matrix

Die Transponierte der Jacobi-Matrix bildet Kräfte (wrenches) am Endeffektor eines Manipulators auf Gelenk-Drehmomente ab. Wir leiten diesen Zusammenhang wieder mit Hilfe der geleisteten Arbeit W her:

$$\int_{t_1}^{t_2} \dot{\theta}^t \tau dt = W = \int_{t_1}^{t_2} (V_{st}^s)^t \cdot F_s dt \quad \text{für alle } t_1 \le t_2 \in \mathbb{R}$$
 
$$\Leftrightarrow \qquad \dot{\theta}^t \tau = (J_{st}^s \cdot \dot{\theta})^t \cdot F_s \quad \text{für alle } \dot{\theta}$$
 
$$\Leftrightarrow \qquad \tau = (J_{st}^s)^t \cdot F_s$$

Offenbar kann man durch Invertierung dieser Gleichung auch die Gelenk-Drehmomente berechnen, die notwendig sind, um einen bestimmten wrench auszuüben. Natürlich besteht wieder das Problem, dass die Inverse eventuell nicht definiert ist, weil der Manipulator in einer singulären Position steht oder die Matrix nicht quadratisch ist, weil der Arm redundant ist oder zuwenig Freiheitsgrade hat.

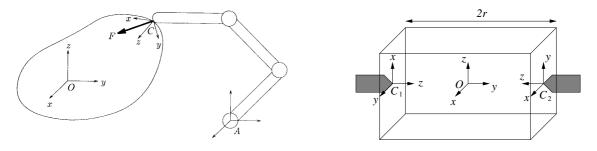


Abbildung 8: Kontaktpunkte an einem zu greifenden Objekt

# 2 Mehrfingriges Greifen

# Motivation

- 1. Vielzahl von Objekten mit einem Endeffektor greifen.
- 2. Feine Bewegungen sind einfacher mit kleinen Fingern als mit mehreren großen Armen realisierbar (Dynamik und Trägheit).
- 3. Manipulierbarkeit erfordert "große" Anzahl von individuell steuerbaren "Fingern", um auch umgreifen zu können.

# 2.1 Kinematik eines Griffs

Wir untersuchen zunächst die Frage, wie ein Griff definiert ist und welche Zusammenhänge zwischen Kräften und Bewegungen der Finger und des Objektes bestehen.

## 2.1.1 Kontaktmodelle

Ein Griff zeichnet sich in erster Linie durch eine bestimmte Anzahl von Kontakten an einem (zu greifenden) Objekt aus. Wir definieren für jeden Kontakt i ein KS  $C_i$ , dessen z-Achse ins Objektinnere entlang der Kontakt-Normalenrichtung zeigt. Im Ursprung dieser KS greift jeweils ein wrench  $F_{c_i}$  an. Die Lage dieser KS ist durch die Transformation  $T_{oc_i}$  relativ zu einem körperfesten KS O angegeben (sinnvollerweise im Schwerpunkt des KS). Die Objektlage selbst ist relativ zu einem Basis-KS P (Palm) festgelegt  $(T_{po})$ . Für jeden Kontakt müssen wir nun noch ein Reibungsmodell festlegen. Damit definieren wir die Menge der übertragbaren Kräfte  $F_{c_i}$  durch Angabe einer wrench-Basis und eines Reibungskegels  $FC_i$ .

• Punktkontakt ohne Reibung Kräfte können nur entlang der Normalenrichtung (z-Achse von  $C_i$ ) übertragen werden. Bei Anwendung tangentialer

Kräfte "rutscht" der Kontaktpunkt einfach weg.

$$F_c = [0, 0, f, 0, 0, 0]^t = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \cdot f = B_c \cdot \vec{f_c} \qquad FC_c = \{ f \ge 0 \}$$

• Punktkontakt mit (Haft-)Reibung Hat man zusätzlich Haftreibung, können auch tangentiale Kräfte ausgeübt werden. Deren maximaler Betrag hängt nach dem Coulomb-Gesetz linear mit der Kraft in Normalenrichtung zusammen (Abb. 8):

$$\|\vec{f_t}\| \le \mu f_n \,,$$

wobei  $\mu$  der (Haft-)reibungskoeffizient ist und von den Materialien der Objekte in Kontakt abhängt. Die Menge der übertragbaren Kräfte liegt damit innerhalb eines Kegels, dessen Achse mit der Normalenrichtung übereinstimmt und dessen Öffnungswinkel  $\alpha = \arctan \mu$  ist. Damit ergibt sich:

• Soft-Finger-Kontakt Realistischer ist ein Kontakt, der auch ein Drehmoment entlang der Normalenrichtung ausüben kann, und daher ein einfaches Modell für einen Flächenkontakt darstellt. Der maximale Betrag des Drehmoments ist wieder linear abhängig von der Normalenkraft ( $|\tau| = |f_4| \le \gamma f_3$ ), wobei  $\gamma$  der torsionale Reibungskoeffizient ist.

$$F_c = [f_1, f_2, f_3, 0, 0, f_4]^t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = B_c \cdot \vec{f_c}$$

$$FC_c = \{ \vec{f_c} \in \mathbb{R}^4 \mid 0 \le \sqrt{f_1^2 + f_2^2} \le \mu f_3, \mid f_4 \mid \le \gamma f_3 \}$$

Ein Kontaktmodell wird also durch Angabe einer Basis  $B_c \in \mathbb{R}^{p \times m}$  von m linear unabhängigen wrench-Komponenten  $(f_1, \ldots, f_m)$  sowie eines Reibungskegels  $FC_c$  dargestellt, wobei  $FC_c$  die Menge der ausübbaren Kräfte durch Angabe einer Reibungsbedingung weiter einschränkt. Da diese Menge konvex ist, gilt folgende Bedingung für die Kräfte:

$$\forall \vec{f_1}, \vec{f_2} \in FC_c \quad \forall \alpha, \beta > 0 \quad : \quad \alpha \vec{f_1} + \beta \vec{f_2} \in FC_c$$

#### 2.1.2 Die Greif-Matrix

Wir fassen die (statischen) Komponenten eines Griffs nun geeignet zusammen. Seien also eine Objektkonfiguration  $T_{po}$  und eine Menge von k Kontakten mit Koordinatensystemen  $C_i$ , Basen  $B_{c_i}$  und Reibungskegeln  $FC_{c_i}$  gegeben. Die wrenches an den einzelnen Kontaktpunkten sind also gegeben durch:

$$F_{c_i} = B_{c_i} \vec{f}_{c_i} .$$

Um den Netto-wrench zu bestimmen, der im Objekt-Schwerpunkt O angreift, müssen wir die wrenches in das Objekt-KS transformieren und summieren:

$$F_{o} = \sum_{i=1}^{k} Ad_{T_{oc_{i}}^{-1}}^{t} F_{c_{i}} = \sum_{i=1}^{k} \begin{pmatrix} R_{oc_{i}} & 0\\ \hat{p}_{oc_{i}} R_{oc_{i}} & R_{oc_{i}} \end{pmatrix} B_{c_{i}} \vec{f}_{c_{i}} \qquad \vec{f}_{c_{i}} \in FC_{c_{i}}$$

Wir definieren die Abbildung von Kontaktkräften  $\vec{f_c}$  auf den Objekt-wrench:

$$G_i := Ad_{T_{oc_i}^{-1}}^t B_{c_i}$$

und erhalten:

$$F_o = \sum_{i=1}^k G_i \vec{f}_{c_i} = [G_1, \dots, G_k] \cdot \begin{bmatrix} f_{c_1} \\ \vdots \\ f_{c_k} \end{bmatrix} = G \cdot \vec{f}_c \qquad \vec{f}_c \in FC$$

wobei

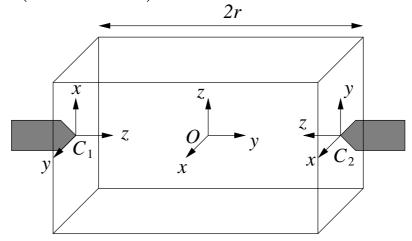
$$\vec{f_c} = (\vec{f_{c_1}}, \dots, \vec{f_{c_k}}) \in \mathbb{R}^m \quad m = \sum_{i=1}^k m_i$$
$$FC = FC_{c_1} \times \dots \times FC_{c_k} \subset \mathbb{R}^m$$

**Definition 2.1** Die Matrix  $G = [Ad_{T_{oc_1}^{-1}}^t B_{c_1}, \dots, Ad_{T_{oc_k}^{-1}}^t B_{c_k}]$  heißt Greifmatrix. Ein Griff ist damit vollständig definiert durch Angabe der Greifmatrix G und der Menge  $FC \subset \mathbb{R}^m$  der gültigen Kontaktkräfte. Die Menge der ausübbaren wrenches auf das Objekt ist damit definiert durch

$$F_o = G\vec{f_c} \quad \text{mit} \quad \vec{f_c} \in FC$$
.

Bemerkung: Diese Definition von G erlaubt es, unterschiedliche Reibungsmodelle für die verschiedenen Kontakte  $C_i$  anzunehmen. Die Kontaktkräfte innerhalb der Reibungskegel können im Prinzip beliebig groß werden – in der Realität benötigt man daher auch noch eine Begrenzung der Kontaktkraft  $f_n$  nach oben. Diese ist letztlich durch die Mechanik der Finger und die maximal ausübbaren Drehmomente in den Gelenken bestimmt.

Beispiel 2.2 (Griff einer Box)



Wir betrachten den abgebildeten Griff einer Box mit zwei Punktkontakten mit Reibung.

$$F_{o} = \sum A d_{T_{oc_{i}}^{-1}}^{t} F_{c_{i}} = \sum A d_{T_{oc_{i}}^{-1}}^{t} B_{c_{i}} \vec{f}_{c_{i}}$$

$$p_{oc_{1}} = (0, -r, 0)^{t} \quad p_{oc_{2}} = (0, r, 0)^{t} \quad B_{c_{i}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}^{t}$$

$$R_{oc_{1}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad R_{oc_{2}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \quad G = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ -r & 0 & 0 & 0 & 0 & 0 \\ 0 & r & 0 & -r & 0 & 0 \end{pmatrix}$$

$$FC_{c_{i}} = \{f \mid \sqrt{f_{x}^{2} + f_{y}^{2}} \leq \mu_{c_{i}} f_{z}\}$$

Der Griff kann also beliebige wrenches auf die Box ausüben, ausser Drehmomenten entlang der Körper-y-Achse. Dazu wäre mindestens ein Soft-Fingerkontakt notwendig. Natürlich müssen die Kräfte auch die Reibungsbedingung erfüllen, so dass die Normalenkraft groß genug sein muss, um entsprechende Kräfte in der x-z-Ebene zu erzeugen.

# 2.2 Bewertung von Griffen

#### 2.2.1 Force Closure

Eine wichtige Eigenschaft eines Griffes ist, externen Kräften entgegenwirken zu können, um das Objekt zu halten, bzw. direkt beliebige Kräfte auf das Objekt auszuüben, um dessen Lage zu verändern. Dabei müssen die aufgebrachten Kräfte immer innerhalb der Reibungskegel liegen, damit das Objekt nicht wegrutscht.

**Definition 2.3 (Force Closure)** Ein Griff (G, FC) heißt force-closure, wenn er jedem externen wrench  $F_e$  widerstehen kann:

$$G(FC) = \mathbb{R}^p \qquad \Leftrightarrow \qquad \forall F_e \in \mathbb{R}^p \quad \exists \vec{f_c} \in FC : G\vec{f_c} = -F_e$$

Force-closure ist durch die Existenz interner Kräfte charakterisiert, die keinen Nettoeffekt auf das Objekt haben:  $Gf_N = 0$  bzw.  $f_N$  ist Element des Nullraumes  $\mathcal{N}(G)$ . Eine Kontaktkraft  $f_N$  heißt interne Kraft, falls  $f_N \in \mathcal{N}(G) \cap FC$ . Liegt  $f_N$  sogar im Innern des Reibungskegels FC, heißt sie strikte interne Kraft.

**Satz 2.4** Ein Griff ist force-closure, gdw. G surjektiv ist und es strikte interne Kräfte  $f_N$  gibt:  $f_N \in \mathcal{N}(G) \cap int(FC) \neq \emptyset$ .

**Beweis**:  $\Leftarrow$ : Wähle  $F \in \mathbb{R}^p$ . Da G surjektiv, existiert ein  $f'_c$  mit  $F = Gf'_c$ .  $f'_c$  liegt jedoch nicht unbedingt innerhalb der Reibungskegel. Durch Hinzunahme von internen Kräften kann man jedoch auch dies erreichen. Sei  $f_N \in \mathcal{N}(G) \cap int(FC)$ . Dann gilt:

$$\lim_{\lambda \to \infty} \frac{f'_c + \lambda f_N}{\lambda} = f_N \in int(FC)$$

Damit existiert ein  $f_c = f'_c + \lambda f_N \in int(FC)$  mit  $Gf_c = Gf'_c = F$ .

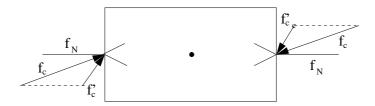


Abbildung 9: Mittels entsprechend hoher interner Kräfte kann jede Kontakt-Kraft in den Reibungskegel "verschoben" werden.

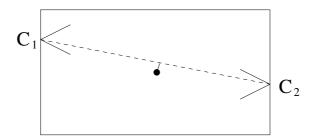
 $\Rightarrow$ : Der Griff sei force-closure. Die Surjektivität von G ist dann laut Definition erfüllt. Sei F beliebig. Dann existieren  $f_1, f_2 \in int(FC)$  mit  $Gf_1 = F$  und  $Gf_2 = -F$ . Es ist aber auch  $f_N = f_1 + f_2 \in int(FC)$  und  $Gf_N = F - F = 0$ .  $\square$ 

## Antipodale Griffe

Satz 2.5 Ein planarer Griff mit zwei Punktkontakten mit Reibung ist force-closure gdw. die Verbindungslinie zwischen den Kontaktpunkten innerhalb der Reibungskegel liegt.

Satz 2.6 Ein Griff in 3D mit zwei Soft-Finger-Kontakten ist force-closure gdw. die Verbindungslinie zwischen den Kontaktpunkten innerhalb der Reibungskegel liegt.

Solche Griffe heißen antipodale Griffe (Griffe mit gegenüberliegenden Kontakten). **Beweis**: Die Bedingung garantiert, dass sich die beiden Kontakte gegenüber liegen, so dass die Greifmatrix G surjektiv ist. Ausserdem sind Kontaktkräfte entlang der Verbindungslinie strikte interne Kräfte: Einerseits heben sich die Kraftanteile genau auf  $(F_1 = -F_2)$  und andererseits heben sich auch die Drehmomente auf, da der Hebel für beide derselbe ist – nämlich die kürzeste Verbindung von der Verbindungslinie zum Schwerpunkt.



#### 2.2.2 Form closure

**Definition 2.7** Ein Griff heißt form-closure, wenn er bereits unter der Annahme von *reibungsfreien Punktkontakten* force-closure ist.

Lemma 2.8 form-closure  $\Rightarrow$  force-closure

Form closure ist das strengere Kriterium und ein Griff der form-closure ist, ist auch force-closure mit jeder anderen Reibungsbedingung. Form closure benutzt lediglich die Griff-Geometrie in Form der Grasp-Matrix G, um den Griff zu bewerten. Die Greifmatrix von reibungsfreien Punktkontakten hat die Form:

$$G = \begin{pmatrix} n_{c_1} & \dots & n_{c_k} \\ p_{c_1} \times n_{c_1} & \dots & p_{c_k} \times n_{c_k} \end{pmatrix} \qquad n_{c_i} = R_{oc_i} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$
$$FC = \{ f \in \mathbb{R}^k \mid f_i \ge 0 \}$$

Für force-closure muss wieder gelten:  $G(FC) = \mathbb{R}^p$ , d.h.

$$\forall F_e \in \mathbb{R}^p \quad \exists f_i \ge 0 : G\vec{f_c} = \sum G_i f_i = -F_e$$

Die Spalten von G müssen den gesamten Raum  $\mathbb{R}^p$  also positiv aufspannen, weil nur drückende Kontaktkräfte  $f_i \geq 0$  erlaubt sind.

**Lemma 2.9** Sei ein Griff G von reibungslosen Punktkontakten gegeben. Dann sind folgende Aussagen äquivalent:

• Der Griff ist form-closure.

- Die Spalten von G spannen den gesamten  $\mathbb{R}^p$  positiv auf. (Es sind nur drückende Kontaktkräfte  $f_i \geq 0$  erlaubt.)
- Die konvexe Hülle der Spalten von G enthält eine  $\varepsilon$ -Umgebung des Ursprungs.

Die konvexe Hülle co(G) ist definiert als

$$co(G) = \{F = \sum f_i G_i \mid \sum f_i \le 1, f_i \ge 0\}.$$

Sie enthält für je zwei Vektoren  $v_i, v_j \in \text{co}(G)$  auch deren Verbindungsstrecke:  $\lambda v_i + (1-\lambda)v_j \in \text{co}(G)$ . Wenn eine  $\varepsilon$ -Umgebung von Null in co(G) enthalten ist, kann offenbar in jede Richtung des  $\mathbb{R}^6$  ein wrench positiv aus den  $G_i$  erzeugt werden. Dessen Länge ist zwar zunächst auf die  $\varepsilon$ -Umgebung beschränkt, jedoch erfüllt auch jeder mit  $\lambda > 0$  skalierte wrench die Reibungsbedingung  $f_i \geq 0$ .

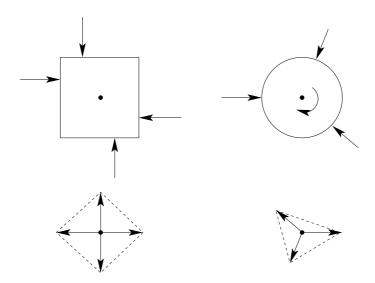


Abbildung 10: Beispiele von 2d-Griffen. Box: form-closure. Kreis: nicht form-closure, da zwar allen externen (linearen) Kräften widerstanden werden kann, aber keine Drehmomente erzeugt werden können. Kugelige Objekte können generell nicht form-closure sein, da die Kontaktkräfte immer auf den Schwerpunkt zeigen und somit kein Hebelarm bleibt, der ein Drehmoment erzeugen könnte.

Unten: Konvexe Hülle der Kräfte – der Ursprung liegt zwar in beiden Fällen innerhalb, aber Drehmomente sind dabei noch nicht berücksichtigt.

Satz 2.10 (Caratheodory) Wenn eine Menge  $X = \{v_1, \dots, v_k\} \subset \mathbb{R}^p$  den gesamten Raum  $\mathbb{R}^p$  positiv aufspannt, gilt  $k \geq p+1$ .

Man braucht also mindestens p+1=7 reibungsfreie Punktkontakte für einen formclosure-Griff. Nach einem Satz von Steinitz kommt man mit 2p=12 Kontakten aus, vorausgesetzt, es existiert überhaupt ein form-closure Griff. Für Kugeln gilt dies z.B. nicht, da alle Kontaktkräfte auf den Schwerpunkt zeigen und daher kein Drehmoment erzeugen können.

## 2.2.3 Qualitätsmaße

Die meisten Qualitätsmaße beruhen auf einer linearen Approximation der Haft-Reibungskegel und anschließender Auswertung der resultierenden konvexen Hülle. Ein (Haft-)Reibungskegel wird damit approximiert durch:

$$FC_{c_i} \approx \{\vec{f}_{c_i} = \sum_{j=1}^n \alpha_{ij} \vec{f}_{ij} \mid \alpha_{ij} \ge 0, \sum_{j=1}^n \alpha_{ij} \le 1\}.$$

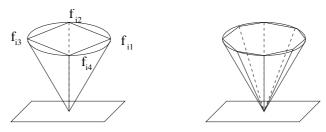


Abbildung 11: Approximation eines Reibungskegels durch eine mehrseitige Pyramide.

Die Kraftvektoren  $\vec{f}_{ij}$  liegen auf dem Rand des Reibungskegels des *i*-ten Kontakts und approximieren ihn mittels einer Pyramide. Man wählt ihren Betrag so, dass der Normalenanteil  $f_i^n = \langle f_{ij}, \hat{n}_i \rangle \hat{n}_i$  jeweils Betrag 1 hat – es kann also maximal eine Normalenkraft vom Betrag 1 ausgeübt werden. Jeder Kraftvektor  $\vec{f}_{ij}$  erzeugt einen wrench im Objekt-KS:

$$F_{ij}^o = Ad_{T_{oc_i}^{-1}}^t \begin{pmatrix} \vec{f}_{ij} \\ 0 \end{pmatrix} .$$

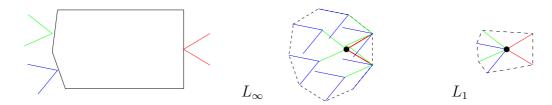
Mittels der  $F_{ij}^o$  wird nun die konvexe Hülle bestimmt. Je nach Wahl der Norm  $\|\vec{f_c}\|$  für die Kontaktkräfte ergeben sich verschiedene Approximationen der Menge W der ausübbaren wrenches durch konvexe Hüllen:

1. 
$$\|\vec{f_c}\|_{\infty} = \max_{i} |f_i^n| \le 1$$
  $W_{L_{\infty}} = \operatorname{co}\left(\bigoplus_{i=1}^k \{F_{i1}^o, \dots, F_{in}^o\}\right)$   $W_{L_1} \subset W_{L_{\infty}}$   
2.  $\|\vec{f_c}\|_{1} = \sum_{i} |f_i^n| \le 1$   $W_{L_1} = \operatorname{co}\left(\bigcup_{i=1}^k \{F_{i1}^o, \dots, F_{in}^o\}\right)$ 

Man beachte, dass in die Berechnung der Norm  $\|\vec{f_c}\|$  immer nur die Normalenanteile  $f_i^n = f_i^z$  eingehen. Um unabhängig von der Objektgröße zu sein, ist es sinnvoll die Drehmomente mit der mittleren Objektgröße r zu skalieren:

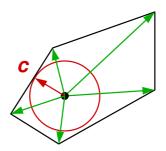
$$F_{ij}^{o} = \begin{pmatrix} R_{oc_i} & 0\\ \frac{1}{r}\hat{p}_{oc_i}R_{oc_i} & R_{oc_i} \end{pmatrix} \begin{pmatrix} \vec{f}_{ij}\\ 0 \end{pmatrix}$$

Beispiel 2.11 Griff eines Rechtecks mit drei Punktkontakten mit Reibung.

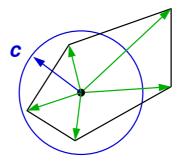


Die wrench-polytope  $W_{L_{\infty}}$  und  $W_{L_{1}}$  kann man nun nutzen, um die Qualität des Griffs zu bestimmen:

- Falls der Ursprung innerhalb des wrench-Polytops liegt, ist der Griff force-closure.
- Bestimme den kleinsten Abstand vom Ursprung zum Rand von  $W_L$ . Dies entspricht dem Radius  $\varepsilon$  der größten (6-dim.) Kugel um den Ursprung, die gerade noch in das Polytop passt. Die zugehörige Richtung, gibt den wrench an, dem am schlechtesten widerstanden werden kann bzw. dem gerade noch widerstanden werden kann, wenn die Kontaktkräfte entsprechend der Norm beschränkt werden.



 $\bullet$ Bestimme den mittleren Abstand vom Ursprung zu den Grenzflächen von  $W_L$ 



### Probleme:

- $\bullet$  Drehmomente hängen von der Wahl des Objekt-KS O ab. Benutze den Massenschwerpunkt.
- ullet wrench-Polytop ist konservative Approximation von W und abhängig von der Ausrichtung der approximierenden Pyramide.

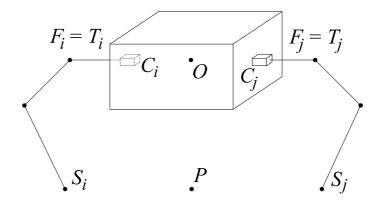


Abbildung 12: Kinematik eines mehrfingrigen Griffes

- Die Anzahl der wrench-Vektoren in der direkten Summe ( $\bigoplus$ ) ist  $N_{L_{\infty}} = n^k$  und in der Vereinigung ( $\bigcup$ )  $N_{L_1} = n \cdot k$ . Rechenaufwand für co(·):  $\mathcal{O}(N \log N)$ .
- Eine gute Approximation mit  $n \leq 8$  Stützpunkten pro Reibungskegel braucht sehr lange. Wähle  $n \approx 6$ .

Wenn man sich nur für bestimmte Unterräume des 6-dim. wrench-Raumes interessiert, z.B. weil man nur Kräfte oder nur Drehmomente ausüben will, kann man auch Projektionen der wrench-polytope W auf diese Räume betrachten und die Maße dort auswerten.

Vortex-Demo

# 2.3 Die Greif-Bedingung

Bislang haben wir Griffe ohne Rücksicht auf die Finger selbst betrachtet und angenommen, dass Kontaktkräfte innerhalb der  $FC_i$  tatsächlich von den Fingern aufgebracht werden können. Nun wollen wir die Manipulierbarkeit des Objektes betrachten, d.h. die Frage welche Objektbewegungen von den Fingern tatsächlich erzeugt werden können. Dazu müssen wir die Hand-Kinematik einbeziehen.

Wir betrachten folgende Koordinatensysteme (Abb. 12):

- in der Handfläche (Palm): P
- $\bullet$  die Fingerbasis  $S_i$  für jeden Finger, fest relativ zu P
- das Objekt-KS O
- $\bullet\,$  Kontakt-KS  $C_i,$  fest relativ zu O
- Finger-Kontakt-KS  $F_i$  im Kontakt, fest relativ zur Fingerspitze

Die Greifbedingung besagt, dass sich die Kontakt-KS  $C_i$  im Objekt und in den Fingern  $(F_i)$  entlang der Richtungen  $B_{c_i}$ , entlang derer Kontakt-Kräfte ausgeübt werden können, nicht relativ zueinander bewegen dürfen:

$$B_{c_i}^t V_{f_i c_i}^b = 0. (*)$$

Wir drücken nun  $V_{f_ic_i}^b$  in bekannten Größen aus. Für die Geschwindigkeiten nutzen wir die Transformationsgleichungen:

$$A \to B \to C: \qquad V_{ac}^b = Ad_{T_{bc}^{-1}}V_{ab}^b + V_{bc}^b$$
 (1)

$$V_{ab}^b = -Ad_{T_{ba}}V_{ba}^b \tag{2}$$

und erhalten

$$F_{i} \to P \to C_{i}: \qquad V_{f_{i}c_{i}}^{b} \stackrel{(1)}{=} Ad_{T_{pc_{i}}^{-1}} V_{f_{i}p}^{b} + V_{pc_{i}}^{b} \stackrel{(2)}{=} -Ad_{T_{pc_{i}}}^{-1} Ad_{T_{pf_{i}}} V_{pf_{i}}^{b} + V_{pc_{i}}^{b}$$

$$P \to O \to C_{i}: \qquad V_{pc_{i}}^{b} \stackrel{(1)}{=} Ad_{T_{oc_{i}}^{-1}} V_{po}^{b} + V_{oc_{i}}^{b} = Ad_{T_{oc_{i}}^{-1}} V_{po}^{b}$$

und zusammen:

$$V_{f_ic_i}^b = -Ad_{T_{pc_i}}^{-1}Ad_{T_{pf_i}}V_{pf_i}^b + Ad_{T_{oc_i}}^{-1}V_{po}^b$$

Dies setzen wir nun in (\*) ein:

$$B_{c_i}^t A d_{T_{pc_i}}^{-1} A d_{T_{pf_i}} V_{pf_i}^b = \underbrace{B_{c_i}^t A d_{T_{oc_i}^{-1}}}_{(A d_{T_{oc_i}^{-1}} B_{c_i})^t = G_i^t} V_{po}^b = G_i^t \cdot V_{po}^b$$

Die Tool-Geschwindigkeit  $V_{pf_i}^b$  relativ zur Handfläche wird durch die Jacobi-Matrix beschrieben:

$$V_{pf_i}^b = J_{pf_i}^b(\theta_i) \cdot \dot{\theta_i} = Ad_{T_{pf_i}(\theta_i)}^{-1} J_{pf_i}^s(\theta_i) \cdot \dot{\theta_i}$$

wobei  $\theta_i$  der Gelenkwinkel-Vektor des *i*-ten Fingers ist. Wir erhalten also:

$$B_{c_{i}}^{t}Ad_{T_{pc_{i}}}^{-1}\underbrace{Ad_{T_{pf_{i}}}^{-1}Ad_{T_{pf_{i}}}^{-1}}_{\mathbf{1}}J_{pf_{i}}^{s}(\theta_{i})\cdot\dot{\theta_{i}}=G_{i}^{t}\cdot V_{po}^{b}\;(=B_{c_{i}}^{t}V_{pc_{i}}^{b})$$

Diese Gleichung hat eine sinnvolle Interpretation: Die rechte Seite ist die Geschwindigkeit  $V_{pc_i}^b$  des *i*-ten Kontakt-KS relativ zur Handfläche, maskiert mit den relevanten Bewegungsrichtungen  $B_{c_i}$ . Die linke Seite gibt die Tool-Geschwindigkeit  $F_i$  relativ zu P an, transformiert diese mittels der Adjungierten  $Ad_{T_{pc_i}}^{-1}$  ins  $C_i$ -KS und maskiert sie wiederum mit  $B_{c_i}$ .

Wir fassen die Bedingungen für alle k Finger zusammen:

**Definition 2.12** Die Hand-Jacobi-Matrix für k Kontakte und Finger (mit  $n_i$  DOFs) ist definiert als:

$$J_{H}(\theta, T_{po}) = \begin{pmatrix} B_{c_{1}}^{t} A d_{T_{pc_{1}}}^{-1} J_{pf_{1}}^{s}(\theta_{1}) & & \\ & \ddots & \\ & & B_{c_{k}}^{t} A d_{T_{pc_{k}}}^{-1} J_{pf_{k}}^{s}(\theta_{k}) \end{pmatrix} \in \mathbb{R}^{m \times n} \qquad n = \sum_{m_{i}} n_{i}$$

$$= \begin{pmatrix} B_{c_{1}}^{t} A d_{T_{s_{1}c_{1}}}^{-1} J_{s_{1}f_{1}}^{s}(\theta_{1}) & & \\ & \ddots & & \\ & & B_{c_{k}}^{t} A d_{T_{s_{k}c_{k}}}^{-1} J_{s_{k}f_{k}}^{s}(\theta_{k}) \end{pmatrix}$$

$$= \begin{pmatrix} B_{c_{1}}^{t} A d_{T_{s_{1}c_{1}}}^{-1} J_{s_{1}f_{1}}^{s}(\theta_{1}) & & \\ & \ddots & & \\ & & B_{c_{k}}^{t} A d_{T_{s_{k}c_{k}}}^{-1} J_{s_{k}f_{k}}^{s}(\theta_{k}) \end{pmatrix}$$

Damit können wir die Greif-Bedingung für die gesamte Hand schreiben als:

$$J_H(\theta, T_{po}) \cdot \dot{\theta} = G^t \cdot V_{po}^b \quad \in \quad \mathbb{R}^m \,. \tag{2.1}$$

**Bemerkung:** Die Greif-Bedingung erfasst nur Kontakte mit Fingerspitzen und es kommen auch nur solche Finger vor, die überhaupt Kontakt haben. Die Vektoren  $\vec{v} = J_H \cdot \dot{\theta} = G^t \cdot V_{po}^b \in \mathbb{R}^m$  fassen die Relativ-Geschwindigkeiten aller Kontakt-KS relativ zur Palm  $(V_{pc_i}^b)$  zusammen und betrachten dabei nur die relevanten Richtungen (maskiert mit  $B_{c_i}$ ).

# 2.4 Manipulierbarkeit

**Definition 2.13** Ein mehrfingriger Griff  $(G, FC, J_H(\theta, T_{po}))$  heißt manipulierbar, falls jede Objektbewegung  $V_{po}^b \in \mathbb{R}^p$  durch entsprechende Gelenkbewegungen  $\dot{\theta}$  erzeugt werden kann, so dass Gleichung (2.1) erfüllt ist.

Satz 2.14 Ein Griff ist manipulierbar gdw.  $\mathcal{R}(G^t) \subseteq \mathcal{R}(J_H(\theta, T_{po}))$ .

#### Beweis:

Das Bild von  $G^t$  ist eine Teilmenge des Bildes von  $J_H$ , d.h. alle Vektoren  $\vec{v} \in \mathbb{R}^m$ , die  $G^t$  erzeugen kann, kann  $J_H$  ebenfalls erzeugen.

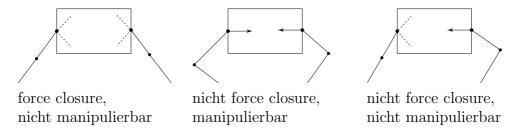
 $\Rightarrow$ : Sei  $\vec{v} \in \mathcal{R}(G^t)$ , d.h. es gibt  $V^b_{po}$  mit  $\vec{v} = G^t \cdot V^b_{po}$ . Da der Griff manipulierbar ist, gibt es dazu auch ein  $\dot{\theta}$  mit  $\vec{v} = G^t \cdot V^b_{po} = J_H \cdot \dot{\theta} \in \mathcal{R}(J_H)$ .  $\Leftarrow$ : Sei  $V^b_{po}$  beliebig, und sei  $\vec{v} = G^t \cdot V^b_{po} \in \mathbb{R}^m$  der zug. Bildvektor. Da  $\vec{v}$  nach

 $\Leftarrow$ : Sei  $V_{po}^b$  beliebig, und sei  $\vec{v} = G^t \cdot V_{po}^b \in \mathbb{R}^m$  der zug. Bildvektor. Da  $\vec{v}$  nach Voraussetzung auch ein Bildvektor von  $J_H$  ist, gibt es also auch ein  $\dot{\theta}$  mit  $\vec{v} = J_H \cdot \dot{\theta}$ .

### Bemerkungen:

- Manipulierbarkeit erfordert nicht, dass  $J_H$  invertierbar ist. Vielmehr können viele Gelenkbewegungen  $\dot{\theta}$  existieren, die dieselbe Objektbewegung  $V_{po}^b$  hervorrufen.
- Falls n > m (Redundanz), existieren z.B. interne Bewegungen  $\dot{\theta} \in \mathcal{N}(J_H)$ , die keine Objekt-Bewegung (in relevanten Richtungen  $B_{c_i}$ ) erzeugen.

• force-closure und Manipulierbarkeit sind unabhängige Konzepte:



#### 2.5 Strukturelle Kräfte

Wir können mittels der Hand-Jacobi-Matrix auch Kontakt-Kräfte in Gelenk-Kräfte umrechnen. Die Herleitung erfolgt mittels der geleisteten Arbeit W:

$$\int_{t_1}^{t_2} \dot{\theta}_i^t \tau_i \, dt = W = \int_{t_1}^{t_2} \left( V_{pc_i}^b \right)^t \cdot F_{c_i} \, dt \quad \text{für alle } t_1 \leq t_2 \in \mathbb{R}$$

$$\dot{\theta}_i^t \cdot \tau_i = \left( A d_{T_{oc_i}^{-1}} V_{po}^b \right)^t \cdot B_{c_i} \vec{f}_{c_i} = \left( B_{c_i}^t A d_{T_{oc_i}^{-1}} V_{po}^b \right)^t \cdot \vec{f}_{c_i}$$

$$= \left( G_i^t V_{po}^b \right)^t \cdot \vec{f}_{c_i} = \left( J_H^i \cdot \dot{\theta}_i \right)^t \cdot \vec{f}_{c_i} = \dot{\theta}_i^t \cdot \left( J_H^i \right)^t \cdot \vec{f}_{c_i}$$
für alle  $\dot{\theta}$ 

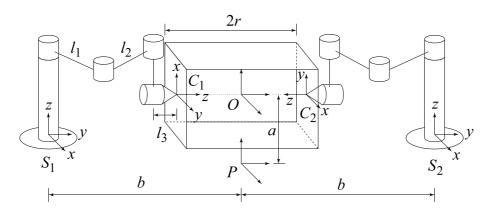
Fassen wir wieder alle Finger i = 1, ..., k zusammen, erhalten wir:

$$\tau = J_H^t \cdot \vec{f_c} \qquad \vec{f_c} \in FC$$

#### Bemerkungen:

- Kontaktkräfte  $\vec{f}_c^s \in \mathcal{N}(J_H^t)$  im Nullraum von  $J_H^t$  heißen strukturelle Kräfte. Diese können nicht durch die Gelenk-Drehmomente  $\tau$  aktiv erzeugt werden, sondern werden passiv von der Hand-Konstruktion aufgebracht. In diesem Falle ist der Griff nicht manipulierbar da das Objekt in den entsprechenden Richtungen nicht aktiv bewegt werden kann –, kann aber trotzdem forceclosure sein da die Konstruktion die notwendigen Kräfte aufbringt.
- Daraus folgt, dass für Manipulierbarkeit die Matrix  $J_H$  vollen Zeilenrang haben muss ansonsten existiert ein nichtleerer Nullraum von  $J_H^t$ . In diesem Fall ist  $J_H$  surjektiv.

Beispiel 2.15 (Zwei SCARAs greifen eine Box) mittels Soft-Finger-Kontakten.



Zur Greif-Matrix G aus Beispiel 2.2 kommt noch die Soft-Finger-Komponente hinzu:

$$G = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -r & 0 & 0 & 0 & 0 & r & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & r & 0 & 0 & -r & 0 & 0 & 0 \end{pmatrix} \quad B_{c_i} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{rang}(G) = 6$$

Die Jacobi-Matrix des SCARA-Manipulators ist aus Beispiel 1.21 bekannt. Zur Vereinfachung setzen wir die Länge der Fingerspitze  $l_3 = 0$ .

$$J_{s_i f_i}^s = \begin{pmatrix} 0 & l_1 \cos \theta_1 & l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) & 0 \\ 0 & l_1 \sin \theta_1 & l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Wir betrachten nur die (einfache) Lage des Objektes in der Zeichnung – für eine allgemeine Lage würde die Greif-Bedingung sehr komplex werden.

$$R_{po} = \mathbf{1}$$
  $\vec{p}_{po} = [0, 0, a]^t$ 

Die Adjungierten  $Ad_{T_{s_ic_i}^{-1}}$  lesen wir direkt ab: Mit  $\vec{p}_{s_1c_1}=[0,b-r,a]^t$  und  $\vec{p}_{s_2c_2}=[0,r-b,a]^t$  können wir die Adjungierten direkt ablesen:

$$R_{s_{1}c_{1}}^{t} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} - \hat{p}_{s_{1}c_{1}} = \begin{pmatrix} b-r & 0 & 0 \\ 0 & a & r-b \\ -a & 0 & 0 \end{pmatrix} \Rightarrow Ad_{T_{s_{1}c_{1}}}^{-1} = \begin{pmatrix} R_{s_{1}c_{1}}^{t} & \begin{bmatrix} b-r & 0 & 0 \\ 0 & a & r-b \\ -a & 0 & 0 \end{bmatrix} \\ 0 & R_{s_{1}c_{1}}^{t} \end{pmatrix}$$

$$R_{s_{2}c_{2}}^{t} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} - \hat{p}_{s_{2}c_{2}} = \begin{pmatrix} r-b & 0 & 0 \\ 0 & a & b-r \\ -a & 0 & 0 \end{pmatrix} \Rightarrow Ad_{T_{s_{2}c_{2}}}^{-1} = \begin{pmatrix} R_{s_{2}c_{2}}^{t} & \begin{bmatrix} 0 & a & b-r \\ r-b & 0 & 0 \\ a & 0 & 0 \end{bmatrix} \\ 0 & R_{s_{2}c_{2}}^{t} \end{pmatrix}$$

$$\vec{p}_{s_{2}c_{2}} = \begin{bmatrix} 0, -b+r, a \end{bmatrix}^{t}$$

Die beiden Teile  $J_i=B^t_{c_i}\cdot Ad^{-1}_{T_{s_ic_i}}J^s_{s_if_i}(\theta_i)$  der Hand-Jacobi-Matrix  $J_H=\begin{pmatrix}J_1\\J_2\end{pmatrix}$  sehen nun wie folgt aus:

$$J_{1} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ -b+r & -b+r+l_{1}c_{1} & -b+r+l_{1}c_{1}+l_{2}c_{12} & 0 \\ 0 & l_{1}s_{1} & l_{1}s_{1}+l_{2}s_{12} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$J_{2} = \begin{pmatrix} +b-r & +b-r+l_{1}c_{1} & +b-r+l_{1}c_{1}+l_{2}c_{12} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -l_{1}s_{1} & -l_{1}s_{1}-l_{2}s_{12} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Der Griff ist force-closure, weil er antipodal ist und Soft-Finger-Kontakte benutzt. Er ist aber nicht manipulierbar, da

$$G^t \cdot [0, 0, 0, 0, 1, 0]^t = [0, 0, 0, 1, 0, 0, 0, -1]^t \notin \mathcal{R}(J_H).$$

Der Nullraum  $\mathcal{N}(J_H^t)$  wird von folgenden strukturellen Kräften aufgespannt:

$$\vec{f_c^s} = [0, 0, 0, 1, 0, 0, 0, 0]^t$$
 und  $\vec{f_c^s} = [0, 0, 0, 0, 0, 0, 0, 1]^t$ .

Diese entsprechen den Objekt-wrenches  $F_o = G\vec{f}_c^s$ :

$$F_o = [0, 0, 0, 0, 1, 0]^t$$
 und  $F_o = [0, 0, 0, 0, -1, 0]^t$ ,

also Drehmomenten um die Objekt-y-Achse, die von dem Griff nicht erzeugt werden können, aber die mittels der Soft-Finger-Kontakte und der Konstruktion aufgebracht werden können, um externen Störungen zu widerstehen.

## 2.6 Zusammenfassung

finger contact object velocity domain 
$$\dot{\theta} \xrightarrow{J_H} B_c^t V_{pc}^b \xleftarrow{G^t} V_{po}^b$$
force 
$$\tau \xleftarrow{J_H^t} \vec{f_c} \xrightarrow{G} F_o$$

Eigenschaft	Beschreibung	in Formeln
force closure	jedem wrench $F_e$ kann (passiv) wider-	$G(FC) = \mathbb{R}^p$
	standen werden	
manipulierbar	jede Objektbewegung kann (aktiv) er-	$\mathcal{R}(G^t) \subseteq \mathcal{R}(J_H)$
	zeugt werden	
interne Kraft	Kontaktkräfte $\vec{f}_N$ , die keinen Nettoef-	$\vec{f}_N \in \mathcal{N}(G) \cap int(FC)$
	fekt auf das Objekt haben	
interne Bewegung	Gelenk-Bewegungen $\dot{\theta}$ , die keine	$\dot{\theta}_N \in \mathcal{N}(J_H)$
	Objekt-Bewegung $V_{po}^b$ erzeugen	
strukturelle Kraft	Kontaktkräfte $\vec{f}_c^s$ , die ohne Gelenk-	$\vec{f_c^s} \in \mathcal{N}(J_H^t)$
	Drehmomente $\tau$ erzeugt werden	

# 2.7 Optimierung von Kontaktkräften $\vec{f_c}$

Wir suchen im folgenden optimale Griffe. Man kann sich verschiedene Optimalitätskriterien und Nebenbedingungen vorstellen:

- stabiler Griff:  $\vec{f_c} \in FC$
- force-closure oder manipulierbarer Griff
- $\bullet$ Begrenzung der Kontaktkräfte:  $\|\vec{f_c}\| < M_f$  (bei zerbrechlichen Objekten)

- Begrenzung der Gelenk-Drehmomente:  $\tau_i^L \leq \tau_i \leq \tau_i^U$
- minimale Kontaktkräfte: min  $\|\vec{f_c}\|$
- minimale Gelenk-Drehmomente: min  $\|\tau\|$
- ullet hohe Griffsicherheit: große Entfernung zum Rand des Reibungskegels FC

Eine Lösung erfolgt i.d.R. numerisch, wobei das Hauptproblem die nichtlinearen (quadratischen) Reibungsbedingungen  $\vec{f_c} \in FC$  sind. Ihre Formulierung mittels linearer Matrixungleichungen (LMI) ermöglicht die Anwendung effizienter (polynomiell konvergenter) Optimierungsverfahren (z.B. Interior-Point-Optimierung).

#### Einschub: Lineare Matrixungleichungen (LMI)

**Definition 2.16** Sei  $S = S^t \in \mathbb{R}^{n \times n}$  eine symmetrische Matrix. Die Schreibweisen

$$S > 0$$
 "S ist positiv [definit]" und  $S \ge 0$  "S ist positiv semidefinit"

verallgemeinern die entsprechenden skalaren Ungleichungen und stehen für

$$S>0$$
  $\Leftrightarrow$   $\vec{x}^t S \vec{x} > 0$  für alle  $\vec{x} \neq 0$   
  $\Leftrightarrow$  alle (reellen) Eigenwerte  $\lambda_i$  von  $S$  sind positiv:  $\lambda_i > 0$   
  $S \geq 0$   $\Leftrightarrow$   $\vec{x}^t S \vec{x} \geq 0$  für alle  $\vec{x} \neq 0$   
  $\Leftrightarrow$  alle  $\lambda_i > 0$ 

**Definition 2.17** Eine lineare Matrixungleichung ist eine Gleichung der Form

$$P(x) = S_0 + x_1 S_1 + x_2 S_2 + \dots + x_k S_k > 0$$

wobei die  $S_i$  symmetrisch sind. Die  $x_i$  sind gesuchte *skalare* Variablen, um die Ungleichung zu erfüllen. Beachten Sie, dass die  $x_i$  linear in die Ungleichung eingehen.

**Satz 2.18** Sei eine LMI der Form  $P(x) = S_0 + \sum_{i=1}^m x_i S_i$  und eine beliebige affine Transformation x = Az + b mit  $A \in \mathbb{R}^{m \times l}$  und  $b \in \mathbb{R}^m$  gegeben. Dann ist P'(z) := P(Az + b) wieder eine LMI in der neuen Variablen  $z \in \mathbb{R}^l$ .

#### **Beweis**:

$$P'(z) = S_0 + \sum_{i=1}^m S_i \left( \sum_{j=1}^l a_{ij} z_j + b_i \right) = \underbrace{S_0 + \sum_{i=1}^m b_i S_i}_{S'_0} + \underbrace{\sum_{j=1}^l z_j \left( \sum_{i=1}^m a_{ij} S_i \right)}_{S'_j}$$

Die  $S_j'$  sind symmetrisch, da sie Summen von symmetrischen Matrizen sind.  $\square$  Ende Einschub.

 ${f Satz}$  2.19 Die Reibungskegel-Bedingungen FC können als LMI ausgedrückt werden.

#### **Beweis**:

• reibungsfreier Punktkontakt:  $P_i = f \ge 0$ 

• Punktkontakt mit Reibung: 
$$P_i = \begin{pmatrix} \mu f_z + f_x & f_y \\ f_y & \mu f_z - f_x \end{pmatrix} \geq 0$$

• Softfinger-Kontakt: 
$$P_i = \begin{pmatrix} f_z & 0 & 0 & \alpha f_x \\ 0 & f_z & 0 & \alpha f_y \\ 0 & 0 & f_z & \beta f_\tau \\ \alpha f_x & \alpha f_y & \beta f_\tau & f_z \end{pmatrix} \ge 0 \qquad \alpha = \mu^{-1} \quad \beta = \gamma^{-1}$$

Wir betrachten lediglich den Punktkontakt mit Reibung. Für die Eigenwerte von  $P_i$  gilt:  $\lambda_{\pm} = \mu f_z \pm \sqrt{f_x^2 + f_y^2}$ . Die Bedingung  $\lambda_{-} \geq 0$  ist damit äquivalent zur Reibungsbedingung  $\sqrt{f_x^2 + f_y^2} \leq \mu f_z$ . Ausserdem kann  $P_i \geq 0$  als LMI geschrieben werden:

$$P_i = f_z \begin{pmatrix} \mu & 0 \\ 0 & \mu \end{pmatrix} + f_x \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + f_y \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \ge 0$$

Das simultane Bestehen der LMI  $P_i \ge 0$  für alle k Kontakte kann in einer einzelnen LMI zusammengefasst werden:

$$\vec{f_c} \in FC \qquad \Leftrightarrow \qquad P(\vec{f_c}) = \begin{pmatrix} P_1(\vec{f_{c_1}}) & & & \\ & P_2(\vec{f_{c_2}}) & & \\ & & \ddots & \\ & & & P_k(\vec{f_{c_k}}) \end{pmatrix} \ge 0$$

Da jeder Matrixblock  $P_i$  eine Linearkombination aus mehreren (drei) Basismatrizen ist, ergibt sich eine entsprechende Darstellung auch für die gesamte Matrix:

$$P(\vec{f_c}) = f_{c_1}^z \begin{pmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_1 & 0 \\ 0 & 0 \end{pmatrix} + f_{c_1}^x \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 \end{pmatrix} + \dots + f_{c_k}^y \begin{pmatrix} 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 \end{pmatrix}$$

#### Bemerkungen:

- $\bullet\,$  Die triviale Lösung  $\vec{f_c}=0$ erfüllt die Reibungsbedingung.
- Wenn die Kontaktkräfte  $\vec{f_c}$  strikt im Inneren der Reibungskegel liegen sollen, fordert man  $P_i > 0$  anstatt  $P_i \geq 0$  die triviale Lösung  $\vec{f_c} = 0$  gibt es dann nicht mehr.
- $\bullet\,$  In die LMI  $P(\vec{f_c})$ können verschiedene Kontaktmodelle eingehen.

Skalare Ungleichungen können durch geeignete Diagonalmatrizen in LMI-Form ausgedrückt werden:

$$m \le x \le M \qquad \Leftrightarrow \qquad \begin{array}{c} -m+x \ge 0 \\ M-x \ge 0 \end{array} \qquad \Leftrightarrow \qquad \begin{pmatrix} -m \\ M \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} x \ge 0$$

Z.B. können wir die ausübbaren Drehmomente auf das Intervall

$$\begin{split} \tau_{\min} &\leq \tau = J_H^t \vec{f_c} \leq \tau_{\max} \\ \Leftrightarrow & \tau_i^{\min} \leq \sum_{j=1}^m J_{ji}^H f_j \leq \tau_i^{\max} \end{split}$$

beschränken, indem wir zusätzlich die LMI  $T = \text{diag}(T_1, \dots, T_n) \ge 0$  berücksichtigen, wobei:

$$T_i = \begin{pmatrix} -\tau_i^{\min} + \sum J_{ji}^H f_j & \\ \tau_i^{\max} - \sum J_{ji}^H f_j \end{pmatrix} \ge 0.$$

Analog können wir die Kontaktkräfte nach oben beschränken, z.B.

$$\|\vec{f_c}\|_1 = \sum f_{c_i}^n \le 1 \qquad \Leftrightarrow \qquad 1 - \sum f_{c_i}^n \ge 0.$$

Wir können nun verschiedene Optimierungsprobleme mittels der LMIs beschreiben.

Force-closure Falls G surjektiv ist, bestimme eine Basis  $N := [n_1, \cdots, n_l] \in \mathbb{R}^{m \times l}$  des Nullraumes von G. Jeder Vektor  $\vec{f_N} \in \mathcal{N}(G) \subset \mathbb{R}^m$  kann dann als  $\vec{f_N} = Nz$  mit  $z \in \mathbb{R}^l$  geschrieben werden. Falls eine Lösung z der resultierenden LMI gefunden werden kann (feasibility problem), ist der Griff force-closure. Die Idee ist also, nur potentielle innere Kräfte  $\vec{f_N} \in \mathcal{N}(G)$  zu betrachten, und die Reibungsbedingungen (LMI) für diese umzuschreiben.

Minimierung der Kontaktkräfte Bei gegebenem externen wrench  $F_e$  werden minimale Kontaktkräfte  $\vec{f_c}$  gesucht, die dem wrench noch widerstehen können. Wir bestimmen zunächst eine beliebige Lösung  $\vec{f_0}$  der Gleichung

$$G\vec{f_c} = -F_e \,.$$

Wir können aber auch beliebige Kräfte  $\vec{f}_N \in \mathcal{N}(G)$  addieren:

$$\mathcal{M} = \{ \vec{f_c} \in \mathbb{R}^m \, | \, G\vec{f_c} = -F_e \} = \{ \vec{f_0} + Nz \, | \, z \in \mathbb{R}^l \}$$

wobei  $N \in \mathbb{R}^{m \times l}$  wieder eine Basis des Nullraumes von G ist. Wir können nun die LMI wieder umschreiben:  $P(\vec{f_c}) = P(\vec{f_0} + Nz)$ , so dass die Reibungsbedingungen bzgl. der  $z \in \mathbb{R}^l$  ausgedrückt sind. Wir minimieren die Summe der Fingerkräfte:

$$\min \Psi(\vec{f_c}) = w^t \cdot \vec{f_c} \qquad \vec{f_c} \in FC$$

wobei der Vektor  $w = [w_1, \dots, w_k]^t$  die Gewichtung der einzelnen Kraftkomponenten angibt, z.B.:  $w_i = [0, 0, 1]^t$  für Punktkontakte mit Reibung, um nur die Normalkräfte zu berücksichtigen. Das Optimierungsproblem können wir also schreiben als:

$$\min \Psi(z) = \Psi(\vec{f_0} + Nz) = \underbrace{w^t \cdot \vec{f_0}}_{const} + (w^t N) \cdot z \qquad P(z) > 0.$$

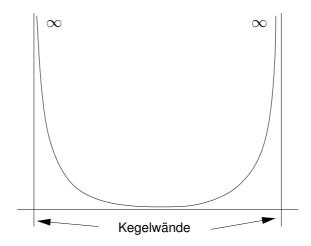
Die optimale Lösung einer solchen linearen Funktion unter konvexen Nebenbedingungen (FC) liegt immer auf dem Rand der Reibungskegel. Daher: möglichst großer Abstand zum Rand der Reibungskegel:

$$\min \Psi(\vec{f_c}) = w^t \cdot \vec{f_c} + \underbrace{\log(\det P^{-1}(\vec{f_c}))}_{-\sum \log(\lambda_i)} \qquad \vec{f_c} \in FC.$$

Die Eigenwerte  $\lambda_i$  von  $P(\vec{f_c})$  bilden ein Maß dafür, wie weit  $\vec{f_c}$  innerhalb der Reibungskegel FC liegt, denn

$$FC = \{\vec{f_c} \mid P(\vec{f_c}) > 0\} = \{\vec{f_c} \mid \lambda_i(P(\vec{f_c})) > 0\},\$$

d.h.  $-\sum \log(\lambda_i)$  hat die Gestalt einer Barrierefunktion, die im Innern der FC klein ist, aber zum Rand steil ansteigt:



Die Lösung des Optimierungsproblems liefert optimale Kontaktkräfte  $\vec{f_c}$ , die dem gewählten externen wrench  $F_e$  widerstehen können.

# 2.8 Anwendungsorientierung und Virtuelle Kontakte

#### 2.8.1 Task-Ellipsoid

In Anwendungen wollen wir i.d.R. Kräfte nur in bestimmten Richtungen ausüben können, z.B. Schieben eines Objektes über den Tisch oder Anheben eines Objektes. In dieser Richtung wollen wir dann eine besonders große Kraft ausüben können, wohingegen uns Störungen in anderen Richtungen nicht so sehr interessieren.





Eine solche Aufgabe kann als Task-Ellipsoid beschrieben werden:

$$\mathcal{E}_{\varepsilon} = \{ \varepsilon Ax + c \mid A \in \mathbb{R}^{6 \times 6}, x, c \in \mathbb{R}^6, ||x|| \le 1 \},$$

wobei die Matrix  $A = U \operatorname{diag}(\sigma_1, \ldots, \sigma_6)U^t$  die Form des Ellipsoids beschreibt: Die Richtungen  $U = [u_1, \ldots, u_6]$  im Körper-KS werden entsprechend ihrer Bedeutung mit  $\sigma_i > 0$  gewichtet. Das Zentrum des Ellipsoids kann mittels c verschoben werden, um besonderes Gewicht auf eine bestimmte Bewegungsrichtung zu legen. Gesucht ist nun das (volumenmäßig) größte Ellipsoid, das noch in das zulässige wrench-Polytop passt:

$$Q = \max_{\varepsilon \in \mathbb{R}^+} \left\{ \varepsilon \, | \, \mathcal{E}_{\varepsilon} \subseteq W_{L_{\infty}} \right\}$$
$$= \max_{\varepsilon \in \mathbb{R}^+} \left\{ \varepsilon \, | \, \forall F \in \mathcal{E}_{\varepsilon} \, \exists \vec{f_c} \in FC : G\vec{f_c} = F \text{ und } \|\vec{f_c}\|_{\infty} \le 1 \right\}$$

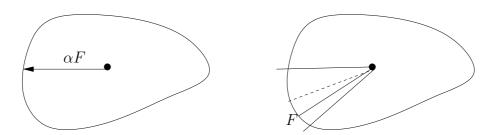
#### 2.8.2 Virtuelle Kontakte

Wir suchen die maximale Kraft  $\alpha F$  (max  $\alpha$ ), die wir in Richtung F ausüben können:

$$G\vec{f_c} = \alpha F$$
  $\Leftrightarrow$   $\left[ G \quad -F \right] \cdot \begin{bmatrix} \vec{f_c} \\ \alpha \end{bmatrix} = 0$   $\vec{f_c} \in FC \text{ und } \alpha > 0$ 

Wir fügen also -F als virtuellen Kontakt in die Greifmatrix G' = [G - F] ein und suchen nach Kontaktkräften  $[\vec{f_c} \ \alpha] \in \mathcal{N}(G') \cap int FC'$  mit  $FC' = FC \times \{\alpha \geq 0\}$ . Der virtuelle Kontakt wird also behandelt wie ein Punktkontakt ohne Reibung.

Bei gegebenem wrench-space W, wird also nach der größten Kraft, gesucht, die in eine bestimmte Richtung F ausgeübt werden kann.



Wenn wir für den virtuellen Kontakt auch einen "Haftreibungskegel" mit Öffnungswinkel  $\tan \alpha = \mu_v$  zulassen, suchen wir in einem ganzen Kegelausschnitt nach der größten ausübbaren Kraft F. Wir optimieren also auch die Kraftrichtung, in der die größte Kraft ausgeübt werden kann.

Ein extern wirkender wrench  $F_e$ , wie z.B. die Gravitation, kann ebenfalls als virtueller Kontakt berücksichtigt werden:

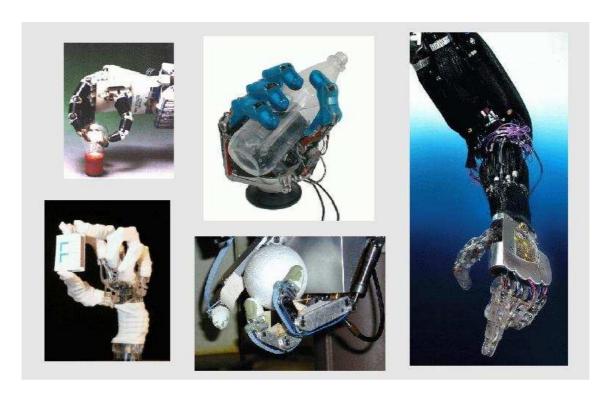
$$G\vec{f_c} = \alpha F - F_e \qquad \Leftrightarrow \qquad \begin{bmatrix} G & -F & F_e \end{bmatrix} \cdot \begin{bmatrix} \vec{f_c} \\ \alpha \\ 1 \end{bmatrix} = 0 \qquad \vec{f_c} \in FC \text{ und } \alpha > 0$$

Wir können also sowohl bevorzugte Kraftrichtungen als auch extern wirkende wrenches im selben framework beschreiben und lediglich einen Algorithmus zur Optimierung der internen Kräfte verwenden.

# 3 Gefügiges Greifen

# 4 Inverse Hand-Kinematik und Griffplanung

# 5 Antriebskonzepte



- Hirzinger Hand  $(4 \cdot 3 = 12 \text{ DOF})$ : (lokale) Elektromotoren
- Bretthauer Hand: Fluid-Aktorik
- Shadow Hand  $(5 \cdot 3 + 1 = 16 \text{ DOF})$ : pneumatische Muskeln und Sehnen

## 6 Lernen zu Greifen

Eine technisch motivierte Zerlegung des Greifprozesses:

- Identifikation des Objektes in der Szene
- Griffplanung in Abhängigkeit von Objekt, Umgebung (Hindernissen) und Aufgabenstellung
- Positionierung der Hand an dem Objekt
- Kontrolliertes Zugreifen, Herstellung von Fingerkontakten
- Lageregelung des Greifobjektes
- haptische Objektidentifikation und Griffbewertung
- Trajektorienplanung für Umgreifprozesse zur Manipulation
- Objekttransport
- Ablegen / Einpassen am Zielort
- Loslassen

### Äußerst komplexer Gesamtprozess!

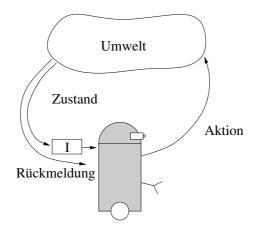
- Greifen ist zu komplex, um es vollständig zu programmieren.
- Lernen von Bewegungssequenzen (Greifen, Laufen, Fahrradfahren, Tennis) und Handlungsabfolgen (z.B. Montage) ist erforderlich.
- Klassische Verfahren überwachten Lernens (Neuronale Netze) sind nicht anwendbar, da Input-Output-Paare nicht verfügbar sind.
- Lernen muss durch selbständige *Exploration* und *Interaktion* mit der Umgebung oder durch Imitation erfolgen.
- Ein einfaches Bewertungssignal muss als Rückmeldung über Güte/Erfolg der Handlung ausreichen

Wir betrachten im folgenden die beiden Lernverfahren:

- Reinforcement Learning
- Imitation Learning

# 7 Reinforcement Learning (Verstärkungslernen)

Ein Reinforcement-Lerner ist ein Agent, der mit seiner Umwelt interagiert und durch trial-and-error versucht, seine Handlungsstrategie (Policy) zu verbessern, d.h. die erhaltene Rückmeldung (Belohnung) zu maximieren.



Das Modell enthält folgende Komponenten:

- eine diskrete, endliche Menge von Umgebungszuständen S, wobei der Agent eventuell nur eine Teilmenge I(s) wahrnehmen kann,
- $\bullet$  eine diskrete, endliche Menge von möglichen Aktionen  $\mathcal{A}$ ,
- eine Menge von Rückmeldungen, typischerweise  $\{0,1\}$  oder  $\mathbb{R}$ ,
- Regeln, die zu einem Zustand  $s_t$  und einer ausgeführten Handlung  $a_t$  zum Zeitpunkt t den nächsten Zustand  $s_{t+1}$  und die Rückmeldung  $r_t(s_t, a_t)$  bestimmen.

Die Zustandsübergänge werden von den Gesetzmäßigkeiten der Umwelt diktiert, während die Reward-Funktion  $r_t(s_t, a_t)$  die Lernaufgabe bestimmt.

Die Zustandsübergange können nicht-deterministisch sein, d.h. für ein Zustands-Aktions-Paar (s, a) kann es mehrere Folge-Zustände s' geben, deren Häufigkeit mit einer Wahrscheinlichkeitsverteilung beschrieben wird.

# 7.1 Markovsche Entscheidungsprozesse (MDP)

**Definition 7.1** Falls der Folgezustand nur vom aktuellen Zustand  $s_t$  und der ausgeführten Aktion  $a_t$  nicht aber von der gesamten Vergangenheit  $(s_t, a_t, s_{t-1}, a_{t-1}, \ldots, s_0, a_0)$  abhängt, spricht man von einem Markovschen Entscheidungsprozess (MDP). Es gibt dann eine Zustandsübergangsfunktion:  $\delta: \mathcal{S} \times \mathcal{A} \to \Pi(\mathcal{S})$ , wobei  $\Pi(\mathcal{S})$  die Menge von Wahrscheinlichkeitsverteilungen über der Zustandsmenge  $\mathcal{S}$  ist. Die Wahrscheinlichkeit im Zustand s bei Ausführung der Aktion a in den Zustand s' überzugehen, bezeichnen wir mit P(s'|s,a).

Die Strategie  $\pi: \mathcal{S} \to \mathcal{A}$ , mit der der Agent seine Aktionen auswählt, heißt *Policy*. Auch diese kann nicht-deterministisch sein. In diesem Falle gibt  $\pi(s, a)$  die Wahrscheinlichkeit an, im Zustand s die Aktion a auszuwählen.

#### Ablauf eines MDP

- 1. Bestimme aktuellen Zustand  $s_t$ .
- 2. Wähle eine Handlung  $a_t = \pi(s_t)$  und führe sie aus.
- 3. Erhalte Reward  $r_t(s_t, a_t)$ .
- 4. Umgebung geht als Reaktion auf  $a_t$  in neuen Zustand  $s_{t+1} = \delta(s_t, a_t)$  über.

Beispiel 7.2 Ein Agent soll lernen, Fahrrad zu fahren.

Zustände: Neigung des Fahrrads, Geschwindigkeit

Aktionen: Drehung des Lenkers nach rechts, links

Rewards: z.B. abhängig von der Neigung des Rades,

streng negativ (Bestrafung), falls das Rad umstürzt.

Der Agent soll lernen, den kumulierten Reward zu maximieren, d.h. möglichst lange aufrecht zu fahren.

#### 7.2 Definitionen des kumulierten Rewards

**Definition 7.3** Der kumulierte Reward oder auch value function gibt den zu erwartenden Reward an, wenn der Agent ausgehend vom Zustand  $s_t$  seine Policy  $\pi$  verfolgt:  $V^{\pi}: \mathcal{S} \to \mathbb{R}$ .

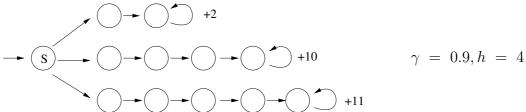
Unterschiedliche Definitionen von  $V^{\pi}$  beziehen zukünftige Rewards in unterschiedlicher Weise ein und beeinflussen so die optimale Strategie  $\pi^*$ :

discounted reward: 
$$V^{\pi} = E_{\pi} \left( \sum_{t=0}^{\infty} \gamma^{t} r_{t} \right)$$
 exponentiell abnehmende Gewards wichtung zukünftiger Rewards

finite-horizon reward: 
$$V^{\pi} = E_{\pi} \left( \sum_{t=0}^{h} r_{t} \right)$$
 gleiche Gewichtung zukünftiger Rewards, endlicher Zeithorizont

average reward: 
$$V^{\pi} = \lim_{h \to \infty} \frac{1}{h} E_{\pi} \left( \sum_{t=0}^{h} r_{t} \right)$$
 gleiche Gewichtung aller zukünftigen Rewards





Nur am Ende des Graphen gibt es einen positiven Reward.

finite-horizon reward: 
$$V^{\pi} = E_{\pi} \left( \sum_{t=0}^{4} r_{t} \right) = \{4; 0; 0\}$$

discounted reward: 
$$V^\pi = E_\pi \bigg( \sum_{t=0}^\infty \gamma^t \, r_t \bigg) = \gamma^n \sum_{t=0}^\infty \gamma^t R = \frac{R \, \gamma^n}{1-\gamma} \approx \big\{ 16; 66; 65 \big\}$$

average reward: 
$$V^{\pi} = \lim_{h \to \infty} \frac{1}{h} E_{\pi} \left( \sum_{t=0}^{h} r_{t} \right) = \{2; 10; 11\}$$

Je nach Wahl des Optimierungskriteriums  $V^{\pi}$  wählt der Agent eine andere Aktion, da er immer versucht den kumulierten Reward  $V^{\pi}$  zu maximieren.

Satz 7.5 Für eine gegebene Policy  $\pi$  muss die value-function  $V^{\pi}$  (discounted reward) die folgenden Gleichungen erfüllen:

$$V^{\pi}(s) = r(s, \pi(s)) + \gamma V^{\pi} \big( \delta(s, \pi(s)) \big) \qquad \forall s \in \mathcal{S}$$
bzw. 
$$V^{\pi}(s) = r(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s)) \cdot V^{\pi}(s') \qquad \forall s \in \mathcal{S}$$
 (7.1)

Der discounted reward im aktuellen Zustand s wird dabei ausgedrückt als Summe des rewards  $r(s, \pi(s))$  aufgrund der Aktion  $a = \pi(s)$  im aktuellen Zeitschritt und der zu erwartenden rewards  $V^{\pi}(s')$  im Folgezustand.

Beweis: Im Falle deterministischer Umwelt gilt:

$$V^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^{t} r_{t} = r_{0}(s, \pi(s)) + \gamma r_{1}(s', \pi(s')) + \gamma^{2} r_{2}(s'', \pi(s'')) + \dots$$

$$= r_{0}(s, \pi(s)) + \gamma \sum_{t=1}^{\infty} \gamma^{t-1} r_{t}$$

$$= r_{0}(s, \pi(s)) + \gamma V^{\pi} (\delta(s, \pi(s)))$$

П

Diese Gleichungen bilden ein lineares Gleichungssystem in  $V^{\pi}(s), s \in \mathcal{S}$ , dessen Lösung die value-function  $V^{\pi}$  liefert.

## 7.3 Policy Iteration

Zu gegebener value-function  $V^{\pi}$  kann man umgekehrt eine neue Policy bestimmen:

$$\pi'(s) := \arg\max_{a} \left( r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot V^{\pi}(s') \right). \tag{7.2}$$

Diese hat für jeden Zustand s eine höhere Bewertung  $V^{\pi'}(s)$ , ist also eine bessere Policy. Wir erhielten genau die alte Policy  $\pi' = \pi$  und die alte value-function  $V^{\pi}$ , wenn wir  $a = \pi(s)$  wählten. Da aber über alle möglichen Aktionen a maximiert wird, verbessert sich die Policy  $\pi'$  bzw. die zugehörige value-function  $V^{\pi'}$ . Wir erhalten damit den folgenden Algorithmus, um die optimale Policy  $\pi^*$  zu lernen:

```
Initialisiere die Policy \pi' zufällig loop \pi:=\pi' Berechne V^\pi durch Lösung von (7.1) Verbessere die Policy entsprechend (7.2) until \pi=\pi'
```

Für die optimale value-function  $V^*$  gelten die Bellman'schen Optimalitätsgleichungen:

$$V^{*}(s) = \max_{a} (r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) \cdot V^{*}(s'))$$

Die Berechnung der value-function  $V^{\pi}$  über das Gleichungssystem (7.1) ist u.U. sehr aufwendig. Das folgende iterative Verfahren ermöglicht eine Approximation von  $V^{\pi}$ :

```
Initialisiere k=0, V_k^\pi=0 loop V_{k+1}^\pi=r(s,\pi(s))+\gamma\sum_{s'\in\mathcal{S}}P(s'|s,\pi(s))\cdot V_k^\pi(s') until V_{k+1}^\pi\approx V_k^\pi
```

#### 7.4 Value Iteration

Alternativ kann man direkt die value-function optimieren:

Initialisiere 
$$V(s)=0$$
 loop 
$$\text{Berechne } Q(s,a) := r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) \cdot V(s')$$
 Update  $V(s) := \max_a Q(s,a)$  until Änderungen von  $V$  sind klein genug

Es kann gezeigt werden, dass dieser Algorithmus tatsächlich zur optimalen valuefunction konvergiert:  $V^* = V^{\pi^*}$ . Schließlich kann man die optimale Policy von der bestimmten optimalen value-function  $V^*$  ableiten:

$$\pi^*(s) := \arg\max_{a} \left( r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) \cdot V^*(s') \right). \tag{7.3}$$

Policy Iteration benötigt zwar weniger Iterationen, ist aber pro Iteration langsamer als Value Iteration. Es ist daher nicht klar, welcher Algorithmus der bessere ist, und es gibt Beispiele, für die der eine oder andere Algorithmus schneller konvergiert. Policy Iteration und Value Iteration sind Beispiele von dynamischer Programmierung. Sie benutzen die Zustandsübergangsfunktion  $\delta(s, a)$  und die Reward-Funktion r(s, a).

## 7.5 Q-Lernen

Policy und Value Iteration benutzen sowohl die Reward-Funktion r(s, a) als auch die Wahrscheinlichkeiten für Zustandsübergänge P(s'|s, a), um die optimale Policy zu finden. Diese Daten sind i.d.R. jedoch nicht ad hoc verfügbar und der Agent muss sie durch Exploration der Umwelt erst lernen. Das Q-Lernen umgeht dieses Modell-Wissen durch Nutzung der Q-Funktion (oder auch action-value function):

$$Q^{\pi}(s, a) := E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) = r(s, a) + \gamma V^{\pi}(\delta(s, a))$$
bzw. 
$$Q^{\pi}(s, a) := E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) = r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot V^{\pi}(s')$$

Sie gibt den erwarteten Reward an, wenn man in Zustand s die Aktion a auswählt und im weiteren der Policy  $\pi$  folgt. Die auszuführende Aktion a im aktuellen Zustand wird also von der eigentlich verfolgten Policy (und der damit verbundenen value-function  $V^{\pi}$ ) entkoppelt. Bei gegebener optimaler value-function  $V^*$  kann diese mittels (7.3) direkt zur Bestimmung der optimalen Policy  $\pi^*$  verwendet werden:

$$\pi^*(s) = \arg\max_{a} Q^*(s, a).$$

Mit der Beziehung  $V^*(s) = \max_a Q^*(s, a)$  folgen für die optimale Q-Funktion  $Q^*$  die Bellman'schen Optimalitätsgleichungen:

$$Q^*(s, a) = r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot \underbrace{\max_{a'} Q^*(s', a')}_{V^*(s')}$$

Diese rekursive Definition von  $Q^*$  bildet die Basis für den Q-Lernalgorithmus:

$$Q'(s,a) := r + \gamma \max_{a'} Q(s',a'), \qquad (7.4)$$

wobei  $\langle s, a, r, s' \rangle$  die *Erfahrung* angibt, im Zustand s bei ausgeführter Aktion a im Zustand s' zu landen und den Reward r erhalten zu haben.

**Temporal Difference Learning:** TD(0) Ein Updateschritt kann auch nur in Richtung des neuen Wertes Q'(s, a) stattfinden:

$$Q(s,a) := Q(s,a) + \alpha \left(\underbrace{r + \gamma \max_{a'} Q(s',a')}_{Q'(s,a)} - Q(s,a)\right), \tag{7.5}$$

Die Lernschrittweite  $\alpha \in [0,1]$  bestimmt, wie groß der Lernschritt in Richtung des upgedateten Q-Wertes Q'(s,a) sein soll. Sie gewichtet den Wert des alten und des neuen Wertes der Q-Funktion und erlaubt einen fließenden Übergang zur neuen Funktion. Für  $\alpha = 1$  erhält man wieder die obige Q-Lernregel.

Damit lautet der Algorithmus:

Initialisiere  $Q\equiv 0$  loop

Wähle Handlung  $a=\arg\max_a Q(s,a)$  und führe sie aus Update (7.5) mit erhaltenem Reward und neuem Zustand s' until Änderungen von Q sind klein genug

**Satz 7.6** Unter folgenden Voraussetzungen konvergiert Q gegen die optimale Q-Funktion  $Q^*$ :

- $|r(s,a)| < \infty \forall s, a,$
- $0 \le \gamma < 1$ ,
- Jedes (s, a)-Paar wird unendlich oft besucht.

## 7.6 Exploration vs. Exploitation

Exploration: meist zufällige Erforschung des Zustandsraumes Exploitation: optimale Ausnutzung des bisher gewonnenen Wissens (Approximation von V oder Q) zur Berechnung der Policy.

- Zu ausgiebiges Erforschen bedeutet, dass der Agent auch nach langem Lernen noch ziellos im meist sehr großen Zustandsraum umherwandert. Dadurch werden auch Bereiche intensiv untersucht, die für die Lösung der Aufgabe gar nicht relevant sind.
- Zu frühes Ausbeuten der gelernten Approximation der Q-Funktion bewirkt möglicherweise, dass sich ein suboptimaler Pfad durch den Zustandsraum etabliert, weil dieser zufällig zuerst gefunden wurde und gute Ergebnisse lieferte. Die optimale Lösung wird jedoch nicht mehr gesucht.

Beispiel 7.7 Mehrere einarmige Banditen, deren Ausschüttung und Erfolgsquote zufällig sind. Wie lange soll man bei anderen Automaten nach höheren Ausschüttungsquoten suchen und ab wann soll man sich auf einen Automaten konzentrieren?

Lösung: teilweise zufällige Auswahl der Aktionen.

 $\varepsilon$ -greedy: Wähle typischerweise die optimale Aktion  $a^*$  aus, aber mit

Wahrscheinlichkeit  $\frac{\varepsilon}{|A|}$  eine zufällige andere Aktion a.

Boltzmann-Gewichtung: Wähle die Aktionen a zufällig entsprechend der Boltzmann-

Verteilung aus:

$$\pi(s, a) = \frac{e^{\beta(Q(s, a) - Q(s, a^*))}}{\sum_{a'} e^{\beta(Q(s, a') - Q(s, a^*))}} \quad \text{mit } a^* = \arg\max_{a} Q(s, a)$$

Die Lernparameter  $\varepsilon \in [0,1]$  und  $\beta \in [0,\infty)$  sollten langsam abnehmen.

- $\varepsilon = 1$  bzw.  $\beta = 0 \Rightarrow$  Gleichverteilung
- $\varepsilon = 0$  bzw.  $\beta = \infty \Rightarrow$  greedy

## 7.7 Temporal Difference Learning: $TD(\lambda)$

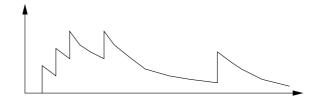
Die Lernregel TD(0) berücksichtigt nur eine einzige Erfahrung  $(s_t, a_t, r_t)$  des Agenten:

$$Q(s,a) := Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right).$$

Damit finden Updates von Q nur für  $(s_t, a_t)$ -Paare statt, die direkt mit einem Reward  $r_t$  verknüpft sind. Falls der Reward stark verzögert kommt, baut sich die richtige Q-Funktion nur sehr langsam auf. Sinnvoller ist es daher bei einem Update auch vorherige Schritte – also den gesamten Weg zum Ziel – zu berücksichtigen. Dazu führen wir die Eignung (eligibility) eines (s, a)-Paares ein:

$$e_t(s, a) = \begin{cases} \gamma \lambda e_{t-1}(s, a) + 1 & \text{falls } (s, a) = (s_t, a_t) \\ \gamma \lambda e_{t-1}(s, a) & \text{sonst} \end{cases}$$

Falls ein Paar (s, a) im Zeitschritt t besucht wird, erhöht sich die Eignung um 1, sonst nimmt sie exponentiell ab:

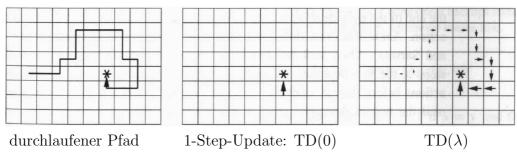


Je höher der Wert e(s, a), desto öfter wurde das Paar besucht und desto wichtiger ist offenbar sein Anteil am erzielten Erfolg (Reward). Dementsprechend nutzen wir folgende Updateregel:

$$Q(s,a) := Q(s,a) + \alpha \cdot e(s,a) \cdot \Delta(s,a,s') \qquad \text{für alle } (s,a) \in \mathcal{S} \times \mathcal{A}$$
 
$$\Delta(s,a,s') = r + \gamma \max_{a'} Q(s',a') - Q(s,a)$$

Die alte Lernregel erhalten wir mit  $e(s, a) = \delta_{sst}\delta_{aa_t}$ , also für  $\lambda = 0$ .

## Beispiel 7.8



# 7.8 Generalisierung:

#### Kontinuierliche Zustands- und Aktionsräume

Alle Algorithmen sind von diskreten Zustands- und Aktionsräumen ausgegangen und speichern die Funktionen V(s) und Q(s,a) in einer Tabelle. Dies ist sehr ineffizient und im Falle von kontinuierlichen Zustands- und Aktionsräumen nicht mehr praktikabel.

**Lösung:** Funktionsapproximation für V(s) bzw. Q(s, a). allg. Ansatz eines Approximators:  $\hat{Q}_w(s, a)$ .

Die Gewichte w haben je nach Wahl des Approximators unterschiedliche Bedeutung:

- Splines/Polynome: Stützstellen
- Multilagen-Perzeptron (MLP): Gewichte im Netz
- SOM, LLM, PSOM: Stützstellen

Ein guter Funktionsapproximator sollte folgende Eigenschaften haben:

- Gute Generalisierung, d.h. Interpolation und Extrapolation bei unbekannten Zuständen und Aktionen.
- Gute Approximation der tatsächlichen Q-Funktion. Dazu muss die Auflösung der Q-Funktion genügend hoch sein, kann aber in unterschiedlichen Bereichen des Raumes  $\mathcal{S} \times \mathcal{A}$  variieren.
- Geringer Speicherbedarf

- Effiziente Berechnung der optimalen Aktion:  $\arg \max_{a \in \mathcal{A}} \hat{Q}_w(s, a)$ . Bei kontinuierlichem Aktionsraum  $\mathcal{A}$  stellt dies ein nichtlineares Optimierungsproblem dar, mit all seinen Problemen (lokale Minima = suboptimale Aktionsauswahl)
- Effiziente Updates der Q-Funktion.
- Lokale Lernverfahren.

Einige dieser Eigenschaften widersprechen einander und man muss einen geeigneten Mittelweg finden, der alle Eigenschaften ausreichend berücksichtigt.

## 7.9 Self-Organizing Map (SOM) / Neural Gas

Die Tabellendarstellung von Q(s,a) weist kontinuierlichen Zustands-Aktions-Paaren zunächst diskrete Werte  $(\bar{s},\bar{a})$  zu und diesen dann den Q-Wert  $Q(\bar{s},\bar{a})$ . Die Q-Funktion ist damit als stückweise konstante Funktion (auf den Zellen des Gitters) dargestellt. SOMs stellen eine Alternative zur Vektorquantisierung (VQ) dar, um kontinuierliche Paare (s,a) auf diskrete  $(\bar{s},\bar{a})$  abzubilden.

Wir wollen die SOM im folgenden lediglich zur Quantisierung der Zustände  $s \in \mathcal{S}$  benutzen – die Aktionen selbst seien weiterhin diskret.

Die allgemeine SOM besteht aus folgenden Komponenten:

- Diskretes Gitter von Knoten oder Neuronen:  $r \in \mathbb{R}^m$ .
- Jedem Knoten r ist ein Gewichtsvektor  $\vec{w}_r \in \mathbb{R}^I \triangleq \mathcal{S}$  im Input-Raum (Zustandsraum) zugeordnet.
- Zu gegebenem Input  $\vec{x} = s$  bestimme den Ausgabe-Knoten (winner):  $r^* = \arg\min_{r \in \mathbb{R}^m} ||\vec{x} \vec{w_r}||.$
- SOM-Lernregel:  $\Delta \vec{w_r} = \varepsilon \cdot h(r, r^*) \cdot (\vec{x} \vec{w_r})$
- Nachbarschaftsgewichtung mit Gaußglocke:  $h(r, r^*) = exp(-\frac{\|r-r^*\|}{2\sigma^2})$ .
- Nachbarschaftsbeziehungen auf dem abstrakten Gitter werden auf den Input-Raum übertragen.
- Das abstrakte Gitter kann man sich als Koordinatensystem für die m-dimensionale Hyperfläche im n-dimensionalen Input-Raum vorstellen.
- Anstatt die Nachbarschaftsgewichtung auf Gitter-Abständen  $||r r^*||$  zu berechnen, kann man auch direkt die Abstände im Input-Raum verwenden:  $||\vec{w_r} \vec{w_{r^*}}||$ . Das abstrakte Gitter ist dann überflüssig und das resultierende Modell nennt man Neural Gas.
- SOM und Neural Gas sind Vektorquantisierer und können daher als Ersatz für die Q-Tabellen-Quantisierung eingesetzt werden.

Voronoi-Parzellierung Die Voronoi-Parzellierung gibt zu gegebenen Quantisierungsvektoren die zugehörigen Regionen im Input-Raum an, die auf diese Vektoren abgebildet werden (aufgrund des kleinsten Abstandes zu diesen Vektoren).

Nach der Quantisierung mit der SOM können alle tabellenbasierten RL-Algorithmen ohne Änderungen angewandt werden. Der SOM-Algorithmus garantiert eine variierende Dichte der Quantisierungsvektoren im Zustandsraum – proportional zur Häufigkeit mit der die entsprechenden Regionen besucht werden.

Die Nachbarschaftsbeziehungen der SOM können nun auch genutzt werden, um kontinuierlich zwischen den diskreten Q-Werten  $Q(\bar{s}, a)$  zu interpolieren. Dazu benutzen wir neben dem winner-Vektor  $\bar{s}^*$  auch alle benachbarten Quantisierungsvektoren  $\bar{s}_i \in \mathcal{N}$ :

$$\hat{Q}(s,a) = \sum_{\bar{s} \in \mathcal{N}} \frac{\|\bar{s} - s\|}{\sum_{\bar{s} \in \mathcal{N}} \|\bar{s} - s\|} Q(\bar{s},a) = \sum_{\bar{s} \in \mathcal{N}} w_{\bar{s}} Q(\bar{s}_i,a) \qquad \sum_{\bar{s} \in \mathcal{N}} w_{\bar{s}} = 1$$

Der Interpolationswert  $\hat{Q}(s, a)$  entsteht also durch Mittelung der Q-Tabellenwerte  $Q(\bar{s}_i, a)$  über alle benachbarten Vektoren  $\bar{s}_i \in \mathcal{N}$ , wobei die Q-Werte entsprechend dem Abstand des Vektors  $\bar{s}_i$  zum tatsächlichen Zustandsvektor s gewichtet werden.

Die  $TD(\lambda)$ -Update-Regel erhält folgende Form:

$$Q(\bar{s}, a) := Q(\bar{s}, a) + \alpha \left(\underbrace{r + \max_{a'} \ \gamma \, \hat{Q}(s', a') - \hat{Q}(s, a)}_{\Delta(s, a, s')}\right) e(\bar{s}, a)$$

Den Fehler  $\Delta(s,a,s')$  berechnen wir also mittels der  $\hat{Q}$ -Approximationen für die tatsächlich beobachteten kontinuierlichen Zustände s und s'. Die Eignung  $e_t(\bar{s},a)$  wird nur noch entsprechend dem jeweiligen Anteil  $w_{\bar{s}}$  der Quantisierungsvektoren  $\bar{s}$  zum Gesamtwert von Q erhöht:

$$e_t(\bar{s}, a) = \begin{cases} \gamma \lambda e_{t-1}(\bar{s}, a) + w_{\bar{s}} & \text{falls } \bar{s} \in \mathcal{N} \text{ und } a = a_t \\ \gamma \lambda e_{t-1}(\bar{s}, a) & \text{sonst} \end{cases}$$

#### 7.10 Verbleibende Probleme

Unvollständige Zustandsinformation erfordert internen Zustand des Agenten, der Historie der Zustände und Aktionen kodiert. Damit ist es möglich Mehrdeutigkeiten in der Zustandsinformation aufzulösen.

Beispiel: Labyrinth. Der Agent sieht aber nur den aktuellen Gang und kann nicht zwischen ähnlichen Gängen unterscheiden.

Schlechte Skalierung mit der Größe (Dimensionalität) des Lernproblems. Zur Lösung komplexer Probleme muss Vorwissen in die Strukturierung der Lernaufgabe eingehen (bias):

Shaping: Präsentation von zunächst einfachen Lernaufgaben, um Basis-Fähigkeiten

zu Erlernen. Übergang zu immer komplexeren Aufgaben. Beispiel: La-

byrinthe.

Lokale Rewards: Rewards, die in zeitlicher Nähe zur verursachenden Handlung ste-

hen, erhöhen die Lerngeschwindigkeit. Schlecht: Rewards erst bei Er-

folg/Misserfolg einer ganzen Handlungssequenz.

Imitation: Einschränkung des Suchraumes durch Nachahmen einer vorgeführten

Handlung.

Reflexe: Eine vorprogrammierte Heuristik kann ebenfalls helfen, den Suchraum

einzuschränken. Sie stellt eine sinnvolle Start-Policy dar, die durch

Erfahrung aber auch verändert werden kann.

Hierarchiches Lernen: Zerlegung des Problems in eine Hierarchie von Teilproblemen,

die separat gelernt werden.

# 8 Imitation Learning

Imitation: Nachahmung einer beobachteten Bewegungssequenz

Das Gebiet ist relativ neu und bislang existieren kaum Anwendungen. Wir zeigen im folgenden ein paar Probleme und Fragestellungen des Imitation Learning auf.

- 1. Was soll nachgeahmt werden?
- 2. Wie wird eine Status- bzw. Sensorsequenz in sinnvolle Teilstücke zerlegt?
- 3. Wie müssen die beobachteten Sequenzen ausgeführt werden, d.h. welche Aktionen führen zur beobachteten Bewegungssequenz?

Wenn Armbewegungen beobachtet werden, sieht man i.d.R. nur den Effekt der Gelenkwinkeländerungen. Welche Gelenkwinkeländerungen notwendig sind, um diesen Effekt zu erzielen, muss der Lerner selbst explorieren.

# 8.1 Was soll nachgeahmt werden?

Handlungsequenzen kann man sich auf verschiedenen Abstraktionsebenen vorstellen. Beispiel Greifen:

- Gelenkwinkel: Welche Arm- und Fingerbewegungen hat der Instruktor gemacht?
- Griffpunkte: Wo hat der Instruktor das Objekt gegriffen?

- Griffpostur: Welchen Grifftyp (Pinzettengriff, Kraftgriff) hat der Instruktor ausgewählt, um das Objekt zu greifen und an welcher Stelle des Objekts greift er zu?
- Was hat der Instruktor gegriffen?

Am Beispiel der Manipulation, z.B. Zusammenbau eines Baufix-Flugzeugs, wird klar, dass man sich typischerweise einen hierarchischen Aufbau vorstellen kann.

- Gelenkwinkel: Welche Handbewegungen sind notwendig, um zwei Teile aufeinander zu legen, eine Schraubverbindung herzustellen, etc.
- In welcher Reihenfolge müssen welche Teile miteinander verbunden werden?
- Kommt es auf die Farbe oder Länge der Objekte an oder sind diese Eigenschaften irrelevant?

Dies motiviert die hierarchische Aufteilung in Handlungsprimitiven. (behavioural primitives).

Beispiel 8.1 Kinder sollen Armbewegungen nachmachen, bei denen

- ein Fleck auf dem Tisch durch die Hand bedeckt wird,
- kein Fleck auf dem Tisch existiert, aber dieselbe Armbewegung ausgeführt wurde.

Falls ein Fleck zu sehen ist, benutzen die Kinder beide Arme mit gleicher Wahrscheinlichkeit, um den Fleck zu verdecken. Ohne Fleck, benutzen die Kinder immer denselben Arm wie der Instruktor. Das Ziel der Handlung wurde als top-down festgelegt.

Auf welche Eigenschaften der vorgeführten Handlungssequenz(en) es ankommt, kann z.B. mittels einer Invarianz-Analyse festgestellt werden: Welche Teile der Handlungssequenz wiederholen sich immer wieder (scheinen also relevant für die Aufgabe zu sein), welche variieren (und sind daher irrelevant)?

# 8.2 Wie soll eine Handlungssequenz in sinnvolle Teilstücke (primitives) zerlegt werden?

- Welche Handlungsprimitiven sind überhaupt sinnvoll?
- Welche Primitiva können in hierarchischer Weise zusammengefasst werden?
- Wie können die Primitiva in einer Handlungssequenz detektiert werden? Lösung: Ansätze aus der Spracherkennung

## 8.3 Imitation auf der Gelenkwinkel-Ebene

Auf unterster Hierarchie-Ebene müssen i.d.R. Gelenkwinkel-Trajektorien reproduziert werden. Es hat sich gezeigt, dass eine direkte Übertragung der beobachteten Gelenkwinkel beim Menschen auf Roboter-Gelenkwinkel (selbst humanoide Roboter) nicht möglich ist, da beide immer eine (leicht) unterschiedliche Kinematik haben – insbesondere variieren Segmentlängen und die Anordnung der Gelenke zueinander.

## Literatur

- [AMOL97] P.K. Allen, A.T. Miller, P.Y. Oh, and B.S. Leibowitz. Using tactile and visual sensing with a robotic hand. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 676–681. 1997.
- [BBD<sup>+</sup>99] T. Bergener, C. Bruckhoff, P. Dahm, H. Janßen, F. Joublin, R. Menzner, A. Steinhage, and W. von Seelen. Complex behavior by means of dynamical systems for an anthropomorphic robot. *Neural Networks*, 12(7-8):1087–1099, 1999.
  - [Bic94] A. Bicchi. On the problem of decomposing grasp and manipulation forces in multiple whole-limp manipulation. *Jorunal of Robotics and Autonomous Systems*, 13:127–147, 1994.
  - [Bic95] A. Bicchi. On the closure properties of robotic grasping. *International journal of Robotics Research*, 14(4), 1995.
  - [Bic00] A. Bicchi. Hands for dextrous manipulation and robust grasping: a difficult road towards simplicity. *IEEE Trans. on Robotics and Automation*, 16(6):652–662, 2000.
  - [BK00] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 348–353. 2000.
- [BKM<sup>+</sup>99] M. Becker, E. Kefalea, E. Maël, C. von der Malsburg, M. Pagel, J. Triesch, J.C. Vorbrüggen, R.P. Würtz, and S. Zadel. Gripsee: A gesture-controlled robot for object perception and manipulation. *Autonomous Robots*, 6(2):203–221, 1999.
  - [CH90] M.R. Cutkosky and R.D. Howe. Human grasp choice and robotic grasp analysis. In *Dextrous Ronot Hands*, pages 5–31. Springer, 1990.
  - [CH97] R. Cipolla and N. Hollinghurst. Visually guided grasping in unstructured environments. Journal of Robotics and Autonomous Systems, 19(3-4), 1997.
  - [CPR01] Jr. J.A. Coelho, J.H. Piater, and Grupen R.A. Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot. *Robotics and Autonomous Systems Journal*, 37(2-3):195–219, 2001.
    - [Erd97] M. Erdmann. An exploration of nonprehensile two-palm manipulation. International Journal of Robotics Research, pages 1–23, 1997.
    - [GC02] R.A. Grupen and Jr. J.A. Coelho. Acquiring state from control dynamics to learn grasping policies for robot hands. *International Journal on Advanced Robotics*, 16(5):427–444, 2002.

- [Gol00] K. Goldberg. Symposium on part feeding and fixturing. In *IEEE International Conference on Robotics and Automation*. 2000.
- [HK96] W.S. Howard and V. Kumar. On the stability of grasped objects. *IEEE Trans. Robotics and Automation*, 12(6):904–917, 1996.
- [Ibe87] T. Iberall. The nature of human prehension. In *Proc. IEEE Int. Conf.* on Robotics and Automation. 1987.
- [Jam85] J.W. Jameson. Analytic techniques for automated grasps. PhD thesis, Stanford University, Department of Electrical Engineering, 1985.
- [KC92] I. Kao and M.R. Cutkosky. Quasistatic manipulation with compliance and sliding. *Journal of Robotics Research*, 11(1):20–40, 1992.
- [Lev97] S.R. Leveroni. *Grasp gaits for planar object manipulation*. PhD thesis, Massachusetts Institute of Technology, 1997.
- [Mat97] Y. Matsuoka. The mechanisms in a humanoid robot hand. *Autonomous Robots*, 4(2):1–11, 1997.
- [MLS94] M. R. Murry, Z. Li, and S. S. Sastry. A Mathematical Introduction to Robotic Manipulation. CRC Press, 1994.
- [MO90] K. Mirza and D.E. Orin. Force distribution for power grasp in the Dgigits system. In CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators. 1990.
- [MP89] X. Markenscoff and C.H. Papadimitriou. Optimum grip of a polygon. *International Journal of Robotics Research*, 8(2):17–29, 1989.
- [MT01] Yoky Matsuoka and W.T. Townsend. Design of life-size haptic environments. In Rus and Singh, editors, *Experimental Robotics VII*, pages 461 470. 2001.
- [OMC97] A.M. Okamura, Turner M.L., and M.R. Cutkosky. Haptic exploration of objects with rolling and sliding. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, volume 3, pages 2485–2490. 1997.
- [OSC00] A.M. Okamura, N. Smaby, and M.R. Cutkosky. An overview of dexterous manipulation. 2000.
- [Pau01] J. Pauli. Learning-based Robot Vision, volume 2048 of Lecture Notes in Computer Science. Springer, 2001.
- [Sal82] J.K. Salisbury. Kinematics and Force Analysis of Articluated Hands. PhD thesis, Stanford University, 1982.
- [SB98] R.S. Sutton and A.G. Barto. Reinforcement learning: An introduction. MIT Press, 1998.

- [Sch99] S. Schaal. Is imitataion learning the route to humanoid robots? *Trends in Cognitive Science*, 3:233–242, 1999.
- [SPB01] S. Schulz, C. Pylatiuk, and G. Bretthauer. A new ultralight anthropomorphic hand. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*. 2001.
- [TAP87] J.C. Trinkle, J.M. Abel, and R.P. Paul. An investigation of frictionless enveloping grasping in the plane. *Int.J. Robotics Research*, 7(3):33–51, 1987.
  - [Tri92] J.C. Trinkle. On the stability and instantaneous velocity of grasped frictionless objects. 8(5), 1992.
  - [Yos96] T. Yoshikawa. Passive and active closures by constraining mechanisms. In *Proceedings IEEE Int. Conf. on Robotics and Automation*, pages 1477–1484. 1996.