

Neuronale Karten in der Robotik: Lernen visuell geführter Greifbewegungen

Thomas Martinetz
Siemens AG
Zentrale Forschung und Entwicklung
81730 München

Wir betrachten das Problem des Erlernens von Greifbewegungen und präsentieren einen Lösungsansatz basierend auf einer hierarchischen Anordnung Kohonen'scher Merkmalskarten. Die Steuerung der Greifbewegungen erfolgt allein über Stereokameras. Die Netzwerkarchitektur zur effizienten Repräsentation der möglichen Objektorientierungen an den verschiedenen Orten des Arbeitsbereiches des Roboters besteht aus einem Satz zweidimensionaler Kohonennetze, angeordnet in einem dreidimensionalen Gitter. Jeder Knoten der Kohonennetze speichert eine lineare Abbildung zur Generierung passender Gelenkwinkelstellungen. Diese linearen Abbildungen ermöglichen eine auf visueller Rückkopplung basierende Greifstrategie.

1. Einleitung

Nachbarschaftserhaltende Merkmalskarten, gekoppelt mit linearen Abbildungen zu sogenannten *Local Linear Maps* (LLM) (Ritter, Schulten 1986), bieten interessante Lösungsansätze für Problemstellungen aus Robotik und Prozeßsteuerung (Ritter et al. 1991, Martinetz 1992). Der sich ergebende neuronale Netzwerktyp zeichnet sich aus durch eine hohe Lernrate sowie stabile Konvergenz, und die explizite Verwendung linearer Abbildungen macht diesen Ansatz insbesondere auch für Aufgabenstellungen geeignet, deren Lösung Rückkopplung erfordert. Wir wollen in diesem Artikel eine hierarchische Variante dieses Verfahrens auf das Problem des Greifens von Objekten mit einem Roboter anwenden. Das Problem der behutsamen Annäherung und kollisionsfreien Umschließung des Objektes mit dem Greifer wollen wir mithilfe visueller Rückkopplung über Stereokameras lösen. Dabei beschränken wir uns auf einen Teil der Fragestellungen bei der Planung von Greifbewegungen, indem wir ein zu greifendes Objekt von relativ einfacher Gestalt, nämlich einen Zylinder, betrachten. Wegen der Symmetrie des Objekts bestimmt dann neben der Position im Arbeitsbereich nur noch die Orientierung einer einzigen Achse, der Symmetrieachse, mögliche Bewegungstrajektorien sowie die von Arm und Greifer einzunehmende Endstellung.

2. Das Robotermodell

In Abb. 1 ist das in den folgenden Computersimulationen verwendete Robotermodell dargestellt. Der Arm des Roboters besitzt drei Freiheitsgrade, womit die erforderliche Armstellung

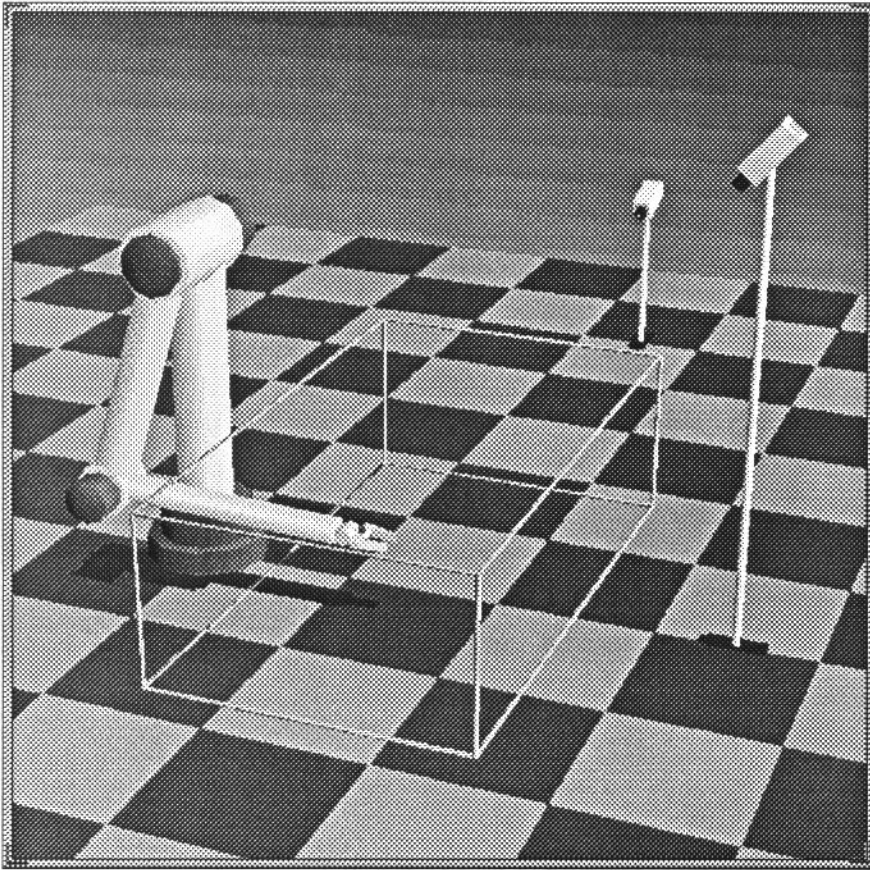


Abb.1: Das im Computer simulierte Robotermodell. In der Computergraphik ist außerdem der Arbeitsbereich sowie die zwei zur Steuerung der Greifbewegungen verwendeten Kameras eingezeichnet.

zu jeder vorgegebenen Objektposition eindeutig bestimmt ist (es liegt keine Redundanz vor). Der Arm kann sich um die vertikale Achse drehen (θ_1), und die Achsen des mittleren (θ_2) sowie äußeren (θ_3) Gelenks stehen parallel zueinander und parallel zur Horizontal-Ebene, die von der x- und y-Achse aufgespannt wird. Zur Orientierung des Greifers stehen zwei Freiheitsgrade zur Verfügung. Der erste Freiheitsgrad ist durch die Achse am "Handgelenk" gegeben, die parallel zum mittleren und äußeren Gelenk des Arms steht. Der zweite Freiheitsgrad erlaubt Drehungen des Greifers um seine eigene Symmetrieachse.

Abbildung 2 liefert eine Skizze des Greifers mit seinen zwei Gelenkwinkeln β_1 und β_2 sowie dem die Orientierung des Greifers beschreibenden Normalenvektor \mathbf{n} . Dieser Vektor steht senkrecht zur Symmetrieachse und senkrecht zur flachen Seite des Greifers. Der eingezeichnete Punkt P bezeichnet das Zentrum des Greifers und muß während des Greifvorgangs ins Zentrum des Zylinders geführt werden. Gleichzeitig ist es erforderlich, den Normalenvektor \mathbf{n} parallel zur Achse des Zylinders auszurichten. Damit dies in jedem Fall möglich ist, muß der Normalenvektor jede beliebige Orientierung einnehmen können. Dies wird durch die beiden Gelenkwinkel β_1 und β_2 gewährleistet.

3. Steuerung über Kameras

Die Aufgabe des neuronalen Netzwerks besteht darin, von zwei Kameras gelieferte Eingangssignale in passende Gelenkwinkel für den Arm sowie den Greifer umzusetzen. Wir gehen im folgenden davon aus, daß ein den Kameras nachgeschaltetes Bildverarbeitungssystem aus

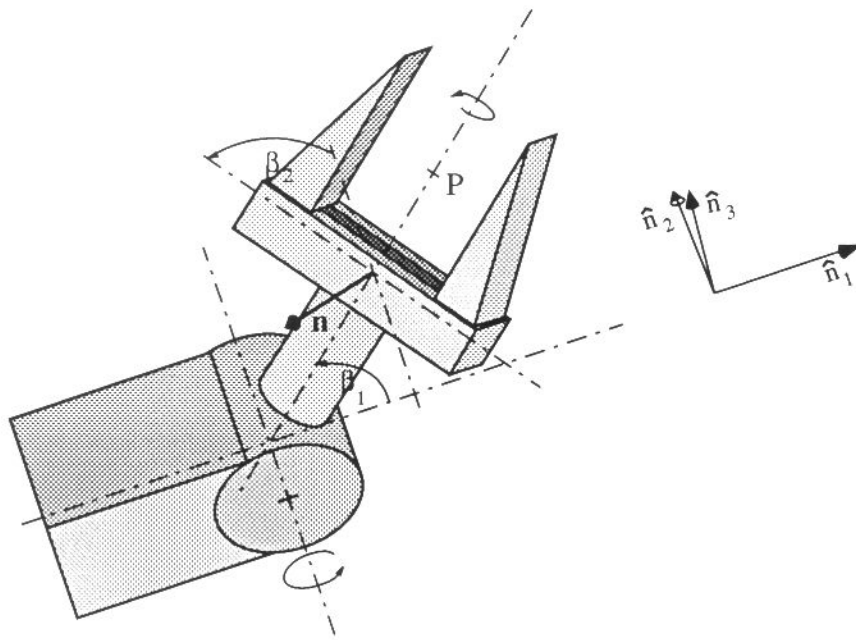


Abb.2: Eine Skizze des Greifers zusammen mit seinen beiden Gelenkwinkeln β_1 und β_2 sowie dem Normalenvektor \mathbf{n} , der die Orientierung des Greifers beschreibt. Der Punkt P bezeichnet das Zentrum und damit den Ort des Greifers, der ins Zentrum des zu greifenden Zylinders geführt werden muß.

den Kamerabildern die erforderlichen Bildkoordinaten des anzufahrenden Objekts extrahieren kann. Um einen Zylinder greifen zu können, benötigt das neuronale Netz allerdings nicht nur Information über die Position des Zylinders im Arbeitsbereich, sondern zusätzlich auch über dessen Orientierung im Raum. An jeder Position im Arbeitsbereich besitzt der Zylinder zwei Freiheitsgrade, die die Orientierung der Zylinderachse und damit auch die zum Greifen notwendige Gelenkwinkelkonfiguration von Arm und Greifer mitbestimmen.

In welcher Form liefern uns die Kameras die Information über Position und Orientierung des Zylinders? In Abb. 3 ist schematisch der Balken dargestellt, den jede Kamera als Projektion des Zylinders auf ihre Bildebene sieht. Der Ort beider Balken auf den Bildebenen enthält implizit die Information über die Raumposition des Zylinders. Jeder Balken besitzt zusätzlich noch eine Orientierung in der Bildebene. Diese Orientierungen liefern implizit die Information über die räumliche Orientierung der Zylinderachse.

Wie aus Abb. 3 ersichtlich, beschreiben wir den Ort des Zentrums jedes Balkens durch dessen zweidimensionale Koordinaten auf der Bildebene der jeweiligen Kamera und fassen beide Koordinaten zu einem vierdimensionalen Vektor \mathbf{u} zusammen. Zur Beschreibung der Orientierung jedes Balkens benutzen wir die Projektionen auf die x- und y-Achse der jeweiligen Kamerabildebene. Um diese Projektionen bezüglich der Vorzeichen eindeutig festzulegen, ist es notwendig, jedem Balken in der Bildebene eine Richtung zuzuordnen, d.h., ein Ende jedes Balkens auszuzeichnen. In Abb. 3 ist das jeweils ausgezeichnete Ende mit einem Pfeilkopf markiert. Welches der beiden Balkenenden jeweils die Richtung angeben soll, ist beliebig. Die ausgezeichneten Enden müssen allerdings zueinander korrespondieren, also zum selben Ende des Zylinders gehören.

Durch die Projektionen ergeben sich so zwei zweidimensionale Vektoren (x_{x1}, x_{y1}) und (x_{x2}, x_{y2}) , die beide Balken mitsamt der gewählten Richtungen eindeutig beschreiben. Zusammengefaßt erhalten wir einen vierdimensionalen Vektor \mathbf{x} , welchen wir noch normieren, um

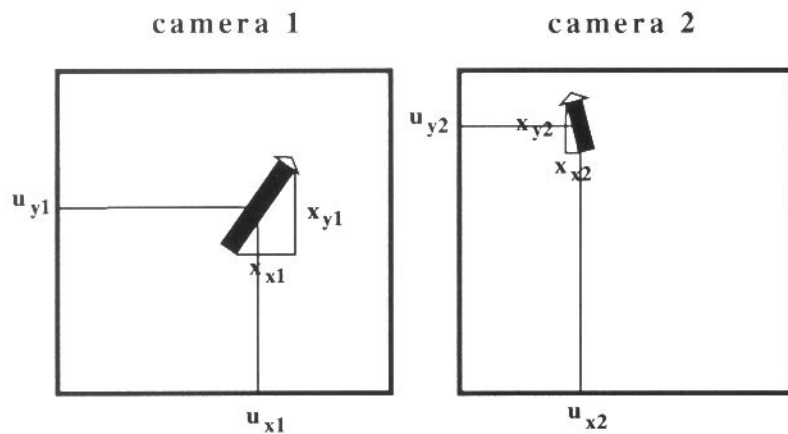


Abb.3: Der Zylinder, gesehen von Kamera 1 und Kamera 2. Mit einem Pfeilkopf ist das ausgezeichnete Ende jedes Balkens markiert (siehe Abschnitt 3).

die hier nicht benötigte Information über die Länge des Zylinders zu beseitigen. Die beiden jeweils vierdimensionalen Vektoren \mathbf{u} und \mathbf{x} tragen nun zusammen die gesamte Information, die das neuronale Netzwerk zur Ausgabe passender Gelenkwinkelstellungen benötigt.

4. Hierarchische Anordnung Kohonen'scher Merkmalskarten

Zur Repräsentation der die Raumposition und Orientierung des Zylinders beschreibenden Eingabesignale \mathbf{u} und \mathbf{x} benutzen wir ein Netzwerk, welches aus einer hierarchischen Anordnung vieler einzelner Kohonen'scher Merkmalskarten besteht (Kohonen 1989). Wie in Abb. 4 dargestellt, besteht die verwendete Netzwerkarchitektur aus einem Satz zweidimensionaler *Unternetze*, die in einem dreidimensionalen *Übernetz* angeordnet sind.

Wegen der drei impliziten Freiheitsgrade der die Raumposition des Objektes bestimmenden Eingabesignale \mathbf{u} wählen wir für deren Repräsentation entsprechend ein dreidimensionales Kohonennetz, welches allerdings aus zweidimensionalen Unternetzen besteht. Jedes Element des dreidimensionalen Übernetzes, also jedes Unternetz, spezialisiert sich während des Lernvorgangs auf einen kleinen Ausschnitt des Arbeitsbereichs. Innerhalb des jeweiligen Ausschnittes bildet das dortige zweidimensionale Unternetz eine nachbarschaftserhaltende Repräsentation der durch zwei implizite Freiheitsgrade festgelegten, die Orientierung des Zylinders bestimmenden Eingabesignale \mathbf{x} .

Jedem Unternetzes \mathbf{r} ist ein vierdimensionaler Gewichtsvektor $\mathbf{w}_{\mathbf{r}}$ zugeordnet. $\mathbf{w}_{\mathbf{r}}$ bezeichnet einen Ort im vierdimensionalen Raum der Kamerakoordinaten \mathbf{u} und korrespondiert damit zu einem Ort im Arbeitsbereich des Roboters. Das Unternetz \mathbf{r} ist für diejenigen Objektpositionen im Arbeitsbereich zuständig, deren Kamerakoordinaten \mathbf{u} dem Gewichtsvektor $\mathbf{w}_{\mathbf{r}}$ am nächsten sind. Dies definiert einen Ausschnitt $V_{\mathbf{r}}$ des Arbeitsbereiches, für den das Unternetz \mathbf{r} zuständig ist. Jedem Knoten \mathbf{p} des Unternetzes \mathbf{r} ist ebenfalls ein vierdimensionaler Gewichtsvektor zugeordnet, bezeichnet mit $\mathbf{z}_{\mathbf{r}\mathbf{p}}$. Der Gewichtsvektor $\mathbf{z}_{\mathbf{r}\mathbf{p}}$ bezeichnet einen Ort im vierdimensionalen Raum der Kameravektoren \mathbf{x} und korrespondiert damit zu einer bestimmten Objektorientierung. Der Knoten \mathbf{p} des Unternetzes \mathbf{r} repräsentiert damit im Raumbereich $V_{\mathbf{r}}$ all diejenigen Objektorientierungen, deren korrespondierenden Kameravektoren \mathbf{x} dem Gewichtsvektor $\mathbf{z}_{\mathbf{r}\mathbf{p}}$ am ähnlichsten sind. Den Knoten \mathbf{p} sind Ausgabegrößen zugeordnet, die den Greifer des Roboters in die entsprechende Position und Orientierung bringen sollen.

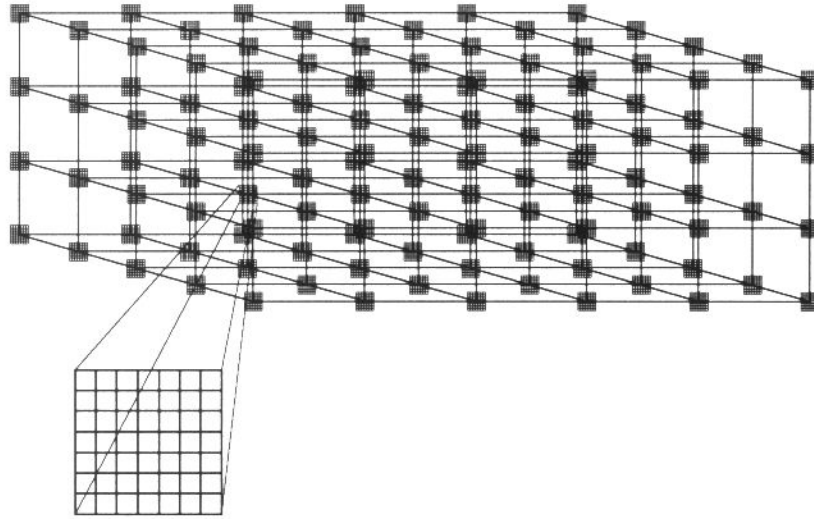


Abb.4: Hierarchische Anordnung von Kohonennetzen. Jedem Element des dreidimensionalen Übernetzes ist ein zweidimensionales Unternetz zugeordnet.

Mathematisch läßt sich die Auswahlprozedur des die Gelenkwinkel bestimmenden Knotens folgendermaßen formulieren: Hat das präsentierte Objekt die Position \mathbf{u} und die Orientierung \mathbf{x} , so wird zuerst dasjenige Unternetz \mathbf{r}^* aus dem Übernetz herausgesucht, für das

$$\|\mathbf{u} - \mathbf{w}_{\mathbf{r}^*}\| \leq \|\mathbf{u} - \mathbf{w}_{\mathbf{r}}\| \quad \text{für alle } \mathbf{r} \quad (1)$$

gilt. Anschließend erfolgt die Auswahl desjenigen Knoten \mathbf{p}^* des Unternetzes \mathbf{r}^* , für welchen

$$\|\mathbf{x} - \mathbf{z}_{\mathbf{r}^*\mathbf{p}^*}\| \leq \|\mathbf{x} - \mathbf{z}_{\mathbf{r}^*\mathbf{p}}\| \quad \text{für alle } \mathbf{p} \quad (2)$$

gilt. Damit ist \mathbf{r}^* dasjenige Unternetz, welches die Objektposition am genauesten beschreibt, und \mathbf{p}^* ist derjenige Knoten des Unternetzes \mathbf{r}^* , welcher die Objektorientierung am genauesten repräsentiert. Dieser Knoten ist sodann für die Ausgabe der passenden Gelenkwinkel zuständig.

5. Die Ausgabewerte und der Positioniervorgang

Durch die Raumposition und Orientierung des Zylinders, also die Vorgabe eines Zielpunktes für P sowie einer Richtung für \mathbf{n} , ist die benötigte Endstellung der Gelenkwinkel eindeutig festgelegt. Den drei Freiheitsgraden für die Raumposition plus den zwei Freiheitsgraden für die Orientierung des Zylinders stehen die insgesamt fünf Freiheitsgrade von Arm und Greifer gegenüber. Es ist nicht möglich, die für die verschiedenen Positionen und Orientierungen des Zylinders erforderlichen Arm- sowie Greiferstellungen als unabhängig voneinander, also entkoppelt, zu betrachten. Mathematisch bedeutet dies, daß die Gelenkwinkel des Arms $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$ sowie die Winkel des Greifers $\vec{\beta} = (\beta_1, \beta_2)$ jeweils gleichzeitig von den Eingabesignalen \mathbf{u} und den Eingabesignalen \mathbf{x} abhängen. Fassen wir alle fünf Gelenkwinkel zu einem Vektor $\vec{\phi} = (\theta_1, \theta_2, \theta_3, \beta_1, \beta_2)$ zusammen, so gilt

$$\vec{\phi}(\mathbf{u}, \mathbf{x}) = \begin{pmatrix} \vec{\theta}(\mathbf{u}, \mathbf{x}) \\ \vec{\beta}(\mathbf{u}, \mathbf{x}) \end{pmatrix}. \quad (3)$$

Das nach Präsentation des Objektes ausgewählte Neuron \mathbf{p}^* des Unternetzes \mathbf{r}^* ist für die Bereitstellung der passenden Ausgabewerte zum Einstellen der Gelenkwinkel $\vec{\phi}$ zuständig.

Dazu speichert jeder Knoten \mathbf{p} jedes Unternetzes \mathbf{r} , im folgenden mit dem Index \mathbf{rp} bezeichnet, einen Term nullter Ordnung, also einen fünfdimensionalen Vektor $\vec{\phi}_{\mathbf{rp}}$. Zur Interpolation erhält jeder Knoten \mathbf{rp} zusätzlich einen Term erster Ordnung, eine Matrix $\mathbf{A}_{\mathbf{rp}}$ von der Dimension 5×8 . Die Darstellung der zu lernenden Transformation $\vec{\phi}(\mathbf{u}, \mathbf{x})$ erfolgt damit durch eine Überdeckung des Eingabesignalraums mit lokal gültigen Linearisierungen von $\vec{\phi}(\mathbf{u}, \mathbf{x})$. Die Linearisierungen erfolgen um die Orte $\tilde{\mathbf{w}}_{\mathbf{rp}} = (\mathbf{w}_{\mathbf{r}}, \mathbf{z}_{\mathbf{rp}})$, wobei $\tilde{\mathbf{w}}_{\mathbf{rp}}$ eine Position im gesamten Eingabesignalraum, dem Produktraum $U \otimes X$, bezeichnet. Fassen wir beide Eingabesignale zu $\tilde{\mathbf{u}} = (\mathbf{u}, \mathbf{x})$ zusammen, so erzeugt das angesprochene Neuron $\mathbf{r}^*\mathbf{p}^*$ als Ausgabesignal die Gelenkwinkel

$$\vec{\phi} = \vec{\phi}_{\mathbf{r}^*\mathbf{p}^*} + \mathbf{A}_{\mathbf{r}^*\mathbf{p}^*}(\tilde{\mathbf{u}} - \tilde{\mathbf{w}}_{\mathbf{r}^*\mathbf{p}^*}). \quad (4)$$

Gleichung (4) bestimmt den ersten Bewegungsschritt, auf den anschließend eine Korrekturbewegung folgt. Dazu ist es zunächst erforderlich, nach dem ersten Bewegungsschritt die erlangte Stellung von Arm und Greifer aus der Sicht der beiden Kameras zu ermitteln. Die Bildkoordinaten des Zentrums P des Greifers in Kamera 1 und Kamera 2 fassen wir zu dem vierdimensionalen Vektor \mathbf{v}_i zusammen. Die Projektion des Normalenvektors \mathbf{n} auf die Kamerabildebene ergibt in jeder Kamera einen zweidimensionalen Vektor. Die Orientierungen beider Vektoren beschreiben die Orientierung des Greifers in Kamerakoordinaten. Das Paar zweidimensionaler Vektoren, das die Orientierung von \mathbf{n} nach dem ersten Bewegungsschritt beschreibt, fassen wir zu dem vierdimensionalen Vektor \mathbf{y}_i zusammen.

Der Positionierungsfehler $\|\mathbf{u} - \mathbf{v}_i\|$ sowie der Orientierungsfehler $\|\mathbf{x} - \mathbf{y}_i\|$ sollen mit dem Korrekturschritt verringert werden. Dazu fassen wir \mathbf{v}_i und \mathbf{y}_i zu dem Vektor $\tilde{\mathbf{v}}_i = (\mathbf{v}_i, \mathbf{y}_i)$ zusammen. Mittels der Differenz $\tilde{\mathbf{u}} - \tilde{\mathbf{v}}_i$ aus Zwischenstellung und erwünschter Endposition können wir dann den Korrekturschritt ausführen. Dieser verwendet die Jacobimatrix des angesprochenen Neurons $\mathbf{r}^*\mathbf{p}^*$ und generiert als Korrektur aller fünf Gelenkwinkel

$$\Delta\vec{\phi} = \mathbf{A}_{\mathbf{r}^*\mathbf{p}^*}(\tilde{\mathbf{u}} - \tilde{\mathbf{v}}_i), \quad (5)$$

womit wir die Endstellung des Arms und des Greifers erhalten, die wir wieder durch die Kameras beobachten und mit $\tilde{\mathbf{v}}_f = (\mathbf{v}_f, \mathbf{y}_f)$ bezeichnen.

Dieser Korrekturschritt kann mehrmals hintereinander ausgeführt werden, womit sich der Positionierungs- und Orientierungsfehler auf Werte verringern läßt, die nur noch durch Unzulänglichkeiten der in der Praxis verwendeten Apparatur nach unten beschränkt sind. Wir werden im letzten Abschnitt zeigen, wie der auf visueller Rückkopplung basierende Korrekturschritt es ermöglicht, das zu greifende Objekt behutsam und kollisionsfrei anzufahren. Zunächst beschränken wir uns auf einen einzigen Korrekturschritt.

6. Das Lernverfahren

Das Lernverfahren besteht aus zwei Teilen. Der erste Teil ist für die Bildung einer nachbarschaftserhaltenden Repräsentation der Kamerakoordinaten \mathbf{u} und \mathbf{x} , also für die Adaption der Gewichtsvektoren $\mathbf{w}_{\mathbf{r}}$ und $\mathbf{z}_{\mathbf{rp}}$, zuständig. Am Ende des Lernvorgangs sollen nur diejenigen Objektpositionen im Arbeitsbereich durch Unternetze und nur diejenigen Orientierungen

durch deren Knoten repräsentiert sein, die tatsächlich während der Lernphase vorkamen. Außerdem soll diese Repräsentation nachbarschaftserhaltend sein. Im Übernetz benachbarte Unternetze sollen am Ende der Lernphase für benachbarte Ausschnitte des Arbeitsbereiches zuständig sein, und in einem Unternetz benachbarte Knoten sollen ähnliche Objektorientierungen repräsentieren. Diese Nachbarschaftserhaltung wird im zweiten Teil des Lernverfahrens für die Adaption der Ausgabegrößen $\vec{\phi}_{\mathbf{r}\mathbf{p}}$ und $\mathbf{A}_{\mathbf{r}\mathbf{p}}$ verwendet.

Die Bildung einer nachbarschaftserhaltenden Repräsentation der Eingabegrößen wird durch Verwendung der Kohonen'schen Lernregeln erzielt (Kohonen 1989), allerdings sind diese nun der hierarchischen Struktur des Netzwerkes geeignet angepaßt (Martinetz 1992). Die Verschiebung der Unternetze im Raum der Kamerakoordinaten U erfolgt durch den bekannten Adaptationsschritt

$$\mathbf{w}_{\mathbf{r}}^{neu} = \mathbf{w}_{\mathbf{r}}^{alt} + \epsilon \cdot h_{\mathbf{r}\mathbf{r}^*}(\mathbf{u} - \mathbf{w}_{\mathbf{r}}^{alt}) \quad \text{für alle } \mathbf{r}. \quad (6)$$

Die Adjustierung der Knotengewichte $\mathbf{z}_{\mathbf{r}^*\mathbf{p}}$ erfolgt ebenso gemäß der Kohonenregel, allerdings wird nun zusätzlich zur Nachbarschaft im Unternetz noch die Nachbarschaft im Übernetz ausgenutzt. Dies führt zu

$$\mathbf{z}_{\mathbf{r}\mathbf{p}}^{neu} = \mathbf{z}_{\mathbf{r}\mathbf{p}}^{alt} + \epsilon \cdot h_{\mathbf{r}\mathbf{r}^*} g_{\mathbf{p}\mathbf{p}^*}(\mathbf{x} - \mathbf{z}_{\mathbf{r}\mathbf{p}}^{alt}) \quad \text{für alle } \mathbf{p}, \mathbf{r} \quad (7)$$

als Adaptationsschritt der Knoten \mathbf{p} der Unternetze \mathbf{r} , wobei $g_{\mathbf{p}\mathbf{p}^*}$ die mit der Entfernung $\|\mathbf{p} - \mathbf{p}^*\|$ im Unternetz abfallende Nachbarschaftsfunktion, und $h_{\mathbf{r}\mathbf{r}^*}$ die mit der Entfernung $\|\mathbf{r} - \mathbf{r}^*\|$ im Übernetz abfallende Nachbarschaftsfunktion darstellt.

Die Lernregeln des zweiten Teils des Lernverfahrens, für die Adaption der Ausgabegrößen $\vec{\phi}_{\mathbf{r}\mathbf{p}}$ und $\mathbf{A}_{\mathbf{r}\mathbf{p}}$, basieren auf Gradientenabstieg auf einer quadratischen Fehlerfunktion (Ritter et al. 1991). Dieser Gradientenabstieg liefert verbesserte Schätzwerte $\vec{\phi}^*$ und \mathbf{A}^* für den Term nullter und den Term erster Ordnung und lautet

$$\vec{\phi}^* = \vec{\phi}_{\mathbf{r}^*\mathbf{p}^*} + \delta_1 \cdot \mathbf{A}_{\mathbf{r}^*\mathbf{p}^*}(\tilde{\mathbf{u}} - \tilde{\mathbf{v}}_i) \quad (8)$$

$$\mathbf{A}^* = \mathbf{A}_{\mathbf{r}^*\mathbf{p}^*} + \delta_2 \cdot \mathbf{A}_{\mathbf{r}^*\mathbf{p}^*}(\tilde{\mathbf{u}} - \tilde{\mathbf{v}}_f)(\tilde{\mathbf{v}}_f - \tilde{\mathbf{v}}_i)^T. \quad (9)$$

Die so erzielten verbesserten Schätzwerte $\vec{\phi}^*$, \mathbf{A}^* werden zum Verbessern der Ausgabegrößen des Knotens $\mathbf{r}^*\mathbf{p}^*$ und seiner Nachbarn verwendet. Es ist möglich, auch die Nachbarn des angesprochenen Knotens $\mathbf{r}^*\mathbf{p}^*$ zu adaptieren, da die durch den ersten Teil des Lernverfahrens gebildete nachbarschaftserhaltende Repräsentation des Eingaberaumes gewährleistet, daß im Netz benachbarte Knoten ähnliche Ausgabewerte lernen müssen.

Wir haben zwei Hierarchien von Nachbarschaften, nämlich einmal eine Nachbarschaft innerhalb des Unternetzes, die wir durch eine Nachbarschaftsfunktion $g'_{\mathbf{p}\mathbf{p}^*}$ beschreiben, als auch eine Nachbarschaft zwischen den Unternetzen innerhalb des Übernetzes, beschrieben durch die Nachbarschaftsfunktion $h'_{\mathbf{r}\mathbf{r}^*}$. Dies bedeutet, daß am Lernerfolg des aktivierten Neurons $\mathbf{r}^*\mathbf{p}^*$ nun einerseits die Nachbarneuronen \mathbf{p} innerhalb des Unternetzes \mathbf{r}^* teilhaben dürfen, zusätzlich aber der Adaptationsschritt innerhalb des Unternetzes \mathbf{r}^* auch noch auf die

benachbarten Unternetze \mathbf{r} übertragen werden kann. Dies führt zu dem Adaptationsschritt

$$\vec{\phi}_{\mathbf{rP}}^{neu} = \vec{\phi}_{\mathbf{rP}}^{alt} + \epsilon' h'_{\mathbf{r}\mathbf{r}^*} g'_{\mathbf{pP}^*} (\vec{\phi}^* - \vec{\phi}_{\mathbf{rP}}^{alt}) \quad (10)$$

$$\mathbf{A}_{\mathbf{rP}}^{neu} = \mathbf{A}_{\mathbf{rP}}^{alt} + \epsilon' h'_{\mathbf{r}\mathbf{r}^*} g'_{\mathbf{pP}^*} (\mathbf{A}^* - \mathbf{A}_{\mathbf{rP}}^{alt}) \quad (11)$$

für die Knoten aller Unternetze. Diese zwei Lernschritte für die Ausgabegrößen haben die gleiche Form wie der Kohonenlernschritt (7) für die Gewichtsvektoren $\mathbf{z}_{\mathbf{rP}}$, welcher auch auf im Unternetz benachbarte Knoten und zusätzlich noch auf benachbarte Unternetze ausgedehnt ist. Anstelle des Eingangesignals \mathbf{x} treten in (10) und (11) die verbesserten Schätzwerte $\vec{\phi}^*$ und \mathbf{A}^* auf. Ausführliche Herleitungen der verwendeten Adaptionsschritte und Kriterien für die richtige Wahl der vorkommenden Lernparameter sind in (Ritter et al. 1991) und (Martinetz 1992) dargestellt.

7. Ergebnis einer Simulation

Für die Steuerung des in Abb. 1 dargestellten und im Computer simulierten Robotermodells verwendeten wir ein Übergitter bestehend aus $4 \times 7 \times 2$ Unternetzen. Jedes dieser Unternetze enthielt 3×3 Knoten. Das neuronale Netz wurde trainiert, indem dem Roboter der Zylinder an zufällig gewählten Orten seines Arbeitsbereiches in zufällig gewählten Orientierungen präsentiert wurde. Mit jeder Präsentation des Zylinders erfolgte eine Anfahrbewegung zum Greifen und ein anschließender Lernschritt des Netzwerkes.

In Abb. 5 sehen wir den Lernerfolg des neuronalen Netzes in Abhängigkeit von der Zahl der ausgeführten Lernschritte. Die beiden Graphen geben den Positionierungs- bzw. Orientierungsfehler des Greifers wieder. Der nach 10 000 Lernschritten erreichte Endwert für den Positionierfehler beträgt 0.004, was 0.6% der Abmessungen des Arbeitsbereiches entspricht. Für den Orientierungsfehler des Greifers erhalten wir am Ende der Lernphase einen Wert von 1.7° Grad. Obwohl jedes Unternetz nur 9 Elemente besitzt, ist die erzielte Orientierungsgenauigkeit des Greifers weit höher als für die Aufgabenstellung benötig.

8. Eine einfache Greifstrategie

In der beschriebenen Simulation versuchte der Roboter, direkt, also schon mit dem ersten Bewegungsschritt (4), sein Greiferzentrum P ins Zentrum des Zylinders zu führen. Dabei kann es häufig passieren, daß der Greifer den Zylinder nicht kollisionsfrei anfährt. Dies ist nicht verwunderlich, denn das Lernverfahren hat bisher die sich durch Einstellen der passenden Gelenkwinkel ergebende Bewegungstrajektorie des Greifers nicht mit in Betracht gezogen. Es ging bisher ausschließlich darum, den Positionierungs- und Orientierungsfehler der Endstellung zu minimieren.

Welche Eigenschaften muß die Bewegungstrajektorie des Greifers haben, damit Kollisionen mit dem zu greifenden Objekt vermieden werden? Menschen führen Greifbewegungen in der Regel so aus, daß die Hand zunächst grob vor das Objekt plaziert und anschließend, über einen durch visuelle Rückkopplung gesteuerten Bewegungsvorgang, um das Objekt gelegt wird. Eine entsprechende Strategie wollen wir auch für unseren Roboterarm wählen. Neh-

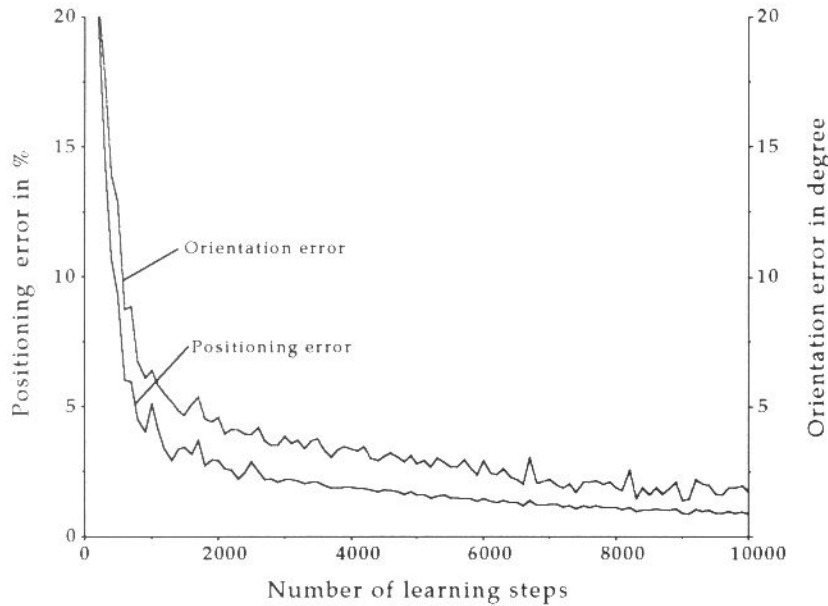


Abb.5: Der mittlere Positionierungs- und Orientierungsfehler mit der Zahl der ausgeführten Lernschritte. Nach 10 000 Versuchsbewegungen beträgt der Positionierungsfehler noch 0.004, was ca. 0.6% der Abmessungen des Arbeitsbereiches entspricht. Der Fehler in der Orientierungsgenauigkeit des Greifers ist nach 10 000 Lernschritten auf einen Wert von 1.7° abgesunken.

men wir an, daß der Greifer die passende Orientierung bereits angenommen hat. Dann genügt als zusätzliche Bedingung zur Erzielung eines kollisionsfreien Ablaufs, daß auf dem letzten Stück der Bewegungstrajektorie die Verlängerung der Symmetrieachse des Greifers die Symmetrieachse des Zylinders zu jedem Zeitpunkt schneidet. Die folgende Bewegungsstrategie gewährleistet dies:

Wir verlagern den Punkt P , der bisher das Zentrum des Greifers bezeichnete und mit dem ersten Bewegungsschritt (4) ins Zentrum des Zylinders geführt wurde, entlang der Symmetrieachse vor den Greifer. Deshalb wird der erste, “feedforward” Bewegungsschritt den Greifer jetzt immer zunächst vor das Objekt plazieren. Gleichzeitig wird der erste Bewegungsschritt bereits die passende Orientierung des Greifers einstellen. Danach beginnt, ähnlich wie bei vom Menschen ausgeführten Greifbewegungen, ein durch visuelle Rückkopplung gesteuerter Bewegungsvorgang, der das Zentrum des Greifer behutsam, ohne Kollision, ins Zentrum des Zylinders führt.

Dazu benutzen wir die durch (5) beschriebene Korrekturbewegung. Diese soll nun nicht den noch vorhandenen Positionierfehler bezüglich des vor dem Greifer liegenden Ortes P , sondern die Abweichung zwischen Greifer- und Zylinderzentrum vermindern. Diese Abweichung ist jetzt relativ groß, abhängig davon, wie weit der Punkt P vor dem Greifer liegt. Daher werden wir den Korrekturschritt diesmal nicht nur einmal, sondern mehrmals durchführen, bis der Restfehler bezüglich Position und Orientierung einen geforderten Mindestwert unterschritten hat. Bezeichnen wir die in den Kameras gesehene Position des Zentrums des Greifers nach Ausführung des ersten Bewegungsschrittes mit \mathbf{v}'_i , so erhalten wir, zusammen mit der in den Kameras gesehenen Orientierung \mathbf{y}_i des Greifers, den Vektor $\tilde{\mathbf{v}}'_i = (\mathbf{v}'_i, \mathbf{y}_i)$. Damit erhalten wir als Korrekturschritt zur Annäherung an das Objekt

$$\Delta\vec{\phi} = \gamma \mathbf{A}_{\mathbf{r} \cdot \mathbf{p}^*} (\tilde{\mathbf{u}} - \tilde{\mathbf{v}}'_i) \quad (12)$$

mit γ als die Schrittweite bestimmender Parameter. Mit dem ersten Bewegungsschritt wird

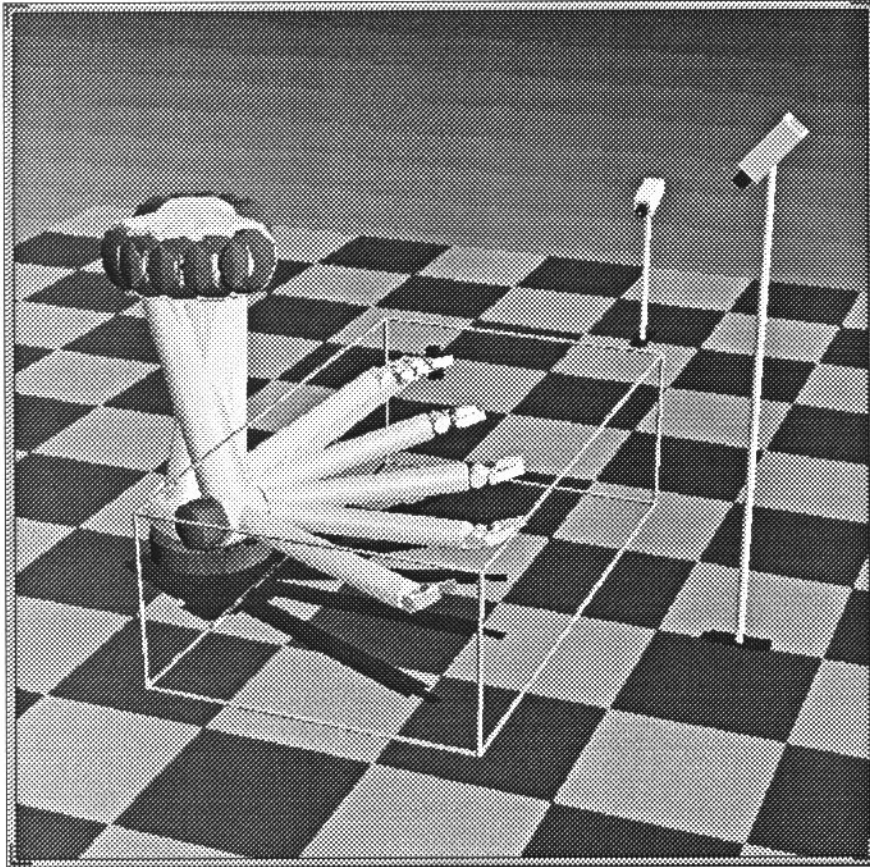


Abb.6: Eine "Stroboskopische Aufnahme" einer zufällig herausgegriffenen Greifbewegung des Roboters. Aufgrund der gewählten Bewegungsstrategie ist der Roboter in der Lage, den Zylinder ohne Kollision mit dem Objekt zu greifen.

erreicht, daß die Symmetrieachse des Greifers die Symmetrieachse des Zylinders schneidet. Mit der durch (12) gesteuerten Annäherung an den Zylinder, bei einer Schrittweite $\gamma \ll 1$ und einer dafür zusätzlichen Zahl von Rückkopplungsschritten, bleibt der Schnittpunkt beider Symmetrieachsen erhalten, und die Annäherung an das Objekt erfolgt behutsam und kollisionsfrei.

Die beschriebene Greifstrategie ist mit der verwendeten Netzwerkarchitektur in natürlicher Weise zu realisieren, da die Jakobimatrizen \mathbf{A}_{rp} einen visuellen Rückkopplungsprozeß, die Korrekturbewegung, ermöglichen. Zur Veranschaulichung des durch die Greifstrategie erzeugten Bewegungsablaufes ist in Abb. 6 eine stroboskopische Aufnahme einer zufällig herausgegriffener Greifbewegung dargestellt. Wir sehen, daß der Roboter den im Arbeitsbereich präsentierten Zylinder von vorne behutsam anfährt und ohne Kollision mit dem Objekt greift.

Literatur

- [1] Ritter H., Schulten K. (1986) Topology Conserving Mappings for Learning Motor Tasks. Neural Networks for Computing, JS Denker (ed.), AIP Conf Proc., 151:376–380, Snowbird, Utah.
- [2] Kohonen T. (1989) Self-organization and associative memory. Springer Series in Information Sciences 8, Heidelberg, 3. Auflage.
- [3] Ritter H., Martinetz T., Schulten K. (1991) *Textbook*. Neuronale Netze – eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke, revised and extended second edition, Addison-Wesley Verlag, Bonn.
- [4] Martinetz T. (1992) *Selbstorganisierende neuronale Netzwerkmodelle zur Bewegungssteuerung*. INFIX Verlag, Dissertationen zur künstlichen Intelligenz, Sankt Augustin.