Robuste Positionierung eines Roboters mittels Visual Servoing unter Verwendung einer Trust-Region-Methode

Nils T Siebel[1,2], Oliver Lang[2], Fabian Wirth[1], Axel Gräser[2]

Zentrum für Technomathematik und Institut für Automatisierungstechnik
Universität Bremen

Kurzfassung

In diesem Beitrag wird ein Verfahren zur bildbasierten Regelung eines Roboters in 6 Freiheitsgraden mit einer am Endeffektor montierten Kamera beschrieben. Der vorgestellte *Image-based Static Look-and-Move*-Regler verwendet ein lineares Modell der Regelstrecke in Form einer im Arbeitspunkt geschätzten Bild-Jacobi-Matrix. Durch die Verwendung einer *Trust-Region-Methode* in der Stellgrößenbestimmung werden Steuerbefehle vermieden, die dazu führen, dass für die Regelung relevante Objektteile (z.B. Objektmarkierungen) nicht mehr im Kamerabild sichtbar sind, ohne bei der Geschwindigkeit der Regelung Abstriche machen zu müssen. Dazu werden auftretende Modellfehler gemessen, um mit Hilfe dieser Information die Stellgrößen adaptiv zu beschränken.
Das wichtigste Problem früherer Regler ist der Umgang mit Nichtlinearitäten der Regelstrecke sowie mit fehlerhaften Modellparametern. Dies führt dazu, dass bei diesen Reglern eine befriedigende Abwägung zwischen den Zielen Zuverlässigkeit und Schnelligkeit kaum möglich ist. Der vorgestellte Ansatz führt hier zu deutlich besseren Ergebnissen.
Zur Untersuchung der Konvergenzeigenschaften des Reglers wurden zusätzlich zu Testläufen mit einem Roboter auch umfangreiche Simulationen durchgeführt.

1 Einleitung

In den letzten 3 Jahrzehnten werden verstärkt visuelle Sensoren, wie etwa Kameras, zur Regelung von Robotern eingesetzt. Dabei ergibt sich das Problem ihrer genauen Modellierung. Eine Methode, um daraus folgende Probleme zu umgehen, ist die Verwendung *bildbasierter* Regler, bei denen Modellierungsfehler tendenziell geringere negative Auswirkungen auf das Regelverhalten haben als bei *positionsbasierten* Reglern. Dieser Artikel beschreibt den Entwurf eines bildbasierten Roboterreglers, dessen Stellgesetz auf einer Trust-Region-Methode beruht. Dafür wird nach einer Beschreibung des Kamera-Roboter-Systems in Abschnitt 2 zunächst ein Modell der Regelstrecke gebildet. In den Abschnitten 4 und 5 finden sich der traditionelle sowie der neue, Trust-Region-basierte Ansatz für einen bildbasierten Regler. Die Validierung der jeweiligen Regler ist in Abschnitt 6 dargestellt.

Ein genauer Vergleich zwischen einem traditionellem und einer nicht adaptiven Variante des Trust-Region-Reglers mit Hilfe eines Simulationsprogrammes findet sich in [Lang et al. 99]. Der Schwerpunkt des vorliegenden Beitrags liegt in der Vorstellung des adaptiven Trust-Region-Reglers, insbesondere der Anpassung der Trust-Region-Parameter während der Regelung.

[1]Zentrum für Technomathematik
Universität Bremen
Postfach 33 04 40, 28 334 Bremen
E-Mail: {nils,fabian}@math.uni-bremen.de

[2]Institut für Automatisierungstechnik
Universität Bremen
Postfach 33 04 40, 28 334 Bremen
E-Mail:{siebel,olang,ag}@iat.uni-bremen.de

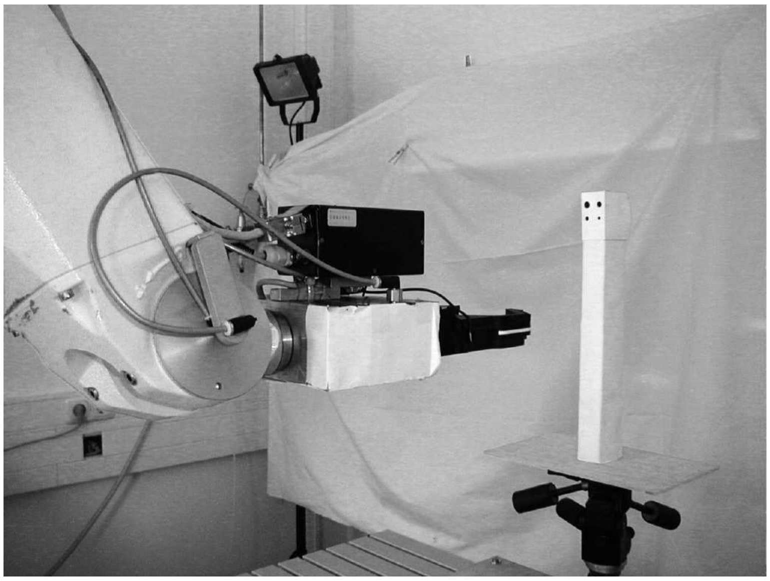


Bild 1: Systemaufbau mit Roboterarm, Kamera und Objekt

2 Image-based Visual Servoing

Der im Rahmen dieses Beitrags vorgestellte Visual-Servoing-Regler beruht auf dem als *Image-based Static Look-and-Move* bekannten Verfahren [Weiss et al. 87]. Aufgabe des Reglers ist es, den Roboter in eine bestimmte Lage (Position und Orientierung) relativ zu einem Objekt zu bewegen, um es von dort manipulieren zu können. Die Lage des Objektes im Raum ist dabei nicht bekannt; dem Regler stehen zur Berechnung der Stellgröße ausschließlich Bildinformationen zur Verfügung. Um die Ziellage zu identifizieren, wurde vorher in einem *Teachvorgang* ein dieser Lage entsprechendes Kamerabild aufgenommen und mit Hilfe von Bildverarbeitungsroutinen Soll-Bildmerkmale $y^\star$ extrahiert (*Teaching-by-Showing*). Während der Regelung findet in jedem Abtastschritt bei stillstehendem Roboter eine Bildaufnahme mit anschließender Bildmerkmalsextrahierung als ein Messvorgang statt. Der Regler soll dann nur anhand der Differenz zwischen Soll- und Ist-Bildmerkmalen (*Bildfehler*) eine Stellgröße berechnen, die der Robotersteuerung übergeben und ausgeführt wird (*Look-and-Move*). Das Ziel ist dabei, den Roboter über die Regelung in eine Lage zu bewegen, in der der Bildfehler einen akzeptierten Restfehler unterschreitet.

2.1 Versuchsaufbau und Überblick über das System

Die experimentellen Tests wurden mit einem `Stäubli Unimation RX-90` Roboter durchgeführt, an dessen Endeffektor eine `Sony EVI-331` Zoomkamera befestigt ist (Abbildung 1). Das Objekt, zu dem der Greifer des Roboters bewegt werden soll, befindet sich im Sichtfeld der Kamera. Eine Marke auf dem Objekt enthält M Markierungen in Punktform, die zur Merkmalsbildung anhand des Kamerabildes verwendet werden. In jeder Roboterpose x_n wird zunächst das Kamerabild von einer Bildverarbeitungskarte der Firma `Cognex` eingelesen und von einer Software zur Bildanalyse untersucht. Die dafür benötigte Zeit beträgt ca. 300 ms. Man erhält für jede auf dem CCD-Sensor abgebildete Objektmarkierung (“Blob”) den horizontalen und vertikalen Anteil der Position ihres Flächenschwerpunktes auf dem Sensor als jeweils ein Bildmerkmal. Diese Bildmerkmale werden paarweise in einem Messvektor abgelegt. Insgesamt steht dem Regler damit ein Messvektor y_n in Form von $m = 2M$ Bildmerkmalen zur Stellgrößenberechnung zur Verfügung (hier: $m = 8$, siehe Abbildung 1). Als Regelfehler im visuellen Regelkreis im Zeitschritt n verwenden wir den *Bildfehler* Δy_n,

$$\Delta y_n := y^\star - y_n,$$

der sich als die Differenz der Ist-Bildmerkmale y_n zum geteachten Sollwert $y^\star$ berechnet. Anhand von Δy_n liefert der Regler eine Stellgröße u_n in Form einer angestrebten relativen

Kameralageveränderung. Sie wird von den Routinen zur Roboteransteuerung unter Verwendung geschätzter externer Kameraparameter in eine neue Roboterpose umgerechnet und über ein Ethernet an den Steuerrechner des Roboters übergeben. Dort tritt nach der Berechnung der inversen Kinematik eine Trajektorieplanung in Kraft, und eine Gelenkregelung sorgt dafür, dass der Roboter eine neue Pose x_{n+1} einnimmt. Für die Kommunikation und die Ausführung des Steuerbefehles werden — je nach Größe der Bewegung im Raum — etwa 200 ms benötigt. Nach der Bewegung

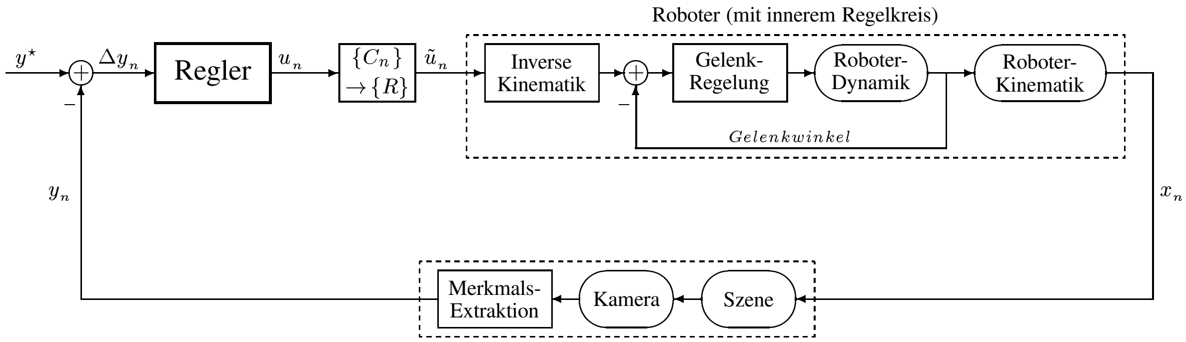


Bild 2: Geschlossener bildbasierter Regelkreis

des Roboters und damit auch der Kamera benötigt der Autofokus der Kamera eine Zeitlang, um sich auf die neue Entfernung des Objektes einzustellen; diese Zeit differierte in den Testläufen teilweise sehr. Zusammen mit der benötigten Zeit für die Kommunikation zwischen dem Computer, auf dem der Regler implementiert ist, dem Steuerrechner des Roboters, sowie der Kamera beträgt die realisierbare *Abtastzeit* des Systems etwa 1 Sekunde.

Aus der relativ großen Abtastzeit folgt die Forderung nach einem Regler, der die Aufgabe des Visual Servoing in möglichst wenigen Abtastschritten bewerkstelligt. Eine weitere Anforderung an den Regler, gerade bei der Verwendung in einem autonomen oder teilautomomen Robotersystem, ist die Sicherstellung einer erfolgreichen Regelung aus möglichst vielen Startlagen.

3 Modellierung der Regelstrecke und Regelziel

Grundlage für die Modellierung der Kamera ist das Prinzip der *Lochkamera*. Wenn die Kamerakoordinaten $({}^{C}x, {}^{C}y, {}^{C}z)^T$ eines Objektpunktes bekannt sind, sind die Koordinaten $({}^{S}x, {}^{S}y)^T$ seines Abbildes auf dem Sensor gegeben durch:

$$ {}^{S}x = \frac{{}^{C}x \cdot f}{{}^{C}z} \qquad \text{sowie} \qquad {}^{S}y = \frac{{}^{C}y \cdot f}{{}^{C}z}. \tag{1} $$

Für die Umkehrung der Abbildungsgleichungen wird zusätzlich zu den Sensordaten aus der Messung die annähernd bekannte Bildweite f der Kamera sowie der Abstand ${}^{C}z_i$ jedes der M Objektpunkte von der Kamera benötigt. Bei den vorgestellten Reglern werden die Tiefeninformationen, ${}^{C}z_i$, während der Regelung mit Hilfe eines Schätzverfahrens über die relative Objektgröße im Kamerabild bestimmt; dieses ist in [Lang et al. 99] dokumentiert.

Unter Zuhilfenahme dieser Größen lässt sich die *Bild-Jacobi-Matrix* J_n als ein im Arbeitspunkt adaptiertes, lineares Modell der Regelstrecke aufstellen. Als Vorhersage für die Änderung der Bildmerkmale durch eine Kameralageveränderung u ergibt sich für hinreichend kleine $\|u\|_2$:

$$y_{n+1} - y_n \approx J_n\, u$$

$$= \begin{pmatrix} -\frac{f}{{}^c z_1} & 0 & \frac{{}^s x_1}{{}^c z_1} & \frac{{}^s x_1\,{}^s y_1}{f} & -f - \frac{{}^s x_1^2}{f} & {}^s y_1 \\ 0 & -\frac{f}{{}^c z_1} & \frac{{}^s y_1}{{}^c z_1} & f + \frac{{}^s y_1^2}{f} & -\frac{{}^s x_1\,{}^s y_1}{f} & -{}^s x_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\frac{f}{{}^c z_M} & 0 & \frac{{}^s x_M}{{}^c z_M} & \frac{{}^s x_M\,{}^s y_M}{f} & -f - \frac{{}^s x_M^2}{f} & {}^s y_M \\ 0 & -\frac{f}{{}^c z_M} & \frac{{}^s y_M}{{}^c z_M} & f + \frac{{}^s y_M^2}{f} & -\frac{{}^s x_M\,{}^s y_M}{f} & -{}^s x_M \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_6 \end{pmatrix}. \qquad (2)$$

Dabei geben wir eine relative Kameralageveränderung u im momentanen Kamerakoordinatensystem (Positionsveränderung sowie Yaw-, Pitch- und Roll-Winkel) an. Zur Herleitung der Matrix J_n siehe [Hutchinson et al. 96] oder [Siebel 99].

3.1 Aufgabenbeschreibung

Die gewünschten Sollbildmerkmale $y^\star$ sind durch einen Teachvorgang gegeben. Im Zeitschritt $n = 0$ befindet sich der Roboter in der *Ausgangspose* x_0. Die Lage des Objektes ist nicht bekannt, doch alle Objektmarkierungen befinden sich im Sichtbereich der Kamera. Unser Ziel ist es nun, den Roboter in eine Lage x_N zu steuern, in der er eine hinreichend ähnliche Lage relativ zum Objekt besitzt wie beim Teachen. Als Kriterium der Übereinstimmung einer Lage x_n mit der gewünschten Lage wird die Größe des Bildfehlers $\|\Delta y_n\|_2$ verwendet; kleine Werte stellen eine gute Übereinstimmung dar. Um die gewünschte relative Lage zu erreichen, gibt man eine obere Schranke $\varepsilon > 0$ für den akzeptierten Restfehler vor und fordert, das keine Komponente von y_n um ε oder mehr von ihrem Sollwert in $y^\star$ abweicht. Das *Regelziel* besteht demnach darin, in eine Lage x_N zu steuern, für die gilt

$$\|\Delta y_N\|_\infty = \|y^\star - y_N\|_\infty < \varepsilon.$$

In den Testläufen zur Validierung implementierter Regler wurden mit dem Wert $\varepsilon = 1$ Pixel bei $m = 8$ Merkmalen gute Erfahrungen gemacht; die resultierende Wiederholgenauigkeit des Roboters ist für die meisten Manipulationsaufgaben ausreichend.

4 Traditioneller Reglerentwurf

Für den Entwurf eines geeigneten Feedbacks für das vorgestellte System finden sich in der Literatur nur wenige verschiedene Ansätze; die meisten Systeme verwenden einen einfachen, “traditionellen” Regelansatz — siehe etwa [Hutchinson et al. 96].

4.1 Allgemeiner Ansatz

Um zu einer gewünschten Veränderung $-\Delta y_n$ der Bildmerkmale im Bild eine möglichst genau entsprechende Kameralageveränderung u_n als Stellgröße des Reglers auszurechnen, wird ein einfaches Optimierungsproblem gelöst. Unter Verwendung eines linearen Systemmodells in Form der Bild-Jacobi-Matrix J_n wählt man

$$u_n \in \underset{u \in \mathcal{U}(x_n)}{\operatorname{argmin}} \|\Delta y_n + J_n\, u\|_2^2, \tag{3}$$

wobei $\mathcal{U}(x_n)$ eine Menge zulässiger Stellgrößen im Zustand x_n bezeichnet. Ist die Beschränkung $u_n \in \mathcal{U}(x_n)$ nicht aktiv, so ergibt sich die Lösung von (3) kleinstmöglicher Norm zu

$$u_n = J_n^+ \, (-\Delta y_n), \tag{4}$$

wobei J_n^+ die *Pseudoinverse* von J_n ist.

Für das vorgestellte System werden 4 koplanare Objektmarkierungen verwendet, das bedeutet $m = 8$ und somit $J_n \in \mathbb{R}^{8\times 6}$. Man kann zeigen, dass die im Matrix J_n im betrachteten Fall Maximalrang besitzt[3], das heißt $\operatorname{rank} J_n = 6$. Damit berechnet sich die Pseudoinverse $J_n^+ \in \mathbb{R}^{6\times 8}$ wie folgt [Deuflhard, Hohmann 93, S. 86–89]:

$$J_n^+ = (J_n^T J_n)^{-1} J_n^T. \tag{5}$$

4.2 Stellgrößenbestimmung beim traditionellem Regler

Der Berechnung einer Stellgröße liegt im Fall des traditionellen Regelansatzes ebenfalls die Pseudoinverse der Bild-Jacobi-Matrix zugrunde. Dabei wird zunächst der volle "Gauß-Newton-Korrekturterm" zur Minimierung des Bildfehlers wie in (4) berechnet:

$$\Delta u_n := J_n^+ \, (-\Delta y_n).$$

Um die Konvergenz des Reglers sicherzustellen, wird der erhaltene Vektor $\Delta u_n \in \mathbb{R}^6$ mit einem "Verstärkungsfaktor" $0 < k \leq 1$ multipliziert, um die Stellgröße u_n des Reglers zu erhalten. Man erhält das Stellgesetz des traditionellen Reglers im Abtastschritt $n \in \mathbb{N}$:

$$u_n := k \cdot \Delta u_n = k \cdot J_n^+ \, (-\Delta y_n). \tag{6}$$

Damit entspricht der traditionelle Regler einem gedämpften Gauß-Newton-Verfahren zur Minimierung des Bildfehlers Δy_n unter Verwendung eines konstanten Dämpfungsfaktors k [Deuflhard, Hohmann 93, S. 106–107, S. 112].

Die Dämpfung des Korrekturterms durch einen geeigneten konstanten Verstärkungsfaktor k ist zwar auf der einen Seite geeignet, die Konvergenz des Reglers zu garantieren, ist auf der anderen Seite jedoch der Grund für ein spezifisches Problem in der Anwendung des Reglers:
Die Bild-Jacobi-Matrix besitzt als ein lineares, geschätztes Modell der Regelstrecke nur auf einer kleinen Umgebung der momentanen Kameralage eine hinreichende Gültigkeit, um adäquate Vorhersagen über eine Merkmalsänderung zu treffen. Im Fall, dass eine Stellgröße u_n "zu groß" ist, besteht daher die Gefahr, eine relativ große, fehlgerichtete Greiferbewegung zuzulassen. Bei einer am Endeffektor montierten Kamera wie in unserem System kann eine solche Bewegung zur Folge haben, dass Objektmarkierungen aus dem Kamerabild verloren werden und die Regelung abgebrochen werden muss.
Um daher eine erfolgreiche Regelung aus möglichst vielen Startlagen zu ermöglichen, muss der Reglerparameter k hinreichend klein gewählt werden. Besonders kritisch ist dabei der *Fernbereich*, denn dort ist der Bildfehler Δy_n relativ groß, sodass das Stellgesetz (6) entsprechend große Stellgrößen liefert. Im *Nahbereich* hingegen ist der Bildfehler um bis zu zwei Größenordnungen kleiner, und das Stellgesetz liefert demgemäß kleine Kameralageveränderungen. Eine hinreichend

[3] Dabei verwendet man, dass keine drei Objektmarkierungen auf einer Geraden liegen, ${}^{C}z_i > 0$ für $i = 1, \ldots, 4$ und die Objektmarkierungen sichtbar sind (das heißt bei koplanaren Markierungen insbesondere, weder alle vier ${}^{C}x_i$ noch alle vier ${}^{C}y_i$ sind gleich 0).

kleine Wahl von k, die eine Konvergenz des Reglers auch im problematischen Fernbereich sicherstellt, bewirkt hier, dass die resultierende Geschwindigkeit des Greifers sehr niedrig ist und die Regelung eine große Anzahl von Abtastschritten benötigt.
In Abschnitt 6 dargestellte Testläufe bestätigen die Relevanz der aufgeführten Probleme; eine ausführlichere experimentelle Evaluation aller dargestellten Regler befindet sich in [Siebel 99].

5 Trust-Region-basierter Reglerentwurf

Der Entwicklung eines neuen bildbasierten Reglers liegt eine *Trust-Region-Methode* zugrunde. Die Inspiration dazu gab ein Visual-Servoing-Regler, wie ihn M. Jägersand in [Jägersand 96] für ein abweichendes System propagierte. Zur Vermeidung der Probleme, die der traditionelle Regler aufweist, wird im Stellgesetz an Stelle von k ein *variabler Reduktionsfaktor* λ_n zur adaptiven Dämpfung des Korrekturterms Δu_n eingesetzt. Unser Ziel ist es, λ_n in jedem Schritt so zu bestimmen, dass zum einen möglichst große Stellgrößen berechnet werden, um eine rasche Regelung zu ermöglichen, auf der anderen Seite jedoch u_n so beschränkt wird, dass die resultierende Bewegung die Kamera nicht aus einer Region herausführt, in der wir dem linearen Modell J_n vertrauen können— der *Trust Region* oder *Model Trust Region* [Fletcher 87].
Um diese Abwägung zwischen Abweichungen des Modells und Schnelligkeit zu erreichen, definieren wir im aktuellen Zustand einen *momentanen Modellfehler* d_n, dessen Relation zu einem akzeptierten Wert d_{soll} zur Adaption einer Schranke α_n für die vorhergesagte Bewegung beobachteter Objektmarkierungen im Bild verwendet wird.

5.1 Das Stellgesetz des Trust-Region-Reglers

Zur Bestimmung des Reduktionsfaktors λ_n definieren wir eine *Schranke* α_n für die prognostizierte Bewegung jedes der M Blobs auf dem CCD-Sensor. Für gegebenes α_n wird bei der Stellgrößenbestimmung zunächst die Länge ℓ_n der maximalen Bewegung auf dem Sensor bei Ausführung der Kamerabewegung $\Delta u_n = J_n^+ (-\Delta y_n)$ vorhergesagt:

$$\ell_n := \max_{i=1,\dots,M} \left\| \begin{bmatrix} (J_n \, \Delta u_n)_{2i-1} \\ (J_n \, \Delta u_n)_{2i} \end{bmatrix} \right\|_2 . \tag{7a}$$

Daraufhin wird im eigentlichen Stellgesetz die Stellgröße derart beschränkt, dass die vorgesagte Bewegung keiner der abgebildeten Objektmarkierungen auf dem Sensor α_n überschreitet:

$$\begin{aligned} u_n &:= \lambda_n \cdot \Delta u_n \\ &= \min \left\{ 1, \frac{\alpha_n}{\ell_n} \right\} \cdot J_n^+ (-\Delta y_n). \end{aligned} \tag{7b}$$

Es gilt nun, α_n in jedem Schritt so einzustellen, dass eine möglichst große Stellgröße zugelassen, auf der anderen Seite jedoch der Modellfehler gering gehalten wird.

Bemerkung 1 *Mit der vorstehenden Bestimmung der Stellgröße wird implizit eine Menge $\mathcal{U}(x_n)$ zulässiger Stellgrößen im Zustand x_n definiert. In der nicht-linearen Optimierung wird diese, wie bereits angedeutet, als* Model Trust Region *des Modells J_n bezeichnet.*

5.2 Messung eines Modellfehlers

Zur Bestimmung eines Maßes für die Abweichung des Modells J_n vom tatsächlichen Verhalten der Regelstrecke definieren wir zunächst zu den Bildmerkmalen y_n im n-ten Abtastschritt, $n > 0$, die entsprechenden *vorhergesagten Bildmerkmale* $\hat{y}_n$, das bedeutet

$$\hat{y}_n := y_{n-1} + J_{n-1} \, u_{n-1}. \tag{8}$$

Damit definieren wir im Abtastschritt $n > 0$ den bei Ausführung von u_{n-1} aufgetretenen *momentanen Modellfehler* d_n für M beobachtete Objektmarkierungen als das Maximum der Abweichungen zwischen vorhergesagten und tatsächlichen Blobpositionen auf dem Sensor:

$$d_n := \max_{i=1,\ldots,M} \left\| \begin{bmatrix} (\hat{y}_n)_{2i-1} \\ (\hat{y}_n)_{2i} \end{bmatrix} - \begin{bmatrix} (y_n)_{2i-1} \\ (y_n)_{2i} \end{bmatrix} \right\|_2 . \tag{9}$$

Bemerkung 2 *Die Wahl der Norm von $\hat{y}_n - y_n$ in der Definition des momentanen Modellfehlers d_n via* (9) *ist dicht an unserem Problem orientiert. Durch die Maximumsbildung werden Fehler bei der Bewegungsvorhersage einzelner Blobpositionen stark gewichtet, denn ein zu großer Fehler bereits bei* einer *Markierung kann dazu führen, diese aus dem Kamerabild zu verlieren.*

Zur Bewertung des Modellfehlers wird zusätzlich zum momentanen Modellfehler d_n ein *zulässiger (akzeptierter) Modellfehler* d_{soll} als Reglerparameter vorgegeben.
Damit definieren wir den Quotienten

$$r_n := \frac{d_n}{d_{soll}} \tag{10}$$

als *relativen Modellfehler*.
Ein kleiner Wert stellt dabei eine gute Übereinstimmung zwischen Modell und Realität dar. Angestrebt zur Abwägung von Modellfehler und einer raschen Regelung ist der Wert $r_n = 1$.
Um eine schnelle Regelung zu ermöglichen und gleichzeitig die Gefahr zu vermindern, die Objektmerkmale aus dem Kamerabild zu verlieren, möchten wir mit Hilfe von α_n den Modellfehler möglichst genau erreichen. Das Ziel dabei ist, α_n in jedem Schritt so zu wählen, dass im nächsten Abtastschritt d_{n+1} den Wert d_{soll} nach Möglichkeit genau erreicht.

5.3 Adaption von α_n

Zur Bestimmung von α_n im Abtastschritt $n > 0$ sind neben dem zulässigen Modellfehler d_{soll} der Wert α_{n-1}, der bei der Berechnung der vorigen Stellgröße verwendet wurde, sowie der Wert d_n des bei der Ausführung jener Bewegung auftretenden Modellfehlers gegeben, und r_n wird anhand von (10) bestimmt.
Die genaue Vorgehensweise bei der α_n-Adaption lässt sich durch Reglerparameter bestimmen. Dieses sind neben d_{soll}

- Minimal- und Maximalwerte α_{min} und α_{max} für α_n. Mit Hilfe dieser Schranken soll sichergestellt werden, dass die zulässige Schrittweite nicht zu große oder zu kleine Werte annimmt.
- Ein *Faktor*, der den Maximalanstieg von α_n pro Abtastschritt beschränkt. Dies geschieht, um bei der α_n-Adaption, beispielsweise durch Fehler bei der Messwerterfassung, aber auch wegen der Nichtlinearität der Regelstrecke, keine zu starke Erhöhung von α_n zuzulassen.

Grundlage hierfür ist die Abschätzung des Zusammenhangs zwischen der vorhergesagten Merkmalsänderung durch die vorige Stellgröße, $\|J_{n-1}\, u_{n-1}\|_2$, und dem resultierenden Modellfehler d_n. Falls $d_n \neq 0$, setzen wir

$$\alpha_n := \alpha_{n-1} \cdot \frac{d_{soll}}{d_n} = \frac{\alpha_{n-1}}{r_n}. \tag{11}$$

Für $d_n = 0$ wird α_n unter Beachtung der weiteren Parameter (etwa, α_{max}) maximal erhöht.

In Abbildung 3 ist die Funktionsweise der Trust-Region-Methode zur Bestimmung einer Stellgröße für *eine* beobachtete Objektmarkierung (2 Bildmerkmale) skizziert. Dabei wurde in diesem Beispiel der zulässige Wert für den Modellfehler leicht unterschritten. Für die Berechnung der nächsten Stellgröße wird α_{n+1} daher größer als α_n gewählt, um durch eine größere zulässige Schrittweite im Bild eine schnellere Regelung zu ermöglichen.

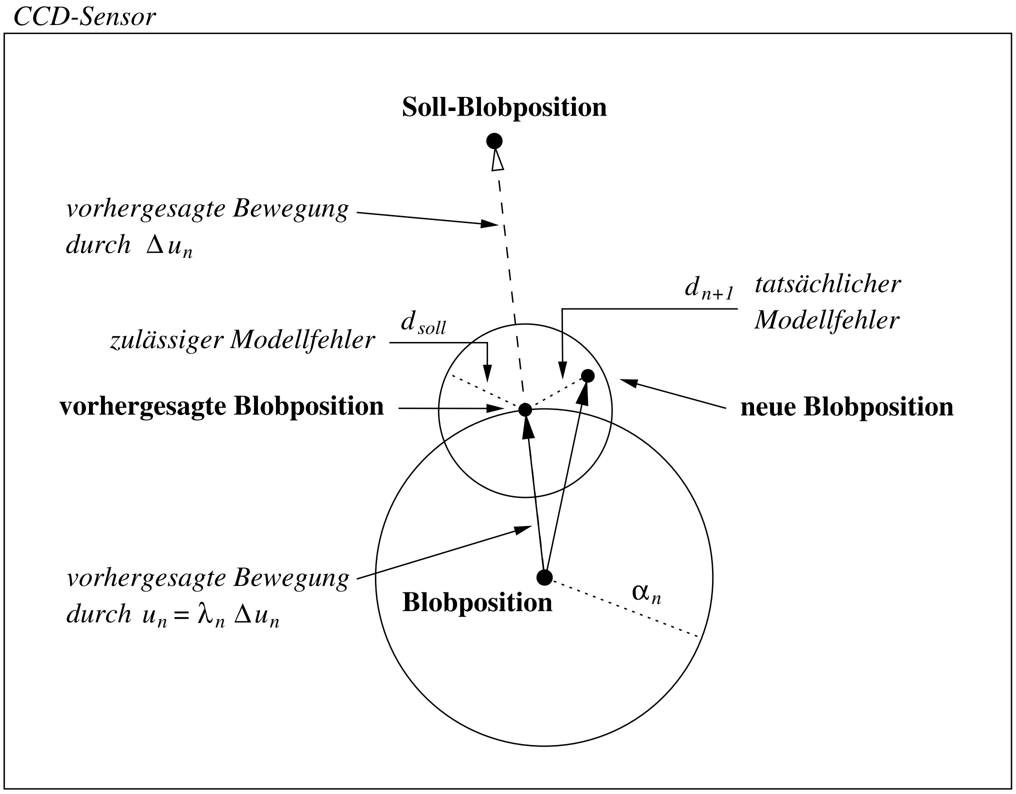


Bild 3: Konstruktion einer Stellgröße mit Hilfe einer bildbasierten Trust-Region-Methode — Betrachtung einer abgebildeten Objektmarkierung (“Blob”) auf dem CCD-Sensor

5.4 Ablauf der Regelung

Das Diagramm in Abbildung 4 beschreibt den vollständigen Ablauf einer Regelung.

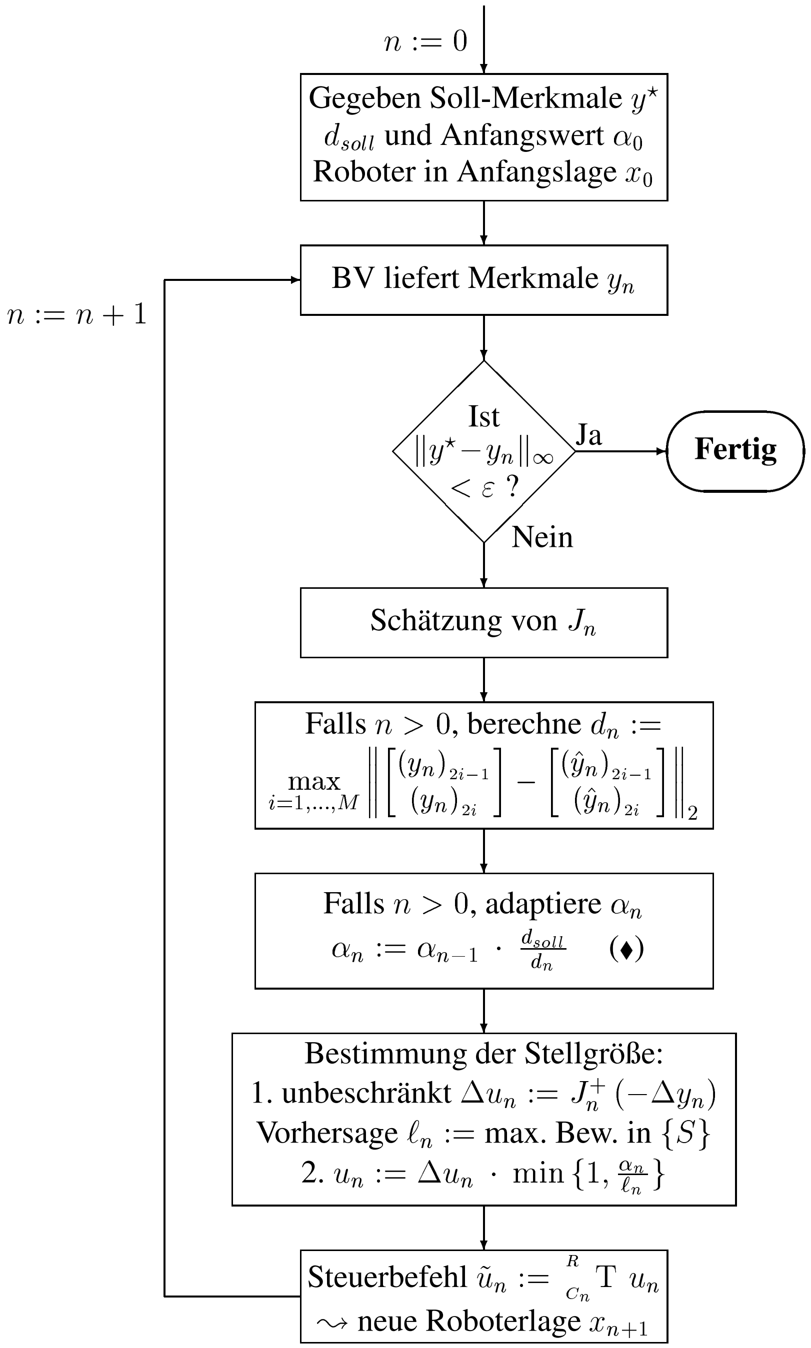


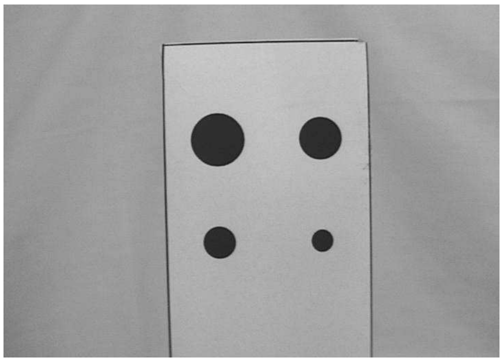
Bild 4: Ablaufplan einer Regelung mit dem Trust-Region-Regler
(♦) abhängig von Reglerparametern; siehe Abschnitt 5.3

6 Testläufe

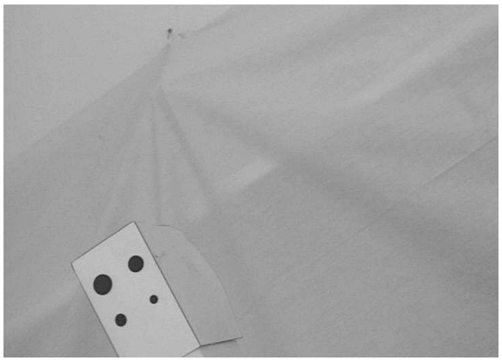
Das Verhalten des vorgestellten Reglers mit adaptierter Schranke α_n am realen Robot-Vision-System wurde am Beispiel von mehreren repräsentativen Ausgangslagen analysiert.

Grundlage für alle Versuche war ein Teachen in der Ziellage mit einer Entfernung von ${}^{C}z \approx 155$ mm zum Objekt. Die verwendete Bildweite ist konstant und beträgt $f \approx 7$ mm.
Nach dem Teachen wurde der Roboter in die jeweilige Ausgangslage bewegt und die Regelung gestartet. Die Regelung wurde als erfolgreich beendet angesehen, wenn alle Komponenten des Bildfehlers weniger als 1 Pixel betrugen. Falls nicht alle Objektmarkierungen im Kamerabild sichtbar waren, kam es zum Abbruch der Regelung.
Im Folgenden wird das Regelverhalten des traditionellen Reglers und dreier Trust-Region-basierter Regler gegenübergestellt. Es werden die Ergebnisse für eine komplizierte Startlage dargestellt.

In Abbildung 5 sind das Kamerabild in der Teachlage und in der exemplarischen Ausgangslage [150 mm,90 mm,-200 mm, 10°,-15°,30°] (relativ zur Teachlage) dargestellt.



(a) Teachlage



(b) Startlage

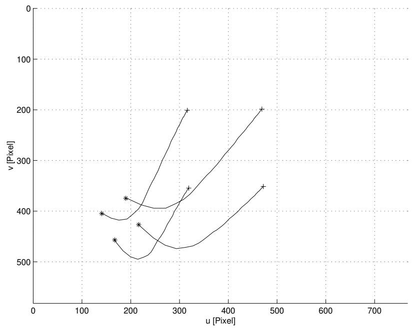
Bild 5: Kamerabild mit Objekt

Betrachtet werden dabei unter anderem die Spuren der abgebildeten Objektmarkierungen auf dem CCD-Sensor. Um eine sichere Regelung zu gewährleisten, sollen sich die Blobs während der Regelung möglichst wenig oder gar nicht auf den Bildrand zubewegen. Der Verstärkungsfaktor für den traditionellen Regler wurde so gewählt, dass das System aus möglichst vielen Lagen in die Ziellage geregelt wird. Durch empirische Versuche wurde $k = 0.07$ als günstiger Wert ermittelt. Zur Untersuchung des Trust-Region-Reglers wurde ein Regler mit konstanter Beschränkung $\alpha = 0.1$ mm sowie die adaptive Variante des Reglers mit zwei verschiedenen Parameterwerten für den akzeptierten Modellfehler, $d_{soll} = 0.04$ mm und $d_{soll} = 0.5$ mm, verwendet.
Als Unter- und Obergrenze für α_n und damit als minimale und maximale Schranke für die vorhergesagte Bewegung der Objektpunkte im Bild wurden $\alpha_{min} = 0.07$ mm und $\alpha_{max} = 0.5$ mm gewählt; dies entspricht etwa 11 und 80 Pixeln. Im ersten Reglerschritt, $n = 0$, liegt noch keine Messgröße für die Bestimmung des momentanen Modellfehlers d_0 vor, anhand derer die Schranke für die Bewegung im Bild adaptiert werden kann. Als Anfangswert für α_n wird daher ein relativ kleiner Wert von $\alpha_0 = 0.07$ mm gewählt.

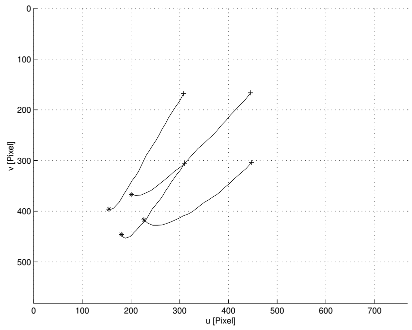
6.1 Bewegung der Objektmarkierungen im Bild

Abbildung 6 zeigt die Bewegung der abgebildeten Objektmarkierungen im Bild während der Regelung. Die Startposition der Blobs ist mit Sternen '$*$', die Zielposition mit jeweils einem Plus '$+$' markiert. Es ist zu sehen, dass die ersten Reglerschritte nicht die Position jedes Blobs im Bild in Richtung der Zielmerkmale führt. Dies ist auf die Nichtlinearität der Regelstrecke sowie Fehler in den Modellparametern zurückzuführen.

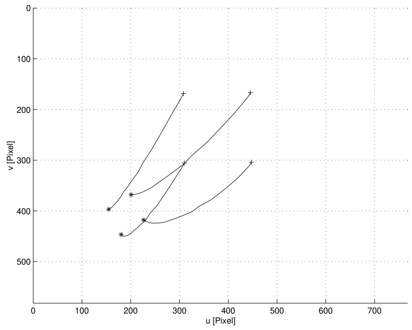
Bei qualitativ gleichartigen Bewegungsverläufen erreichen die Trust-Region-Regler ihr Ziel deutlich schneller. Der traditionelle Regler benötigte für die Regelung 85 Abtastschritte, wobei ein Verstärkungsfaktor von $k = 0.07$ verwendet wurde. Versuche zeigten, dass der Regler mit Parameterwerten von über 0.07 die Regelung nicht erfolgreich durchführen kann, da die Kamera bereits im ersten Schritt eine Objektmarkierung aus dem Bild verliert. Der gewählte Wert $k = 0.07$ führt zu entsprechend kleinen Schrittweiten; dennoch ist während der ersten Regelschritte deutlich eine Bewegung der Merkmale zum Bildrand hin zu erkennen.
Der Verlauf der Blobs bei dem Trust-Region-Regler mit konstantem $\alpha = 0.1$ mm und bei jenem mit $d_{soll} = 0.04$ mm ist vergleichbar. Beide Regler vermeiden fast vollständig eine Bewegung der Merkmale in Richtung des Bildrandes und erreichen dadurch eine sichere Konvergenz. Dabei zeigt sich in der Anzahl der Regelschritte eine Verbesserung durch die adaptive Variante von 27 auf 16 Schritte. Bei einer Erhöhung des Soll-Modellfehlers auf einen Wert von $d_{soll} = 0.5$ mm ergibt sich eine weitere Reduzierung der Schrittzahl auf 8 Schritte. Zugleich werden jedoch gerade zu



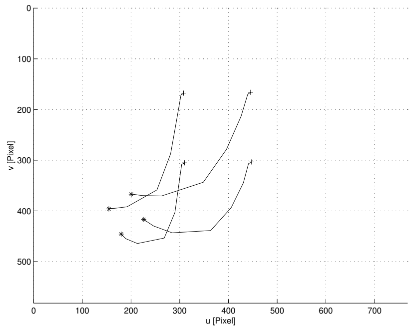
(a) Traditioneller Regler, $k = 0.07$ (85 Schritte)



(b) Trust-Region-Regler, $\alpha = 0.1$ mm (27 Schritte)



(c) Trust-Region-Regler, $d_{soll} = 0.04$ mm (16 Schritte)



(d) Trust-Region-Regler, $d_{soll} = 0.5$ mm (8 Schritte)

Bild 6: Spur der Bildmerkmale auf dem CCD-Sensor

Beginn größere Kamerabewegungen zugelassen, wodurch sich die Merkmale beinahe so stark auf den Bildrand zubewegen wie bei dem traditionellen Regler.

6.2 Verläufe der Trust-Region-Parameter: Regler mit konstantem $\alpha = 0.1$ mm

Zur Verdeutlichung der Unterschiede im Regelverhalten des traditionellen Reglers auf der einen Seite und der Trust-Region-Regler auf der anderen Seite sind in den folgenden Graphiken die Verläufe der Trust-Region-Parameter dargestellt.
Abbildung 7 zeigt den Verlauf des Reduktionsfaktors λ_n während der Regelung für den Regler mit konstantem $\alpha = 0.1$ mm. Dieser Wert entspricht dem Verstärkungsfaktor k des traditionellen Reglers; zum Vergleich mit diesem ist in Höhe 0.07 eine Hilfslinie dargestellt.
Es zeigt sich, dass dieser Regler in den ersten 9 Schritten eine größere Reduktion der berechneten Kamerabewegung erzwang als der verglichene traditionelle Regler mit konstanter Verstärkung (das heißt, $\lambda_n < 0.07$ für $n = 0, \ldots, 8$).
Im weiteren Verlauf der Regelung hatte die Schranke α_n immer weniger Einfluss auf die ausgeführte Bewegung; die letzten beiden Stellgrößen schließlich wurden gar nicht beschränkt, der Reduktionsfaktor hatte den Wert $\lambda_n = 1$. Grund hierfür ist, dass zum Schluss der Regelung der Bildfehler bereits so weit ausgeregelt ist, dass sich die Positionen der 4 Blobs im Bild weniger als α_n von ihren Sollwerten entfernt befinden. Ihre durch die Bild-Jacobi-Matrix vorhergesagte notwendige Kamerabewegung zum Ausgleich des Bildfehlers würde daher für keinen Blob eine Bewegung im Bild um mehr als α_n bewirken, und so wird die berechnete Lageveränderung vollständig ausgeführt.

Dies zeigt den konzeptionellen Unterschied zum traditionellen Regleransatz mit einem konstanten Verstärkungsfaktor k. Auch eine konstante Beschränkung α der Bewegung im Bild bewirkt eine über

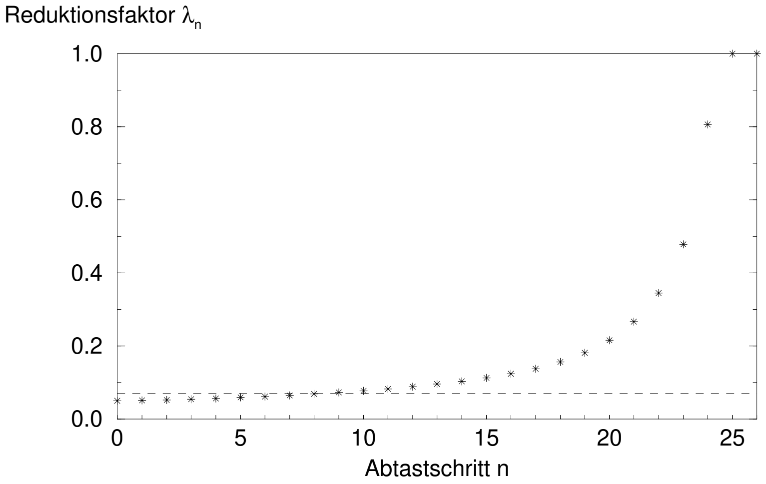
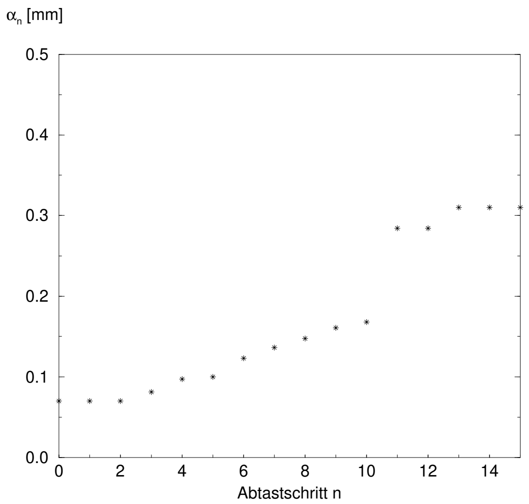
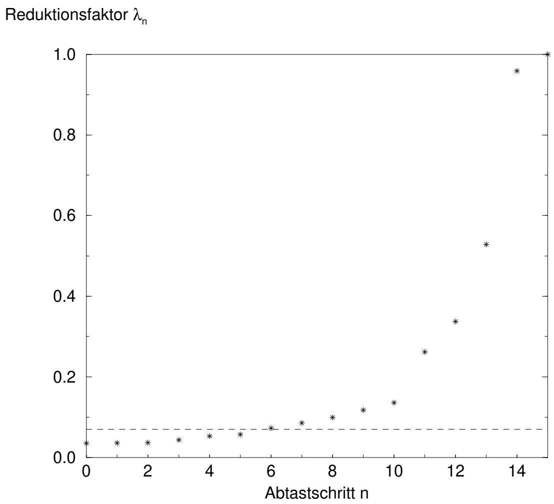


Bild 7: Erfolgte Reduktion der Kamerabewegung (Faktor λ_n)



(a) Verlauf der Schranke α_n



(b) Erfolgte Reduktion der Kamerabewegung

Bild 8: Adaption der Trust-Region-Parameter, $d_{soll} = 0.04$ mm

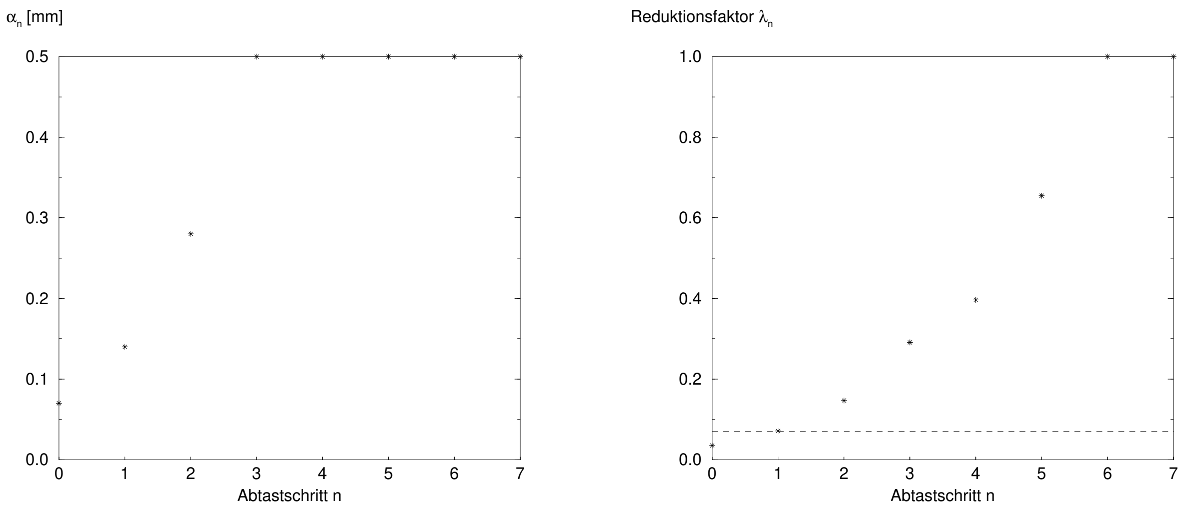
die Regelung veränderliche effektive "Verstärkung" der Kamerabewegung, da das Maß, in dem eine *Schranke* die Stellgröße beeinflusst, von der Größe des momentanen Bildfehlers abhängt.

6.3 Verläufe der Trust-Region-Parameter: Regler mit variablem α_n

Abbildung 8 zeigt den Verlauf der Trust-Region-Parameter der adaptiven Variante mit $d_{soll} = 0.04$ mm. Mit der Schranke α_n steigt der entsprechende Reduktionsfaktor von knapp $\lambda_0 = 0.035$ im ersten Abtastschritt, in dem er eine um die Hälfte kleinere Kamerabewegung erzwingt wie der Regler mit $k = 0.07$, stetig an und überschreitet die Marke $\lambda_n = 0.07$ zuerst bei der Berechnung des siebenten Regelschrittes. Danach steigt α_n stetig weiter an, und lässt so die Reduktion der Kamerabewegung schließlich im letzten Abtastschritt inaktiv werden, $\lambda_{15} = 1$.

Wie Abbildung 9(a) zeigt, ergibt sich bei dem Regler mit $d_{soll} = 0.5$ mm während der Regelung der größtmögliche Anstieg der Beschränkung α_n in Form einer Verdopplung des Anfangswertes 0.07 mm pro Schritt, bis das Maximum $\alpha_{max} = 0.5$ mm erreicht ist. Damit reduzierte dieser Regler laut Graphik 9(b) nur die erste Stellgröße stärker und die zweite bereits etwa gleich stark wie der traditionelle Regler mit $k = 0.07$; die weiteren Kamerabewegungen fallen deutlich größer aus. Mit dieser starken Anhebung des α_n-Wertes verschlechterte sich das Verhalten des Reglers gegenüber dem mit einem niedrigeren Wert insofern, dass bereits zu Beginn der Regelung, wo

Ungenauigkeiten im Modell erfahrungsgemäß deutlichere Auswirkungen haben, relative große Schrittweiten zugelassen werden. Auf diese Weise ist die Gefahr, Objektmarkierungen aus dem Kamerabild zu verlieren, tendenziell größer als bei jenem Regler — wenngleich er in allen getesteten Lagen rasch und zuverlässig konvergierte.



(a) Verlauf der Schranke α_n

(b) Erfolgte Reduktion der Kamerabewegung

Bild 9: Adaption der Trust-Region-Parameter, $d_{soll} = 0.5$ mm

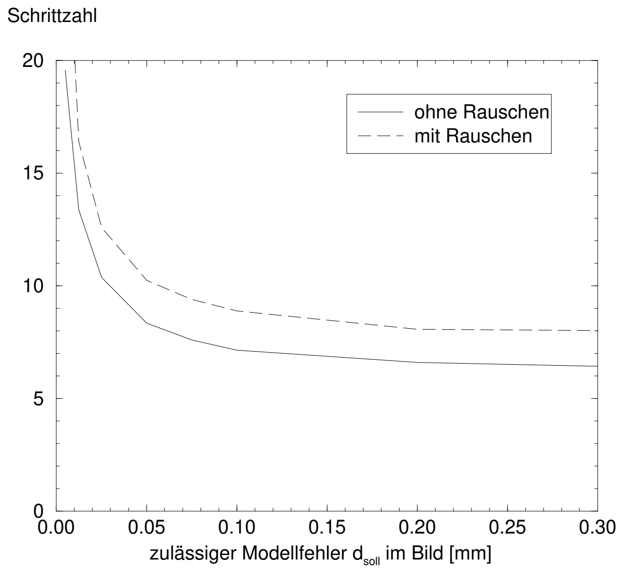


Bild 10: Anzahl Regelschritte bei adaptiver Beschränkung des Schrittes im Bild

6.4 Simulationen

Neben den Versuchen am realen Robot-Vision-System wurde die Leistungsfähigkeit der Regler unter Berücksichtigung möglichst vieler Ausgangslagen auf Simulationsebene untersucht. Für verschiedene Werte der Reglerparameter wurde die Regelung aus jeweils 1265 Startlagen heraus durchgeführt, wobei anfallende Kenndaten für eine spätere statistische Auswertung genutzt wurden. Ein in das Programm integrierter Pseudo-Zufallszahlengenerator erlaubte es fakultativ, die Messgrößen der simulierten Bildaufnahme während der Regelung mit einem Pixelrauschen zu belegen. Ein nähere Beschreibung dieses sogenannten *Multilagentests* ist in [Lang et al. 99] zu finden. Dort sind auch die Untersuchungen des traditionellen Reglers und die des Reglers mit konstantem α beschrieben. Im Folgenden wird daher nur auf das Regelverhalten mit variablem α_n eingegangen.

In Abbildung 10 ist die durchschnittliche Anzahl notwendiger Regelschritte über d_{soll} aufgetragen (Maximalanstieg der Schranke um den Faktor 2.0 pro Schritt). Es ergibt sich im dargestellten Bereich durchgängig eine Erfolgsquote von $100\,\%$, dass heißt, aus allen 1265 Startlagen wurde in weniger als 200 Abtastschritten das Regelungsziel erreicht (zur genaueren Definition der Erfolgsquote siehe auch [Lang et al. 99]). Wie die Abbildung zeigt, kann durch die adaptive Erhöhung der Schranke α_n über die Regelung eine Reduzierung der Zahl notwendiger Abtastschritte erreicht werden. Dieses ist konsistent mit den Untersuchungen am realen Robot-Vision-System. Das Rauschen bewirkt eine Erhöhung der Schrittzahl um etwa 1.5 Pixel.

6.5 Fazit

Die Untersuchung hat deutlich gemacht, dass der traditionelle Regler den Roboter auch mit einem relativ niedrigen Wert des Verstärkungsfaktors wie etwa $k = 0.1$ in einigen Fällen nicht in die Ziellage bewegen kann, weil er eine Stellgröße berechnet, die die Bildmerkmale aus dem Kamerabild herausbewegt. Diese kritische Situation kann selbst dann eintreten, wenn die Regelung in einer Lage gestartet wird, in der die Bildmerkmale einen beträchtlichen Abstand vom Bildrand haben. Das zugrunde liegende Problem besteht darin, dass der Regler zu Beginn der Regelung, wenn eine große Regelabweichung vorliegt, eine entsprechend große Stellgröße berechnet. Bei einer entsprechend großen Kamerabewegung stimmt das Modell der Strecke nicht mehr, der Arbeitspunkt wird zu weit verlassen. Zudem haben Fehler in den externen Kameraparametern, aber auch andere Modellfehler sowie verrauschte Messgrößen einen großen Einfluss, und die Bildmerkmale werden leicht aus dem Kamerabild verloren. Um daher eine Konvergenz aus möglichst vielen Ausgangslagen sicherzustellen, muss der Reglerparameter k genügend klein gewählt werden. Ein

Reglertyp und Parameter	Lage 1	Lage 2	Lage 3	Lage 4	Simulation
Traditioneller Regler, $k = 0.07$	85	85	79	72	∅ 86.9
Traditioneller Regler, $k = 0.1$	$\infty^\dagger$	50	48	49	∅ 53.1
Trust-Region-Regler, $\alpha = 0.07$ mm	37	14	18	16	∅ 25.8
Trust-Region-Regler, $\alpha = 0.1$ mm	27	10	13	13	∅ 18.7
Trust-Region-Regler, $d_{soll} = 0.04$ mm	16	11	14	12	∅ 11.1
Trust-Region-Regler, $d_{soll} = 0.5$ mm	8	5	7	8	∅ 7.9

Tabelle 1: Vergleich der Anzahl notwendiger Regelschritte verschiedener Regler in vier verschiedenen, repräsentativen Startlagen sowie die durchschnittliche Zahl der Regelschritte im Multilagentest
† Regler konvergierte nicht (Objekt wurde aus dem Bild verloren).

über gesamte Regelung konstanter, hinreichend kleiner Wert wird zwar die Konvergenz des Reglers sicherstellen, die Anzahl der Reglerschritte jedoch gerade für den letzten Teil der Regelung inakzeptabel hoch halten.

Die Validierung des Trust-Region-Reglers zeigt, dass er die Nachteile des traditionellen Reglers, nicht aufweist.

Die *Konvergenzprobleme* des traditionellen Reglers durch zu große Stellgrößen in den ersten Regelschritten wurden mit Hilfe einer hinreichenden Beschränkung der Bewegung im Bild vollständig vermieden. Damit entfällt das Risiko, das Objekt durch Ungenauigkeiten in der Modellierung und damit der Vorhersage der Änderung im Bild aus dem Bereich des Kamerabildes zu verlieren. Die besten Ergebnisse lieferte diesbezüglich der Regler mit einer adaptiven Beschränkung der vorhergesagten Bewegung im Bild und einem zulässigen Modellfehler von 0.04 mm. Ein vergleichbares Verhalten zu Beginn ergab sich mit einer festen Schranke von $\alpha = 0.1$ mm für die Bewegung im Bild.

Gleichzeitig mit der auf $100\,\%$ verbesserten Erfolgsquote konnte durch die Verwendung des Trust-Region-Reglers die *Anzahl der notwendigen Abtastschritte* gegenüber dem traditionellen Regler entscheidend reduziert werden. Dies verdeutlicht Tabelle 1, in der die Regelergebnisse der Testläufe am realen System, darunter die dargestellte Ausgangslage als Lage 1, und des Multilagentests für den traditionellen und die vorgestellten Trust-Region-Regler zusammengefasst sind.

Der Trust-Region-Regler mit einem *konstanten* Wert für α zeigt im Hinblick auf die Anzahl der Abtastschritte bereits ein gutes Ergebnis und eine ausgeprägte Verbesserung gegenüber dem traditionellen Regler. Durch die schrittweise Erhöhung von α_n über die Regelung vermag der *adaptive* Trust-Region-Regler diesen Vorteil noch auszubauen. Dabei vermindert der Regler mit einem zulässigen Modellfehler von $d_{soll} = 0.04$ mm die Zahl der Abtastschritte ohne Kompromisse im Konvergenzverhalten. Bei dem Regler mit deutlich erhöhtem zulässigen Modellfehler von $d_{soll} = 0.5$ mm erkauft man eine weitere Reduzierung der Schrittzahl mit einer höheren Wahrscheinlichkeit, das Objekt aus dem Bild zu verlieren.

7 Abschluss und Ausblick

Der neu implementierte Trust-Region-Regler ist zusammen mit den verwendeten Methoden zur Schätzung der Bild-Jacobi-Matrix für die Aufgabe des Image-based Visual Servoing sehr gut geeignet. Im Vergleich zum traditionellen Regleransatz bringt der Trust-Region-Ansatz Vorteile durch die Sicherstellung der Konvergenz des Reglers auch für schwierige Startlagen sowie eine deutlich höhere Konvergenzgeschwindigkeit. Dabei kann der Regler gut mit den nur ungenau bekannten externen und internen Kameraparametern sowie mit Mess- und Modellfehlern umgehen; durch die adaptive Beschränkung der Kamerabewegung stellt er sich auf diese Abweichungen ein und sichert die Konvergenz des Reglers. Insbesondere hat der Trust-Region-Regler in großen

Bereichen für die Werte der Reglerparameter in keinem Testlauf eine Stellgröße ausgerechnet, die bewirkte, dass die Kamera eine Objektmarkierung aus ihrem Sichtfeld verlor. Vor dem Hintergrund dieser guten Ergebnisse wird der Regler zur Zeit in das am Institut für Automatisierungstechnik entwickelte teilautonome Robotersystem `FRIEND` integriert [Lang et al. 99].

Literatur

[Weiss et al. 87] Lee E. Weiss, Arthur C. Sanderson, Charles P. Neuman: "Dynamic Sensor-Based Control of Robots with Visual Feedback". In *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 5, Oktober 1987.

[Hutchinson et al. 96] Seth Hutchinson, Greg Hager, Peter Corke: *A Tutorial on Visual Servo Control*. Yale University, New Haven, Mai 1996.
`ftp://ftp.cs.yale.edu/pub/hager/tutorial.ps.gz`

[Jägersand 96] Martin Jägersand: "Visual Servoing using Trust Region Methods and Estimation of the Full Coupled Visual-Motor Jacobian". In *Proceedings of the IASTED Applications of Control and Robotics*, 1996, S. 105–108.
`ftp://ftp.cs.rochester.edu/pub/u/jag/iastedARC96.ps.Z`

[Siebel 99] Nils T Siebel: *Bildbasierte Roboterregelung in sechs Freiheitsgraden unter Verwendung einer Trust-Region-Methode*. Diplomarbeit, Institut für Automatisierungstechnik, Universität Bremen, August 1999.
`http://www.math.uni-bremen.de/˜nils/work/Diploma.ps.gz`

[Lang et al. 99] Oliver Lang, Ralf Vogel, Nils T Siebel, Axel Gräser: "Vergleich verschiedener bildbasierter Regler zur Realisierung teilautonomer Greifvorgänge". In *15. Fachgespräch Autonome Mobile Systeme (AMS'99), Technische Universität München*, Springer-Verlag, 1999.

[Fletcher 87] Roger Fletcher: *Practical Methods of Optimization*. 2. Auflage, John Wiley & Sons, New York, Chichester 1987.

[Deuflhard, Hohmann 93] Peter Deuflhard, Andreas Hohmann: *Numerische Mathematik I. Eine algorithmisch orientierte Einführung*. 2. Auflage, de Gruyter, Berlin und New York, 1993.