

Sensorgeführte Roboter für industrielle Aufgaben

Friedrich Lange und Gerd Hirzinger

*Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR), Oberpfaffenhofen
e-mail: Friedrich.Lange@dlr.de*

Zusammenfassung

Der Artikel zeigt auf, dass die Probleme früherer sensorgestützter Bahnplanungs- und Regelungsverfahren weitgehend überwunden sind. Dazu wird eine Architektur vorgestellt, die sich leicht in übliche Robotersteuerungen integrieren lässt. Sie besteht aus verschiedenen Modulen, die es dem Anwender erlauben, in industriellen Umgebungen, auch bei hohen Geschwindigkeiten, kostengünstig Sensoren einzusetzen, die die Genauigkeit vorgegebener Bewegungen bezüglich messbarer Referenzen deutlich erhöhen. Das Verfahren erlaubt die Ausführung von am CAD-System geplanten Bahnen ohne weiteres Teach-In.

Abstract

The article shows that former problems with sensor based path planning and control methods have been solved. An architecture is presented which easily can be integrated into usual robot control devices. The method consists of several modules which allow the user at a reasonable price to apply sensors to improve the accuracy of given motion with respect to measurable invariants, even at high robot speed. The method allows the application of CAD-based planned paths without further teach-in.

1 Einleitung

Bisher wurde Sensorik bei Robotern hauptsächlich in prototypischen Anwendungen verwendet, vor allem von Hochschulen und Forschungseinrichtungen. Diese Systeme waren zwar in der Lage, mit weitgehend unbekanntem Umgebungen fertig zu werden, zur Serienreife gelangten sie aber nicht, da sie meist teuer, empfindlich und langsam waren. So blieben als Anwendung nur Bereiche, in denen Menschen nicht oder nur mit hohem Aufwand arbeiten können, z. B. in der Raumfahrt [10].

Industriell wird Sensorik bei Robotern nur dann eingesetzt, wenn sie kostengünstig ist, auch bei ungünstigen Bedingungen wie Staub robust funktioniert und die Zeit zur Bearbeitung einer Aufgabe nicht verlängert. Dabei erlaubte der letzte Punkt in der Vergangenheit meist nur einzelne Messungen (looking - then moving), da wegen der hohen Bearbeitungsgeschwindigkeiten keine online Sensordatenrückführung möglich war.

Auf der anderen Seite sind industrielle Bedingungen in mancherlei Hinsicht recht einfach. So ist das Umfeld bis auf kleine Unsicherheiten bekannt oder zumindest konstant. Das betrifft sowohl kinematische als auch dynamische Effekte, die, soweit sie nicht schon modelliert sind, offline gelernt und dann kompensiert werden können. Die online Sensorik hat somit die Aufgabe, verschleißbedingte Änderungen des Systems und kleine Abweichungen in der Position oder Form von Werkstücken zu erfassen und die Bewegung entsprechend online zu modifizieren.

Industrielle Systeme ohne Sensorik erlauben es nicht, off-line geplante Bahnen direkt einzusetzen. Stattdessen müssen durch einen Teachvorgang eines Bedieners die Stützpunkte der

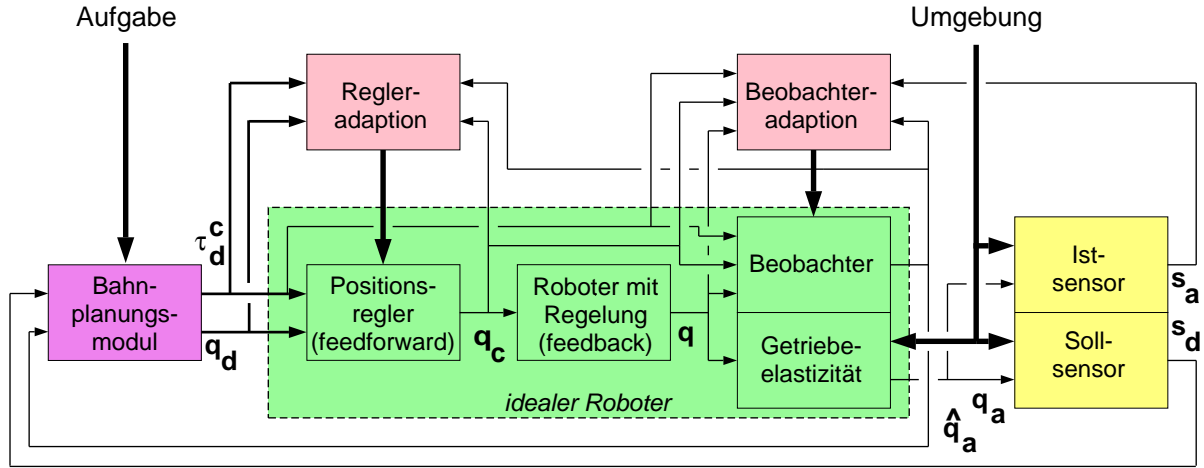


Abbildung 1: Architektur

Bahn zunächst statisch versetzt und dann in Abhängigkeit der Geschwindigkeit noch weiter verschoben werden. Die maximale Geschwindigkeit ist schließlich erreicht, wenn Schwingungen des Systems angeregt werden. Unberücksichtigt bleiben die Einflüsse durch Verschleiß oder Alterung des Manipulators sowie durch Ungenauigkeiten der vorangegangenen Fertigungsprozesse.

In diesem Artikel wird daher aufgezeigt, wie Sensordaten auch bei hohen Geschwindigkeiten online zur Korrektur der programmierten Bahn eingesetzt werden können, um die genannten Einschränkungen weitgehend aufzuheben. Die maximale Bahngeschwindigkeit ergibt sich dann aufgrund der maximalen Antriebsmomente. Das Teach-In einer offline geplanten Bahn kann entfallen.

Im nächsten Abschnitt wird eine Übersicht über die vorgeschlagene Struktur der Steuerung gegeben. Danach werden die einzelnen Module erläutert.

2 Steuerungsarchitektur mit online Sensorik

Die Architektur besteht aus mehreren Modulen (Abbildung 1): Sie definiert einen *idealen Roboter*, der aus dem realen System und zwei zusätzlichen Blöcken besteht. Weiterhin sind Module vorgesehen, die die beiden Blöcke adaptieren und dadurch den Roboter ideal machen und schließlich gibt es Module zur Interpretation der Aufgabenbeschreibung.

Eingang des *idealen Roboters* ist die Sollbahn des tool center points (TCP), die im Achsraum punktweise für jeden Abtastschritt gegeben ist. Ein *kartesischer idealer Roboter* ließe sich auch definieren (Abbildung 2). Davon wird hier aber abgesehen, da Ungenauigkeiten in den kinematischen Transformationen sich i. allg. sensor geregelt ausgleichen lassen. Auf der anderen Seite können ggf. vorhandene Freiheitsgrade der Rücktransformation bei der Bahnplanung genutzt werden.

Ausgang des *idealen Roboters* ist die Istposition des TCP, ebenfalls im Achsraum. Während am Ausgang zum Zeitpunkt k nur die aktuelle Istposition $\hat{\mathbf{q}}_a(k)$ definiert ist, werden am Eingang mehrere Sollpositionen $\mathbf{q}_d(k) \cdots \mathbf{q}_d(k + n_d - 1)$ vorgegeben, da nur dadurch die Einhaltung der Sollbahn erreicht werden kann. Dabei entspricht n_d größenordnungsmäßig der Zahl der Abtastschritte innerhalb der Zeitkonstante des dynamischen Systems.

Sofern der Roboter kraftschlüssig mit der Umgebung verbunden ist, müssen auch die der erwarteten (vorgegebenen) Kraft \mathbf{F}_d entsprechenden Achsmomente τ_d^c dem *idealen Roboter* mitgeteilt werden, und zwar auch für den Zeitraum von einigen Abtastschritten. Erst mit die-

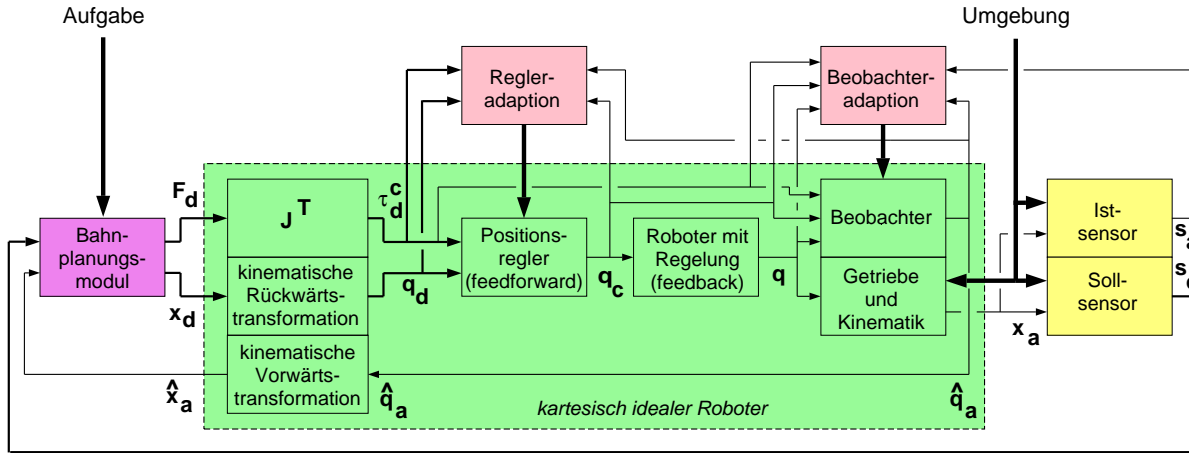


Abbildung 2: Alternative Architektur mit *kartesisch idealem Roboter*

sen Momenten sind die Informationen ausreichend, um den Roboter eine Sollbewegung korrekt ausführen zu lassen.

Der *ideale Roboter* selbst besteht aus drei Teilen. Das ergibt sich aus der Forderung nach einer leichten Integrierbarkeit in bestehende Robotersteuerungen. So beschreibt der mittlere Block den mechanischen Roboter mit seinen Antrieben und der Steuerung, soweit sie üblicherweise von Roboterherstellern verkauft wird. Der in diesem Artikel interessierende Teil der Steuerung regelt die Armpositionen aufgrund von Achskommandos q_c (meist Sollwert genannt), auf die über eine Sensorschnittstelle in jedem Abtastschritt zugegriffen werden kann.

Zur internen Regelung wird normalerweise eine Kaskadenregelung eingesetzt, die die Motorströme berechnet. Diese Regelung allein erlaubt kein ideales Bahnfahren entlang einer Solltrajektorie, da der Kaskadenregelung die Sollbahn jeweils nur in Form einer Momentaufnahme des Kommandovektors vorliegt. Weitere Zeitpunkte der Sollbahn sind unbekannt.

Es empfiehlt sich daher, einen weiteren Regler vorzuschalten, der den oben definierten Abschnitt von n_d Zeitpunkten der Sollbahn verarbeiten kann. Das wird in Abschnitt 3.1 ausgeführt. Dieser Regler wird adaptiv angesetzt, damit man ihn an die Roboterdynamik anpassen kann. Es zeigt sich, dass in vielen Fällen eine Vorsteuerung ausreicht, also keine Rückkopplung der Istposition nötig ist.

Bei genauerer Betrachtung beschreiben die von der Robotersteuerung ausgegeben Istwerte nicht die Gelenkwinkel der Arme q_a sondern die, mit dem entsprechenden Übersetzungsverhältnis skalierten Gelenkwinkel der Motoren q . Man spricht hierbei auch von antriebsseitiger Messung im Gegensatz zu den die wirkliche Position des TCP definierenden abtriebsseitigen Gelenkwinkeln. Diese beiden Messmethoden unterscheiden sich bei großen Beschleunigungen, sofern die Getriebe elastisch sind, was auch bei starren Industrierobotern angenommen werden muss. Maßnahmen zur Berücksichtigung von Elastizitäten werden im Abschnitt 3.2 vorgestellt. So kann man die Motoristwerte q bei elastischen Getrieben nicht zur Regleradaption verwenden. Stattdessen werden die Armistwerte q_a beobachtet. Der Beobachter selbst ist auch adaptiv angesetzt. Zur Adaption wird neben den vorliegenden Achswerten und den Kontaktkräften auch die aktuelle Position des TCP benötigt. Sie wird sensorisch bestimmt. Sofern die Sensorik in der Lage ist, online sämtliche Freiheitsgrade zu messen, kann man den Beobachter auch einsparen und stattdessen die Sensorwerte in Achswerte umrechnen.

Alternativ zu dieser Struktur des *idealen Roboters* mit adaptierten Modulen ist auch eine modellbasierte Implementierung von Regler und Beobachter ohne Verwendung von Positionskommandos q_c möglich. In dem Fall entfallen die Adaptionen sowie die Trennung der beiden

Regler des *idealen Roboters*. Unter Umständen bietet sich dann auch eine andere Schnittstelle an, die die Vorgabe von Ableitungen der Sollposition zum aktuellen Zeitpunkt anstelle zukünftiger Sollbahnpunkte vorsieht.

Ein anderer Block in der Architektur nach Abbildung 1 interpretiert die Aufgabenbeschreibung. Diese ist i. Allg. symbolisch gegeben und wird vom Bahnplanungsmodul (Abschnitt 3.3) in eine Sollbahn mit vorgegebenem Geschwindigkeitsprofil umgewandelt. Dabei müssen die Sensordaten integriert werden, sofern die Bahn aufgabenabhängig modifiziert werden soll. So wird z. B. die Aufgabe „Spritze Klebstoff im Abstand von 1 cm von der Kante“ entsprechend der CAD-Daten über die Kante in eine Nominalbahn konkretisiert und in einen Regler, sofern die Lage der Kante unsicher ist. Die Sensordaten beschreiben hierbei die Lage der Kante und damit indirekt die Sollbahn.

Abbildung 1 trennt zwischen dem Sensor für die Sollbewegung und dem Sensor für die Istbewegung. Letzterer wird nur bei Elastizitäten oder unsicherer Kinematik benötigt und auch dann nur während der Adaptionsphase, nicht aber während der Aufgabenausführung. Die Trennung der beiden Sensorarten ist logisch gemeint. Beide Systeme können aus derselben Hardware bestehen, z. B. einer Kamera, die sowohl raumfeste Marken als auch die Merkmale der Sollbahn detektiert. Die Unterscheidung der beiden Arten von Sensoren ist nötig, da man die Istposition bei der aufgabenunabhängigen Adaption des Reglers des *idealen Roboters* braucht.

Zu den Abbildungen kann bemerkt werden, dass nicht nur Signalflüsse dargestellt sind, sondern auch Abhängigkeiten, z. B. der Einfluss der Roboterposition auf die Sensorwerte.

3 Mögliche Module

In diesem Abschnitt werden drei mögliche Module aufgezeigt, die unabhängig voneinander einsetzbar sind und für deren Realisierung in der Literatur verschiedene Möglichkeiten vorgestellt wurden.

3.1 Erhöhung der Bahngenauigkeit

Dieser Abschnitt betrachtet die Realisierung des *idealen Roboters*. Das bedeutet einen Reglerentwurf, durch den jede Sollbahn im Bereich des jeweiligen Arbeitspunktes exakt abgefahren wird.

Bei der Architektur nach Abbildung 2 ist zur Realisierung eines *kartesisch idealen Roboters* auch eine Verbesserung der i. allg. ungenauen kinematischen Transformationen nötig. Dies geschieht üblicherweise durch Korrektur der Denavit-Hartenberg Parameter (siehe z. B. [3]). Zusätzlich spielen Elastizitäten eine Rolle bei der statischen Genauigkeit [5]. Beide Punkte sind von untergeordneter Bedeutung, wenn die Bahnplanung sensorgeführt erfolgt, da statische Ungenauigkeiten sich ähnlich wie Abweichungen der Sollbahn von der Nominalbahn auswirken. Trotzdem erleichtern statisch absolut genaue Roboter die Regelung, da der Einfluss kinematischer Fehler wegen der nichtlinearen Transformationen nur annähernd statisch ist.

In diesem Artikel wird nur auf die dynamische Bahngenauigkeit eingegangen. Modellbasierte Ansätze [24, 21] greifen in die vorhandene Robotersteuerung ein und kommandieren so direkt die Achsmomente. Dies erfordert großen Hardwareaufwand und genaue Modellkenntnisse und soll daher hier nicht weiter verfolgt werden. Ein anderer Ansatz optimiert die Parameter der herkömmlichen Regler [19]. Hier werden stattdessen die Blöcke „Positionsregler“ und „Regleradaption“ aus Abbildung 1 oder 2 ausgefüllt. Neben einfachen adaptiven Regelansätzen bietet sich eine prädiktive Regelung [8, 7] an, die den Regler aufgrund der identifizierten Modelldaten entwirft. Hier wird dagegen ein von den Autoren entwickeltes iteratives Verfahren [13] kurz

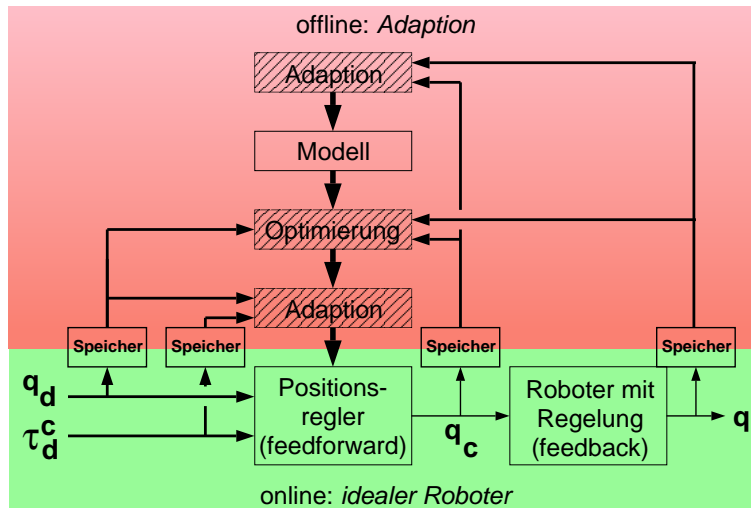


Abbildung 3: Struktur der Adaption (ohne Elastizität)

umrissen, das auch eine Verfeinerung des Regelgesetzes aufgrund verbleibender Regeldifferenzen erlaubt (Abbildung 3).

Die Adaption erfolgt offline in 3 Stufen (schraffierte Blöcke) anhand von ausgewählten Trajektorien. Nach dem ersten Testlauf wird der Roboter (mit feedback Regelung) identifiziert (1. Stufe). Dann wird der Positionsregler iterativ adaptiert, indem jeweils die letzte Testtrajektorie a posteriori optimiert wird (2. Stufe), woraufhin die Reglerparameter angepasst werden (3. Stufe). Dies geht über *iterative learning control* (ILC) [20, 17, 4] hinaus, da dort kein Regler adaptiert wird. Für den *idealen Roboter* ist aber ausdrücklich ein Verfahren gesucht, das zumindest im Bereich des aktuellen Arbeitspunktes beliebige Bahnen anhand der in Abschnitt 2 definierten Eingänge optimal steuert.

Der iterative Ansatz erlaubt auch bei nur grob identifiziertem Modell eine hohe Güte, da verbleibende Fehler immer wieder korrigiert werden. Daher wird für den Positionsregler neben dem linearen Ansatz der Einsatz neuronaler Netze zur Berücksichtigung von Kopplungen und, im Kontaktfall, auch ein Regler zur Kompensation externer Kräfte vorgeschlagen (Abbildung 4). Die Alternative eines umfassenden Netzes, das die Kommandos q_c direkt berechnet, ist nicht möglich, da die Genauigkeit nicht ausreicht. So muss genutzt werden, dass der Roboter sich durch die großen Motordrehmomente annähernd linear verhält, so dass die nichtlinearen Anteile der Regelung feiner skaliert werden können. Zur Abbildung werden mehrschichtige Perzeptrons mit geeigneten Trainingsverfahren [22, 11] vorgeschlagen.

Eine Rückkopplung von Messwerten ist wegen der hohen Wiederholgenauigkeit moderner Industrieroboter dagegen nicht nötig. Beim Positionsregler handelt es sich also um eine Vorsteuerung. Bei modellbasierten Verfahren entspricht dies der *computed torque method*, da die Rückkopplung (in dem vom Hersteller gelieferten Teil der Steuerung) unabhängig von der gewünschten Bewegung ist.

3.2 Berücksichtigung von Getriebeelastizitäten

Bei der adaptiven Struktur nach Abbildung 1 müssen Elastizitäten nur dann gesondert berücksichtigt werden, wenn die Position des TCP nicht direkt messbar ist. Man muss dann zwischen zwei Fällen unterscheiden: Im einen Fall hat man eine externe Messeinrichtung für die Istpositionen des TCP und kann damit wie gewohnt den Regler adaptieren und die Sollbahn bestimmen. Am einfachsten ist es, wenn man für Istsensoren und Sollsensoren dieselbe Hardware verwendet.

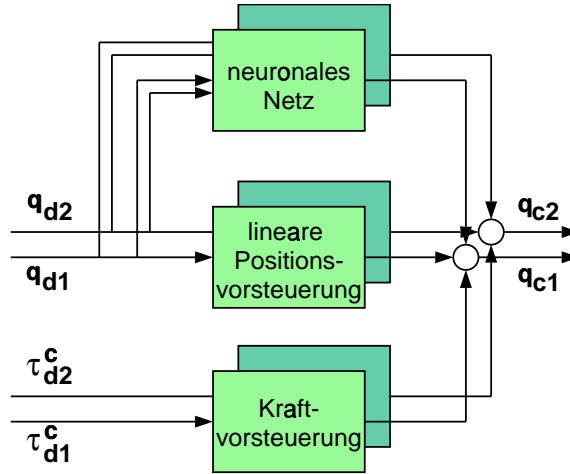


Abbildung 4: Module des Positionsreglers bei einem 2-achsigen Roboter

Dazu eignen sich aufgrund des hohen Informationsgehalts insbesondere bildgebende Sensoren. In [6, 15] wird daher eine handelsübliche CCD-Kamera zur hochgenauen Bestimmung der Sollposition eingesetzt.

Im anderen Fall steht die Messeinrichtung nur eingeschränkt zur Verfügung, z. B. weil die Daten offline übertragen werden, weil nur in wenigen Freiheitsgraden gemessen wird oder weil die Messeinrichtung nach der Adaptionsphase aus Kosten- oder Platzgründen abgebaut wird. Dann muss, zumindest zur online Erfassung der Sollbahn nach Abschnitt 3.3, ein Beobachter entworfen oder adaptiert werden, der die TCP-Position im laufenden Betrieb zur Verfügung stellt.

So ein Beobachter muss besser als das Modell nach Abbildung 3 sein, da durch Elastizitäten schwach gedämpfte Schwingungen angeregt werden können. Das Gedächtnis des Beobachters muss daher, sofern es sich auf die in Abbildung 1 gezeigten messbaren Eingänge beschränkt, einen Zeitraum von mehreren Sekunden abdecken. Daneben ist eine hohe Genauigkeit erforderlich, da sich sonst Phasenverschiebungen gegenüber der darzustellenden Schwingung ergeben. Dies ist ggf. nur durch einen modellbasierten Ansatz oder durch zusätzliche Sensorik (wie in [1]) möglich. Zu diesem Punkt sind noch weitere Untersuchungen nötig.

Daher wird in [14] ein Verfahren beschrieben, bei dem der Positionsregler aufgrund von offline verfügbaren Daten adaptiert wird. Dabei wird wegen der zweidimensionalen kartesischen Messung auch auf das Problem der Transformation von den Messwerten zu den Roboterachsen eingegangen. Mit $\hat{\mathbf{x}}_a \approx \mathbf{x}_d$ erlaubt das Verfahren näherungsweise auch eine online Bahnplanung wie in [15].

3.3 Sensorgestützte Bahnplanung

Dieser Abschnitt beschreibt die Integration von Sensordaten in die Steuerung. Dabei wird eine allgemeine Darstellung gewählt (Abbildung 5), die sowohl zur Kraftregelung [12] als auch zur Regelung mit ortsfesten oder robotergeführten CCD-Kameras [15] oder sonstigen Sensoren geeignet ist. Im Gegensatz zu direkten Sensorregelungen, z. B. einer Kraftregelung, ist die vorgeschlagene Struktur nicht auf einzelne Sensoren begrenzt. Vielmehr können unterschiedliche Sensortypen fusioniert werden, sofern jeder Sensor eine Positionsänderung ergibt.

Abbildung 5 enthält, wie bei Industrierobotern üblich, eine durch kartesische Punkte \mathbf{x}_p und die entsprechend gewünschten Sensorwerte \mathbf{s}_p gespeicherte Bahn, für die zunächst die Werte zu den Abtastzeitpunkten interpoliert werden. Dies bedeutet u. A. die Festlegung eines Geschwin-

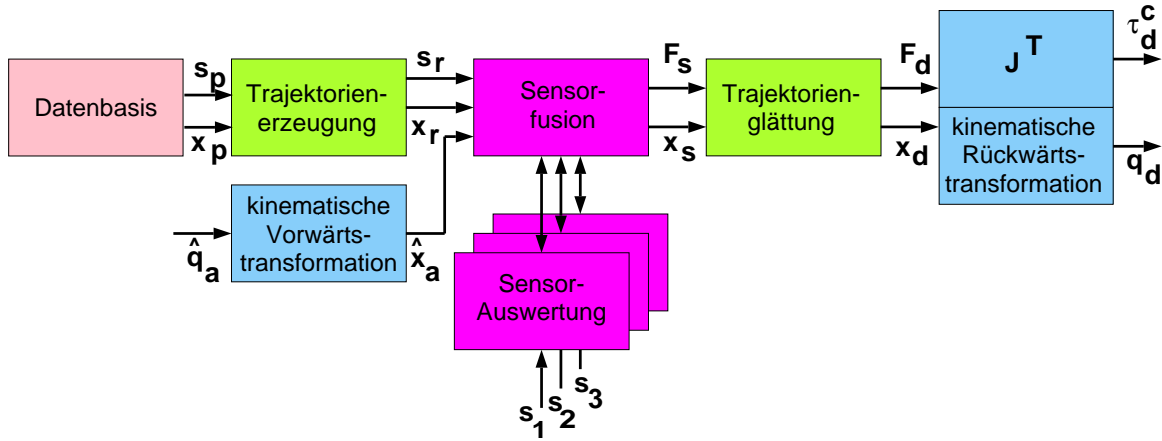


Abbildung 5: Allgemeines Bahnplanungsmodul

digkeitsprofils. Meist wird dabei die Gesamtzeit modellgestützt minimiert [2, 23].

Im nächsten Block werden die aktuellen Sensorwerte integriert. Das Prinzip der in Abbildung 5 dunkel unterlegten Blöcke besteht im Gegensatz zu üblichen Regelungen darin, dass die aus Positionen \mathbf{x}_r und Sensorwerten \mathbf{s}_r bestehende Referenztrajektorie mit den Istwerten verglichen wird.

Vor der Weitergabe an den Eingang des *idealen Roboters* muss die Sensortrajektorie \mathbf{x}_s zumindest bei grob auflösenden Sensoren noch geglättet werden, da der *ideale Roboter* auch große Beschleunigungen nach Möglichkeit fehlerfrei ausführt. Wegen der Verwendung im *idealen Roboter* werden auch die auftretenden Kontaktkräfte berechnet. Andere Sensorwerte spielen dagegen keine Rolle mehr.

Außerdem sind die kinematischen Transformationen aus Abbildung 2 zu berechnen. Dabei wird die Rückwärtstransformation absolut berechnet, also nicht durch die inverse Jacobi genähert, da sich sonst ein leichtes Driften in unregulierten Raumrichtungen ergeben kann.

Die in Abbildung 5 dunkel unterlegten Blöcke sollen näher erläutert werden. Dabei wird angenommen, dass sich aufgrund der Sensorwerte und der aktuellen Position, bei der die Sensorwerte gemessen werden, eindeutig eine gewünschte Position für den TCP ergibt. Diese Formulierung schließt sowohl eine direkte Kraftregelung als auch eine Impedanzregelung ein, sofern die natürliche Steifigkeit des Systems bekannt oder messbar ist. Das ist immer erfüllt, wenn ein Roboter über die Positionsschnittstelle kraftgeregelt werden kann [12]. Die Darstellung nach Abbildung 5 ist auch allgemein für sensorabhängige Fügebewegungen [18] geeignet.

Ein einfacher Ansatz interpretiert die Differenz zwischen Soll- und Istkraft als Positionsfehler. Die Sollposition ergibt sich in sensorgeregelten Richtungen als Summe von aktueller Position \mathbf{x}_a und Positionsfehler $\Delta\mathbf{x}_a$, ansonsten durch die Referenztrajektorie \mathbf{x}_r .

$$\mathbf{x}_s = \begin{cases} \mathbf{x}_a + \Delta\mathbf{x}_a & \text{in Krafrichtung} \\ \mathbf{x}_r & \text{sonst} \end{cases} \quad (1)$$

mit

$$\Delta\mathbf{x}_a = \frac{\mathbf{F}_r - \mathbf{F}}{E} \quad (2)$$

Bei einer Kraftregelung mit bekannter Elastizität E ergibt die Differenz zwischen Referenzkraft \mathbf{F}_r und Istkraft \mathbf{F} in Richtung des Kraftvektors die Korrektur, aus der die Sensortrajektorie bestimmt wird.

Etwas umgeschrieben ergeben sich aus Gleichung (1) die Fusionsgleichung (3) und die Prädiktionsgleichung (4).

$$\Delta \mathbf{x}_r = \begin{cases} \Delta \mathbf{x}_a - (\mathbf{x}_r - \mathbf{x}_a) & \text{sofern sensierbar} \\ 0 & \text{sonst} \end{cases} \quad (3)$$

$$\mathbf{x}_s = \mathbf{x}_r + \Delta \mathbf{x}_r \quad (4)$$

Dabei versteht sich die Fallunterscheidung in der Fusionsgleichung räumlich, während das vorhandene $\Delta \mathbf{x}_r$ jeweils so lange gültig ist, bis neue Sensorwerte vorliegen. Die Fusionsgleichung kann für die orthogonalen Projektionen verschiedener Sensoren erweitert werden. Dabei ist \mathbf{x}_a die Position, an der die sensierte Regeldifferenz $\Delta \mathbf{x}_a$ gemessen wurde. Die Sensorposition \mathbf{x}_s in Abbildung 5 ist die Summe von Referenzposition und sensierter Abweichung $\Delta \mathbf{x}_r$ der wahren Anordnung vom Referenzfall.

Unter der Annahme gleichbleibender oder sich nur langsam ändernder Abweichungen der Sensorwerte von den erwarteten Werten benötigt unser unser Verfahren im Gegensatz zu leistungsfähigen Realisierungen in der Literatur (siehe z. B. [9, 16]) keine hohe Sensorbandbreite. Im Extremfall reicht eine einmalige Sensormessung, z. B. *eine* Kraftmessung oder *eine* Kameraaufnahme, aufgrund derer die Solltrajektorie in jedem Abtastschritt nach Gleichung (4) aktualisiert und im internen Robotertakt eingeregelt wird. So reicht in vielen Fällen die Integration der Bildverarbeitung durch Gleichung (3) im 50 Hz Takt üblicher CCD-Kameras, damit der *ideale Roboter* die Solltrajektorie \mathbf{x}_d mit hoher Güte abfährt, auch bei hoher Bahngeschwindigkeit.

4 Diskussion

Der Beitrag beschreibt eine Architektur, die „Sensorregelung“ in Roboterregelung und Bahnplanung trennt und aufzeigt, dass in beiden Bereichen leistungsfähige Module existieren. Die zitierten Arbeiten der Autoren weisen in Experimenten an einem elastischen KUKA-Roboter Genauigkeitserhöhungen um bis zu 70 % nach. Der Regelfehler entlang einer durch online Bildverarbeitung sensierten Linie beträgt auch bei voller Robotergeschwindigkeit weniger als einen Millimeter.

Diese Verbesserungen werden, abgesehen von den Sensoren, nur durch Software erreicht. Somit kann das Verfahren auf üblichen Robotersteuerungen ohne Mehrkosten implementiert werden. Nun liegt es an der Industrie, die entwickelten Methoden einzusetzen, um damit die Umrüstzeiten drastisch zu verkürzen oder neue Aufgabenbereiche zu erschließen.

Literatur

- [1] T. Alban and H. Janocha. Dynamic calibration of industrial robots with inertial measurement systems. In *Proc. European Control Conference*, Karlsruhe, Germany, August / Sept. 1999.
- [2] J. Bals and E. Auer. Industrial robot control - in-loop mops feedback-parameter tuning, 1997. <http://www.op.dlr.de/FF-DR-ER/research/irobot/feedback.html>.
- [3] R. Bernhardt. RoboCal. <http://www-robotics.ipk.fhg.de/rc/RoboCal/RoCal-sum-e.html>, 1997. Fraunhofer Institut für Produktionsanlagen und Konstruktionstechnik.
- [4] Y. Chen. Webpages for iterative learning control (ILC) research, 1999. <http://ilc.ee.nus.edu.sg/>.

- [5] P. Drouet, S. Dubowsky, and C. Mavroidis. Compensation of geometric and elastic deflection errors in large manipulators based on experimental measurements: Application to a high accuracy medical manipulator. In J. Lenarčič and M. L. Husty, editors, *Advances in Robot Kinematics: Analysis and Control*. Kluwer Academic Publishers, 1998.
- [6] J. A. Gangloff, M. de Mathelin, and G. Abba. Visual servoing of a 6 DOF manipulator for unknown 3D profile following. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 3236–3242, Detroit, Michigan, May 1999.
- [7] J. A. Gangloff, M. F. de Mathelin, and G. Abba. High performance 6 dof visual servoing using generalized predictive control. In *Proc. European Control Conference*, Karlsruhe, Germany, August / Sept. 1999.
- [8] B. Geyer und H. Rake. Prädiktive Regelung von Roboterantrieben mit linearen Zustandsraummodellen. In *GMA-Kongress „Mess- und Automatisierungstechnik“*, VDI Bericht Nr. 1282, Seiten 367-376, Baden-Baden, Sept. 1996.
- [9] M. Ishikawa, T. Komuro, A. Namiki, and I. Ishi. 1ms sensory-motor fusion system. In *Int. Symp. of Robotics Research (ISRR'99)*, pages 291–296, Snowbird, Utah, Oct. 1999.
- [10] K. Landzettel, B. Brunner, K. Deutrich, G. Schreiber, B.-M. Steinmetz, and G. Hirzinger. DLR's experiments on the ETS VII space robot mission. In *Proc. Int. Conference on Advanced Robotics, ICAR 99*, Tokyo, Japan, Oct. 1999.
- [11] F. Lange. Fast and accurate training of multilayer perceptrons using an extended Kalman filter (EKFNet), Sept. 1995. <http://www.robotic.dlr.de/Friedrich.Lange/>.
- [12] F. Lange and G. Hirzinger. Learning force control with position controlled robots. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 2282–2288, Minneapolis, Minnesota, April 1996.
- [13] F. Lange and G. Hirzinger. Adaptive minimization of the maximal path deviations of industrial robots. In *Proc. European Control Conference*, Karlsruhe, Germany, August / Sept. 1999.
- [14] F. Lange and G. Hirzinger. Learning accurate path control of industrial robots with joint elasticity. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 2084–2089, Detroit, Michigan, May 1999.
- [15] F. Lange, J. Langwald, and G. Hirzinger. Predictive feedforward control for high speed tracking tasks. In *Proc. European Control Conference*, Karlsruhe, Germany, August / Sept. 1999.
- [16] R. Maaß, V. Zahn, M. Dapper, and R. Eckmiller. Hard contact surface tracking for industrial manipulators with (SR) position based force control. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 1481–1486, Detroit, Michigan, May 1999.
- [17] K. L. Moore. Iterative learning control: An expository overview. In B. N. Datta, editor, *Applied and Computational Control Signals, and Circuits, Vol. 1*. Birkhäuser, Boston, 1999.
- [18] W. S. Newman, M. S. Branicky, H. A. Podgurski, S. Chhatpar, L. Huang, J. Swaminathan, and H. Zhang. Force-responsive robotic assembly of transmission components. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 2096–2102, Detroit, Michigan, May 1999.

- [19] M. Otter and S. Pieters. Industrial robot control - optimal motion planning, 1997. <http://www.op.dlr.de/FF-DR-ER/research/irobot/optmp.html>.
- [20] M. Pandit und K. Buchheit. Iterativ lernende Regelung zyklischer Produktionsprozesse. *Automatisierungstechnik at*, 44(1):21–31, 1996.
- [21] J.-X. Shi, A. Albu-Schäffer, and G. Hirzinger. Key issues in the dynamic control of lightweight robots for space and terrestrial applications. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 490–497, Leuven, Belgium, May 1998.
- [22] P. van der Smagt. Feed-forward networks, 1998. <http://www.robotic.dlr.de/Smagt/research/feed-forward/>.
- [23] B. Tondu and S. A. Bazaz. The three-cubic method: An optimal online robot joint trajectory generator under velocity, acceleration, and wandering constraints. *The International Journal of Robotics Research*, 18(9):893–901, 1999.
- [24] A. Verl and M. Schlemmer. Control design engineering - lightweight robot control, 1997. <http://www.op.dlr.de/FF-DR-ER/research/lrobot/lrobot.html>.