

EASY-ROB, Stefan Anton
FH Darmstadt - FB-Informatik, Prof. Dr. Ing. Thomas Horsch

Automatische Programmierung kooperierender Robotersysteme

Zusammenfassung

Mit zunehmendem Automatisierungsgrad wachsen auch die Anforderungen an das Bewegungsvermögen eingesetzter Industrieroboter. Ein Roboter mit sechs Achsen ist heute der Regelfall, wenig verbreitet sind jedoch programmierbare Steuerungssysteme, die es erlauben, auch mehrere Roboter kontinuierlich bahnsynchron kooperieren zu lassen und hierbei die Bewegungsmöglichkeiten der Kinematiken d. h. aller angetriebenen Achsen, speziell auch die der redundanten Achsen, voll auszuschöpfen. Derartige Aufgabenstellungen lassen sich mit herkömmlichen Teach-in-Verfahren nur sehr ineffizient programmieren. Dies bedeutet, dass mit sehr hohem Programmieraufwand Anwenderprogramme erzeugt werden, die das real existierende Bewegungsvermögen der eingesetzten Roboter nur in sehr beschränktem Umfang nutzen. Hilfreich ist hierbei ein entsprechendes Bewegungsplanungsverfahren, das nahezu automatisch, optimiertes Bewegungsverhalten für alle beteiligten Kinematiken erzeugt.

Um komplexe Handhabungs-, Montage- und Bearbeitungsaufgaben effizient automatisieren zu können, verwendet man zunehmend Systeme kontinuierlich kooperierender Kinematiken. Während die technische Realisierung der Bewegungssysteme mit industriell verfügbaren Komponenten durchführbar ist, neigen herkömmliche Programmierverfahren im allgemeinen zu einer erheblichen Einschränkung der erreichbaren Leistungsfähigkeit. Mit Hilfe modellgestützter Bewegungsplanungsverfahren im Rahmen der Programmierung lässt sich die real vorhandene Mobilität kooperierender Kinematiken bestmöglich im Sinne der zu automatisierenden Fertigungsaufgabe nutzen. Hiermit lassen sich die erforderlichen Ausführungszeiten unter Berücksichtigung der mechanischen Beanspruchung der verwendeten Kinematiken erheblich reduzieren und die Qualität des zu fertigenden Produktes erhöhen. Da moderne Robotersteuerungen heute alle wesentlichen Funktionen zum synchronen Betrieb von Zusatzachsen beinhalten, wird es mittelfristig möglich sein, das entwickelte Verfahren auch auf

Technische Änderungen und Verbesserungen sind vorbehalten

breiter Ebene industriell anzuwenden. Der industrielle Einsatz bahnsynchron kooperierender Robotersysteme setzt als Erweiterung der üblichen industriellen Steuerungskonzepte voraus, dass die beteiligten Kinematiken als Roboter und nicht nur als Zusatzachsen interpretiert werden. Mehrere Hersteller industrieller Robotersysteme bieten solche Steuerungssysteme bereits an, die es erlauben, das Bewegungsverhalten von Robotern quasi kontinuierlich zu synchronisieren.

Die im Rahmen dieses Forschungsberichtes dargestellten Ergebnisse ermöglichen eine automatisierte Programmierung kooperierender Robotersysteme. Diese Problemstellung wurde zweistufig realisiert und bewertet. In einer ersten Stufe wurde sie als lineares Gleichungssystem formuliert und dieses System mithilfe der Singulärwertzerlegung gelöst. Aufgrund gewisser Einschränkungen, die im Rahmen dieses Berichts näher erläutert werden, wurde in einem zweiten Schritt diese Problemstellung als Optimierungsproblem formuliert und gelöst. Entsprechende Ergebnisse sind in Form von Animationen abrufbar.

Zielsetzung und Motivation

Die Fähigkeit, herausragende Produkte herzustellen, basiert unter anderem auch auf fortschrittlichen Fertigungsmethoden. Im Automobilbau werden vielfach Roboter eingesetzt, um Fertigungsverfahren mit gleichbleibend hoher Qualität zu realisieren. Gleichzeitig besteht die Forderung, die Fertigungsprozesse zu beschleunigen. Ein möglicher Ansatz besteht darin, mehrere Roboter kooperierend an einem Werkstück einzusetzen. Dieser Ansatz erlaubt einen weitaus flexibleren und effizienteren Einsatz der Roboter. Vereinfacht dargestellt: Dieselben Roboter können schneller, vielseitiger und präziser agieren als bisher. Werkstücke können nun auch während der Weitergabe zur nächsten Montagestation bearbeitet werden, dadurch reduzieren die „intelligenteren“ Maschinen die nichtproduktiven Transportzeiten.

Roboterhersteller (Firmen wie z.B. KUKA, ABB, Fanuc, Motoman, ...) entwickeln derzeit Steuerungen, die in der Lage sind, mehrere Roboter synchronisiert zu bewegen. Die Programmierung solcher kooperierender Roboter ist derzeit noch ungelöst. Traditionell werden sogenannte Programmierhandgeräte (Programmierung durch „Teach-In“) oder Offline Programmiersysteme eingesetzt, um solche Systeme zu programmieren. Diese Verfahren sind interaktiv und bei mehreren kooperierenden System nicht mehr sinnvoll einsetzbar, um in angemessener Zeit zu einer Lösung im Sinn mehrerer Roboterprogramme zu kommen.

Ausgehend von dieser Situation ist die Zielsetzung die prototypische Realsierung eines Verfahrens, das die Programmierung kooperierender Robotersysteme automatisiert. Hierbei erfolgt die Aufgabe prozeß- und werkstückorientiert. Die Angabe der Prozeßorientierung erfolgt durch eine geeignete Angabe von Koordinatensystemen (Frames) entlang der Bearbeitungsbahn, die ggfs. mit einer unterlagerten Werkstückbewegung synchronisiert werden muss.

Industrielle Relevanz

Mit steigendem Automatisierungsgrad treten in der industriellen Praxis vermehrt fertigungstechnische Problemstellungen auf, die zu erhöhten Anforderungen an das Bewegungsvermögen der Roboterkinematiken führen. Eine in diesem Zusammenhang charakteristische Aufgabenklasse umfasst verschiedenste fertigungstechnische Problemstellungen, die sich aufgrund des Bewegungsvermögens einer einzelnen Kinematik nicht oder nur bedingt automatisieren lassen. Dies sind primär Bearbeitungsaufgaben, die das Einhalten einer Vorzugslage des Werkstückes erfordern. Aufgabenspezifisch definierte Vorzugslagen sind unter anderem beim Schweißen in Wannennlage, und beim Wasserstrahl- oder Laserschneiden aus Wirtschaftlichkeits-, Qualitäts- oder Sicherheitsgründen einzuhalten. Aufgabenstellungen dieser Art lassen sich mit Hilfe kontinuierlich kooperierender Kinematiken effizient automatisieren, weil Werkzeug und Werkstück gleichermaßen als Funktion der Zeit während des Bearbeitungsvorganges geführt werden können [Münch 2001].

Anforderungen an die Programmierung

Für ein Verfahren, das die synchrone koordinierte Bewegung mehrerer Roboter ermöglicht, werden folgende Anforderungen definiert.

- (A1) Vermeidung von kinematischen Zwangslagen (Singularitäten / Gelenkansschläge)
- (A2) Berücksichtigung von Prozeßeinschränkungen (z.B. Schweißen in Wannennlage)
- (A3) Berücksichtigung von Prozeßredundanzen (z.B. Schweißen in Wannennlage erlaubt ein Abweichen von Vorzugslagen in gewissen Grenzen)
- (A4) Priorisierung einzelner Achsen (z.B. „Einfrieren“ der Handachsen)
- (A5) Einsetzbarkeit unabhängig von der Anzahl der eingesetzten Robotersysteme
- (A6) Einsetzbarkeit unabhängig von der kinematischen Struktur der eingesetzten Roboter

Die Anforderung A1 beschreibt die Notwendigkeit, die Achsbeschränkungen eines Roboters einzuhalten. Eine generierte Bewegung muss im gültigen Bereich der jeweiligen Achse liegen. Zudem sind Einschränkungen an die mögliche Achsgeschwindigkeit sowie Achsbeschleunigung vorzunehmen. Die Anforderung A2 definiert prozessspezifische Vorgaben bei einer Bearbeitungsaufgabe. Betrachtet man z.B. eine Schweissaufgabe, so muss gewährleistet sein, dass das Schweissgut aufgrund der Gravitation von oben aufgetragen wird, d.h. der Prozess definiert Anforderung zusätzlich an die Orientierung der Bearbeitungsaufgabe bzgl. des Werkstücks (Vorzugslage). Anforderung A3 berücksichtigt Prozessredundanzen. Übertragen auf das Beispiel Schweißen in Wannelage bedeutet dies, dass man eine gewisse Abweichung von der Nominalorientierung zulässt. Das Einhalten der Position ist für die Bearbeitungsqualität wichtig, die Einhaltung der Orientierung kann aber durchaus innerhalb eines „Toleranzkegels“ erfolgen. Über die Anforderung A4 ist man in der Lage, einzelne Achsen zu gewichten, um zum Beispiel redundante Freiheitsgrade auf bestimmte Achsen abzubilden. So kann man durch eine entsprechende Gewichtung der Achsen 1-3 (der sogenannten Hauptachsen bei typischen Roboterkinematiken) eine Bewegung weitestgehend auf die Handachsen abbilden (Achsen 4-6 bei typischen Roboterkinematiken), sofern die Bearbeitungsaufgabe diese Freiheitsgrade zulässt. Die Anforderungen A5 und A6 fordern die Nutzung für (fast) beliebige Roboterkinematiken und ermöglichen somit den breiten Einsatz des Verfahrens.

Voraussetzungen und Grundlagen

Um die weiteren Ausführungen verstehen zu können, sind Grundkenntnisse in Robotik erforderlich. Eine gute Grundlage bieten diverse Lehrbücher in diesem Umfeld ([Weber 2002], [Craig 1989], [Paul 1981]).

Realisierung

In einem ersten Schritt wurde ein bereits existierendes Verfahren [Duelen 1987] implementiert. Es beruht auf einer inkrementellen Betrachtung mit Hilfe einer sogenannten akkumulierten Jakobimatrix.

Im Fall eines Roboters stellt die Jakobimatrix J die Linearisierung der Gleichung der Vorwärtstransformation im Arbeitspunkt q dar. Die Vorwärtstransformation bezeichnet die Abhängigkeit der – in der Regel 6-dimensionalen - kartesischen Koordinaten x des Tool-Center-Points (TCP) von der aktuellen Konfiguration des Roboters, dargestellt in den Gelenkkordinaten

q.

$$x = f(q)$$

Das resultierende lineare Ersatzproblem lautet:

$$\Delta x = J(q)\Delta q$$

In dieser Gleichung bezeichnet man J als Jakobimatrix. Die Komponenten der Jakobimatrix J sind die einzelnen partiellen Ableitungen:

$$J_{i,j}(q) = \frac{\partial f_i}{\partial q_j}$$

Diese Gleichung drückt nun aus, wie sich kleine Änderungen in den Gelenkwinkeln (dq) auf die Position und Orientierung des TCP's (dx) auswirken. In der Regel gibt man kleine kartesische Änderungen vor und ist an den daraus resultierenden Gelenkwinkeländerungen interessiert. Hierzu muss man die Jakobimatrix invertieren:

$$\Delta q = J^{-1}(q)\Delta x$$

Hierbei ist zu beachten, dass es Stellungen des Roboters gibt, die zu einer Jakobimatrix führen, die nicht mehr vollen Rang hat, also insbesondere nicht invertierbar ist. Solche Stellungen nennt man singular. Es empfehlen sich hier „robustere“ Verfahren zur Invertierung wie z.B. die Singulärwertzerlegung (singular value decomposition), ein Verfahren, das eine Art Diagonalisierung für nichtquadratische Matrizen darstellt.

Bei der Problemstellung kooperierender Roboter wird dieses Verfahren verallgemeinert. Folgende Abbildung zeigt zwei kooperierende Roboter. Gedanklich führt der eine Roboter das Werkstück (work object WO) und der andere das Werkzeug (tool object TO)

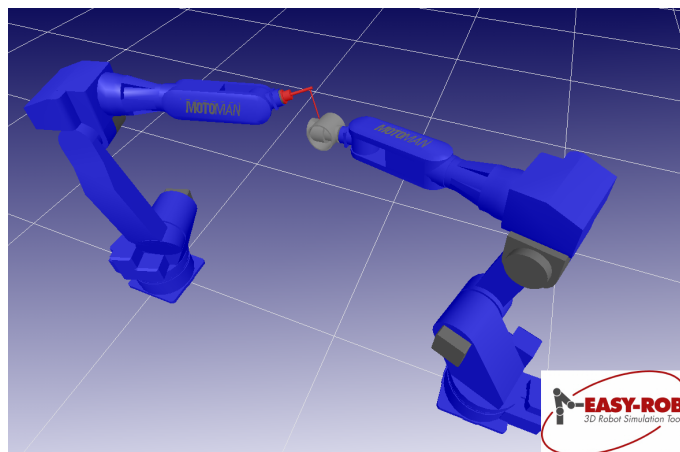


Abbildung 0: Werkstück- und werkzeugführende Kinematik

Zudem definiert eine sogenannte virtuelle Kinematik das Bearbeiten in Vorzugslage. Der Begriff virtuell deutet an, dass diese Kinematik nicht existiert, jedoch eine Möglichkeit darstellt, Vorzugslagen zu modellieren (siehe Abbildung 1). Diese virtuelle Kinematik definiert die Bewegung des Werkstückes. Mit dem Hilfsmittel der virtuellen Kinematik ist man zudem in der Lage Toleranzen zwischen den beteiligten Roboter und dem Werkstück zu modellieren (siehe z.B. Anforderung A3).

Die Modellierung einer Bearbeitungsaufgabe soll beispielhaft anhand einer Aufgabe dargestellt werden, an der zwei Roboter beteiligt sind.

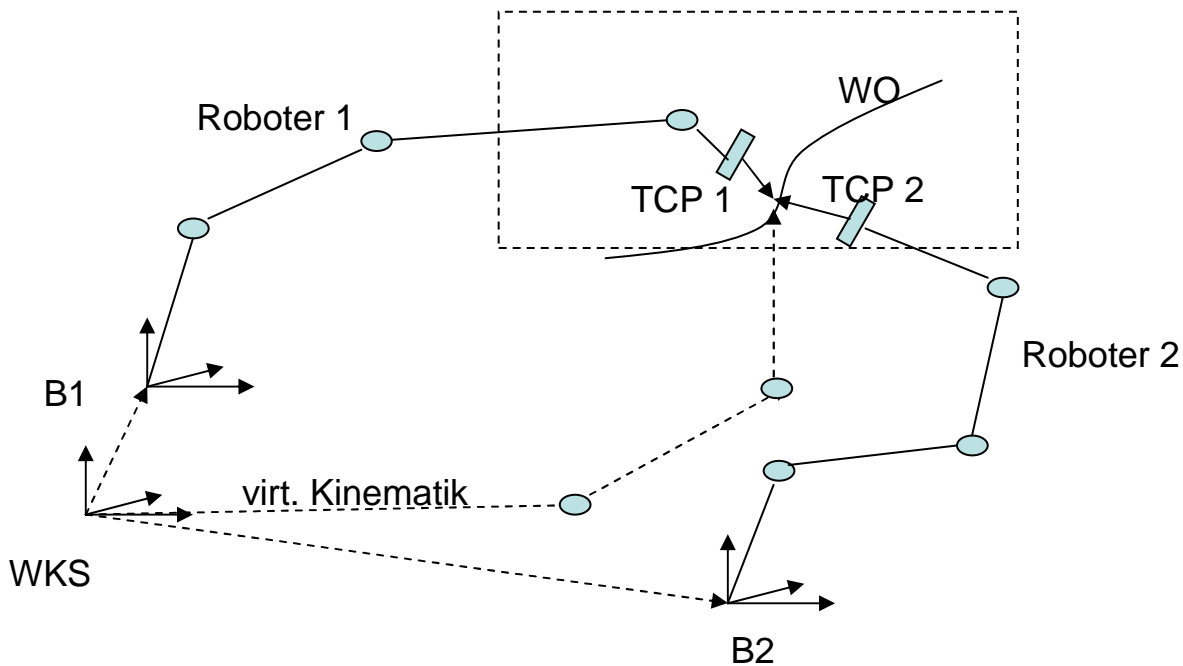


Abbildung 1: Kooperierendes Robotersystem mit virtueller Kinematik (gestrichelt)

- WKS Weltkoordinatensystem
- B1 Basis des ersten Roboters
- R1 Flansch des ersten Roboters
- TO Werkzeugkoordinatensystem (TCP 1)
- B2 Basis des zweiten Roboters
- R2 Flansch des zweiten Roboters
- WO Werkstückkoordinatensystem

Hierzu werden drei (im Allgemeinen $n+1$, wenn n Roboter beteiligt sind) Gleichungen mit Transformationsmatrizen aufgestellt. Die ersten beiden dieser drei Gleichungen stellen die sogenannte Schliessungsbedingung zu dem Werkstück dar. Die dritte Gleichung beschreibt die Bedingung für die Vorzugslage.

Die beiden Schliessungsbedingungen lauten:

$${}_{WKS}^{B1}T \cdot {}_{B1}^{R1}T(q_1) \cdot {}_{R1}^{TO1}T = {}_{WKS}^{WO}T(q_V) \cdot {}_{WO}^{TO1}T(t) \quad (1.1)$$

$${}_{WKS}^{B2}T \cdot {}_{B2}^{R2}T(q_2) \cdot {}_{R2}^{TO2}T = {}_{WKS}^{WO}T(q_V) \cdot {}_{WO}^{TO2}T(t) \quad (1.2)$$

mit

q_1 Konfiguration (Achsstellungen) des ersten Roboters (z.B. werkzeugführend)

q_2 Konfiguration (Achsstellungen) des zweiten Roboters (z.B. werkstückführend)

q_V virtueller Roboter (definiert Bewegung des Werkstücks)

Der linke Teil der Gleichungen (1.1) und (1.2) beschreibt jeweils die Koordinatentransformation vom Weltkoordinatensystem (WKS) über die Roboterbasis und der mechanischen Schnittstelle (Flansch) zum Werkzeug des entsprechenden Roboters. Der rechte Teil beschreibt eine Bewegung des Werkstücks sowie der zu bearbeitenden Prozessbahn. Handelt es sich um einen werkstückführenden Roboter, so ist in der Regel die zweite Transformation auf der rechten Seite konstant und somit nicht zeitabhängig.

Die Vorzugslagenbedingung des Werkstücks lautet:

$${}_{WKS}^{TO}T(q_V) = {}_{WKS}^{TO}T(t) \quad (1.3)$$

Der linke Teil der Gleichung (1.3) beschreibt die Bewegung der Vorzugslage. Formal wird diese Bewegung über eine virtuelle Kinematik modelliert. Es handelt sich hierbei um eine Kinematik mit sechs Freiheitsgraden mit folgender Transformationskette:

$$Transl(x) * Transl(y) * Transl(z) * Rot(x) * Rot(y) * Rot(z)$$

Sechs Freiheitsgrade sind mindestens notwendig um beliebige Lagen (Verschiebung und Orientierung) eines Werkstücks im dreidimensionalen (kartesischen) Raum zu realisieren.

Aus den Gleichungen (1.1) bis (1.3) ergeben sich nun folgende Fehlerfunktionen, wenn man diese Gleichungen als Differenzgleichung formuliert

$$\Delta f_1 = f(q_1) - f(q_v, t) \quad (1.4)$$

$$\Delta f_2 = f(q_2) - f(q_v, t) \quad (1.5)$$

$$\Delta f_v = f(q_v) - f(t) \quad (1.6)$$

Zielsetzung ist es, diese jeweils 6-dimensionalen Fehlervektoren „klein“ im Sinne der Prozesstoleranzen zu halten. In Abhängigkeit der Aufgabenstellung können auch selektiv Fehlerkomponenten einzeln betrachtet werden. Ignoriert man z.B. die Komponenten des Fehlervektors in Gleichung (1.6), so stellt man implizit keine Forderungen an die Bewegung des Werkstücks, d.h. die Lage des Werkstücks ist für eine solche Aufgabe nicht relevant.

Gleichungen (1.4) und (1.5) ermitteln den Fehler bzgl. der Abweichung zwischen der Werkzeuglage (TO) und dem aktuellen Bearbeitungspunkt (WO)

Die Gleichung (1.6) ermittelt den Fehler bzgl. der Abweichung zwischen dem aktuellen Bearbeitungspunkt (WO) und der explizit definierten Vorzugslage.

Die Grundidee des Verfahrens ist es, diese nichtlinearen Fehlerfunktionen zu linearisieren und in einem Gleichungssystem zusammenzufassen. Linearisiert man die Funktionen (1.4) – (1.6) mit Hilfe der jeweiligen Jakobimatrix und definiert folgende Größen:

$$\Delta f_{acc} = \begin{bmatrix} \Delta f_1 \\ \Delta f_2 \\ \Delta f_v \end{bmatrix}, \quad J_{acc} = \begin{bmatrix} J_{R1}(q_1) & 0 & -J_{VK}(q_v) \\ 0 & J_{R2}(q_2) & -J_{VK}(q_v) \\ 0 & 0 & J_{VK}(q_v) \end{bmatrix}, \quad \Delta q_{acc} = \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_v \end{bmatrix} \quad (1.7)$$

so ergibt sich Gleichung (1.8) mit den akkumulierten Größen:

$$\Delta f_{acc} = J_{acc} \Delta q_{acc} \quad (1.8)$$

Die akkumulierte Jakobimatrix J_{acc} koppelt die beiden Schliessungsbedingungen über die virtuelle Kinematik. Diese Matrix hat (im Fall zweier kooperierender Roboter mit jeweils 6 Freiheitsgraden und der virtuellen Kinematik mit 6 Freiheitsgraden) die Dimension 18 x 18.

Sind z.B. vier Roboter (mit jeweils 6 Freiheitsgraden) an einer Kooperation beteiligt, so ist die Dimension der akkumulierten Jakobimatrix 30 x 30. Durch Einsatz von Robotersystemen mit Freiheitsgraden ungleich 6 wird diese Matrix ggfs. nichtquadratisch. Die Lösung solcher Gleichungssysteme erfolgt über die Singulärwertzerlegung.

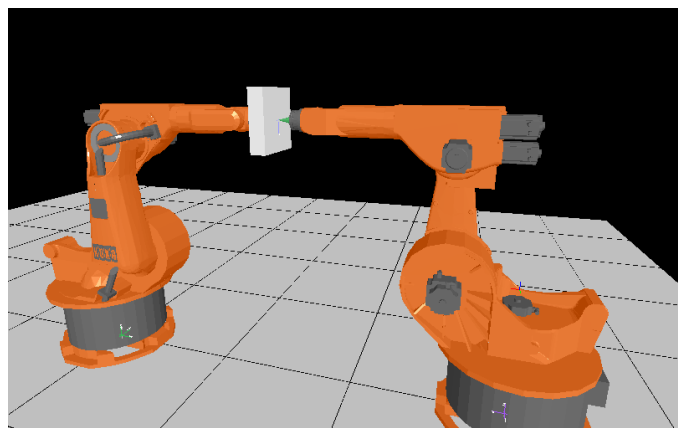
Dieses Verfahren wurde in einer ersten Implementierung realisiert und einer Bewertung unterzogen. Im Rahmen dieser Bewertung wurde erwartungsgemäß festgestellt, dass die gewünschten (oben definierten) Anforderungen allerdings nur zum Teil abgedeckt werden können. Es ist in der Tat nicht möglich, Bedingungen, die mathematisch den Charakter einer Ungleichungsbedingung haben (z.B. das Einhalten von Gelenkansschlägen), zu berücksichtigen.

Daher wurde dieses Verfahren in einer Form weiterentwickelt, so dass die Idee des o.g. Verfahrens weiterhin zu Grunde liegt, zusätzlich alle Anforderungen umsetzt. Dies erfolgte durch die Formulierung der Kooperationsaufgabe als quadratisches Optimierungsproblem [Goldfarb 1983].

Ergebnisse

Das Verfahren wurde anhand von drei Szenarien verifiziert.

Applikation 1:

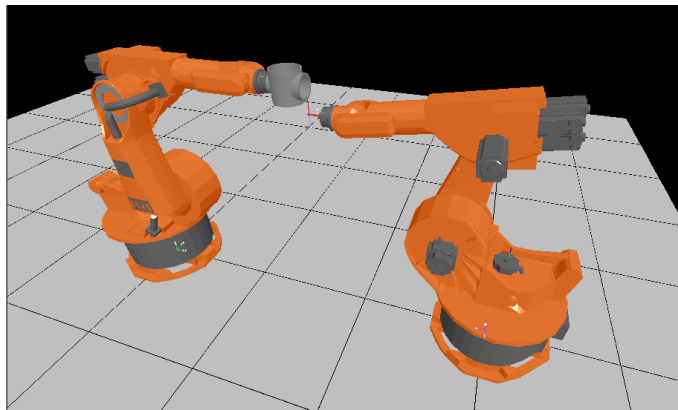


Ein werkzeugführender Roboter (rechts) fährt entlang einer Bahn am Rand des Werkstücks (graue Platte), die von dem werkstückführenden Roboter (links) geführt wird.

Dieses Szenario soll die grundlegenden Aspekte des Verfahrens aufzeigen und dessen Realisierung verifizieren. Insbesondere die Robustheit des Verfahrens in singulären Stellungen soll getestet werden.

Applikation 2:

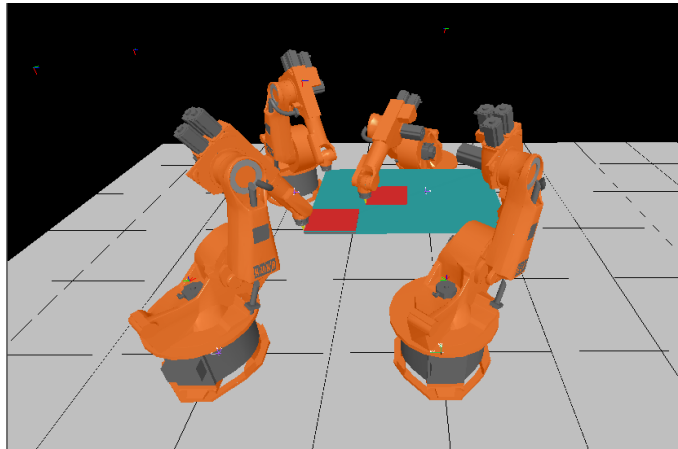
Hierbei kooperieren zwei Roboter: ein werkzeugführender Roboter führt eine Aufgabe auf einem T-Rohr durch, das von einem werkstückführenden Roboter gehalten wird.



Dieses Szenario soll insbesondere das Verfahren bzgl. Aufgabenstellungen überprüfen, wo einzelne Freiheitsgrade Toleranzen haben sowie Vorzugslagen einzuhalten sind (z.B. Wannennlage beim Schweißen).

Applikation 3:

Vier Roboter kooperieren miteinander: zwei werkzeugführende Roboter führen jeweils eine Aufgabe auf einem Werkstück durch, das von zwei werkstückführenden Robotern gehalten und geführt wird.



Dieses Szenario soll insbesondere die Verallgemeinerung des Verfahrens mit mehr als zwei Robotern überprüfen.

Einige Simulationsergebnisse können unter [HORSCH 2006] abgerufen werden. Derzeit liegen folgende Simulationen als AVI und VRML Dateien vor, die z.B. mit dem Windows Media Player oder einem VRML Plugin (z.B. Cortona) für einen Internetbrowser angeschaut werden können. Bei den AVI Dateien empfiehlt es sich, die Wiedergabegeschwindigkeit zu reduzieren.

Werkzeuge

Zur Realisierung des Verfahrens wurden folgende Werkzeuge benutzt:

- Programmierung unter Microsoft Visual Studio C++
- Roboter Simulationssystem EASY-ROB™ und des Application Programming Interfaces API-UserDll zur Einbindung und Verifikation des Verfahrens. <http://www.easy-rob.com>
- Nutzung der Open Source Software MICROB. <http://www.robotique.ireq.ca/microb/en>

Das vorgestellte Verfahren erlaubt es eine automatisierte Programmierung kooperierender Roboter zu realisieren. Es erfüllt die genannten Anforderungen an eine solche Programmierung. Folgende Eigenschaften sind charakteristisch für die Implementierung.

Allgemeiner Ansatz

- unabhängig von der kinematischen Struktur

- serielle Kinematiken werden unterstützt
- basiert nicht auf speziellen kinematischen Beschreibungen (z.B. DH)
- unabhängig von der Anzahl der beteiligten Roboter
- Anforderungen werden abgedeckt
beispielsweise die Gewichtung der Achsen durch modifizieren der Zielfunktion

Ausblick

Der entwickelte Prototyp ist mit modernen Softwareentwicklungsmethoden realisiert. Insbesondere der objektorientierte Ansatz bietet eine gute Wartbarkeit und Erweiterbarkeit. Mit diesem Prototyp können Offlineprogrammiersysteme mit der Fähigkeit ergänzt werden kooperierende Robotersysteme zu programmieren.

Die entwickelte Software wurde in ihrem Kern in verhältnismäßig kurzer Zeit entwickelt. Die Aufwände für die Bereitstellung und Anpassung an benutzte Werkzeuge, das Visualisierungssystem EASY-ROB™, das Optimierungsmodul und die VECTMATH Bibliothek zur Nutzung von Methoden der linearen Algebra, waren nicht vernachlässigbar. Ein unerwartet hoher Aufwand lag im Softwaretest.

Die Robustheit des Verfahrens / der Software hängt von den Anfangsbedingungen ab. Das Optimierungsverfahren minimiert eine quadratische Funktion unter linearen Nebenbedingungen. Der Einsatz eines solchen Optimierungsverfahrens ist durch die Linearisierung des Problems in Form der beschriebenen akkumulierten Jakobimatrix motiviert, d.h. das Verfahren arbeitet mit differenziellen Änderungen, funktioniert also im „Kleinen“. Ist das Problem fehlerhaft beschrieben bzw. konfiguriert, so erzeugt das Verfahren unvorhersagbare Ergebnisse.

Man kann sich verschiedene Ansätze der Erweiterung dieses Verfahrens vorstellen, die das Verfahren (noch) variabler machen. Ein interessanter Punkt wäre zu untersuchen, inwieweit ein nichtlineares Optimierungsverfahren die oben beschriebene Robustheit erhöht.

Der derzeit benutzte Ansatz erlaubt es zudem, weitere bisher nicht betrachtete Anforderungen zu integrieren. Zum Beispiel spielt die Kollisionsfreiheit bei Roboterbewegungen eine grosse Rolle. Das Verfahren kann um diese Anforderung durch Integration weiterer Ungleichungsbedingungen für das Optimierungsproblem ergänzt werden.

Die Ergebnisse dieses Forschungssemesters wurden am 21. Feb. 2006 im Rahmen eines mathematischen Seminars an dem Institut für Angewandte Geometrie der Johannes Kepler Universität Linz vorgestellt und im Rahmen der Messe AUTOMATICA 2006, einer bedeutenden Fachmesse für Robotik und Automation (München), am Stand der Firma EASY-ROB auszustellen.

Literatur

- [Craig 1989] J.J. Craig: Introduction to robotics, mechanics and control, Addison Wesley Verlag, 1989
- [Paul 1981] Paul, R.P.: Robot Manipulators, Mathematics, Programming and Control MIT Press 1981
- [Duelen 1987] G. Duelen: Verteilung der Bewegung auf die Freiheitsgrade des Werkzeugs und der Werkstückaufspannung, DFG Endbericht 1987
- [Goldfarb 1983] D. Goldfarb, A. Idnani: A numerically stable dual method for solving strictly convex quadratic programs. Mathematical Programming 27, (1983), 1-33
- [Horsch 2000] Prof. Thomas Horsch: Introduction to Robotics: „Module Trajectory generation and robot programming“ FH Darmstadt, Summer 2000, <http://www.easy-rob.com/data/LectureRobotics.pdf>
- [Münch, 2001] H. Münch: Effizienter industrieller Einsatz bahnsynchron kooperierender Roboter, Hochschule Magdeburg-Stendal, interner Forschungsbericht
- [Weber 2002] Prof. Wolfgang Weber: FH Darmstadt
Industrieroboter: Methoden der Steuerung und Regelung, Carl Hanser Verlag 2002
- [Horsch 2006] Prof. Thomas Horsch: FH Darmstadt
<http://www.fbi.h-da.de/organisation/personen/horsch-thomas.html>

Autorenangaben:

Dipl. Ing. Stefan Anton
EASY-ROB
Hans-Thoma-Str. 26a
60596 Frankfurt/Main
Tel.: 069 677 24 287
Fax: 069 677 24 320
E-mail: stefan.anton@easy-rob.com

Prof. Dr. Ing. Thomas Horsch
FH Darmstadt - FB Informatik
Schöfferstrasse 8b
64295 Darmstadt
Tel.: 06151 16 8443
Fax.: 06151 16 8935
E-mail: t.horsch@fbi.h-da.de