

The revised version of this preprint appeared in:
K.-H. Hoffmann, T. Lohmann, W. Jäger, H. Schunck (eds.):
Mathematik – Schlüsseltechnologie für die Zukunft (Springer-Verlag, 1997) 551–564.

Parameteridentifikation, Bahnoptimierung und Echtzeitsteuerung von Robotern in der industriellen Anwendung

A. Heim¹, O. von Stryk¹, H. J. Pesch², H. Schöffler³ und K. Scheuer⁴

¹ Mathematisches Institut, Technische Universität München,
E-mail: stryk@mathematik.tu-muenchen.de,
URL: <http://www.mathematik.tu-muenchen.de/~heim/Roboter>

² Institut für Mathematik, Technische Universität Clausthal,
Erzstr. 1, D-38678 Clausthal-Zellerfeld

³ BMW AG, München

⁴ BMW AG, Dingolfing

Abstract. A way for the improvement of roboter based production lines of cars using mathematical optimization methods is outlined. The complete task is subdivided into several related mathematical subproblems, namely the *modeling* of the dynamical system and of the constraints, the *identification* of unknown parameters, the *robot trajectory optimization* by methods of optimal control, and the aspects of real-time control implementation, visualization, and computer simulation.

1 Projektbeschreibung und Ziele

1.1 Problembeschreibung

Bei der Fertigung von Kraftfahrzeugen durchlaufen die Karosserie bzw. Teile davon Produktionsstraßen, die aus mehreren von Robotern bedienten Stationen bestehen (siehe Abb. 1). Hierbei sind beispielsweise pro Auto circa 5000 Schweißpunkte zu setzen. Die Optimierung der Produktionsstraße hinsichtlich der benötigten Taktzeit pro Auto erfolgt derzeit im wesentlichen durch Eingriffe der Programmierer vor Ort während Nacht- oder Wochenendzeiten.

Ein wichtiger Aspekt zur Reduktion der Taktzeit pro Auto ist die Robotersteuerung, beispielsweise beim Anfahren aufeinanderfolgender Schweißpunkte oder bei Handlingsaufgaben. Die Programmierung der Roboter zur Erledigung dieser Aufgaben erfolgt derzeit im wesentlichen durch manuelle Bahnvorgabe (teaching) im Hinblick auf eine möglichst optimale Bewegung der Roboterarme.

1.2 Mathematisierung der Problemstellung

Der Verlauf der Bewegung der Roboterarme hängt von den während der Bewegung angelegten Spannungen in den Gelenkmotoren ab. Die mathematische Beschreibung des dynamischen Systems, das das Bewegungsverhalten eines Industrieroboters beschreibt, geschieht durch sehr komplexe Formeln,

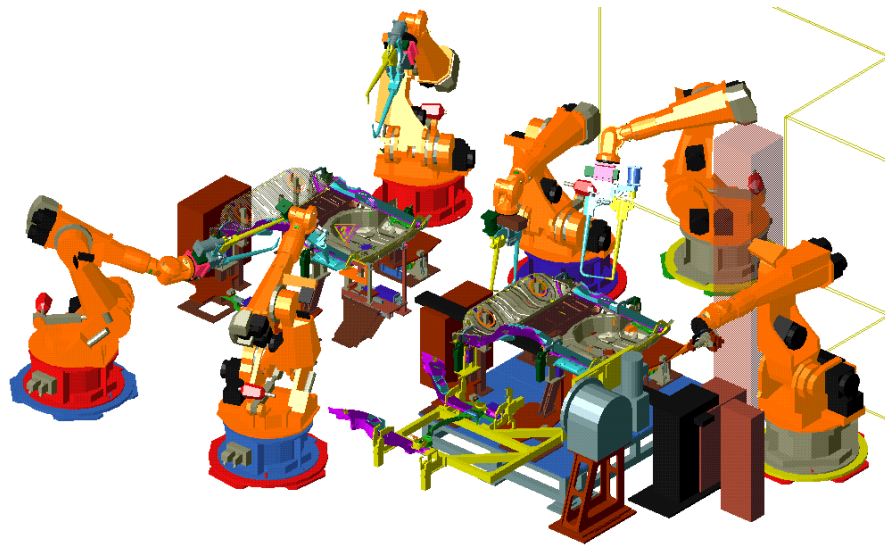


Abb. 1. Ausschnitt einer Fertigungslinie beim Industriepartner.

nichtlineare gewöhnliche Differentialgleichungen. Diese Formeln können mit Hilfe von Computern und speziellen Verfahren der Mechanik (Mehrkörperalgorithmen) aufgestellt werden. Die resultierenden Beziehungen sind in einer Computersprache geschrieben und können ausgedruckt Hunderte von Schreibmaschinenseiten füllen.

Die schnellste Bewegung eines Roboterarmes zwischen zwei vorgegebenen Punkten A und B verläuft selten entlang einer Geraden. Diese zeitoptimale Bewegung kann mit Hilfe der Mathematik, der Theorie der optimalen Steuerung, gefunden werden. Trotz der umfangreichen Formelmengen können die Verläufe der Motorspannungen und Gelenkwinkel mit neuen effizienten numerischen Verfahren berechnet werden, so daß die Zeit für die Bewegung zwischen zwei vorgegebenen Punkten minimal wird. Außerdem können zusätzliche, für die Praxis besonders wichtige Beschränkungen an die optimale Bahn berücksichtigt werden: Es dürfen die Kräfte in den Gelenken gewisse Maximalwerte nicht überschreiten, oder es gibt Hindernisse im Raum, die mit dem Arm umfahren werden müssen.

Richtet man sich nur nach dem Kriterium Zeit, so gibt es zwei gravierende Nachteile. Die zeitoptimalen Bahnen belasten die Gelenke außerordentlich hoch und die beim realen Roboter auftretenden kleinen Abweichungen von den Soll-Werten führen zu Instabilitäten in der zeitoptimalen Bewegung, weil keine Reserven für deren Ausgleich mehr übrig sind. Ist jedoch die schnellstmögliche Zeit für eine Bewegung bekannt, so kann man eine nur 10 (oder 20) % langsamere Zeit als Schranke zur Berechnung einer energiem-

die Roboterbewegungen sehr schnell sind, müssen bei der Bahnoptimierung auch Massenverteilungen, Trägheitsmomente, sowie Zentrifugal-, Coriolis-, Gravitations- und Reibungskräfte berücksichtigt werden. Als Mehrkörpersystem in Minimalkoordinaten wird das *dynamische* Verhalten durch ein System von n Differentialgleichungen in den Gelenkkoordinaten beschrieben (n ist die Anzahl der rotatorischen und translatorischen Robotergelenke):

$$M(q(t)) \ddot{q}(t) = u(t) + h(q(t), \dot{q}(t), t), \quad t \in [0, t_f] \quad (1)$$

bzw. für $n = 3$

$$\begin{pmatrix} M_{11}(q(t)) & M_{12}(q(t)) & M_{13}(q(t)) \\ M_{21}(q(t)) & M_{22}(q(t)) & M_{23}(q(t)) \\ M_{31}(q(t)) & M_{32}(q(t)) & M_{33}(q(t)) \end{pmatrix} \begin{pmatrix} \ddot{q}_1(t) \\ \ddot{q}_2(t) \\ \ddot{q}_3(t) \end{pmatrix} = \begin{pmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{pmatrix} + \begin{pmatrix} h_1(q(t), \dot{q}(t), t) \\ h_2(q(t), \dot{q}(t), t) \\ h_3(q(t), \dot{q}(t), t) \end{pmatrix}. \quad (2)$$

Die Gelenkkoordinaten $q = (q_1(t), q_2(t), q_3(t))^T$ sind die Zustandsvariablen, und die normalisierten Momentensteuerungen $u = (u_1(t), u_2(t), u_3(t))^T$ sind die Steuervariablen. Die Matrix $M(q)$ von Trägheitsmomenten ist positiv definit und symmetrisch. Die durch Coriolis-, Zentrifugal-, Gravitations- und Reibungskräfte bedingten Momente sind in $h(\dot{q}(t), q(t), t)$ enthalten. Zur rechnergestützten Formulierung der Differentialgleichungen stehen effiziente Mehrkörperalgorithmen zur Verfügung [Schi].

Außer technischen Beschränkungen, wie den maximalen Motorspannungen, maximalen Gelenkwinkelgeschwindigkeiten und maximalen Koordinatenverläufen, sind geometrische Beschränkungen zur Kollisionsvermeidung mit der Arbeitszelle oder anderen Objekten im Arbeitsbereich des Roboters bei der Berechnung optimaler Bewegungen unbedingt zu berücksichtigen. Im Rahmen des Projektes wurde hierzu eine besonders effiziente Methode zur Modellierung der Kollisionsvermeidung entwickelt [Sta].

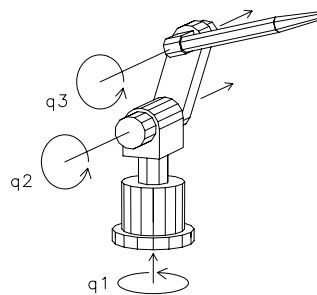


Abb. 3. Robotermodell mit $n = 3$ Freiheitsgraden (rotatorische Gelenke).

2.2 Parameteridentifikation

Um das reale kinematische oder dynamische Verhalten eines konkreten Roboters hinreichend genau beschreiben zu können, müssen Parameter wie geome-

trische und kinematische Grössen, Massen der Teilkörper, Trägheitsparameter, Reibungskoeffizienten u. a. bekannt sein. Unterschiedliche Methoden zur Roboterkalibrierung durch *direkte* oder *indirekte Vermessung* sind verfügbar, um die Genauigkeit kinematischer Modelle zu verbessern [BeAl]. Bei der indirekten Vermessung werden die kinematischen Parameter durch Vergleich von Soll- und Ist-Bahnen berechnet (Identifikation, inverses Problem). Auf geeigneter Modellierung und numerischer nichtlinearer Ausgleichsrechnung beruhende Verfahren zur indirekten Vermessung kinematischer Parameter sind mit dem Industriepartner in den Vorarbeiten zum laufenden Projekt bereits entwickelt und implementiert worden [Mü].

Die Bestimmung dynamischer Parameter ist jedoch problematischer. Zur direkten Vermessung dynamischer Parameter, wie den Massenmittelpunkten und den Trägheitstensenoren, ist die Zerlegung des Roboters und experimentelle Vermessung jedes einzelnen Teilkörpers mit relativ aufwendigen Meßapparaturen notwendig [Tü]. Für einen industriellen Einsatz ist es jedoch wünschenswert, die fehlenden (oder nicht ausreichend genau bekannten) Daten des Roboters möglichst am Arbeitsplatz und ohne die Notwendigkeit einer Zerlegung des gesamten Roboters bestimmen zu können.

Zur Identifikation dynamischer Parameter durch indirekte Vermessung werden geeignete Testbahnen mit dem Roboter gefahren. Die unbekannt Modellparameter werden dann so angepaßt, daß die computersimulierten Bahnen möglichst gut mit den gemessenen Bahnverläufen übereinstimmen.

Ein erster Ansatz hierzu nützt aus, daß die zu bestimmenden Trägheitsparameter linear in den Elementen der Massenmatrix M in Gleichung (1) auftreten. Wenn man nun annimmt, daß die Steuergröße $u(t)$ (input) sowie alle Zustandsgrößen $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ (output) für eine Anzahl von Meßzeitpunkten $t = t_1, \dots, t_m$ gemessen werden können, so erhält man damit aus Gleichung (1) ein überbestimmtes lineares Gleichungssystem für die unbekannt Trägheitsparameter [Hö]. Mit akzeptablem Aufwand können in der Praxis jedoch weder die zweiten noch die ersten Ableitungen von $q(t)$ gemessen werden. Daher müssen als Ersatz „künstliche“ Meßwerte für die Ableitungen von einer Reihe von Meßwerten $\tilde{q}(t_i)$, $i = 1, \dots, m$ bestimmt werden, beispielsweise durch Differenzenapproximationen erster und zweiter Ordnung. Zusätzliche Schwierigkeiten werden dadurch verursacht, daß die Messungen $\tilde{q}(t_i)$ mit Meßfehlern behaftet sind [Hö].

Andererseits können konsistente Werte für $\dot{q}(t)$ und $\ddot{q}(t)$ auch allein aus Messungen von $q(t)$ und mit Hilfe numerischer Integration der Differentialgleichungen (1) berechnet werden. Die Trägheitsparameter erhält man dann als Lösung eines nichtlinearen beschränkten Ausgleichsproblems der Form:

Man minimiere ℓ_2 hinsichtlich der unbekannt Parameter c und den (unbekannt) Anfangswerten $q(0)$, $\dot{q}(0)$

$$\ell_2(c, q(0), \dot{q}(0)) = \sum_{j=1}^m \sum_{i=1}^n (\tilde{q}_i(t_j) - q_i(t_j))^2 / \omega_{i,j}^2 \quad (\omega_{i,j} = \text{const}), (3)$$

$$\text{wobei} \quad M(q(t), c) \ddot{q}(t) = u(t) + h(q(t), \dot{q}(t), t, c), \quad t \in [0, t_f] \quad (4)$$

$$\text{und } 0 = r(q(0), q(t_f), c). \quad (5)$$

D. h. in Gleichung (3) ist $q(t)$ die (numerische) Lösung der von den unbekannt-ten Parametern c abhängenden Differentialgleichungen, die das dynamische Verhalten des Roboters beschreiben, mit optional vorgegebenen Anfangs-oder Endwerten (Gleichung (5)). Die numerische Lösung der resultierenden gleichungsbeschränkten nichtlinearen Ausgleichsprobleme erfolgt mit verallgemeinerten Gauß-Newton Verfahren oder angepassten Verfahren der Sequentiellen Quadratischen Programmierung (SQP) [BoEiSchl], [GiMuSaWr], [He]. Dazu ist eine effiziente und zuverlässige Berechnung der Gradienten $(\partial q/\partial c)(t)$, $(\partial q/\partial q(0))(t)$ erforderlich. Mehrere Ansätze zur Gradientenberechnung wurden entwickelt, implementiert und verglichen [BuHiKi], [He].

Bemerkungen: Es kann Trägheitsparameter geben, die konstruktionsbedingt durch indirekte Vermessung nicht bestimmt werden können (z. B. die Trägheitsmomente bzgl. der horizontalen Rotationsachsen im Basisgelenk 1 von Abb. 3). Andere Parameter wiederum sind möglicherweise nur bis auf einen gemeinsamen homogenisierenden Faktor eindeutig bestimmbar.

Beispiel: In Abb. 4 ist ein Vergleich zwischen Computersimulation und Experiment für die im nächsten Abschnitt beschriebene verbrauchsmi- nimal- e Bahn aus Abb. 6 für einen Roboter vom Typ Manutec r3 wiedergegeben. Beim Basisgelenk gibt es keinen sichtbaren Unterschied zwischen dem berechneten und experimentellen Verlauf des Gelenkwinkels $q_1(t)$. Die Unterschiede zwischen der berechneten und tatsächlich benötigten Steuerung $u_1(t)$ sind im Mittelteil der Bewegung auf die im Simulationsmodell [OtTü] nicht bertück- sichtigte Gleitreibung und am Anfang und am Ende der Bewegung zusätz- lich auf Elastizitäten in den Getrieben zurückzuführen. Zur Verbesserung des Modells werden die drei Koeffizienten $c_{F,i}$, $i = 1, 2, 3$ eines Coulomb- Reibungsansatzes

$$R_i = c_{F,i} \arctan(100 \dot{q}_i(t))/(\pi/2), \quad i = 1, 2, 3 \quad (6)$$

durch Parameteridentifikation aus 400 Messungen berechnet.

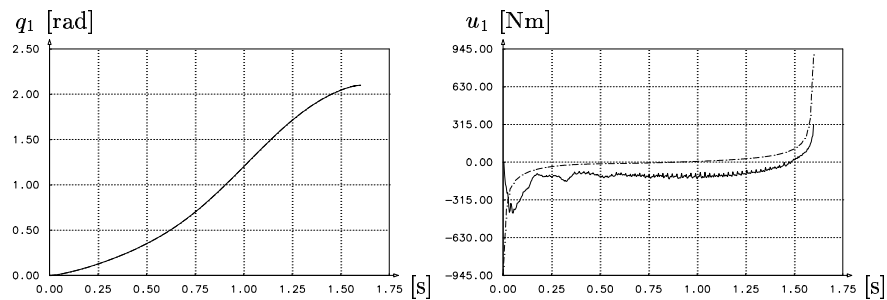


Abb. 4. Verlauf von Winkel q_1 und Steuerung u_1 des Basisgelenks für die ver- brauchsminimale Steuerung (Abb. 6) in Simulation (---) und Experiment (—).

2.3 Bahnoptimierung

Nachdem die dynamischen Gleichungen in geeigneter Weise bestimmt wurden, kann die Optimierung der Bewegungen untersucht werden. Zur Berechnung der optimalen Steuerung $u(t)$ und des zugehörigen optimalen Zustands- und Geschwindigkeitsverlaufes $q(t)$, $\dot{q}(t)$ ist ein geeignetes Funktional $J[u, t_f]$ hinsichtlich der Steuervariablen u und der ggf. freien Verfahrzeit t_f zu minimieren (optimale Steuerung). Dabei sind die dynamischen Gleichungen (1), evtl. vorgeschriebene Anfangs- und Endbedingungen ($q(0) = q_0$, $q(t_f) = q_f$, etc.) sowie weitere Beschränkungen der Form $g(q(t), \dot{q}(t)) \geq 0$ einzuhalten.

Unterschiedliche Aufgaben erfordern unterschiedliche Gütekriterien J für optimale Bahnen. Wenn, wie bei Punktschweiß- oder Handlingsaufgaben, nur Anfangs- und Endstellung der Bewegung vorgegeben sind, sind typischerweise zeitoptimale Punkt-zu-Punkt Bahnen gesucht

$$J_1[u, t_f] = t_f \rightarrow \min . \quad (7)$$

Bei Klebevorgängen ist es andererseits erforderlich, während der gesamten Bewegung eine vorgeschriebene Bahn optimal abzufahren.

Selbst für komplizierteste Robotermodelle kann leicht mit den Mitteln der Optimalsteuerungstheorie nachgewiesen werden, daß bei zeitminimaler Steuerung immer mindestens ein Motor an seinem Limit operieren muß. Andererseits zeigen die numerischen Ergebnisse, daß zeitminimale Punkt-zu-Punkt Bewegungen meist eine unerwartete Schaltstruktur mit vielen Umschaltungen zwischen maximalen Antriebs- und Bremsmomenten besitzen (sogenannte bang-bang Steuerungen) und dabei enorme Belastungen auf die Gelenke ausüben. Hier bieten schnelle, energieminimierende Bewegungskriterien wie

$$J_2[u] = \int_0^{t_f} \left(\sum_{i=1}^n u_i^2(t) \right) dt \rightarrow \min , \quad (8)$$

die bei geeignet vorgeschriebener Endzeit t_f nur wenig langsamere Bewegungen als die zeitminimale liefern, einen guten Kompromiß zwischen Verfahrzeit und Verschleiß [vS], [vSSch]. Die vorab berechnete theoretisch schnellstmögliche Verfahrzeit dient dabei als Schranke in der Berechnung einer z. B. 20 % langsameren, energieminimalen Bewegung.

Das Bahnoptimierungsverfahren muß eine problemlose Optimierung für komplexe Roboterdynamiken, sowie sehr unterschiedliche Gütekriterien und Beschränkungen ermöglichen. Hierzu wurde ein sogenanntes direktes Kollokationsverfahren entwickelt und implementiert [vS]. Die Methode beruht auf einer Diskretisierung der Zustandsvariablen durch stückweise kubische Splinefunktionen, die die Differentialgleichungen an den Gitterpunkten der Diskretisierung und den dazwischen liegenden Mittelpunkten erfüllen (Kollokation). Die Steuerungen werden durch stückweise lineare Funktionen approximiert. Durch diese Diskretisierung wird das optimale Steuerungsproblem in ein endlich-dimensionales nichtlineares beschränktes Optimierungsproblem

übergeführt, welches mit SQP-Verfahren gelöst werden kann [GiMuSaWr]. Durch diesen direkten Lösungsansatz kann eine optimale Steuerung näherungsweise berechnet werden, ohne mit den oft schwierigen notwendigen Bedingungen der Optimalsteuerungstheorie (adjungierte Differentialgleichungen etc.) umgehen zu müssen (indirekter Ansatz).

Verglichen mit anderen numerischen Methoden der optimalen Steuerung ist dieses direkte Kollokationsverfahren relativ einfach zu verwenden (da Kenntnisse der Theorie optimaler Steuerungen nicht notwendigerweise zur Bedienung erforderlich sind), robust (da sehr wenig Information über die Lösung a priori benötigt wird) und zuverlässig (da die erreichbaren Genauigkeiten für die Roboterbahnoptimierung völlig ausreichend sind) [vS].

Während die beschriebene Kollokationsmethode allgemein zur numerischen Lösung optimaler Steuerungsprobleme eingesetzt werden kann, wurde im Projekt eine weitere Diskretisierung entwickelt und implementiert, die zusätzlich die spezielle Struktur der Roboterdynamik (1) (Mehrkörpersystem in Minimalkoordinaten) ausnützt. Dabei wird die Transformation des semi-impliziten Systems von n Differentialgleichungen zweiter Ordnung (1) auf die Standardform eines Systems erster Ordnung mit $2n$ Differentialgleichungen ($\dot{q}(t) = v(t)$, $\dot{v}(t) = M^{-1}(q(t))(u(t) + h(q(t), v(t), t))$) vermieden. Im Vergleich zu Standardverfahren führt die neue Diskretisierung zu sehr viel kleineren nichtlinearen Optimierungsproblemen. In zusätzlicher Kombination mit Sparse-Optimierungsmethoden können die Berechnungszeiten so um einen Faktor im Bereich von 10 bis 20 verkürzt werden [OlvS]. Die optimale Bahnplanung ist damit im Bereich von etwa einer Minute auf Standardrechnern (PCs oder Workstations) durchführbar geworden.

Beispiel: Schnelle Punkt-zu-Punkt Bahnen von A nach B werden für einen Roboter vom Typ Manutec r3 [OtTü] in drei Freiheitsgraden ($n = 3$) und bei einer Last von 0 kg untersucht:

$$A : q(0) = \begin{pmatrix} 0.0 \\ 1.3 \\ 0.0 \end{pmatrix}, \quad B : q(t_f) = \begin{pmatrix} 2.1 \\ -0.6 \\ 0.0 \end{pmatrix}, \quad \dot{q}(0) = 0, \quad \dot{q}(t_f) = 0. \quad (9)$$

Achtzehn technische Beschränkungen

$$|q_i(t)| \leq q_{i,\max}, \quad |\dot{q}_i(t)| \leq \dot{q}_{i,\max}, \quad |u_i(t)| \leq u_{i,\max}, \quad i = 1, 2, 3 \quad (10)$$

müssen bei der Planung optimaler Bahnen berücksichtigt werden [OtTü], [vS], [vSSch]. Die berechnete zeitoptimale Bahn mit $t_{f,\min} = 1.321$ s ist in Abb. 5 dargestellt. Danach wird die verbrauchsminimale Bahn (vgl. [PfRe])

$$J_3[u] = \int_0^{t_f} \sum_{i=1}^3 (\dot{q}_i(t) u_i(t))^2 dt \rightarrow \min \quad (11)$$

für eine feste, vorgeschriebene Endzeit von 1.600 s berechnet, welche 21 % langsamer ist (Abb. 6). Das minimale Integralkriterium ist $J_{3,\min} = 6.136$. Der entsprechende Wert für die zeitminimale Bewegung liegt bei 306.4.

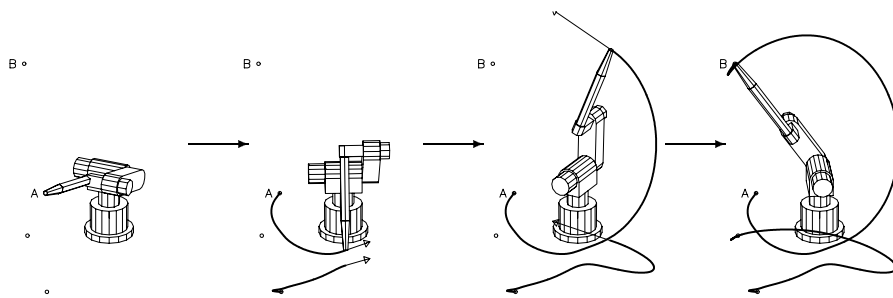


Abb. 5. Zeitminimale Bewegung von A nach B in $t_f = 1.321$ s.

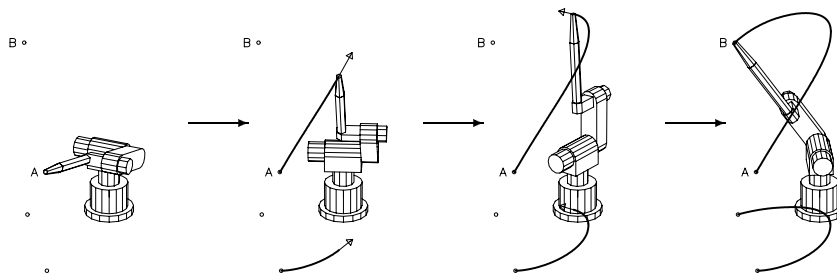


Abb. 6. Verbrauchsminimale Bewegung von A nach B in $t_f = 1.600$ s.

2.4 Echtzeitsteuerung

Die berechneten Bahnen müssen unter den Echtzeitbedingungen des realen Systems umsetzbar sein. D. h. die Einflüsse von Schleppfehlern, eine beschränkte maximale Stellgliedgeschwindigkeit (d. h. $|\dot{u}_i(t)| \leq \dot{u}_{i,\max} = \text{const}$) und Überschießen des Armes am Ende der (zeitminimalen) Bewegung müssen bereits in der Simulation geeignet berücksichtigt werden. Dies kann durch zusätzliche Nebenbedingungen in der Bahnoptimierung erfolgen.

Zur numerischen Approximation von optimalen Echtzeitsteuerungen wurden zwei, im Prinzip auch auf die Roboterbahnoptimierung übertragbare Methoden entwickelt [Br], [PeGaMiBr]. Bei beiden Methoden wird zunächst ein Bündel von optimalen Lösungen berechnet, so daß der z. B. für eine spezielle Roboterbewegung relevante Teil des Zustandsraumes durch optimale Trajektorien hinreichend dicht ausgefüllt ist. Längs dieser Trajektorien sind dann die optimalen Steuerungen als sogenannte Open-Loop-Steuerungen bekannt. Aus den Informationen der Open-Loop-Lösungen (Steuerungen, Zustände, adjungierte Variablen etc.) werden dann die Rückkopplungssteuerungen synthetisiert. Zur Interpolation oder Approximation der Rückkopplungssteuerungen aus den Daten der Open-Loop-Steuerungen längs der Trajektorien des Bündels bieten sich verschiedene Möglichkeiten an, z. B. mit Hilfe der

linearen oder nichtlinearen Ausgleichsrechnung, mit global geglätteten Taylorreihen oder mit Hilfe neuronaler Netze. Das Training dieser parametrisierten Ansätze, d.h. die Optimierung der Parameter in den Ansätzen, erfolgt dabei aus den Daten längs der vorab berechneten Open-Loop-Trajektorien. Dadurch gelingt es, den Rechenaufwand für die Online-Berechnungen zu minimieren. Zusätzlich kann man Schwankungen von Modellparametern, sofern deren Schwankungsbreite bekannt ist, als Steuerungen eines antagonistischen Gegenspielers in einem Null-Summen-Differentialspiel modellieren. Die im Sinne der Differentialspieltheorie optimalen Lösungen stellen den schlechtest möglichen Fall im Sinne der zu optimierenden Zielgröße dar, der bei Schwankungen der Parameter eintreten kann. Eine Bereitstellung hinreichend vieler Open-Loop-Lösungen ermöglicht dann wieder die Synthese der Rückkopplungssteuerungen aus den Daten über die Open-Loop-Steuerungen. In der Arbeit [PeGaMiBr] werden die Rückkopplungssteuerungen mit Hilfe neuronaler Netze synthetisiert und anhand eines einfachen Differentialspiels exemplarisch angewendet. In der Arbeit [Br] werden Luftdichteschwankungen während des atmosphärischen Wiedereintritts eines Space-Shuttles als Steuerungen eines Gegenspielers in einem Differentialspiel aufgefaßt und die fast-optimalen Rückkopplungssteuerungen mit Hilfe global geglätteter Taylorreihen approximiert. Die erforderlichen Rechenzeiten für die Online-Rechnungen sind sehr gering. Der Nachweis, daß diese Methoden auch für die Roboterbahnoptimierung geeignet sind, muß aber noch geführt werden.

2.5 Visualisierung und Computersimulation

Um die komplexen Arbeitsvorgänge bei Industrierobotern besser planen und die Ergebnisse der Bahnoptimierung verifizieren und validieren zu können, sowie zur Erhöhung der Akzeptanz der Ergebnisse bei den Anwendern ist die Visualisierung der berechneten optimalen Bahnen unbedingt erforderlich.

In den letzten Jahren wurden allgemein durch die Entwicklung preiswerter und leistungsfähiger Rechner die Voraussetzungen für realitätsnahe Animationen in der Computergrafik geschaffen. Mittlerweile können sogar auf gängigen PCs Computerbilder mit aufwendigen Beleuchtungsmodellen erstellt werden, wo noch vor kurzem mit einfachen Drahtgittermodellen, ohne Beleuchtungseffekte und ohne Berechnung verdeckter Flächen gearbeitet wurde. Dabei ist es sogar ohne Einsatz zusätzlicher Grafikkhardware möglich, mehrere Bilder pro Sekunde darzustellen und damit fließende Animationen zu ermöglichen. Mit zusätzlicher Hardware lassen sich bei Modellen von mittlerem Komplexitätsgrad, wie sie für Roboter verwendet werden, sogar Animationen mit zwanzig bis fünfzig Bildern pro Sekunde erreichen.

Diese Leistungssteigerung der Rechner hat sich in gleichem Maße auch auf das Gebiet der Computersimulation ausgewirkt. In der Simulation von Roboterbewegungen muß man für die Vorausberechnung des Bewegungsverhaltens die zugehörigen Differentialgleichungen (1) numerisch integrieren. Obwohl diese Gleichungen ausgesprochen kompliziert sind, ist mit heutigen Rech-

nen der Zeitaufwand für die Simulation vergleichsweise gering. Innerhalb von Sekunden ist es möglich, das Bewegungsverhalten des Roboters über den Zeitraum von mehreren Stunden vorherzusagen (ohne Optimierung oder Parameteridentifizierung). Im Vergleich dazu ist der Zeitaufwand, eine vorhergesagte Stellung des Roboters mit Computergrafik zu visualisieren, sehr viel größer. Computersimulation und Visualisierung liefern zusammen ein unersetzliches Werkzeug für die Bahnplanung beim Einsatz von Industrierobotern. So können vor dem tatsächlichen Einsatz der Roboter in der Fertigung die Bewegungsabläufe im Detail studiert und überprüft werden. Durch die grafische Darstellung ist es dann leicht möglich, Kollisionen von Robotern untereinander oder mit Werkstücken zu erkennen und diese wiederum in der Bahnplanung zu berücksichtigen und abzuwenden.

In Abb. 7 ist eine im Rahmen des Projektes entwickelte grafische Benutzeroberfläche dargestellt, bei der Simulation von Roboterbewegung und dreidimensionale Visualisierung in einem interaktiven Programm verbunden sind. Im Fenster links oben sieht man den Verlauf einer zeitminimalen Roboterbewegung, rechts unten das Kontrollfenster zur interaktiven Motorsteuerung und im Fenster rechts oben den zeitlichen Verlauf der Gelenkwinkel. Die gleichzeitige Berechnung der Roboterbewegung und der animierten Grafik erfolgt in Echtzeit und kann auf mehreren Rechnern verteilt ablaufen.

2.6 Software-Engineering

Die numerischen Verfahren zur Bahnoptimierung und Parameteridentifizierung werden an der Hochschule entwickelt. Sie sind ursprünglich als eigenständige, an der mathematischen Problembeschreibung orientierte Algorithmen angelegt, die unabhängig von anderer Software eingesetzt werden können. Als durchschnittlicher Benutzer der Programme wird in der Regel ein Fachexperte vorausgesetzt. Vom industriellen Anwender kann jedoch weder das Fachexpertenwissen in Mathematik noch die an der abstrakten mathematischen Problemformulierung orientierte Bedienung der Verfahren verlangt werden. Die mathematischen Optimierungsverfahren werden daher über eine gemeinsam definierte Schnittstelle, die mathematische Details vor dem Anwender verbirgt, mit den beim Industriepartner eingesetzten Roboter-CAD-Systemen verbunden. Dabei stellt sich die Frage, wie eine sinnvolle Anbindung der Optimierungs- und Parameteridentifizierungsalgorithmen an die Roboter-Software erfolgen kann.

Einer direkten Implementierung der mathematischen Verfahren *innerhalb* der derzeit beim Industriepartner eingesetzten Roboter-CAD-Systeme ist eine Verbindung über eine wohldefinierte *Schnittstelle* vorzuziehen. Zum einen kann das Roboter-CAD-System so bei Bedarf durch ein anderes Produkt ersetzt werden, ohne daß die Optimierungsprogramme mitersetzt werden müssen. Zum anderen ist die Pflege und Erneuerung der Optimierungsverfahren, wenn beispielsweise neue Diskretisierungen oder Sparse-Optimierungsverfahren entwickelt worden sind, nur dann bei vernünftigem Aufwand unter

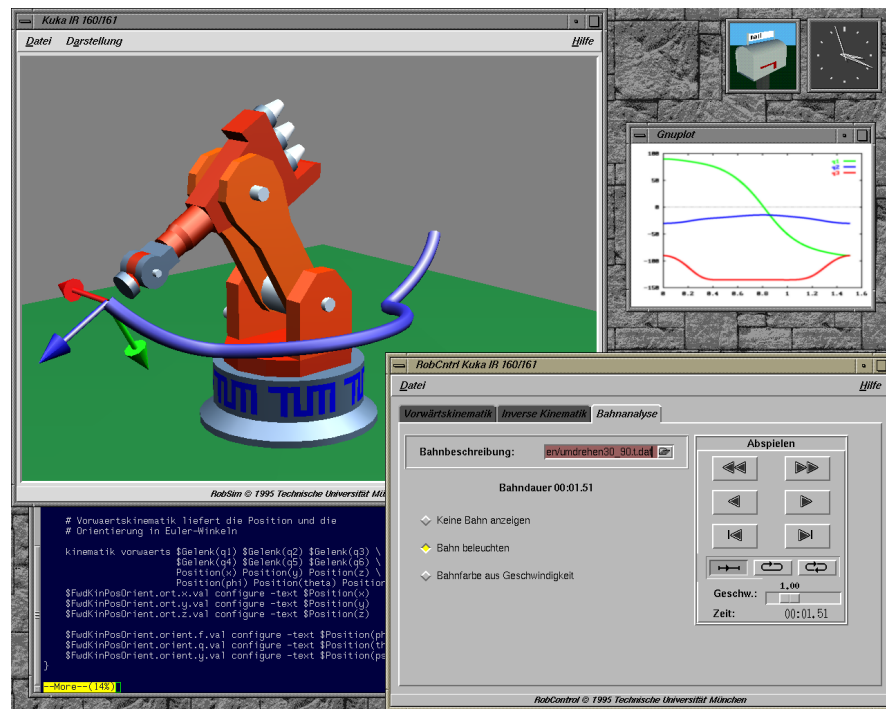


Abb. 7. Benutzeroberfläche der Testumgebung für dynamische Robotermodelle.

den Möglichkeiten der Hochschule durchführbar, wenn die beim Industriepartner eingesetzte Implementierung im wesentlichen mit der universitären übereinstimmt.

Daher wird ein modulares Konzept realisiert, bei dem die Software beider Projektpartner weitgehend ihre Eigenständigkeit beibehalten kann. Zur Steuerung der einzelnen Programmteile wird das Programmiersystem Tcl/Tk [Ou] eingesetzt. Dieses hat sich in den letzten Jahren auf UNIX-Systemen unter X-Windows sowohl im universitären als auch im industriellen Bereich als mächtiges Werkzeug zur Erstellung grafischer Benutzeroberflächen etabliert. Tcl (Tool command language) ist eine Skriptsprache, ähnlich zu Shell-Skripten, die vom Benutzer beliebig um eigene Funktionen erweitert werden kann. Man koppelt diese eigenen Funktionen als Unterprogramme in der Programmiersprache C an das System an. Durch Tk (Toolkit) wird die Skriptsprache um eine Sammlung von Elementen für grafische Benutzeroberflächen, wie Knöpfe, Texteingabefelder etc. erweitert. Die Funktionalität von auf diesem System beruhenden Programmen ist in mehrere Module getrennt, welche als Funktionen der Skriptsprache zur Verfügung stehen. Das Zusammenspiel der einzelnen Programmteile untereinander und die Interaktion mit dem Benutzer, der über die grafische Oberfläche eingreift, wird

dann durch Tcl-Skripten geregelt. In diesen Skripten wird der Ablauf des Programms ereignisorientiert beschrieben. Etwa so: „Drückt der Anwender auf diesen Knopf, dann führe einen Iterationsschritt aus und teile das Ergebnis der Anzeige mit“. Diese abstrahierte Beschreibung der Interaktion einzelner Programmteile und des Benutzers bildet eine ideale Plattform zum Anknüpfen unterschiedlicher Softwarekomponenten. Die gemeinsame Schnittstelle ist dabei die Anbindung an die Skriptsprache Tcl.

3 Fazit und Ausblick

Numerische Methoden der optimalen Steuerung und Parameteridentifizierung in gewöhnlichen Differentialgleichungen werden zur Bahnoptimierung bei Industrierobotern eingesetzt. Dieses Vorgehen hat gegenüber den herkömmlichen Bahnplanungsmethoden den Vorteil, daß die resultierenden Roboterbewegungen am besten im Sinne der verwendeten Gütekriterien sind. Bessere Bahnen gibt es nicht. Ein Nachteil ist die Notwendigkeit einer geeigneten Modellierung der Roboterdynamik und Nebenbedingungen. Zur Bestimmung dynamischer Roboterparameter, die der direkten Messung kaum zugänglich sind, wurden Verfahren zur indirekten Vermessung durch modellgestützte Parameteridentifizierung in Differentialgleichungen entwickelt. Zur robusten und effizienten Berechnung der optimalen Roboterbahnen wurden spezielle numerische Verfahren der optimalen Steuerung entwickelt und implementiert. Die numerischen Verfahren werden über Schnittstellen mit den entsprechenden Planungssystemen beim Industriepartner verbunden. Die bisher an exemplarischen Problemstellungen erzielten Ergebnisse werden im weiteren Verlauf des Projektes an den Einsatzorten erprobt werden.

Danksagung: Die Autoren danken Prof. R. Bulirsch für die wertvolle Unterstützung in diesem Projekt.

Literatur

- [BeAl] Bernhardt, R., Albright, S. L. (Hrsg.): Robot Calibration. (London: Chapman & Hall, 1994).
- [BoEiSchl] Bock, H. G., Eich, E., Schlöder, J. P.: Numerical solution of constrained least squares boundary value problems in differential-algebraic equations. In: K. Strehmel (Hrsg.): Numerical Treatment of Differential Equations, Proc. of the NUMDIFF-4 Conference, Halle-Wittenberg, 1987, Teubner Texte zur Mathematik **104** (1988) 269-280
- [Br] Breitner, M. H.: Robust optimale Rückkopplungssteuerungen gegen unvorhersehbare Einflüsse: Differentialspielansatz, numerische Berechnung und Echtzeitapproximation. Dissertation, Technische Universität Clausthal (1995)
- [BuHiKi] Buchauer, O., Hiltmann, P., Kiehl, M.: Sensitivity analysis of initial-value problems with application to shooting techniques. Numerische Mathematik **67** (1994) 151-159

- [BulKr] Bulirsch, R., Kraft, D. (Hrsg.): Computational Optimal Control. International Series of Numerical Mathematics **115** (Birkhäuser, 1994)
- [GiMuSaWr] Gill, P. E., Murray, W., Saunders, M. A., Wright, M. H.: User's guide for NPSOL (Version 4.0). Report SOL 86-2, Dep. of Operations Research, Stanford University, California, USA (1986)
- [He] Heim, A.: Parameteridentifizierung in differential-algebraischen Gleichungssystemen. Diplomarbeit, Mathematisches Institut, Technische Universität München (1992)
- [Hö] Hölzl, J.: Modellierung, Identifikation und Simulation der Dynamik von Industrierobotern. Fortschritt-Berichte VDI, Reihe 8, Nr. 372 (Düsseldorf: VDI-Verlag, 1994)
- [Mü] Münch, A.: Parameteridentifikation bei Robotern: Modellierung und Numerik der indirekten Vermessung. Diplomarbeit, Mathematisches Institut, Technische Universität München (1992)
- [OlvS] Olbrich, M., von Stryk, O.: A direct transcription method for robot trajectory optimization utilizing dynamical structure. In Vorbereitung.
- [OtTü] Otter, M., Türk, S.: The DFVLR Models 1 and 2 of the Manutec r3 Robot. DFVLR-Mitt. 88-13, Institut f. Dynamik der Flugsysteme, DLR, Oberpfaffenhofen (1988)
- [Ou] Ousterhout, J. K.: Tcl and the Tk Toolkit. Addison-Wesley Professional Computing Series (1994)
- [Pe] Pesch, H.J.: Solving optimal control and pursuit-evasion game problems of high complexity. In: [BulKr] (1994) 43-61
- [PeGaMiBr] Pesch, H. J., Gabler, I., Miesbach, S., Breitner, M. H.: Synthesis of optimal strategies for differential games by neural networks. In: G. J. Olsder (Hrsg.): New Trends in Dynamic Games and Applications, Annals of the International Society of Dynamic Games **3** (Boston: Birkhäuser, 1995) 111-141
- [PfrRe] Pfeiffer, F., Reithmeier, E.: Roboterdynamik. (Teubner Verlag, 1987)
- [Schi] Schiehlen, W. O. (Hrsg.): Multibody system handbook. (Berlin, Heidelberg, New York, Tokyo: Springer-Verlag, 1990)
- [Sta] Stahlhut, A.: Effiziente Modellierung der Kollisionsvermeidung in der Roboterbahnoptimierung. Diplomarbeit, Mathematisches Institut, Technische Universität München (1994)
- [StoBul] Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis. 2nd ed. (Springer-Verlag, 1993)
- [vS] von Stryk, O.: Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen. Fortschritt-Berichte VDI, Reihe 8, Nr. 441 (Düsseldorf: VDI-Verlag, 1995)
- [vSSch] von Stryk, O., Schlemmer, M.: Optimal control of the industrial robot Manutec r3. In: [BulKr] (1994) 367-382
- [SZ] Süddeutsche Zeitung: Wie Roboter die Kurve kratzen – Mathematiker suchen nach optimaler Bewegung von Maschinen. Von Karlhorst Klotz (10.02.1994)
- [Tü] Türk, S.: Zur Modellierung der Dynamik von Robotern mit rotatorischen Gelenken. Fortschritt-Berichte VDI, Reihe 8, Nr. 211 (Düsseldorf: VDI-Verlag, 1990)