

Virtual Prototyping und Optimierung freier Parameter eines dreiachsigen Roboters

H. Henrichfreise, M. Pähle und S. Oedekoven
Labor für Mechatronik, Fachhochschule Köln
Betzdorfer Straße 2, 50679 Köln

Zusammenfassung: Zur Verringerung der Produktentwicklungszeiten werden Entwicklungsabläufe zunehmend parallelisiert. Dies bedeutet für ein mechatronisches Produkt die gleichzeitige Entwicklung von Konstruktion, Steuerung, Regelung und Benutzeroberfläche. Hier bietet Virtual Control Prototyping die Möglichkeit, vor Fertigstellung der Konstruktion die Software für Steuerung, Regelung und Benutzeroberfläche frühzeitig unter realistischen Bedingungen zu testen und zu optimieren. Die Regelstrecke wird dabei durch die Echtzeitsimulation eines aus ersten konstruktiven Entwürfen abgeleiteten dynamischen Modells ersetzt. Dieser virtuelle Prototyp kann natürlich auch für die Optimierung der Konstruktion herangezogen werden. Erst dadurch kann wirkliche Optimalität eines mechatronischen Gesamtsystems und die gewünschte Interdisziplinarität der verschiedenen Entwicklungsarbeiten erreicht werden.

Das Virtual Prototyping und die Optimierung eines mechatronischen Systems wird im vorliegenden Beitrag am Beispiel der Entwicklung eines dreiachsigen Roboters beschrieben. Insbesondere wird das erschließbare Verbesserungspotential durch die Einbeziehung freier Konstruktions- und Steuerungsparameter in den Optimierungsprozeß verdeutlicht. Der Aufbau des virtuellen Prototyps und die Durchführung der Optimierung erfolgt mit am Markt verfügbaren, leistungsfähigen Entwicklungswerkzeugen. Das Beispiel ist damit geeignet, noch zurückhaltende Konstrukteure vom Potential der mechatronischen Entwicklungsmethodik für die Entwicklung optimaler Produkte in kürzester Zeit zu überzeugen.

1 Einleitung

Bei der Entwicklung eines mechatronischen Systems ist die Inbetriebnahme des ersten Prototyps der entscheidende Augenblick. Hier zeigt sich, ob die Konstruktion und die entworfene Steuerung und Regelung im realen Betrieb zum gewünschten Systemverhalten führen. Daher wird das Experiment am Prototyp zu einem möglichst frühen Zeitpunkt des Produktentwicklungsprozesses angestrebt. Für die schnelle Realisierung der Steuerungs- und Regelungsalgorithmen und den Test des mechatronischen Gesamtsystems wurden in den letzten Jahren die regelungstechnischen Entwurfsumgebungen um Werkzeuge und Methoden zum Rapid Control Prototyping (RCP) erweitert. Sie enthalten die erforderliche leistungsfähige Echtzeithardware, automatische Codegenerierung sowie Software zur Experimentsteuerung und Echtzeitdatenerfassung, die eine enge Kopplung mit der Entwurfsumgebung ermöglicht. Durch die automatische Inbetriebnahme der Algorithmen lassen sich verschiedene Ansätze in kürzester Zeit überprüfen und Aussagen über die später im Produkt benötigte Rechnerhardware können getroffen werden.

Zur Verringerung der Entwicklungszeit erfolgt die Konstruktion sowie der Entwurf von Steuerung, Regelung und Benutzeroberfläche jedoch zunehmend parallel, so daß erst gegen Ende der Produktentwicklung ein mechanischer Prototyp zur Verfügung steht. Um in diesem Fall den Test von Steuerung und Regelung nicht ausschließlich in der Simulation durchführen

zu müssen, kann auf das Virtual Control Prototyping (VCP) zurückgegriffen werden. VCP nutzt die Möglichkeiten des RCP vor Fertigstellung des mechanischen Prototyps. Dabei erfolgt die Inbetriebnahme der Steuerung und Regelung wie am realen System auf der verfügbaren Prototypinghardware. Der mechanische Prototyp wird durch die Echtzeitsimulation eines geeigneten mathematischen Modells ersetzt. Auf diese Weise können realitätsnahe Tests der Steuerungs- und Regelungssoftware bereits zu einem frühen Zeitpunkt des Entwicklungsprozesses durchgeführt werden. Zusätzlich läßt sich eine enge Kopplung von Synthese und Realisierung zur Verwirklichung eines optimalen Gesamtsystems erreichen.

Dies soll am Beispiel der Entwicklung eines dreiachsigen Roboters demonstriert werden.

2 Das Roboterprojekt im CLM

Im Labor für Mechatronik der Fachhochschule Köln (Cologne Laboratory of Mechatronics) wird in einem Musterprojekt ein dreiachsiger Roboter entwickelt. Ziel des Projektes ist ein schneller und genauer Mechanismus für wiederholte Bewegungen. Der Endeffektor soll vorgegebene Bahnsequenzen in minimaler Zeit durchfahren und gleichzeitig maximale Positioniergenauigkeit erreichen. Ferner soll der Mechanismus an verschiedene Aufgabenstellungen anpaßbar sein. **Bild 2.1** zeigt die gewählte konstruktive Lösung mit SCARA-ähnlicher Konfiguration.

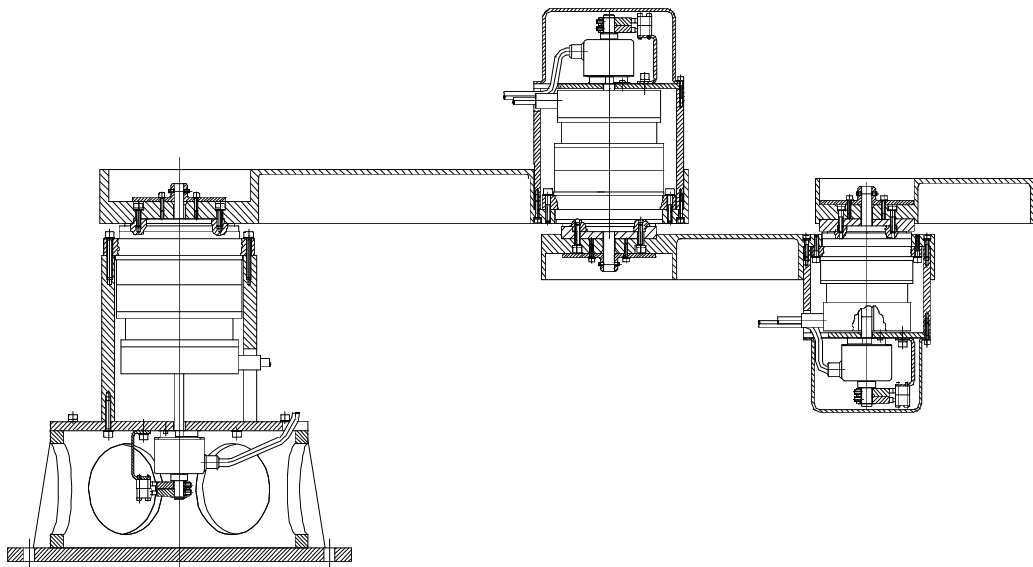


Bild 2.1: Dreiachsiger modularer Roboter

Die geforderte Anpaßbarkeit wird durch einen modularen Aufbau der Konstruktion mit austauschbaren Antrieben und Armen erreicht. Die verwendeten Harmonic Drive Antriebe vereinigen auf geringem Bauraum einen AC-Motor mit Bremse und inkrementalem Winkelgeber sowie ein Harmonic Drive Getriebe, dessen Übersetzung in einem großen Bereich wählbar ist. Ein abtriebsseitiges Kreuzrollenlager erlaubt den direkten Anschluß der Arme. Aufgrund ihrer speziellen Bauweise sind Harmonic Drive Getriebe praktisch spielfrei, weisen aber erhebliche Elastizität auf.

Die Ansteuerung der Motoren erfolgt über stromgeregelte Servoverstärker. Dadurch stehen als Aktoren Momentensteller mit Verzögerungsverhalten erster Ordnung und hoher Bandbreite zur Verfügung.

Die in den Antrieben integrierten Winkelgeber stellen die für die Regelung verwendeten Motorwinkel bereit. Zusätzliche Winkelgeber mit sinusförmigen Ausgangssignalen liefern hochauflösende Messungen der Abtriebswinkel. Aus diesen läßt sich später zusammen mit den Motorwinkeln die Getriebetorsion ermitteln, die für eine schwingungsdämpfende Regelung und Reibungskompensation verwendet werden soll.

Der Entwurf von Steuerung, Regelung und einer Benutzerschnittstelle erfolgt parallel zum konstruktiven Entwurf. Dabei soll zur Minimierung der Kosten und Entwicklungszeit ein bezüglich der genannten Ziele Schnelligkeit und Genauigkeit optimales Gesamtsystem mit nur einem mechanischen Prototyp erreicht werden.

Zur Erfüllung der Entwicklungsvorgaben wurde der Weg des VCP beschritten. Am virtuellen Prototyp können frühzeitig Realisierungseffekte studiert und realistische Systemtests durchgeführt werden. Zudem bildet der virtuelle Prototyp in dieser Arbeit die Basis für die Optimierung des mechatronischen Gesamtsystems.

3 Virtual Control Prototyping

Bild 3.1 zeigt die für das VCP gewählte Systemstruktur und ihre Aufteilung auf eine Echtzeitentwicklungshardware. Ein für die jeweilige Entwicklungsstufe geeignetes dynamisches Modell des Roboters wird auf einem leistungsfähigen Prozessor in Echtzeit simuliert. Die Inbetriebnahme der Steuerung und Regelung erfolgt durch RCP auf einem anderen sogenannten Regelungsprozessor. Als Option können, wie in **Bild 3.1** dargestellt, in der Echtzeitsimulation reale Sensorsignale generiert und die Peripherieschnittstellen der Regelungshardware einbezogen werden, wodurch der virtuelle Prototyp und die damit durchgeführten Experimente noch realistischer werden. Der Datenaustausch zwischen Regelungs- und Simulationshardware erfolgt über Speicherschnittstellen.

Die graphische Animation der Ergebnisse aus der Echtzeitsimulation des Robotermodells in Form eines bewegten Gelenkmechanismus vermittelt einen Eindruck über die Systemfunktion, insbesondere über die Schnelligkeit und Genauigkeit der Bewegungen.

Mit der dargestellten Systemstruktur erlaubt das VCP einen einfachen Übergang zu anderen Formen des Prototyping. So gelangt man nach Fertigstellung des mechanischen Prototyps durch dessen Anschluß an die bis dahin entworfenen Steuerungs- und Regelungsalgorithmen direkt zum RCP. Der Anschluß erfolgt dabei über die beim VCP bereits vorgesehenen Peripherieschnittstellen. Bei Verwendung des Produktcodes für die Steuerung und Regelung in den entsprechenden Blöcken aus **Bild 3.1** liegt eine SIL¹-Simulation vor. Zur HIL²-Simulation gelangt man bei Anschluß der Produkthardware für die Steuerung und Regelung an die Echtzeitsimulation des Streckenmodells oder bei Einbeziehung anderer rückgekoppelter Hardwarekomponenten, wie z.B. eines Kraft-/Momentensensors, in die Echtzeitsimulation.

¹ Software in the Loop

² Hardware in the Loop

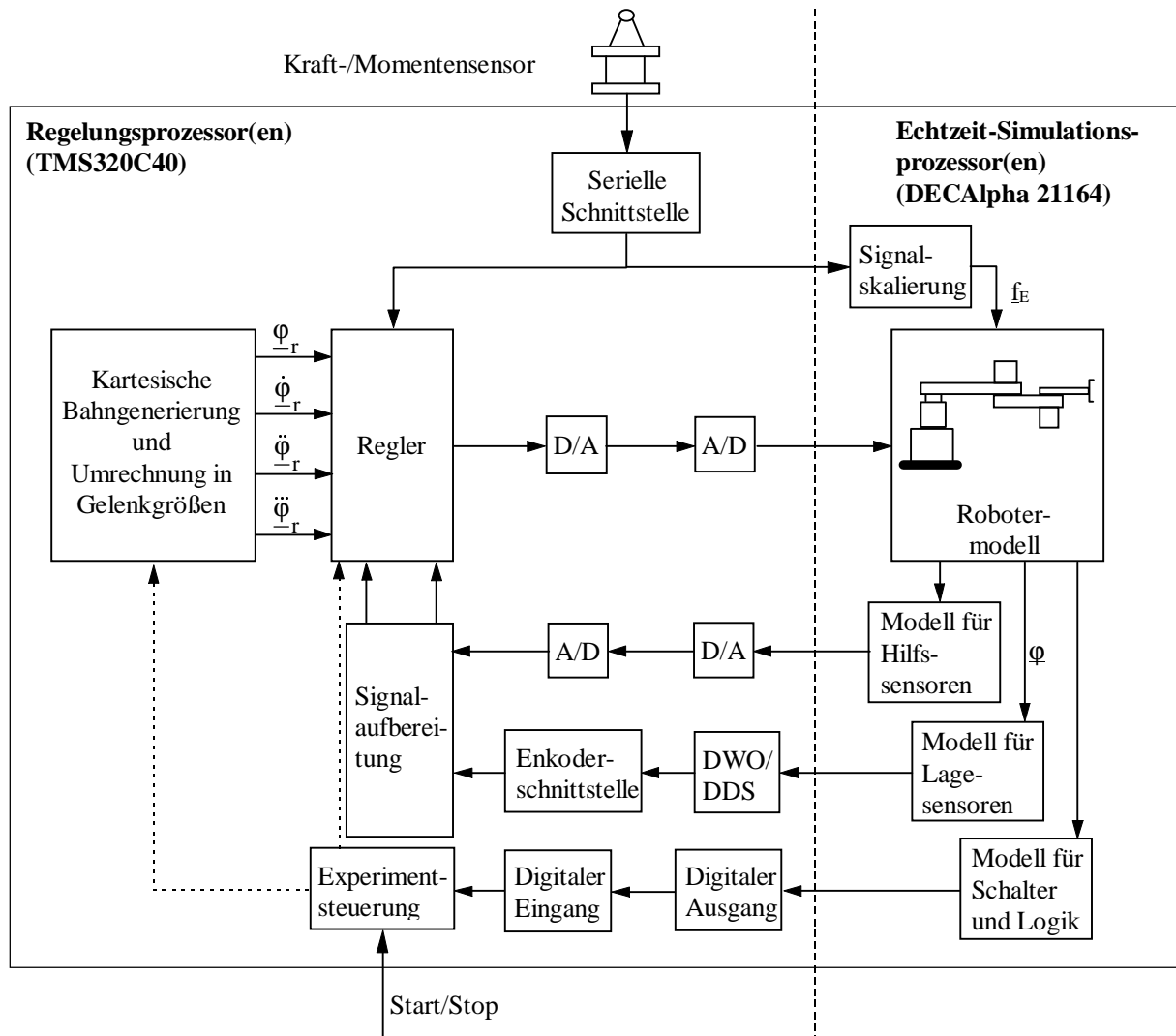


Bild 3.1: Systemstruktur für das Virtual Control Prototyping

Dabei ist besonders hervorzuheben, daß der Übergang zwischen den verschiedenen Formen des Prototyping innerhalb der selben Entwicklungsumgebung stattfindet. Die Entwicklungsmethodik wird nicht gewechselt. Eine Neuprogrammierung der während des VCP erstellten Testsequenzen ist nicht erforderlich. Optimierungssoftware aus der Phase des VCP kann weiterverwendet werden. Dies alles trägt erheblich zu einer Verringerung der Entwicklungszeit bei.

Ein weiterer Vorteil des Beginns einer Systementwicklung mit VCP liegt in der Möglichkeit des Test von Steuerungssoftware für kritische Betriebssituationen. So können z.B. Notausfunktionen für den Fall, daß der Mechanismus mit voller Geschwindigkeit in die Endschalter fährt überprüft werden, ohne ein Risiko der Beschädigung eines realen Prototyps. Solche Tests sind direkt vom Softwareentwickler für den Produktcode durchführbar.

Mit VCP erfolgt nach Fertigstellung der Konstruktion die sofortige Inbetriebnahme der bezüglich aller Komponenten getesteten und voroptimierten Steuerung und Regelung. Die Feinabstimmung des Systems kann danach mit der Optimierungssoftware aus dem VCP beginnen.

Der Kernpunkt des VCP ist ein geeignetes Modell der Regelstrecke. Dabei wird die Systementwicklung meist mit einem einfachen Modell begonnen. Mit Fortschritt der Konstruktion und zunehmender Komplexität der Steuerung und Regelung und der dafür

durchzuführenden Systemtests wird das Modell mehr und mehr verfeinert. Am Ende besteht eine gute Übereinstimmung zwischen dem Modell und dem realen System.

Im folgenden Abschnitt wird auf das für diese Arbeit verwendete Modell des dreiachsigen Roboters und eine dafür angesetzte Regelung eingegangen.

3.1 Modellbildung und Regelung

In der hier vorliegenden frühen Phase des VCP kommt ein relativ einfaches Modell für den dreiachsigen Roboter zum Einsatz [1]. Das Modell enthält eine gute Beschreibung des Starrkörperverhaltens und erlaubt durch die Einbeziehung der Getriebeelastizität Untersuchungen des Stabilitätsverhaltens angesetzter Regelungen. Damit ist es für die in Abschnitt 4.2 beschriebene Optimierung freier Systemparameter geeignet.

Bild 3.2 zeigt das für die Formulierung der Bewegungsgleichungen verwendete physikalische Ersatzmodell des Gelenkmechanismus.

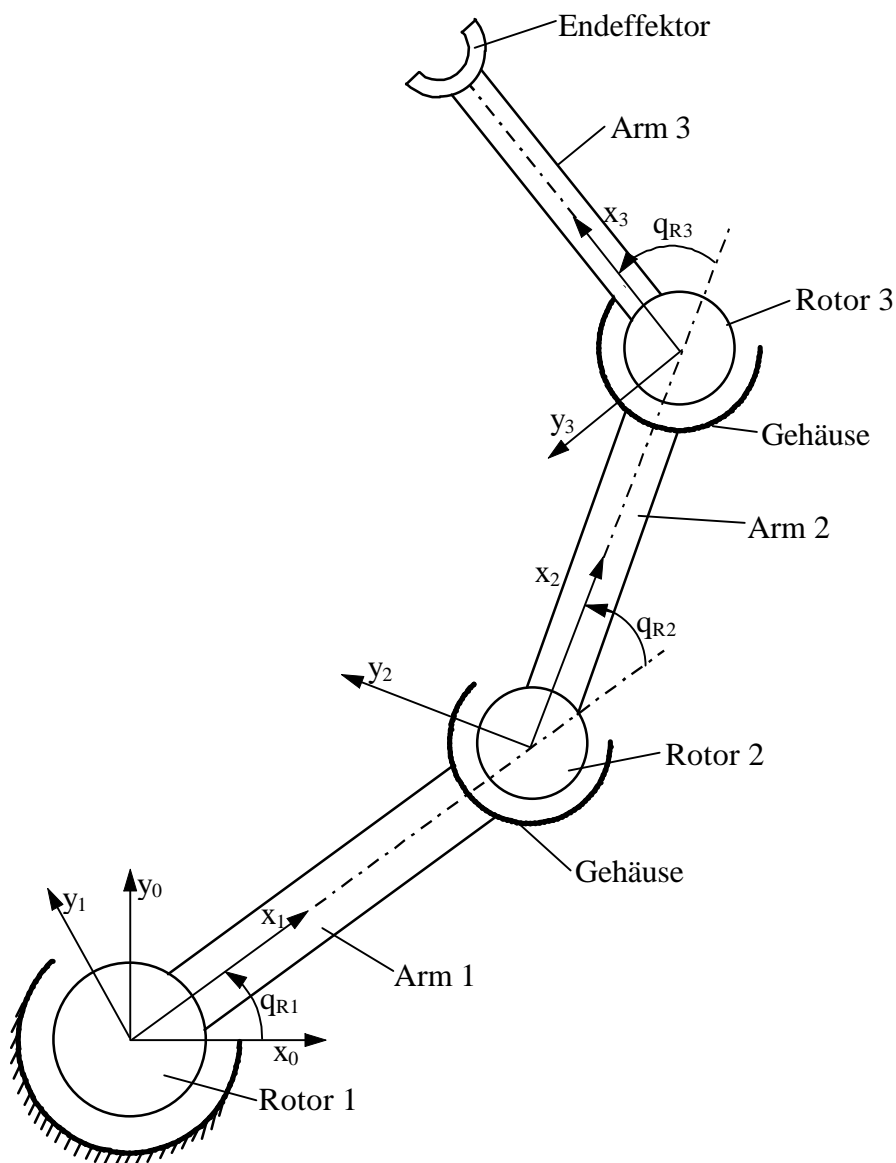


Bild 3.2: Physikalisches Ersatzmodell des Gelenkmechanismus

Dieses Ersatzmodell setzt sich gemäß **Bild 3.3** wiederum aus starren Arm- und elastischen Gelenkmodulen zusammen. In ihnen sind die geometrischen Eigenschaften, Trägheits-,

Steifigkeits- und Dämpfungseigenschaften sowie die Freiheitsgrade und äußeren Kräfte und Momente des Systems spezifiziert.

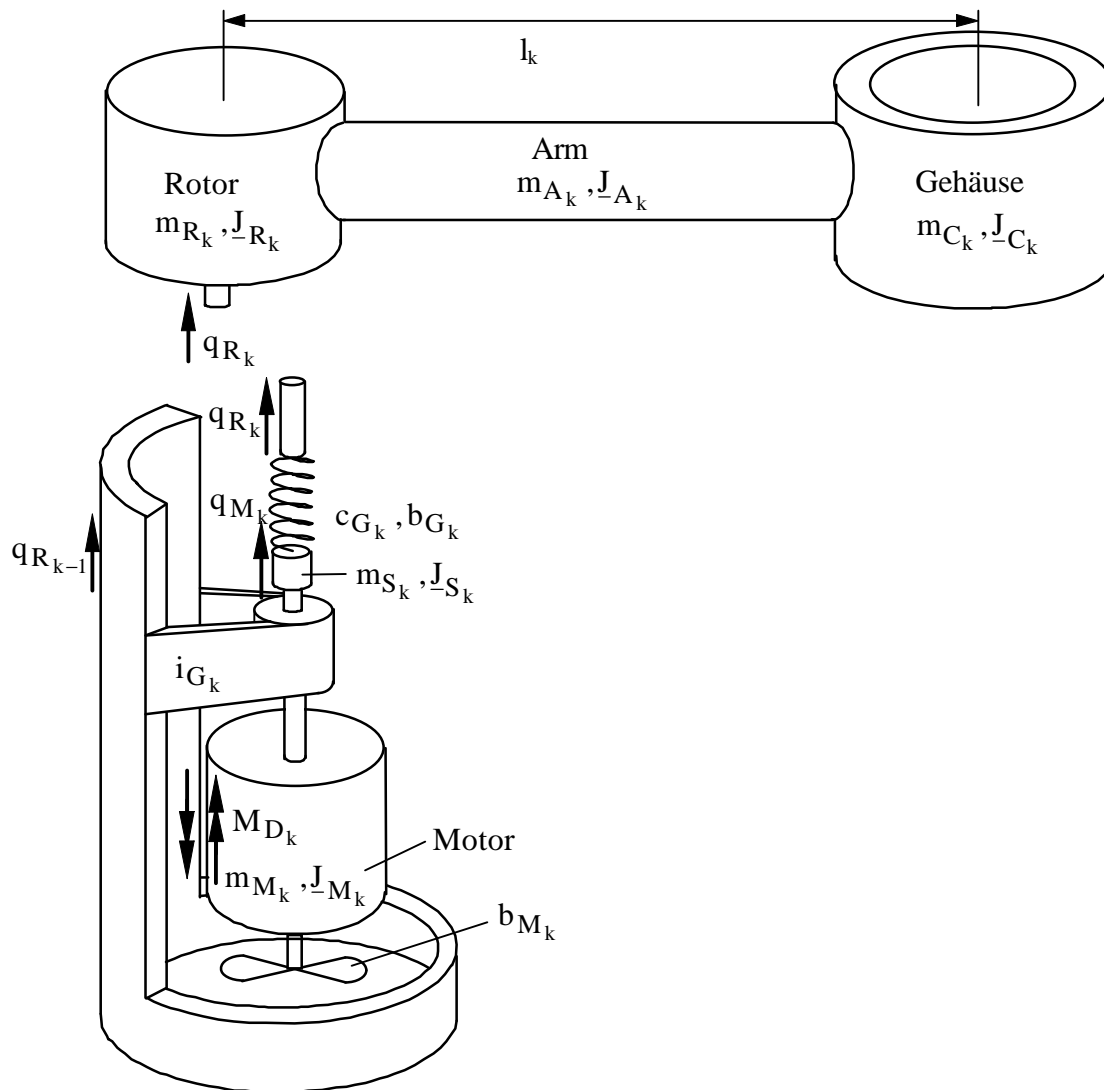


Bild 3.3: Arm- und Gelenkmodul

Ein Armmodul besteht jeweils aus einem Rotor, einem starren Arm der Länge l sowie dem Gehäuse für das nachfolgende Gelenk. Der Rotor faßt dabei die abtriebsseitigen Trägheitseigenschaften des vorangehenden Gelenks zusammen. Die Trägheitseigenschaften eines jeden Körpers werden durch seine Masse m und den Tensor \underline{J} der Hauptachsträgheitsmomente beschrieben.

Ein Gelenkmodul besteht aus dem Motor mit geschwindigkeitsproportionalem Dämpfer mit Konstante b_M , einem reibungsbehafteten Getriebe mit Übersetzung i_G und einem als masselos angenommenen Gehäuse, das in das Gehäuse des vorangehenden Armmoduls eingeschoben wird. Die Getriebeelastizität wird durch eine Feder mit Steifigkeit c_G und Materialdämpfung b_G berücksichtigt. Jedes Gelenkmodul trägt zum Gesamtsystem einen Rotorfreiheitsgrad q_R und einen Motorfreiheitsgrad q_M bei. Es handelt sich dabei um die relativen Drehwinkel zum Gehäuse. Der Motorfreiheitsgrad wurde dabei für eine einfachere Formulierung des Getriebefedermomentes auf der Abtriebsseite des Getriebes angesetzt. Angetrieben wird ein Gelenk durch das Antriebsmoment M_D , das auf die Motordrehmasse und das Gehäuse wirkt.

Der Zusammenbau der Module ist denkbar einfach. Die abtriebsseitige Welle eines Gelenks wird mit dem Rotor eines Armmoduls verbunden. In das Gehäuse auf der anderen Seite des Armmoduls wird das nächste Gelenk eingeschoben. Beide Gehäuse werden dabei fest miteinander verbunden. Am Ende der so entstehenden kinematischen Kette befindet sich anstelle eines Gehäuses der Endeffektor.

Der modulare Aufbau des physikalischen Ersatzmodells mit relativen Gelenkfreiheitsgraden ermöglicht die rekursive Beschreibung der zur Formulierung der Bewegungsgleichungen benötigten Lagekinematik (Ortsvektoren und Drehmatrizen). Die Bewegungsgleichungen werden über die Lagrangeschen Gleichungen [2] hergeleitet. Dabei sind die symbolischen Rechnungen unter MAPLE durch die Verwendung eines Gleichungsgenerators [3] vollkommen automatisiert. Als Eingaben sind dem Gleichungsgenerator die Lagekinematik, die Trägheitseigenschaften der Körper und die von außen an den Körpern des Systems angreifenden Kräfte und Momente bereitzustellen. Letztere sind die an den Motoren angreifenden Motor- und Dämpfermomente sowie in den Getrieben wirkende Reibungs- und Federmomente. Weiterhin werden für einen möglichen Kontakt mit der Umgebung am Endeffektor Störkräfte in der x,y-Ebene und ein Moment um die z-Achse angesetzt. Für die Bewegungsgleichungen wird für ihre Verwendung in einer SIMULINK S-Funktion (siehe Abschnitt 3.2) ebenfalls mit MAPLE C-Code generiert.

Zur Vervollständigung des Modells der Regelstrecke sind dem Modell des Gelenkmechanismus noch die Modelle der Aktorik und Sensorik hinzuzufügen.

Die Aktorik eines Gelenks umfaßt die Erzeugung des Antriebsmomentes M_D aus einer Servospannung u_{servo} . Das Antriebsmoment wird im Luftspalt zwischen Motordrehmasse und Gehäuse durch den Motorstrom induziert. Eine Stromregelung ermöglicht die Modellierung der Verhältnisse zwischen der Sollwertspannung u_{servo} in den Stromregler und dem Antriebsmoment als Verzögerungsverhalten erster Ordnung.

Als Sensoren kommen für die Regelung die motorseitigen Winkelgeber in den Gelenken zum Einsatz. Das Modell eines Sensors beinhaltet eine proportionale Meßverstärkung sowie eine Quantisierungskennlinie, die das Sensorsignal aus dem Motorwinkel $\varphi = i_G \cdot q_M$ realistisch nachbilden.

Für die in dieser Arbeit verfolgte Zielsetzung werden mit den motorseitigen Winkelmessungen zunächst einfache lokale Gelenkregelungen angesetzt. Der Entwurf eines Gelenkreglers erfolgt unter den idealisierenden Annahmen eines starren Mechanismus und einer entkoppelten linearen Bewegungsgleichung mit konstanten Koeffizienten für jedes Gelenk [4]. **Bild 3.4** zeigt die verwendete Reglerstruktur ohne Aktor- und Sensormodelle. Einer Kompensation von Trägheits- und Dämpfungseigenschaften des Gelenkmodells ist eine PD-Rückführung des Motorwinkels überlagert. Die Aufschaltung des Sollwinkels, der Sollwinkelgeschwindigkeit und -beschleunigung garantiert für die obigen Annahmen stationäre Genauigkeit des Gelenkwinkels für beliebige Sollwertverläufe.

Am hergeleiteten realistischen Streckenmodell mit gekoppelten Bewegungsgleichungen und veränderlichen Trägheitsmomenten weisen die beschriebenen Gelenkregelkreise zwar eine winkelabhängige Dynamik auf, arbeiten aber bei einer richtigen Festlegung der Reglerparameter für nicht zu hohe Genauigkeitsansprüche gut. Die Aufschaltung der Ableitungen des Sollwinkels führt zumindest zu einer erheblichen Reduktion stationärer Bahnfehler.

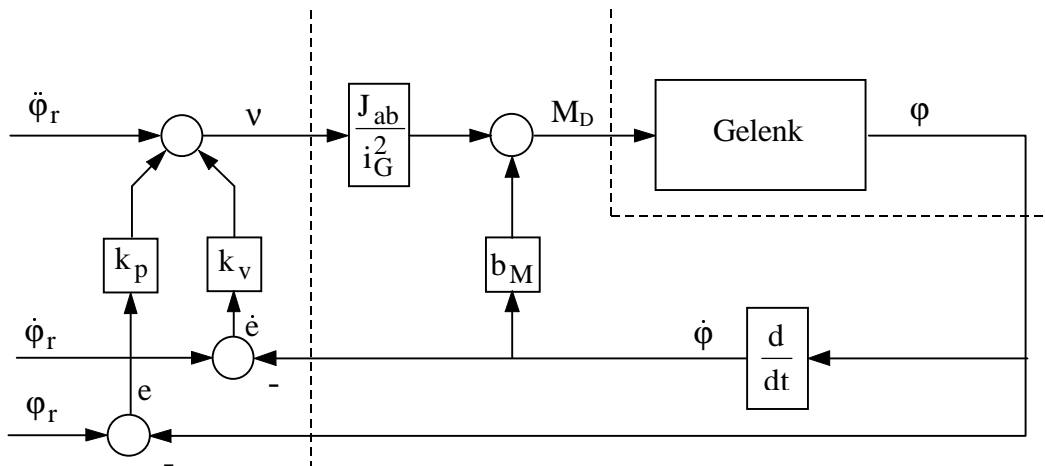
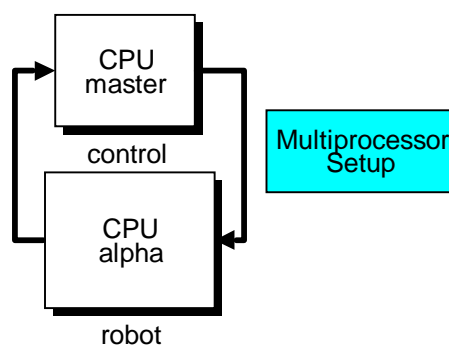


Bild 3.4: Gelenkregelung

Die Sollwertverläufe für die einzelnen Gelenke stellt ein Bahngenerator bereit. Dieser berechnet im Reglertakt aus kartesischen Vorgaben die Sollwinkel, -winkelgeschwindigkeiten und -winkelbeschleunigungen. Die erforderlichen Steuerungsparameter, das sind Zielpunkte des Endeffektors in kartesischen Koordinaten sowie die für die zugehörigen Bahnsegmente maximalen Geschwindigkeiten und Beschleunigungen, werden vom Benutzer in einer graphischen Oberfläche eingegeben.

3.2 Realisierung des virtuellen Prototyps

Der virtuelle Prototyp des Robotersystems enthält die im vorangegangenen Abschnitt vorgestellten Teilsysteme: Modell der Regelstrecke, Regelung und Bahnsteuerung. Diese werden unter der integrierten Entwicklungsumgebung MATLAB/SIMULINK/RTW mit dSPACE TDE auf einem dSPACE Multiprozessorsystem durch Rapid Prototyping realisiert. **Bild 3.5** zeigt das SIMULINK Blockschaltbild des Prozessornetzwerks für die automatische Codegenerierung.

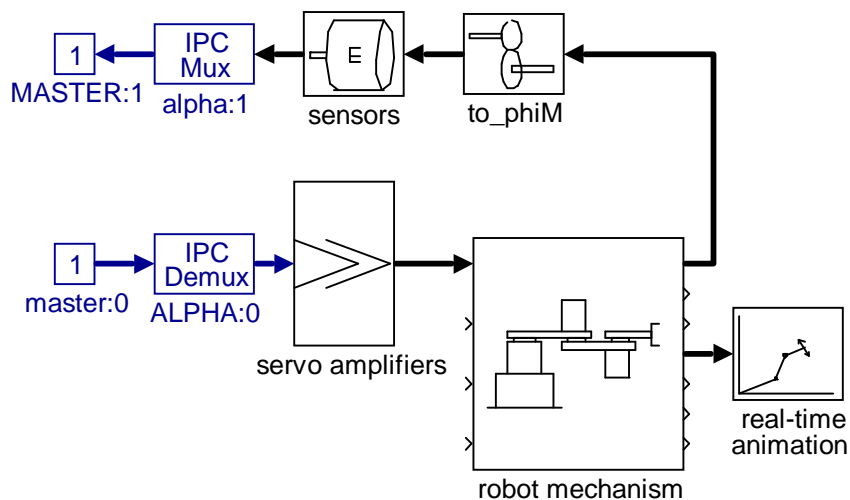


3.5: Prozessornetzwerk

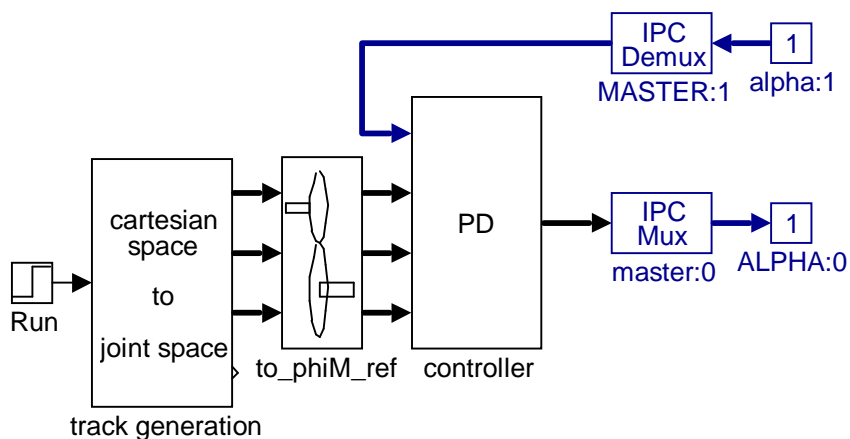
Ein TMS 320C40 (CPU master) übernimmt wie später am realen mechanischen Prototyp die Steuerung und Regelung. Die Echtzeitsimulation des Roboters erfolgt auf einem DEC-Alpha 21164 (CPU alpha). Obwohl der Master-CPU die gesamte erforderliche Peripherie zur Verfügung steht, findet der Datenaustausch zwischen den Prozessoren über IPC (Inter Processor Communication) Schnittstellen statt. Peripherieschnittstellen, wie in **Bild 3.1** dargestellt, bleiben in dieser Stufe der Systementwicklung noch unberücksichtigt.

Unterhalb der Netzwerkebene befinden sich die Blockschaltbilder für die Regelstrecke, Steuerung und Regelung.

Gemäß **Bild 3.6** werden auf dem DEC-Alpha als Simulationsprozessor die Teilmodelle Aktorik (servo amplifiers), Gelenkmechanismus und Sensorik (to_phiM, sensors) der Regelstrecke realisiert. Bei letzteren wird berücksichtigt, daß die in Abschnitt 3.1 beschriebenen Motorfreiheitsgrade q_M zur Formulierung der Motorwinkel als Meßgrößen mit den Getriebeübersetzungen i_G zu multiplizieren sind, bevor sie den Sensormodellen mit Meßverstärkung und Quantisierungskennlinie zugeführt werden. Ein zusätzlicher Block "real-time animation" berechnet aus den Rotorfreiheitsgraden q_R die Ortsvektoren zur Darstellung der Bewegung des Gelenkmechanismus während der Simulation. Weitere Ausgänge des mechanischen Subsystems sind die an- und abtriebsseitigen Winkelgeschwindigkeiten und -beschleunigungen sowie die Getriebeabtriebsmomente. Zusätzliche Eingänge sind die beschriebenen Störkräfte und das Störmoment am Endeffektor.



3.6: Blockschaltbild der Regelstrecke



3.7: Blockschaltbild der Steuerung und Regelung

Das Subsystem "control" des Prozessornetzwerkes auf dem TMS 320C40 enthält die Steuerung und Regelung aus **Bild 3.7**. Die Ausgänge des Bahngenerators stellen die abtriebsseitigen Sollwertverläufe für die Gelenke bereit, die passend zu den Winkelmessungen durch Multiplikation mit den Getriebeübersetzungen (Block to_phiM_ref) auf die Motorseite umgerechnet werden.

Die Inbetriebnahme des virtuellen Prototyps erfolgt ausgehend vom vorgestellten Blockschaltbild mit dem Real-Time Workshop von The MathWorks und dem dSPACE Real-Time Interface für Multiprozessorsysteme durch automatische Codegenerierung. Die Experimentsteuerung, Erfassung und Darstellung der Echtzeitverläufe sowie die Animation der Bewegung des Gelenkmechanismus wurden mit Hilfe der dSPACE Funktionsbibliotheken MLIB und MTRACE unter MATLAB programmiert. Damit steht eine flexible Umgebung für Experimente mit dem Prototyp zur Verfügung.

Das beschriebene Blockschaltbild kann auch für die Offline-Simulation unter SIMULINK verwendet werden. Virtual Prototyping bietet aber neben der Berücksichtigung der Realisierungsbedingungen für die Steuerung und Regelung bei der vorliegenden Modellgröße den weiteren Vorteil einer deutlichen Verringerung der Rechenzeit. Die Ausführungszeit für einen Simulationsschritt und einen Steuerungs- und Regelungstakt beträgt auf der TMS 320C40/Alpha-Combo etwa 250 μ s. Dabei erfolgt die numerische Simulation des kontinuierlichen Streckenmodells mit einem Runge-Kutta Verfahren 4. Ordnung.

Der virtuelle Prototyp dient nicht nur dem Test von Steuerung und Regelung und der Echtzeitanimation der Bewegungen des Gelenkmechanismus. Er bildet ferner, wie im folgenden Abschnitt beschrieben, die Basis für die Systemoptimierung.

4 Optimierung des Systemverhaltens

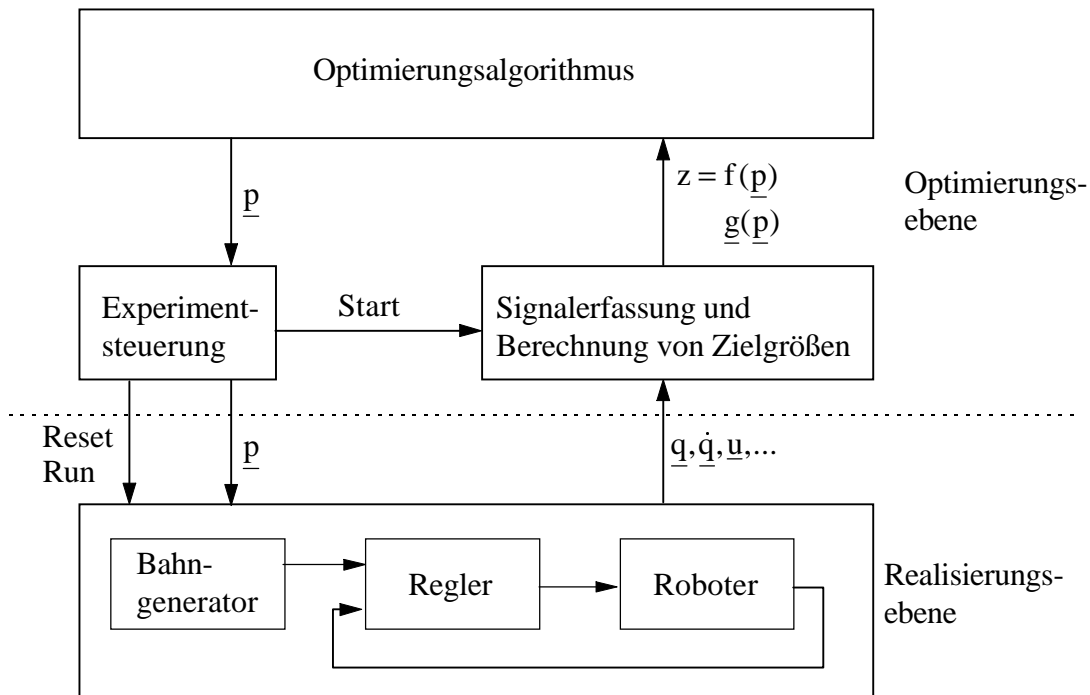
Das im Abschnitt 3 beschriebene Robotersystem enthält eine Vielzahl von Parametern. Diese werden bei herkömmlicher Entwicklungsmethodik zunächst mit einfachen überschlägigen Formeln und intuitiv für die Konstruktion festgelegt. Mit den daraus resultierenden Parametern der Regelstrecke erfolgt danach für stationäre Genauigkeit und ein gutes Einschwingverhalten die Berechnung der Steuerungs- und Regelungsparameter. Das Ziel des mechatronischen Systementwurfs, ein optimales Gesamtsystem, ist mit dieser sequentiellen Vorgehensweise jedoch kaum zu erreichen. Zu einem optimalen System gelangt man nur durch die gleichzeitige Betrachtung der verkoppelten Teilsysteme und die Minimierung der für die gegebene Aufgabenstellung formulierten Zielgrößen. Dies geschieht durch geeignete Variation der freien Parameter in allen Teilsystemen. Zusätzlich sind durch die konstruktive Gestaltung sowie Beschränkung von Größen in der Steuerung und Regelung Nebenbedingungen zu beachten. Dabei verlangt besonders die Formulierung der Entwurfszielgrößen und Nebenbedingungen sowie die Wahl einer geeigneten Optimierungsstrategie ingenieurmäßiges Denken. Zuverlässige Lösungsverfahren für die beschriebene Optimierungsaufgabe sind in heutigen regelungstechnischen Entwicklungsumgebungen vorhanden.

Auf die gewählte Optimierungsstrategie für das vorliegende Robotersystem und die Formulierung geeigneter Zielgrößen und Nebenbedingungen wird in Abschnitt 4.2 näher eingegangen. Zunächst soll die softwaremäßige Realisierung der Optimierung und die Einbindung des virtuellen Prototyps aus Abschnitt 3.2 im Vordergrund stehen.

4.1 Optimierungsumgebung

Zur Durchführung der Optimierung des Systemverhaltens sind das Systemmodell, Funktionen für Zielgrößen und Nebenbedingungen und ein geeigneter Optimierungsalgorithmus zusammenzubringen. Dies geschieht mit der in **Bild 4.1** dargestellten Optimierungsumgebung. Die Basis bildet das Systemmodell, das ohne jede Veränderung aus dem Virtual Prototyping übernommen wird. Es stellt auf der verwendeten Echtzeithardware die Realisierungsebene für das betrachtete mechatronische System dar. Diese Realisierungsebene wird zur Durchführung der Optimierung um Funktionen zur Experimentsteuerung, Signalerfassung, Berechnung von Zielgrößen und Nebenbedingungen und zur Optimierung erweitert. Diese sogenannte

Optimierungsebene wird im Gegensatz zur Realisierungsebene auf dem Hostrechner des Echtzeitsystems unter MATLAB implementiert. Für die Optimierung selbst kommt ein Algorithmus aus der MATLAB Optimization Toolbox zur Anwendung. Der erforderliche Datenaustausch mit der Realisierungsebene erfolgt ebenfalls von MATLAB aus mit Hilfe der zur Echtzeithardware gehörigen Funktionsbibliotheken MLIB und MTRACE. Durch die Verwendung des virtuellen Prototyps aus Abschnitt 3 erfolgt nicht nur der Test von Steuerung und Regelung sondern auch die Optimierung unter realitätsnahen Bedingungen.



4.1: Optimierungsumgebung

Während der Optimierung spielen die in **Bild 4.1** dargestellten Ebenen und Funktionen wie folgt zusammen: Der Optimierungsalgorithmus ruft für einen bestimmten Parametervektor \underline{p} die Experimentsteuerung auf. Diese parametrisiert das Systemmodell in der Realisierungsebene, setzt das Modell zurück, initialisiert und startet die Erfassung der für die Bestimmung der Zielgrößen und Nebenbedingungen benötigten Zeitverläufe aus der Echtzeitsimulation und startet die Simulation. Nach Beendigung des Experiments werden aus den erfaßten Zeitverläufen die Werte der Funktionen für die Entwurfszielgröße $f(\underline{p})$ und Nebenbedingungen $\underline{g}(\underline{p})$ berechnet. Diese werden zur Bestimmung eines neuen Parametersatzes, der die Zielgrößen in Richtung eines Minimums bewegt, zurück an den Optimierungsalgorithmus übergeben.

Durch die Verwendung des virtuellen Prototyps ist die beschriebene Optimierungsumgebung ohne große Änderungen und aufwendige Neuprogrammierung für nachfolgende Formen des Prototyping weiterverwendbar. Mit dem virtuellen Prototyp sind alle Parameter der Subsysteme des mechatronischen Gesamtsystems optimierbar. Beim Übergang zur HIL-Simulation reduziert sich die Anzahl der optimierbaren Parameter lediglich um die Parameter der Teile, die in Form von Hardware eingebunden werden. Beim Rapid Control Prototyping mit der realen Regelstrecke verbleiben die Steuerungs- und Regelungsparameter.

Im folgenden Abschnitt wird eine Anwendung der Optimierungsumgebung für den virtuellen Prototyp vorgestellt.

4.2 Optimierung des virtuellen Prototyps

Schon während der ersten konstruktiven Entwürfe des dreiachsigen Roboters stellte sich bei der Festlegung des Arbeitsraumes und der Auswahl der Antriebseinheiten die Frage nach optimalen Armlängen und Getriebeübersetzungen für gegebene Handhabungsaufgaben. Da diese Frage mit den üblichen konstruktiven Berechnungsmethoden nicht beantwortet werden konnte, wurden die Größen zunächst intuitiv und mit Hilfe überschlägiger dynamischer Berechnungen für minimale Verfahrzeit festgelegt. Durch die mit dem konstruktiven Entwurf parallel vorangetriebene Entwicklung des virtuellen Prototyps und der Optimierungsumgebung stand jedoch ein Mittel zur Optimierung des Systemverhaltens über die noch freien Konstruktionsparameter zur Verfügung. Ausgangspunkt war der oben beschriebene nominale Parametersatz aus dem ersten konstruktiven Entwurf. Die im folgenden geschilderte Vorgehensweise und die resultierenden Ergebnisse stellen das mit einem optimalen konstruktiven Entwurf erschließbare Verbesserungspotential für ein mechatronisches System in anschaulicher Weise dar. Ähnliche Ergebnisse für ein anderes System wurden bereits in [5] erzielt.

Von den Entwurfszielen wird zunächst die Forderung nach maximaler Schnelligkeit des Robotermechanismus betrachtet. Durch eine geeignete Festlegung der freien Konstruktionsparameter und Parameter der Bahnsteuerung soll eine bestimmte Bahn in minimaler Zeit durchlaufen werden. Diese Bahn wird in Form der Start- und Zielpunkte des Endeffektors sowie der Bahnsollbeschleunigung und maximalen Geschwindigkeit auf den einzelnen Bahnabschnitten vorgegeben und vom Bahngenerator im virtuellen Prototyp generiert. Das Durchlaufen der Bahnsequenz stellt für die Optimierungsumgebung das Experiment dar. Die Minimierung der Bahndurchlaufzeit wird mit Hilfe der folgenden zweistufigen Optimierungsstrategie [6] erreicht. Ein Vorteil dieser Strategie liegt neben ihrer Anschaulichkeit in der Konfliktfreiheit der in den einzelnen Stufen verfolgten Subziele, durch die sich ein zügiger Verlauf der numerischen Optimierung einstellt. Für die auftretenden skalaren Zielfunktionen und vektoriellen Nebenbedingungen kommt der Constraint-Algorithmus aus der MATLAB Optimization Toolbox zur Anwendung.

Die gewählte Optimierungsstrategie arbeitet wie folgt:

In der ersten Stufe werden bei konstanten Sollbeschleunigungen auf den einzelnen Bahnabschnitten die Stellgrößen (die Eingangsspannungen der Servoverstärker) aller drei Antriebe minimiert. Dies geschieht mit Hilfe der Armlängen und Getriebeübersetzungen als freie Konstruktionsparameter. Die verwendete Zielfunktion ist das Maximum der Maximalwerte der im Experiment auftretenden absoluten Stellgrößenverläufe. Als Nebenbedingungen sind zu berücksichtigen, daß die Armlängen positiv und die Bahnpunkte mit den jeweiligen Armlängen erreichbar sein müssen. Ferner soll die Summe der Armlängen für einen gewünschten maximalen Arbeitsraum konstant sein.

In der zweiten Stufe der Optimierung erfolgt für jeden Bahnabschnitt mit der zugehörigen Sollbeschleunigung als freiem Parameter die Minimierung der Zielgröße Bahndurchlaufzeit. Dabei gilt als Nebenbedingung, daß bis auf kurzzeitige Ausreißer keine der zuvor minimierten Stellgrößen ihren maximalen Aussteuerbereich von ± 10 V verlassen darf.

Die beschriebenen Optimierungsstufen können wiederholt durchlaufen werden, bis für die gesamte Bahnsequenz keine weitere Verringerung der Durchlaufzeit mehr erzielt wird.

Besonders wichtig für den erfolgreichen Verlauf der Optimierung ist die sorgfältige mathematische Formulierung der Zielfunktionen zur Bildung der Zielgrößen sowie der Gleichungen und Ungleichungen für die zugehörigen Nebenbedingungen. Die Berechnung von Werten für Zielgrößen und Nebenbedingungen erfolgt in der verwendeten Optimierungsumgebung, wie bereits in Abschnitt 4.1 beschrieben, durch Auswertung der Zeitverläufe aus dem Experiment mit Hilfe der genannten Funktionen und Gleichungen.

Das folgende Beispiel für eine einfache Bahnsequenz zeigt die Verbesserung des Systems durch die zweistufige Optimierung. Der Endeffektor durchfährt die geradlinigen Bahnabschnitte vom Koordinatenpunkt (0.7;0) zum Punkt (0.7;0.6) und zurück.

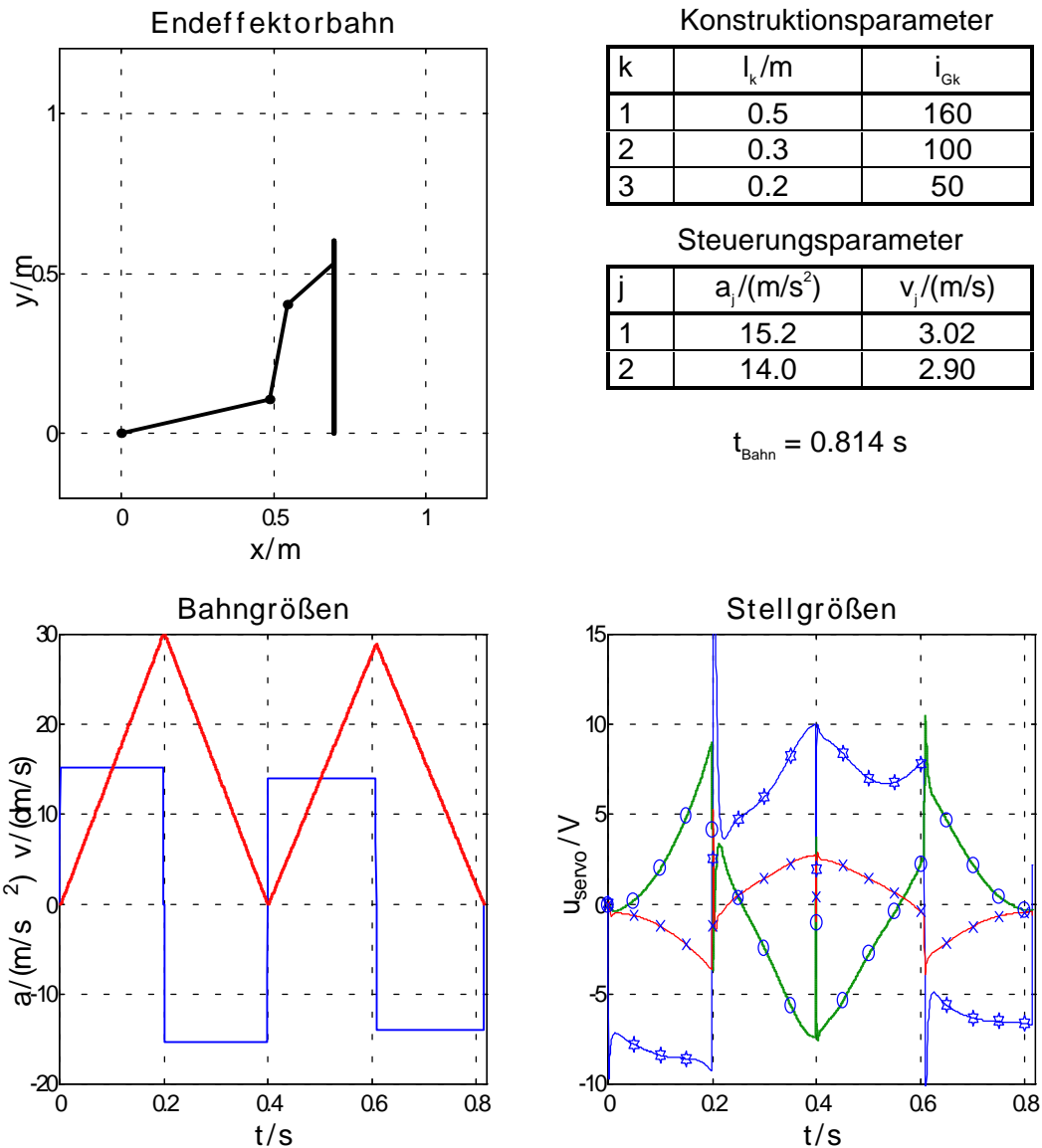


Bild 4.2: Bahnbewegung mit nominalen Konstruktionsparametern

Zunächst wird als Referenz das System mit nominalen Parametern aus dem ersten konstruktiven Entwurf betrachtet. Für dieses System erfolgt allein mit der zweiten Stufe der Optimierung die Minimierung der Bahndurchlaufzeit. **Bild 4.2** faßt das Ergebnis zusammen.

Das Bild links oben stellt die Bahn und den Gelenkmechanismus mit den nominalen Armlängen aus der Tabelle der Konstruktionsparameter dar. Der Endeffektor befindet sich bereits wieder auf dem Rückweg zum Startpunkt der Bahnsequenz (Bahnabschnitt 2). Durch die Optimierung wurden lediglich die Bahnsollbeschleunigungen a_j für die Bahnabschnitte $j = 1, 2$ verändert. Die maximalen Bahngeschwindigkeiten aus der Tabelle der Steuerungsparameter sind von den Beschleunigungen abhängige Größen. Mit der zweiten Stufe der Optimierung wird für den Endeffektor durch maximal mögliche Beschleunigungen von etwa 15 m/s^2 eine minimale Bahndurchlaufzeit von 0.814 s erreicht. Bis auf kurzzeitige Ausreißer durch Einschwingvorgänge in den Regelkreisen bleiben die Stellgrößen (unten rechts) innerhalb der

geforderten Grenzen von ± 10 V. Bei der Auswertung der Optimierungsnebenbedingungen bleiben die Ausreißer unberücksichtigt. Die Zeitverläufe unten links geben die Sollverläufe der Bahnbeschleunigung und -geschwindigkeit wieder. Die Unterschiede zwischen den Beschleunigungen und maximalen Geschwindigkeiten auf den Bahnabschnitten 1 und 2 sind auf die winkelabhängige Dynamik der Gelenkregelkreise zurückzuführen (vgl. Abschnitt 3.1). Dadurch ergeben sich für die Hin- und Rückbewegung des Endeffektors unterschiedliche Stellgrößenverläufe, so daß die Optimierung infolge der Nebenbedingungen mit unterschiedlichen Parameterwerten (Bahnsollbeschleunigungen) beendet wird.

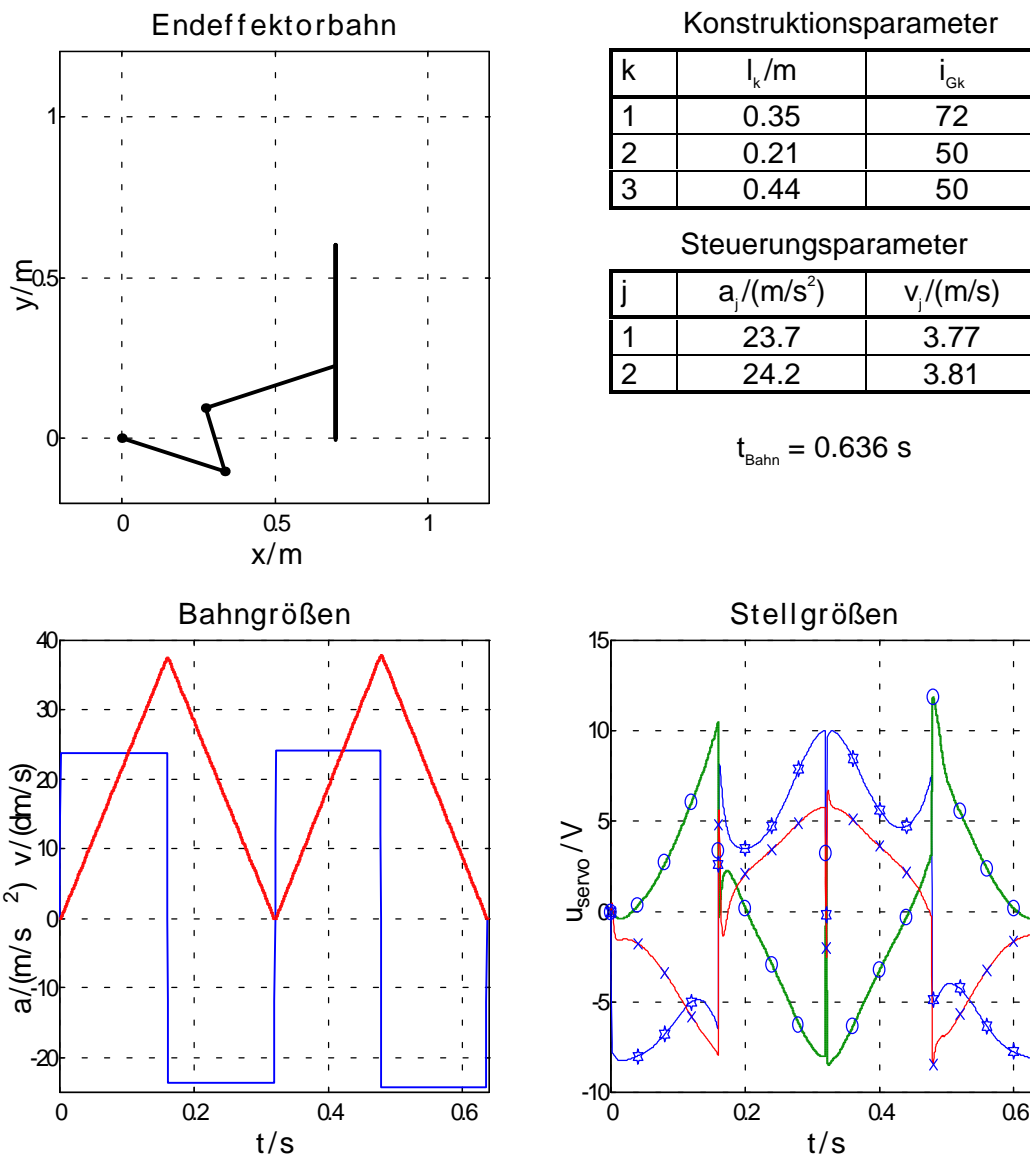


Bild 4.3: Bahnbewegung mit optimierten Parametern

Nun soll die Verbesserung des Systemverhaltens durch die Einbeziehung der Konstruktionsparameter in den Optimierungsprozeß gezeigt werden. Das Durchlaufen beider Stufen der gewählten Optimierungsstrategie liefert das in **Bild 4.3** zusammengefaßte Ergebnis.

Die für die betrachtete Bahnsequenz optimalen Konstruktionsparameter unterscheiden sich erheblich von ihren ursprünglichen Werten. Das Erscheinungsbild des Mechanismus im Bild oben links ist stark verändert. Der Endeffektor befindet sich wieder auf dem Rückweg zum Startpunkt, ist aber zum gleichen Zeitpunkt bereits deutlich weiter als in **Bild 4.2**. Dies veranschaulicht den Vorsprung des Systems mit optimierter Konstruktion, dessen

Bahndurchlaufzeit um 22% auf 0.636 s verringert wurde. Gleichzeitig erhöhen sich die Bahnsollbeschleunigungen von etwa 15 m/s^2 auf etwa 24 m/s^2 . Aus den Zeitverläufen der Stellgrößen läßt sich folgern, daß diese Verbesserung unter anderem durch eine gleichmäßigere Nutzung der Stellgrößen erzielt wird.

Wie am obigen Beispiel bereits deutlich wurde, ist die Optimalität des Systems lediglich für die in der Optimierung verwendete Bahnsequenz gegeben. Für jede Bahnsequenz erhält man einen eigenen optimalen Satz von Konstruktions- und Bahnsteuerungsparametern. Die vorgestellte Optimierung liefert damit spezielle Problemlösungen für spezielle Aufgabenstellungen. Diese Eigenschaft hat ihre besondere Bedeutung für fest vorgegebene schnelle wiederholte Bewegungen in einer Produktionsanlage. Viele einzelne optimale Mechanismen könnten die Fertigungstaktzeit erheblich reduzieren. Aber auch bei wechselnden Handhabungsaufgaben kann davon ausgegangen werden, daß für Bewegungen in der Umgebung einer optimalen Bahn ein nahezu optimales Verhalten vorliegt. In diesem Fall wären bei einmal festgelegten Konstruktionsparametern jeweils die Bahnsollbeschleunigungen mit der zweiten Stufe der Optimierungsstrategie anzupassen.

Eine dritte Stufe der Optimierung, auf die in dieser Arbeit nicht mehr eingegangen wird, ist die Minimierung des Bahnfehlers mit Hilfe der Reglerparameter. Diese Stufe ist auch bei bereits vorgegebener Konstruktion durchführbar. Wirkliche Optimalität des Gesamtsystems im Sinne der zu Beginn genannten Zielsetzung Schnelligkeit und Genauigkeit erhält man aber nur bei Einbeziehung der Parameter aller Teilsysteme und Durchlaufen aller Optimierungsstufen.

5 Schlußbetrachtung

Das Roboterprojekt im CLM ist ein geeignetes Beispiel zur Darstellung der Vorgehensweise bei der Entwicklung eines optimalen mechatronischen Produktes. Es zeigt, daß wirkliche Optimalität nur durch die Einbeziehung aller Systemkomponenten, auch der Konstruktion, in den Optimierungsprozeß erreicht werden kann. Der Beginn der Systementwicklung mit Virtual Control Prototyping ermöglicht realitätsnahe Systemtests, besonders für kritische Betriebssituationen, bereits vor Fertigstellung des mechanischen Prototyps. Die Systemoptimierung auf der Basis des virtuellen Prototyps bezieht nicht nur frühzeitig reale Systemkomponenten ein, sie fördert zudem parallele Entwicklungsabläufe und die gewünschte Interdisziplinarität der konstruktiven, regelungstechnischen und steuerungstechnischen Entwürfe und ihrer Realisierungen. Der Übergang vom VCP zu anderen Formen des Prototyping, z.B. zum Rapid Control Prototyping, ist ohne Wechsel der Entwicklungsmethodik und Werkzeuge möglich. Während des VCP erstellte Optimierungssoftware ist weiterverwendbar. Diese Durchgängigkeit der Entwicklungsumgebung reduziert die Entwicklungszeit. Die gewählte Entwicklungsmethodik verringert die Gefahr von Fehlschlägen. Unzulänglichkeiten und Fehler stellen sich durch die systematische Vorgehensweise auf dem Weg zum optimalen System frühzeitig heraus. Am Ende des Entwicklungsprozesses steht mit nur einem realen Prototyp das nachgewiesene optimale Produkt.

Der Aufbau der beschriebenen Prototyping- und Optimierungsumgebung stellt mit den am Markt verfügbaren Werkzeugen keine Schwierigkeit dar. Die vorgestellte Methodik ist bei ähnlichem Verbesserungspotential auf andere Anwendungsbereiche übertragbar.

Literatur

- [1] Pähle, M.: Entwicklung einer Umgebung zum Virtual und Rapid Prototyping in der Robotik. Diplomarbeit im CLM, FB Konstruktionstechnik, FH Köln 1997.
- [2] Parkus, H.: Mechanik der festen Körper. Springer-Verlag 1981.
- [3] Krohn, M.: Rechnerunterstützte Ermittlung der symbolischen Bewegungsgleichungen hybrider Mehrkörpersysteme. Diplomarbeit im Institut für Automatisierungstechnik, FB Maschinentechnik, Universität GH Paderborn 1986.
- [4] Craig, J.J.: Introduction to Robotics, Mechanics and Control. Addison-Wesley Publishing Company 1989.
- [5] Lückel, J., Kasper, R. und Jäker, K.: Interactive Optimization of Controller and Plant Parameters in the case of Multiple Design Objectives. 3rd IFAC/IFIP International Symposium CADCE'85, Copenhagen, Denmark 1985.
- [6] Oedekoven, S.: Optimierung freier Konstruktionsparameter für einen dreiachsigen Roboter. Diplomarbeit im CLM, FB Konstruktionstechnik, FH Köln 1998.

Kontakt:

Henrichfreise, Hermann: Hermann.Henrichfreise@FH-Koeln.de
http://www.fh-koeln.de/fb/fb-kt/deutsch/labore/mt_lab.htm