

Der mechatronische Entwicklungsprozess am Beispiel der Regelung eines elektromechanischen Positioniersystems

H. Henrichfreise, J. Jusseit und M. Weller
Labor für Mechatronik, Fachhochschule Köln
Betzdorfer Straße 2 ,50679 Köln
Website: www.clm-online.de, E-Mail: info@clm-online.de

Kurzfassung

Am Beispiel der Lageregelung eines elektromechanischen Positioniersystems wird der gesamte mechatronische Entwicklungsprozess durchlaufen. Dabei ist das erste Teilziel der Entwicklung ein optimaler Prototyp der Regelung, der als Referenz für die anschließende Entwicklung von Produktionscode für ein Festkomma-Steuergerät dient. Die dem mechatronischen Entwicklungskreislauf folgende systematische Vorgehensweise und die lückenlose Unterstützung der Entwicklungsstationen durch eine zugeschnittene Werkzeugkette führen in kürzester Zeit zu den gewünschten Ergebnissen. Dabei wird die Bedeutung von Methodenwissen und des Werkzeugeinsatzes auch für die Entwicklung des Steuergerätescodes deutlich. Die Realisierung der Regelung mit Festkomma-Arithmetik zeigt das gleiche sehr gute Führungs- und Störverhalten wie der Gleitkomma-Prototyp.

Einleitung

Innovative Produkte des Maschinenbaus leben vom engen Zusammenwirken mechanischer, elektrotechnischer und informationstechnischer Komponenten. Für die Entwicklung solcher Produkte ist ein neuer ganzheitlicher Ansatz erforderlich, der in der Mechatronik seinen Ausdruck findet. Der zugrundeliegende Entwicklungsprozess verfolgt zunächst die Entwicklung eines optimalen Prototyps für ein Produkt und anschließend dessen Weiterentwicklung zur Serienreife und damit zu einem vermarktbareren Produkt.

Die Vorgehensweise zur Entwicklung eines optimalen Prototyps spiegelt sich im mechatronischen Entwicklungskreislauf wider. Dabei werden ausgehend von einer Produktidee sukzessive die Stationen Modellbildung, Parameteridentifikation, Analyse, Synthese (Steuerungs- und Regelungsentwurf) und Optimierung durchlaufen. Anschließend erfolgt für geeignete Systementwürfe die prototypische Realisierung und die experimentelle Verifikation der Produktidee. Am Ende dieser Entwicklung steht der optimale Prototyp als Vorgabe für die Weiterentwicklung zur Serienreife.

Auf diese Weise wird das im Systemkonzept vorhandene Optimierungspotential ausgeschöpft und werden innovative Produkte kostengünstig und in minimaler Zeit realisiert.

Die Bewältigung der beschriebenen umfassenden Entwicklungsaufgabe erfordert den Einsatz zugeschnittener Software- und Hardwarewerkzeuge. Am Markt weit verbreitet sind Entwicklungsumgebungen wie MATLAB/Simulink und dSPACE-Echtzeitsysteme. Die enthaltenen Toolboxes und Algorithmen sind sehr leistungsfähig, verlangen aber vom Anwender eigene umfangreiche Programmierarbeit und die vollständige Kenntnis der theoretischen Hintergründe. Auf dieser Basis sind zugeschnittene Erweiterungen erforderlich, die es auch dem weniger geübten Anwender ermöglichen, umfassende Entwicklungsaufgaben für mechatronische Produkte zu bearbeiten, und die mit den kommerziell verfügbaren

Das beschriebene EMPS stellt eine Musteranwendung im CLM dar, die von grundlegender Bedeutung für unterschiedlichste Anwendungsbereiche ist. Viele mit diesem System erzielte Ergebnisse konnten bereits auf entsprechende Systeme aus der Fahrzeugtechnik, dem Flugzeugbau und der Antriebstechnik übertragen werden.

Für das EMPS soll im folgenden eine Lageregelung entworfen werden. Natürlich sind einer solchen Regelung im Fahrzeug, Flugzeug oder einem antriebstechnischen Gesamtsystem weitere Steuerungen und Regelungen überlagert. Zugunsten der Darstellung der Entwicklungsmethodik und des Werkzeugeinsatzes beschränkt sich dieser Beitrag jedoch auf die Ebene der Lageregelung.

Modellbildung und Parameteridentifikation

Die erste Station des Entwicklungskreislaufes ist die Modellbildung. Hier wird für die Regelstrecke mit ihren relevanten Eigenschaften ein mathematisches Modell formuliert und auf dem Rechner abgebildet.

Die Abbildung erfolgt für den Fall des nichtlinearen Modells der Regelstrecke in Form des in Bild 2 dargestellten Simulink-Blockdiagramms. Das Modell wird später für die Simulation des geregelten Systems verwendet.

Die oberste Modellebene zeigt die Modelle der Subsysteme Aktor, Mechanismus (für die mechanische Führungs- und Tragstruktur) und Sensorik. Durch die Verwendung von Blöcken aus einer im CLM entwickelten Blockbibliothek mit elektrischen und mechanischen Vierpolen ist der Aufbau der Subsysteme sehr übersichtlich. Für die Entwicklung einer präzisen Lageregelung müssen die an den einzelnen Massen des EMPS angreifenden Reibmomente, die nichtlinear von anderen Systemgrößen abhängen, möglichst realitätsnah nachgebildet werden. Hierfür wurde im CLM ein detailliertes Simulationsmodell entwickelt, das in den Blöcken für die Massen im Subsystem Mechanismus des EMPS zum Einsatz kommt. Nur so kann der Einfluss der Reibmomente auf das Systemverhalten untersucht und können gezielte Gegenmaßnahmen entwickelt werden.

Das Modell für den Aktor enthält den stromgeregelten Motorstromkreis mit PI-Stromregler, Integrator-Anti-Windup, Stromsollwertbegrenzung, Leistungsverstärker, Begrenzung der verfügbaren Klemmspannung und einer Rauschquelle, die Störeinstreuungen global erfasst.

In den Modellen für die Sensoren im Subsystem Sensorik wird die Welligkeit der Tachospannung, Sensorrauschen und Quantisierung berücksichtigt. Diese Effekte spielen wie die Reibung bei hohen Anforderungen an die Lageregelung bezüglich Genauigkeit und Bandbreite (Schnelligkeit) eine wichtige Rolle.

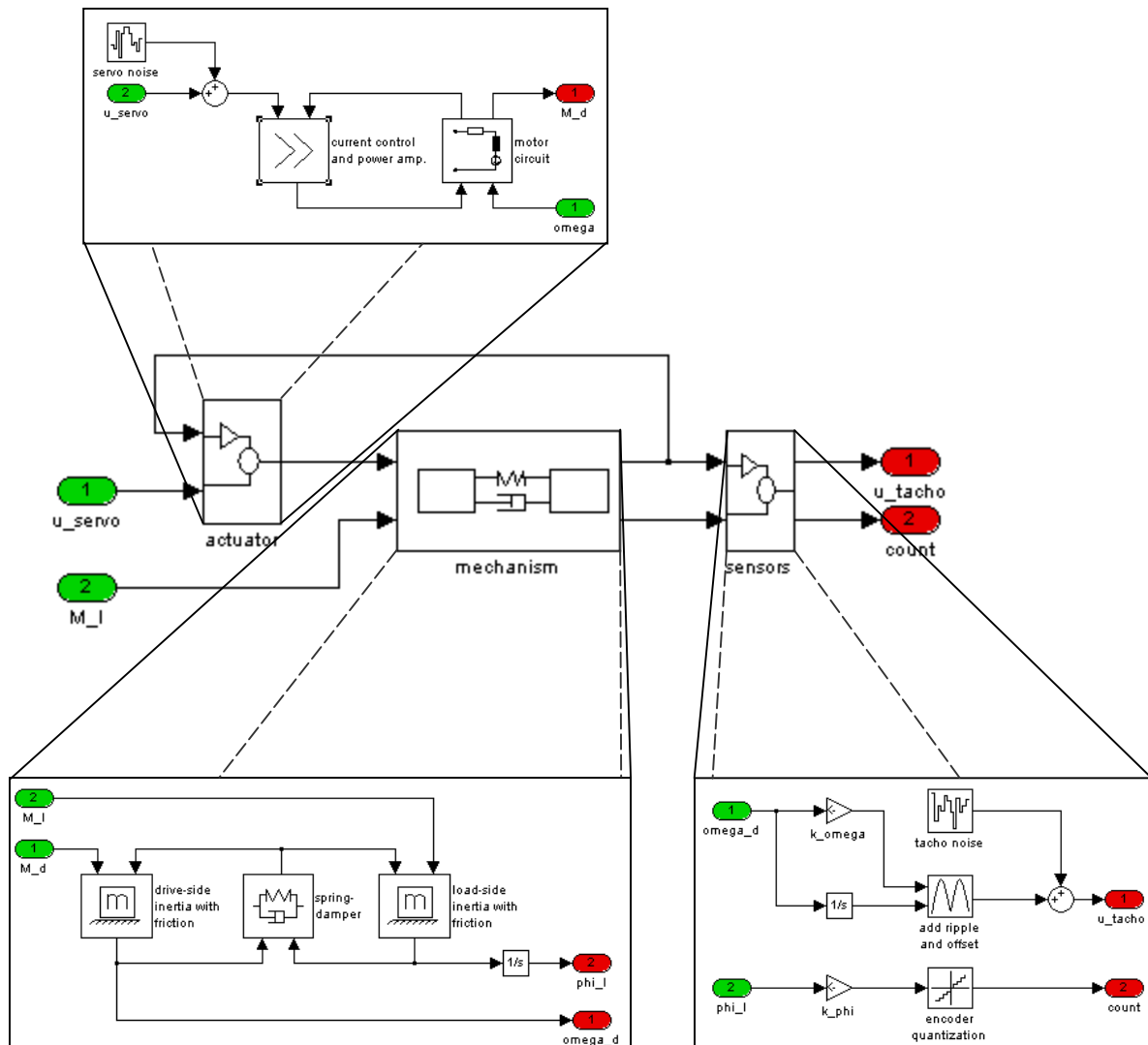


Bild 2: Simulink-Modell der nichtlinearen Regelstrecke

Für die lineare Systemanalyse und den Entwurf einer Regelung mit den Mitteln der linearen Systemtheorie wird weiterhin ein linearisiertes Modell der Regelstrecke benötigt. Die Herleitung dieses Modells findet man in [1].

Nach der Modellbildung folgt als zweite Station des Entwicklungskreislaufes die Parameteridentifikation. Hier werden die Modelle der Regelstrecke an das reale System angepasst. Dies erfolgt bei der bereits festgelegten Modellstruktur (siehe Bild 2) durch die Ermittlung bestmöglicher Parameter für das nichtlineare und lineare Modell. Die Identifikation erfolgt werkzeuggestützt mit den folgenden im CLM entwickelten Tools.

Zunächst wird das Verhalten des realen Systems experimentell ermittelt. Diese Aufgabe übernimmt das Trace und Analysis (TA) Tool. Basierend auf dSPACE Echtzeithardware werden geeignete Anregungssignale erzeugt und die Systemantworten erfasst. Die Weiterverarbeitung der Signale erfolgt auf der Ebene von MATLAB. Das TA-Tool stellt die erfassten Zeitantworten dar, berechnet und plottet auf Knopfdruck gewünschte Funktionen wie Wahrscheinlichkeitsdichten, Spektren, Korrelation, Frequenzgänge und mehr. Die Signalerfassung, Berechnungen und Darstellungen erfolgen gleichzeitig. Das TA-Tool stellt aufbauend auf MATLAB und dSPACE Echtzeitsystemen ohne zusätzliche Investitionen in neue Umgebungen die Funktionalität eines Spektralanalysators zur Verfügung. Bild 3 zeigt

die Benutzeroberfläche mit dem Ergebnis einer experimentelle Analyse für das EMPS in Form von gemessenen Zeitantworten und Frequenzgängen.

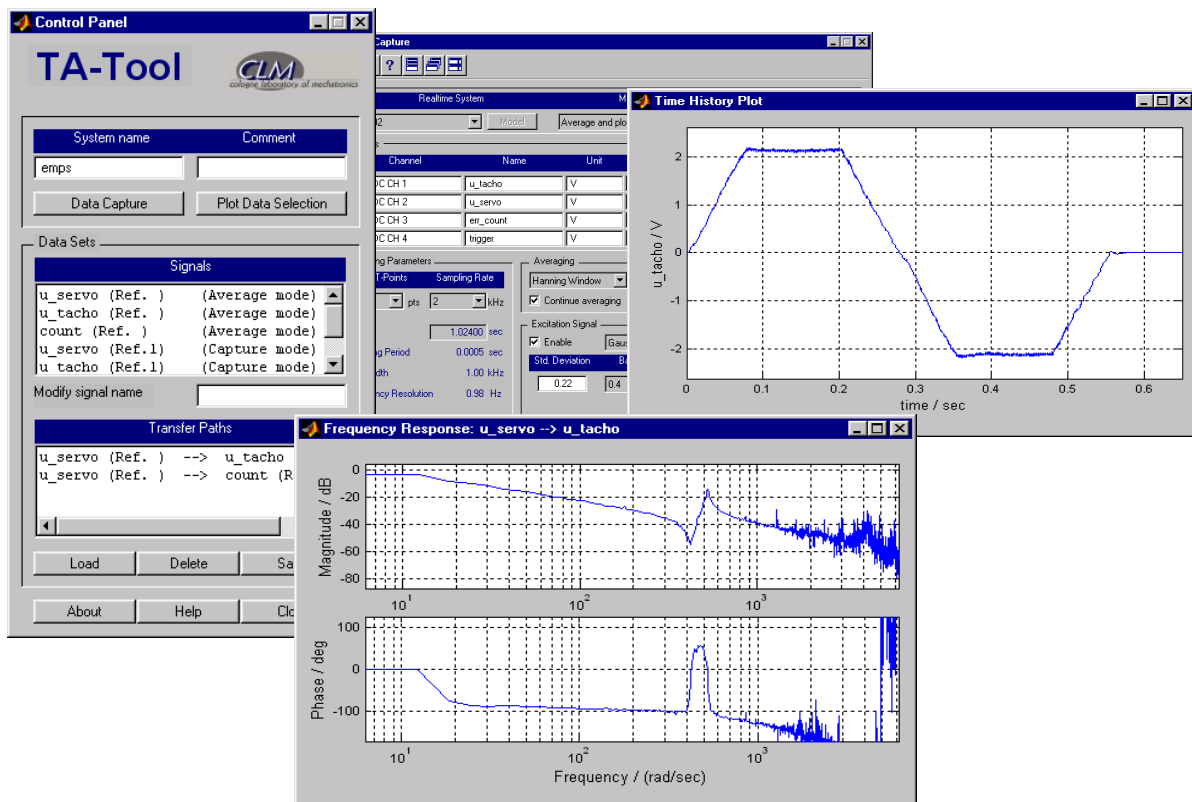


Bild 3: TA-Tool mit gemessenen Zeitantworten und Frequenzgängen

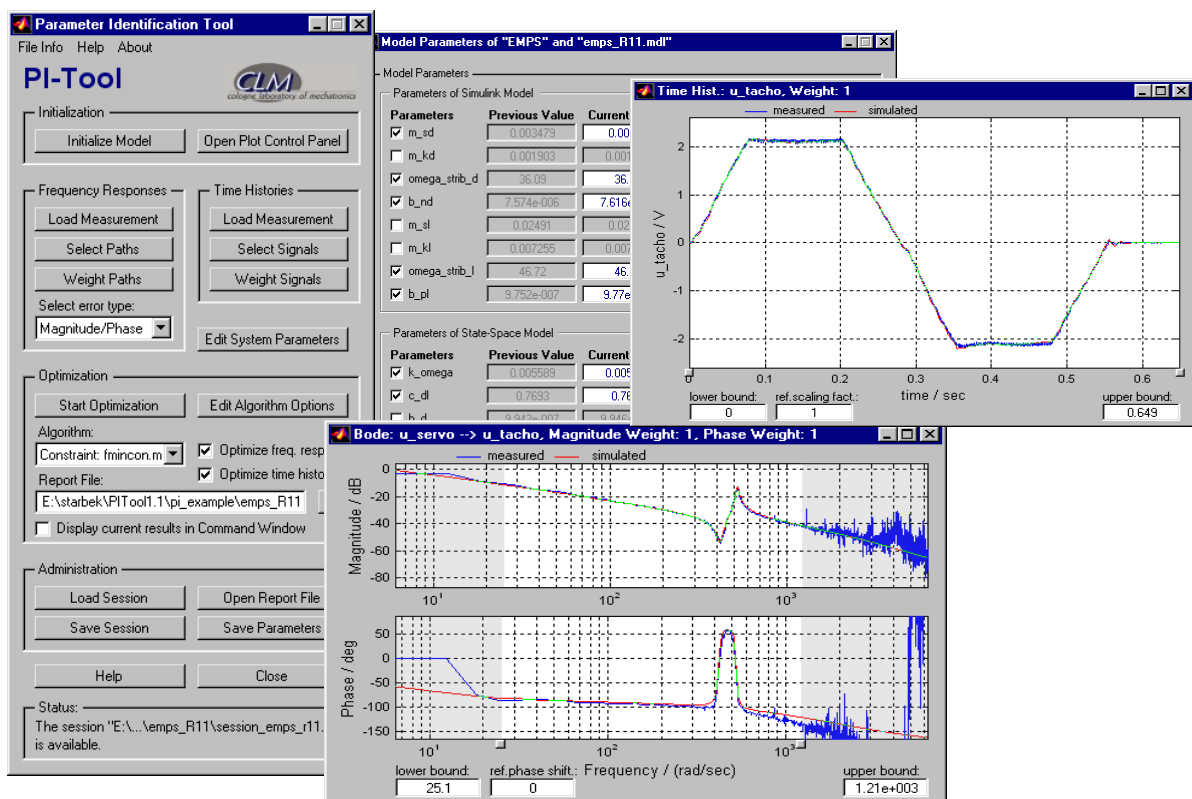


Bild 4: PI-Tool mit Ergebnissen der Modellanpassung

Die eigentliche Parameteridentifikation erfolgt nun mit Hilfe des Parameter Identification (PI) Tools. Ausgangspunkt ist ein Startparametersatz aus Datenblättern, Handrechnungen und Schätzungen. Das Tool minimiert den aus gemessenen und simulierten Zeitantworten und Frequenzgängen gebildeten Modellfehler in Abhängigkeit der für die Identifikation ausgewählten Parameter des nichtlinearen und linearen Modells. Die Modelle in Form des Simulink-Blockdiagramms und eines linearen Zustandsmodells stammen aus der Modellbildung. Als Ergebnis der Identifikation erhält man einen Parametersatz, der für die gewählte Modellstruktur die bestmögliche Übereinstimmung der Antworten der Modelle und des realen Systems ergibt. Bild 4 zeigt die Benutzeroberfläche des PI-Tools und die Ergebnisse der Anpassung der Modelle für das EMPS an die zuvor gezeigten Messergebnisse.

Man erkennt eine sehr gute Übereinstimmung von Modellen und realem System, sodass mit den Modellen eine realistische Basis für den folgenden Reglerentwurf gegeben ist.

Entwurf und prototypische Inbetriebnahme einer optimalen Lageregelung

Bild 5 stellt die allgemeine Struktur einer Lageregelung für das betrachtete Anwendungsbeispiel dar.

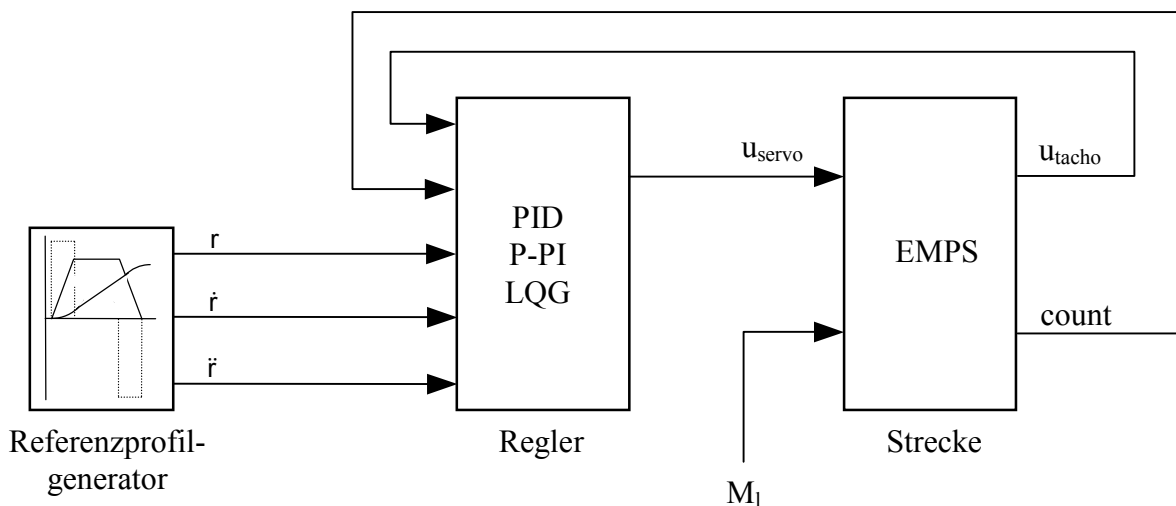


Bild 5: Allgemeine Struktur einer Lageregelung für das EMPS

Mit der Struktur werden die folgenden Ziele verfolgt:

Durch die Rückführung der Messgrößen (u_{tacho} und count) soll der Regelkreis ein möglichst gutes Einschwingverhalten (d.h. kurze Einschwingzeit und geringe Fehleramplituden für den Lagefehler) bei Führungsanregung (r , \dot{r} , \ddot{r}) und Störanregung (M_1) aufweisen. Ferner wird stationäre Genauigkeit (d.h. Lagefehler Null nach dem Einschwingen des Regelkreises) bei Störanregung angestrebt.

Die Aufschalten der Führungsgrößen (Solllage r , Sollgeschwindigkeit \dot{r} , Sollbeschleunigung \ddot{r}) verfolgt das Ziel, stationäre Genauigkeit für die betrachtete Klasse von Sollsignalen zu erreichen.

Geeignete klassische Reglerstrukturen sind die einschleifige Rückführung der Lagemessung count über einen PID-Regler oder eine Kaskadenregelung mit P-Regler für die Lage und unterlagertem PI-Regler für die Hilfsregelgröße u_{tacho} . In die dargestellte allgemeine Struktur

fügt sich aber auch eine optimale lineare Zustandsregelung (LQG-Regelung) ein. In Bild 6 sind zum Vergleich für einen parabelförmigen Sollverlauf typische Verläufe der Lagefehler für die drei genannten Reglerstrukturen dargestellt [2].

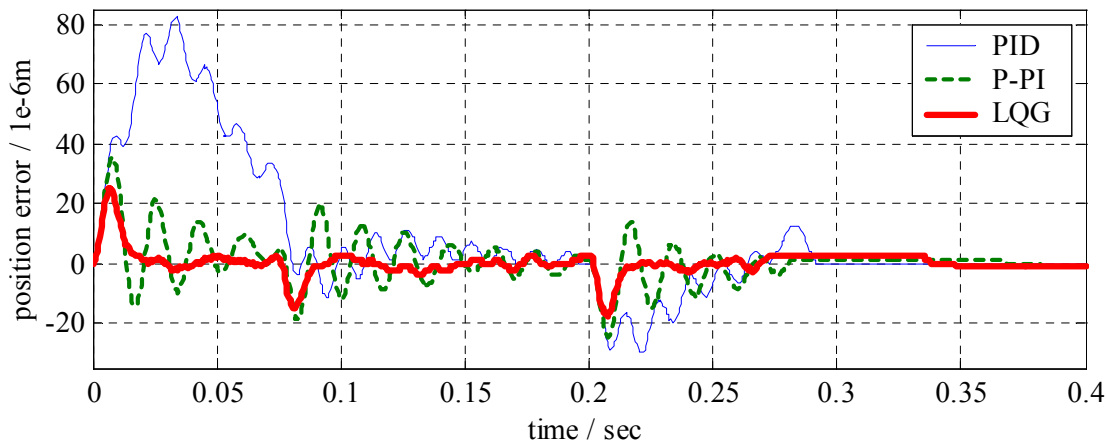


Bild 6: Lagefehler für verschiedene Reglerstrukturen

Die LQG-Regelung erfüllt die Regelungsziele am besten. Im folgenden wird der werkzeuggestützte Entwurf und die Inbetriebnahme der Regelung an diesem Beispiel dargestellt.

Bild 7 zeigt den geschlossenen Regelkreis mit LQG-Kompensator.

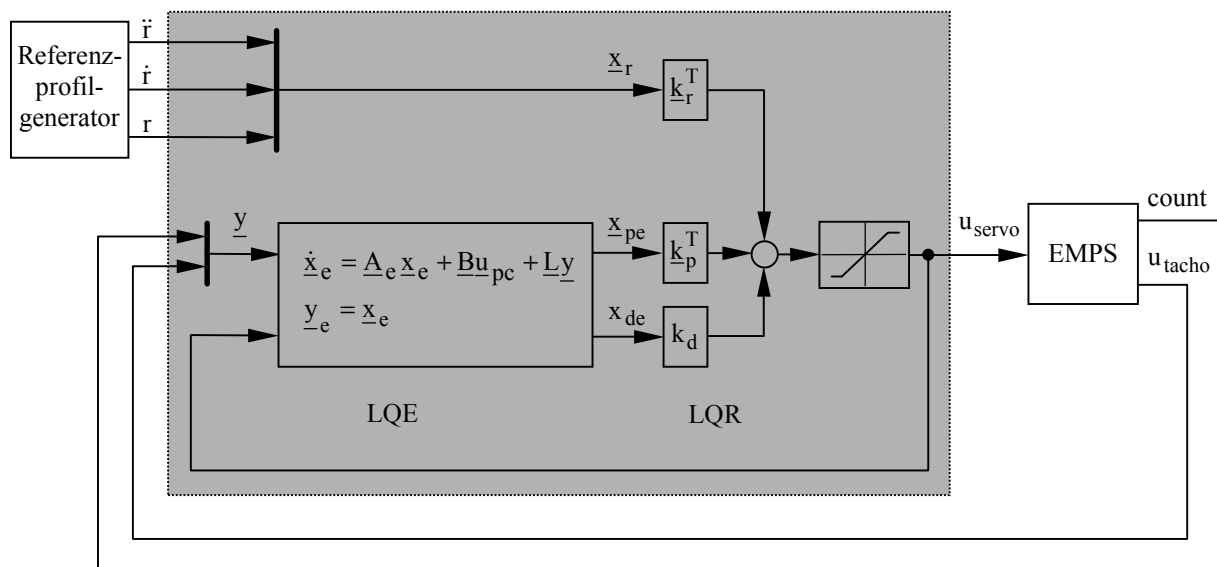


Bild 7: Regelkreis mit LQG-Kompensator

Der dynamische Kompensator entsteht durch die Verkopplung eines statischen Zustandsreglers mit den Reglerverstärkungen \underline{k}_r^T , \underline{k}_p^T und \underline{k}_d und eines linearen dynamischen Beobachters. Die Führungsgrößen werden von einem Referenzprofilgenerator bereitgestellt und über die Verstärkungsmatrix \underline{k}_r^T im Regler auf die Stellgröße aufgeschaltet. Ferner enthält der Regler die Rückführung des Zustandsvektors der Regelstrecke über die Verstärkungsmatrix \underline{k}_p^T und die Aufschaltung lastseitiger Störmomente (infolge Reibung und

externer Störkraft F_1 am Schlitten) über die Verstärkung k_d . Die nicht messbaren Zustands- und Störgrößen werden vom Beobachter aus den Messgrößen und der Stellgröße rekonstruiert. Der Entwurf der Kompensatorsysteme Zustandsregler und Beobachter ist in [1] ausführlich beschrieben.

An dieser Stelle soll auf die Unterstützung des Entwurfes und der Realisierung der Regelung durch weitere Werkzeuge in der bisher beschriebenen Werkzeugkette eingegangen werden.

Das wichtigste Werkzeug ist die in Bild 8 dargestellte Entwurfssuite.

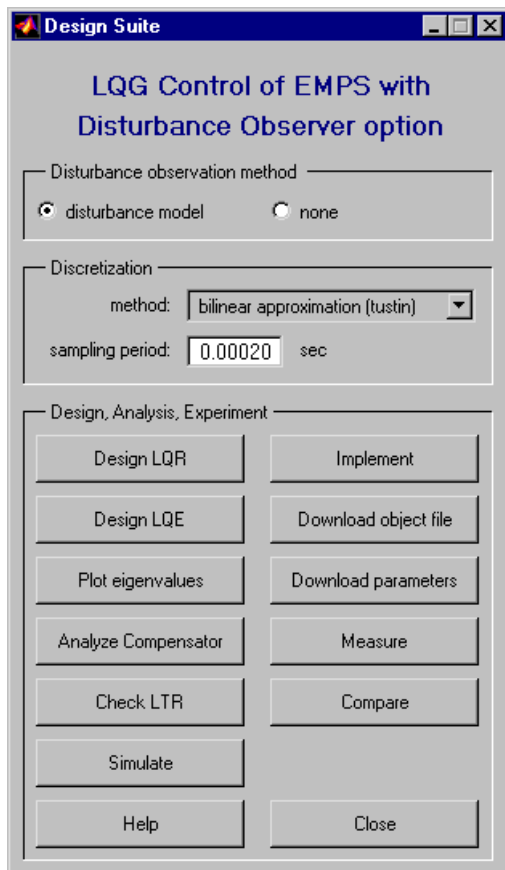


Bild 8: *Entwurfssuite*

Solche Suiten werden im CLM zu jeder Anwendung erzeugt. Die Suite stellt eine anwenderfreundliche Aufbereitung des Wissens (theoretische Grundlagen, Methodik und Werkzeugeinsatz) für den Reglerentwurf und die Realisierung dar. Sie erlaubt eine schnelle interaktive Grobabstimmung der Regelung mit Hilfe von Entwurfsparametern. Dabei wird das Führungs- und Störverhalten und die Robustheit des geregelten Systems durch lineare Analyse im Zeit- und Frequenzbereich und durch Simulation überprüft, die Regelung in Betrieb genommen und im Experiment verifiziert. Vom Entwurf bis zum Experiment geht alles per Knopfdruck. Bild 9 und Bild 10 zeigen das für die Simulation mit der Suite verwendete Simulink-Modell und das für die Inbetriebnahme der Regelung verwendete Echtzeitmodell.

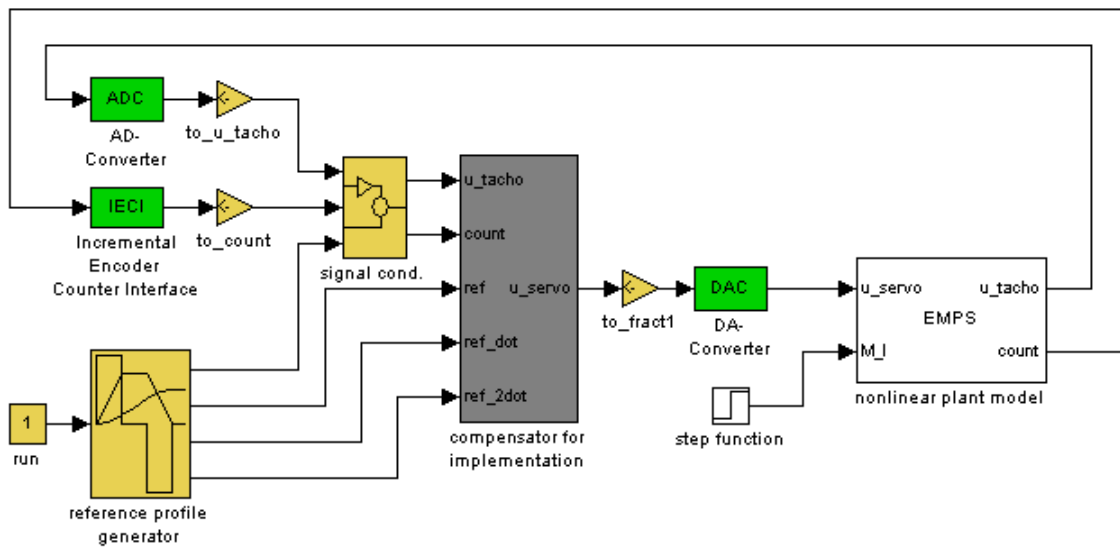


Bild 9: Simulationsmodell für den geschlossenen Regelkreis

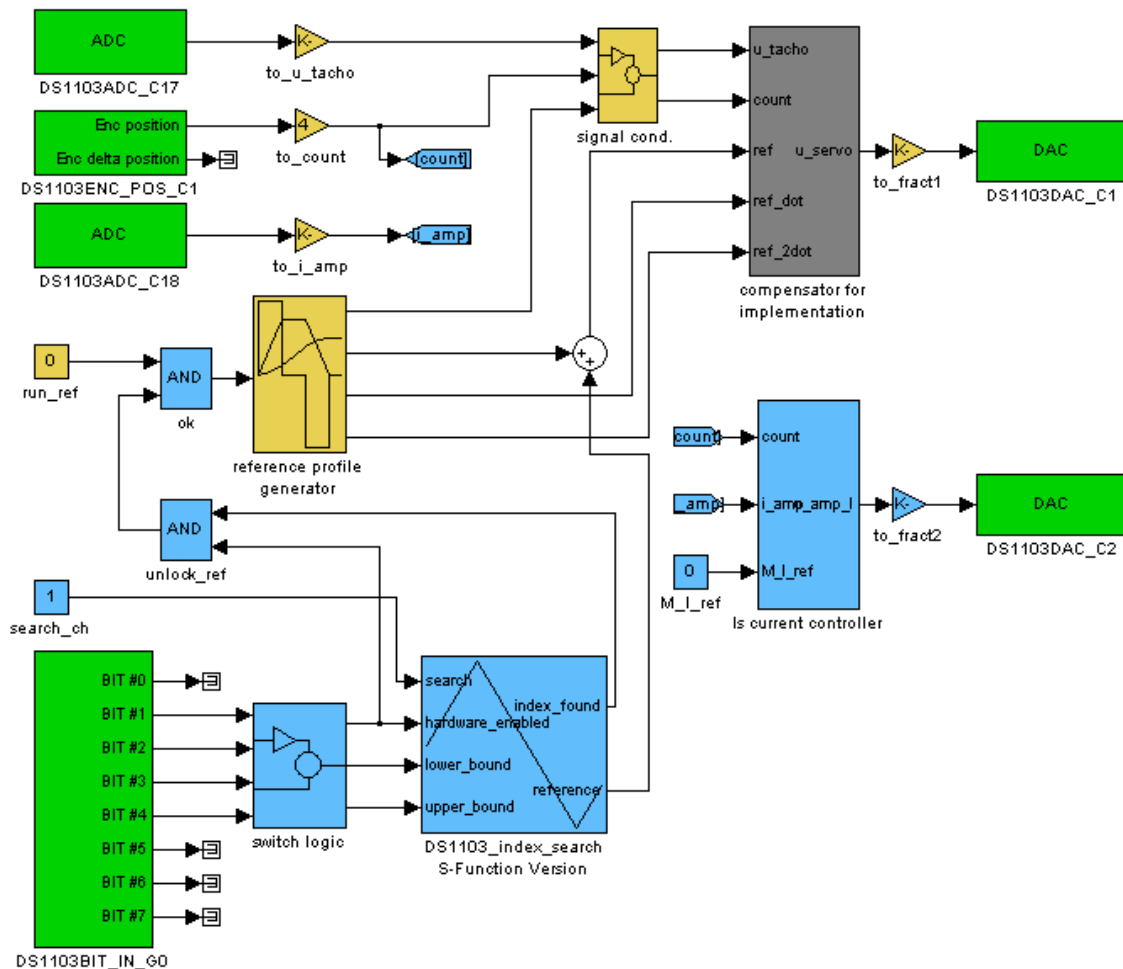


Bild 10: Echtzeitmodell für die Inbetriebnahme der Regelung

Das Simulationsmodell enthält das nichtlineare Streckenmodell aus der Modellbildung, den für die Realisierung diskretisierten Kompensator, den Referenzprofilgenerator und ein Subsystem zur Aufbereitung der Sensorsignale. Modelle der Schnittstellen berücksichtigen

die Abtast- und Haltevorgänge sowie die Quantisierung der I/O-Signale bei der digitalen Realisierung der Regelung. Ergebnisse aus der Simulation werden später zusammen mit Messergebnissen gezeigt.

Mit Hilfe des Echtzeitmodells erfolgt die Inbetriebnahme der Regelung durch Rapid Control Prototyping auf einem dSPACE-Echtzeitsystem. Es enthält den Kompensator, den Profilgenerator und die Signalaufbereitung aus dem Simulationsmodell. Die Modelle für die I/O-Schnittstellen sind durch Blöcke ersetzt, die mit der automatischen Codegenerierung die Zugriffe auf die I/O der Echtzeithardware implementieren. Weitere Blöcke sind für die Initialisierung der inkrementellen Lagemessung, das sogenannte Homing, erforderlich. Ferner wird für die experimentelle Analyse des Störverhaltens eine Stromregelung für den Lastmotor implementiert.

Die Entwurfssuite ist mehr als ein Werkzeug. Zusammen mit den zugehörigen Quelltexten und Modellen stellt sie eine Dokumentation der Vorgehensweise und Ergebnisse dar und dient damit der Konservierung und Weitergabe des Wissens. In dieser Eigenschaft werden im CLM Suiten als Muster für unterschiedlichste Anwendungen entwickelt. Die Verwendung solcher Suiten verbessert die Akzeptanz neuer Theorien und Vorgehensweisen beim Anwender.

Nach der schnellen Grobabstimmung mit Hilfe der Entwurfssuite kommt für die optimale Abstimmung der Regelung ein weiteres Werkzeug aus der CLM-Toolkette zur Anwendung. Das SO-Tool ermöglicht die Optimierung von Systemen bezüglich beliebiger Zielfunktionen und Nebenbedingungen mittels freier Systemparameter, wobei alle Komponenten eines Systems (Konstruktion, Steuerung, Regelung, ...) und des Systementwurfs einbezogen werden können. Bild 11 zeigt die zugehörige Benutzeroberfläche. Hier wurde das SO-Tool für die optimale Einstellung der LQG-Regelung über die zugehörigen Entwurfsparameter eingesetzt. Dabei konnten die Entwurfsvorgänge und Modelle unverändert aus der Suite übernommen werden.

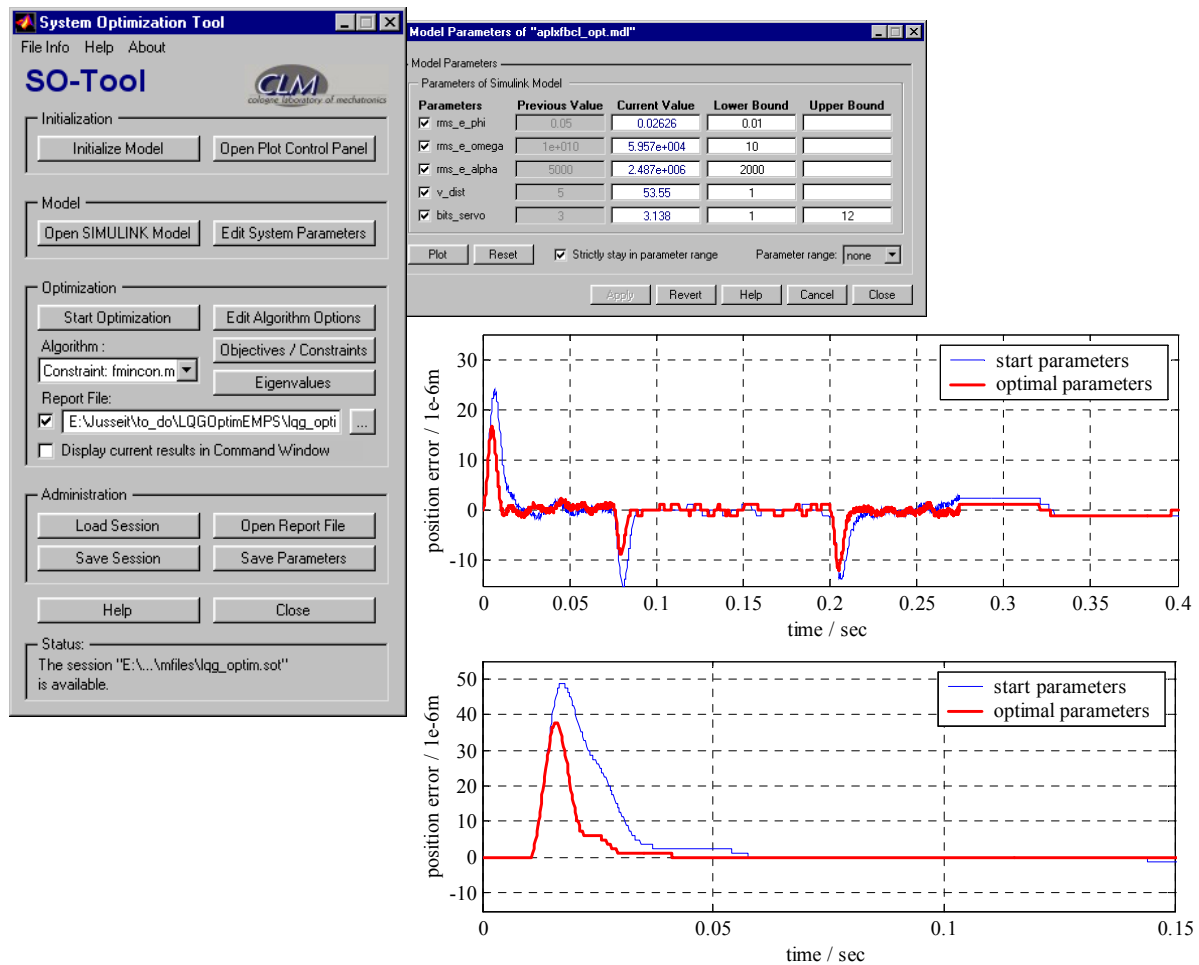


Bild 11: SO-Tool mit Ergebnissen für den LQG-Entwurf

Ausgangspunkt des Optimierungsprozesses sind die Entwurfsparameter aus der Grobabstimmung. Die anschließende Optimierung basiert auf Zeitantworten aus der Simulation des nichtlinearen Gesamtsystems. Mit Hilfe der Zeitantworten werden durch sukzessiven Reglerentwurf die quadratischen Gütefunktionale für den LQG-Entwurf numerisch minimiert. Die Minimierung erfolgt über die Entwurfsparameter, die schließlich in die Entwurfssuite als Default-Parameter übernommen werden.

Im folgenden werden die mit dieser zweistufigen Vorgehensweise mit Entwurfssuite und SO-Tool erzielten Ergebnisse aus Simulation und Experiment vorgestellt. Für beide Fälle wurden jeweils die gleichen Anregungssignale verwendet. Zur Untersuchung des Führungsverhaltens der Regelung dienen als Sollsignale ein parabelförmiger Sollverlauf für den Schlitten und die zugehörigen Geschwindigkeits- und Beschleunigungsverläufe aus Bild 12. Als Störانregung wird ein sprungförmiges Lastmoment, im Experiment über die Stromregelung des Lastmotors aufgeschaltet.

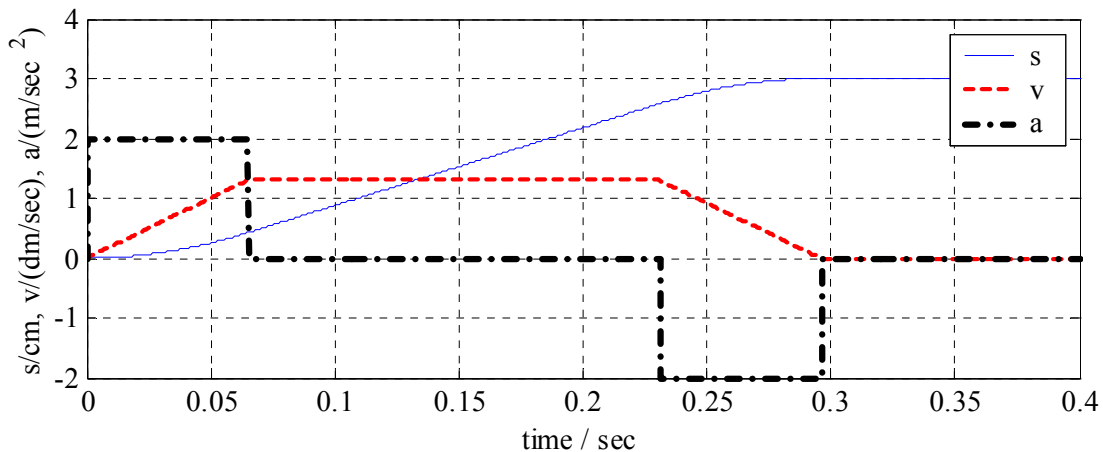


Bild 12: Sollsignale zur Untersuchung des Führungsverhaltens

Die resultierenden Führungs- und Störantworten des geregelten Systems zeigen Bild 13 und Bild 14.

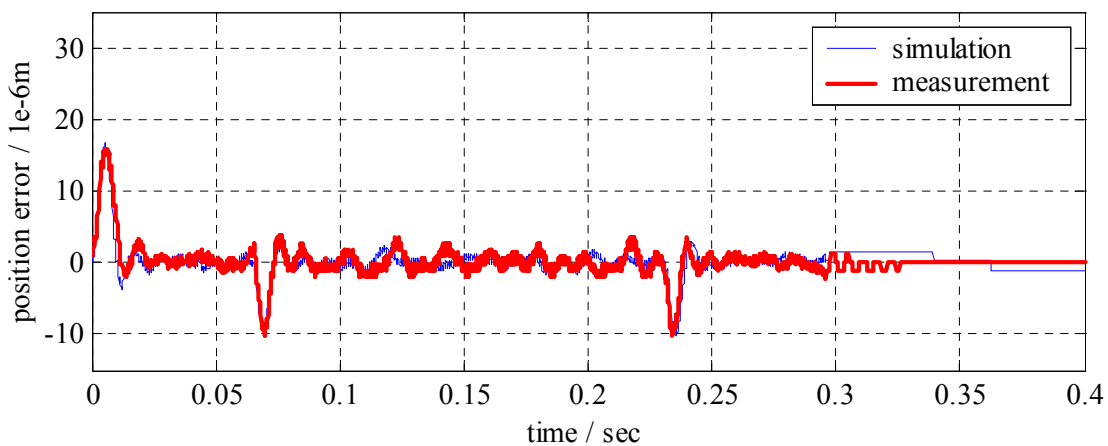


Bild 13: Führungsantwort

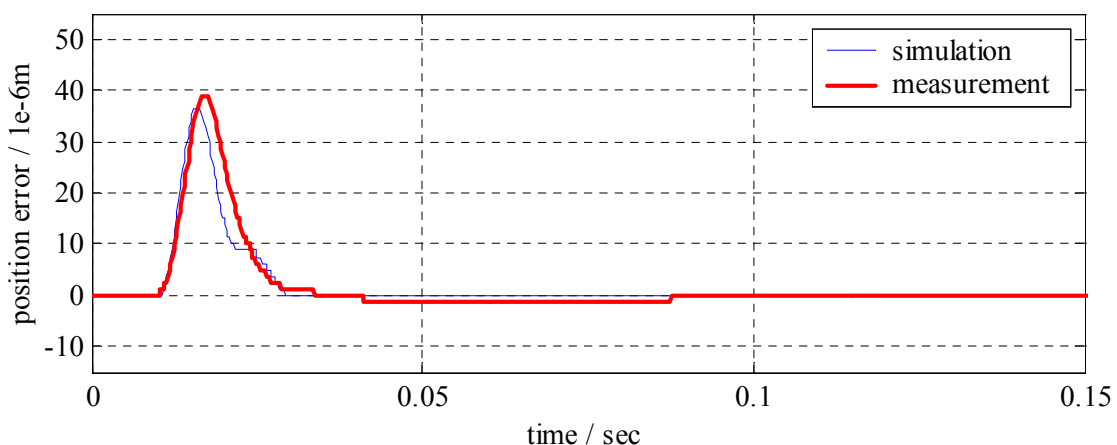


Bild 14: Störantwort

Man erkennt eine sehr gute Übereinstimmung der Ergebnisse aus Simulation und Experiment. Die Führungsantwort zeigt nach Änderungen der Sollbeschleunigung ein schnelles Einschwingen des Lagefehlers. Die elastischen Schwingungen im System werden praktisch ausgeglet. Die Regelung ist aufgrund der Führungsgrößenaufschaltung in allen

Bewegungsphasen des Schlittens (Beschleunigung, konstante Geschwindigkeit, Verzögerung) stationär genau. Auch im Stillstand treten keine stationären Regelabweichungen auf. Dies ist sowohl der Führungs- als auch der Störantwort zu entnehmen. Die Auswirkung von Reibung und externen Störmomenten auf den Lagefehler wird durch die Beobachtung und Aufschaltung dieser Momente im Kompensator stationär kompensiert.

Die Qualität der entwickelten Regelung wird zusätzlich bei Betrachtung der Auflösung von $1.25 \mu\text{m}$ des Lagegebers deutlich. Die stationären Lagefehler bewegen sich im Bereich der Auflösung.

Nach Durchlaufen des Entwicklungskreislaufes wird der Kreislauf mit einem optimalen Prototyp für die beschriebene Lageregelung verlassen. Dafür mitverantwortlich sind die vorgestellte Entwicklungsmethodik und der Werkzeugeinsatz, die zudem eine Reduktion der Entwicklungszeiten bewirken.

Mit dem optimalen Prototyp ist die Entwicklung des Produktes noch nicht abgeschlossen. Der Prototyp stellt zusammen mit der Entwurfssuite die Spezifikation für die anschließende Weiterentwicklung zur Serienreife dar.

Weiterentwicklung zur Serienreife

Die Weiterentwicklung des optimalen Prototyps betrifft alle Komponenten des mechatronischen Systems, die Konstruktion, Sensorik, Aktorik und die Komponenten der Informationsverarbeitung (Hardware und Algorithmen). Im folgenden soll für das behandelte Beispiel auf die Entwicklung von Festkomma-Code für die entworfene Lageregelung eingegangen werden. Wie beim Prototyping sind auch hier für eine effiziente und erfolgreiche Durchführung der Arbeiten eine systematische Vorgehensweise und Werkzeugeinsatz erforderlich.

Ausgangspunkt für die Weiterentwicklung der Regelung ist der kontinuierliche Kompensator aus Bild 7. Die erforderlichen Schritte zur Erzeugung von Festkomma-Code für ein Steuergerät sind eine spezielle Aufbereitung der Kompensatorgleichungen, ihre Diskretisierung sowie die automatische Skalierung und Code-Generierung.

Die Aufbereitung beginnt mit der Zusammenfassung der Kompensatorsubsysteme Regler und Beobachter zu einem Zustandsmodell. Das Ergebnis ist in Bild 15 dargestellt.

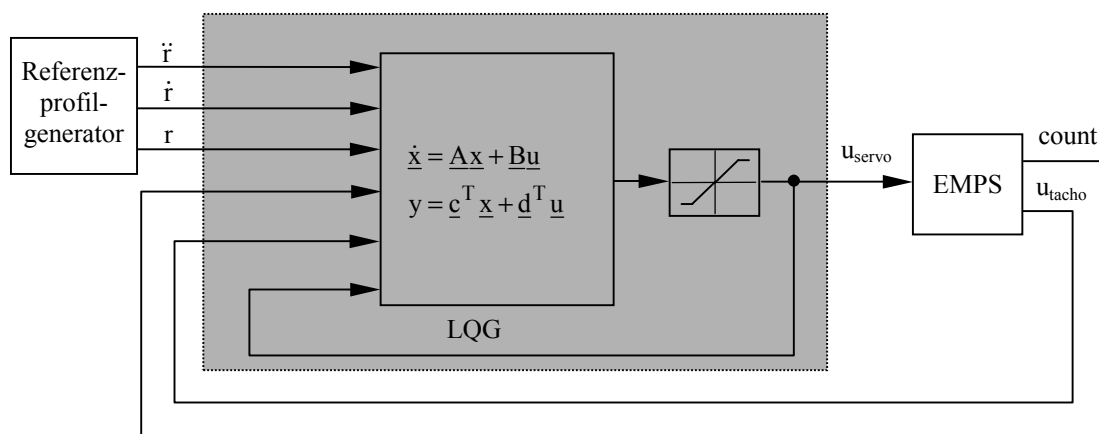


Bild 15: Regelkreis mit zusammengefasstem Kompensator aus dem Entwurf

Diese Maßnahme verringert die Anzahl der in Echtzeit durchzuführenden Rechen- und Speicheroperationen und ist für die weitere Aufbereitung des Kompensators für Festkomma-Arithmetik günstig.

Für den folgenden Schritt sind die Anforderungen an die betrachtete Lageregelung verantwortlich. Die Regelgröße enthält absolute Lageinformation mit einem großen Aussteuerbereich. Gleichzeitig wird eine hohe Genauigkeit der Lage verlangt. Bei einem Aussteuerbereich von ± 0.15 m und einer Auflösung der Lagemessung von $1.25 \mu\text{m}$ beträgt der Unterschied mehr als 5 Größenordnungen. Das gleiche gilt auch für andere Größen in den Kompensatorgleichungen, die Lageinformation enthalten. Die Realisierung eines großen Aussteuerbereiches bei gleichzeitiger Forderung einer hohen Auflösung ist mit Festkomma bei einer begrenzten Wortlänge von beispielsweise 16 Bit nicht darstellbar.

Zur Erfüllung der Forderung nach hoher Auflösung dient die folgende Transformation der Kompensatorgleichungen auf die sogenannte Relativform [3;4]. Bild 16 zeigt das Ergebnis für das Blockdiagramm.

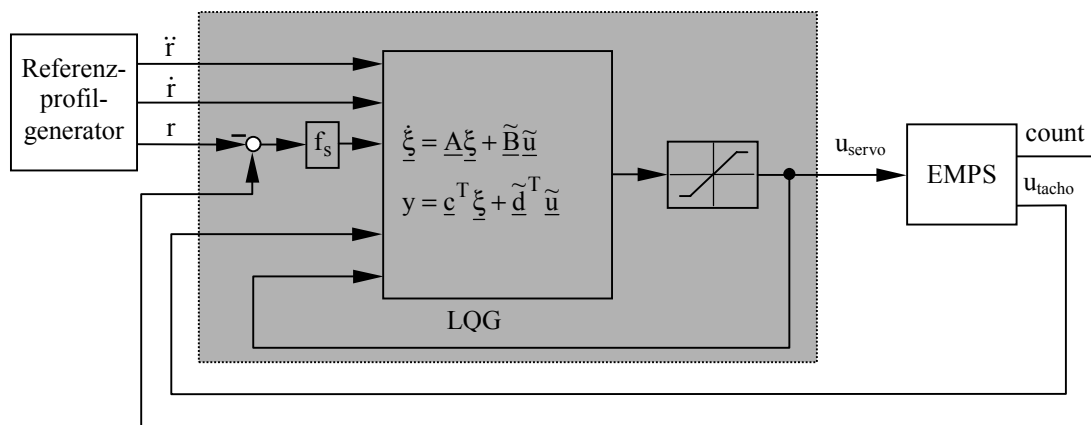


Bild 16: Regelkreis mit Kompensator in Relativform

Für die Relativform tritt als Eingang in den Kompensator anstelle der absoluten Soll- und Istlage der Lagefehler. Der Maximalwert des Lagefehlers ist zusätzlich zur Erhöhung der Rechengenauigkeit auf den Zahlenbereich der Zielhardware hochskaliert. Entsprechend verkleinern sich die zugehörigen Reglerverstärkungen. Die Rechnung der Kompensatorgleichungen kann vollständig mit 16-Bit-Arithmetik erfolgen. Allein der hochskalierte Lagefehler wird durch Subtraktion zweier 32-Bit Variablen gebildet und als 16-Bit Variable gespeichert.

An die Stelle des Zustandsvektors \underline{x} mit absoluter Lageinformation aus Bild 15 tritt ein relativer Zustandsvektor $\underline{\xi}$, der sich um einen aus der Sollgeschwindigkeit \dot{r} gebildeten, gleitenden Mittelwert bewegt.

Die Transformation der Kompensatorgleichungen auf Relativform erfolgt in MATLAB mit Hilfe einer Funktion aus der CLM-Funktionsbibliothek.

Weitere Schritte für eine gute Implementierung des Kompensators mit Festkomma-Arithmetik sind die Transformation der Kompensatorgleichungen auf reelle Modalform und die Diskretisierung. Das am besten geeignete Diskretisierungsverfahren und die größtmögliche Abtastperiode wird mit Hilfe der Entwurfssuite ermittelt.

Der diskrete Kompensator kann nun in die Entwicklungsumgebung TargetLink [5] für die Festkomma-Codegenerierung importiert werden. Die Blöcke für den Gleitkomma-Prototyp werden per Knopfdruck durch entsprechende Festkommablöcke ersetzt. Bild 17 zeigt das Resultat dieser automatischen Konvertierung. Der Zustandsblock enthält die diskreten Zustandsgleichungen des Kompensators in Relativform.

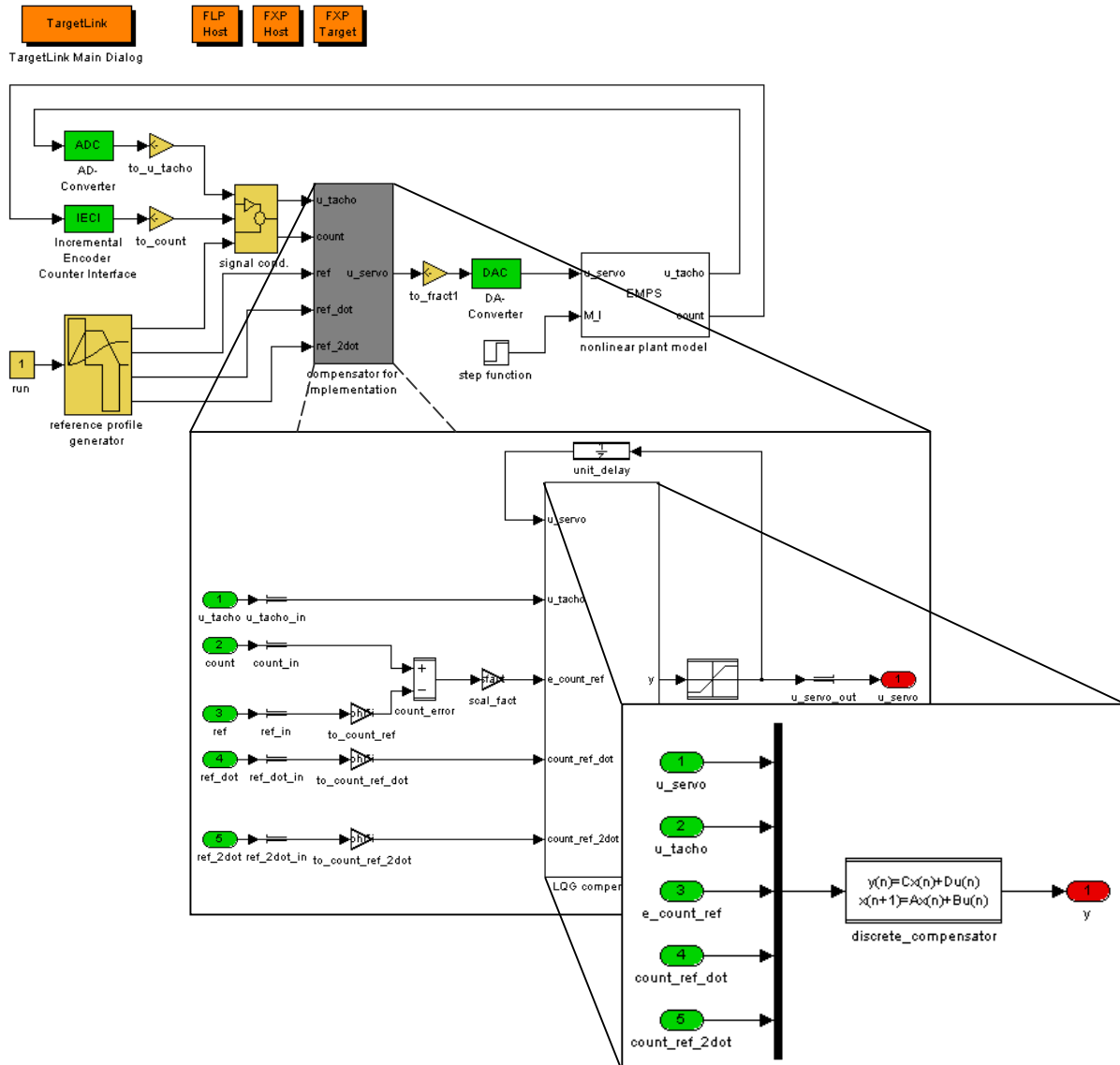


Bild 17: Simulink-Modell mit Festkommaspezifikation des Kompensators

In dieser Festkommaspezifikation des Kompensators sind für die einzelnen Festkommablöcke weitere Angaben über die Datentypen (Default ist 16-Bit Integer) und Skalierungsfaktoren für die Ausgangsgrößen erforderlich. Für Eingangsgrößen gelten die Angaben zu den Ausgangsgrößen der vorgeschalteten Blöcke. Die entsprechenden Skalierungen der Größen und Koeffizienten werden automatisch durchgeführt.

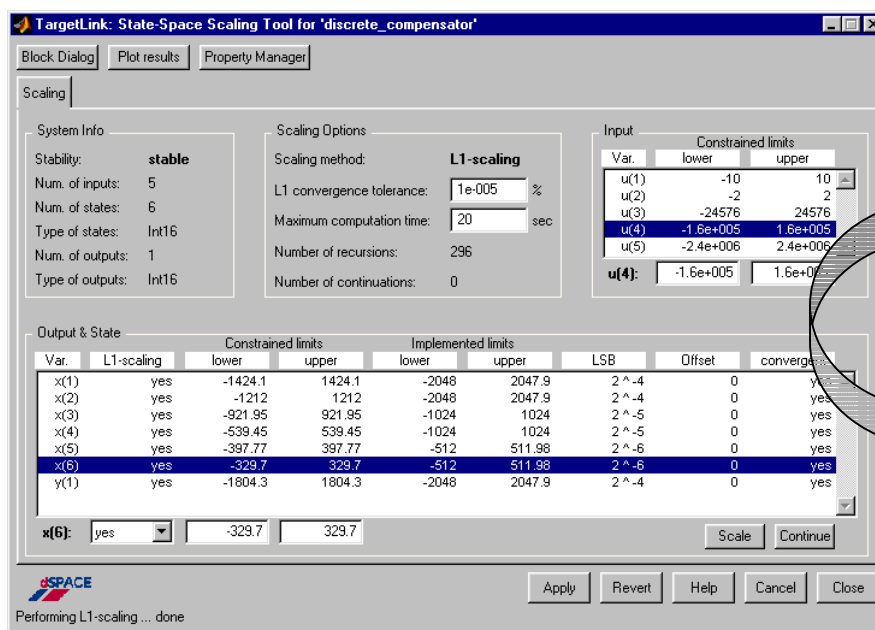
Für die einfachen Blöcke wie Inports, Gains, Summierer, Unit-Delay und Saturation erhält man den Skalierungsfaktor im einfachsten Fall aus der Beziehung

$$y = S_y y'$$

Darin ist y die ursprüngliche physikalische Ausgangsgröße (z.B. eine Spannung mit einem Aussteuerbereich von ± 10 V) und y' das zugehörige Integer-Äquivalent (z.B. mit einem Zahlenbereich von $-32768 \dots 32767$ für 16-Bit Integer als Datentyp). Der Skalierungsfaktor S_y entspricht dem physikalischen Wert des LSB des Integer-Äquivalents. Er wird so gewählt, dass der Zahlenbereich des Integer-Äquivalents den Aussteuerbereich der physikalischen Größe vollständig abdeckt. Dabei sind wegen einer effizienteren Rechnung für den Skalierungsfaktor Zweierpotenzen vorzuziehen. Für das oben genannte Zahlenbeispiel wurde der Skalierungsfaktor $S_y = 2^{-11}$ V gewählt, der für die physikalische Größe y einen Aussteuerbereich von -16 V ... 15.999512 V erlaubt.

Für dynamische Subsysteme, wie den Zustandsblock für den Kompensator, gestaltet sich die Berechnung der Skalierungsfaktoren nicht so einfach. Hier sind alle Zustands- und Ausgangsgrößen auf den Zahlenbereich des gewählten Festkomma-Datentyps zu skalieren. Deren Aussteuerbereiche sind aufgrund der Dynamik des Subsystems zunächst nicht bekannt.

Zur Bestimmung der Aussteuerbereiche der Variablen dynamischer Systeme und der zugehörigen Skalierungsfaktoren wurde daher im CLM ein Skalierungswerkzeug entwickelt und in die Entwicklungsumgebung TargetLink integriert. Das Werkzeug erhält man nach Doppelklick auf den betrachteten Block in einem Untermenü. Bild 18 zeigt die zugehörige Dialogbox.



Vorgabe der Aussteuerbereiche der Eingangsgrößen

worst-case Aussteuerbereiche und Skalierungsparameter für Zustands- und Ausgangsgrößen

Bild 18: Dialogbox des Skalierungstools

Nach Vorgabe der bekannten Aussteuerbereiche der Eingangsgrößen ermittelt das Werkzeug automatisch die worst-case Aussteuerbereiche der Zustands- und Ausgangsgrößen. Mit den Grenzen der Aussteuerbereiche werden die für die gewählten Datentypen erforderlichen Skalierungsfaktoren ermittelt und die Blockgleichungen automatisch skaliert. Weitere Details zum Skalierungstool sind in [6] nachzulesen.

Nach Angabe der Skalierungsinformationen (Datentypen, Skalierungsfaktoren) und automatischer Skalierung der Blockgleichungen erfolgt nun wie beim Rapid Control Prototyping die Inbetriebnahme des Kompensators durch automatische Codegenerierung, jetzt allerdings für die Zielhardware mit Festkomma-Arithmetik, und der Test der Regelung,

zunächst in der Simulation, dann im Experiment. Bild 19 und Bild 20 zeigen Ergebnisse mit dem generierten Festkomma-Code aus einer Prozessor-in-the-Loop-Simulation. Prozessor-in-the-Loop heißt, dass der Code auf einem Evaluierungsboard für den Zielprozessor im geschlossenen Kreis mit der Simulation der nichtlinearen Regelstrecke betrieben wird.

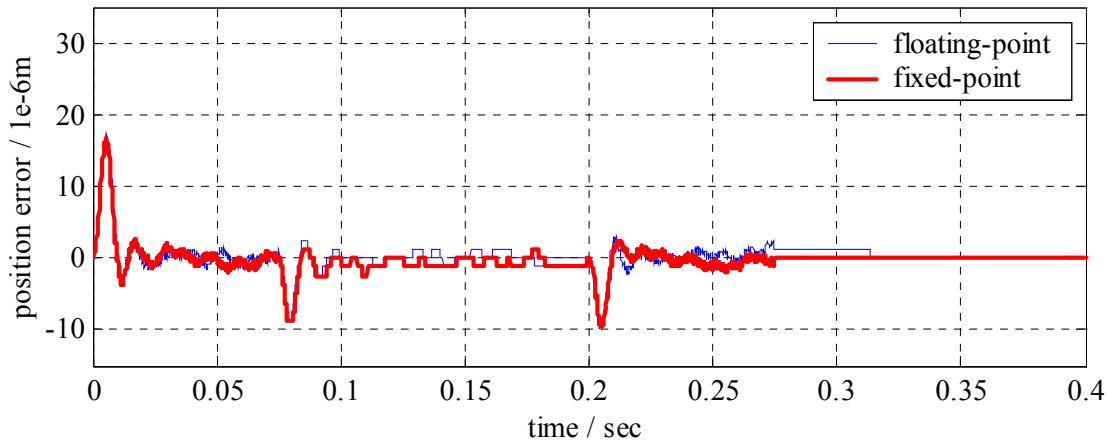


Bild 19: Führungsantwort mit Festkomma-Arithmetik

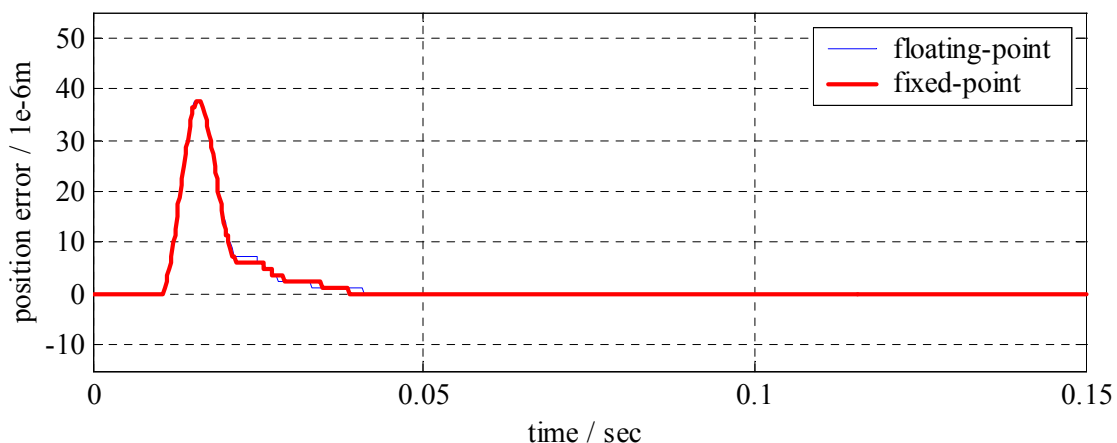


Bild 20: Störantwort mit Festkomma-Arithmetik

Im Vergleich mit den Ergebnissen für den Gleitkomma-Prototyp der Regelung bestehen kaum Unterschiede. Solche Ergebnisse mit Festkomma-Arithmetik sind nicht selbstverständlich. Sie sind allein auf die vorgestellte Vorgehensweise von der Aufbereitung des Kompensators bis zu Codegenerierung zurückzuführen. Dabei spielt der vorgestellte Werkzeugeinsatz für die effiziente und fehlerfreie Durchführung der erforderlichen Schritte eine wesentliche Rolle.

Zusammenfassung

Am Beispiel einer Regelung für ein elektromechanisches Positioniersystem wurde der gesamte Entwicklungsprozess, von der Aufgabenstellung bis zum Produktionscode der Regelung für ein Festkomma-Steuergerät, durchlaufen.

Die dem mechatronischen Entwicklungskreislauf folgende Methodik und der Einsatz einer zugeschnittenen Werkzeugkette führen in kürzest möglicher Zeit zu einem optimalen Prototyp. Dieser Prototyp dient als Referenz für die anschließende Weiterentwicklung der Regelung zur Serienreife. In diesem Zusammenhang spielt die zusammen mit dem Prototyp

entwickelte Entwurfsuite eine wichtige Rolle. Sie ermöglicht den Transfer des Wissens über den Entwurf und die Realisierung der Regelung und stellt gleichzeitig eine ausführbare Spezifikation des entworfenen Systems dar.

Aus dem Themenkreis der Weiterentwicklung des Gesamtsystems zur Serienreife wurde die Entwicklung des Steuergerätes für die Regelung beschrieben. Ziel war eine gleichwertige Implementierung der Regelung mit Festkomma-Arithmetik. Hierbei wurde gezeigt, dass durch die systematische Aufbereitung der Reglergleichungen und eine entsprechende Werkzeugunterstützung die Erzeugung von Festkomma-Code für Steuergeräte genauso beherrschbar ist, wie das Rapid Control Prototyping mit leistungsfähiger Gleitkomma-Hardware. Die Festkommaimplementierung zeigt nahezu übereinstimmende Ergebnisse.

Der Beitrag macht deutlich, dass sowohl die Entwicklung eines optimalen Prototyps als auch die erfolgreiche Weiterentwicklung einer Regelung zur Serienreife ähnliche Anforderungen an Methodenwissen und Werkzeugeinsatz stellen. Die vorgestellte Werkzeugkette stellt eine Erweiterung der am Markt eingeführten Entwicklungsumgebungen dar und ermöglicht es auch dem weniger geübten Anwender, umfassende Entwicklungsaufgaben zu bearbeiten.

Quellenverzeichnis

- [1] H. Henrichfreise: *Prototyping of LQG-Compensator for a Compliant Mechanical Drive System with Friction*. 1. Workshop TransMechatronik - Entwicklung und Transfer von Entwicklungssystemen der Mechatronik, HNI-Verlagsschriftenreihe, Band 23, 1. Edition, Hrsg: Jürgen Gausemeier, Paderborn 1997. Aufsatz verfügbar über die CLM-Homepage <http://www.clm-online.de/>.
- [2] H. Henrichfreise: *Regelung von Servoantriebssystemen mit Elastizität und Reibung*. Seminarunterlagen, DMecS Development of Mechatronic Systems GmbH & Co. KG, Köln, 2001.
- [3] H. Hanselmann: *Low-Resolution Implementation of High Resolution Position Control*. IEEE Transaction on Automatic Control, Vol. 33, No. 11, S. 1074-1078, 1988.
- [4] R. Starbek, H. Henrichfreise und U. Kiffmeier: *Skalierung dynamischer Systeme zur automatischen Codegenerierung für Festkommaprozessoren*. Labor für Mechatronik, FH-Köln, 2001.
- [5] U. Kiffmeier, L. Köster, M. Meyer und C. Witte: *Automatische Generierung von Produktionscode für Seriensteuergeräte*. Automatisierungstechnik 47 (1999), Heft 7, S. 295-304.
- [6] dSPACE: *TargetLink Production Code Generation Guide*. dSPACE GmbH, Paderborn, 2001.