



Technische Universität München
Lehrstuhl für Bildverstehen
und autonome Systeme

Bachelorarbeit

Admittanz-Regelung einer omnidirektionalen Roboterbasis

Tobias Fuchs

DRAFT

Inhaltsverzeichnis

1	Einleitung	4
1.1	Problemstellung und Motivation	4
1.2	Zielsetzung	4
2	Grundlagen	6
2.1	Einführung in die Regelungstechnik	6
2.1.1	Regelkreise	6
2.1.2	Lineare Regelstrecken	7
2.2	Mathematische Grundlagen	8
2.2.1	Funktionaltransformationen	8
2.2.2	Numerische Differentiation, Integration und Faltung	8
2.3	Digitale Filter	10
2.3.1	Filter mit unendlicher Impulsantwort (IIR)	10
2.3.2	Kalman-Filter	11
2.4	Systemdynamische Bausteine	12
2.4.1	P-Systeme	13
2.4.2	I-Systeme	14
2.4.3	D-Systeme	14
2.4.4	PT_n Systeme	14
2.4.5	Totzeit-Glieder	16
2.4.6	Rechnen mit Übertragungsfunktionen	17
2.5	Reglertypen	18
2.6	Methoden der Systemidentifikation	19
3	Stand der Technik	21
3.1	Konkrete Problemstellung am Lehrstuhl Radig	21
3.1.1	Mecanum Wheels	21
3.1.2	Kommunikation mit der Basis	22
3.2	Interaktive Regelung	23
3.2.1	Impedanz- und Admittanzregelung	24
4	Experimentelle Systemidentifikation	25
4.1	Charakterisierung der Störgrößen	25
4.2	Versuchsaufbau	26
4.3	Bestimmung eines linearen, zeitinvarianten Systems (LTI)	27
5	Analytische Systemidentifikation	31
5.1	Modellierung physikalischer Systeme	31
5.2	Modellierung des Systems	31
5.2.1	Modellierung der Trägheit	31
5.2.2	Modellierung des Rollwiderstands	32
5.2.3	Modellierung des Getriebes	36
5.2.4	Klassifizierung des Modells	36
5.3	Simulation und Validierung	37
6	Umsetzung	40
6.1	Umsetzung einer Geschwindigkeitsregelung	40
6.1.1	Verwendung des inversen Modells	40
6.1.2	Parametrierung durch Einstellregeln	40

6.2	Entwurf des Admittanzreglers	42
6.2.1	Grundlage der Stellregeln	42
6.2.2	Besonderheiten des Regelkreises	43
6.2.3	Entwurf einer Impedanzregelung	43
6.2.4	Entwurf einer Admittanzregelung	45
6.3	Vorbereitung eines Echtzeit-Betriebssystems	45
6.3.1	Notwendigkeit einer Echtzeit-Umgebung	45
6.3.2	RTAI	46
6.4	Interprozesskommunikation unter RTAI	46
7	Bewertung und Ausblick	50

1 Einleitung

Am Lehrstuhl für Bildverstehen und wissensbasierte Systeme der Technischen Universität München werden im Zusammenhang mit der Forschungsinitiative CoTeSys (Cognition for Technical Systems) Roboter entwickelt, die sich teilweise selbsttätig in die Arbeits- und Lebensumgebung von Menschen eingliedern sollen.

Ziel ist die Entwicklung von Systemen, die nicht ausschließlich nach einem fest programmierten Plan arbeiten, sondern laufend Informationen aus ihrer Umwelt in ihre Aktionen mit einbeziehen.

Hierfür gibt es bereits Anwendungsfälle aus der industriellen Fertigung, wie beispielsweise CNC-Fräsroboter, die auf den Bruch ihres Bohrers reagieren und Hilfe anfordern.

Vor besondere Herausforderungen werden derartige Systeme allerdings gestellt, wenn sie in den Lebensalltag eingebunden werden sollen, um Menschen bei bestimmten Tätigkeiten zu unterstützen.

Anders als in einer Produktionshalle sollen sich Roboter nicht nur weitestgehend autonom, sondern auch intuitiv verhalten, so dass ein Anwender keine technische Vorerfahrung benötigt um mit ihnen zu interagieren.

Nicht zuletzt deshalb beschäftigen sich nicht nur Informatiker und Elektrotechniker mit diesen Aufgabenstellungen, sondern im Fall von CoTeSys auch Psychologen, vorwiegend aus dem Bereich der Verhaltenspsychologie. Deren Untersuchungen ergeben die genauen Anforderungen an das Verhalten eines solchen autonomen Robotersystems. Bekannt ist, dass durch die Simulation von Verhalten, das Menschen bereits aus ihren alltäglichen Erfahrungen nachvollziehen können, die Akzeptanz eines Roboters im Lebensumfeld erhöht und der Umgang mit ihnen im Idealfall zur Gewohnheit wird.

1.1 Problemstellung und Motivation

Auch das in dieser Arbeit behandelte Problem hat seinen Ursprung in der Forderung nach intuitivem Umgang mit autonomen Systemen und der Simulation von erwartetem Verhalten. Steht ein Roboter im Weg während er einen Plan abarbeitet, so ist die intuitive Reaktion des Menschen, ihn auf die Seite schieben zu wollen.

Bei einer mobilen Basis von über 250 Kilogramm wird ihm dies allerdings nur mit Mühe gelingen - besonders, wenn man berücksichtigt, dass vor allem Pflegebedürftige mit dem System umgehen können sollen.

Die Basis mit einer Fernsteuerung auf die Seite zu bewegen ist natürlich möglich, allerdings setzt dies für den Menschen neu zu erlernende technische Schnittstellen voraus. Die Bewegung über einen Joystick fordert ein vergleichbar hohes Abstraktionsvermögen und verkompliziert die Bedienung.

1.2 Zielsetzung

Gewünscht ist die aktive Unterstützung der Roboterbasis, wenn ein Anwender sie durch Anschieben bewegen will. Der hierfür benötigte Kraftaufwand muss möglichst gering sein, so dass dies auch für pflegebedürftige Personen ohne große Anstrengung möglich ist. Es wird somit angestrebt, die eigentliche Masse der Basis zu verbergen.

Im Folgenden soll ausgeführt werden, wie diese Anforderungen mit Methoden der Regelungstechnik und Echtzeit- Softwareanwendungen erfüllt werden können.

2 Grundlagen

Das Verständnis für die Grundlagen der Regelungstechnik und des Systembegriffs ist notwendig, um die gegebenen Problemstellungen und die Motivation hinter den erarbeiteten Lösungsansätzen nachvollziehen zu können.

In diesem Kapitel werden einerseits später verwendete Begriffe definiert, andererseits auch Methoden der Analyse und Modellierung von Systemen im Ansatz erläutert.

2.1 Einführung in die Regelungstechnik

Die Regelungstechnik ist eine system- und informationsorientierte Ingenieursdisziplin. Sie befasst sich mit dem Verständnis und der gezielten Beeinflussung realer Systeme, wie zum Beispiel der Einhaltung einer bestimmten Raumtemperatur, oder der Geschwindigkeit eines Fahrzeugs.

Systemorientiert bezieht sich in diesem Zusammenhang auf die mathematische Modellierung eines Problems, um es auf universelle Grundprinzipien zu reduzieren. Sie ist zudem informationsorientiert, da sie durch physikalische und energetische Fragestellungen bestimmt wird. Ziel ist die Automatisierung von Prozessen aus den unterschiedlichsten Anwendungsbereichen. Dies begründet den grundsätzlichen Unterschied zur manuellen Regelung und Steuerung.

2.1.1 Regelkreise

Ein Beispiel zur manuellen Regelung ist das Fahren mit konstanter Geschwindigkeit. Bedingt durch Störungen, wie Wind und Steigung der Fahrbahn, genügt es nicht, das Gaspedal auf einer festen Position zu halten - die wäre eine manuelle Steuerung. Stattdessen muss bei regelmäßiger Prüfung des Tachos mehr oder weniger beschleunigt werden. Abbildung 2.1 zeigt dieses Beispiel als Regelkreis.

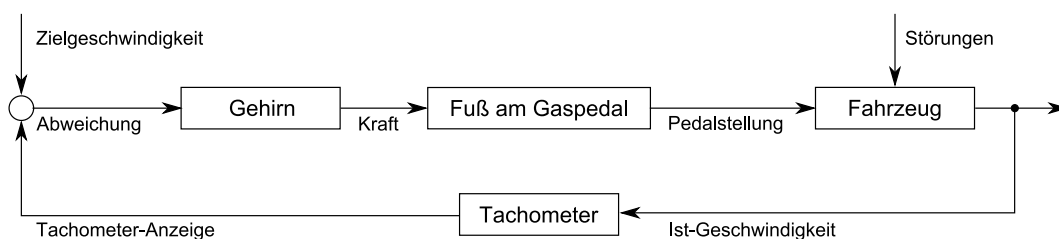


Abbildung 2.1: Einfaches Beispiel eines geschlossenen Regelkreises

Anhand dieses Beispiels können die entsprechenden Begriffe aus der Regelungstechnik eingeführt werden.

Aufgabengröße Die eigentlich zu regelnde Größe (hier: Geschwindigkeit des Fahrzeugs)

Führungsgröße Der durch den Regler einzuhaltende Sollwert für die Aufgabengröße

Regelgröße Die durch den Sensor erfassbare Größe, die von der Aufgabengröße, z.B. durch Sensorrauschen, abweichen kann.

Regeldifferenz auch Regelfehler. Differenz aus Führungsgröße und Regelgröße. Im Idealfall wird sie durch den Regler auf 0 gehalten.

Aktoreingang Eingangsgröße des Stellgliedes, normalerweise Ausgangsgröße des Reglers

Stellgröße auch Aktorenausgang. Eingangsgröße und somit Beeinflussung des zu regelnden Systems

Störgrößen Nicht beeinflussbare Größen, die an verschiedenen Stellen der Regelschleife wirken können.

Abbildung 2.2 zeigt diese Elemente im Gesamtzusammenhang des Regelkreises. Es ist ersichtlich, dass die aktive Veränderung der Aufgabengröße eine Änderung der Regeldifferenz bewirkt, wodurch ein geschlossener Wirkungskreislauf entsteht. Das Vorhandensein eines derartigen Kreislaufs unterscheidet die Regelung von der Steuerung.

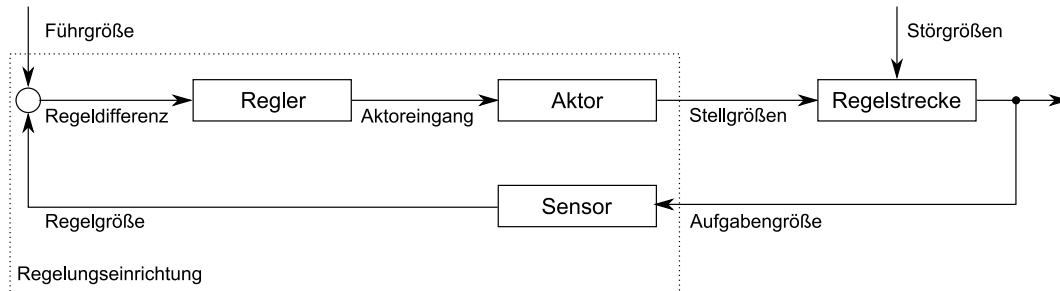


Abbildung 2.2: Aufbau eines geschlossenen Regelkreises

Dieses Regelungsprinzip der Prozessbeobachtung und -beeinflussung hat universellen Charakter, es ist also unabhängig vom Einsatzgebiet gültig. Abgesehen von technischen Aufgabenfeldern bildet zum Beispiel auch ein Hormonkreislauf oder die Verwaltung eines Aktienfonds einen Regelkreis.

Nicht zuletzt deswegen werden Systeme in der Regelungstechnik abstrakt behandelt und mit den oben benannten Begriffen beschrieben. Man spricht regelungstechnisch also von "Regelgröße", egal ob es sich um Geschwindigkeit eines Fahrzeugs oder den Wert eines Fonds handelt.

2.1.2 Lineare Regelstrecken

Vor dem Entwurf eines Reglers muss das Verhalten der Regelstrecke, also des zu regelnden Systems, hinreichend bekannt sein.

Im Allgemeinen sind hierbei Zusammenhänge zwischen Eingangs- und Ausgangsgrößen von Interesse. Diese lassen sich als Signale betrachten, was die mathematische Abstraktion des Regelungsproblems erleichtert.

Die Beschreibung des Systemverhaltens wird allgemein *Modellierung* genannt und in einem eigenen Kapitel behandelt.

Grundsätzlich ist für die Modellierung von Interesse, ob es sich um ein sogenanntes lineares, zeitinvariantes (engl.: "Linear, time-invariant", LTI) System handelt oder als solches angenähert werden kann.

Ein System gilt als zeitinvariant, wenn sich die für seine Modellierung bedeutsamen Charakteristiken für den Zeitraum der Anwendung des Modells nicht ändern. LTI-Systeme können im Allgemeinen durch lineare Differentialgleichungen mit konstanten Koeffizienten beschrieben werden. Diese können zu Differenzgleichungen umgeformt werden, die quasi direkt für die diskrete Umsetzung des Modells verwendet werden können.

Allgemein ist ein System *linear*, wenn es sein Ausgabeverhalten durch proportionale Zusammenhänge und dem Superpositionsprinzip beschrieben werden kann, und somit gilt:

$$a \cdot G(x) + b \cdot G(y) = a \cdot b \cdot G(x + y)$$

Zeitliche Invarianz eines Systems besagt, dass es zu jedem Zeitpunkt dasselbe Verhalten zeigt, sich die Parameter seiner mathematischen Beschreibung also nicht ändern. Vereinfacht ausgedrückt ist ein System *zeitinvariant*, wenn sein Verhalten zeitdeterministisch ist,

eine spezielle Eingangsfunktion also immer dieselbe Systemantwort liefert.

Tatsächlich sind physikalischen Systeme fast nie wirklich linear: Verschiedene Formen der Reibung und Hysterese-Effekte sind meistens vorhanden und stellen Streckenglieder dar, die sich nicht linear beschreiben lassen. Oft können diese ignoriert und eine gute lineare Annäherung der Strecke gebildet werden. Ist dies nicht der Fall, so behilft man sich unterschiedlicher Mittel. Zur Behandlung von Haftreibung werden beispielsweise oft Lookup-Tabellen oder abschnittsweise definierte Funktionen verwendet, um sie im Modell behandeln zu können.

2.2 Mathematische Grundlagen

In diesem Abschnitt werden kurz mathematische Grundlagen wiederholt, die zur Herleitung der Methoden dieser Arbeit verwendet wurden. Von Bedeutung ist auch die numerische Umsetzung üblicher mathematischer Verfahren, auf die gesondert eingegangen wird.

2.2.1 Funktionaltransformationen

Die *Laplace-Transformation* ist die wichtigste im Laufe dieser Arbeit verwendete Funktionaltransformation. Bekanntlich diese eine lineare Integraltransformation dar, die einer Originalfunktion im Zeitbereich $f(t)$ eine eindeutige Bildfunktion im Frequenzbereich $F(s)$ zuordnet (Unb08, S. 51):

$$F(s) = \mathcal{L}\{f\}(t) = \int_0^{+\infty} f(t) \cdot e^{-st} dt, \quad s \in \mathbb{C} \quad (2.1)$$

Diese Übertragung ist durch die *Inverse Laplace-Transformation* umkehrbar. Sie wird durch ein komplexes Linienintegral beschrieben:

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi i} \cdot \lim_{T \rightarrow \infty} \int_{\gamma-iT}^{\gamma+iT} F(s) \cdot e^{st} ds, \quad s \in \mathbb{C}, \gamma \in \mathbb{R} \quad (2.2)$$

Diese Rücktransformation ist für die Implementierung eines diskreten Regelkreises von besonderer Bedeutung, da Systeme oftmals im Frequenzbereich modelliert, aber im Zeitbereich behandelt werden.

Ihre Berechnung ist allerdings weit aufwändiger als die der Laplace-Transformierten. Sie hängt unter anderem auch von den Polstellen der Bildfunktion ab, die gegebenenfalls erst über den Residuensatz behandelt werden müssen (Unb08, S. 60).

2.2.2 Numerische Differentiation, Integration und Faltung

Bei der numerischen Umsetzung mathematischer Operationen entstehen neue konzeptionelle Probleme hinsichtlich der Effizienz und Genauigkeit der entsprechenden Algorithmen. Im Folgenden werden Lösungsansätze gezeigt, mit denen die für die Umsetzung dieser Arbeit bedeutendsten Verfahren numerisch realisiert werden können.

Numerische Differentiation

Zur Auswertung verschiedener Experimente, und nicht zuletzt für die Implementierung eines D-Gliedes, besteht der Bedarf, die Ableitung einer Funktion zu bilden.

Dies ist insofern problematisch, als dass es sich im digitalen Umfeld hierbei um diskrete Funktionen handelt, über die außer dem Rauschverhalten keine Vorkenntnisse vorhanden sind. Sie können daher nur numerisch differenziert werden. Bekanntlich ist die einfachste

Lösung hierbei ein Differentiationsoperator in der Form von Newton's Differentialquotienten als Vorwärts-Differenzenquotient ¹ über die Sekante durch $f(t-h)$ und $f(t)$. Wählt man das Ableitungsgebiet h groß genug, so wird ein einfacher, der Ableitung vorgeschaltener Tiefpass erster Ordnung realisiert. Allerdings erhöht sich die Latenz der Ableitung proportional mit der Größe - und somit Dauer - des Gebiets. Durch eine Gewichtung über das Ableitungsintervall an je fünf Stellen der Funktion kann das Rauschverhalten mit kleineren Werten für h verbessert werden:

$$(f[t])' \approx \frac{-(f[t+2h]) + 8(f[t+h]) - 8(f[t-h]) + (f[t-2h])}{12h}$$

In Echtzeit-Anwendungen wie einem Regelkreis kann allerdings nur auf Werte der Vergangenheit zurückgegriffen werden (Kausalität). Daher wird die Gleichung umgeformt:

$$(f[t])' \approx \frac{-(f[t]) + 8(f[t-h]) - 8(f[t-3h]) + (f[t-4h])}{12h}$$

Es gibt eine Vielzahl weiterer Methoden, neben der 5-Punkt-Ableitung wurde im Laufe der Arbeit aber nur die Ableitung durch einen Kalman-Filter verwendet.

Numerische Integration

Ein numerisches Verfahren zur Integration mittels polynomialer Interpolation lässt sich über die Quadraturformel ² herleiten:

$$\int_{t_0}^t f(t) dt \approx t \sum_{j=0}^k \alpha_j \cdot f(t \cdot t_j) \quad (2.3)$$

Letztendlich wird das Integral in k *Stützstellen* t_j unterteilt, die mit einem jeweiligen Koeffizienten α_j gewichtet werden. In der einfachsten Implementierung sind die Stützstellen äquidistant und alle Koeffizienten $\alpha_j = (b-a)/k$. Dies entspricht der Intuition, dass ein Integral über die Summation von Rechtecken unter der Funktion angenähert werden kann. Bekanntlich kann diese Annäherung durch das Trapez-Verfahren verbessert werden ³:

$$\int_{t_0}^t f(t) dt \approx \frac{1}{2} \cdot T_A \cdot \sum_{j=0}^k f(t_{j-1}) + f(t_j)$$

mit T_A als konstante Abtastzeit zwischen $f(t)$ und $f(t+1)$.

Numerische Faltung

Die Faltung zweier Funktionen h und x im Zeitbereich ist definiert als (Söd99, S. 106)

$$y(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t-\tau) d\tau \quad (2.4)$$

Diese Gleichung kann direkt als Algorithmus umgesetzt werden. Man spricht dann von einer *diskreten, linearen Faltung*.

Zur effizienten numerischen Faltung gibt es zwei grundsätzliche Herangehensweisen, die in der Umsetzung dieser Arbeit verwendet wurden: Die Verkürzung des Faltungsbereichs und die sog. *schnelle Faltung* unter Zuhilfenahme der *Fast Fourier Transformation*.

¹Siehe u.a. (GW01, 235)

²Diese Herleitung findet sich u.a. in (GW01, S. 235)

³Siehe u.a. (MR04, S. 327)

```

1 t = 0.0
2 while (t < t_end) do
3     sum = 0
4     tau = 0.0
5     while tau < t do
6         sum += f(tau) * g(t-tau) * T
7         tau += T
8     end
9     print(sum)
10    t += T
11 end

```

Abbildung 2.3: Implementierung der einer diskreten, linearen Faltung in Ruby

Im ersten Fall wird der Bereich des Faltungsintegrals von $[0, t]$ verkürzt auf $[t - m, t]$, um zu vermeiden, dass der Integrationsbereich linear mit der Zeit wächst. Die Faltung wird dadurch beschleunigt, aber auch ungenauer. Bei der schnellen Faltung werden beide Funktionen über eine FFT in den Frequenzbereich überführt, wo sie zur Berechnung der Faltung nur multipliziert werden müssen. Die Komplexität sinkt damit von $O(n \cdot m)$ auf $O(2 \cdot \log(n) \cdot n)$. Dieses Verfahren ist linearen Implementierungen also hinsichtlich der Effizienz überlegen, bringt durch die Natur der FFT aber auch eine schlechtere Dynamik mit sich.

2.3 Digitale Filter

Für die notwendigen Experimente am realen System, besonders aber die Umsetzung einer digitalen Regelschleife, müssen Daten von verschiedenen Sensoren der Basis interpretiert werden. Um die Zuverlässigkeit der vorgestellten numerischen Verfahren zu erhöhen werden die Sensordaten an manchen Stellen zunächst gefiltert, um ihr Rauschverhalten zu verbessern.

In diesem Abschnitt werden verschiedene Arten digitaler Filter vorgestellt und verglichen.

2.3.1 Filter mit unendlicher Impulsantwort (IIR)

Das IIR-Filter realisiert einen einfachen Tiefpass. Er wird im Zeitbereich beschrieben durch

$$y[t] = a_1 \cdot f[t - 1] + a_2 \cdot f[t - 2] \dots a_n \cdot f[t]$$

Die Filterkoeffizienten $a_1 \dots a_n$ beschreiben seinen Frequenzgang, die Anzahl der Koeffizienten die resultierende Phasenverschiebung im diskreten Zeitbereich. In seiner einfachsten und häufigsten Implementierung verwendet ein IIR nur den vorangegangenen Messwert:

$$y[t] = a \cdot f[t - 1] + b \cdot f[t]$$

mit $a < b$ und $a + b = 1$. In der Anwendung wird er typischerweise indirekt rekursiv verwendet:

```

double value_in, value_out;
double last = 0.0;
while(value_in = f(t)) {
    value_out = k * last + (1-k) * value_in;
    // ...
    last = value_out;
}

```

Da jeder Rückgabewert eines IIR-Filters Einfluss auf alle folgenden Filterwerte hat, spricht man von einer unendlichen Impulsantwort. In der Praxis ist die Impulsantwort diskreter IIR-Filter, bedingt durch den begrenzten numerischen Darstellungsraum, aber tatsächlich

endlich.

Aus dem Implementierungsbeispiel wird auch deutlich, dass er zu einer Latenz von genau einem Abtastintervall führt.

2.3.2 Kalman-Filter

Ein Kalman-Filter ist ein Base-Filter, der als Beobachter einer Größe fungiert. Aus einem ungenauen, beispielsweise verrauschten, Eingangssignal wird rekursiv ein Zustand eines linearen, dynamischen Systems geschätzt. Anders als FIR- und IIR-Filter behilft sich ein Kalman-Filter zusätzlich eines bekannten Modells des beobachteten dynamischen Systems. Ein beliebtes Beispiel ist die Ermittlung von Position und Geschwindigkeit eines über Radar beobachteten Fahrzeugs. In der Regel werden Positionsdaten über Radar mit niedriger Frequenz und für eine stetige Geschwindigkeitsermittlung zu ungenau erfasst.

Ein entsprechend eingestellter Kalman-Filter kann durch Kenntnis des dynamischen Modells des Fahrzeugs seine Position und Geschwindigkeit mit hoher Genauigkeit schätzen. Vereinfacht ausgedrückt ist das deswegen möglich, weil die meisten theoretisch möglichen Bewegungen des Fahrzeugs aufgrund seiner Fahrdynamik ausgeschlossen werden können. So wird ein PKW bei einer Geschwindigkeit von 100 km/h nicht sprunghaft seine Fahrrichtung oder Geschwindigkeit ändern können.

Seinen rekursiven Charakter erhält der Kalman-Filter durch den permanenten Abgleich seiner letzten Schätzung und den tatsächlich gefolgten Wert. Deren Abweichungen führen zur Anpassung einer Schätzmatrix (Covarianz).

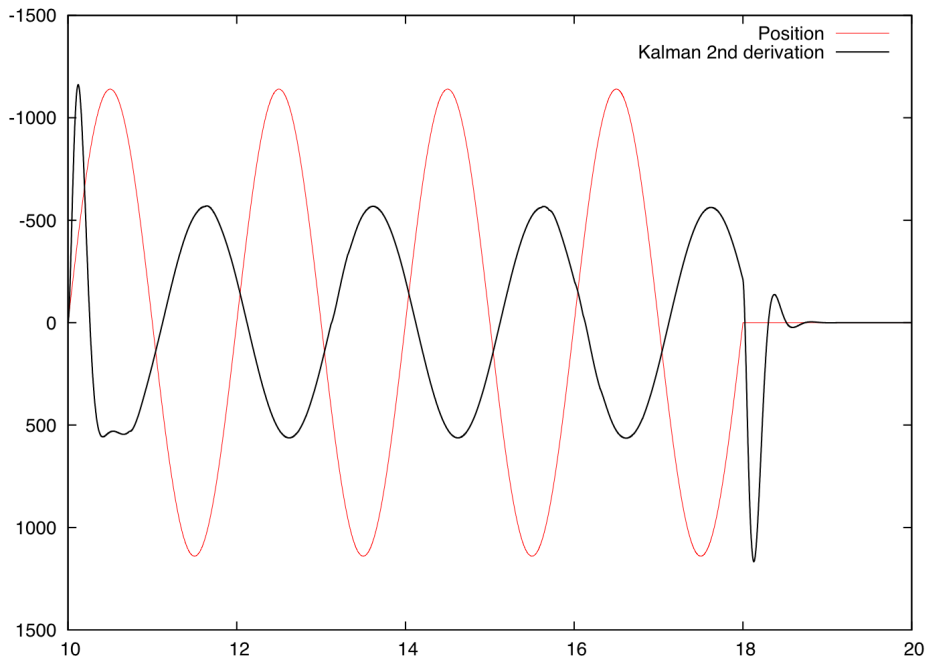


Abbildung 2.4: Provokation von Schätzfehlern eines Kalman-Filters

Daher wurde zur Systemidentifikation, beispielsweise bei der Ableitung der Geschwindigkeit aus Positionsdaten, ein vergleichsweise einfache 5-Punkt-Ableitung verwendet. Diese hat zwar ein ungünstigeres Rauschverhalten und höhere Latenz, verfälscht aber in keinem Fall die Charakteristik des Ergebnisses. Abbildung 2.4 zeigt beispielhaft das Ergebnis eines

Kalman-Filter, der für die Berechnung der zweiten Ableitung eines trägen Systems eingestellt ist. Wird dieser mit einem synthetischen, nicht-monotonen Signal konfrontiert, so sind die Schätzfehler unübersehbar. Würden die Einschwingphasen des Filters als Systemantwort aufgefasst, so hätte dies eine schwere Fehlinterpretation des Modells zur Folge.

Vergleicht man das Ergebnis mit dem der einfachen Differentiation in Abbildung 2.5, so wird aber auch das wesentlich verbesserte Rauschverhalten deutlich. Um die Stärken beider Methoden auszuspielen wurde die 5-Punkt-Differentiation zur Verarbeitung von experimentellen Daten verwendet - die schließlich manuell interpretiert werden - und die Ableitung durch ein Kalman-Filter in der Umsetzung des D-Gliedes im Regler.

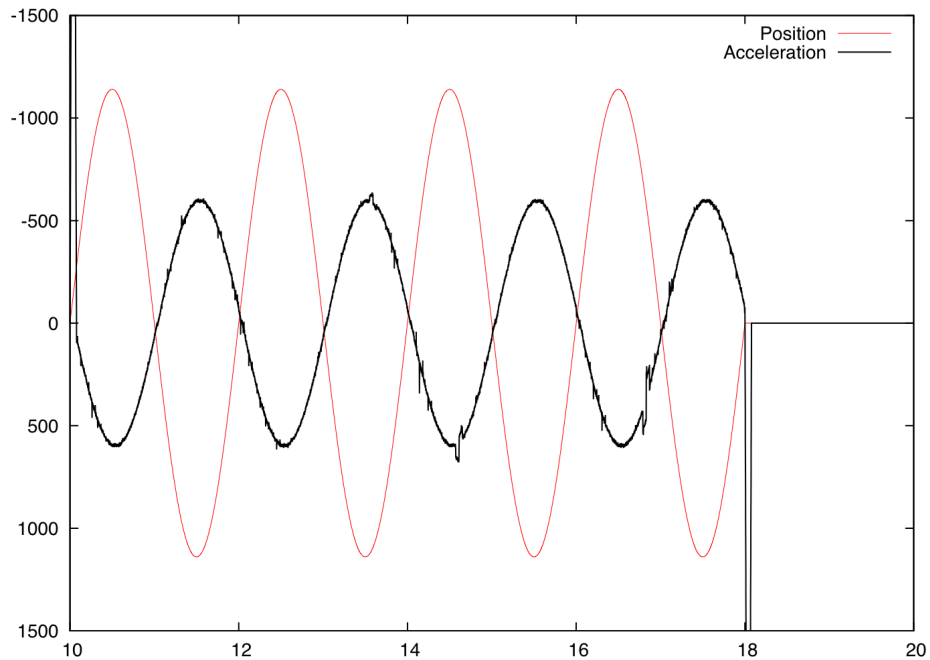


Abbildung 2.5: Bildung der zweiten Ableitung durch 5-Punkt-Methode

2.4 Systemdynamische Bausteine

Das Verhalten eines Systems wird normalerweise durch die Wirkzusammenhänge seiner einzelnen Bausteine, auch Glieder genannt, beschrieben. Ein Systembaustein beschreibt hierbei die Beeinflussung einer bestimmten Art auf Energie oder Materie, die durch das System wandern, in Abhängigkeit bestimmter Parameter, wie Geschwindigkeit oder Temperatur. Typischerweise wird das Verhalten eines systemdynamischen Bausteins auf folgende Arten beschrieben:

- Als Differentialgleichung im Zeitbereich
- Als Gewichtungsfunktion im Zeitbereich
- Als Funktion im Frequenzbereich

Dem ersten Anschein nach sind diese Darstellungsformen äquivalent. Es wird sich aber zeigen, dass es auch Fälle gibt, in denen das in der Praxis nicht der Fall ist.

Mechanische Systeme lassen sich bekanntlich gut durch Differentialgleichungen modellieren, etwa Trägheit in Abhängigkeit von der Beschleunigung, also der zweiten Ableitung der Position über die Zeit.

Weniger intuitiv ist die Beschreibung im Frequenzbereich. Hier wird eine komplexwertige Funktion $F(s)$ aufgestellt, mit der die Impulsantwort des Systems beschrieben wird. In der Literatur zur Regelungstechnik spricht man von einer Übertragungsfunktion.

Die Übertragungsfunktion, oft auch *Systemfunktion* oder *Frequenzgang* genannt, ist definiert als

$$F(s) = \frac{Y(s)}{U(s)} = \frac{\text{Wirkungsfunktion}}{\text{Ursachefunktion}} \quad (2.5)$$

mit $Y(s)$ und $U(s)$ als Laplace-Transformierte der entsprechenden diskreten Funktionen $y[t]$ und $u[t]$ im Zeitbereich. Sie beschreibt das Übertragungsverhalten eines LTI-Systems vollständig (Söd99, S. 104).

Abbildung 2.6 zeigt den Gesamtzusammenhang der Funktionen schematisch.

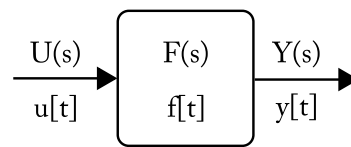


Abbildung 2.6: Schematische Darstellung von Ursache-, System- und Wirkungsfunktion

2.4.1 P-Systeme

Ein P-System (P = proportional) realisiert einen linearen Zusammenhang zwischen Eingabe- und Ausgangsgröße. Mit x_a als Ausgangs- und x_e als Eingangsgröße wird es beschrieben durch (Bus04):

$$x_a(t) = K_P \cdot x_e(t)$$

Und der Übertragungsfunktion

$$F(s) = K_P \quad (2.6)$$

Der Faktor K_P ist verstärkend für $K_P > 1$ oder dämpfend für $K_P < 1$. Abbildung 2.7 zeigt die Übergangsfunktion eines P-Systems.

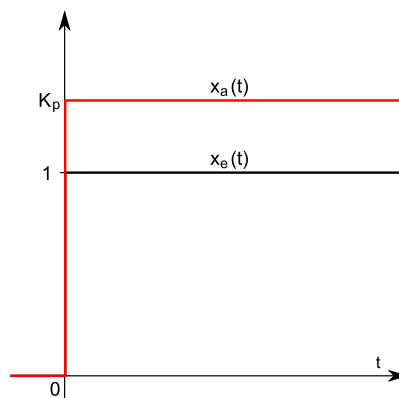


Abbildung 2.7: Übergangsfunktion P-System

Es ist leicht nachvollziehbar, dass ein P-System die Phase und Frequenz des Eingangssignals nicht beeinflusst.

2.4.2 I-Systeme

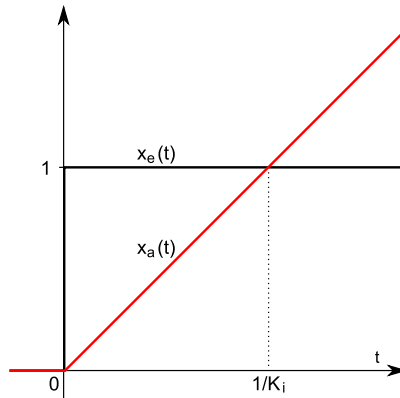


Abbildung 2.8: Übergangsfunktion I-System

Hat das Integral über die Eingangsgrößen der Vergangenheit Einfluss auf die Ausgangsgröße, dann kann dies durch ein I-System (I: Integral) beschrieben werden (Bus04).

$$x_a = K_I \cdot \int_0^t x_e(\tau) d\tau$$

Die Übertragungsfunktion lautet:

$$F(s) = \frac{K_I}{s} \quad (2.7)$$

Der Füllstand eines Tanks x_a wird beispielsweise durch die in der Vergangenheit zugeführte Flüssigkeit x_e bestimmt, und die bisherige Geschwindigkeit eines Zuges hat Einfluss auf seine aktuelle Position. Diese kann dann durch Integration über die Geschwindigkeit ermittelt werden. Die zugehörige Darstellung im Zeitbereich ist demnach (Bus04):

$$\dot{x}_a(t) = K_I \cdot x_e(t)$$

2.4.3 D-Systeme

Ein System wird als D-System (D: Differential) bezeichnet, wenn es proportional zu der *Änderung* einer Eingangsgröße reagiert.

$$x_a(t) = K_D \dot{x}_e(t)$$

mit der zugehörigen Übertragungsfunktion

$$F(s) = K_D \cdot s \quad (2.8)$$

Es existiert keine reale Entsprechung für ein reines D-System, da seine Sprungantwort einem Dirac-Impuls entspricht. Als Annäherung gelten vielleicht Piezo-Kristalle, da sie Spannung in Abhängigkeit einer Druckveränderung erzeugen. In realen Systemen treten D-Glieder aber nur in Zusammenhang mit Verzögerungen auf. Jedes physikalische System beinhaltet eine Form von Energiespeicher, so dass man D-Systeme nur in Kombination mit PT_n -Gliedern modelliert.

2.4.4 PT_n Systeme

Beinhaltet ein System beispielsweise einen Energiespeicher, so dass die Auswirkung einer Eingangsgröße verzögert eintritt, so spricht man von einem PT-System. Hierbei kann man sich ein PT_n -Glied als System mit n verketteten Energiespeichern vorstellen.

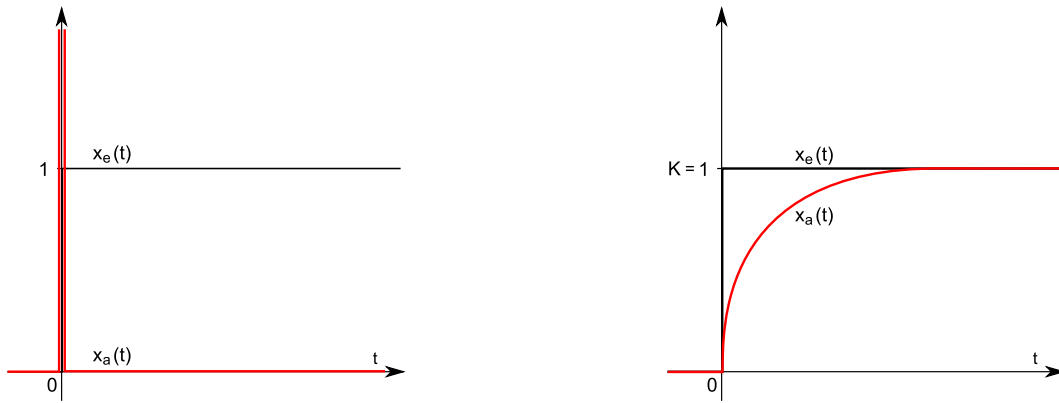


Abbildung 2.9: Übergangsfunktion D-System und PT1-System

Ein verzögerndes System wirkt träge, wie man in Abbildung 2.9 sehen kann, und stellt einen Tiefpassfilter dar.

Bei Systemen höherer Ordnung ist zusätzlich eine mögliche Schwingungsfähigkeit zu berücksichtigen.

Das in der Mechanik oft als Beispiel herangezogene Feder-Masse-Dämpfer-System besteht aus den beiden hintereinander geschalteten Energiespeichern Feder und Masse. Die der Feder zugeführte Energie x_e wird der Masse verzögert weitergegeben, deren Lage sich trägheitsbedingt wiederum verzögernd ändert. Die Übertragungsfunktion eines PT_2 -Systems wird im s -Bereich formuliert als (Bus04):

$$F(s) = K \cdot \frac{\omega_0}{s^2 + 2 \cdot D \cdot \omega_0 \cdot s + \omega_0^2} \quad (2.9)$$

Die Konstante D in dieser Gleichung ist der Dämpfungsgrad, ω_0 die Eigenfrequenz des Systems unter Vernachlässigung der Dämpfung. Seine Pole sind

$$p_{1,2} = -\omega_0 D \pm \omega_0 \sqrt{D^2 - 1}$$

Wird ein Pol des Systems komplexwertig, so ist es schwingungsfähig. Es ist instabil, wenn mindestens ein Pol einen positiven Realteil besitzt. Insgesamt gibt es drei Fälle in Abhängigkeit von D zu unterscheiden:

1. $D \leq 0$: Instabil
2. $0 < D < 1$: Periodisch stabil (schwingungsfähig)
3. $D \geq 1$: Aperiodisch stabil (nicht schwingungsfähig)

Für nicht schwingungsfähige PT_2 -Systeme ergibt sich eine typische sigmoidale Kurve, wie in Abbildung 2.10). Besonders ein derartiges System wird später von Bedeutung sein.

Ein nicht schwingungsfähiges, verzögerndes System n -ter Ordnung hat die Übertragungsfunktion

$$F_{PT_n}(s) = K \cdot \frac{1}{(1 + sT_t/n)^n} \quad (2.10)$$

Für $n = 1$ ergibt sich demnach

$$F_{PT_1}(s) = K \cdot \frac{1}{1 + sT_t} \quad (2.11)$$

PT_1 -Systeme sind niemals schwingungsfähig. Es existiert in diesem Fall nur der reelle Pol $-1/T_t$.

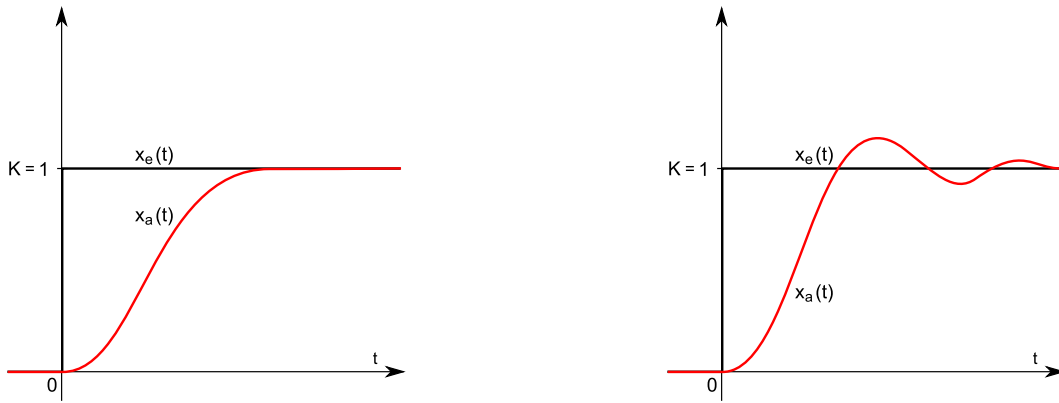


Abbildung 2.10: Übergangsfunktion nicht-schwingungsfähiges und schwingungsfähiges PT2-System

2.4.5 Totzeit-Glieder

Totzeitglieder werden gerne als Effekt eines Transportvorgangs beschrieben. Generell drückt ein Totzeitglied den Umstand aus, dass die Auswirkung eines Effekts - zum Beispiel Zufuhr eines Stoffes in einen Wasserkreislauf - erst zeitlich versetzt gemessen werden kann.

Im Unterschied zu PT-Systemen findet bei Totzeitgliedern bis zum Zeitpunkt T_u keine Änderung der Ausgangsgröße des Systems statt, also auch keine sichtbare Annäherung.

Aus mechanischer Trägheit entsteht zumindest per Definition keine Totzeit, da eine Positionsveränderung normalerweise sofort festgestellt werden kann, auch wenn sie sehr klein ist. Totzeitglieder können nur im Frequenzbereich, also nicht durch Differenzialgleichungen, dargestellt werden. Für derartige Systeme kann eine Regelung zudem nur schwer und meist unter Kompromissen in der Güte umgesetzt werden. Sie sind bei der Regelung daher äußerst unerwünscht, und es wird viel Forschungsaufwand in Verfahren investiert, die ihre Regelung verbessert und erleichtert.

Totzeit wird im Frequenzbereich formuliert als

$$F(s) = K \cdot e^{-sT_t}$$

Der Verschiebungssatz ist zu erkennen. Das ist leicht nachvollziehbar: Im Frequenzbereich bewirken reine Totzeitsysteme eine positive Phasenverschiebung.

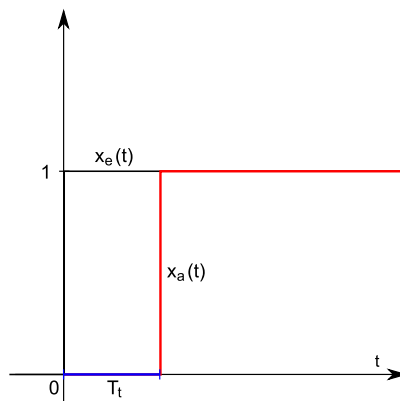


Abbildung 2.11: Übergangsfunktion T-System

Um die Analyse zu vereinfachen, nähert man Totzeitverhalten als Reihe verzögernder Glieder (PT_n) an, nach Gleichung 2.10 also

$$e^{-sT_t} \approx \frac{1}{(1 + sT_t/n)^n} \quad (2.12)$$

Das bedeutet aber auch: Selbst wenn PT2-Systeme per theoretischer Definition keine echte Totzeit erzeugen, so können sie doch derart verzögern, dass man in der Praxis von Totzeit sprechen muss.

Bedeutsam ist die genaue Art der Verzögerungen. Diese klassifiziert man durch die Wendetangente in der Sprungantwort, wobei die erste Verzögerung als Verzugszeit T_u , und die zweite als Ausgleichszeit T_g bezeichnet wird.

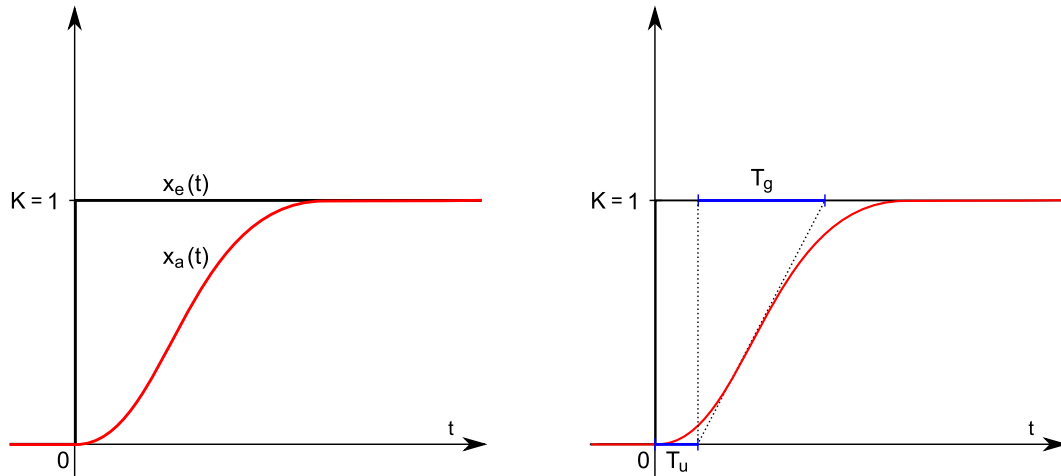


Abbildung 2.12: Klassifizierung von PT2-Systemen nach Ausgleichszeit und Verzugszeit

Die Regelbarkeit von Strecken höherer Ordnung wird entscheidend vom Verhältnis Ausgleichszeit zu Verzugszeit bestimmt. Je größer dieses Verhältnis, umso besser ist die Strecke zu regeln. Bei einer Strecke 1. Ordnung ist T_u gleich Null und somit das Verhältnis T_g/T_u unendlich groß. Diese Strecke ist tatsächlich stabil und mit einfachen Mitteln zu regeln. Im Zuge der Modellreduktion wird daher oft untersucht, ob ein Verzögerungssystem zweiter Ordnung auch in erster Ordnung angenähert werden kann. Für Werte von T_g/T_u einer Strecke gilt die Klassifizierung (Bög07, Sektion R13):

> 5 gut regelbar

2.5 - 5 mäßig regelbar

1.2 - 2.5 schlecht regelbar

< 1.2 sehr schlecht regelbar

Liegt der Quotient unter 2.5, so sollte T_u als Totzeit behandelt werden.

2.4.6 Rechnen mit Übertragungsfunktionen

Die vorgestellten Komponenten treten in einem System in verschiedenen Kombinationen auf. Vor allem in linearen Systemen, in denen das Superpositionsprinzip gilt, ist ein Systemverhalten oft die Summe verschiedener systemdynamischer Bauteile, deren Wirkung als Parallelschaltung dargestellt wird.

Für die Summe zweier Glieder $G_1(s)$ und $G_2(s)$ gilt (Unb08, S. 74):

$$Y(s) = X_a(s) = G_1(s) + G_2(s) \quad (2.13)$$

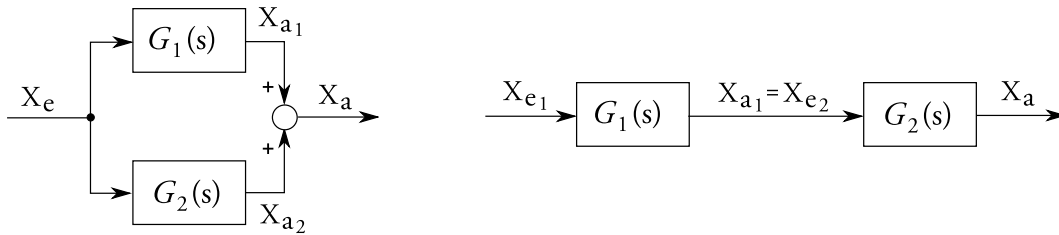


Abbildung 2.13: Parallelschaltung und Verkettung zweier systemdynamischer Bausteine

Wirken diese Komponenten nicht gemeinsam sondern nacheinander, so werden sie verkettet und es gilt (Unb08, S. 74):

$$Y(s) = X_a(s) = G_1(s) \cdot G_2(s) \quad (2.14)$$

Das Produkt zweier Funktionen im s -Bereich entspricht deren Faltung im Zeitbereich (Unb08, S. 58). Diese ist, wie die Multiplikation, kommutativ.

Ist die Gesamtübertragungsfunktion F des Systems bekannt, so liefert sie nach Faltung mit einer Eingangsfunktion eine entsprechende Systemantwort, denn nach Gleichung 2.14 und 2.5 gilt:

$$Y(s) = F(s) \cdot U(s) \quad (2.15)$$

Das Ergebnis der Faltung mit der Heaviside-Sprungfunktion θ bezeichnet man als *Übertragungsfunktion*.

Wird die Übertragungsfunktion durch die inverse Laplace-Transformation in den Zeitbereich überführt, so kann die Faltung auch numerisch geschehen. Dadurch wird Simulationen und Validierung des Modells durch Software erleichtert.

2.5 Reglertypen

Regler werden unter verschiedenen Gesichtspunkten klassifiziert. Diese sind beispielsweise⁴:

Implementierung Analoge Regler realisieren einen kontinuierlichen Regelkreis, im Gegensatz zu digitalen Reglern, die über eine ausreichend hochfrequente Abtastung ein diskretes, quasikontinuierliches Verhalten erreichen.

Stellgrößenraum Liegen die möglichen Stellgrößen in einem kleinen, diskreten Raum (beispielsweise "an" und "aus"), so spricht man von einer *ereignisdiskreten Steuerung*⁵.

Adaptionsfähigkeit Adaptive Regler sind in der Lage, beispielsweise über die selbstständige Änderung der Zielgrößen, ihr Regelungsverhalten der Systemumgebung anzupassen.

Führungsgrößenverlauf als *Festwert-* oder *Folgewertregelung* und deren Mischformen.

Allgemein kann ein Regler auch als System im vorgestellten Sinn verstanden werden. Analog zu den vorgestellten Arten von Systemen existieren Reglertypen, die gezielt ein entsprechendes Systemverhalten realisieren. Der Regler selbst stellt somit selbst ein System dar, das aus den beschriebenen Gliedern besteht.

Mit einfachen linearen Reglertypen (P, I und D - Komponenten) wird normalerweise versucht, das Verhalten des zu regelnden Modells zu kompensieren, um eine neutrale Übertragungsfunktion $F(s) = 1$ bestmöglich anzunähern. Beispielsweise könnte man ein I-System mit einem D-Regler ausgleichen. Im Idealfall wird ein Regler nach dem exakten inverse Modell des Systems parametrisiert.

⁴Siehe u.a. (Bus04)

⁵Hierzu (Bus04, 1-15)

Die reine Modellinvertierung ist = in der Theorie - trivial: Das invertierte Modell ist die Umkehrfunktion des Vorwärtsmodells. Wie man das Vorwärtsmodell eines realen, physikalischen Systems ermitteln kann, wird im folgenden Abschnitt beschrieben.

2.6 Methoden der Systemidentifikation

Um jegliche reale Vorgänge gezielt beeinflussen zu können ist eine konzeptionelle Grundvorstellung ihrer Zusammenhänge notwendig. Modellbildung ist der Prozess, in welchem diese Zusammenhänge erarbeitet, und deren Vorstellung formell beschrieben wird.

Ein Modell ist *"eine Imitation der Realität, und das mathematische Modell eine spezielle Form der Darstellung"* (KMH01, S. 4). Ein Modell kann also, muss aber nicht zwingend mathematisch formuliert sein. Um ein Modell für einen Regelungsprozess bilden zu können muss das System identifiziert werden, also die Fragen zu beantworten

- Welche Systemkomponenten vorhanden sind
- In welcher Kombination sie auftreten (z.B. als Summe, Verkettung ...), und
- Wieviel Einfluß sie auf die Strecke nehmen

Werden hierzu ausschließlich experimentelle Ergebnisse interpretiert, um das System modellhaft zu beschreiben, spricht man von *Blackbox-Modellierung*. Theoretische Überlegungen zu den dem System zugrundeliegenden, meist physikalischen, Eigenschaften werden hierbei also außer Acht gelassen.

Der Vorgang ein Modell ohne experimentelle Unterstützung, also nur durch theoretische Überlegungen anzufertigen, wird *Whitebox-Modellierung* genannt, da man das genaue Systemverhalten noch nicht empirisch untersucht hat, oder dies nicht möglich ist.

Tatsächlich ist praktisch weder eine reine White- noch Blackbox-Modellierung möglich: Eine analytische Beschreibung muss mindestens empirisch validiert werden, und die Interpretation experimentell ermittelter Daten setzt analytisches Vorwissen voraus (KMH01, S. 493).

In der Praxis werden also Mischformen dieser Ansätze verwendet, die unter dem Begriff *Greybox-Modellierung* zusammengefasst werden. Ist das System in einer geeigneten Testumgebung zugänglich, so werden normalerweise erst sehr allgemeine Experimentreihen ange stellt, um auf grundlegende Eigenschaften des Systems schließen zu können.

Im Anschluß versucht, das experimentell ermittelte Systemverhalten bestmöglich theoretisch zu formulieren. Dies kann nur gelingen, wenn die Ergebnisse der bisherigen Experimente im Ansatz richtig interpretiert wurden. Voraussetzung hierfür ist wiederum ein Grundverständnis der Vorgänge innerhalb des Systems, über die im Detail aber nur spekuliert werden kann. Erfahrung in der Modellierung ist also sehr hilfreich.

Um den Grad der Vollständigkeit des analytischen Modells ermitteln zu können, wird es mit denselben Werten angeregt, wie sie in den Experimenten verwendet wurden. Abweichungen vom erwarteten Verhalten werden erneut interpretiert und fließen, eventuell durch neue Experimentdaten ergänzt, in die Verbesserung des Modells ein. Dieser Prozess wird so lange wiederholt, bis reales und simuliertes Systemverhaltens ausreichend übereinstimmen, und das Modell als Grundlage für die weitere Umsetzung dienen kann.

Letztendlich lässt sich ein Modell nicht endgültig beweisen: Ein mathematisches Modell realer Vorgänge gilt als verifiziert, wenn man es empirisch nicht falsifizieren kann. Zum einen wird aber jedes Experiment eine Abweichung vom Modellverhalten zeigen, zum anderen können nie alle theoretisch möglichen Simulationen an einem Modell durchgeführt werden. Wie das Konzept des Regelkreises ist auch die Modellierung nicht auf die Ingenieursdisziplinen beschränkt: Ein Kardiologe stellt beispielsweise mit einer EKG-Messung ein Experiment an, das nach einer theoretischen Vorstellung interpretiert wird und zu einer Diagnose führt.

Es ist unmöglich, in einem Modell jede tatsächlich vorhandene Komponente eines Systems

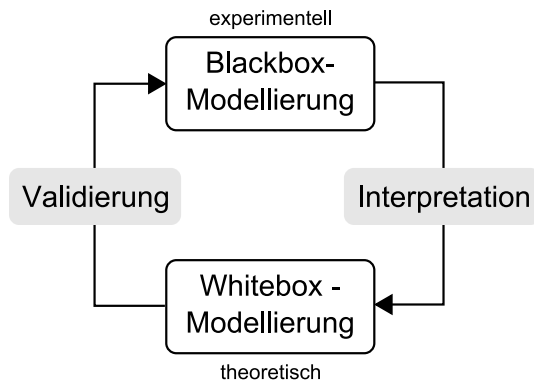


Abbildung 2.14: Allgemeiner Ablauf der Greybox=Modellierung

zu berücksichtigen. Diese sind gerade bei physikalischen Systemen auch nur eingeschränkt bekannt.

Daher wird das Augenmerk auf vermutlich besonders ausschlaggebende Komponenten gelegt, und Systemglieder mit weniger Einfluss werden ignoriert, solange das Modell hinreichend genau ist. Stellt man umgekehrt fest, dass ein Modell von ausreichender Genauigkeit auch mit weniger Komponenten beschrieben werden kann, so werden die überflüssigen Glieder im Zuge einer *Modellreduktion* entfernt.

3 Stand der Technik

Um den Kontext der Problemstellung verstehen zu können werden in diesem Kapitel zunächst die technischen Gegebenheiten vorgestellt, unter denen sie bearbeitet wurde.

3.1 Konkrete Problemstellung am Lehrstuhl Radig

Die mobile Basis, die im weiteren Verlauf behandelt wird, ist ein Prototyp der Firma KUKA Roboter GmbH. In ihrer gegenwärtigen Konfiguration hat sie Abmessungen von ca. 93.4cm Länge, 60cm Breite und 180cm Höhe und wiegt etwa 290kg.

An der Basis sind vier Mecanum-Räder montiert, mit denen omnidirektionale Bewegungen der Basis möglich sind. Die Räder werden über vier Motoren des Typs DSD 36 S der Firma Baumüller Nürnberg GmbH angetrieben. Die Steuerung der Motoren übernehmen vier Baumüller b maXX BM3411 Motor-Controller, die mit konventionellen Cat-5-Ethernet-Kabeln in einer Ringtopologie verbunden sind.

Zusätzlich ist in die Basis ein PC integriert, auf dem Software zur Steuerung und Regelung der Basiskomponenten ausgeführt wird. Dieser ist mit den Motor-Controllern ebenfalls über ein Ethernet verbunden.

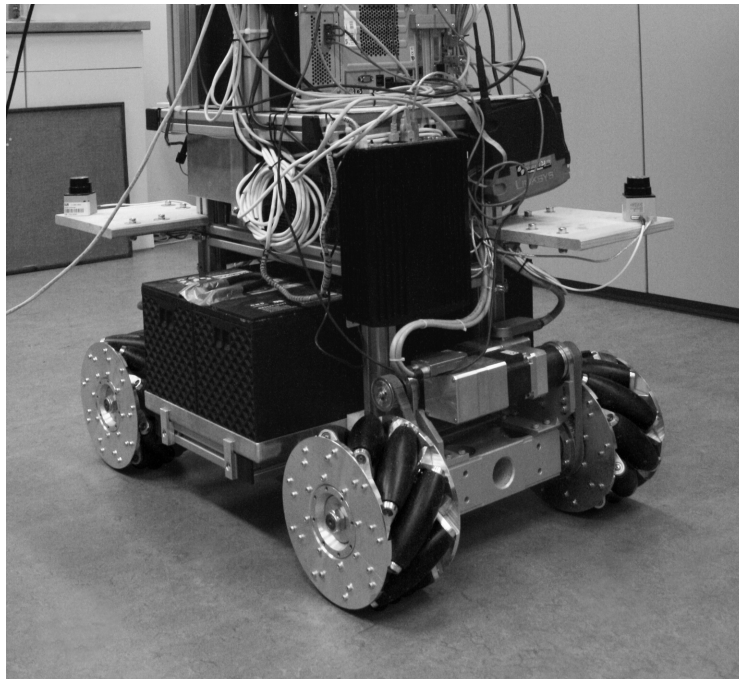


Abbildung 3.1: Mobile Basis am Lehrstuhl für Bildverstehen und wissensbasierte Systeme der TU München

3.1.1 Mecanum Wheels

Das Prinzip des *Mecanum Wheel* wurde 1973 Bengt E. Ilon bei der schwedischen Firma Mecanum AB entwickelt. Man bezeichnet es nach seinem Ursprung auch als *Swedish Wheel* oder *Ilon Wheel*.

Derartige Räder bestehen aus einem konventionellem Rad, auf dem freibewegliche Rollen in einem Winkel von 45° montiert sind (siehe Abb. 3.3). Wird eine Basis mit üblicherweise vier solcher Räder ausgestattet, die unabhängig voneinander angesteuert werden können (Abb. 3.2), so ist durch entsprechende Kombination der Radgeschwindigkeiten eine omnidirektionale Bewegung der Basis möglich.

Omnidirektional bedeutet in diesem Zusammenhang, dass neben konventionellen Vorwärts-, Rückwärts- und Rotationsbewegungen auch eine seitliche Bewegung möglich ist, und diese frei kombiniert werden können. In Abbildung 3.4 sind die Freiheitsgrade mit ihrer entsprechenden Ansteuerung zusammengefasst.

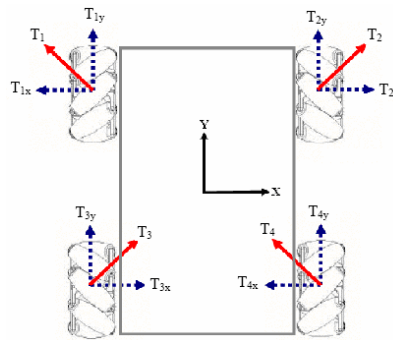


Abbildung 3.2: Schema der Basis mit Koordinatensystemen der Räder (Quelle: (Sal06))

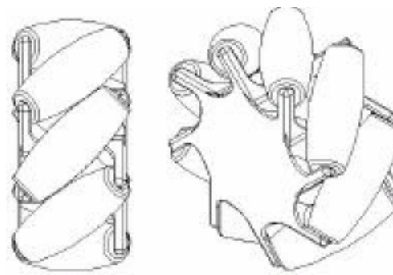


Abbildung 3.3: Schema eines Mecanum Wheels mit 8 Rollen (Quelle: (Sal06))

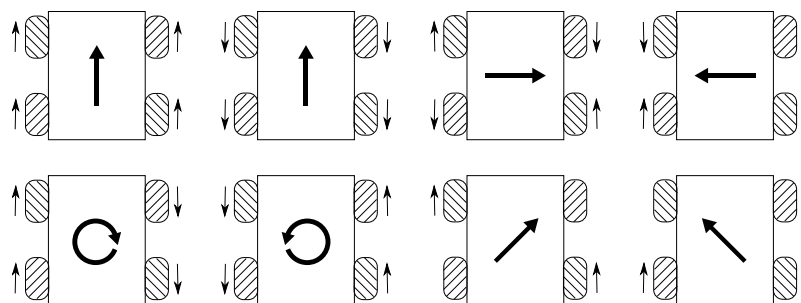


Abbildung 3.4: Freiheitsgrade der Basis mit ihrer Steuerung (nach (Sal06))

3.1.2 Kommunikation mit der Basis

Die Kommunikation mit der Basis beschränkt sich auf die Ansteuerung der Motor-Controller EtherCAT.

EtherCAT ist ein speziell für Echtzeit-Anforderungen entwickelter Feldbus, der auf Ethernet (IEEE 802.3) basiert. Die Teilnahme an einem EtherCAT-Netzwerk benötigt somit neben einem Standard-NIC keine spezielle Hardware. Über einen eigenen Ethertype können

EtherCAT-Frames von anderen Ethernet-Datenpaketen unterschieden werden, sodass es gemeinsam mit anderen Ethernet-Protokollen anderer Ethernet-basierter Anwendungen betrieben werden kann.

Von diesen unterscheidet es sich besonders im Umgang mit den Ethernet-Frames, die von jedem Empfänger nach bevor ihrer Verarbeitung an den nächsten Busteilnehmer weitergeleitet werden.

Kinematik

Die Transformation der Radpositionen (F_1, F_2, F_3, F_4) in die Darstellung im kartesischen Raum als Vektoren (x, y, α) geschieht durch die Berechnung einer Jacobi-Matrix, die für eine Projektion $\mathbb{R}^4 \mapsto \mathbb{R}^3$ verwendet wird.

Eine Jacobi-Matrix ist eine Ableitungsmatrix. Ihre Einträge sind hierbei die partiellen Ableitungen der Positionen im Originalraum nach Positionen im Bildraum¹:

$$J_F = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

Durch die fehlende Dimension im Bildraum hat sie die Form 4×3 , sodass für die Rückprojektion nur eine Pseudo-Inverse Jacobimatrix gebildet werden kann.

In der Vorwärtskinematik wird ausgehend von einem gegebenen Zielvektor (x, y, α) im kartesischen Raum der entsprechende Kraftvektor (F_1, F_2, F_3, F_4) berechnet, während in der inversen Kinematik die resultierende Lage der Basis zu gegebenen Motor Kräften ermittelt wird. Für die Geschwindigkeiten (x, y, a) und Radgeschwindigkeiten $(w_1 \dots w_4)$ gilt:

$$J \cdot \begin{pmatrix} x \\ y \\ a \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix}$$

Eine Steuerung unter Verwendung einer auf Jacobi-Matrizen basierenden Vorwärts-Kinematik war vor der Umsetzung dieser Arbeit bereits vorhanden. Allerdings wurde vermutet, dass diese für die Implementierung einer Admittanzregelung nicht geeignet ist. Diese Zusammenhänge in der Steuerung können für eine spätere Regelung aber relevant werden.

Software

Die Firmware der Basis bietet von sich aus einen eingebetteten PID-Controller mit einer Regelkreisfrequenz von 8 kHz (Bau07, S. 135), dieser wird in dieser Arbeit aber nicht verwendet. Stattdessen wird eine Echtzeit-Umgebung auf einem externen Host aufgesetzt, welche die Basis über Ethercat steuert. Zum einen wird dadurch eine höhere Flexibilität während der Umsetzung erreicht, da man nicht auf einen PID-Regler beschränkt ist, zum anderen kann erst nach Abschluss der Arbeit festgestellt werden, ob und inwiefern sich ein PID-Regler für das gegebene Problem eignet. Die anfallenden Experimente müssen zudem in jedem Fall über einen externen Host gesteuert werden, die Installation einer Echtzeit-Umgebung stellt also keinen unnötigen Mehraufwand dar.

3.2 Interaktive Regelung

Die bisher vorgestellten Grundprinzipien der Regelung zielten auf das Einhalten einer vorgegebenen Regelgröße unter kleinstmöglicher Abweichung ab. Dies wurde am Beispiel der Bewegungs- und Lageregelung ausgeführt.

¹So u.a. in (Shi05)

Neue Aspekte kommen hinzu, wenn die Regelung eine physikalische Interaktion mit der Umgebung berücksichtigen soll, die über den Ausgleich von Störgrößen und der Streckendynamik hinaus geht.

Dies ist beispielsweise im Fall der Kraftregelung gefordert, in der nicht die Position oder Geschwindigkeit des Systems im Vordergrund steht, sondern die Ausübung einer Kraft.

3.2.1 Impedanz- und Admittanzregelung

Impedanz- und Admittanzregelung sind Formen der Interaktionsregelung, also einer Regelung, die durch Kräfteinflüsse aus der Umwelt beeinflusst wird.

Normalerweise ist das Ziel der Impedanzregelung, ein Verhalten der Endeffektoren in der Form zu erreichen, dass sie Kontaktkräften ausweichen oder nachgeben. Die Impedanzregelung regelt hierbei den Grad des Widerstands gegen Kontaktkräfte, die Admittanzregelung analog den Grad der Nachgiebigkeit des Endeffektors. Um Kontaktkräfte zu messen werden üblicherweise Kraftsensoren an den Endeffektoren angebracht, also beispielsweise am Handgelenk oder den Fingerspitzen eines Roboterarms.

Ein Verhalten dieser Art kann aus unterschiedlichen Gründen erwünscht sein. Beispielsweise gestalten sich Teach-In-Vorgänge für einen Roboterarm wesentlich komfortabler, wenn der Arm vom Menschen direkt geführt werden kann, statt durch einen Joystick. Für die gegebene Aufgabe ist sie erwünscht, da die Basis auf Schieben oder Ziehen des Benutzers durch eine dem Menschen nachvollziehbare Positionsveränderung reagieren soll.

Eine Interaktionsregelung kann passiv oder aktiv umgesetzt sein. In der passiven Variante werden die betroffenen Endeffektoren nicht über einen Regelkreis gesteuert, sondern sind allein aufgrund ihres Aufbaus - beispielsweise durch elastische Arme - in der Lage, Kontaktkräften ihrer Umgebung nachzugeben (shr08, 162).

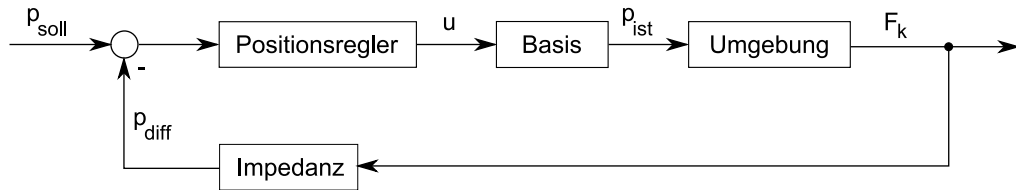


Abbildung 3.5: Blockdiagramm Impedanzregelung (nach (Win06, S. 21))

Das Wunschverhalten der Nachgiebigkeit des Systems wird als Masse-Dämpfer-Feder-System beschrieben, also nach der Differentialgleichung

$$f = M\Delta\ddot{x} + D_k\Delta\dot{x} + K_k\Delta x \quad (3.1)$$

Mit M , D_k , K_k als virtuelle Masse, Dämpfung, Steifheit.

Aus dieser Gleichung und Abbildung 3.5 geht hervor, dass der Admittanzregler die Kontaktkraft F_K unter Beeinflussung des Stellwerts für einen untergeordneten Positionsregler regelt. Der Stellalgorithmus für die auf die Sollposition aufgeschaltene Differenz geht aus der Lösung der Differentialgleichung 3.1 hervor.

4 Experimentelle Systemidentifikation

Für eine regelungstechnische Aufgabe ist zunächst von Interesse, wie das gegebene System mit Hilfe linearer Systembausteine beschrieben werden kann. Es hierfür auch untersucht, ob vorhandene Nichtlinearitäten ignoriert oder ausreichend linearisiert werden können. Ist dies nicht der Fall, so erhöht sich die Komplexität des Reglers erheblich.

Bei der experimentellen Systemidentifikation mechanischer Prozesse ist es der übliche Ansatz, das beobachtete System anzuregen und mit der Reaktion oder Ausgabe des Systems in Zusammenhang zu setzen.

Ein hypothetisches System, das keinen Einflüssen gleich welcher Art ausgesetzt ist, würde das Eingangssignal unverändert zurückgeben. Unterschiede zwischen Eingangs- und Ausgangsdaten können daher ausschließlich auf Einflüsse auf das System zurückgeführt werden. Allerdings liefert das Ergebnis nur die Summe aller Einflüsse, es kann also keinen Aufschluß über deren genaue Zusammensetzung oder Wechselwirkungen geben.

Die Vorstellung von einem System als Übertragungsfunktion von Eingangssignalen $u(t)$ und Ausgangssignalen $y(t)$ unter Berücksichtigung von systemfremden Störungen in der Ausgabe $v(t)$ wird formuliert als (Cod05, S. 8):

$$\begin{aligned} y(t) &= G_0(s)u(t) + v(t) & \text{mit } sx(t) &= \dot{x}(t) & \text{im kontinuierlichen, oder} \\ y(t) &= G_0(z)u(t) + v(t) & \text{mit } z^{-1}x(t) &= x(t-1) & \text{im diskreten Fall} \end{aligned} \quad (4.1)$$

Es kann auch als Operator definiert werden (KMH01, 20):

$$u : \mathcal{T} \rightarrow \mathcal{U} \subseteq \mathcal{R}^r, \quad y : \mathcal{T} \rightarrow \mathcal{Y} \subseteq \mathcal{R}^v, \quad \mathbf{S} : \mathcal{U} \rightarrow \mathcal{Y}, \quad y = \mathbf{S}[u] \quad (4.2)$$

Die Indizes r und v geben die Dimension der Eingangs- und Ausgangsvektoren des Systems an, diese Signale können also Vektoren sein. Im Regelfall klassifiziert man Systeme als *SISO* (Single Input, Single Output) oder *MIMO* (Multiple Input, Multiple Output). Abbildung 4.1 zeigt die konzeptionelle Vorstellung eines Systems mit den beschriebenen Größen. Es wird auch deutlich, dass die Systemantwort nur in Kombination mit systemfremden Störungen gemessen werden kann. Alle bekannten Störgrößen, also auch die in der Messschleife, müssen vor dem Versuchsaufbau identifiziert, bewertet und gegebenenfalls eliminiert werden, um sich auf die Aussagekraft der experimentellen Ergebnisse verlassen zu können.

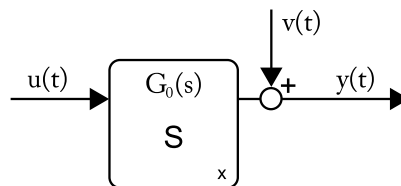


Abbildung 4.1: Allgemeine konzeptionelle Beschreibung eines Systems

4.1 Charakterisierung der Störgrößen

Bekanntlich kann nur so gut geregelt werden, wie gemessen und modelliert wird. Die Identifikation von Störgrößen hilft daher, die Güte der Regelung bereits zu einem frühen Zeitpunkt zu verbessern. Andererseits liefert sie auch Aufschluss über die maximal erreichbare Regelgüte und den Grad des nötigen Aufwandes für die Modellierung.

Auf der Hand liegen Störeinflüsse durch Messrauschen. Dieses kann am einfachsten durch die Differenz eines synthetischen Eingangssignals und der direkt gemessenen Werte ermittelt werden. Für den in diesem Fall verwendeten Encoder ist es allerdings nicht möglich, ein synthetisches Signal anzulegen um daraus das Rauschverhalten zu ermitteln, da die anschließend gemessene Lageveränderung auch anderen Einflüssen, wie dem Getriebe, unterworfen ist.

Damit bleibt als einzige Möglichkeit, die Werte des Sensors in Ruhelage als Abweichungen vom realen Nullwert und somit als Grundrauschen zu interpretieren. Daraus lassen sich aber keine Rückschlüsse auf die Genauigkeit des Sensors im Bezug auf die Dynamik des Systems ziehen: Messabweichungen und somit Drift sind beispielsweise weiter unbekannt.

Glücklicherweise wurden vom Hersteller des Sensors hinreichend genaue Angaben zum Rauschverhalten geliefert.

Erwartungsgemäß muss demnach neben dem gemessenen Grundrauschen von 15 Encoder-Ticks über die Auflösung von 3994930 Ticks pro Meter Positionsveränderung in diesem Fall kein weiteres Störverhalten berücksichtigt werden.

4.2 Versuchsaufbau

Wie in Kapitel 2 erwähnt können über den Speed Controller der Basis Geschwindigkeitswerte der einzelnen Räder angefordert werden. Auf diese Möglichkeit muss für die folgenden Experimente aber verzichtet werden, da der Speed Controller durch die Regelung der Motorgeschwindigkeiten Einfluss auf das System nehmen würde. Er wird daher deaktiviert indem die Basiscontroller in den reinen Torque Control Modus gestellt werden.

Statt durch den Speed Controller werden Daten aus dem Encoder-Modul interpretiert. In einer sehr hohen Auflösung von 16 Bit pro Umdrehung und einem Umdrehungszähler 16 Bit messen die Encoder-Sensoren die Positionsveränderung am Anker jedes Motors. Im Torque-Control-Modus kann die Rotationsgeschwindigkeit eines Motors nicht direkt von einem Sensor gelesen werden, die erste Ableitung über die Positionswerte ergibt aber sehr exakte Werte für die Geschwindigkeit der Basis. Die Güte der Geschwindigkeitsdaten hängt aber auch von der verwendeten Methode der Ableitung ab.

Abbildung 4.2 zeigt schematisch den Aufbau der Messschleife. Der rot markierte Signalfluss bildet keinen geschlossenen Kreis, da die gemessene Systemantwort keinen Einfluß auf die Steuerung hat. Das bewusste Entfernen der Rückkopplung bezeichnet man als *Öffnen des Regelkreises*.

Bei der gegebenen Regelungsaufgabe handelt es sich um ein MIMO-System, gemäß Gleichung 4.2 mit $r = v = 4$, da jeder Motor einzeln angesteuert wird, und zu jedem Motor Encoder-Daten gelesen werden.

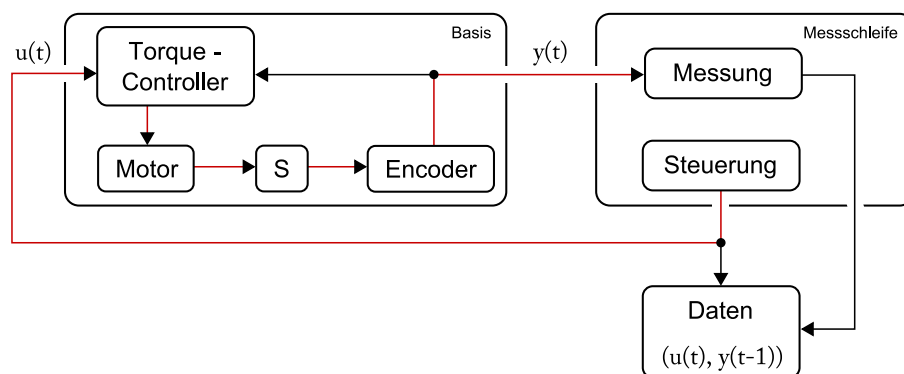


Abbildung 4.2: Schema der Messschleife mit offenem Regelkreis

4.3 Bestimmung eines linearen, zeitinvarianten Systems (LTI)

Wie bereits in Kapitel 4 erläutert lassen sich lineare Systeme wesentlich einfacher modellieren als nichtlineare Systeme. Es ist vor der Implementierung des Reglers also von Interesse, ob das physikalische Verhalten der Basis linear modelliert werden kann.

Gemeinhin bezeichnet man ein System als linear, wenn für die relevanten Wirkzusammenhänge gilt:

$$\exists t_c \quad \forall t_s > t_c: \quad x_a(t_s) = K \cdot x_e(t_s)$$

Also - unter Vorbehalt systembedingter Dynamik wie Verzögerungen und Einschwingphasen - ab einem Zeitpunkt t_c ein proportionaler Zusammenhang zwischen Eingangs- und Ausgangsgrößen besteht. Für die Bestimmung eines LTI wird aber zudem gefordert, dass für alle Zustandsübergänge des Systems das Superpositionsprinzip angewendet werden kann. Dies schließt demnach auch das Systemverhalten vor seiner Stabilisierung (t_s) ein.

Die Vermutung liegt nahe, dass sich Kraft- und Positionskontrolle als lineares System modellieren lassen, da für Newton'sche Kräfte das Superpositionsprinzip gilt. Allerdings kann hiervon nicht im Allgemeinen ausgegangen werden, da letztendlich die konkrete Physik des Systems entscheidet. Hydraulische Systeme können zum Beispiel selten linear formuliert werden.

Eine Reihe von Experimenten soll nun ersten Aufschluß über die Linearität des Systems geben.

Die Ergebnisse dieser Versuchsreihen sind später bei der Validierung des theoretischen Modells hilfreich: Eine korrekte theoretische Abbildung des Systems würde dieselben Ausgangssignale erzeugen wie das reale System im Versuch.

Anders als beispielsweise bei der Analyse von Filtern üblich wird hierzu nicht die Impulsantwort interpretiert, die sich aus einem angenäherten Dirac-Stoß ergibt. Dieser würde zu keiner messbaren Reaktion der Basis führen. Die verwendeten Signale, die üblicherweise zur Blackbox-Modellierung herangezogen werden, sind:

Sprung zur Ermittlung des Störverhaltens

Rampe zur Ermittlung des Führungsverhaltens

Sinus-Sweep zur Ermittlung des Frequenzverhaltens

In der ersten Versuchsreihe wird die Sprungfunktion als konstanter Drehmoment auf die Motoren der ruhenden Basis gegeben. Die daraus resultierende Geschwindigkeit der Basis wird als Systemantwort aufgefasst.

Bedenkt man die hohe Masse der mobilen Basis, so ist davon auszugehen, dass besonders Reibung und Trägheit großen Einfluss auf das Bewegungsverhalten des Systems nehmen.

Abbildung 4.4 zeigt das Ergebnis des Experiments. Der rote Graph zeigt das Eingangssignal (Drehmoment auf einen Motor), der grüne Graph die Reaktion des Systems als Sprungantwort (Ableitung der Position über die Zeit).

Interessant ist die Reaktion der Basis in Bewegung: Die Geschwindigkeit steigt nicht linear, sondern in einer asymptotischen Annäherung bis zum Ende des Impulses.

Wichtig ist, dass die Sprungantwort unverzögert erfolgt. Es konnte in diesem Experiment also keine messbare Totzeit nachgewiesen werden. Insgesamt wirkt die Sprungantwort typisch für ein Verzögerungssystem.

Die Sprungantwort hat die augenscheinliche Charakteristik eines PT_n -Systems, da sie einen sigmoidalen Verlauf beschreibt. Für die Verzögerungszeiten ergibt sich $T_u = 52.3$ und $T_g = 652.4$, und somit einem Quotienten von 12.47. Das Verhalten als PT_2 -System angenähert werden, da $T_g/T_u = 12.47 > 10$ ist (Unb08, S. 197). Eine zu berücksichtigende Totzeit liegt

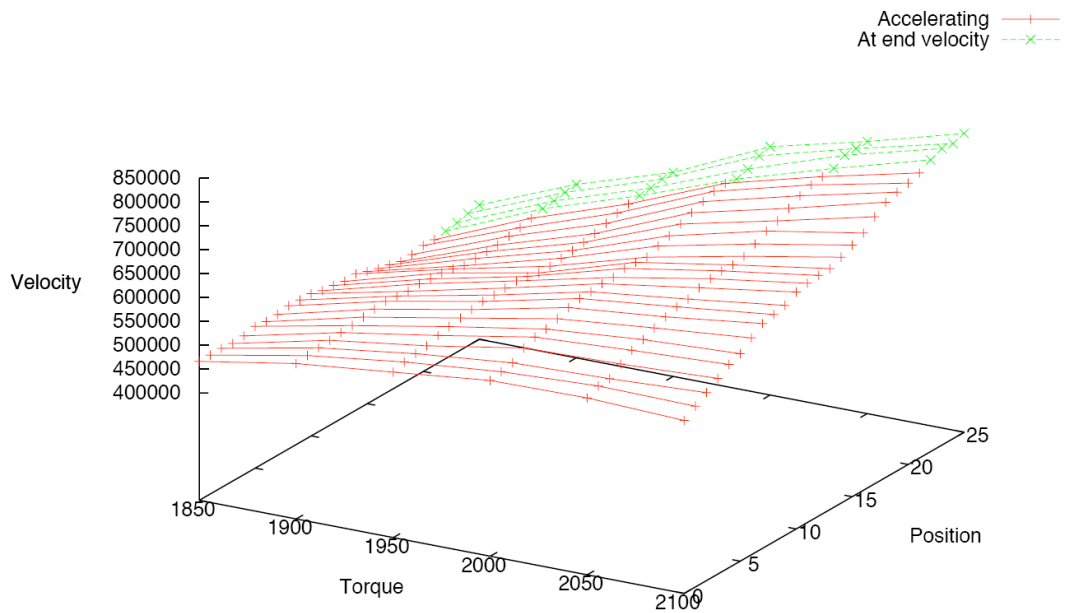


Abbildung 4.3: Ergebnis der Experimentreihe zum linearen Verhalten mit Verlauf in der Beschleunigung

zumindest nach diesem Experiment nicht vor.

Ob ein PT_2 -Glied das System tatsächlich brauchbar beschreibt, lässt sich zu diesem Zeitpunkt nicht sagen.

Nach dem Prinzip eines Sweeps wird die Frequenz der Kraftschwingungen mit jeder Testreihe erhöht. Erwartungsgemäß nimmt die Amplitude der Systemantwort mit der Erhöhung der Frequenz ab, bis das System mit keiner Schwingung antwortet, die Basis sich also nicht mehr bewegt.

Dies zeigt erneut den Tiefpasscharakter des Systems. Interessant ist hierbei die Grenzfrequenz, bei der die Basis nicht mehr mit Bewegung reagiert. Abbildung 4.5 zeigt den gesamten Verlauf des Experiments bis zum Stillstand der Basis.

Intuitiv ist ersichtlich, dass eine Bewegung erst ab einem bestimmten Drehmoment beobachtet werden kann. Zu diesem Zeitpunkt kann noch keine definitive Aussage darüber getroffen werden, welche Aspekte, und in welcher Gewichtung, hierfür verantwortlich sind. Dieses Phänomen entspricht allerdings den Erwartungen hinsichtlich Reibung und Trägheit.

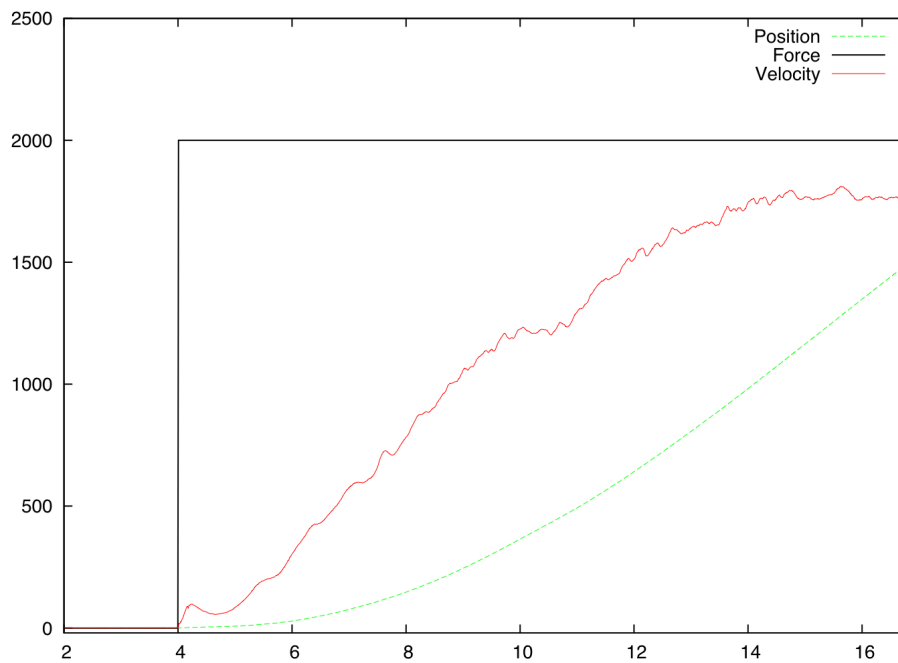


Abbildung 4.4: Sprungfunktion auf Torque (schwarz) und Sprungantwort (rot)

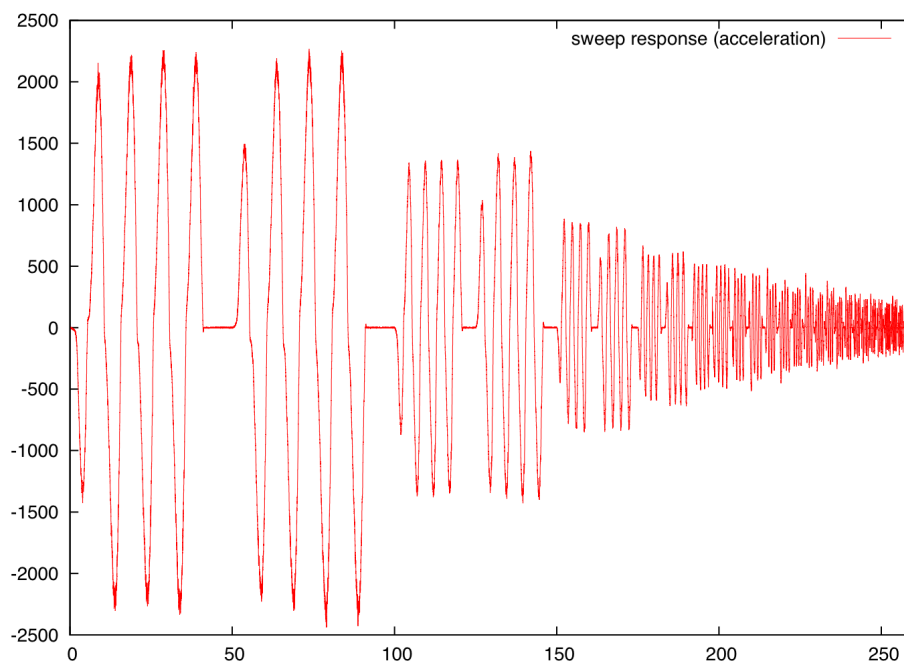


Abbildung 4.5: Systemantwort auf Torque-Sweep

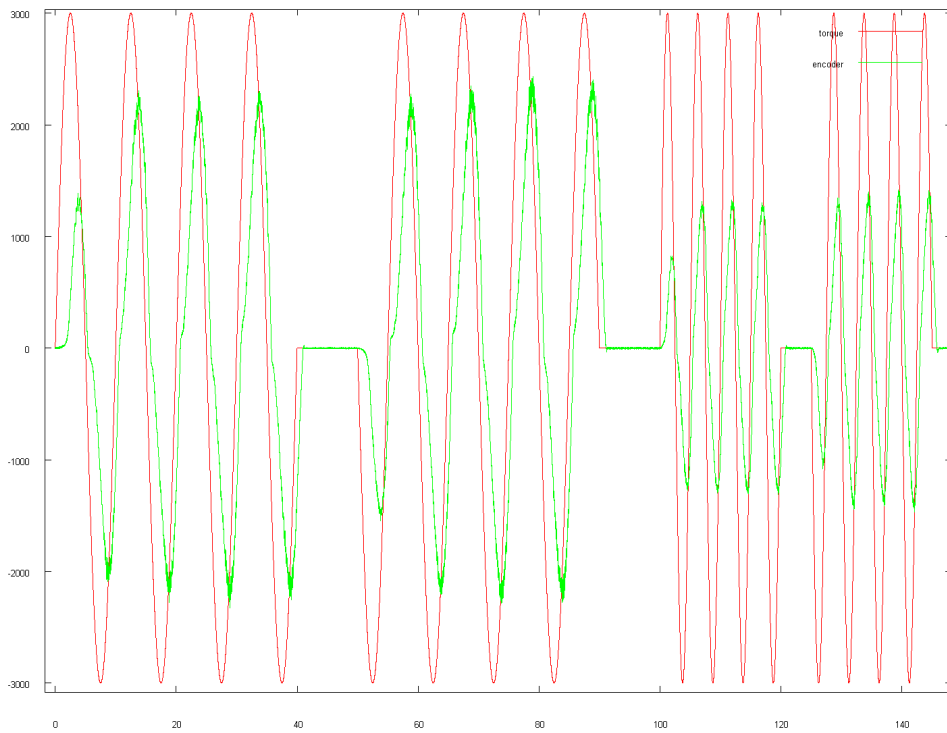


Abbildung 4.6: Sinus-Sweeps (rot) mit Frequenzgang (grün)

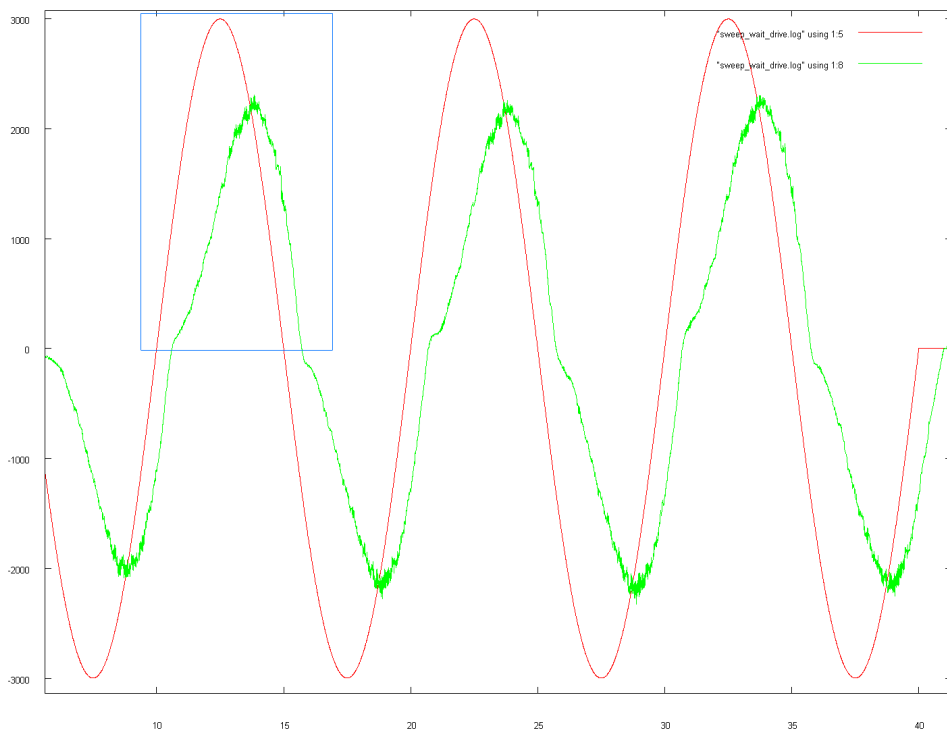


Abbildung 4.7: Einzelne Sinusfolge (rot) mit Frequenzgang (grün)

5 Analytische Systemidentifikation

Wie bereits beschrieben gehen in der Modellbildung analytische und experimentelle Prozesse Hand in Hand und bedingen sich im Greybox-Modelling gegenseitig.

In diesem Kapitel wird exemplarisch ein Ansatz ausgeführt, durch Interpretation der Ergebnisse aus den Versuchsreihen ein mathematisches Modell des Systems zu bilden. Hierzu werden die identifizierten Systemkomponenten einzeln betrachtet und jeweils eine Methode gezeigt, diese analytisch zu beschreiben.

5.1 Modellierung physikalischer Systeme

Die Frage, welche mathematische Ausdrucksform sich für die Beschreibung eines Systemverhaltens am besten eignet, kann nicht allgemein beantwortet werden. Für manche Probleme eignen sich Differentialgleichungen sehr gut, andere können - wie beispielsweise Totzeiten - nur im Frequenzbereich ausgedrückt werden.

Die Modellbildung eines physikalischen Systems basiert daher auf

- der Zerlegung des Systems in einzelne Subsysteme (Analyse)
- der anschließenden Modellierung dieser Glieder als abgeschlossene Teilprobleme unter Verwendung einer jeweils geeigneten Darstellungsform
- der abschließenden Kombination der Submodelle in ein Gesamtmodell (Synthese)

Übliche Darstellungsformen für die Submodelle sind, unter anderem

Differentialgleichungen beispielsweise für Wirkzusammenhänge in Abhängigkeit Newton'scher Kräfte

Übertragungsfunktionen zur Formulierung eines Frequenzgangs in Abhängigkeit von Größen im s -Bereich

Algorithmische Darstellung als prozedurale Beschreibung des Verhaltens in Abhängigkeit von z.B. der Zeit, für nicht-lineare Komponenten oft auch durch Lookup-Tabellen

Oft sind Darstellungsvormen äquivalent, und die einzelnen Methoden schließen sich gegenseitig nicht aus: Differentialgleichungen können beispielsweise in den Laplace-Bereich transformiert, dort gelöst, und die Lösung abschließend in den Zeitbereich überführt werden. Dies ist sogar ein gängiges Verfahren¹.

Es ist theoretisch auch möglich, die Übergangsfunktion eines Systems im Zeitbereich durch polynomielle Approximation zu ermitteln. Es ist aber leicht nachvollziehbar, dass ein derartiges Modell nur im approximierten Bereich gültig ist, und außerhalb von diesem drastisch vom realen Systemverhalten abweicht. Dies ist für polynomiell angenäherte Modelle typisch (KMH01, S. 512). In der Praxis werden komplexe und insbesondere nicht-lineare Systeme in Software wie SIMULINK modelliert, die auch die Simulation des Systems durchführen kann.

5.2 Modellierung des Systems

5.2.1 Modellierung der Trägheit

Für die Modellierung wird die Masse der Basis als wichtigstes Kriterium betrachtet. Es liegt nahe, dass folglich auch ihre Trägheit großen Einfluss auf ihre Dynamik nehmen muss.

¹Siehe u.a. (Unb08, S. 60)

Nach dem zweiten Newton'schen Gesetz ist die Trägheit eines Körpers in Abhängigkeit von Beschleunigung und Masse definiert als:

$$F = \dot{v} \cdot m + \dot{m} \cdot v = \dot{v} \cdot m \quad \text{Für } \dot{m} = 0$$

Das durch Trägheit bedingte Übertragungsverhalten wird als Tiefpass erster Ordnung aufgefasst, also als PT_1 -System. Ein durch Trägheit bedingtes Totzeitverhalten wird demnach nicht vermutet. Die in den Sinus-Sweep-Experimenten beobachtete Phasenverschiebung wird zum Großteil diesem Subsystem zugeschrieben.

In der Analyse werden die Massen der Motoren vorerst nicht betrachtet, da diese bereits durch die verwendeten Drehmomentenregler in den Motor-Controllern ausgeglichen werden sollten.

5.2.2 Modellierung des Rollwiderstands

In den bisherigen Experimenten wurden Nichtlinearitäten festgestellt, die sich als Anrollwiderstand bemerkbar machen, sich aber noch nicht eindeutig bestimmen lassen.

Das beobachtete Verhalten im Anrollen und im Umkehrpunkt der Bewegung hat eine große Übereinstimmung mit Systemverhalten, wie es von Reibungskomponenten bekannt ist. Der Rollwiderstand wird demnach wie eine Komposition von Reibungskräften modelliert.

Motivation hinter der Ermittlung eines Modells für den Rollwiderstand ist der Versuch, das System durch ein lineares Verhalten beschreiben zu können: Sollte das Modell die nichtlinearen Faktoren ausreichend umfassend beschreiben, dann das inverse Modell diese - zumindest theoretisch - kompensieren.

Das Erstellen von präzisen Reibungsmodellen ist ein komplexer Vorgang, der detaillierte Kenntnisse aus der Physik erfordert. Das bekannte Gesetz zur Gleitreibung

$$F_R = \mu \cdot F_N$$

verleitet zu der Annahme, ein Reibungskoeffizient μ genüge, um eine konstante Reibungskraft für ein System mit Gesamtmasse M zu ermitteln. Dies ist aber keineswegs der im Allgemeinen der Fall: Spätestens bei Phänomenen wie elastischer und plastischer Deformation kann ein vollständiges Reibungsmodell nur unter hohem Aufwand ermittelt werden.

Glücklicherweise ist es für die gegebene Problemstellung nicht zwingend notwendig, ein mathematisches Modell aller Reibungskomponenten zu ermitteln. Es gibt Annäherungsmethoden, die zwar nur unter bestimmten Bedingungen eine ausreichende Güte liefern, aber richtig eingesetzt trotz großer Vereinfachung ein brauchbares Modell liefern. Der übliche Ansatz ist die Aufstellung eines globalen Reibungsmodells, in die alle Reibungskomponenten einfließen, deren genaue Zusammensetzung aber nicht bekannt sein muss.

Grundsätzlich analysiert man Reibung unter anderem durch folgende Grundmodelle (AS02, p. 23):

- Viskose Reibung
- Gleitreibung
- Coulomb-Reibung
- Stribeck-Reibung (Übergang von Haftreibung zu Gleitreibung)

Da es sich bei Reibung um Newton'sche Kräfte handelt, gilt für sie das Superpositionsprinzip. Gleitreibung und viskose Reibung sind aus regelungstechnischer Sicht wenig problematisch, da beide Kräfte auch in Kombination durch lineare Funktionen beschrieben werden können.

Gleitreibung

Die Gleitreibung ist eine konstante Dämpfung, also beschrieben durch

$$F_{RC} = \mu_C \cdot F_N \tag{5.1}$$

Die Gleitreibung ist demnach nicht abhängig von der Geschwindigkeit, im Gegensatz zur viskosen Reibung, die linear mit der Geschwindigkeit steigt:

$$F_{Rv} = \mu_v \cdot \dot{x} \cdot F_N \quad (5.2)$$

Unangenehme Formen der Reibung sind die Haftreibung und die Coulomb-Reibung, da sie um den Nullpunkt der Geschwindigkeit unstetig, also nichtlinear sind. Das gilt auch für die Stribeck-Reibung, da sie den Übergang von Haft- zu Gleitreibung beschreibt und somit die Unstetigkeit der Haftreibung beinhaltet.

Diese Formen der Reibung können nicht durch kontinuierliche Funktionen über die Geschwindigkeit, sondern nur abschnittsweise definiert werden. Es gibt aber kontinuierliche Annäherungen, die sich des Tangens bedienen, da dieser um den Nullpunkt zwar keine Sprungstelle, aber einen ähnlichen Verlauf beschreibt.

Die Behebung der Unstetigkeit durch eine Annäherung ist auch aus numerischer Sicht erwünscht, da Sprungstellen in der ersten Ableitung zu Instabilität führen.

Außerdem können Reibungskomponenten zusätzliche Abhängigkeiten aufweisen, die sich im Allgemeinen nur schwer brauchbar modellieren lassen. Beispielsweise können Coulomb- und Haftreibung abhängig von Temperatur und Alter der Bauteile sein, und somit zu einem zeitvarianten System führen (Win06, S. 103).

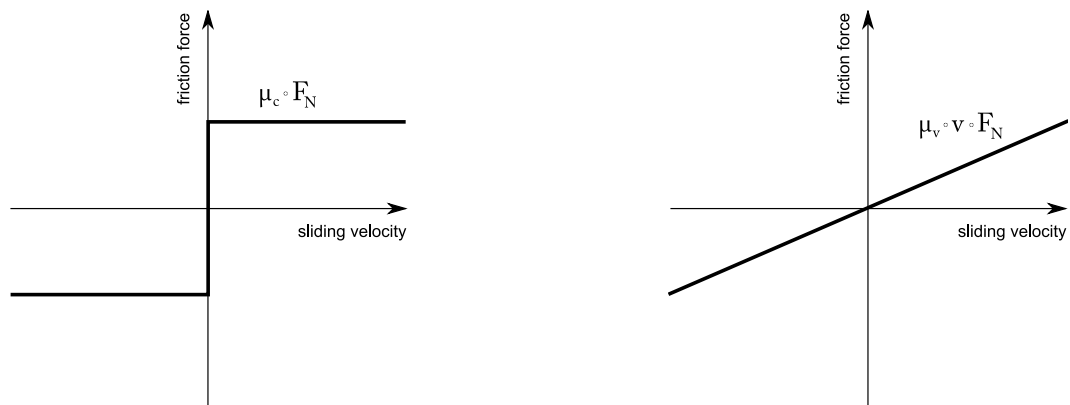


Abbildung 5.1: Coulomb-Reibung und viskose Reibung, schematisch

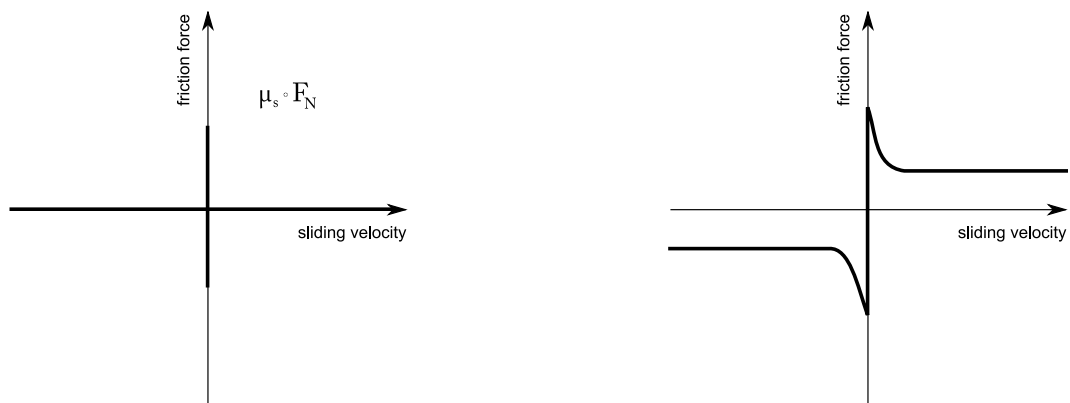


Abbildung 5.2: Haftreibung und Stribeck-Effekt, schematisch

Nach den physikalischen Gegebenheiten der Basis wird eine Kombination von Haft- und Gleitreibung vermutet, die durch einen Stribeck-Effekt ineinander übergehen. Diese Annahme wurde aus dem Verzögerungsverhalten bei der Antwort auf die Sprungfunktion geschlossen, und aus den aufgetretenen Nichtlinearitäten in der Systemantwort auf den Sinus-Sweep. Ein

Stribeck-Effekt wird vermutet, da ein Losbrechen in keinem Fall zu beobachten war, der Übergang von Haft- zu Gleitreibung demnach kontinuierlich verläuft.

Coulomb-Reibung

Um die Reibung so genau wie möglich beschreiben zu können, wird in Experimentreihen der Versuch unternommen, die Coulombsche Reibung isoliert zu messen. Dies ist nur möglich, wenn die Basis sich in Bewegung befindet, die Haftreibung also überwunden ist. Problematisch ist hierbei, dass in Bewegung der Basis Reibung und Trägheit immer kombiniert auftreten. Die Basis darf also außerdem nach Erreichen einer Endgeschwindigkeit nicht weiter beschleunigt werden, da sonst Trägheitskräfte auf die Basis wirken würden, die dann nicht mehr vom Einfluß der Reibung unterschieden werden könnten.

In den folgenden Experimenten wurde ein konstanter Torque mit den Motoren erzeugt, bis sich eine stabile Endgeschwindigkeit eingestellt hat. Diese wurde zu jedem Torque gemessen. Jedes Experiment wurde mehrmals wiederholt, und aus den Ergebnissen jeweils Mittelwerte gebildet. So kann letztendlich zu jedem verwendeten Torque die Endgeschwindigkeit ermittelt werden. Setzt man die Endgeschwindigkeiten in Relation zum Torque, so lässt sich die Linearität des Verhaltens unter Vernachlässigung der Haftreibung untersuchen.

In Abbildung 5.3 ist eine Auswahl der Ergebnisse als Graphen von Geschwindigkeit über die Zeit dargestellt.

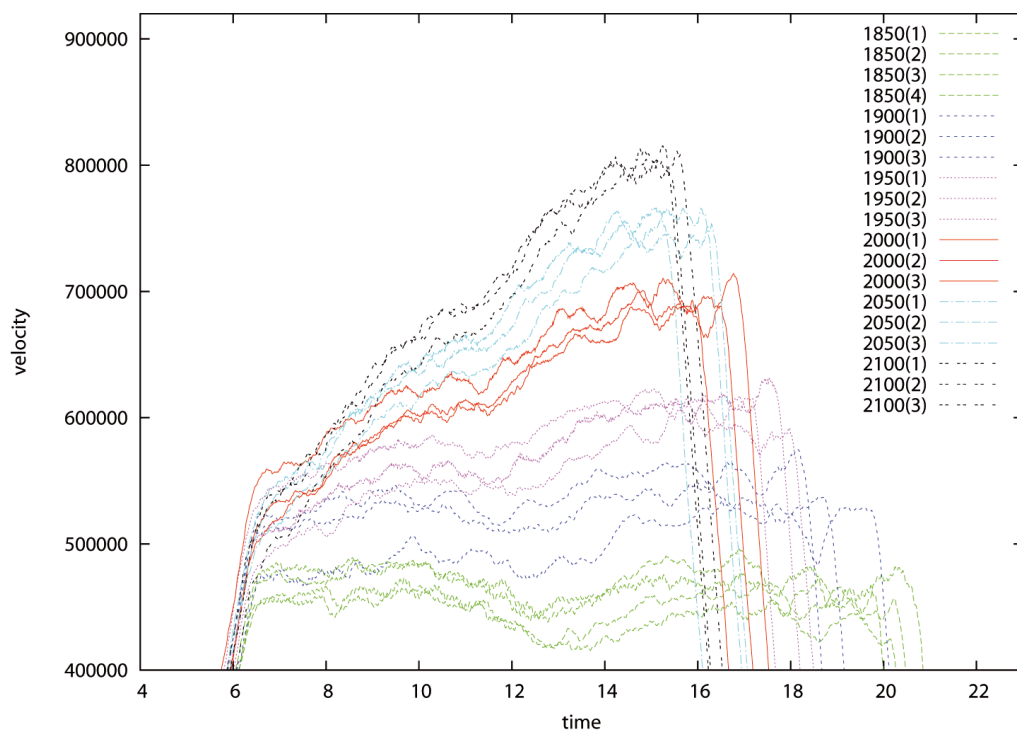


Abbildung 5.3: Experimentelle Ergebnisse zur Reibungsmodellierung ($v:t$)

Schließlich werden die Graphen nach den Positionsdaten, statt nach der Zeit, angeordnet. Im Ergebnis liegen die Abschnitte zu denen sich die Basis in einer quasi-konstanten Geschwindigkeit befand, auf der x-Achse übereinander. Abbildung 5.4 zeigt das Ergebnis, mit jeweils einem Graphen für die Durchschnittswerte einer Experimentreihe.

Bildet man die Differenzen der Endgeschwindigkeiten nach der Ausrichtung nach Positionsdaten, so kann die Linearität des Rollwiderstands in Abhängigkeit von der zugeführten Kraft ermittelt werden. Abbildung 4.3 wurde durch die Darstellung dieser Differenzen über alle Graphen gebildet. Es entsteht eine Fläche, deren Verlauf den Zusammenhang zwischen

konstantem Torque und den Endgeschwindigkeiten beschreibt. Rot markiert ist der Bereich, in dem zu jedem Torque eine Endgeschwindigkeit erreicht wurde. Hier ist der proportionale Zusammenhang zwischen Torque und Geschwindigkeit zu erkennen.

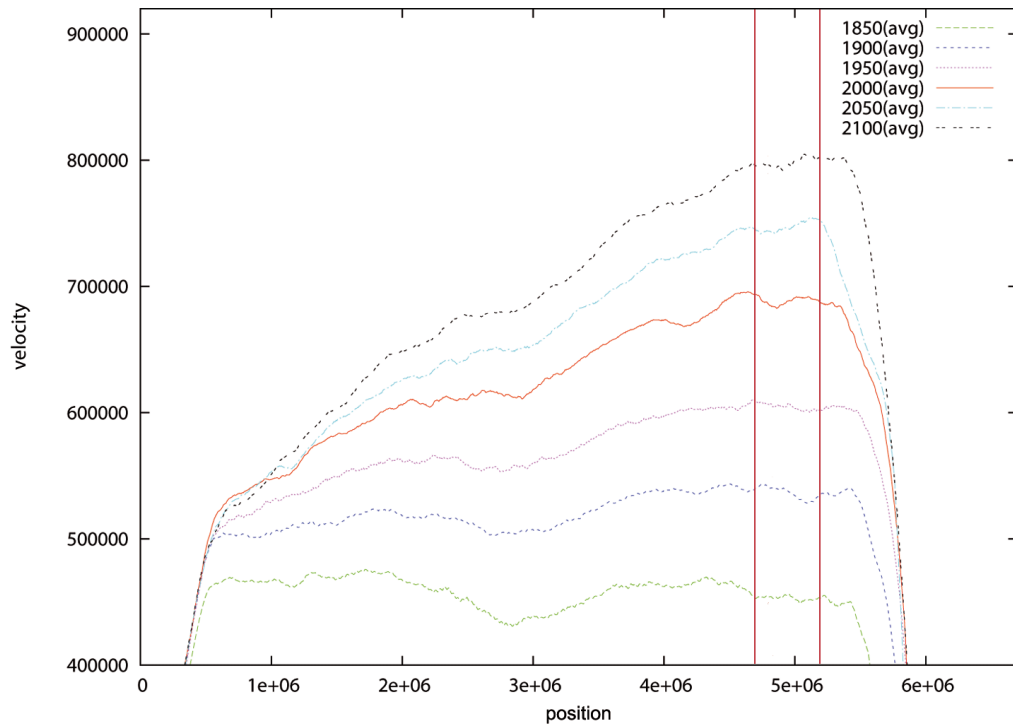


Abbildung 5.4: Experimentelle Ergebnisse zur Reibungsmodellierung im Durchschnitt (v:p)

Die Experimente zur Identifikation der Reibungskomponenten haben somit gezeigt, dass die Gleitreibungskomponente proportional zur der dem System zugeführten Kraft, hier also dem Torque, modelliert werden kann. Bei höheren Beschleunigungen nimmt die jeweilige Endgeschwindigkeit immer weniger, was einer viskosen Reibungskomponente zugeschrieben wird. Diese wird aber vorerst als unbedeutend betrachtet.

Wären bei den Experimenten weitere nichtlineare Zusammenhänge deutlich geworden, dann hätten weitere Faktoren im Reibungsmodell berücksichtigt werden müssen.

Theoretisch könnte man die Rollwiderstände auch über die Beobachtung des Abbremsens der Basis aus einer Bewegung modellieren, da ausschließlich Reibungskräfte für das Entschleunigen eines in Bewegung befindlichen Körpers verantwortlich sind, der keiner äußeren Kraft ausgesetzt ist.

Anrollwiderstand als Haftreibung

Offen ist noch die Untersuchung des Anrollwiderstands. Dieser hat sich in den Experimenten ähnlich einer Haftreibung gezeigt und wird demnach im Modell als solche aufgefasst, auch wenn tatsächlich keine Haftreibung im Sinne ihrer Definition vorhanden ist.

Die Haftreibung ist eine Zwangskraft, die nur wirksam ist, solange das System sich nicht bewegt. Sie ist immer genau so groß wie die von außen zugeführte Kraft, die in Abwesenheit von Reibung das System in Bewegung setzen würde - also die Superposition aller äußeren Kräfte.

Befindet sich das System auch nur in geringer Bewegung, so ist der Anrollwiderstand überwunden und seine Gegenkraft ist vollständig kompensiert. Für die Haftreibung wird der maximale Torque angesetzt, bei dem sich die Basis gerade nicht von der Stelle bewegt.

Die dynamische Reibung kann aufgrund ihrer nachgewiesenen Linearität durch den experimentell festgestellten Reibungskoeffizienten modelliert werden.

Das Modell ist demnach abhängig von der Geschwindigkeit und dem gewünschten effektiven Torque (siehe [1])

$$F_R(v = 0) = F_c + \text{sign}(v) + d \cdot v$$

Da der proportionale Zusammenhang von Geschwindigkeit und Kraft nachgewiesen wurde, kann v durch das Produkt von Torque und einem Reibungskoeffizienten ersetzt werden.

Bis zum letzten Stand der Umsetzung ist die Rollwiderstandskomponente nicht in die Modellierung eingeflossen, da das Gesamtverhalten als PT_2 -System modelliert wurde und somit die Reibung nicht als einzelnes Systemglied verwendet wird.

Sollte dies später notwendig werden finden sich Ansätze zu Reibungsmodellen in ref[55]schuster08.

5.2.3 Modellierung des Getriebes

Nach Herstellerangaben werden die Räder über Planetengetriebe angetrieben. Diese werden üblicherweise als linear betrachtet, und bislang hat kein Experiment an der Basis diese Einschätzung für den konkreten Fall widerlegt. Es wurden aber experimentell Phänomene beobachtet, die sich am einfachsten durch Spiel bzw. Hysterese im Getriebe erklären lassen, die konkrete Ausprägung dieser Effekte war allerdings marginal. Das Getriebe fließt also vorerst nicht in die Modellierung ein.

5.2.4 Klassifizierung des Modells

Nach den bisherigen Ergebnissen kann man das Modell eines Antriebsmotors der Basis unter folgenden Aspekten beschreiben:

Trägheit als Energiespeicher (PT1-System)

Reibungen als eigenes Reibungsmodell im Zeitbereich mit konstanter Dämpfung (insgesamt PT1-System)

Hysterese im Getriebe als wahrscheinlich unkritisches T_t -System im Umkehrpunkt der Bewegung

Die Grundlage des analytischen Modells bilden zwei hintereinander geschaltene PT1-Systeme, wovon das zweite als Trägheit interpretiert wird. Das erste Verzögerungsglied wird der Summe aller Rollwiderstände zugeschrieben. Die Totzeit im Getriebe wird vorerst vernachlässigt.

Wie bereits in der Interpretation der Experimente ausgeführt wird das System als aperiodisch stabiles PT2-System modelliert, das wiederum als Folge zweier PT1-Glieder beschrieben werden kann.

Ein PT1-System wird im Frequenzbereich wie erwähnt beschrieben durch

$$F(s) = \frac{K}{1 + s \cdot T} \quad (5.3)$$

PT2-Glieder werden allgemein formuliert durch die Übertragungsfunktion

$$F(s) = K \cdot \frac{\omega_0}{s^2 + 2 \cdot D \cdot \omega_0 \cdot s + \omega_0^2} \quad (5.4)$$

In diese Formel fließt die Eigenfrequenz ω_0 des Systems ein. Diese lässt sich durch eine Modellierung des Systems als homogene Differentialgleichung ermitteln, aber sie ist bereits aus den bisherigen Experimenten bekannt: Bringt man das System in Schwingung, dann ist die Amplitude seiner Reaktion abhängig von der Frequenz. Abbildung 4.5 zeigt das Ergebnis eines entsprechenden Experiments an der Basis. Entspricht die Frequenz der Schwingung der Eigenfrequenz des Systems, dann kommt es zu Resonanz, sodass die Amplitude der Systemantwort verhältnismäßig hoch ist, oder sich sogar zunehmend verstärkt. Die höchste Amplitude wurde bei den Sweep - Experimenten um 0.6 Hz festgestellt. Vorerst wird für ω_0 dieser Näherungswert verwendet.

Zur Verkettung zweier Systeme werden deren Übertragungsfunktionen multipliziert (Unb08, 74). Im aperiodisch stabilen Fall eines PT2-Systems, wie es hier vermutet wird, ist dies das Produkt zweier PT1-Systeme mit je einer spezifischen Verzögerungskonstante und einer gemeinsamen Dämpfung ²:

$$F_{PT2,D \geq 1} = \left(\frac{K}{1 + \tau_2 \cdot s} \right) \cdot \left(\frac{1}{1 + \tau_1 \cdot s} \right) \quad (5.5)$$

5.3 Simulation und Validierung

Im Folgenden soll die ermittelte Übertragungsfunktion des Modells numerisch simuliert werden. Um den Grad deren Gültigkeit zu validieren, wird die Simulation abschließend mit den Daten aus den Experimenten am realen System verglichen.

Hierzu wird die Übertragungsfunktion zunächst mittels der inversen Laplace-Transformation in den Zeitbereich überführt. Für die daraus resultierende *Gewichtungsfunktion* $g(t)$ gilt:

$$g(t) = \mathcal{L}^{-1}(F(s))$$

In diesem Fall:

$$g(t) = \mathcal{L}^{-1} \left(\frac{K}{T^2 \cdot s^2 + 2 \cdot D \cdot T \cdot s + 1} \right) \quad (5.6)$$

Für reelle und positive Variablen ist das Ergebnis

$$g(t) = -\frac{e^{t \cdot \left(-\frac{\sqrt{D^2-1}}{T} - \frac{D}{T} \right)} - e^{t \cdot \left(\frac{\sqrt{D^2-1}}{T} - \frac{D}{T} \right)}}{2 \cdot \sqrt{D^2-1} \cdot T} \quad (5.7)$$

Die Gewichtungsfunktion kann jetzt einfach in Software implementiert werden. Nur die Parameter D und T sind noch festzulegen. T ist die Länge des verwendeten zeitdiskreten Abtastintervalls.

In der Simulation bestehen keine Echtzeitanforderungen, da nicht mit einem realen System gearbeitet wird. Sie muss daher nicht hardwarenah, üblicherweise in C, umgesetzt werden. In diesem Fall wurde Ruby verwendet, um die Faltung der Gewichtungsfunktion mit der Heaviside-Funktion zu simulieren.

²siehe auch (Sch08, S. 25)

```

1 F = 0.01
2
3 def f(t) # Heaviside
4     return 0 if t == 0.0
5     return 1
6 end
7
8 def g(t) # Model
9     t1 = 2.29
10    t2 = 2.3
11    dm = t1-t2
12    k * (Math.exp(-t/t1)/dm)-
13        (Math.exp(-t/t2)/dm)
14 end
15
16 File.open("./out.dat", "w") { |f|
17     t = 0.0
18     while(t < 20) do
19         sum = 0
20         tau = 0.0
21         while tau < t do
22             sum += f(tau) * g(t-tau) * F
23             tau += F
24         end
25         f.write([t, f(t), sum].join("\t")+ "\n")
26         t += F
27     end
28 }

```

In den Zeilen 19-24 des Listings ist die bereits ausgeführte numerische Faltung implementiert. Abbildung 5.5 zeigt den Vergleich der simulierten Systemantwort mit dem entsprechenden experimentellen Ergebnis an der Basis, normiert auf einen gemeinsamen, dimensionslosen Darstellungsraum.

[Erklären, warum P-Anteil durch lineares Reibungsmodell hier noch ignoriert]

Obwohl noch nicht alle Modellparameter exakt ermittelt wurden, ist die Übereinstimmung mit dem realen System bereits sehr hoch.

Gemäß den beschriebenen Verfahren zur iterativen Verbesserung eines Reglers wird dieses Modell für erste Tests am System verwendet. Es wird sich dann zeigen, ob und inwiefern die Regelparameter verbessert werden müssen.

Hierzu müssen Regelkreise entworfen, und unter Echtzeit-Anforderungen implementiert werden, für die dann auch bestimmte Stabilitätsanforderungen gelten. Die nächsten Kapitel befassen sich mit dieser Aufgabe.

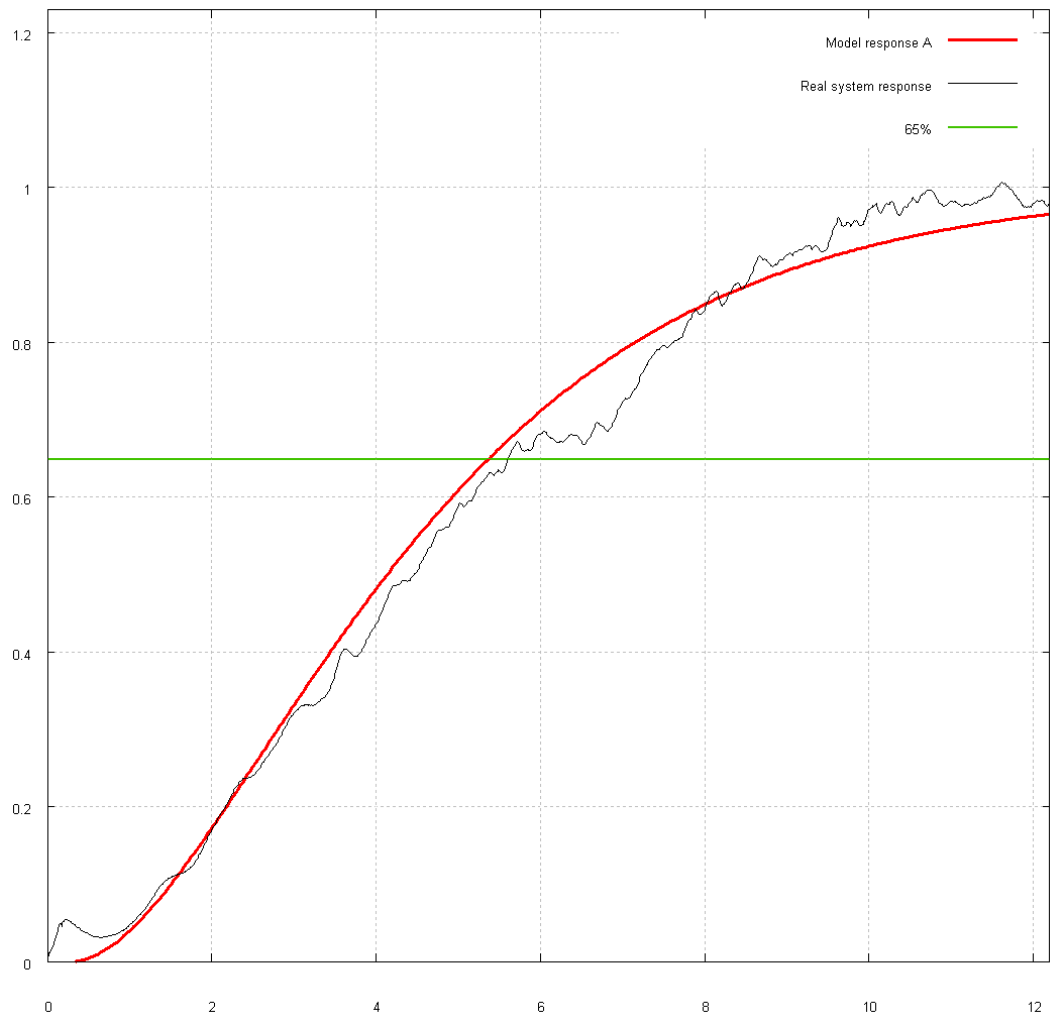


Abbildung 5.5: Modellantwort (rot) mit realer Systemantwort (schwarz)

6 Umsetzung

Nach einigen Iterationsschritten der Modellbildung und der Validierung soll schließlich ein Regelkreis für die Admittanzregelung implementiert werden.

In diesem Kapitel werden Ansätze für den Entwurf des Regelkreises beschrieben, und seine Realisierung in einer Echtzeit-Umgebung. Abschließend werden Methoden vorgestellt,

6.1 Umsetzung einer Geschwindigkeitsregelung

Die vorhandenen Erkenntnisse aus der Modellbildung sollen zunächst exemplarisch für die Umsetzung eines Regelkreises dienen, der die Dynamik der Basis kompensiert, wie sie in den Experimenten beobachtet wurde.

Hierzu kann ein Regelkreis auf unterschiedliche Weise entworfen werden. Den Ansätzen gemein ist die Verwendung eines eingebetteten Kraftreglers zur Kompensation der Streckendynamik. Im Folgenden werden zwei Herangehensweisen vorgestellt: Die Verwendung eines inversen Modells als Vorsteuerung der Regelschleife, und die reine Parametrierung eines Reglers durch Einstellregeln.

6.1.1 Verwendung des inversen Modells

Während das mathematische Modell einer Strecke die erwartete Systemreaktion in Abhängigkeit von den Eingangsgrößen liefert, kann das inverse Modell prognostizieren, welche Eingangsgröße angelegt werden muss, um eine gewünschte Systemantwort zu erhalten. In diesem Fall kann ein inverses Modell zu einer gewünschten Geschwindigkeit den notwendigen Torque ermitteln.

Es wird dem Regelkreis vorgeschaltet und erfährt keine Rückkopplung, dient also als Vorsteuerung zum Regler, der dann nur noch die Abweichung des inversen Modells vom realen Systemverhalten ausgleichen muss (Win06, S. 19)

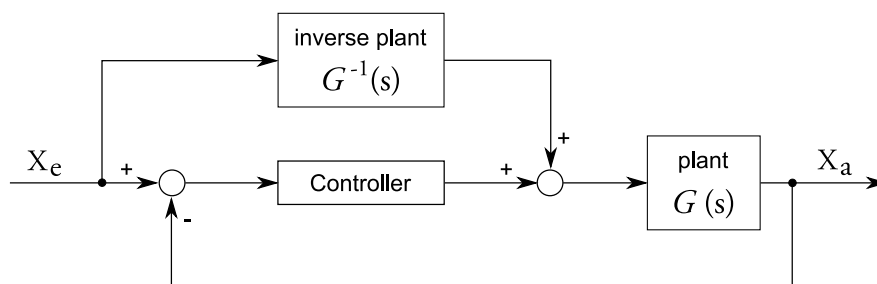


Abbildung 6.1: Reglerentwurf mit inversem Modell als Vorsteuerung

Diese Methode bietet den Vorteil, dass - ein ausreichend genaues Modell der Strecke vorausgesetzt - der benötigte Aktoreneingang im Vorfeld bekannt ist, man also nicht die Reaktion der Strecke abwarten muss, um schließlich nach einigen Durchläufen der Regelschleife den eigentlich erwünschten Effekt zu erreichen.

6.1.2 Parametrierung durch Einstellregeln

Im Vergleich zur Ermittlung eines korrekten Modells der Strecke liefert die Anwendung von Schätzverfahren vergleichsweise schnelle Ergebnisse, die für viele Regelungsaufgaben bereits

ausreichend sind. Zwar bietet ein Regelkreis ohne Vorsteuerung eine geringere Dynamik, doch gerade wegen des geringen Aufwands empfiehlt es sich, zunächst das Ergebnis eines durch Einstellregeln parametrisierten Reglers zu bewerten, bevor unnötigerweise viel Zeit in die mathematische Modellierung investiert wird.

Es wurden daher erste Experimente mit einem PI-Regler angestellt, der mit den Einstellregeln nach Chien, Hrones und Reswick konfiguriert wurde.

Dieses Verfahren ist für Strecken höherer Ordnung anwendbar, wenn Ausgleichszeit T_g und Verzugszeit T_u experimentell am offenen Regelkreis ermittelt werden können. Beide Werte sind wurden bereits ermittelt, um die Regelbarkeit des Prozesses bewerten zu können. Nach (MR04, S. 227) lassen sich die gesuchten Parameter K_P und $K_I = 1/T_N$ für einen PI-Regler mit möglichst günstigem Führungsverhalten und aperiodischen Regelverlauf wie folgt ermitteln:

$$K_P = 0.35 \cdot \frac{T_g}{T_u \cdot K_s} \quad \text{und} \quad T_n = 1.2 \cdot T_g$$

Die Implementierung des resultierenden Regelalgorithmus ist in Listing 6.1 aufgeführt. Daraufhin wurde das Systemverhalten am offenen und geschlossenen Regelkreis verglichen.

Die Ergebnisse dieser Experimente ist in Abbildung 6.2 im Vergleich zum Systemverhalten ohne Regelung dargestellt. Eine überaus deutliche Verbesserung des Führungsverhaltens wird deutlich. Die so erreichte Kompensation der Streckendynamik wird vorerst als ausreichend betrachtet und der Regelkreis für die weitere Umsetzung zumindest zur Kompensation der Trägheit verwendet. Sollte später eine höhere Dynamik des Reglers notwendig werden, so wird er um eine Vorsteuerung erweitert.

Listing 6.1: Auszug aus der Implementierung des PI-Controllers

```

1 double Ks, Kp, Ki, Tn, Tu, Tg, w, d;
2 Ks = 0.5;
3 Tg = 10.8 * 1000;           // multiplied with controller frequency
4 Tu = 1.2 * 1000;           // multiplied with controller frequency
5 Kp = 0.39 * (Tg / (Tu * Ks));
6 Tn = 1.4 * Tg;
7 Ki = 1/(Tn);
8
9 sensor_data_t cur;
10 actor_data_t tar;
11
12 while(true) {
13
14     load_sensor_data(&cur);
15
16     for(int i=0; i<4; i++) {
17
18         // First derivation of position, resolved using
19         // Kalman filter.
20         encoder_1st_der[i] = kalman__der_predict(i, cur.position[i], 1);
21         w = 2100 * heaviside(current_time(), i);
22         d = w - 0.003 * encoder_1st_der[i];
23
24         tar.torque_set_value[i] = (Kp * d + Ki * integral_d[i]);
25         integrate(&integral_d, d, i);
26     }
27
28     send_actor_data(&tar);
29 }

```

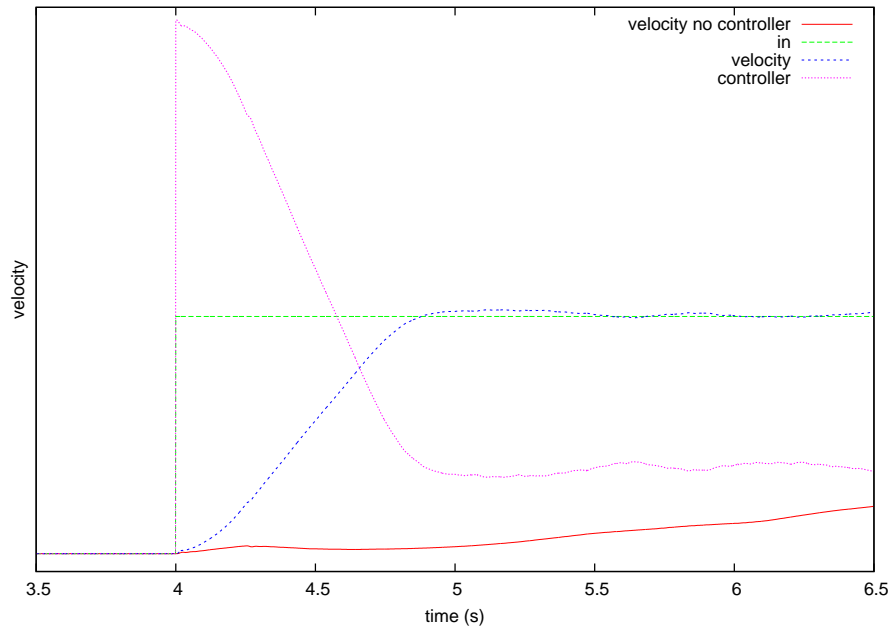



Abbildung 6.2: Übertragungsverhalten der Basis mit und ohne PI-Controller

6.2 Entwurf des Admittanzreglers

6.2.1 Grundlage der Stellregeln

Das Wunschverhalten als Masse-Dämpfer-Feder-System wie es für Impedanz- und Admittanzregler üblich ist, wurde bereits in Gleichung 3.1 beschrieben. Dem Verhalten des Reglers zufolge ist eine *Kontaktkraft* F_K gegeben, zu der eine Positionsdiffrenz ermittelt wird, die auf die Stellgröße des Lagereglers aufgeschalten wird. Ist die Kontaktkraft verfügbar, so muss nur noch die Differentialgleichung zum Wunschverhalten gelöst werden.

Diese Lösung ist am einfachsten zu ermitteln, indem die Differentialgleichung zunächst in den s-Bereich transformiert wird¹:

$$\frac{F(s)}{x(s)} = m \cdot s^2 + d \cdot s + k \quad (6.1)$$

Das Stellgesetz wird durch die Umkehrung von Ursache- und Wirkungsfunktion im s-Bereich ermittelt:

$$\frac{p_{soll}(s)}{F_{ist}(s)} = \frac{1}{m \cdot s^2 + d \cdot s + k}$$

Als Lösung für die Steifigkeit ergibt sich

$$\frac{p_{soll}(s)}{F_{ist}(s)} = \frac{1}{k}$$

¹Diese Herleitung und Details finden sich u.a. in (Bus04, 2-5) und (Win06, S. 21)

und für die Dämpfungsregelung

$$\frac{\dot{p}_{soll}(s)}{F_{ist}(s)} = \frac{1}{d}$$

Da die Admittanz die Umkehrung der Impedanz ist, muss zur Ermittlung ihrer Lösungen der Kehrwert zu Gleichung 6.1 gebildet werden. Die Admittanzregelung bildet demnach nicht die Kontaktkraft aus der Sollposition, sondern eine Positions- oder Geschwindigkeitsänderung aus einer gegebenen Sollkraft (Win06, S. 23). Da eine Impedanzregelung die Kontaktkraft über eine Positionsveränderung regelt, bezeichnet man sie als *indirekte Kraft- / Momentenregelung* im Gegensatz zur *direkten Kraft- / Momentenregelung* wie sie die Admittanzregelung darstellt.

Im Verlauf der Umsetzung hat sich nicht endgültig herausgestellt, welche der beiden Herangehensweisen für die gegebene Problemstellung vorzuziehen ist. Im Folgenden werden Ansätze für Umsetzungen beider Varianten beschrieben.

6.2.2 Besonderheiten des Regelkreises

In der Literatur zur Kraft- und Momentenregelung, speziell zum Entwurf von Impedanzreglern, wird implizit gefordert, dass die Kontaktkraft F_K von einem speziellen Sensor zur Verfügung gestellt wird. Hier ist das aber nicht der Fall: Der Einfluss des Benutzers kann entweder als Positionsveränderung der Basis, oder als zusätzlicher Drehmoment von den Motor-Controllern ermittelt werden, also immer in Kombination mit dem von der Regelschleife selbst verursachten Aktorenausgängen.

Sobald beispielsweise die Basis über den Drehmomenten-Regler bewegt wird und der Benutzer die Basis der entstehenden Bewegung entgegen drückt, kann nicht ohne Weiteres ermittelt werden, welcher Anteil des resultierenden Drehmoments durch äußeren Einfluss entstanden ist.

Es müssen demnach Ansätze entwickelt werden, die es zunächst ermöglichen, die Kontaktkraft getrennt zu ermitteln.

6.2.3 Entwurf einer Impedanzregelung

Abbildung 6.3 zeigt den Entwurf eines Impedanz-Regelkreises. Der Impedanzregler erhält die Ist-Kraft, also die Summe von Kontaktkraft und durch den Regelkreis gesteuerte Kraft, als Drehmoment. Diese ist für die Anfangsbedingung 0. Ändert sich die Ist-Kraft durch Beeinflussung des Benutzers, so wird aus dieser die Impedanz berechnet. Der Ausgang des Impedanzreglers ist der modifizierte Positionswert, und seine zweite Ableitung als Beschleunigung. Mit einem inversen Modell als Vorsteuerung zum Drehmomenten-Regler wird die für die gewünschte Positionsveränderung notwendige Kraft aus dem Beschleunigungswert des Impedanzreglers ermittelt. Im einfachsten Fall beinhaltet das inverse Modell nur die Trägheit der Basis, also $F_{fwd} = m \cdot a$.

Für ein ideales inverses Modell wird erwartet, dass der Positionsregler keinen Regelfehler mehr feststellen kann, da die Vorsteuerung bereits den exakt benötigten Wert für die Positionsveränderung gesetzt hat. Die als Vorsteuerung gesetzte Kraft wird dem Kraft-Ist-Wert abgezogen und das Ergebnis als ermittelte Kontaktkraft an die Impedanzregelung übergeben. Auf diese Weise wurde die externe Kontaktkraft demnach von der Summe aller Kräfte, die als Drehmoment gemessen wurden, herausgezogen.

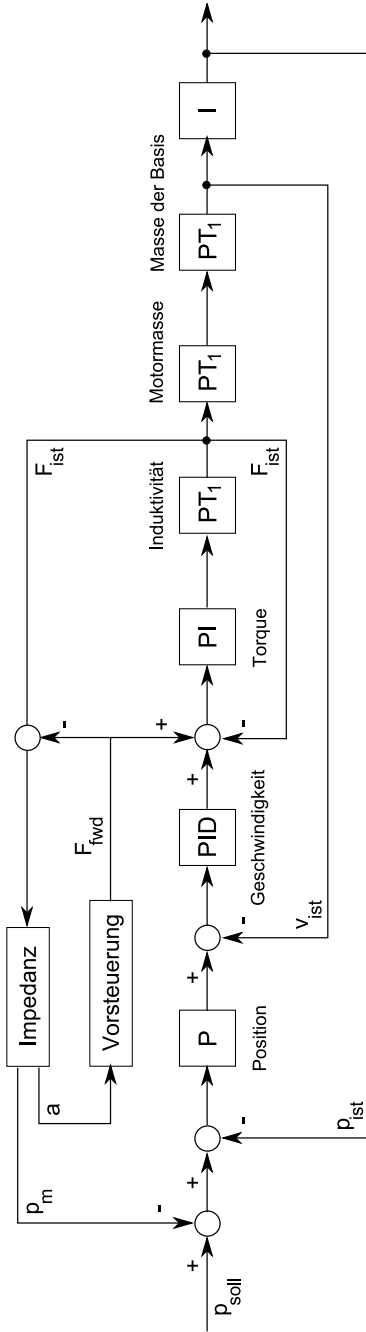


Abbildung 6.3: Blockdiagramm Impedanzregelung mit Vorwärtsmodell in der Vorsteuerung

6.2.4 Entwurf einer Admittanzregelung

In der Umsetzung hat sich der vorgestellte Ansatz zur Impedanzregelung als äußerst schwierig herausgestellt, was unter anderem der zusätzlichen Regelkreisdynamik durch die kaskadierte Kraftsteuerung zugeschrieben wurde. Letztendlich ist aber auch nicht geklärt, ob der Ansatz überhaupt korrekt ist und wie beabsichtigt umgesetzt werden kann.

Alternativ wurde ein Admittanzregelkreis entworfen wie er in Abbildung 6.4 dargestellt ist, der mit vermutlich einfacher umgesetzt werden kann. Ein der Admittanzregelung übergeordneter Kraftregler (PI) gleicht die Dynamik der Strecke aus. Dies kann mit dem bereits vorgestellten Geschwindigkeitsregler erreicht werden. Die Kontaktkraft wird aus der Differenz eines Vorwärtsmodells der Strecke und dem tatsächlichen Systemzustand gebildet. Diese wird von der Sollgröße des Kraftreglers abgezogen, sodass die Drehmomente verringert oder umgekehrt werden. Zwischen Vorwärtsmodellvergleich und dem Kraftregler bestimmt ein Admittanzregler den Grad der Nachgiebigkeit des Systems, indem den tatsächlich aktuierten Drehmomenten eine Differenzkraft aufgeschaltet wird.

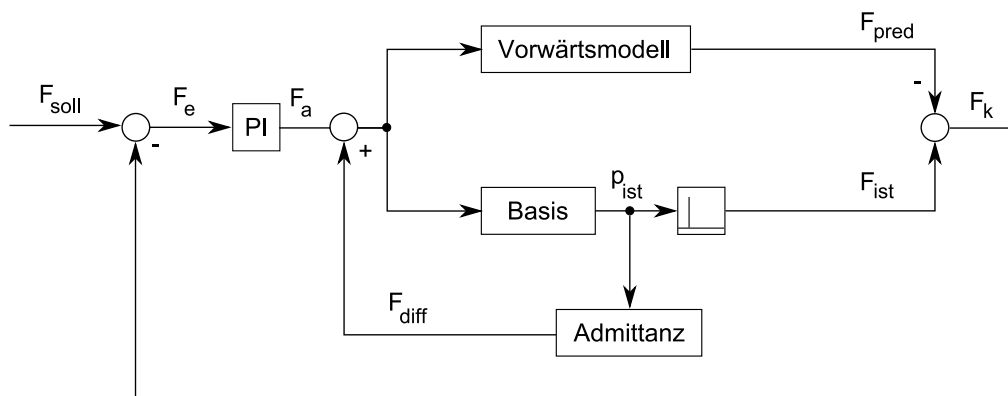


Abbildung 6.4: Blockdiagramm Admittanzregelung mit Vorwärtsmodell und Komparator

6.3 Vorbereitung eines Echtzeit-Betriebssystems

6.3.1 Notwendigkeit einer Echtzeit-Umgebung

Um sich des Anspruchs des Begriffs *Echtzeit* bewusst zu werden, muss er zunächst für den hier gegebenen Kontext abgegrenzt werden.

Oft wird von Echtzeit gesprochen, wenn die Rechenzeit bis zur Reaktion einer Anwendung für den Menschen unmerklich klein ist. Dies ist zum Beispiel in 3D-Anwendungen der Fall, in denen das Bild zur Laufzeit berechnet und noch während der Interaktion sichtbar wird. Beispielsweise wird die Darstellung von mindestens 25 Bildern pro Sekunde angestrebt, es ist aber nicht problematisch, wenn es im Einzelfall weniger sind. In diesem Zusammenhang handelt es sich um sogenannte *Weiche Echtzeit* oder *Soft Real-time*.

Für die Messung, Steuerung und somit Regelung eines realen Systems kann aber nicht akzeptiert werden, dass einzelne Datenpakete mangels verfügbarer Rechenzeit verloren gehen. Es ist klar, dass ein Regler nicht zuverlässig arbeiten kann, wenn nicht gesichert ist, dass die Stellgrößen am System ankommen oder Sensordaten vollständig sind. Beinhaltet eine Anwendung kritische Zeitintervalle, so spricht man von *Harter Echtzeit* oder *Hard Real-Time* (Sup05, S. 118), die in diesem Abschnitt behandelt wird.

Dies ist insofern relevant, als dass durch die Berechnungen im digitalen Regelkreis eine *Rechentotzeit* entsteht, die näherungsweise der Abtastzeit T_A entspricht. Erst wenn T_A im Vergleich zu den kleinsten kritischen Zeitkonstanten sehr klein ist, kann von einer quasikontinuierlichen Regelung gesprochen werden (Web07).

Hohe Rechenleistung reicht für Echtzeitumgebungen also lange nicht aus. Die Umgebung muss zudem vorhersagbare Zeitintervalle einhalten, um Jitter zu vermeiden. Selbst auf den leistungsstärksten Systemen wäre dies ohne ein entsprechend ausgelegtes Betriebssystem

nicht möglich, da die Ausführungszeitpunkte eines Prozesses - und somit auch die Ausführung der Echtzeit-Schleife - beliebig vom Scheduler gesetzt werden können. Auch bei höchster Priorisierung eines Prozesses gibt es keine Garantie, dass er nicht immer wieder zum gewünschten Takt verdrängt wird. Ein besonderes Problem stellen in diesem Zusammenhang Interrupts dar, die noch schwerer beeinflusst werden können, zumal sie von der Hardware ausgelöst werden.

Zudem bieten Echtzeit-Betriebssysteme für die Ermittlung von Periodendauern eine zuverlässige Auflösung der Hardware-Zeit bis in den Nanosekundenbereich.

6.3.2 RTAI

Bekannte Betriebssysteme mit diesem Anspruch sind beispielsweise VXWorks und QNX. In den letzten Jahren wurde zudem RTAI in Industrie und Forschung immer beliebter. RTAI ist ein Patch für den (Vanilla und Debian) Linux Kernel, der diesen dahingehend verändert, dass ein weiterer echtzeitoptimierter Scheduler noch vor die Ausführungspriorität des Linux-Kernels eingehängt werden kann. Somit ist das eigentliche Betriebssystem den in Echtzeit ausgeführten Prozessen untergeordnet, und kann diese auch nicht verdrängen. Allerdings haben manche Interrupts immer noch Vorrang gegenüber dem RTAI-Scheduler.

Im Vergleich zu dedizierten Echtzeit-Betriebssystemen darf RTAI-Linux nicht als ideales Echtzeitsystem bezeichnet werden. Auch nach den Kernel-Patches sind spezielle Anpassungen notwendig, um so wenige Interrupts wie möglich zu erzeugen.

Auch in diesem Fall mussten im BIOS des Rechners auf der Basis diverse Einstellungen vorgenommen werden, um diesen Problemen zu begegnen. Unter anderem wurden ACPI und USB deaktiviert, da beide Funktionen problematische Interrupts erzeugen und verzichtbar sind.

Abbildung 6.5 zeigt schematisch die Architektur eines RTAI-Linux-Betriebssystems. Sie zeigt unter anderem, dass

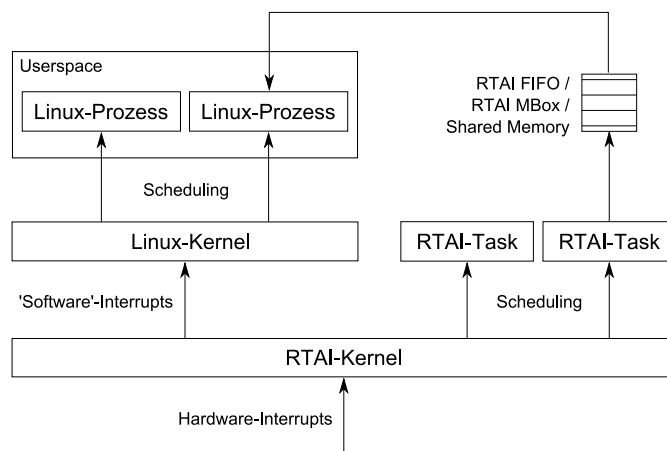


Abbildung 6.5: Architektur eines RTAI-Linux-Betriebssystems (nach (Abb03, S. 159))

6.4 Interprozesskommunikation unter RTAI

Ein wesentlicher Aspekt im Entwurf von RTAI-Anwendungen ist die Kommunikation zwischen dem Kernel-Modul und der Anwendung im User Space.

Mit gewöhnlichen Kernelmodulen wird üblicherweise mittels der Funktion `ioctl` kommuniziert. Dies ist dann unproblematisch, wenn die Kommunikation beider Seiten entweder nicht synchron sein muss, oder der Prozess so langsam erfolgt, dass die Übertragung nicht gefährdet ist.

Bei Echtzeit-Anwendungen wird zu jedem Prozesstakt mindestens ein Datenpaket zwischen Kernel-Modul und User Space übertragen. Die Kommunikation ist also höchst zeitkritisch.

Geht man von einer Messchleife von 2 kHz und einer Paketgröße von 64 Byte aus, so wie es hier der Fall ist, dann ist klar, dass die Kommunikation zusätzlich abgesichert werden muss. Die Grundproblematik der Datenübertragung entspricht dem bekannten Erzeuger/Verbraucher-Problem: Echtzeit-Anwendung und der lesende Prozess im Userspace müssen dank einer FIFO zwar nicht in einem gemeinsamen Takt laufen, allerdings muss sichergestellt sein, dass der lesende Prozess die übertragenen Daten schnell genug verarbeiten und somit schnell genug aus dem FIFO-Puffer lesen kann, da sonst Werte verloren gehen würden.

Listing 6.2: Grundgerüst eines Kernel-Moduls mit RTAI

```

1 #include <linux/module.h>
2 #include <asm/io.h>
3 #include <math.h>
4 #include <rtai.h>
5 #include <rtai_sched.h>
6 #include <rtai_fifos.h>
7
8 #define TICK_PERIOD 1000000
9 #define TASK_PRIORITY 1
10 #define STACK_SIZE 10000
11 #define FIFO 0
12
13 static RT_TASK rt_task;
14
15 static void rt_loop(int t)
16 {
17     int count = 0;
18     float value;
19
20     while (1) {
21         value = sin(2*M_PI*1*rt_get_cpu_time_ns()/1E9);
22         rtf_put(FIFO, &count, sizeof(count));
23         rtf_put(FIFO, &value, sizeof(value));
24         count++;
25         rt_task_wait_period();
26     }
27 }
28
29 int init_module(void)
30 {
31     RTIME tick_period;
32     rt_set_periodic_mode();
33     rt_task_init(&rt_task, rt_loop, 1,
34                 STACK_SIZE, TASK_PRIORITY, 1, 0);
35     rtf_create(FIFO, 8000);
36     tick_period = start_rt_timer(nano2count(TICK_PERIOD));
37     rt_task_make_periodic(&rt_task,
38                          rt_get_time() + tick_period,
39                          tick_period);
40     return 0;
41 }
42
43 void cleanup_module(void)
44 {
45     stop_rt_timer();
46     rtf_destroy(FIFO);
47     rt_task_delete(&rt_task);
48     return;
49 }

```

Listing 6.3: Grundgerüst einer RTAI-Anwendung im Userspace

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <sys/types.h>
4 #include <sys/mman.h>
5 #include <sys/stat.h>
6 #include <fcntl.h>
7 #include <signal.h>
8
9 #define FIFO_DEV "/dev/rtf0"
10
11 static int end;
12
13 static void sigint_handler(int dummy)
14 {
15     end = 1;
16 }
17
18 int main (void)
19 {
20     int fd_fifo, count;
21     float value;
22
23     if ((fd_fifo = open(FIFO_DEV, O_RDONLY)) < 0) {
24         fprintf(stderr, "Error opening %s: %d\n",
25                 FIFO_DEV, fd_fifo);
26         exit(1);
27     }
28
29     signal(SIGINT, sigint_handler);
30
31     while (!end) {
32         read(fd_fifo, &count, sizeof(count));
33         read(fd_fifo, &value, sizeof(value));
34
35         printf("Count: %d Value: %f\n", count, value);
36     }
37
38     return 0;
39 }
```


7 Bewertung und Ausblick

Zu Beginn dieser Arbeit wurden die Grundlagen der Regelungstechnik und der Systembegriff eingeführt. Darauf aufbauend wurden das Prinzip des Greybox-Modelling sowie Methoden der experimentellen und analytischen Systemidentifikation dargelegt. Abschließend wurden Entwürfe für die Impedanz- und Admittanzregelung des bearbeiteten Modells erläutert und Ansätze zu ihrer Implementierung unter RTAI Linux entwickelt.

In der Kürze der Zeit konnte neben einem Geschwindigkeitsregler keiner der Ansätze tatsächlich vollständig realisiert werden, in der kommenden Entwicklung wird an dieser Stelle aber weiterentwickelt und die Implementierung der Admittanzregelung durch weitere Experimente untersucht und stabilisiert. Es ist geplant, anschließend unter Verwendung der bereits vorhandenen Sensorik an der Basis Kontrollstrukturen einzuführen, um eine Gefährdung der Umwelt durch ihr Verhalten auszuschließen.

Es hat mich gefreut, zu sehen, wie die Informatik dem interdisziplinären Feld der Regelungstechnik beitragen kann, und somit auch festzustellen, wie sie in anderen Aufgabengebieten eingebettet wird und ihre stets postulierte analytische Vorgehensweise zum Tragen kommt.

Abbildungsverzeichnis

2.1	Einfaches Beispiel eines geschlossenen Regelkreises	6
2.2	Aufbau eines geschlossenen Regelkreises	7
2.3	Implementierung der einer diskreten, linearen Faltung in Ruby	10
2.4	Provokation von Schätzfehlern eines Kalman-Filters	11
2.5	Bildung der zweiten Ableitung durch 5-Punkt-Methode	12
2.6	Schematische Darstellung von Ursache-, System- und Wirkungsfunktion . . .	13
2.7	Übergangsfunktion P-System	13
2.8	Übergangsfunktion I-System	14
2.9	Übergangsfunktion D-System und PT1-System	15
2.10	Übergangsfunktion nicht-schwingungsfähiges und schwingungsfähiges PT2-System	16
2.11	Übergangsfunktion T-System	16
2.12	Klassifizierung von PT2-Systemen nach Ausgleichszeit und Verzugszeit	17
2.13	Parallelschaltung und Verkettung zweier systemdynamischer Bausteine	18
2.14	Allgemeiner Ablauf der Greybox=Modellierung	20
3.1	Mobile Basis am Lehrstuhl für Bildverstehen und wissensbasierte Systeme der TU München	21
3.2	Schema der Basis mit Koordinatensystemen der Räder (Quelle: (Sal06)) . . .	22
3.3	Schema eines Mecanum Wheels mit 8 Rollen (Quelle: (Sal06))	22
3.4	Freiheitsgrade der Basis mit ihrer Steuerung (nach (Sal06))	22
3.5	Blockdiagramm Impedanzregelung (nach (Win06, S. 21))	24
4.1	Allgemeine konzeptionelle Beschreibung eines Systems	25
4.2	Schema der Messschleife mit offenem Regelkreis	26
4.3	Ergebnis der Experimentreihe zum linearen Verhalten mit Verlauf in der Beschleunigung	28
4.4	Sprungfunktion auf Torque (schwarz) und Sprungantwort (rot)	29
4.5	Systemantwort auf Torque-Sweep	29
4.6	Sinus-Sweeps (rot) mit Frequenzgang (grün)	30
4.7	Einzelne Sinusfolge (rot) mit Frequenzgang (grün)	30
5.1	Coulomb-Reibung und viskose Reibung, schematisch	33
5.2	Haftreibung und Stribeck-Effekt, schematisch	33
5.3	Experimentelle Ergebnisse zur Reibungsmodellierung (v:t)	34
5.4	Experimentelle Ergebnisse zur Reibungsmodellierung im Durchschnitt (v:p) .	35
5.5	Modellantwort (rot) mit realer Systemantwort (schwarz)	39
6.1	Reglerentwurf mit inversem Modell als Vorsteuerung	40
6.2	Übertragungsverhalten der Basis mit und ohne PI-Controller	42
6.3	Blockdiagramm Impedanzregelung mit Vorwärtsmodell in der Vorsteuerung .	44
6.4	Blockdiagramm Admittanzregelung mit Vorwärtsmodell und Komparator . .	45
6.5	Architektur eines RTAI-Linux-Betriebssystems (nach (Abb03, S. 159))	46

Literaturverzeichnis

- [Abb03] ABBOTT, Doug: *Linux for embedded and real-time applications*. Newnes, 2003. – ISBN 978-0-750-67546-88
- [AS02] ALBU-SCHÄFFER, Alin: *Regelung von Robotern mit elastischen Gelenken am Beispiel der DLR-Leichtbauarme*, Lehrstuhl für Steuerungs- und Regelungstechnik der TU München, Diss., 2002
- [Bau07] BAUMÜLLER NÜRNBERG GMBH (Hrsg.): *b maXX 4400 Motor Parameter Manual*. 5.03039.06. Ostendstr. 80 - 90 / D-90482 Nürnberg / Germany: Baumüller Nürnberg GmbH, 11 2007. – Firmware Version 03
- [Bög07] BÖGE, Alfred (Hrsg.): *Vieweg Handbuch Maschinenbau*. Vieweg+Teubner, 2007 . – ISBN 978-3-8348-0110-4 (Print) 978-3-8348-9092-4 (Online)
- [Bus04] BUSS, Martin: *Steuerungs- und Regelungstechnik I*. Lehrstuhl für Steuerungs- und Regelungstechnik der TU München, 6 2004. – Skriptum zur Vorlesung Steuerungs- und Regelungstechnik I
- [Cod05] CODRONS, Benoît (Hrsg.): *Process Modelling for Control*. Springer London, 2005 (Advances in Industrial Control). – ISBN 978-1-85233-918-0 (Print) 978-1-84628-247-8 (Online)
- [GW01] GÜNTER WENISCH, Fritz B. (Hrsg.): *Lehr- und Übungsbuch numerische Mathematik*. Fachbuchverlag Leipzig, 2001 . – ISBN 978-3446213753
- [KMH01] KATALIN M. HANGOS, I. T. C. (Hrsg.): *Process Modelling and Model Analysis*. Academic Press, 2001 (Process Systems Engineering). – ISBN 978-0121569310
- [MR04] MANFRED REUTER, Serge Z.: *Regelungstechnik für Ingenieure: Analyse, Simulation und Entwurf von Regelkreisen*. 12. Vieweg Verlag, 2004. – ISBN 978-3-8348-0018-3
- [Sal06] SALIH, Jefri Efendi M.: Designing Omni-Directional Mobile Robot with Mecanum Wheel. In: *American Journal of Applied Sciences 3 (5): 1831-1835, 2006* (2006). – ISSN 1546-9239
- [Sch08] SCHUSTER, Hans: *Hochverstärkungsbasierte Regelung nichtlinearer Antriebssysteme*, Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München, Diss., 2008
- [Shi05] SHIMADA, A.: Mecanum-wheel Vehicle Systems Based on Position Corrective Control. In: *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE 6* (2005), November, Nr. 10. <http://dx.doi.org/10.1109/IECON.2005.1569224>. – DOI 10.1109/IECON.2005.1569224
- [shr08] *Springer Handbook of Robotics*. Bd. 1. Springer Berlin / Heidelberg, 2008 . – ISBN 978-3-540-23957-4
- [Söd99] SÖDER, Günter: *Skriptum: Simulationenmethoden in der Nachrichtentechnik*. Lehrstuhl für Nachrichtentechnik der Technischen Hochschule München, 3 1999
- [Sup05] SUPRIJADI, Eko B.: *Kinematische Echtzeit-Regelung und auf vereinfachter Dynamik basierte Kraftregelung einer vierbeinigen Gehmaschine*, Fachbereich Ingenieurwissenschaften. Abteilung Maschinenbau der Universität Duisburg-Essen, Diss., 2005

- [Unb08] UNBEHAUEN, Heinz (Hrsg.): *Regelungstechnik I*. Vieweg+Teubner, 2008 . – ISBN 978-3-8348-0497-6 (Print) 978-3-8348-9491-5 (Online)
- [Web07] WEBER, Wolfgang: *Industrieroboter: Methoden der Steuerung und Regelung*. 2. Hanser Verlag, 2007. – ISBN 978-3-446-41031-2
- [Win06] WINKLER, Alexander: *Ein Beitrag zur kraftbasierten Mensch-Roboter-Interaktion*, Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität Chemnitz, Diss., 2006