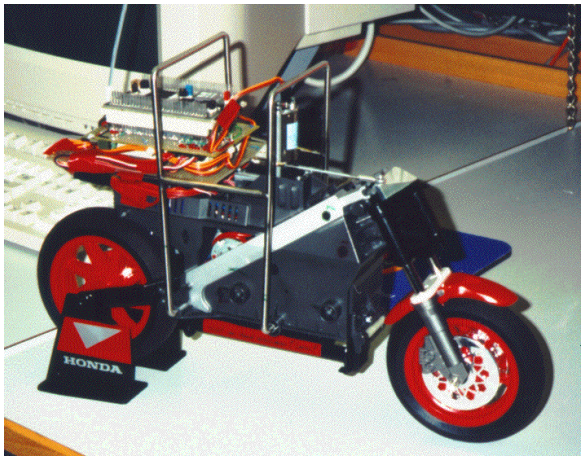
Semesterarbeit autonomes Zweirad

Modellierung, Konstruktion und Implementation eines autonom fahrenden Zweiradmodells

Ausgabe

Dateiname:	Seiten:	Autoren:	Datum:
bericht.doc	179	M. Neff, R. Odermatt, A. Stucki	06.02.98



Rechte

Für dieses Dokument behalten wir uns alle Rechte vor.

© Copyright 1998 M. Neff, R. Odermatt, A. Stucki

© Copyright 1998 SA

Inhaltsverzeichnis

1. Vorwort 5
2. Aufgabenstellung 6
3. Einleitung 9
4. Motorrad 11
4.1. Elektrobike 11
4.2. Kleinfahrrad 12
4.3. Eigenbau eines Zweirades 12
4.4. Modellbauzweirad 13
4.4.1. Modellmotorradtypen 13
4.5. Bericht zum Besuch bei Graupner 15
4.6. Funktionsweise des gesteuerten Modells 16
4.7. Umbau des Motorrades 18
4.7.1. Originalzustand 18
4.7.2. Umbau 19
4.7.3. Tips und Tricks zum Umbau 21
4.8. Prüfstand 22
5. Sensoren / Aktoren / Modellbauteile 23
5.1. Lagesensor 23
5.1.1. Gyrometer 23
5.1.2. Inklinometer 24
5.2. Geschwindigkeitsmessglied 24
5.2.1. Evaluierung eines Geschwindigkeitssensors 24
5.3. Servo 27
5.4. Modellbauteile 28
5.4.1. Fahrtregler 28
5.4.2. Empfänger 28
5.4.3. Fernsteuerung 29
5.4.4. Leistungsakku 30
5.4.5. Ladegerät 31
6. Modellierung 32
6.1. Modellierung des Motorrades 32
6.1.1. Modellierung als Vielkörpersystem 32
6.1.2. Modellierung als invertiertes Pendel 33
6.2. Modellierung des Inklinometers 40
6.2.1. Eingangswinkel des Inklinometers 40
6.2.2. Identifikation der Übertragungsfunktion des Inklinometers 41
6.3. Modellierung des Servos 44
6.4. Bestimmung der Parameter 45
6.5. Abschätzung des Modellierungsfehlers 48
6.5.1. Auflagepunkt des Vorderrades 48
6.6. Gesamtmodell der Strecke 56
7. Regler 57
7.1. Regelkonzept 57
7.2. Polfestlegung 59
7.2.1. Fehlerrückführung 59
7.2.2. Zustandsrückführung 59
8. Hardware 61

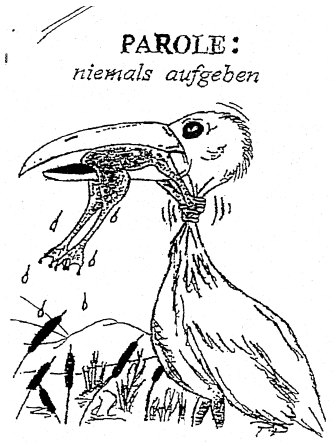
8.1. Wahl des Rechner ... 61
8.2. Aufgaben der Hardware ... 61
8.2.1. Eingangs-und Ausgangssignale ... 61
8.2.2. Codierung der Signale und Bindung an die Hardware ... 62
8.3. Der Microcontroller MC68332 ... 64
8.3.1. Time Processor Unit (TPU) ... 65
8.3.2. Queued Serial Modul (QSM) ... 67
8.3.3. System Integration Modul (SIM) ... 71
8.4. Das Evaluationboard M68332EVS ... 73
8.5. Das Zusatzboard ... 75
8.5.1. Spannungsversorgung ... 75
8.5.2. Pinbelegung der Pfostenstecker ... 75
8.5.3. Modusumschaltung ... 76
8.5.4. Ein- und Ausschlaufen der Signale ... 76
8.5.5. AD-Wandler ... 77
9. Software ... 78
9.1. Programmablauf ... 78
9.2. Einteilung der Software in Module ... 79
9.2.1. Tpu_func.c ... 79
9.2.2. Qsm_func.c ... 80
9.2.3. Sim_func.c ... 81
9.2.4. Reg_func.c ... 81
10. Zusammenfassung ... 82
11. Ausblick ... 83
12. Anhang ... 84
12.1. Literaturverzeichnis ... 84
12.2. Verwendete Variablen ... 85
12.3. Wichtige Adressen ... 86
12.4. Datenblätter ... 88
12.4.1. Modellbau ... 88
12.4.2. Sensoren ... 88
12.4.3. Elektronik Bauteile Zusatzboard ... 88
12.5. MatLab-File für Regler und Simulation ... 167
12.6. C-Programme ... 170
12.6.1. main.c ... 170
12.6.2. Tpu_func.c ... 170
12.6.3. Qsm_func.c ... 175
12.6.4. Sim_func.c ... 177
12.6.5. Reg_func.c ... 178

 © Copyright 1998 SA

1. Vorwort

Der Traum von Freiheit eines jeden spiegelt sich doch oft in der Art seines Fahrzeuges. Ein Motorrad ist schnell, wendig und steht für ein bestimmtes Lebensgefühl: "Freiheit". Nur frei ist das Motorrad selbst nicht. Es braucht einen Fahrer, der es durch die Welt steuert. Ganz auf sich gestellt würde es nicht allzu weit kommen. Könnte man nicht dem Motorrad diese Freiheit, welche es verkörpert, mitgeben? Ein Motorrad als eingebettetes System mit einem Prozessor, Sensoren und Aktoren, welche es vor dem Umfallen bewahren und auf der vorgegebenen Bahn halten, könnte diesen Traum ermöglichen. Mit eigener Energieversorgung wäre so das Motorrad endlich unabhängig und dem Traum von Freiheit ein grosses Stück näher.

Die Zielsetzung für diese Semsterarbeit war einfach formuliert : "Es soll ein autonom fahrendes Zweirad erstellt werden". Somit musste zuerst ein passendes Zweirad gefunden werden, welches der Verwendung als Anschauungsobjekt einer Vorlesung gerecht wurde. Bereits hier zeigte sich, in wie viele Bereiche ein solches Projekt hineinreicht. Nebst Mechanik und Elektrotechnik spielten nun auch Themen wie Material und Preis eine wichtige Rolle. Diese für uns unbekannten Aspekte ermöglichten es uns, einen Einblick in die vor uns liegende Welt des Ingenieurs zu erhalten. Die Probleme stecken bekanntermassen im Detail. Über Lieferprobleme, komplexen mathematischen Modellierungsaufgaben bis hin zu defekten Bauteilen hangelten wir uns durch zu einem fertigen Prototypen. Ganz im Sinne unseres kleinen Mottos : "niemals aufgeben".



Trotz aller Schwierigkeiten war die Arbeit sehr lehrreich und machte viel Spass. Wir werden mit Interesse die weiteren Schritte dieser Arbeit verfolgen und wünschen den Studenten, welche hier weiterarbeiten, viel Geduld und den nötigen Mut, um diese Vision wahr werden zu lassen.

Dank der hervorragenden Zusammenarbeit mit Herrn M.Bühler von der Elektrotechnik-Werkstatt, der uns mit seinem Fachwissen grosszügig und hilfreich unterstützt hat, konnte die benötigte Mechanik für den Umbau des Motorrades in kürzester Zeit bereitgestellt werden.

Wir bedanken uns herzlich bei unseren Betreuern, Prof. Dr. Glattfelder und Matthias Gries, welche uns jederzeit mit ihren kompetenten Ratschlägen zur Seite standen und durch ihren unermüdlichen Einsatz und ihre grosszügige Unterstützung einen Grossteil zum Erfolg dieser Arbeit beigetragen haben.

Zürich, den 6. Februar 1998

M. Neff R. Odermatt A. Stucki

2. Aufgabenstellung

© Copyright 1998 SA

© Copyright 1998 SA

3. Einleitung

Viele Geräte in unserer Umwelt sind eingebettete Systeme, deren Leistung wir im Alltag manchmal gar nicht wahrnehmen, da die Elektonik unter Abdeckungen steckt – gut getarnt und miniaturisiert. Eingebettete Systeme bestehen aus einem oder mehreren Rechner, welche über Kommunikationsschnittstellen, Sensoren und Aktoren mit der Umwelt kommunizieren und ihren Verlauf aufnehmen und beeinflussen.

Wie bringt man Studenten bei, was ein eingebettetes System ist und welche Probleme zu lösen sind? Mit einem praktischen, sichtbaren Beispiel wird das Verständnis für technische Lösungen gefördert. Diese Arbeit schafft ein Anschauungsmodell, welches man als offenes eingebettetes System bezeichnen kann. Ein Zweirad wird mit einer selbständigen Rechnereinheit erweitert, um eine autonome Fahrweise zu ermöglichen.

Die Arbeit ist ein Konglomerat aus verschiedenen Ingenieurgebieten: Mechanik, Regelungstechnik, Konstruktion, Analog- und Digitalelektronik, Informatik und Hardware/Software-Codesign. Deshalb wird das Projekt interdisziplinär sowohl vom Institut für technische Informatik und Kommunikationsnetze, als auch vom Institut für Automatik betreut.

Um das Ziel eines autonom fahrenden Zweiradmodells zu erreichen, wird in folgenden Teilschritten vorgegangen:

Komponentenevaluation und Umbau des Zweirades

Um das Zweirad in einer Vorlesung einzusetzen, soll es als Anschauungsobjekt im Hörsaal verwendbar sein. Sensoren und Aktoren müssen evaluiert, beschafft und montiert werden. Die Kosten der Komponenten dürfen einen vertretbaren Rahmen nicht überschreiten. Da weiterführende Arbeiten geplant sind, muss das umgebaute Zweirad einfach erweitert werden können.

Modellierung eines Zweirades

Zuerst muss das Verhalten des Originalzweirades verstanden werden. Mit diesem Verständnis kann das Zweirad effizient auf ein Modell abgebildet werden und es kann beurteilt werden, welche Teile man vernachlässigen kann. Dabei soll Schritt für Schritt von einer einfachen zu einer komplexen Mechanik modelliert werden.

Auslegung und Simulation eines Reglers

Anhand der Modellierung wird ein Regler ausgelegt und in Simulationen getestet. Der Regler soll vorerst einfach gehalten werden und mit wenigen Parametern einstellbar sein, um das Implementieren und Testen am realen Zweirad handhabbar zu machen. Später kann ein erweitertes Reglerkonzept entwickelt und optimiert werden.

Anhängen einer Rechnerplatine

Eine Rechnerplatine, welche den Regleralgorithmus durchführt, soll auf das Zweirad aufgebaut werden. Es müssen Anforderungen bezüglich Grösse, Energieverbrauch, Rechnerleistung und Speicherbedarf erfüllt werden. Alle nötigen Signale werden über getestete Schnittstellen an den Prozessor angehängt.

Schreiben der Software

Der Regleralgorithmus soll in strukturierter und übersichtlicher Form im Rechner implementiert werden. Die Routinen für die externe Kommunikation sollen in separaten Programmen ausgeführt werden und einfach aufzurufen sein.

Um am Schluss keine Überraschungen zu erleben, ist es wichtig das möglichst alle Einzelkomponenten vor der Gesamtimplementation vollständig ausgetestet sind. Für einen Test des vollständigen Motorrades, ist eine geeignete Umgebung zu finden oder zu erstellen.

In der Arbeit, wird darauf geachtet, dass das Zweirad flexibel bezüglich dem Anbau weiterer Sensoren und Aktoren bleibt. Auch soll es möglich sein jederzeit das Modell wieder in den ursprünglichen Zustand zurückzuversetzen. Im Ausblick werden ein Reihe von Möglichkeiten aufgezählt, um das Modell zu erweitern.

© Copyright 1998 SA

4. Motorrad

Aufgrund der Aufgabenstellung musste zuerst ein Zweirad gefunden werden, welches einen einfachen Umbau zum geregelten Fahrzeug gestattet. Es standen mehrere Varianten zur Diskussion, für welche in diesem Kapitel die Vor- und Nachteile aufgeführt sind. Schliesslich wurde ein Modellbaumotorrad der Firma Graupner gewählt und ausgetestet. Die Funktionsweise des gekauften Modells ist anhand von Videoaufnahmen beschrieben. Der letzte Teil des Kapitels ist dem Umbau des Motorrades gewidmet. Die Teile, welche auf das Motorrad gebaut wurden, sind im folgenden Kapitel beschrieben.

4.1. Elektrobike

Als Grundlage für das geregelte Zweirad wurde abgeklärt, ob ein normales Fahrrad mit Elektroantrieb geeignet ist. Es wurden mehrere Typen überprüft. Als Beispiel ist unten das elektrisch angetriebene Fahrrad der Firma BKTech AG, Burgdorf abgebildet.

Elektrobike Flyer

Vorteile

Der Antrieb und die Spannungsversorgung sind bereits vorhanden. Es gibt keine Platzprobleme für die Reglerplatine.

Nachteile

Es ist zu gross, um im Vorlesungssaal betrieben zu werden. Das Ansteuern der Lenkung benötigt einen grossen Servomotor und damit auch die nötige Leistungselektronik.

Schlussfolgerung

Ein normales Fahrrad ist zu gross für unsere Zwecke. Zudem ist der finanzielle und zeitliche Aufwand erheblich. Es muss ein kleineres Zweirad gefunden werden.

4.2. Kleinfahrrad

Als weitere Variante stand der Kauf eines Kinderfahrrades zur Diskussion. Dabei müssten sämtliche Stell- und Messorgane sowie der Antrieb durch eine Werkstatt montiert werden.

Das kleinste Modell, welches gefunden wurde, hat einen Raddurchmesser von 16“ (ca. 40 cm).

Kinderfahrrad

Vorteile

Gegenüber einem Elektrofahrrad ist es kleiner und wesentlich günstiger. Trotzdem steht genügend Platz zur Verfügung, um die nötigen Bauteile zu befestigen.

Nachteile

Für den Umbau könnte man Modellbauservos verwenden. Es ist aber unsicher, ob deren Leistung ausreicht. Der Antrieb und die Batterie müssten zuerst angebaut werden.

Schlussfolgerung

Ein Kinderfahrrad ist, obwohl kleiner als ein normales Fahrrad, zu unhandlich für den Hörsaal. Auch bei dieser Variante ist der zeitliche und technische Aufwand für den Umbau zu gross.

4.3. Eigenbau eines Zweirades

Als dritte Variante wurde der Bau eines Kleinstfahrrades durch die Werkstatt und der Einbau von Modellbaukomponenten diskutiert.

Das folgende Bild zeigt als Beispiel ein ehemals von Graupner betriebenes Testmodell.

 © Copyright 1998 SA

Fahrrad Graupner

Auch Bruno Paillard, University of Sherbrooke, Kanada hat diese Lösung für seine zwei Prototypen gewählt. Dabei wurde das gesamte Modell aus Aluminium gefräst [Pai].

Vorteile

Man kann standardisierte Komponenten aus dem Modellbau verwenden und ist vollkommen frei im Design.

Nachteile

Das ganze Modellfahrrad muss neu konstruiert werden, da in dieser Grössenordnung keine Baupläne vorhanden sind. Ein bestehendes Fahrradmodell gibt es nicht zu kaufen.

Schlussfolgerung

Diese Lösung ist sehr flexibel. Sie verlangt aber eine intensive Beschäftigung mit dem Design des Zweirades. Mit dieser Lösung lässt sich der Zeitrahmen einer Semesterarbeit nicht einhalten.

4.4. Modellbauzweirad

Bei dieser Variante, welche dann beschafft wurde, wird ein Modellbaumotorrad zugekauft.

Vorteile

Das Modell ist ohne einen Umbau fahrfähig. Der Antrieb und die Lenkung sind bereits integriert und die Spannungsversorgung der Reglerplatine kann direkt aus dem Modellbauakku erfolgen.

Nachteile

Der Platz auf dem Modellbaumotorrad ist beschränkt. Die Platine mit dem Prozessor muss klein genug sein, damit sie die Fahreigenschaften nicht negativ beeinflusst. Sämtliche Signale vom Empfänger zum Servo sind unbekannt und müssen zuerst ausgemessen oder recherchiert werden.

Schlussfolgerung

Da diese Variante im Hinblick auf den zeitlichen Rahmen und den mechanischen Umbau unseren Anforderungen am besten entspricht, wurde diese Lösung gewählt.

4.4.1. Modellmotorradtypen

Es musste ein passendes Motorradmodell gefunden werden. Dabei wurde vor allem auf die Grösse, die Robustheit und die Möglichkeiten zum Umbau geachtet. Im folgenden sind die Resultate beschrieben, welche sich nach den Abklärungen ergeben haben.

Es gibt keine Modellbausätze mehr, sondern nur noch komplett montierte Motorräder mit oder ohne Fernsteuerung, da die Kundennachfrage zu gering ist. Ein Grund für die geringe Nachfrage ist, dass die Modelle schwer zu steuern sind.

Zwei Modelle sind zur Zeit erhältlich. Beide haben einen elektrischen Antrieb und massstabsgetreue Ausmasse. Beide Modelle sind mit einer stufenlosen proportionalen Fernsteuerung ausgestattet und haben montierbare Sturzbügel oder Stützräder. Gemäss der Beschreibung in den Katalogen sind die Beschleunigung und die Geschwindigkeit eher zu gross, um damit im Hörsaal herumzufahren.

Modell Kyosho

Hersteller:	Kyosho
Grössenverhältnis:	1 : 8
Länge:	270 mm
Radstand:	185 mm
Breite:	98 mm
Höhe:	190 mm
Bodenfreiheit:	10 mm
Gewicht:	750 g
Antriebsakku:	7,2 V / 600 mA
Preis:	360.- sFr mit 2-Kanal-Fernsteuerung 160.- sFr ohne Fernsteuerung

Dieses Modell ist mit einer beweglichen Fahrerpuppe ausgestattet. Die Zwei-Kanal-Fernsteuerung ist etwas unbefriedigend, da sie aus einem Pistolengriff und einem Drehknopf besteht und keine Ausbaumöglichkeiten zulässt. Bei einer grösseren Empfängereinheit könnte es passieren, dass sie nicht mehr unter die Abdeckung passt und damit die Verschalung nicht mehr montiert werden könnte, was für unser Vorhaben aber kaum ins Gewicht fallen dürfte.

Modell Graupner

Modellmotorrad Graupner

Vertrieb:	Graupner
Typ:	Graupner 460
Bestell-Nr:	5020
Grössenverhältnisse:	1 : 5

 © Copyright 1998 SA

Länge:	410 mm
Radstand:	290 mm
Breite:	135 mm
Höhe:	310 mm
Durchmesser Hinterrad:	128 mm
Durchmesser Vorderrad:	117 mm
Gewicht:	2,2 kg
Preis:	406.- sFr mit 4-Kanal-Fernsteuerung

Dieses Modell verfügt über eine eingebaute Gewichtsverlagerung, bei welcher der Akku, der ganz unten montiert ist, über einen Servo seitlich verschoben wird, was eine Neigung von über 45° zulasse. Gemäss Katalog besitzt dieses Modell auch eine Bremse, welche etwa in der Art einer Scheibenbremse anspricht. Graupner stellt dieses Modell nicht selbst her, sondern kauft es komplett zusammengebaut ein.

Schlussfolgerung

Das Modell Graupner ist etwas grösser und wesentlich schwerer als das Modell Kyosho. Auf den Bildern im Katalog macht es auch einen robusteren Eindruck. Da der Importeur für die Schweiz in der Nähe von Zürich beheimatet ist, war es möglich, vor dem definitiven Kauf das Modell noch genauer zu überprüfen.

4.5. Bericht zum Besuch bei Graupner

Wir besuchten die Vertretung von Graupner am Freitag den 19. September 1997. Folgende Erkenntnisse konnten aus dem Besuch gezogen werden:

Steuerung

Das Vorderrad des Modells wird nicht wie bei einem normalen Fahrrad gesteuert, sondern es wird zusammen mit der Batterie seitlich ausgelenkt, was einer Gewichtsverlagerung entspricht. Das Vorderrad ist frei drehbar, so dass das Fahren mit dem Modellmotorrad einem freihändigen Fahren entspricht. Dies hat zur Folge, dass das Modell recht schnell fahren muss, um eine gewisse Stabilität zu erhalten, und dementsprechend wird der Kurvenradius grösser.

Modell

Es besteht vollständig aus Kunststoff. Nur Motoren, Schrauben und die Gabel des Vorderrades sind aus Metall. Dadurch ist das Modell nicht vollständig starr und erträgt Schläge und Stürze gut. Die einzigen Verschleissteile seien gemäss Auskunft von Herrn Lutz (Graupner Schweiz) die Verschalung und eventuell die Gabel des Vorderrades, wenn man direkt in eine Wand fährt. Wenige Teile sind fix verklebt. Die meisten sind zusammengeschraubt. Dabei sind die Schrauben meistens in den Plastik hineingedreht und nicht mit Muttern befestigt.

Standardteile

Für die Empfängereinheit und die Ansteuerung des Motors wurde eine einzige, offene und auf dieses Modell zugeschnittene Printplatte verwendet. Sämtliche Teile des Modells können einzeln bestellt werden.

Lager

Sämtliche Lager, das heisst die Radlager und die Lenkungslager, sind Gleitlager.

Kraftübertragung

Der Motor ist fest mit dem Gehäuse verschraubt. Über einen Zahnriemen wird die Hinterradachse angetrieben.

Lieferzeiten

Die maximale Lieferzeit beträgt bei einer Bestellung bei Graupner Schweiz ca. 10 Tage und über den Fachhandel maximal 3 Wochen. Wenn das Modell verfügbar ist, wird es sofort zugesandt.

Umbau

Das Modell lässt folgende Umbauten zu:

- Die Printplatte kann durch separat eingekaufte Standartteile ersetzen werden (Empfängereinheit, Fahrtregler etc.)
- Die direkte Lenkung des Vorderrades ist durch Verwendung des "Gewichtverlagerungs"-Servos oder durch einen zusätzlichen Servo möglich.
- Es ist genügend Platz für Sensoren, Aktoren und eine kleine Reglerplatine vorhanden und es ist möglich, diese durch Schrauben oder Kleben zu befestigen
- Die Räder sind mit Schrauben und Muttern am Rahmen befestigt. Ein Geschwindigkeitssensor kann montiert werden.

Probleme

Als grösstes Problem wird die Auslenkung des Vorderrades erachtet. Sie beträgt maximal ±18 Grad, was einem Kurvenradius von mehr als einem Meter entspricht. Als weiteres Problem muss die Befestigung eines Servos an der Lenkung gelöst werden.

Fazit

Das Modell entspricht nicht den Erwartungen in Bezug auf Stabilität, Materialverwendung und Lenkung. Dem Kinderfahrrad als Alternative ist aber dieses Modell vorzuziehen, da es robust ist und viele Teile bereits korrekt dimensioniert vorhanden sind. Natürlich ist auch in diesem Fall ein Umbau für unsere Zwecke unumgänglich.

4.6. Funktionsweise des gesteuerten Modells

Als Grundkonstruktion für das Zweirad wird das Motorradmodell der Firma Graupner verwendet. Dieses Motorrad ist über eine Fernsteuerung sowohl in der Geschwindigkeit als auch in der Richtung steuerbar. Bei genügender Geschwindigkeit stabilisiert es sich selbst durch seine Geometrie und das frei drehbare Vorderrad.

Die Gabel zieht das Vorderrad hinter sich her. Die Gabel hat einen Nachlauf, das heisst, dass die Achse des Gabelrohres vor der Achse des Vorderrades verläuft. Das Verhalten ist vergleichbar mit den Rollen an einem Stuhl. Die Rollen richten sich in die Fahrtrichtung aus. Im Gegensatz dazu hat ein vom Menschen gesteuertes Zweirad normalerweise einen Vorlauf. Weshalb man dabei aber trotzdem freihändig fahren kann wird im Kapitel 6.5 (Abschätzung des Modellierungsfehlers) erklärt. Beim Modellmotorrad stabilisiert sich das Vorderrad so, dass der Schwerpunkt in einer stabilen Lage zwischen Gravitation und Fliehkraft liegt. Die Kräfte, welche durch die Änderung des Dralls der Räder wirken, dämpfen die Bewegungen des Motorrades. Das heisst grundsätzlich, dass es sich beim gesteuerten Motorradmodell um ein ab einer gewissen Geschwindigkeit stabiles System handelt.

Um zu zeigen, wie sich das Modell im ursprünglichen Zustand verhält, wurden Videoaufnahmen gemacht. Bei diesen Testfahrten zeigte das Modell ein erstaunlich gutes Verhalten. Die Aufnahmen zeigen das Motorrad sowohl auf einem glatten als auch einem rauhen Boden, also bei unterschiedlicher Haftung. Bei ausreichender Geschwindigkeit fuhr das Modell sowohl geradeaus als auch in Kurven stabil. Wurde jedoch die Geschwindigkeit verlangsamt, nahm die Stabilität ab, und unterhalb von ca. 2 m/s kippte es um.

Aufnahme in der Turnhalle des MM-Gebäudes

Die erste Aufnahme ist in der Turnhalle des MM-Gebäudes unter dem ETH Hauptgebäude entstanden. Da der Turnhallenboden sehr glatt ist, rutscht das Motorrad bei starker Schräglage schnell weg und kommt zu Fall. Alle Versuche mussten mit Stützrädern ausgeführt werden, damit das Motorrad nicht ständig umfällt.

Aufnahme im Freien auf der Tartanbahn beim Ballon

Aufgrund des glatten Bodens in der Turnhalle wurden die zweiten Aufnahmen auf einem Hartplatz im Freien durchgeführt. Die Haftung auf dem Boden war wesentlich besser. Es konnte auf die Stützräder verzichtet werden.

 © Copyright 1998 SA

Versuche

An beiden Standorten wurden folgende Versuche aufgenommen:

Modellmotorrad in der ursprünglichen Ausführung

Das Motorrad fährt ohne jegliche Änderung umher, so wie es verkauft wurde. Dabei zeigt sich ein gutmütiges Verhalten. Bei genügender Geschwindigkeit und genügend Platz ist das Motorrad einfach zu steuern. Mit etwas Übung kann eine grosse Acht gefahren werden. Lässt man den Steuerknüppel nach einer Schrägfahrt zurück in die Nullage springen, ergibt sich ein starkes Überschwingen. Es zeigte sich, dass das Motorrad sehr robust gegen Unebenheiten ist.

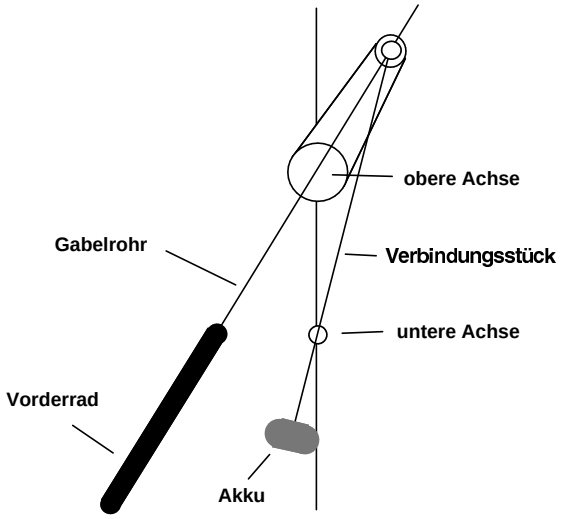
Modellmotorrad ohne Verschalung und Puppe

Durch die Demontage der Verschalung und der Puppe liegt der Schwerpunkt weiter unten, und das Modell hat weniger Gewicht. Bei stabiler Fahrt wird das Motorrad träger, weil es weniger Gewicht hat. Das Auslenken des Gabelrohrs wirkt sich nicht mehr so stark auf das Verhalten des Motorrades aus. Auch bei starker Auslenkung legt sich das Motorrad kaum in die Kurve.

An der Stabilitätsgrenze kippt das Modell jedoch schneller um, da der Schwerpunkt tiefer liegt. Es kippt schneller in die Kurve und kommt darum auch schneller zu Fall.

Modellmotorrad mit versteiftem Gabelrohr (Lenkung nur noch mit Gewichtsverlagerung)

Wird der Lenkmechanismus durch das drehbare Gabelrohr versteift und nur noch die Gewichtsverlagerung zur Lenkung verwendet, wird das Steuern äusserst schwierig. Bei hohen Geschwindigkeiten (im Stabilitätsbereich) legt sich das Motorrad auch bei voll ausgelenktem Gewicht kaum in die Kurve, und es ergeben sich riesige Kurvenradien. Damit auch engere Kurven gefahren werden können, muss das Motorrad bis an die Stabilitätsgrenze verlangsamt werden. Enge Kurven können dann nur noch mit der Geschwindigkeitssteuerung gefahren werden. Die Gewichtsverlagerung bewirkt nur, dass es eine Links- oder eine Rechtskurve gibt.



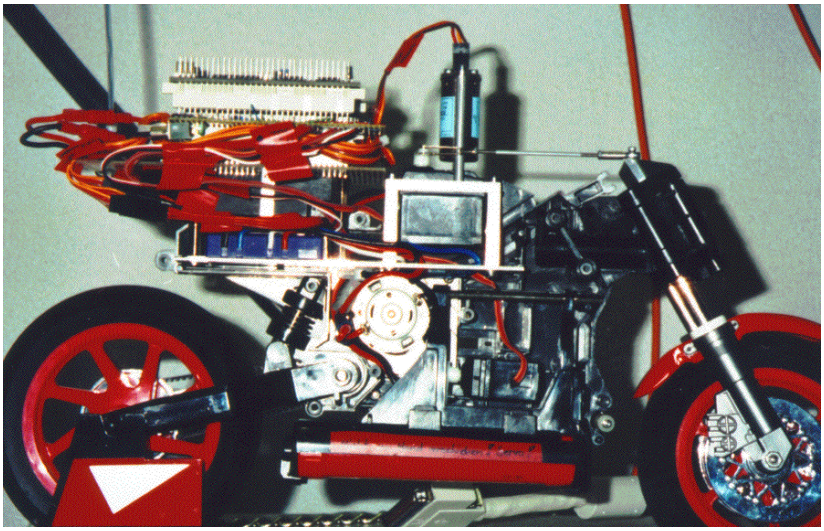
Lenkmechanismus des Gabelrohrs

Schlussfolgerung

Das Motorrad wird vor allem durch die seitliche Auslenkung der Gabel gesteuert. Wird die Gabel nach rechts ausgeschwenkt, legt sich das Motorrad in eine stabile Rechtskurve. Es ist erstaunlich, dass sich das Motorrad in derselben Richtung in die Kurve legt, wie die Gabel ausgelenkt wird. Die Gewichtsverlagerung unterstützt diese Bewegung nur. Das Motorrad ist durch seine Konstruktionsart mit dem frei drehbaren Vorderrad ab etwa 5 km/h stabil und wird durch das Erhöhen der Geschwindigkeit noch zusätzlich stabilisiert.

4.7. Umbau des Motorrades

Zuerst wird der Originalzustand, soweit noch nicht erklärt, beschrieben. Danach werden die gemachten Umbauten erklärt. Das folgende Photo zeigt das Innere des Modellmotorrades nach dem Umbau.



Motorrad offen, nach Umbau

4.7.1. Originalzustand

Elektonik

Wie bereits beim Besuch festgestellt, befindet sich die gesamte Elektronik des Modells auf einer Platine, welche ins Innere des Chassis eingeschoben werden kann. Auf dieser Platine befinden sich die Empfängereinheit, der Fahrtregler zur Ansteuerung des Motors und die Regelung des Servos zur Gewichtsverlagerung.

Da die Modellbaufirmen aus Gründen des Kopierschutzes grundsätzlich keine Schaltpläne herausgeben, ist es unmöglich, die benötigten Signale von dieser Platine abzugreifen. Die Elektronik zur Regelung des Servos ist ebenfalls auf dieser Platine untergebracht. Zum Servo führen darum fünf Kabel. Bei Standardservos werden hingegen nur drei Kabel (Speisung, Erde, Impuls) verwendet.

Deshalb wird die gesamte Elektronik durch Standardkomponenten ersetzt. Der Servo muss durch einen handelsüblichen Modellbauservo ersetzt werden.

Die mitgelieferte Sendeeinheit besitzt nur zwei Kanäle und ist nicht ausbaufähig. Da vorgesehen ist, auf dem Modell eventuell auch einen MIMO-Regler zu implementieren, kann diese Fernsteuerung nicht verwendet werden. Es muss mehr als nur die Geschwindigkeit und Richtung gesteuert werden können. Es ist nötig, dass auch die Gewichtsverlagerung angesteuert werden kann.

Steuerung

Die Steuerung erfolgt über einen Servo, welcher oberhalb der Batterie im Innern des Gehäuses eingeklemmt ist. Er ist weder geschraubt noch geklebt, sondern nur sauber eingeklemmt. Er lässt sich jederzeit einfach ausbauen. Dieser Servo lenkt das Batteriefach, welches sich unten am Motorrad befindet, seitlich aus. Über eine starre Verbindung wird auch das Gabelrohr ausgelenkt. Zwei Federn an der Seite des Gabelrohrs dämpfen diese Bewegung. Das Drehen des Vorderrads selbst, wie es von einem normalen Fahrrad bekannt ist, wird nicht angesteuert. Das Vorderrad lässt sich bis an die Anschläge frei drehen.

Verkleidung

Zum Originalmodell gehört auch eine massstabsgetreue Verkleidung, welche sich aufstecken und festschrauben lässt. Die Verkleidung besteht aus zwei Teilen. Das Teil, welches das Motorrad

 © Copyright 1998 SA

unten und vorne schützt, ist einfach zu montieren. Das Teil mit der Fahrerpuppe, wird von oben schräg eingeschoben. Mit etwas Krafteinsatz kann das Teil in die richtige Position geschoben werden, damit man es festschrauben kann. Die Puppe ist mit einigen Schrauben an die Verkleidung montiert.

4.7.2. Umbau

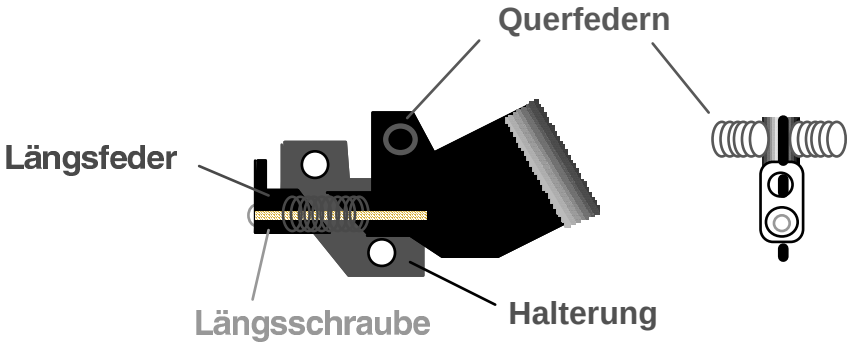
Antrieb

Der Antrieb wurde belassen. Der Motor, das Getriebe und die Übertragung auf das Hinterrad haben keine Änderungen erfahren.

Versteifung der Gabel

Im Originalzustand kann die Gabel seitlich ausgelenkt werden. Da dieses Auslenken eine vereinfachte Modellierung als Einkörpersystem unmöglich macht und die Auslenkung nur über die Gewichtsverlagerung erfolgen kann, wird die Gabel versteift.

Wie das Schema unten zeigt, sind im Aufhängemechanismus der Gabel ursprünglich drei Federn eingesetzt, welche durch entsprechende Aluminiumröhrchen ersetzt wurden. Mit diesen Röhrchen ist die Gabel vorne völlig starr. Es ist zu beachten, dass auch die Längsfeder im Gabelmechanismus ersetzt wurde. Diese Feder liess eine leichte Verschiebung des Gabelrohres in Fahrtrichtung zu. Ohne diese Feder wird die Gabelaufhängung bei einer Frontalkollision stärker belastet.



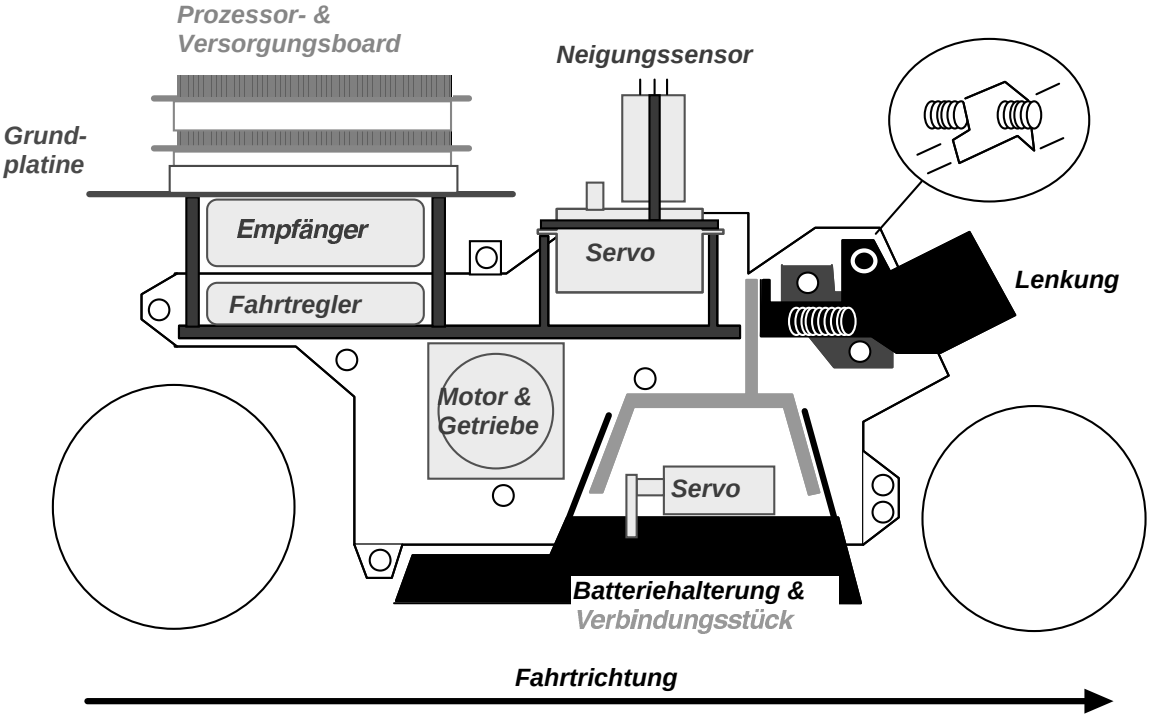
Aufhängemechanismus Gabel

Elektronik

Die bestehende Elektronik inklusive dem Servo wurde entfernt und durch neue Komponenten ersetzt. Die Evaluierung wurde mit grosszügiger Mithilfe von Herrn Meili, Meili Modellbau, Frauenfeld, durchgeführt.

Die Platine des Originalmotorrades wurde durch eine Aluminiumplatte ersetzt, welche als Grundplatte für den weiteren Aufbau verwendet wurde. Der Fahrtregler wurde direkt auf diese Grundplatte montiert. Über zwei Distanzstücke wurde eine Basisplatine montiert, welche als Basis für das Prozessor- und das Versorgungsboard dient. Der Empfänger wurde unter diese Basisplatine montiert.

Im folgenden Schema sind die Positionierung der Servos und Elektronikteile im Modellmotorrad eingezeichnet.



Detailschema Umbau Motorrad

Gewichtsverlagerungsservo

Da der Servo C512 die gleichen Masse wie der Originalservo hat, kann die bestehende Halterung für den neuen Servo verwendet werden. Der Gewichtsverlagerungsservo wird also nicht verschraubt, sondern lediglich ins Gehäuse eingelegt. Fixiert wird er vom Gehäuse und den Gehäuseschrauben.

Lenkservo und die mechanische Verbindung zur Gabel

Der Servo für die Lenkeransteuerung wurde mit zwei zugefrästen Aluminiumstücken auf die Grundplatte gesetzt, damit er in der richtigen Höhe aus dem Chassis herausragt und einfach zur Ansteuerung der Gabel verwendet werden kann. Für die mechanische Verbindung zur Gabel bzw. zum Gewicht werden Servoflügel auf die Servos gesteckt und mit Schrauben fixiert. Da die Flügel und der Servokopf wie Zahnräder ineinander greifen, kann die Position der Flügel nicht beliebig genau eingestellt werden. Die genaue Positionierung muss mittels der Trimmung an der Fernsteuerung vorgenommen werden.

Für die Ansteuerung des Vorderrades mit dem hinter der Gabel sitzenden Servo wurde ein Aluminiumstück auf die Gabel geschraubt, worauf eine Verbindungstange fixiert wurde. Als Verbindungsstange wurde ein Gabelanschluss mit Federstahlbacken verwendet.

Der Servo steht horizontal und lenkt die Verbindungsstange horizontal aus. Da die Gabel aber nicht senkrecht steht, sondern um 30° aus der Senkrechten nach vorne zeigt, bewegt sich die Verbindungsstange beim Fixierungspunkt an der Gabel leicht auf und ab. Deswegen wurde ein Kugelgelenkträger mit Gewinde an der Gabel befestigt, damit dieser Bewegung kein Widerstand entgegengesetzt wird. Die Länge der Verbindungsstange kann durch Gewinde an beiden Enden entsprechend angepasst und mit Muttern fixiert werden.

Da der Servo eine Geschwindigkeitsbeschränkung hat und die Berechnung einfacher werden, wurden die Hebelarme am Servo und der Gabel gleichlang gewählt. Es ergibt sich dadurch eine 1:1-Übertragung des Winkels vom Servo auf die Gabel. Der Servo wird nicht bis an seine maximale Auslenkung von ±45° betrieben, sondern bewegt sich nur im Bereich ±18°. Als weiterer Vorteil werden so nichtlineare Effekte weitgehend verhindert. Die Ansteuerung der Gabel darf nur über eine Verbindungsstange erfolgen, da sich zwei Verbindungsstangen gegenseitig behindern.

 © Copyright 1998 SA

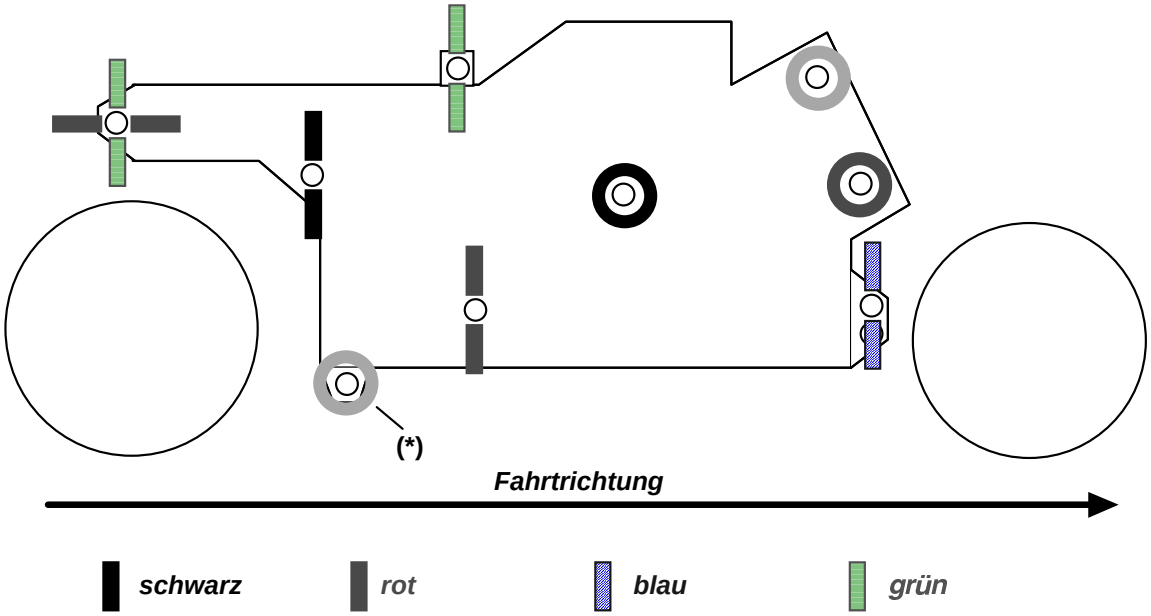
Überrollbügel

Damit der ganze Aufbau mit dem Prozessor- und Versorgungsboard bei einem Sturz nicht beschädigt wird, wurde ein Überrollbügel montiert. Dieser Überrollbügel ist so geformt, dass er in die Halterung der ursprünglichen Stützräder passt. Die vorderen Löcher der Halterung mussten etwas vergössert werden, damit der Überrollbügel eingeführt werden kann. Somit können die ursprünglichen Stützräder nur noch lose montiert werden.

4.7.3. Tips und Tricks zum Umbau

Das Steckersystem von Graupner ist verpolsicher. Die Ecken der Stecker sind auf der einen Seite abgerundet. Somit können die Stecker an den Modellbauteilen nicht falsch eingesteckt werden. Dies trifft aber nicht auf das Versorgungsboard und den Lagesensor zu.

Die Schrauben, welche das Modell zusammenhalten, sind sehr weich. Es ist wichtig, dass ein genügend grosser Kreuzschraubenzieher verwendet wird. Die meisten Schrauben sind in den Plastik gedreht, weshalb sie leicht ausgedreht werden können. Damit die Schrauben am richtigen Platz montiert werden, wurden die Löcher und die Schrauben farblich gekennzeichnet. Eine Schraube, welche mit (*) gekennzeichnet ist, wird von der anderen Seite hineingedreht. Im folgenden Schema sind die entsprechenden Farben zu den Löchern eingezeichnet.



Schraubenschema

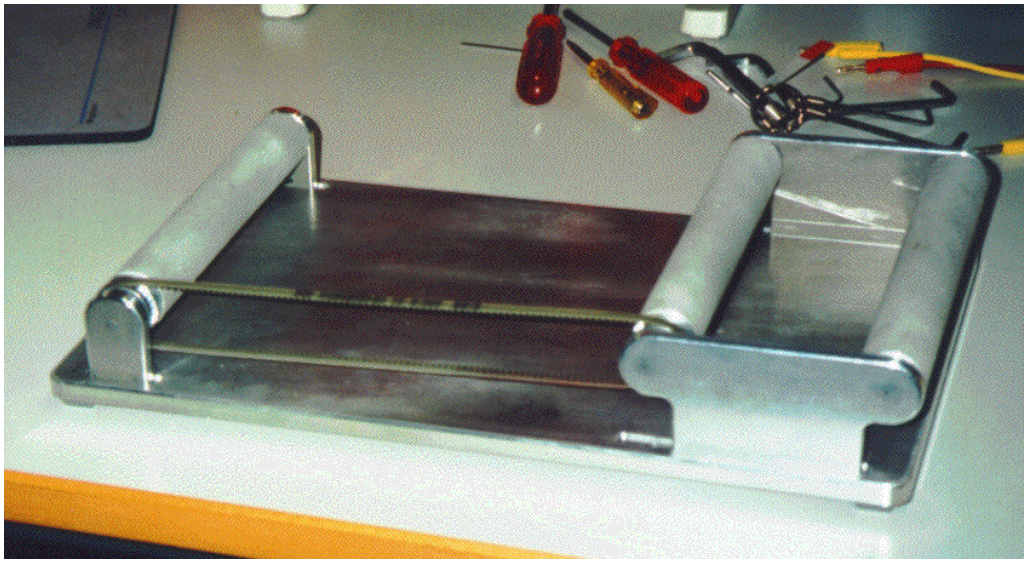
Die Herstellung der hinzugefügten Metallteile wurde durch die Elektrotechnik-Werkstatt im ETL durchgeführt. Der Betreuer für dieses Projekt war Herr M. Bühler, der wertvolle Tips und Anregungen für den Umbau geben konnte.

Es ist stets darauf zu achten, dass zuerst die Fernsteuerung und erst dann der Empfänger eingeschaltet wird. Ist nur der Empfänger eingeschaltet, werden irgendwelche Signale empfangen, welche ausserhalb der normalen Ansteuerung liegen. Der Antriebsmotor und die Servos führen in diesem Fall nicht kontrollierbare Bewegungen durch. Die Servos werden durch diese Signale an die Anschläge getrieben, was die Mechanik oder die Servos selbst zerstören kann. Da die Signale über den Prozessor geführt werden, konnte eine Routine eingebaut werden, welche dieses Verhalten verhindert.

4.8. Prüfstand

Um das Verhalten des Motorrades bei der Geradeausfahrt testen zu können, wurde ein Prüfstand gebaut. Dieser Prüfstand besteht aus einer Grundplatte und drei etwa 30 cm breiten Rollen. Die Stütze für das Vorderrad besteht aus einer Rolle und das Hinterrad kommt auf den zwei anderen Rollen zu liegen. Die zwei hinteren Rollen haben einen Abstand von 7 cm, so dass das Motorrad auf den Rollen bleibt und nicht etwa nach vorne ausbricht.

Damit die Rollen das Geradeausfahren simulieren können, wird die vordere Rolle mit einem Zahnriemen von der mittleren Rolle angetrieben. So dreht sich auch das Vorderrad. Steht das Vorderrad schräg zum Motorrad, so ergibt das Abrollen des Vorderrades auf der Rolle eine seitliche Bewegung, welche einer Kurve auf dem Boden entsprechen würde. Mit dieser Auslenkung lässt sich das Motorrad auf den Rollen stabilisieren, so dass es eine Geradeausfahrt vollzieht.



Prüfstand

 © Copyright 1998 SA

5. Sensoren / Aktoren / Modellbauteile

Dieses Kapitel beschreibt die verschiedenen Sensoren, Aktoren und Modellbauteile, welche im Verlauf der Semesterarbeit beschafft wurden. Zu jedem Teil ist beschrieben, warum es gekauft wurde und wie es funktioniert. Es ist vorteilhaft, wenn die einzelnen Teile, die beschrieben werden, dem Leser vorliegen.

5.1. Lagesensor

Um das seitliche Kippen des Motorrades zu messen, musste ein passender Sensor gefunden werden. Das dynamische Messen des Neigungswinkels α ist ein schwieriges physikalisches Problem. Der Winkel kann nicht direkt gemessen werden, da keine direkte Verbindung zum Boden besteht.

Am Naheliegensten ist die Verwendung eines Gyrometers, welcher die Drehgeschwindigkeit misst. Auch Bruno Paillard, University of Sherbrooke, Kanada hat diese Lösung für seine zwei Prototypen gewählt [Pai]. Leider führt die Drift bei Gyrometern zur langsamen Änderung der Nullage, welche kompensiert werden müsste.

Als anderes Messglied ist es möglich, einen Inklinometer einsetzen. Das Inklinometer misst die Auslenkung eines Pendels, welches in die Richtung der resultierenden Kraft zeigt. Diese Sensoren haben das Problem, dass sie wegen der Messart eine grosse Verzögerung besitzen.

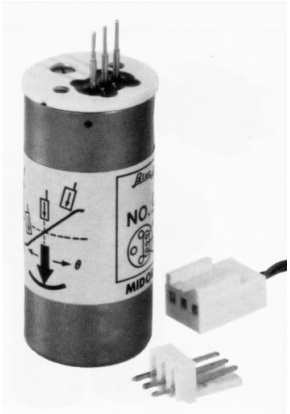
Da sich die Beschaffung eines Gyrometers als schwierig und teuer erwies, ist momentan im Motorrad ein Inklinometer eingebaut. Wegen der Zeitverzögerung des Inklinometers wird eventuell die Beschafftung eines Gyrometers nötig. Die Kombination von Inklino- und Gyrometer könnte die Nachteile der beiden Messysteme gegenseitig aufheben. Folgende Informationen über Lagesensoren sind während der Semesterarbeit zusammengekommen:

5.1.1. Gyrometer

Im Internet (http://www.gyration.com) wurde ein kleiner und direkt auf ein Board lötbarer Gyrometer gefunden. Der MicroGyro 100 von Gyration Inc., Saratoga, USA, hat aber den Nachteil, dass zuerst ein Evaluationboard beschafft werden muss. Dieses Board kostet Fr. 650.– und war zu Beginn der Semesterarbeit noch nicht lieferbar.

Da sich herausstellte, dass der Inklinometer eine Einschwingzeit von 150 ms hat, wurden bereits weitere Abklärungen gemacht. Dabei wurde herausgefunden, dass im Projekt „Schwebende Plattform“ am Institut für Automatik Gyrometer eingesetzt werden. Auf der Plattform wird ein piezoelektrischer Gyrometer Gyrostar ENC-05EA der Firma Murata verwendet. Dieser Gyro ist preiswert und wird in der Schweiz durch die Firma AGELTRO, Mönchaltdorf vertrieben. Es scheinen keine Lieferschwierigkeiten zu bestehen. Falls zur Regelung zusätzlich ein Gyrometer nötig wird, ist dieser Gyrometer sicher empfehlenswert, da im Projekt „Schwebende Plattform“ zufriedenstellende Ergebnisse erreicht wurden.

5.1.2. Inklinometer



Inklinometer

Als Lagesensor wurde der Inklinometer PMP-S20T der Firma Pewatron, Wallisellen verwendet [Pew]. Das Inklinometer besitzt einen analogen Ausgang, welcher zuerst gewandelt werden muss, um vom Prozessor verarbeitet werden zu können. In vertikaler Position ist der Wert des Ausgangs die halbe Versorgungsspannung. Je nach Richtung der Auslenkung nimmt die Ausgangsspannung zu beziehungsweise ab. Im Anhang befindet sich das Datenblatt dieses Inklinometers. Mit seinem Preis von Fr. 150. – ist er die günstigste Möglichkeit, um die Winkellage des Motorrades zu messen.

Das dynamische Verhalten des Inklinometer wurde durch die Sprungantwort identifiziert. Die Messung der Sprungantwort ist im Kapitel 6.2 (Modellierung des Inklinometers) beschrieben. Falls das dynamische Verhalten zu langsam ist, steht die Möglichkeit eines Gyrometers oder der folgende Inklinometer zur Auswahl.

Eine wesentlich teurere Variante stand ebenfalls zur Diskussion. Der Zerotronic Sensor der Wyler AG, Winterthur hat ein schnelleres Ansprechverhalten als der Inklinometer von Pewatron und ein digitales Ausgangssignal. Dieser Sensor ist mit mehr als Fr. 2300.– jedoch viel teurer. Es wurde der günstigere Inklinometer gekauft.

5.2. Geschwindigkeitsmessglied

In dieser Semesterarbeit wurde angenommen, dass sich das Modell mit konstanter Geschwindigkeit bewegt und weder beschleunigt, noch bremst. Zuerst war vorgesehen, dass das Sollwertsignal von der Fernsteuerung beziehungsweise dem Empfänger als Geschwindigkeitsangabe verwendet werden kann. Die Beziehung zwischen dem Sollwertsignal und der resultierenden Geschwindigkeit ist stark nichtlinear und abhängig von der Batterieleistung und der Belastung des Motorrades. Auf dem Prüfstand braucht das Motorrad wesentlich mehr Leistung als auf dem Boden. Es war lediglich möglich, die Einstellungen für die Sollgeschwindigkeit von 3 m/s zu ermitteln. Diese Einstellung liegt circa bei Rasterpunkt 16 des Knüppels 2 Richtung D an der Fernsteuerung (siehe Kapitel 5.4.3 Fernsteuerung).

Aufgrund dieser Schwierigkeiten wurde klar, dass der Einsatz eines Geschwindigkeitsmessgliedes bereits für konstante Geschwindigkeiten sinnvoll ist. In einem späteren Schritt sollten auch Geschwindigkeitsänderungen in freier Fahrt möglich sein, was auf jeden Fall eine Geschwindigkeitsmessung nötig macht.

5.2.1. Evaluierung eines Geschwindigkeitssensors

Eine berührungslose Messung mit einem optischen Sensor oder einem Hallsensor sind Lösungen für die Geschwindigkeitsmessung. Dafür wurden verschiedene Firmen angefragt und Informationsmaterial angefordert. Die Adressen der angefragten Firmen sind im Anhang aufgeführt.

<u>Lichtschranken mit Codescheiben</u>

Die Umlaufgeschwindigkeit eines Rades kann mit einer Codescheibe und einer Lichtschranke gemessen werden. Die Codescheibe unterbricht in regelmässigen Abständen den Lichtstrahl der

 © Copyright 1998 SA

Lichtschranke, was den Strom einer Photodiode verändert. Dieses Signal kann als digitales Signal vom Prozessor eingelessen und gezählt werden.

Aus dem erhaltenen Informationsmaterial ergab sich, dass die Firma Hewlett Packard die grösste Auswahl an Sensoren hat, welche in Bezug auf Grösse, Preis und Leistung den Anforderungen entsprechen. Viele Varianten waren entweder zu gross, zu teuer oder hatten eine zu geringe Auflösung des Drehwinkels, das heisst zuwenig Schlitze pro Umdrehung.

Ein Sensor, welcher die Anforderungen am besten erfüllt, ist die Lichtschranke HEDS-9700 mit dem Codewheel HEDS-5120 der Firma Hewlett Packard, welche über die Firma Elbatex AG, Wettingen bezogen werden können.

Gegen die Montage einer solchen Lichtschranke am Motorrad sprechen einige Gründe. Die Codescheibe muss auf das Hinterrad des Töffs montiert werden. Es gibt aber keine Codescheiben mit einem Innendurchmesser von 12mm. Somit müsste entweder die Achse des Hinterrades abgedreht werden oder aber der Innendurchmesser der Coderscheibe vergrössert werden.

Das Hinterrad läuft auf einer starren Metallachse von 8mm Durchmesser. Als Lager werden zwei Kupferstücke verwendet, welche in die Plastikachse des Rades eingeschoben sind. Die Wandstärke der Plastikachse ist lediglich einige Milimeter stark. Das Abdrehen dieser Achse kommt deshalb nicht in Frage, weil das zu einem Bruch der Achse führen kann.

Man kann jedoch auch nicht den Innendurchmesser einer Codescheibe genügend vergrössern, um die Codescheibe auf die Achse zu stülpen. Zudem ist es schwierig, die Codescheibe genau zu montieren. Eine Unwucht kann zu ungenauen Messresultaten oder bei einer Berührung der Lichtschranke sogar zur Zerstörung führen.

Winkelschrittgeber

Eine andere Möglichkeit der Geschwindigkeitsmessung sind Winkelschrittgeber, auch Encoder genannt. Sie bestehen ebenfalls aus einem optischen Sensor und einer Codescheibe, sind aber in einem Gehäuse verpackt. Sie haben eine Achse oder eine Hohlwelle als Verbindung zu einem Motor oder einem Rad.

Beim Motorrad dreht sich die Hinterradachse selbst nicht. Es ist nicht möglich, einen Encoder mit einer Hohlwelle direkt an die Achse zu montieren. Es müssen zwei Zahnräder zur Übertragung verwendet werden. Ein Zahnrad wird am Hinterrad und das andere an der Achse des Encoders befestigt.

Das Moment, welches aufgewendet werden muss, um den Winkelschrittgeber anzutreiben, ist klein gegenüber den anderen Drehmomenten. Das Moment stellt also kein Problem dar. Die Verwendung eines Encoders lässt Spielraum für die Montage offen, da die Grösse der Zahnräder unabhängig vom Sensor gewählt werden kann.

Als Bezugsquelle für einen Winkelschrittgeber bietet sich ebenfalls Hewlett Packard an. Die meisten Sensoren anderer Firmen sind mit stabilen Gehäusen und abgeschirmten Kabeln versehen, welche eher für Industrieandwendungen gedacht sind. Sie sind darum auch wesentlich teurer als die Winkelschrittgeber von Hewlett Packard. Die Sensoren benötigen lediglich einige Widerstände als externe Beschaltung, damit die Signale abgegriffen werden können. Die Datenblätter von HP sind ausführlich, was beim Anschliessen ans Prozessorboard vorteilhaft ist.

Abschätzung der benötigten Auflösung

Der Prozessor kann im gewählten Arbeitspunkt Impulse der Länge 1.9 µs bis 125 ms verarbeiten. Diese Grenzen ergeben auch die maximale und minimale Anzahl Codierungsstriche auf der Codescheibe bezüglich der Geschwindigkeit des Motorrades. Die folgende Tabelle zeigt, dass hohe Auflösungen bis zu 512 cpr (cycles per revolution) für Geschwindigkeiten zwischen 0.03 m/s und 5.5 m/s verwendet werden können.

Impulsbreite, welche vom Prozessor noch verarbeitet werden kann									
TminP (ms)		TmaxP (ms)							
0.0019		125.0000							
Geschwingikeit									
km/h	m/s		d=Durchmesser Hinterrad						
20.0000	5.5556	Vmax2	(m)	PI					
15.0000	4.1667	Vmax1	0.1280	3.1416					
10.8000	3.0000	VnA							
10.0000	2.7778	Vn							
5.0000	1.3889	Vsoll3	Umfang des Hinterrades (U=2*Pi*r=Pi*d)						
3.6000	1.0000	Vsoll2a	(m)						
3.0000	0.8333	Vsoll2	0.4021						
1.8000	0.5000	Vsoll1a							
1.0000	0.2778	Vsoll1							
0.5000	0.1389	Vmin1							
0.1000	0.0278	Vmin2							
Drehzahl =v/U=f (Hz)			Periode T=1/f (s)						
	Hz			(s)		(ms)			
Vmax2/U	13.8155	fmax2		0.0724	Tmax2	72			
Vmax1/U	10.3616	fmax1		0.0965	Tmax1	97			
VnA/U	7.4604	fnA		0.1340	TnA	134			
Vn/U	6.9078	fn		0.1448	Tn	145			
Vsoll3/U	3.4539	fsoll3		0.2895	Tsoll3	290			
Vsoll2a/U	2.4868	fsoll2a		0.4021	Tsoll2a	402			
Vsoll2/U	2.0723	fsoll2		0.4825	Tsoll2	483			
Vsoll1a/U	1.2434	fsoll1a		0.8042	Tsoll1a	804			
Vsoll1/U	0.6908	fsoll1		1.4476	Tsoll1	1,448			
Vmin1/U	0.3454	fmin1		2.8953	Tmin1	2,895			
Vmin2/U	0.0691	fmin2		14.4765	Tmin2	14,476			
Timp=T/cpr			(ms)	[Es werden die steigenden Flanken vom Prozessor erkannt]					
clocks per round									
	K	C	D	E	F	G	H	A	I
	96	100	192	200	256	360	400	500	512
	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)
Tmax2	0.7540	0.7238	0.3770	0.3619	0.2827	0.2011	0.1810	0.1448	0.1414
Tmax1	1.0053	0.9651	0.5027	0.4825	0.3770	0.2681	0.2413	0.1930	0.1885
TnA	1.3963	1.3404	0.6981	0.6702	0.5236	0.3723	0.3351	0.2681	0.2618
Tn	1.5080	1.4476	0.7540	0.7238	0.5655	0.4021	0.3619	0.2895	0.2827
Tsoll3	3.0159	2.8953	1.5080	1.4476	1.1310	0.8042	0.7238	0.5791	0.5655
Tsoll2a	4.1888	4.0212	2.0944	2.0106	1.5708	1.1170	1.0053	0.8042	0.7854
Tsoll2	5.0265	4.8255	2.5133	2.4127	1.8850	1.3404	1.2064	0.9651	0.9425
Tsoll1a	8.3776	8.0425	4.1888	4.0212	3.1416	2.2340	2.0106	1.6085	1.5708
Tsoll1	15.0796	14.4765	7.5398	7.2382	5.6549	4.0212	3.6191	2.8953	2.8274
Tmin1	30.1593	28.9529	15.0796	14.4765	11.3097	8.0425	7.2382	5.7906	5.6549
Tmin2	150.7964	144.7646	75.3982	72.3823	56.5487	40.2124	36.1911	28.9529	28.2743

Sollte eine höhere Auflösung für niedrigere Geschwindigkeiten benötigt werden, so kann ein Übersetzungsverhältnis der Zahnräder von 1 : 2 gewählt werden.

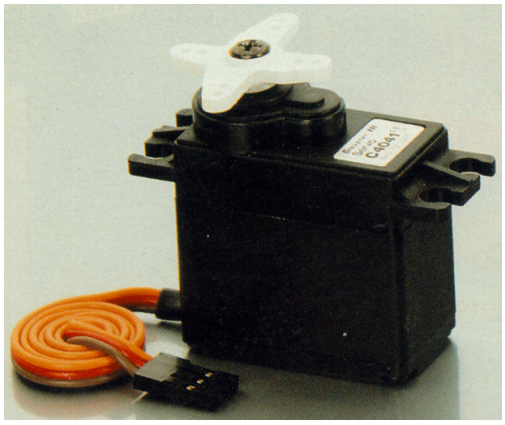
Es wurde der Sensor HEDS-5700-I00 von Hewlett Packard bei der Firma Elbatex AG, Wettingen bestellt. Die Lieferzeit beträgt etwa 6 Wochen. Der Geschwindigkeitssensor wird erst in der nächsten Arbeit mit dem Motorrad zur Verfügung stehen.

 © Copyright 1998 SA

5.3. Servo

Um den originalen Gewichtsverlagerungsservo zu ersetzen und die Lenkung anzusteuern, mussten zwei Servos beschafft werden.

Hersteller	Graupner
Typ	ECO-Servo C512
Bestellnummer	4096
Preis	Fr. 28.00
Versorgungs- und Signalspannung	4.8 - 6 V
Geschwindigkeitsbeschränkung	0.15 s/40°
Stellmoment	27.5 Ncm
Nennfrequenz des PWM-Signals	50 Hz
maximale Frequenz des PWM-Signals	150 Hz



Servo

Zur Fernsteuerung wurde bereits ein Servo vom Typ C512 mitgeliefert. Da dieser Servo äusserst preiswert ist und bereits Erfahrung mit ihm bestehen, wurde entschieden, sowohl für die Lenkeransteuerung als auch für die Gewichtsverlagerung diesen Servo einzusetzen. Auch liegen die Parameter dieses Servos im gewünschten Rahmen. Der Servo braucht 0.15 Sekunden für eine Bewegung von 40°. Dies entspricht einer Geschwindigkeit von 267°/s. Der Vergleich mit anderen Servos von Robbe und Graupner zeigt, dass dieser Servo bereits einer der schnellsten ist. Sollte sich dieser Servo als untauglich erweisen, wird es schwierig, einen besseren Servo zu finden. Servos mit Metallgetriebe sind zwar präziser, aber kaum schneller als dieser Servo mit einem Plastikgetriebe.

Wird der Servo mit einem PWM-Signal von 50 Hz betrieben, ergibt sich eine maximale Totzeit von 20 ms. Da die Zeitkonstante des Motorrades um 126 ms beträgt, ist diese Totzeit relevant für die Modellierung. Mit einem Pulsgenerator wurde der Servo einem Test unterzogen. Die Frequenz des PWM-Signals hat keinen Einfluss auf die Winkellage des Servos. Die Winkellage des Servos hängt nur von der Pulsbreite ab. Wird die Pulsfrequenz über ca. 170 Hz erhöht, beginnt der Servo zu zittern und zu schwingen. Durch die Schwingungen kann die Mechanik des Servos beschädigt werden. Deshalb wurde die maximale PWM-Frequenz auf 150 Hz festgelegt, womit die Totzeit maximal nur noch 6.6 ms beträgt.

Da Graupner grundsätzlich keine Angaben zu ihren Bauteilen macht, konnten die dynamischen Parameter des Servos nicht in Erfahrung gebracht werden. Auf einen Messaufbau, um den Servo auszumessen, wurde aus Zeitgründen verzichtet.

5.4. Modellbauteile

In diesem Abschnitt wird die Modellbauelektonik beschrieben, welche im Motorrad eingebaut wurde. Alle beschriebenen Teile wurden bei Meile Modellbau, Frauenfeld beschafft.

5.4.1. Fahrtregler

Der Fahrtregler wird im Motorrad verwendet, um den Antriebsmotor mit Energie zu versorgen. Er erhält ein PWM-Signal vom Prozessorboard. Vorläufig wird dieses Signal äquivalent zum Eingangssignal vom Prozessor erzeugt, das heisst, dass das Geschwindigkeitssteuerungssignal von der Fernsteuerung direkt zum Fahrtregler weitergegeben wird.

Hersteller	Robbe
Typ	Fahrtregler Rohkraft 120E
Bestellnummer	8231
Preis	Fr. 124.00
Regelbereich	6 - 12 V

Der Fahrtregler versorgt den Motor, den Empfänger und zwei Servos mit der benötigten Energie. Damit ist der Fahrtregler an seiner Betriebsgrenze. Es können keine weiteren Servos angeschlossen werden. Als Schutz vor der Zerstörung der elektornischen Bauteile besitzt er eine automatische Überhitzungsabschaltung.

Der Fahrtregler hat die Eigenschaft, auch im unteren Tourenbereich fein zu regeln. Er besitzt die Modi „Vorwärts“, „Stop“ und „Rückwärts“. Er lässt sich über die beiden Potentiometer „Stop“ und „Geschw.“ einstellen.

Die „Stop“-Einstellung legt fest, in welcher Position sich der Steuerknüppel befindet, wenn der Fahrtregler den Motor im Modus „Stop“ hält. Das Potentiometer wurde so eingestellt, dass sich das Motorrad im „Stop“-Modus befindet, wenn der Steuerknüppel am unteren Anschlag ist. Das Modell kann momentan also nur vorwärts fahren kann. Durch verändern dieser Einstellung ist es aber möglich, das Motorrad auch rückwärts fahren zu lassen.

Die „Geschw.“-Einstellung legt die Maximalgeschwindigkeit fest. Momentan ist die Maximalgeschwindigkeit so eingestellt, dass das Modell die Sollgeschwindigkeit auf dem Prüfstand ohne Probleme erreicht.

Das genaue Vorgehen für die Einstellungen, der Beschaltungsplan und weitere technische Daten sind aus dem Datenblatt im Anhang ersichtlich.

5.4.2. Empfänger

Der Empfänger wandelt die Steuersignale von der Fernsteuerung in PWM-Signale um. Der Empfänger ist zusammen mit der Fernsteuerung in einem Set geliefert worden.

Hersteller	Graupner
Typ	Empfänger C 16 FM S 40
Bestellnummer	4037
Preis	Fr. 125.00 (falls separat bestellt)

Falls Probleme mit diesem Empfänger auftreten, ist es auch möglich, einen teueren und kleineren Empfänger zu kaufen. Der Empfänger C 19 FM S 40 beispielsweise ist in SMD-Technik gefertigt und darum robuster gegenüber Vibrationen.

Der Empfänger hat auf einer Seite eine Steckerleiste für neun Stecker. Auf einem Steckplatz muss die Versorgung des Empfängers und der angeschlossenen Bauteile eingesteckt werden. In den Empfänger können nur Stecker des Graupner Steckersystems eingesteckt werden. Die Stifte am Rand des Gehäuses sind miteinander verbunden und müssen an Ground angeschlossen werden. Auch die Stifte in der Mitte sind miteinander verbunden und müssen an die Versorgungsspannung angeschlossen werden. Es spielt also keine Rolle, an welcher Stelle der Fahrtregler, welcher die Energie liefert, eingesteckt wird.

 © Copyright 1998 SA

Die Steckerbelegung am Empfänger ist momentan wie folgt festgelegt:

1	Fahrtregler
4	Lenkservo
7	Gewichtsverlagerungsservo

Die Belegung wurde so gewählt, damit ein besserer Zugang zu den Steckern besteht. Ist der Empfänger auf dem Motorrad montiert, können die Stecker nicht mehr aus dem Empfänger entfernt werden. Um trotzdem das Prozessorboard vom Empfänger trennen zu können, wurden kurze Zwischenkabel mit systemkonformen Steckern eingefügt.

Die Signale, welche der Empfänger liefert, sind pulsbreitenmoduliert (PWM). Die Pulsfrequenz beträgt 50 Hz, was einer Pulsperiode von 20 ms entspricht. Die minimale Pulsbreite beträgt 1 ms und die maximale 2 ms. Diese Angaben entsprechen dem Ausgangssignal der Fernsteuerung, wobei „maximal“ bzw. „minimal“ der maximale bzw. minimalen Auslenkung des Steuerknüppels und der Trimmung entspricht.

Die Pulshöhe ist äquivalent zur Versorgungsspannung vom Fahrtregler. Der Empfänger arbeitet unabhängig der Versorgungsspannung und kann ohne Probleme zwischen 4 und 6 Volt betrieben werden.

5.4.3. Fernsteuerung

Da die ursprüngliche Empfängereinheit ausgebaut wird und mehr als vier Kanäle benötigt werden, musste eine neue Fernsteuerung beschafft werden.

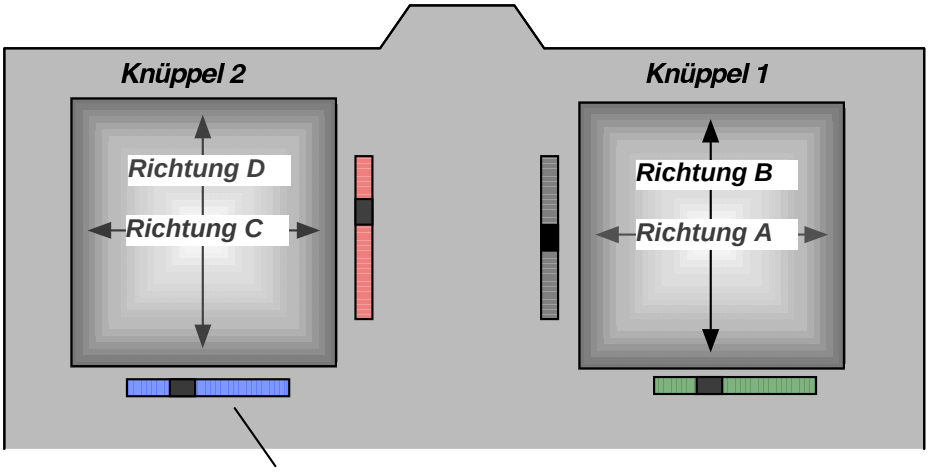
Hersteller	Graupner
Typ	Fernsteuerung FM 314
Bestellnummer	4746
Preis	Fr. 297.00
Kanäle	8/14 Kanälel
Modulation	40 MHz FM

Diese Fernsteuerung wurde gewählt, da die nächst grösseren Fernsteuerungen bereits ein Computersystem integriert haben. Diese erlauben zwar vorbereitete Abläufe automatisch ausführen zu lassen, aber die Preise bewegen sich ab circa Fr. 500.– aufwärts. Die nächst kleineren Fernsteuerungen haben bereits nur noch vier Kanäle, was für unsere Anforderungen nicht ausreicht.

Die Bezeichnung 8/14 Kanäle ist etwas verwirrend. Jede Bewegungsrichtung eines Knüppels (vorwärts, rückwärts, rechts und links) wird als ein Kanal bezeichnet, obwohl intern nur zwei Kanäle pro Knüppel verwendet werden. Das heisst, dass die Fernsteuerung eigentlich aus vier Kanälen besteht und auf sieben Kanäle ausbaubar ist.

Die Kanalbelegung ist momentan wie folgt festgelegt:

Knüppel 1	Richtung A seitwärts	Kanal 4	Lenkung
Knüppel 1	Richtung B vorwärts	Kanal ?	unbelegt
Knüppel 2	Richtung C seitwärts	Kanal 7	Gewichtsverlagerung, Auslenken der Batterie
Knüppel 2	Richtung D vorwärts	Kanal 1	Geschwindigkeit



Kanalbelegung Fernsteuerung

Standardmässig sind alle Knüppel mit einer Neutralisationsfeder ausgestattet. Zusätzlich kann eine Rasterfeder eingesetzt werden. Bei der Rasterfeder wird ein Metallplättchen an ein Raster am Steuerknüppel angedrückt, so dass der Knüppel nur noch in diskreten Schritten bewegt werden kann. Die Neutralisationsfeder bringt den Knüppel wieder in die Nullposition zurück, was insbesondere für die Links-Rechts-Steuerung sinnvoll ist.

Für den Knüppel 1 seitlich wurde eine Rasterfeder eingesetzt, damit leichter eine konstante Kurve gefahren werden kann. Für den Knüppel 2 längs wurde die Neutralisationsfeder ausgehängt, die Neutralisationsmechanik entfernt und ebenfalls eine Rasterfeder eingesetzt. Damit lässt sich der Fahrtregler nun auf eine konstante Geschwindigkeit einstellen, ohne dass der Knüppel ständig festgehalten werden muss. Das Raster lässt von Anschlag zu Anschlag 31 Rasterschritte zu.

Die Laufrichtung der Servos kann umgedreht werden, indem man den entsprechenden Anschlussstecker am Senderboard umdreht, was auch in der Betriebsanleitung auf Seite 4 und 7 im Anhang beschrieben ist. Die Fernsteuerung ist für beträchtliche Distanzen ausgelegt. Dennoch ist es sinnvoll, immer die Antenne einzusetzten, um auch bei schwacher Batterie einen sicheren Empfang zu garantieren.

Als Spannungsversorgung wurde nicht die mitgelieferte Batteriehalterung für 8 Batterien (1,5 V) verwendet, sondern ein NC-9,6V-Akku ins Batteriefach eingesetzt. Die Fernsteuerung kann mit einem Kabel, ohne dass sie geöffnet werden muss, direkt am Ladegerät aufgeladen werden. Die Leistung der Batterie bricht relativ schnell zusammen, was auf der Batterieanzeige gut beobachtet werden kann. Der Akku der Fernsteuerung muss nach einer Gesamtentladung 14 Stunden aufgeladen werden, damit er wieder seine volle Leistung bringt.

Damit die Nullwerte, welche von der Fernsteuerung bzw. vom Empfänger an den Prozessor geschickt werden, immer dieselben sind, hätten eigentlich die Trimmungen an der Fernsteuerung entfernt werden sollen. Leider können sie ohne Beschädigung der Fernsteuerung nicht entfernt werden. Um ein unbeabsichtigtes Verstellen zu verhindern, können die Trimmungen lediglich abgedeckt werden.

Die benötigten Manipulationen und weitere Details zur Fernsteuerung kann man der Betriebsanleitung im Anhang entnehmen.

5.4.4. Leistungsakku

Für den Antrieb und die Vorsorgung der Bordelektronik des Motorrades musste ein passender Akku gekauft werden.

Hersteller	Cadnica
Typ	Standardakku NI-CD GN-1700SCR
Preis	Fr. 70.00
Spannung	7.2 V
Speicherenergie	1700 mAh

 © Copyright 1998 SA

Der Einsatz eines Standardakkus genügt, da die Elektonikplatine keine speziellen Leistungsanforderungen stellt. Mit diesem Akku ist es also möglich, sowohl den Antrieb, als auch die Elektonik zu versorgen. Da ein Akku nach ca. 15 Minuten voller Fahrt entladen ist, wurden zwei Akkus gekauft. Damit entstehen bei Tests keine längere Wartezeiten.

5.4.5. Ladegerät

Um die Leistungsakkus und den Akku der Fernsteuerung aufladen zu können, wurde ein Ladegerät benötigt. Folgendes Produkt wurde für diese Zwecke beschafft:

Typ	CB 97
Hersteller	Arlec
Bestell-Nr.	22.13.010
Preis	Fr. 160.00

Es gibt auch einfachere Ladegeräte von der Firma Arlec. Bei diesen muss man aber selbst wissen, wie lange man den Akku aufladen soll.

Ein anderes Ladegerät wird von der Firma Robbe hergestellt. Dieses Netzgerät hat eine automatische Abschaltung und Anschlussmöglichkeiten für einen Akku und die Fernsteuerung. Leider hat dieses Gerät keine Entladefunktion, welche für eine längere Lagerung der Batterien benötigt wird.

Das Ladegerät CB 97 von Arlec hat eine automatische Abschaltung und eine Entladefunktion. Mit diesem Ladegerät können gleichzeitig drei Akkus aufgeladen werden, nämlich ein Leistungsakku (Motorrad), ein Senderakku und ein Empfängerakku. Da der Empfänger auf dem Motorrad vom Leistungsakku gespiesen wird, konnte auf einen Empfängerakku verzichtet werden.

Mit der Schnelladefunktion des Ladegerätes (4 A) ist ein Leistungsakku in etwa 1.5 Stunden aufgeladen. Das Ladegerät zeigt die Beendigung des Ladevorgangs an. Der Senderakku wird nur mit 160 mA aufgelden. Darum braucht der Sender Akku etwa 14 Stunden, bis er voll aufgeladen ist.

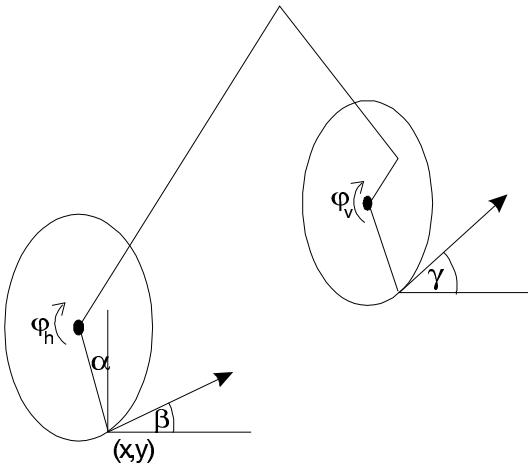
Grundsätzlich ist ein langsames Aufladen der Batterien besser, da dann die einzelnen Zellen sauber aufgeladen werden. Zur Lagerung sollten vor allem die Leistungsakkus vollständig entladen werden, damit ihre Leistungsfähigkeit erhalten bleibt.

6. Modellierung

Wie soll das Verhalten eines Zweirades modelliert werden? In diesem Kapitel wird das Motorrad als invertiertes Pendel modelliert. Die Modellierung des Inklinometers und der Servos vervollständigt die gesamte Strecke. Eine Aufstellung aller Parameter der Strecke und die verwendeten Messverfahren sind ebenfalls beschrieben. Im Abschnitt 6.5 wird neben einer qualitativen Abschätzung des Modellierungsfehlers auch erklärt, warum ein Zweirad selbststabilisierend ist. Mit dem Gesamtmodell am Ende des Kapitels, wird im folgenden Kapitel ein Regler entworfen.

6.1. Modellierung des Motorrades

Ein Zweirad besteht aus vier Körpern: dem Hinterrad, dem Vorderrad, dem Rahmen und der Gabel. Das Hinterrad und der Rahmen sind durch die Achse fest miteinander verbunden. Sie können sich nur um die Achse gegeneinander drehen. Das Gleiche gilt für das Vorderrad und die Gabel. Auch die Gabel und der Rahmen sind fix verbunden und können sich nur um die Achse des Gabelrohrs gegeneinander drehen. Beide Räder stehen auf dem Boden.



Zweirad

Um die Position und Ausrichtung im Raum zu beschreiben, benötigen wir folgende Koordinaten:

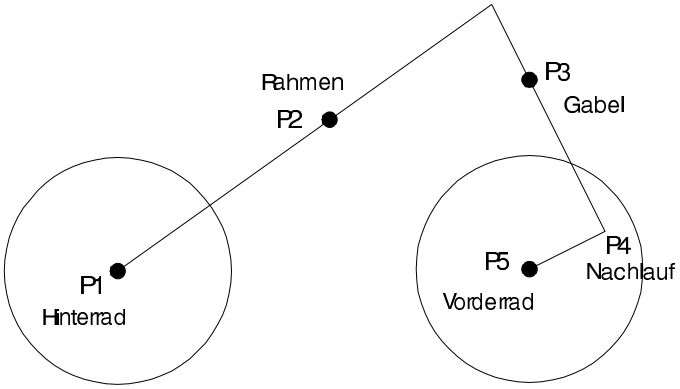
Schräglage des Hinterrades	α
Fahrtrichtung des Hinterrades	β
Fahrtrichtung des Vorderrades	γ
Ort des Auflagepunktes des Hinterrades	x, y
Winkellage des Hinter- und Vorderrades	φ_h, φ_v

Ein lenkbares Zweirad hat also immer 7 Freiheitsgrade.

6.1.1. Modellierung als Vielkörpersystem

Ein Zweirad ist ein Vielkörpersystem. Vielkörpersysteme werden in der Mechanik meistens mit dem Aufstellen der Lagrange-Gleichung gelöst. Nur wenn im Vielkörpersystem keine geschlossene Geometrie vorkommt, kann das System mit der herkömmlichen Mehrkörperdynamik behandelt werden. Eine geschlossene Geometrie ist dann vorhanden, wenn mit Durchfahren mehrerer Körper und der Auflageumgebung eine geschlosse Schleife gebildet werden kann, wenn also zwei Körper des gleichen Systems die Auflagefläche berühren. Ein Zweirad hat auch eine geschlossene Geometrie, da sowohl das Vorderrad, als auch das Hinterrad ständig den Boden berühren. Die Dynamik eines Zweirades kann also nur mit Aufstellen der Lagrange-Gleichung gelöst werden.

 © Copyright 1998 SA



Zweirad

Bei dieser Methode werden für jeden Körper Koordinaten für sämtliche Freiheitsgrade eingeführt. Danach werden Bindungsgleichungen aufgestellt, welche den Bezug zwischen den Körpern herstellen und die Gesamtzahl der Freiheitsgrade verringern. Auch für Einschränkungen, welche die Umgebung vorgibt, werden Bindungsgleichungen aufgestellt. Dabei muss zwischen holonomen und nichtholonomen Gleichungen unterschieden werden. Holonome Bindungsgleichungen beschreiben die Geometrie des Vielkörpersystems, z. B. den Abstand zwischen zwei Körpern. Nichtholonome Bindungsgleichungen beschreiben die Bewegungsrichtung gegenüber der Umgebung, das heisst, die Gleichungen hängen auch von der Ableitung einer Koordinate ab. Nichtholonome Gleichungen beschreiben zum Beispiel die Fahrtrichtung eines Rades auf dem Boden. Somit handelt es sich bei einem Zweirad um ein nichtholonomes System.

Für die Modellierung des nichtholonomen Mehrkörpersystems „Zweirad“ muss ein Funktionaldifferentialgleichungssystem gemäss der Lagrange-Mechanik aufgestellt werden. Dieses Differentialgleichungssystem kann mit geeigneten DAE-Tools (Differential Algebraic Equation) gelöst werden. Sowohl Neil Getz [Get] als auch Dirk von Wissel [Wis] benutzten für ihre Simulationen diese Methode. Bei beiden Veröffentlichungen handelt es sich aber um theoretische Abhandlungen der Möglichkeiten zur Regelung eines Zweirades. Eine Implementation der vorgeschlagenen Regelungsalgorithmen ist aufgrund des Rechenleistungsbedarfs illusorisch. Die Modellierung eines Vielkörpersystems nur für Simulationszwecke sprengt die zeitlichen Grenzen dieser Semesterarbeit.

Wie soll ein Vielkörpersystem einfacher modelliert werden?

Gesucht ist ein Einkörpersystem, welches sich möglichst ähnlich einem Zweirad verhält. Betrachtet man ein Zweirad bei der Fahrt von hinten, schwenkt es jeweils vor einer Kurve aus der senkrechten Lage aus, legt sich in die Kurve und richtet sich danach wieder auf. Die Bewegung ist vergleichbar mit der Bewegung eines invertierten Pendels, welches sich so auf einer ebenen Trajektorie fortbewegt, dass es nicht umfällt.

Im folgenden wird das Motorrad als invertiertes Pendel modelliert. Dieses Vorgehen birgt aber das Risiko, dass die Modellierung nicht ausreicht, und der Regler nicht die gewünschten Ergebnisse bringt. Das Motorrad und das invertierte Pendel sind zwei unterschiedliche Systeme. Falls das Projekt an der Modellierung scheitert, müsste das Motorrad als Mehrkörpersystem modelliert werden.

6.1.2. Modellierung als invertiertes Pendel

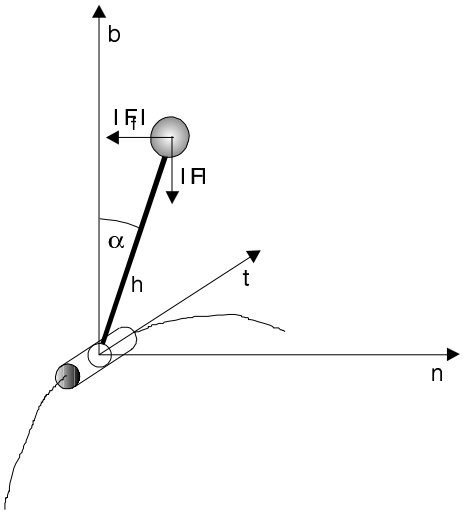
Zuerst wird ein mathematisches Modell des bewegten invertierten Pendels aufgestellt. Es wird danach noch mit den Drallvektoren der Räder erweitert. Als Ergebnis erhält man die Zustandsdarstellung der Bewegungen des invertierten Pendels.

Wird das Zweirad als invertiertes Pendel modelliert, fallen in erster Linie die Geometrie und die Ausdehnung des Zweirades weg. Damit verschwinden die Kräfte zwischen den einzelnen Körpern, insbesondere zwischen dem Rahmen und der Gabel. Diese Dynamiken sind im Modell des invertierten Pendels nicht vorhanden.

Das bewegte invertierte Pendel

Beim bewegten invertierten Pendel handelt es sich um ein Einkörpersystem. Das Pendel hat seine gesamte Masse konzentriert in einem Massepunkt im Schwerpunkt, das heisst im Abstand h vom Auflagepunkt. Die Auflage ist ein langes Lager um die ebene Trajektorie, das heisst, das Pendel kann sich nur um die Drehachse des Lagers bewegen. Durch diese Beschränkung besitzt das System nur noch vier Freiheitsgrade: Die Koordinaten x, y des Auflagepunktes, den Winkel β der Richtung der Trajektorie und den Winkel α zwischen der Vertikalen und dem Pendel.

Die Trajektorie entspricht der Verbindung zwischen den Auflagepunkten der beiden Räder auf dem Boden. Das Pendel bewegt sich vorwärts auf dieser ebenen Trajektorie. Um die Krümmung der Trajektorie zu ändern, muss der Lenkwinkel geändert werden. Durch die Lenkung wird der Verlauf der Trajektorie bestimmt. Die Richtung des Lagers wird also durch die Fahrtrichtung vorgegeben.



invertiertes Pendel

Um die physikalischen Zusammenhänge zu erfassen, wird im Auflagepunkt ein begleitendes Dreibein eingeführt. Durch die Einführung eines begleitenden Koordinatensystems verschwinden die Positionskoordinaten x, y und β. Das invertierte Pendel hat in diesem Koordinatensystem nur noch einen Freiheitsgrad, den Winkel α. Das Dreibein hat folgende Koordinaten:

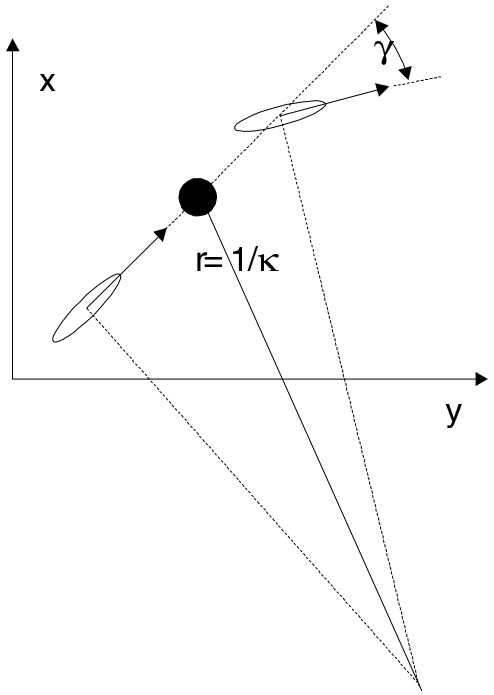
n-Koordinate normal (senkrecht) auf der Trajektorie in der Auflageebene
t-Koordinate tangential zur Trajektorie in der Auflageebene
b-Koordinate binormal (senkrecht) auf den anderen Koordinaten

Das Pendel kann sich nur in der b/n-Ebene drehen. Der Winkel α ist der Winkel zwischen der Koordinatenachse b und dem Pendel. Die n-Koordinate liegt immer auf der Geraden durch den Auflagepunkt und den Mittelpunkt des Krümmungskreises an die Trajektorie.

Das Pendel bewegt sich mit gleichbleibender Geschwindigkeit v auf der Trajektorie. Sämtliche Kräfte in tangentialer Richtung werden durch die Trägheit des Körpers und das Drehmoment des geregelten Motors aufgenommen. Sie bewirken also keine Geschwindigkeitsänderung.

 © Copyright 1998 SA

Krümmung der Trajektorie



Lenkwinkel und Kurvenkrümmung

Um die Kräfte zu berechnen, wird der Radius des Krümmungskreises benötigt, welcher direkt mit dem Lenkwinkel des Motorrades zusammenhängt. Der Abstand zwischen Schwerpunkt und Mittelpunkt des Krümmungskreises beträgt:

$$r_k = \frac{1}{\kappa} - h \cdot \sin\alpha \qquad \kappa = \frac{\tan\gamma}{l}$$

Die Krümmung κ hängt direkt mit dem Lenkwinkel γ und dem Radstand l zusammen. Wird für den Schwerpunkt angenommen, dass er direkt über dem Auflagepunkt des Hinterrades liegt, dann ergibt sich für κ folgende Gleichung:

$$\kappa = \frac{\tan\gamma}{l}$$

Wird für den Schwerpunkt angenommen, dass er direkt über dem Auflagepunkt des Vorderrades liegt, dann ergibt sich für κ folgende Gleichung:

$$\kappa = \frac{\sin\gamma}{l}$$

Linearisiert man die beiden Gleichungen erhält man für beide Schwerpunkte dieselbe linearisierte Gleichung:

$$\left.\begin{aligned} \sin\gamma &\cong \gamma \\ \tan\gamma &\cong \gamma \end{aligned}\right\} \qquad \kappa = \frac{\gamma}{l}$$

In der Realität befindet sich der Schwerpunkt zwischen den beiden Auflagepunkten. Um die weiteren Berechnungen nicht unnötig zu verkomplizieren, wird der Schwerpunkt bis zur Linearisierung über dem Auflagepunkt des Hinterrades angenommen. Durch die Linearisierung wird der Schwerpunkt zwischen den beiden Auflagepunkten zu liegen kommen.

Kräfte im relativen System

Führungskraft in n-Richtung

$$\underline{F}_f = m \cdot \underline{a}_f = -mv^2 \cdot \frac{\kappa}{1 - h \cdot \kappa \cdot \sin(\alpha)} \cdot \underline{e}_n = -mv^2 \cdot \frac{\tan(\gamma)}{l - h \cdot \sin(\alpha)\tan(\gamma)} \cdot \underline{e}_n$$

wobei m die Masse, $\underline{a}_f$ die Zentripetalbeschleunigung, κ die Krümmung der Fahrkurve, v die Geschwindigkeit, γ der Lenkwinkel und l der Radstand sind.

Corioliskraft in t-Richtung

$$\underline{F}_C = m \cdot \underline{a}_C = 2 \cdot m \cdot \underline{v}_r \times \underline{\omega}_f = 2 \cdot m \cdot \dot{\alpha} \cdot h \cdot v \cdot \kappa \cdot \sin(\alpha) \cdot \underline{e}_t$$

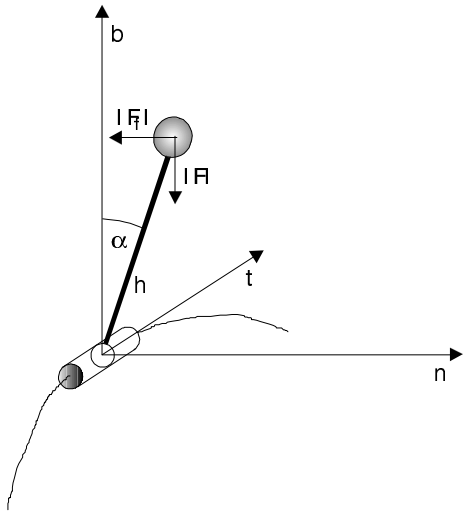
wobei $\underline{a}_C$ die Coriolisbeschleunigung, $\underline{v}_r$ die Relativgeschwindigkeit, $\underline{\omega}_f$ die Führungskreisfrequenz sind.

Wirkliche Kraft in b-Richtung

$$\underline{F} = -m \cdot g \cdot \underline{e}_b$$

wobei g die Erdbeschleunigung ist.

Gemäss Voraussetzung bewegt sich unser System mit gleichbleibender Geschwindigkeit, das heisst, dass Kräfte in t-Richtung keine Wirkung haben. Die Corioliskraft steht senkrecht auf der b/n-Ebene. Sie hat also keinen Einfluss auf die Bewegung des Pendels.



Kräfte am invertierten Pendel

Die Führungskraft und die Gravitationskraft wirken auf die Bewegungsrichtung des Pendels im relativen Koordinatensystem. Daraus ergibt sich folgende Differentialgleichung:

$$J\ddot{\alpha} = mgh \cdot \sin(\alpha) - mv^2 \cdot \frac{\tan(\gamma)}{l - h \cdot \sin(\alpha)\tan(\gamma)} \cdot h \cdot \cos(\alpha)$$

wobei J das Trägheitsmoment des Pendels ist.

Für das invertierte Pendel gilt, da es sich um einen Massepunkt handelt, dass das Trägheitsmoment folgenden Wert hat:

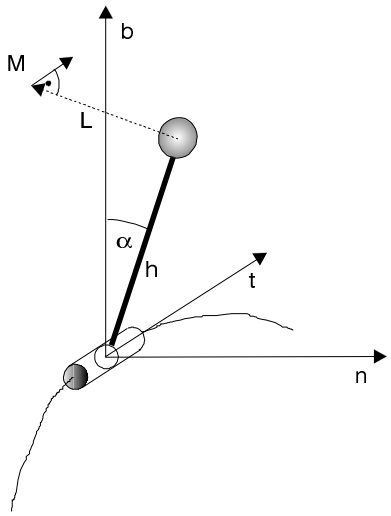
$$J = m \cdot h^2$$

Das reale Modell ist ausgedehnt. Es besitzt ein grösseres Trägheitsmoment als der Massepunkt. Das Trägheitsmoment des Motorradmodells wird im Kapitel 6.4 (Bestimmung der Parameter) gemessen.

 © Copyright 1998 SA

Erweitertes Pendel mit Drallvektoren

Das Pendel wird nun mit den Drallvektoren der beiden Räder erweitert. Die Dralle können an einer beliebigen Stelle am invertierten Pendel angreifen, da die Kraftwirkungen, das heisst die Drehmomente, unabhängig vom Angriffspunkt sind.



Drehmoment am invertierten Pendel

Der Drallvektor des Hinterrades liegt in der b/n-Ebene und steht senkrecht zum Pendel. Der Drall des Hinterrades beträgt:

$$L_h = J_h \cdot \frac{v}{r_h}$$

wobei J_h das Trägheitsmoment und r_h der Radius des Hinterrades ist.

Der Drallvektor des Vorderrads steht ebenfalls senkrecht zum Pendel, ist jedoch um den Winkel γ aus der b/n-Ebene abgelenkt. Der Drall des Vorderrades beträgt:

$$L_v = J_v \cdot \frac{v}{r_v}$$

wobei J_v das Trägheitsmoment und r_v der Radius des Vorderrades ist.

Da der Winkel γ nur kleine Werte annimmt, kann für die Bewegung des gesamten Pendels die Summe der beiden Dralle verwendet werden. Für das Auslenken des Vorderrades darf aber nur der Drall des Vorderrades betrachtet werden.

$$L = L_h + L_v$$

Die Beschreibung, wie die Trägheitsmomente der Räder gemessen wurden, befindet sich im Kapitel 6.4 (Bestimmung der Parameter).

$$\dot{L} = M$$

Gemäss Drallsatz erzeugt die Änderung des Dralls ein Moment in Richtung der Änderung. Die Anteile der Momente, welche in der b/n-Ebene liegen, haben keinen Einfluss auf die Bewegung des Pendels. Nur die Anteile der Momente, welche in t-Richtung zeigen, üben ein Moment auf das Pendel aus.

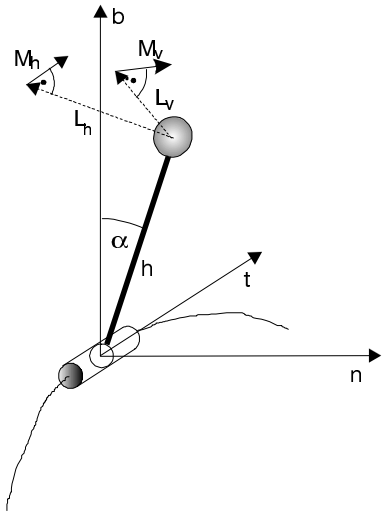
Dies ergibt, dass die Momente, welche durch die Änderung von α erzeugt werden und alle in der b/n-Ebene liegen, keinen Einfluss auf den Winkel α haben.

Dralländerungen durch die Kurvenfahrt erzeugen ein Moment in t-Richtung. Es steht also senkrecht auf der b/n-Ebene.

$$\underline{M} = \underline{\omega}_f \times \underline{L} = v \cdot \frac{\tan(\gamma)}{l} \cdot L \cdot \cos(\alpha) \cdot \underline{e}_t$$

Durch das Lenken wird eine Dralländerung am Vorderrad erzeugt, welche bei kleinem γ in t-Richtung zeigt:

$$\underline{M}_v = \dot{\gamma} \cdot L_v \cdot \cos(\alpha) \cdot \underline{e}_t$$



Drehmomente am invertierten Pendel

Nichtlineare Differentialgleichung

Die Differentialgleichung des erweiterten Pendels lautet:

$$J\ddot{\alpha} = mgh \cdot \sin(\alpha) - mv^2 \cdot \frac{\tan(\gamma)}{l - h \cdot \sin(\alpha)\tan(\gamma)} \cdot h \cdot \cos(\alpha) + \frac{vL}{l}\cos(\alpha)\tan(\gamma) + L_v \cdot \dot{\gamma} \cdot \cos(\alpha)$$

Linearisieren

Da vorerst nur mit kleinen Winkeln (Geradeausfahrt) gerechnet wird, können im ersten Schritt die trigonometrischen Funktionen wie folgt linearisiert werden:

$$\sin(x) \cong x, \qquad \cos(x) \cong 1, \qquad \tan(x) \cong x$$

$$J\ddot{\alpha} = mgh \cdot \alpha - mv^2 h \cdot \frac{\gamma}{l - h \cdot \alpha \cdot \gamma} + \frac{vL}{l} \cdot \gamma + L_v \cdot \dot{\gamma}$$

Der zweite Summand kann durch eine Taylor-Reihe, welche nach dem ersten Glied abgebrochen wird, um den Punkt $\gamma_0 = \alpha_0 = 0$ linearisiert werden. Dabei muss die Taylor-Reihe nach zwei Variablen entwickelt werden.

$$f(\gamma,\alpha) = \frac{\gamma}{l - h \cdot \alpha \cdot \gamma}$$

$$f(\gamma_0 + \gamma, \alpha_0 + \alpha) \cong \underbrace{f(\gamma_0, \alpha_0)}_{=0} + \underbrace{\gamma \cdot \left(\frac{\partial}{\partial \gamma} f(\gamma,\alpha) \Big|_{\substack{\gamma=\gamma_0 \\ \alpha=\alpha_0}} \right)}_{=\frac{\gamma}{l}} + \underbrace{\alpha \cdot \left(\frac{\partial}{\partial \alpha} f(\gamma,\alpha) \Big|_{\substack{\gamma=\gamma_0 \\ \alpha=\alpha_0}} \right)}_{=0}$$

Um das System besser regelbar zu machen, wird zudem eine kleine Dämpfung d eingeführt (d = 0.01). Die linearisierte Differentialgleichung lautet:

 © Copyright 1998 SA

$$J\ddot{\alpha} = mgh \cdot \alpha - \frac{mv^2 h}{l} \cdot \gamma + \frac{vL}{l} \cdot \gamma + L_v \cdot \dot{\gamma} - mgh \cdot d \cdot \dot{\alpha}$$

Die Koeffizienten werden dimensionsbefreit. Damit kann man abschätzen, wie schnell sich das System bewegt, denn es ergibt sich auf der linken Seite der DGL eine Zeitkonstante, welche die Geschwindigkeit des Systems beschreibt.

$$T^2\ddot{\alpha} = \alpha - d \cdot \dot{\alpha} - \frac{v}{gl}\left(v - \frac{L}{mh}\right) \cdot \gamma + \frac{L_v}{mgh} \cdot \dot{\gamma}$$

wobei $T^2 = \frac{J}{mgh}$

setzt man $J = m \cdot h^2$ ein, ergibt sich als erste Abschätzung $T = \sqrt{\frac{h}{g}}$. Die Höhe des Schwerpunktes des Modellmotorrades beträgt 11,4 cm, was für die Zeitkonstante einen ersten Anhaltspunkt von 108 ms ergibt.

Die Differentialgleichung wird in ein Differentialgleichungssystem mit folgenden Grössen umgeschrieben, damit man das System im Zustandsraum darstellen kann:

$$x_1 = \alpha, \qquad x_2 = \dot{\alpha}, \qquad u_m = \gamma$$

$$\begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -d \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \cdot u_m$$

wobei $b_2 = -\frac{v}{gl}\left(v - \frac{L}{mh}\right) = -\frac{v^2}{gl}\left(1 - \frac{gl}{mh}\left(\frac{J_h}{r_h} + \frac{J_h}{r_h}\right)\right)$

und $b_1 = \frac{1}{T}\frac{L_v}{mgh} = v\frac{J_v}{r_v J}$.

Die Ableitung des Eingangssignals γ wurde eliminiert, indem das Signal bereits nach dem ersten Integrator auf das System wirkt. Durch den Integrator wird die Ableitung kompensiert.

Nur die Koeffizienten des Eingangssignals hängen von der Geschwindigkeit v ab. Das System verhält sich unabhängig von der Geschwindigkeit immer gleich, wenn das Eingangssignal null ist.

Die Zustandsraumdarstellung dieser Gleichung lautet:

$$\underline{\dot{x}} = A \cdot \underline{x} + B \cdot u_m$$

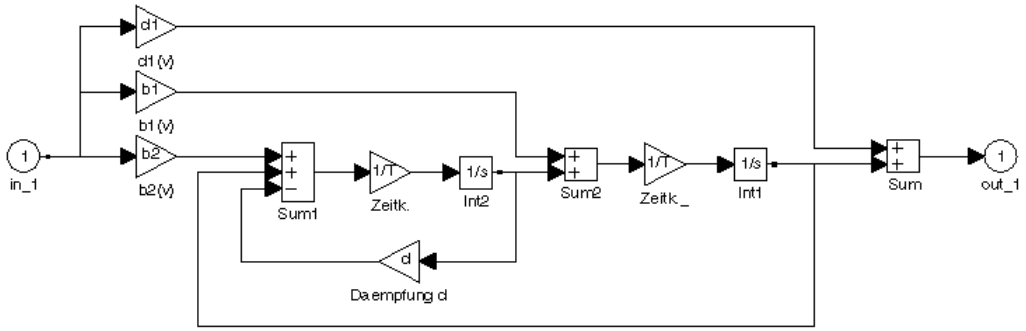
$$A = \frac{1}{T}\begin{bmatrix} 0 & 1 \\ 1 & -d \end{bmatrix}, \qquad B = \frac{1}{T}\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Das System besitzt zwei Pole

$$s_{1,2} = \frac{-d \pm \sqrt{d^2 + 4}}{2T}$$

Die Pole liegen beinahe symmetrisch zur Imaginärachse in der linken / stabilen und rechten / instabilen Halbebene.

Aus den obigen Berechnungen wurden die Matrizen A und B der Zustandsraumdarstellung bestimmt. Damit ist der Zusammenhang zwischen der Lenkbewegung und den internen Zuständen modelliert. Das System A, B ist steuerbar. Im folgenden Abschnitt werden auch noch die Matrizen C und D für den Ausgang berechnet. Daraus kann abgeleitet werden, dass das System auch beobachtbar ist. Aus dieser Darstellung kann in Simulink ein Modell aufgebaut werden, welches später zur Simulation eines Reglers dient.



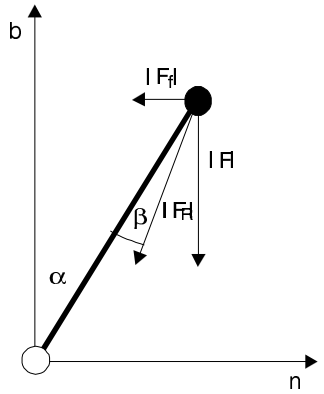
Simulink-Modell des Motorrades

6.2. Modellierung des Inklinometers

In diesem Teil wird zuerst der Eingangswinkel des Inklinometers berechnet. Als Resultat erhält man die Ausgangsmatrizen der Pendelmodellierung von oben. Da das Inklinometer stark schwingt und eine erhebliche Ausregelzeit hat, musste ein Modell für das Inklinometer identifiziert werden. Mit der Hilfe eines mechanischen Aufbaus wurde die Sprungantwort aufgenommen, woraus man die Parameter ablesen konnte.

6.2.1. Eingangswinkel des Inklinometers

Das Inklinometer zeigt den Winkel β zwischen der Schräglage α des Motorrades und der resultierenden Kraft aus der Führungskraft und der Gravitationskraft.



Eingangswinkel des Inklinometers

Die Auslenkung β des Inklinometers berechnet sich wie folgt aus den Kräften:

$$\beta = \alpha - \tan\left(\frac{F_f}{F}\right)$$

Die Kräfte betragen, wie oben berechnet:

$$\underline{F}_f = -mv^2 \cdot \frac{\tan(\gamma)}{l - h \cdot \sin(\alpha)\tan(\gamma)} \cdot \underline{e}_n \qquad \underline{F} = -m \cdot g \cdot \underline{e}_b$$

Linearisiert lauten die Kräfte:

$$\underline{F}_f = -\frac{mv^2}{l} \cdot \gamma \cdot \underline{e}_n \qquad \underline{F} = -m \cdot g \cdot \underline{e}_b$$

 © Copyright 1998 SA

Eingesetzt in der Gleichung und linearisiert für kleine Winkel, ergibt sich folgende Beziehung:

$$\beta = \alpha - \frac{v^2}{gl} \cdot \gamma$$

Der Ausgang des Inklinometers hängt also sowohl von der Schräglage, als auch direkt vom Lenkwinkel γ ab.

Für die obige Zustandsraumdarstellung ergeben sich folgende Ausgangsmatrizen:

$$y_m = C \cdot \underline{x} + D \cdot u_m$$

$$C = [1 \quad 0], \qquad D = [d_1], \qquad d_1 = -\frac{v^2}{gl}$$

Der Durchgriff D ist dominant gegenüber C. Das heisst, dass bereits im Eingangssignal des Inklinometers vor allem Information über den Lenkwinkel, aber nur wenig über den Kippwinkel α enthalten ist. Es ist schwierig, den Winkel α aus diesem Signal genau zu bestimmen.

Die Übertragungsfunktion des Motorrades lautet:

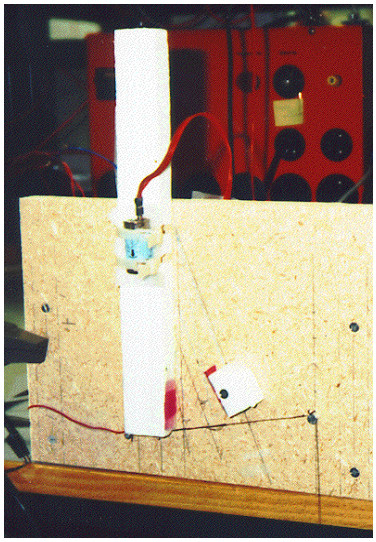
$$G_m(s) = \frac{s^2 d_1 + s\frac{1}{T}(d_1 d + b_1) + \frac{1}{T^2}(d b_1 + b_2 - d_1)}{s^2 + s\frac{d}{T} - \frac{1}{T^2}}$$

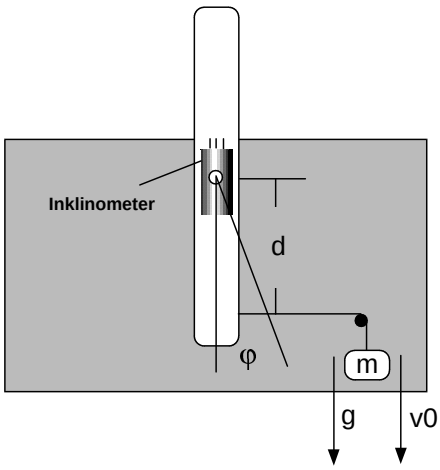
Die Nullstellen der Übertragungsfunktion liegen ebenfalls beinahe symmetrisch zur Imaginärachse in der linken und rechten Halbebene.

6.2.2. Identifikation der Übertragungsfunktion des Inklinometers

Gemäss Herstellerangaben [Pew] beträgt die Ausregelzeit ca. 0.3 Sekunden. Aufgrund der Messart mit einem in Öl gelagerten Pendel ist ein starkes Schwingen zu erwarten. Wegen diesem Verhalten musste das Vorhaben, kein Modell des Inklinometers aufzustellen, fallen gelassen werden. Für eine exakte Modellierung anhand von physikalischen Gesetzen wäre der genaue innere Aufbau und Wissen über die Mechanik in viskosen Flüssigkeiten nötig gewesen. Deswegen musste das Inklinometer identifiziert werden, um ein Modell des physikalischen Verhaltens des Inklinometers zu erhalten.

Mit Hilfe eines mechanischen Messaufbaus wurden mehrere Sprungantworten aufgenommen.





Messaufbau

Beim Messaufbau handelt es sich um ein Holzgestell mit einem Styroporpendel. Das Pendel kann eine Bewegung von 20° machen, bevor ein Anschlag aus Styropor das Pendel stoppt. Damit das Pendel vom Anschlag nicht zurückprallt, wurde ein Klettverschluss so befestigt, dass das Pendel beim Aufprall hängenbleibt. Im Drehpunkt des Pendels wird das Inklinometer befestigt. Es macht die Bewegungen des Pendels mit. Damit der Sprung auf dem Pendel möglichst steil ist, wurde Styropor verwendet. Aufgrund der kleinen Masse treten keine Verzögerungen auf.

Beim Bewegen des Pendels aus der senkrechten Lage wird ein Kontakt eines Stromkreises geöffnet. Das Signal wurde als Trigger verwendet und ist jeweils in der unteren Hälfte der Bilder aufgezeichnet. Die Signale wurden mit einem Speicherkathodenstrahloszilloskop aufgenommen. Das Signal aus dem Inklinometer besitzt einen Offset von 2.5 Volt. Das KO wurde so eingrichtet, dass nach Abzug des Offsets die Null-Volt-Linie in der Mitte des Bildes liegt.

Der Sprung wurde beim ersten Versuch mit einem Gewicht ausgelöst. Dabei wurde das Gewicht fallengelassen und zog bei hoher Geschwindigkeit das Pendel an den Anschlag. Anhand der Physik von fallenden Körpern kann die Steilheit der Rampe berechnet werden.

Die Bewegung, welche der Sensor auf dem Messaufbau durchläuft, wenn die Reibung und die Trägheitsmomente vernachlässigt werden, lässt sich mit folgender Formel beschreiben:

$$\varphi = 0.5 \cdot \frac{g}{d} \cdot t^2 + \frac{v_0}{d} \cdot t$$

wobei g die Erdbeschleunigung, d der Abstand zwischen dem Drehpunkt des Pendels und dem Befestigungspunkt der Zugschnur und v_0 die Anfangsgeschwindigkeit sind.

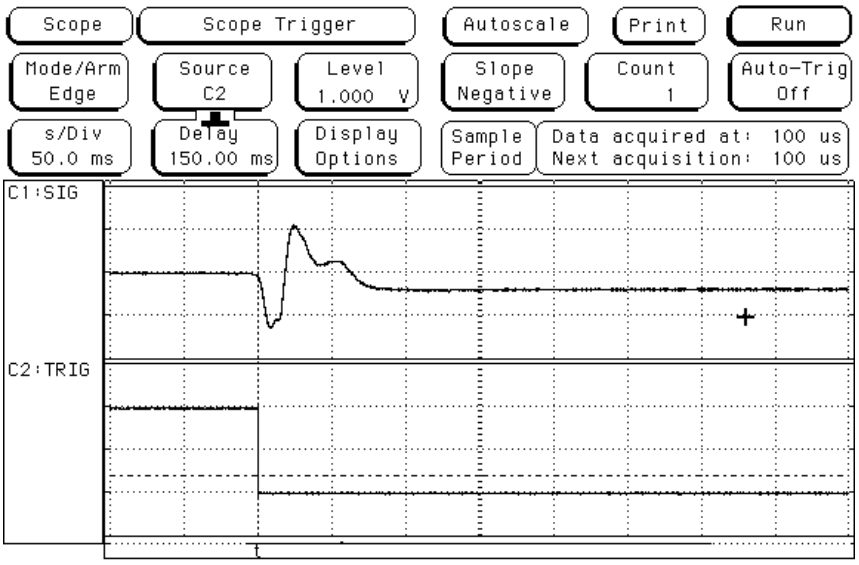
Die Bewegung der Masse wird beschrieben durch:

$$x = 0.5 \cdot gt^2 + v_0 t$$

Soll das Pendel eine Drehbewegung von 20 Grad machen, muss die Masse an der Zugschnur vertikal um 35 mm nach unten verschoben werden. Die Masse hat am Anfang keine Geschwindigkeit, das heisst, $v_0 = 0$. Soll die Drehbewegung von 20 Grad innerhalb von 10 ms gemacht werden, dann muss die Masse aus einer Höhe von 1.75 m fallengelassen werden. Daraus resultiert eine Rampe für die Auslenkung, welche für eine Messung des Antwortverhaltens des Sensors verwendet werden kann.

Im zweiten Versuch wurde das Pendel per Hand ausgelöst. Mit dem Zeigefinger wurde ans Pendel geschlagen um einen möglichst steilen Sprung zu erhalten.

Versuch 1: Sprung ausgelöst durch ein Gewicht.

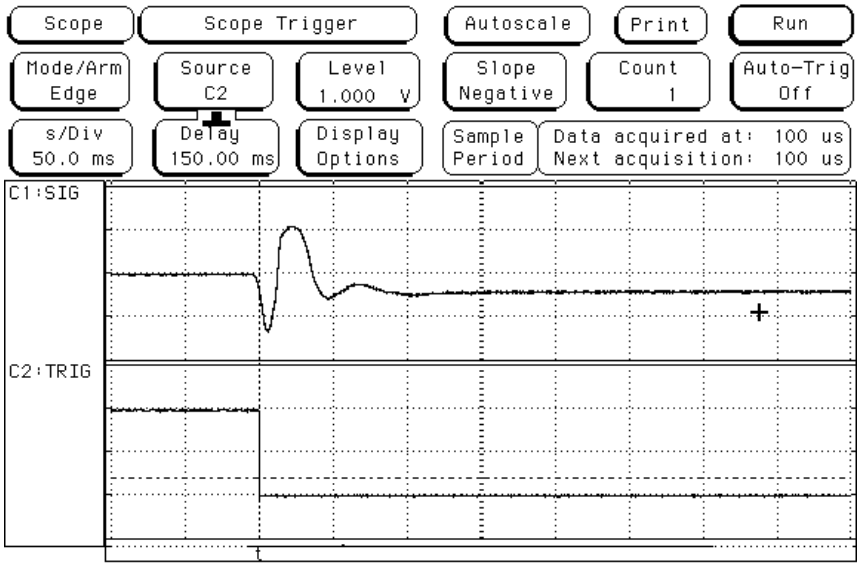


Versuch 1 / Auflösung 50 ms/Div

 © Copyright 1998 SA

Die Zeitachse ist in 50 ms je Teilung unterteilt. Es sind 0.5 Sekunden abgebildet. Die Spannungsachse ist in 1 Volt je Teilung unterteilt.

Versuch 2: Sprung von Hand ausgelöst.



Versuch 2 / Auflösung 50 ms/Div

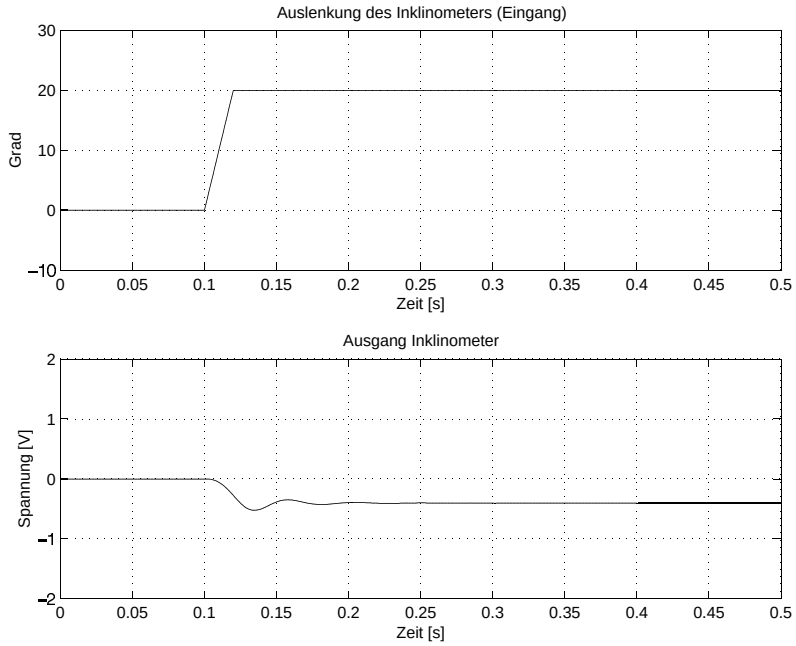
Aus den Messungen ist ersichtlich, dass das Inklinometer ein starkes Überschwingen hat. Das Inklinometer wird als lineares Glied zweiter Ordnung identifiziert, das heisst die Übertragungsfunktion lautet wie folgt:

$$G_i(s) = \frac{k_{ink}}{\frac{s^2}{\omega^2} + s\frac{2d}{\omega} + 1}$$

Die Verstärkung k_{ink} hängt direkt mit der Versorgungsspannung des Inklinometers zusammen. Anhand von Spannungsmessungen an verschieden Winkelpositionen wurde die genaue Verstärkung ermittelt. Die Schwingfrequenz ω und die Dämpfung d können aus obigen Messungen herausgelesen werden. Es ergeben sich die Werte ω = 210 rad/s und d = 0.15.

Im folgenden Kapitel wird für alle Parameter erklärt, wie und wo sie gefunden oder gemessen wurden. Auch befindet sich im Anhang das Parameterfile param.m der MatLab-Simulationen, aus welchem die Zusammenhänge und die verwendeten Werte ersichtlich sind.

Das folgende Bild zeigt den Spannungsverlauf bei einer sprungförmigen Auslenkung des Inklinometers. Es wurde mit den gefundenen Parameterwerten in MatLab simuliert.



Simulation Inklinometer

Das Modell des Inklinometers muss mit Vorsicht aufgenommen werden, da mit einer linearen Modellierung das gemessene Überschwingen gar nicht erreicht werden kann. Eventuell wurden hier Fehler im Messaufbau gemacht. Es wäre sicher lohnenswert, die Identifikation des Inklinometer etwas zu verfeinern.

6.3. Modellierung des Servos

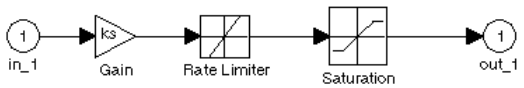
Die Servos für die Lenkung und für die Gewichtsverlagerung sind identisch. Es wurden zwei ECO-Servos vom Typ C512 [Gra] der Firma Graupner zugekauft. Die Servos haben im dynamischen Betrieb zwei Einschränkungen.

Leider war es nicht möglich, eine Messung des dynamischen Verhaltens des Servos aufzunehmen. Probleme bereiten vor allem die Messung der mechanischen Bewegung des Servos. Darum beruhen folgende Aussagen auf Herstellerangaben aus dem Katalog [Gra] oder Vermutungen.

Es wird angenommen, dass der Servo keine Totzeit hat. Da das maximale Stellmoment 0.3 Nm beträgt, wird die Dynamik des Vorderrades vernachlässigt. Es werden keine linearen dynamischen Verzögerungen in die Modellierung einbezogen.

Als dynamische Angabe steht im Katalog eine minimale Stellzeit von 0,15 Sekunden für 40°, was einer maximalen Geschwindigkeit von 260° pro Sekunde entspricht. Die Stellgeschwindigkeit des Servos ist dadurch begrenzt. Zudem lässt die Geometrie des Motorrades nur Lenkwinkel zwischen -18° und +18° zu.

Aus diesen Angaben entsteht folgendes nichtlineare Modell:



Simulink-Modell des Servos

Um das Gesamtmodell mit dieser Nichtlinearität durch einen linearen Regler zu stabilisieren, muss darauf geachtet werden, dass die Beschränkungen nicht überschritten werden.

 © Copyright 1998 SA

6.4. Bestimmung der Parameter

In diesem Teil sind sämtliche Parameter des Motorrades, welche zur Modellierung verwendet wurden, nochmals aufgeführt. Bei jedem Parameter ist der Wert aufgeführt, welcher für die Simulationen und den Regler verwendet wurde. Anschliessend sind für die Parameter, bei welchen es nicht ersichtlich ist, wie sie gemessen wurden, die Messverfahren erklärt.

Geometrie des Motorrades

Radstand aus Katalog [Gra]	$l = 0.29$ m
Radius des Hinterrades aus Katalog [Gra]	$r_h = 0.064$ m
Radius des Vorderrades aus Katalog [Gra]	$r_v = 0.059$ m
Höhe des Schwerpunktes über Boden	$h = 0.114$ m
maximaler Lenkwinkel	$\gamma_{max} = 18°$

Motorrad

Masse des gesamten Motorrades	$m = 2.200$ kg
Geschwindigkeit des Motorrades	$v = 3$ m/s
Dämpfungskoeffizient des Motorrades	$d = 0.01$

Trägheitsmomente

Trägheitsmoment des Vorderrades	$J_v = 3.78 \cdot 10^{-4}$ kg m^2
Trägheitsmoment des Hinterrades	$J_h = 4.81 \cdot 10^{-4}$ kg m^2
Trägheitsmoment des Motorrades um die Kippachse	$J = 3.89 \cdot 10^{-2}$ kg m^2

Inklinometer

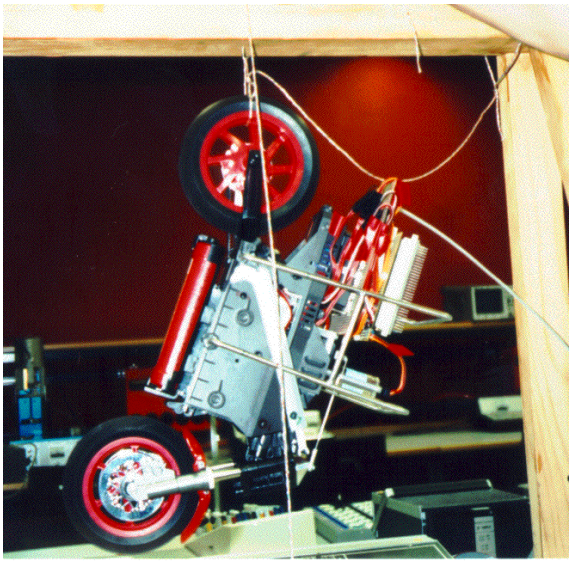
Schwingfrequenz des Inklinometers	$\omega = 140$ rad/s
Dämpfung des Inklinometers	$d = 0.25$
Verstärkung des Inklinometers	$k_{ink} = -1.156$ V/rad

Servo

max. Geschwindigkeit des Servos aus Katalog [Gra]	$\omega_{smax} = 4.65$ rad/s

<u>Messung Schwerpunkt</u>

Der Schwerpunkt wird gefunden, indem das Motorrad am Vorderrad aufgehängt wird und die Senkrechte durch den Aufhängepunkt auf das Motorrad gezeichnet wird. Dasselbe wird mit dem Hinterrad als Aufhängepunkt gemacht. Der Schnittpunkt der beiden Linien ist der Ort des Schwerpunktes. Um keine Fehler bei der Messung zu begehen, wird der Schwerpunkt auf beiden Seiten des Motorrades eingezeichnet und deren Position verglichen.



Messung Schwerpunkt

Messung Trägheitsmomente der Räder

Die Trägheitsmomente der Räder werden mit einem Schwingversuch gemessen. Dabei wird ein Gewicht von 50 Gramm an ein Rad montiert. Das Motorrad wird aufgebockt, so dass das Rad sich frei drehen kann. Das Rad wird bis maximal 45° ausgelenkt und die Dauer von ein paar Perioden gemessen. Die Anzahl der Perioden hängt mit der Dämpfung der Schwingung zusammen. Je stärker die Dämpfung, desto weniger Perioden können erfasst werden. Die Periodendauer T und das Gesamtträgheitsmoment J_G (Rad mit Gewicht) hängen wie folgt zusammen:

$$\frac{2\pi}{T} = \sqrt{\frac{m_G r_G g}{J_G}}$$

wobei m_G die Masse des Gewichts und r_G Abstand des Gewichts von der Drehachse ist.

Um auf das Trägheitsmoment J des Rades zu kommen, muss vom Gesamtträgheitsmoment J_G noch das Trägheitsmoment des Gewichts abgezogen werden.

$$J = J_G - m_G r_G^2 = \frac{T^2}{4\pi^2} m_G r_G g - m_G r_G^2$$

Für jedes Rad wurden 6 Messungen aufgenommen. Die grösste und die kleinste Messung wurden gestrichen, und der Mittelwert aus den 4 verbleibenden Messungen wurde als Messwert verwendet.

Als Verfikation wird das Resultat mit berechneten Trägheitsmomenten verglichen. Aus der Geometrie des Rades können maximale und minimale Schranken für das Trägheitsmoment berechnet werden. Das wirkliche Trägheitsmoment ist mindestens so gross wie das Trägheitsmoment eines Zylinders mit dem Radradius r (Minimum) und maximal so gross wie das Trägheitsmoment eines Hohlzylinders mit dem Radradius r als Aussenradius und dem Felgenradius r_f als Innenradius.

$$\frac{1}{2} m r^2 < J < \frac{1}{2} m \left(r^2 + r_f^2\right)$$

wobei m die Masse des Rades ist.

Bei beiden Rädern lag die Messung zwischen dem Minimal- und dem Maximalwert. Die gemessenen Trägheitsmomente können somit als glaubwürdig erachtet werden.

Messung Trägheitsmoment des Motorrades

Das Trägheitsmoment des Motorrades wird mit einem Schwingversuch gemessen. Das Modell schwingt um die Achse durch den Schwerpunkt, welche parallel zur Kippachse liegt. Mit der gemessenen Periodendauer T kann das Trägheitsmoment berechnet werden. Im folgenden wird erklärt, wie der Schwingversuch bewerkstelligt wurde.

Das Motorrad wird mit Gummizügen in einen Rahmen gespannt. Die oberen Gummizüge werden unterhalb des Gabelrohrs eingeschlauft und das Modell am oberen Hacken eingehängt. Das Modell wird einige Male um die eigene Achse gedreht, damit die oberen Gummizüge angespannt sind. Die unteren Gummizüge werden am hinteren Ende des Rahmens befestigt und im unteren Haken eingehängt. Durch die Anspannung der oberen Gummis dreht sich das Modell in eine Gleichgewichtslage zwischen den Drehmomenten der oberen und der unteren Gummizüge. Bei der Befestigung muss darauf geachtet werden, dass alle Befestigungspunkte der Gummizüge auf einer Geraden durch den Schwerpunkt liegen und die Gerade parallel zur Kippachse verläuft. Das Motorrad wird nun um den Winkel α_f aus der Gleichgewichtslage ausgelenkt. Dabei muss beachtet werden, dass man nicht zu schwach aufzieht, da dann aufgrund der Dämpfung zuwenig Perioden gemessen werden können. Bei einer zu starken Auslenkung können aber auch Probleme auftreten. Wenn die Drehachse nicht genau durch den Schwerpunkt verläuft, schwingt das Motorrad während der Messung nicht nur um die Drehachse, sondern auch im Rahmen hin und her.

In der Ausgangsposition wird die Grösse der Federkonstanten mit einer Federwaage bestimmt. Im Abstand a_f von der Drehachse wird eine Federwaage angehängt und die angreifende Kraft F_f gemessen. Die Federkonstante c_f ergibt sich aus folgender Rechnung:

 © Copyright 1998 SA

$$c_f = \frac{F_f \cdot a_f}{\alpha_f}$$

wobei α_f der Winkel ist mit dem das Motorrad aufgezogen ist.

Nun wird die Dauer von einigen Perioden gemessen. Die Anzahl der Perioden hängt mit der Dämpfung der Schwingung zusammen. Je stärker die Dämpfung, desto weniger Perioden können erfasst werden. Die Periodendauer T und das Trägheitsmoment J_s durch die Schwerpunktsachse parallel zum Boden hängen wie folgt zusammen:

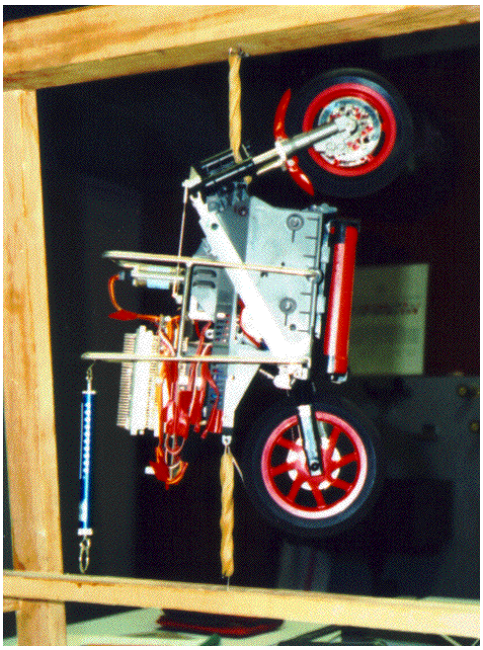
$$\frac{2\pi}{T} = \sqrt{\frac{c_f}{J_s}}$$

Um auf das Trägheitsmoment J des Motorrades um die Kippachse zu kommen, muss zum Trägheitsmoment J_s noch die Verschiebung um die Höhe h aus der Schwerpunktsachse dazugezählt werden. Nach dem Satz von Steiner beträgt das Trägheitsmoment J:

$$J = J_s + mh^2 = \frac{T^2}{4\pi^2} c_f + mh^2$$

wobei m die Gesamtmasse des Motorrades und h die Höhe des Schwerpunktes ist.

Es wurden 6 Messungen aufgenommen. Die grösste und die kleinste Messung wurden gestrichen, und der Mittelwert aus den 4 verbleibenden Messungen wurde als Messwert verwendet.



Messung Trägheitsmoment Motorrad

Als Verfikation wird das Resultat mit berechneten Trägheitsmomenten verglichen. Aus der Geometrie des Motorrades kann folgendes geschlossen; das wirkliche Trägheitsmoment ist mindestens so gross wie das Trägheitsmoment eines Zylinders mit dem Radius der halben Schwerpunktshöhe h (Minimum) und maximal so gross wie das Trägheitsmoment eines Zylinders mit dem Radius der Schwerpunktshöhe h,

$$\frac{1}{2} m \frac{h^2}{4} < J < \frac{1}{2} mh^2 ,$$

wobei m die Gesamtmasse des Motorrades ist.

Die Messung lag zwischen dem Minimal- und dem Maximalwert. Die gemessenen Trägheitsmomente können somit als glaubwürdig erachtet werden.

6.5. Abschätzung des Modellierungsfehlers

Wie gross sind die Fehler, welche durch die Modellierung des Motorrades als Pendel eingeführt wurden? Kann die Grösse des Fehlers überhaupt abgeschätzt werden? Warum kann mit einem Zweirad überhaupt freihändig gefahren werden?

Um das Verhalten eines Zweirades besser zu verstehen, sind im folgenden einige Grafiken aufgezeichnet. Danach wird anhand dieser Grafiken erklärt, warum sich ein Zweirad bei genügender Geschwindigkeit stabilisiert. Die Erklärungen beruhen auf der Anschauung und nicht auf mathematischen Beweisen, sind also nur qualitativ.

Durch die Modellierung als Pendel wurden in erster Linie die Kräfte zwischen Rahmen und Gabel vernachlässigt. Eine Abschätzung, wie gross dieser Fehler ist, ist äusserst schwierig. Was aber berechnet werden kann, ist die Bewegung des Auflagepunktes des Vorderrades gegenüber dem Rahmen. Durch diese Bewegung des Auflagepunktes ändert sich der Radstand, was das Verhalten des Zweirades beeinflusst. Die Berechnungen lassen eine ungefähre Abschätzung zu, wie gross der Fehler beim Pendel ist, welches nur einen Auflagepunkt hat. Als Nebenprodukt dieser Berechnungen kann gezeigt werden, warum es möglich ist, mit einem Zweirad freihändig zu fahren.

6.5.1. Auflagepunkt des Vorderrades

Zuerst werden zwei Koordinatensysteme eingeführt. Um auf das Verhalten des Zweirades in stabiler Fahrt schliessen zu können, wird ein Koordinatensystem D eingeführt, welches nach der resultierenden Kraft ausgerichtet ist. Das Koordinatensystem D hat die Koordinaten d, e, f. Aus den Grafiken, welche in diesem Koordinatensystem dargestellt werden, kann auf das qualitative Verhalten des Fahrrades geschlossen werden. Die Berechnungen können aber wesentlich einfacher in einem zweiten Koordinatensystem X durchgeführt werden. Das Koordinatensystem X ist an das Problem angepasst und hat die Koordinaten x, y, z.

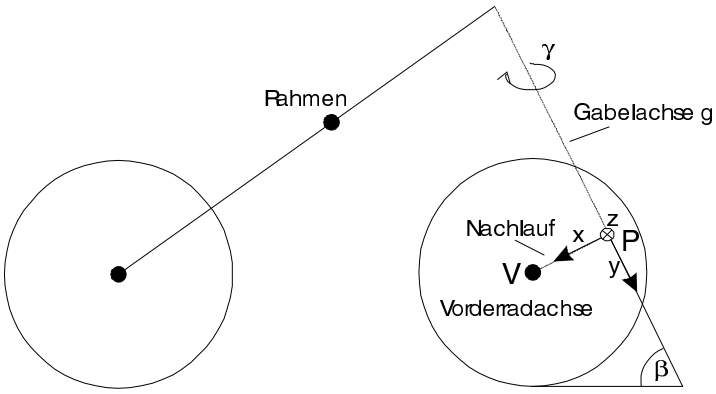
<u>Koordinatensystem X</u>

Das Koordinatensystem X ist körperfest bezüglich des Rahmens, das heisst, dass das Koordinatensystem immer im selben Abstand und gleich ausgerichtet ist wie der Rahmen. Der Ursprung P des Koordinatensystems liegt auf der Gabelachse g, das ist die verlängerte Drehachse, um welche sich die Gabel dreht. Der Ursprung P liegt im Schnittpunkt mit der Senkrechten zur Vorderradachse V.

Die Koordinate x steht senkrecht auf der Gabelachse g und zeigt zum Rahmen. Die Koordinate z liegt auf der Gabelachse g und zeigt zum Boden. Die x/z-Ebene entspricht der Ebene, welche durch die Gabelachse g und den Rahmen aufgespannt wird. Die Koordinate y steht senkrecht auf dieser Ebene und zeigt nach links, wenn man das Motorrad von hinten betrachtet. Die Koordinaten bilden ein Rechtsystem.

Der Winkel γ misst die Drehung der Gabel. Die Drehachse ist die Gabelachse g, also die z-Koordinate, und die Drehrichtung ist von x nach y, also mathematisch postitiv. Der Winkel α misst die Neigung des Zweirades. Drehachse k ist die Richtung des Hinterrades am Boden, welche der Kippachse entspricht. In vertikaler Position des Zweirades ist $\alpha = 0$. Der Winkel α nimmt zu, wenn das Zweirad nach rechts kippt. Der Winkel β ist die Neigung der Gabelachse g gegenüber dem Boden. Er wird als konstant angenommen.

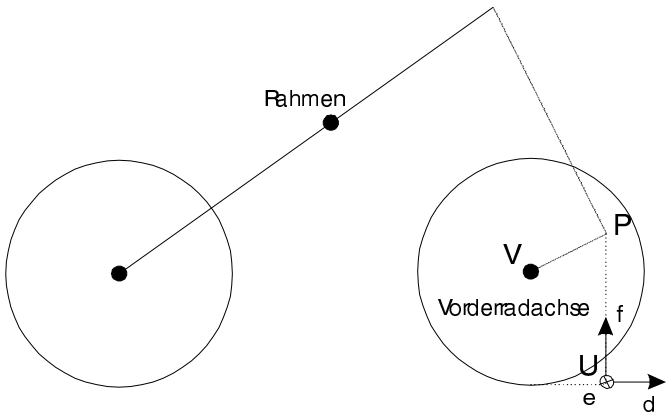
 © Copyright 1998 SA



Zweirad mit Koordinatensystem X

Koordinatensystem D

Der Urspung U des Koordinatensystems D liegt auf der Kippachse k, welche die Verlängerung der Fahrtrichtung des Hinterrades ist. Der Urspung U liegt im Schnittpunkt mit der Senkrechten zum Koordinatenursprung P. Steht das Motorrad in vertikaler Lage, liegt U genau unter P auf dem Boden. Die Koordinate d liegt auf der Kippachse k und zeigt nach vorne. Die Koordinate e steht senkrecht auf der Kippachse k und senkrecht auf der Ebene, welche durch den Rahmen und die Gabelachse g aufgespannt wird, also der x/z-Ebene. Die Koordinate e zeigt nach links, wenn man das Motorrad von hinten betrachtet. Die Koordinate f liegt in der x/z-Ebene und zeigt nach oben. Die Koordinaten bilden ein Rechtssystem.



Zweirad mit Koordinatensystem D

Bezug zwischen X und D

Der Bezug zwischen den Koordinatensystemen X und D wird durch eine Koordinatentransformation hergestellt. Zu jeder Koordinate im System D wird erklärt, was ihr Wert bedeutet.

Die Koordinate d gibt an, um wieviel der Auflagepunkt A vor dem Koordinatenurspung U liegt.

$$d = z\cos(\beta) - x\sin(\beta)$$

Die Koordinate e gibt an, um wieviel der Auflagepunkt A neben der Kippachse k liegt.

$$e = y$$

Die Koordinate f gibt die Distanz zwischen den Koordinatenursprüngen U und P an.

$$f = x\cos(\beta) + z\sin(\beta) - y\tan(\alpha)$$

Die Koordinate f liegt parallel zur resultierenden Kraft, welche das Zweirad in einer stabilen Lage hält. Darum wird dieser Wert für die weiteren Betrachtungen eine entscheidende Rolle spielen.

Betrachtung bei vertikalem Zweirad $\alpha = 0$

Um den Bezug zum Modellmotorrad zu haben, werden folgende Parameter ausgemessen:

Nachlauf	l = 2.25 cm
Neigung der Gabelachse	$\beta = 60°$
Radius des Vorderrades	r = 5.9 cm

Diese Parameter werden im folgenden benötigt, um den Ort des Auflagepunktes A zu berechnen und in Grafiken darzustellen.

Ort der Vorderradachse V bezüglich dem Koordinatenursprung P

Die Vorderradachse V liegt in der x/y-Ebene auf einem Kreis mit Radius l, wobei l der Abstand der Vorderradachse von der Gabelachse ist, das heisst der Nachlauf des Zweirades. Der Nachlauf zeigt in x Richtung. Daraus ergeben sich die für die einzelnen Koordinaten folgende Werte:

$$x_l = l\cos(\gamma); \qquad y_l = l\sin(\gamma); \qquad z_l = 0;$$

Ort des Auflagepunktes A bezüglich der Vorderradachse V

Um den Ort des Auflagepunktes A bezüglich der Vorderradachse V zu bestimmen, wird das Koordinatensystem X in den Punkt V verschoben. Die Ausrichtung bleibt dieselbe wie zuvor. Es handelt sich nur um eine Translation des Koordinatensystems X.

Der Auflagepunkt A liegt in der Ebene, welche durch die Gabeldrehachse g und den Nachlauf l aufgespannt wird. Die Gleichung dieser Ebene lautet:

$$y = x\tan(\gamma)$$

Weiter liegt der Auflagepunkt A auch in der Ebene, welche durch die Radachse, die senkrecht zum Nachlauf l und der Gabelachse steht, und der Senkrechten zum Boden aufgespannt wird. Die Gleichung dieser Ebene lautet:

$$x = -y\tan(\gamma) + \frac{z}{\tan(\beta)}$$

Drittens liegt der Auflagepunkt A auf der Kugeloberfläche um V mit dem Vorderradradius r. Die Kugelgleichung lautet:

$$x^2 + y^2 + z^2 = r^2$$

Diese drei Gleichungen werden nach den drei Variablen aufgelöst. Mit Hilfe von Maple erhält man Funktionen in Abhängigkeit von γ.

$$x_v(\gamma); \qquad y_v(\gamma); \qquad z_v(\gamma);$$

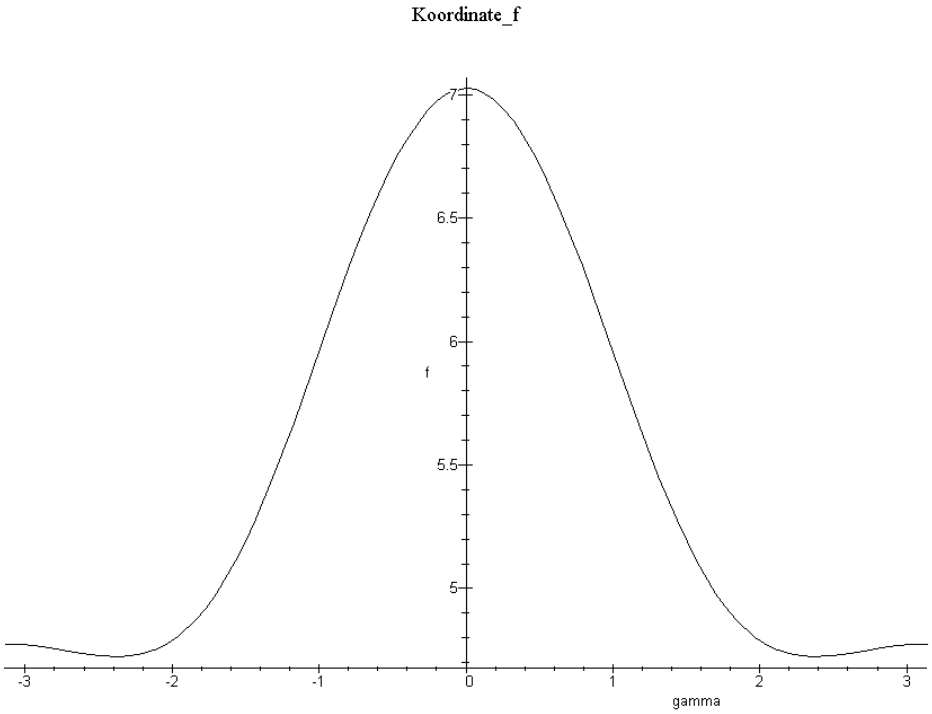
Ort des Auflagepuntes A im Koordinatensystem D

Um die genaue Position des Auflagepuntes A zu erhalten, müssen die Distanzen noch addiert werden.

$$x(\gamma) = x_l + x_v; \qquad y(\gamma) = y_l + y_v; \qquad z(\gamma) = z_l + z_v;$$

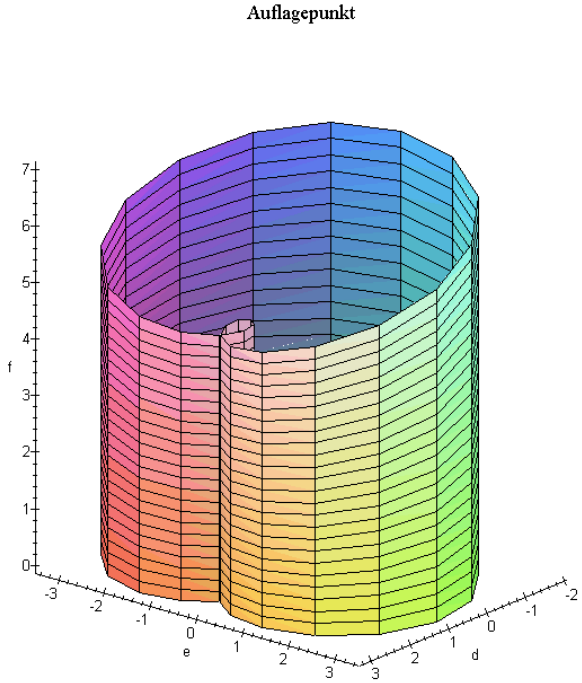
Aus dem Bezug zwischen X und D kann nun der Wert der Koordinaten d, e und f berechnet und zur Veranschaulichung in verschiedenen Grafiken dargestellt werden. In der folgenden Grafik sieht man, wie sich die Koordinate f verändert, wenn die Gabel um 360° gedreht wird.

 © Copyright 1998 SA



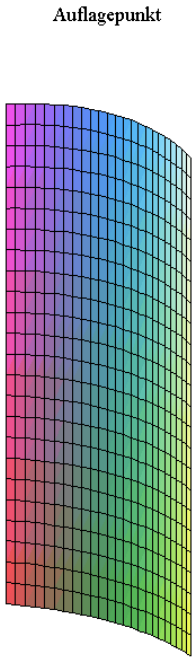
Grafik 1: Koordinate f abhängig vom Winkel γ

Die folgende Grafik zeigt den Ort des Auflagepunktes A im Koordinatensystem D bei einer Drehung der Gabel um 360°. Über dem Auflagepunkt ist der Wert der Koordinate f aufgezeichnet.



Grafik 2: Ort von A bei vertikalem Zweirad

Die folgende Grafik zeigt den Ausschnitt aus obiger Grafik, welcher mit der beschränkten Lenkung des Motorrades erreicht werden kann. Die Variable γ bewegt sich also nur zwischen -18° und +18°.



Grafik 3: Ort von A bei vertikalem Zweirad und Lenkanschlägen

Ein Kommentar zu obigen Grafiken folgt nach der Betrachtung des schiefen Zweirades.

Betrachtung bei schiefem Zweirad $\alpha = 10°$

Ort der Vorderradachse V bezüglich dem Koordinatenursprung P

Der Ort der Vorderradachse V bezüglich dem Koordinatenursprung P ändert sich durch das Kippen des Zweirades nicht.

Ort des Auflagepunktes A bezüglich der Vorderradachse V

Wie oben wird das Koordinatensystem in den Punkt V verschoben. Das Koordinatensystem bleibt aber immer noch nach dem Rahmen ausgerichtet, das heisst, dass die Verschiebung nur eine Translation ist.

Der Auflagepunkt A liegt auf der Ebene, welche durch die Gabeldrehachse und den Nachlauf l aufgespannt wird. Die Gleichung dieser Ebene lautet:

$$y = x\tan(\gamma)$$

Weiter liegt der Auflagepunkt A auch in der Ebene, welche durch die Radachse, die senkrecht zum Nachlauf l und der Gabelachse steht, und der Senkrechten zum Boden aufgespannt wird. Die Gleichung dieser Ebene lautet:

$$x = -y\tan(\gamma) + z\left(\frac{1}{\tan(\beta)} - \frac{\tan(\alpha)\tan(\gamma)}{\sin(\beta)}\right)$$

Drittens liegt der Auflagepunkt A auf der Kugeloberfläche um V mit dem Vorderradradius r. Die Kugelgleichung lautet:

$$x^2 + y^2 + z^2 = r^2$$

Diese drei Gleichungen werden nach den drei Variablen aufgelöst. Mit Hilfe von Maple erhält man Funktionen in Abhängigkeit von γ.

$$x_v(\gamma); \qquad y_v(\gamma); \qquad z_v(\gamma);$$

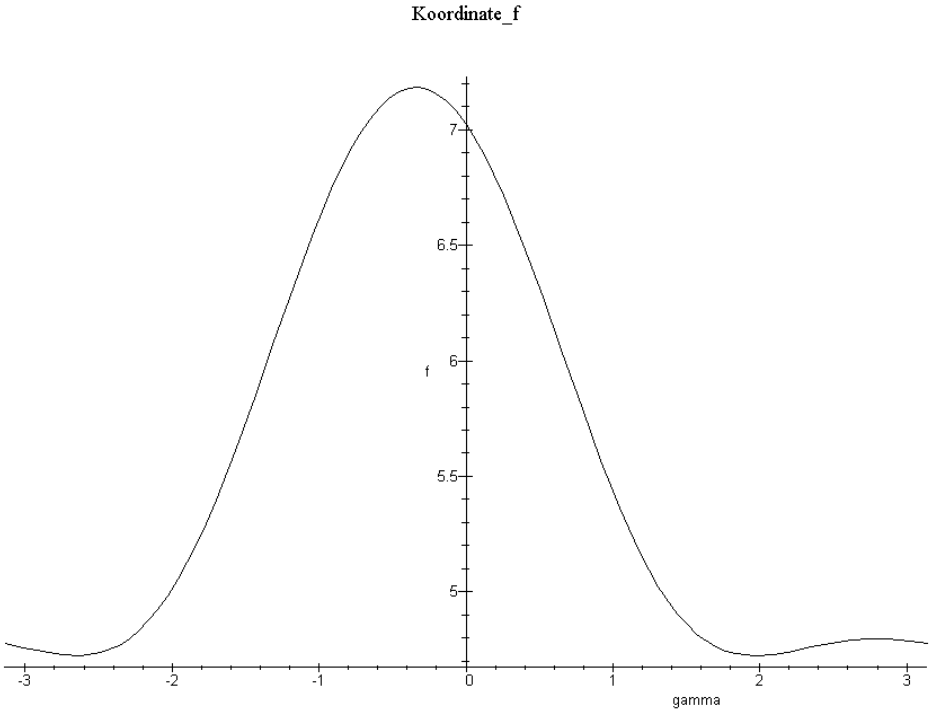
 © Copyright 1998 SA

Ort des Auflagepuntes A im Koordinatensystem D

Um die genaue Position des Auflagepuntes A zu erhalten müssen die Distanzen noch addiert werden.

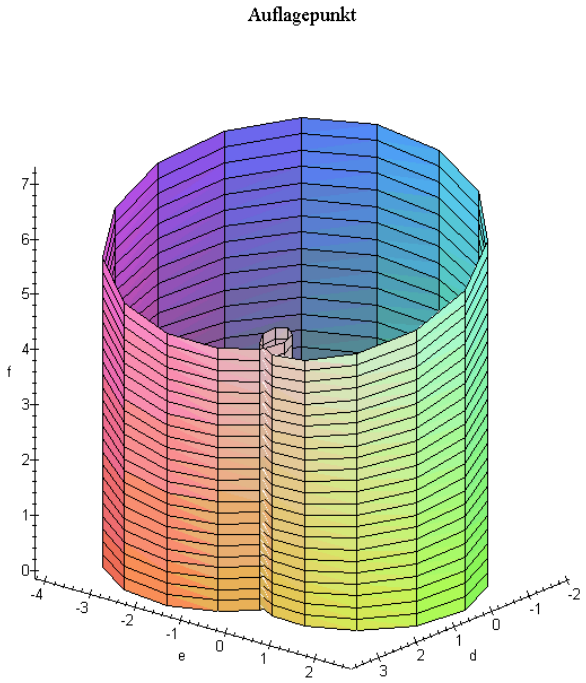
$$x(\gamma) = x_l + x_v; \qquad y(\gamma) = y_l + y_v; \qquad z(\gamma) = z_l + z_v;$$

Aus der Beziehung zwischen X und D kann nun der Wert der Koordinaten d, e und f berechnet und zur Veranschaulichung in verschiedenen Grafiken dargestellt werden. In der folgenden Grafik sieht man, wie sich die Koordinate f verändert, wenn die Gabel um 360° gedreht wird bei einer Schräglage α von 10°.



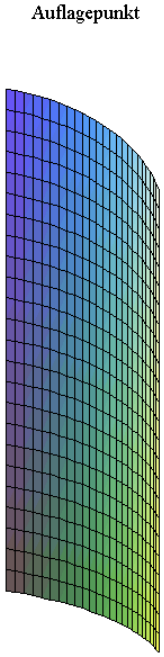
Grafik 4: Koordinate f abhängig vom Winkel γ / α = 10°

Die folgende Grafik zeigt den Ort des Auflagepunktes A bei einer Drehung der Gabel um 360° und einer Schräglage α = 10°. Über dem Auflagepunkt ist der Wert der Koordinate f aufgezeichnet.



Grafik 5: Koordinate f abhängig vom Winkel γ / α = 10°

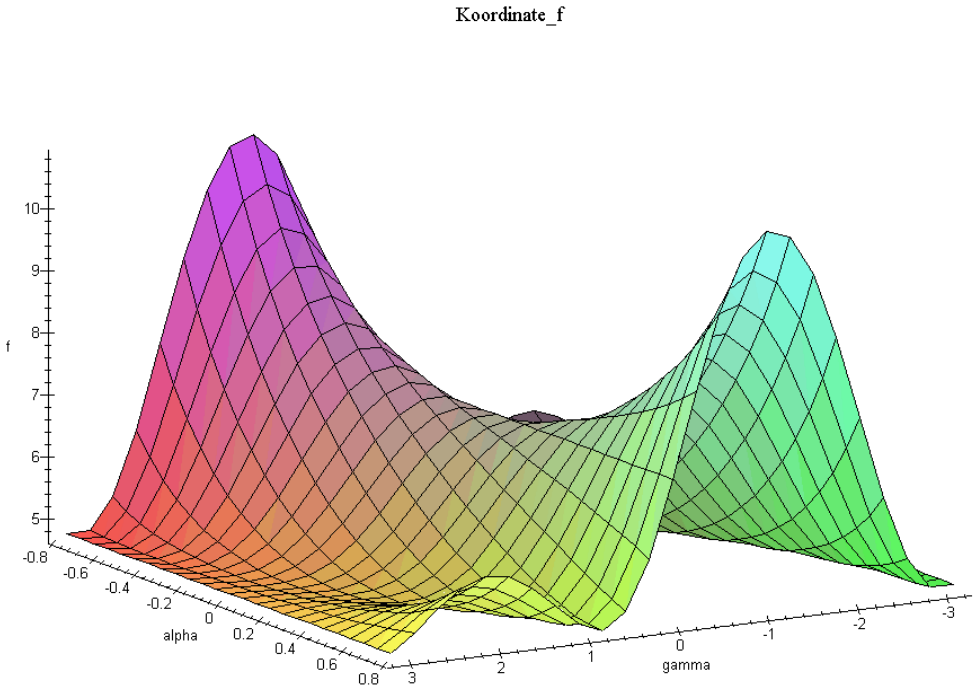
Die folgende Grafik zeigt den Ausschnitt aus obiger Grafik, welcher mit der beschränkten Lenkung des Motorrades erreicht werden kann. Die Variable γ bewegt sich also nur zwischen -18° und +18°.



Grafik 6: Koordinate f abhängig vom Winkel γ / α = 10°

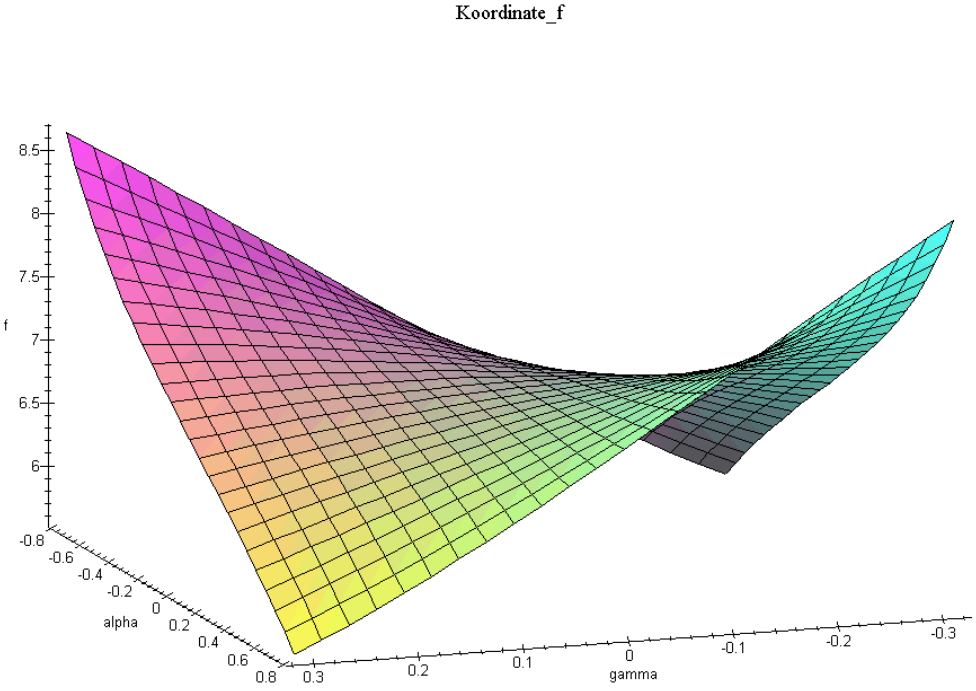
In der folgenden Grafik sieht man, wie sich die Koordinate f verändert, wenn die Lenkung um 360° gedreht wird und die Schräglage zwischen -45° und +45° beträgt.

 © Copyright 1998 SA



Grafik 7: Koordinate f abhängig von den Winkeln γ und α

Die folgende Grafik zeigt den Ausschnitt aus obiger Grafik, welcher mit der beschränkten Lenkung des Motorrades erreicht werden kann. Die Variable γ bewegt sich also nur zwischen -18° und +18°.



Grafik 8: Koordinate f abhängig von den Winkeln γ und α

<u>Kommentar zu den Grafiken</u>

Die Koordinate f zeigt in entgegengesetzter Richtung der resultierenden Kraft bei stabiler Fahrt. Betrachtet man die resultierende Kraft als Potentialkraft, so versucht das Zweirad in eine Position mit weniger Höhe einzulaufen. Geht man von einer stabiler Geradeausfahrt aus und lässt das Zweirad langsam kippen, das heisst, dass α grösser wird, kann man aus den Grafiken sieben und

acht herauslesen, dass bei $\gamma = 0$ die Lenkung auf eine schiefe Ebene hinausläuft. In der Folge nimmt γ zu, das heisst, dass das Vorderrad einlenkt. Durch dieses Einlenken wird das Zweirad wieder in die aufrechte Position zurückgebracht. Die Dämpfung dieser Bewegungen ist bei genügender Geschwindigkeit so gross, dass das Motorrad in eine stabile Geradeausfahrt einschwingt.

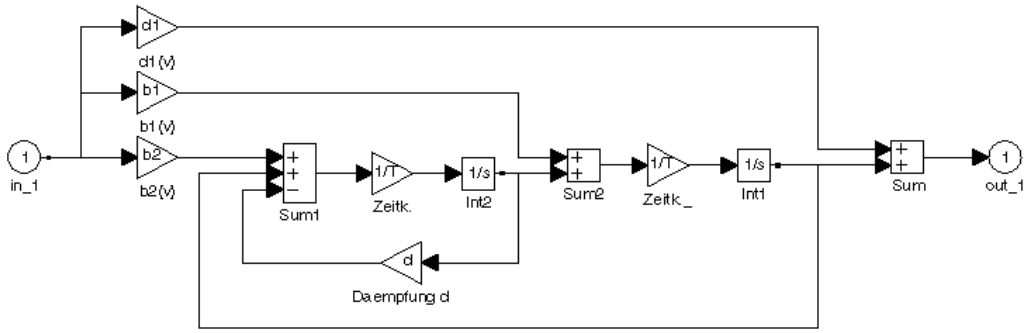
Hat das Zweirad einen Vorlauf, wie normalerweise einem Fahrrad, dann spielt sich dasselbe in der kleinen Schleife ab, welche im Vordergrund der Grafiken zwei und fünf sichtbar ist.

Aus den Grafiken drei, sechs und acht ist ersichtlich, dass die Änderungen des Auflagepunktes bei eingeschränkter Lenkung relativ klein sind und das Verhalten des Modells kaum beeinflussen. Die Vereinfachungen, welche gemacht wurden, können mit dieser Feststellung begründet werden.

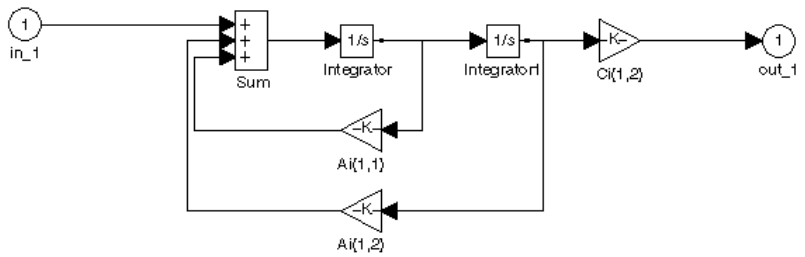
6.6. Gesamtmodell der Strecke

Sämtliche Angaben aus obigen Abschnitten können zu einem Simulink-Modell des gesamten Zweirades zusammengestellt werden. Aufgrund dieses Modells wird im folgenden Kapitel ein Reglerkonzept entworfen, welcher im Motorrad zum Einsatz kommt.

Das linearisierte Gesamtmodell des Zweirades ist vierter Ordnung und kann aus folgenden Simulink-Modellen zusammengestellt werden:



linearisiertes Modell des Zweirades



Modell des Inklinometers

 © Copyright 1998 SA

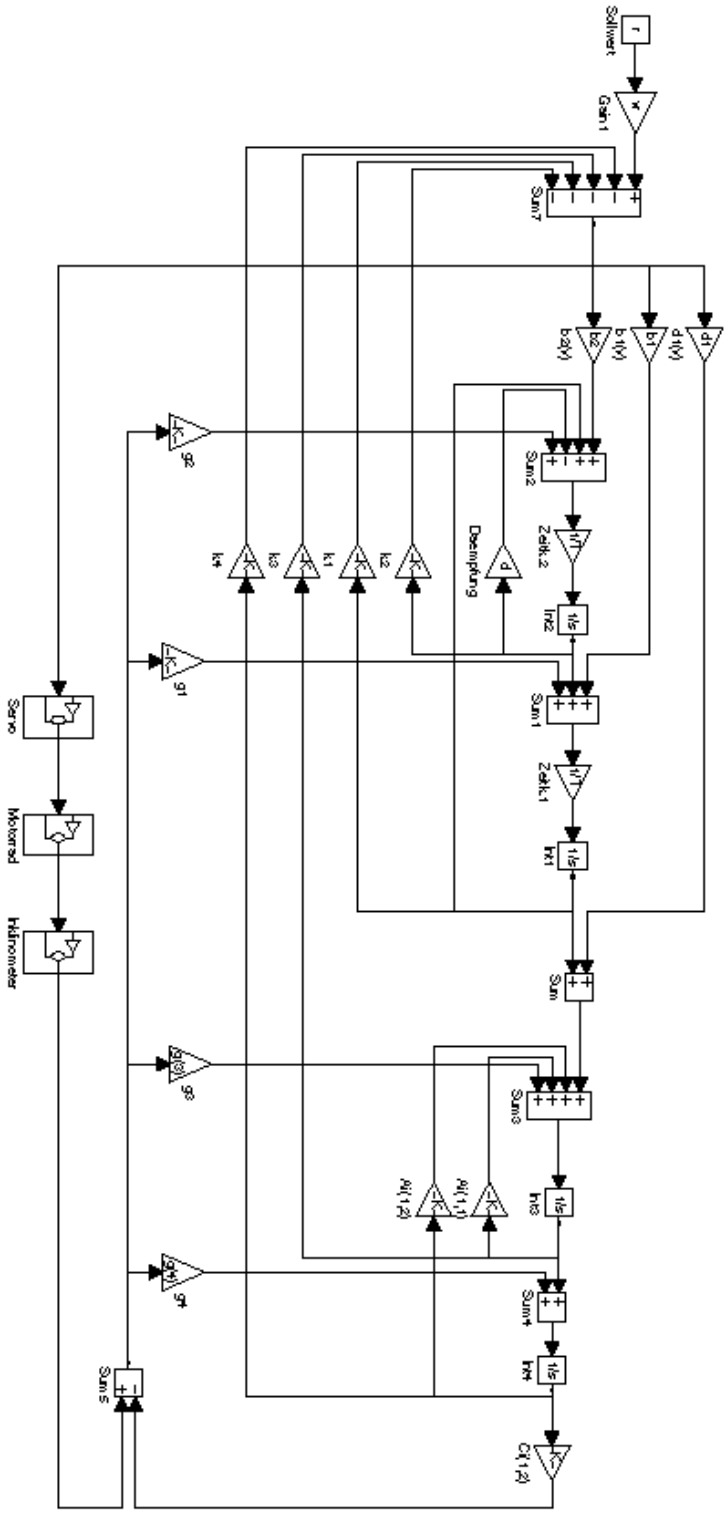
7. Regler

Auf MatLab wurde für das Gesamtmodell aus dem vorherigen Kapitel ein Regler entworfen.

7.1. Regelkonzept

Die Strecke des Motorrades ist vierter Ordnung und instabil. Zwei Pole liegen symmetrisch zur imaginären Achse in der linken und rechten Halbebene der komplexen Ebene. Die zwei weiteren Pole des Inklinometers liegen symmetrisch zur reellen Achse in der linken Halbebene. Die Pole des Inklinometers wurden durch die Sprungantwort identifiziert. Leider ist die Identifikation dieser Pole ziemlich unsicher. Abweichungen zur Realität sind zu erwarten. Die ganze Strecke wurde linearisiert. Das Modell stimmt also nur für einen beschränkten Bereich.

Wegen dieser Gründe würden alle einfachen Konzepte scheitern, welche auf einem PID-Regler mit einer Ausgangsrückführung beruhen. Um das System 4. Ordnung zu stabilisieren, muss ein Beobachter eingeführt werden. Damit kann das System mit einem Zustandsregler stabilisiert werden. Im folgenden Simulink-Bild ist der Regler mit Beobachter parallel zur Strecke eingeführt worden:



Beobachter mit Zustandsregler

 © Copyright 1998 SA

7.2. Polfestlegung

Im folgenden werden die Pole für die Rückführfaktoren der Fehlerrückführung und der Zustandsrückführung festgelegt. Die Pole wurden für den ersten Versuch von Hand festgelegt. Mit dieser Methode konnten spezielle Eigenschaften der Strecke berücksichtigt werden. Die Zustandsrückführfaktoren wurden so festgelegt, dass die Faktoren der Inklinometerzustände null sind. Eine nachfolgende Optimierung würde sicher noch bessere Ergebnisse liefern.

Zur Polfestlegung ist auch das MatLab-File param.m im Angang zu betrachten.

Um bereits vor der vollständigen Modellierung mit ersten Regelungskonzepten zu experimentieren, wurden die Pole allgemein festgelegt. Als Basis für die Polfestlegung wurden die Pole und Nullstellen aus der Strecke so zu einem Grundvektor zusammengesetzt, so dass ein stabilisierendes Verhalten erwartet werden kann.

Dieser Grundvektor setzt sich wie folgt zusammen:

erster Pol	negative Nullstelle der Strecke z(1)
zweiter Pol	negative Polstelle des Motorrades p(2)
dritter und vierter Pol	negative Polstellen des Inklinometers p(3) und p(4)

7.2.1. Fehlerrückführung

Für die Fehlerrückführung wird der Realteil des Grundvektors mit einem Faktor f_g multipliziert. Der Faktor kann als einfacher Designparameter für die Polfestlegung verwendet weden. Die Pole der Fehlerrückführung sind also wie folgt festgelegt (in MatLab Schreibweise):

$$p_g = f_g \cdot real([z(1) \quad p(2) \quad p(4) \quad p(4)]) + i \cdot imag([z(1) \quad p(2) \quad p(4) \quad p(4)]);$$

Der Faktor f_g hat den Wert vier. Die Fehlerrückführung ist damit um den Faktor vier schneller als die Strecke.

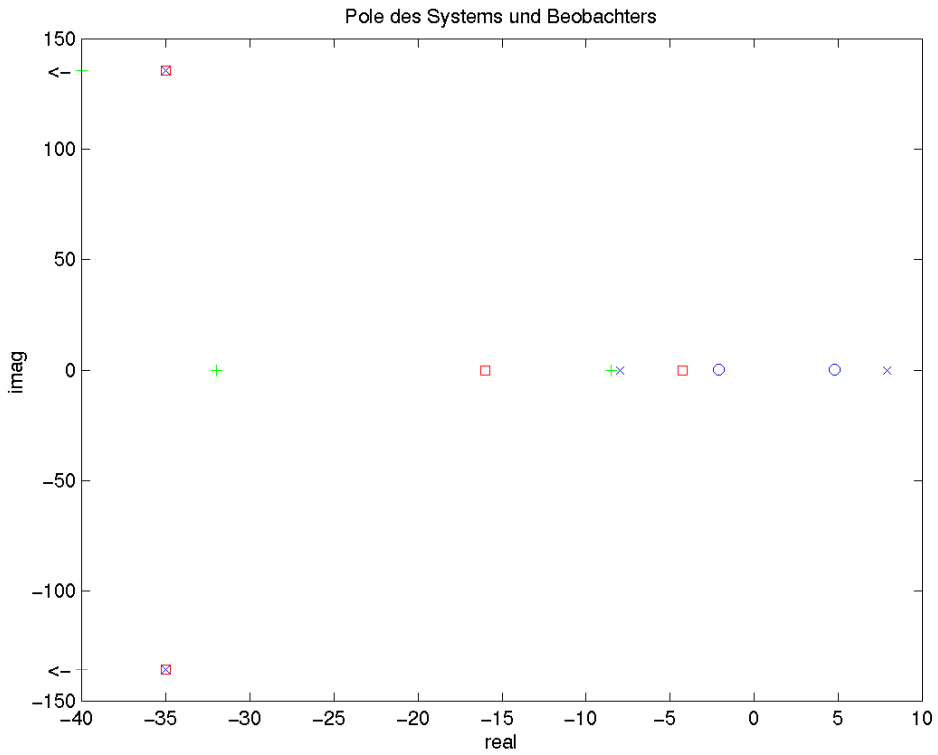
7.2.2. Zustandsrückführung

Die Pole für die Zustandsrückführung werden festgelegt, indem nur die langsamen Pole, dass heisst die ersten beiden (z(1) und p(2)), mit einem Faktor f_k multipliziert werden. Es werden nur die langsamen Pole mit einem Faktor multipliziert, weil damit die Rückführfaktoren der Inklinometerzustände beinahe null werden. Sie werden im Regler gleich null gesetzt. Die Pole der Zustandsrückführung sind also wie folgt festgelegt (in MatLab Schreibweise):

$$p_k = [f_k \cdot real([z(1) \quad p(2)]) \quad p(4) \quad p(4)] + i \cdot imag([z(1) \quad p(2) \quad p(4) \quad p(4)]);$$

Der Faktor f_k hat den Wert zwei. Die Zustandsrückführung ist damit doppelt so schnell wie die Strecke.

Folgendes Bild zeigt die Lage der Pole und Nullstellen in der komplexen Ebene:



Pole (System und Rückführung) und Nullstellen (System)

Wobei die Symbole für folgende Nullstellen bzw. Pole stehen

x	Pole der Strecke
o	Nullstellen der Strecke
+	Pole der Fehlerrückführung
□	Pole der Zustandsrückführung

Mit diesen Rückführfaktoren werden in Simulationen gute Resultate erreicht.

Im folgenden Kapitel wird auf die harwaremässige Implementation des Regleralgorithmus eingegangen.

 © Copyright 1998 SA

8. Hardware

Es gibt verschiedene Möglichkeiten, die Hardware und somit die Rechenleistung für die Regelung der Motorradstabiliserung bereitzustellen.

In einem ersten Schritt wurde vorgesehen, den Regleralgorithmus extern auf einem PC mit entsprechender Interfacekarte zu rechnen. Das Motorrad wäre so über eine Nabelschnur geregelt worden. Dies ist eine Möglichkeit, einen bestehenden Regleralgorithmus in einer frühen Phase auszuprobieren, ohne sich gross mit Hardware auseinanderzusetzen.

In diesem Projekt lief jedoch die Synthese des Reglers und der mechanische Umbau des Motorrades simultan ab. Das heisst, das Motorrad stand gar nicht zur Verfügung für frühe Testphasen. Wir entschieden uns deshalb, parallel den Aufbau der Elektronik für einen Onboardrechner voranzutreiben. Um in der nützlichen Frist eine Hardwareumgebung zur Verfügung zu haben, war es unsere Absicht, ein „Evaluation-System“ einzusetzen. Die Entwicklung von Elektronik sollte sich also nur auf wenige Aufgaben beschränken, wie z.B die Bereitstellung der Versorgungsspannung.

In einer zukünftigen Phase des Projekts wäre es sinnvoll, eine exakt auf die Bedürfnisse des Regelungsproblems zugeschnittene Hardware aufzubauen und dies auf einem kompakten Board zu realisieren. Um dies bereits jetzt durchzuführen, war die Zeit jedoch zu knapp.

8.1. Wahl des Rechner

Die Aufgabe der Stabilisierung eines Motorrades stellt verschiedene Anforderungen an einen Rechner. Man muss Signale von den Sensoren einlesen, verarbeiten und darauf die Aktoren ansteuern können. Der Rechner muss all diese Aufgaben in einer nützlichen Frist abarbeiten.

Es bestehen also harte Echtzeitanforderungen. Die Abtastzeit, die der eingesetzte Rechner noch bewältigen kann, sollte deutlich kleiner sein als die Zeitkonstante des zu regelnden Systems. Aus diesem Grund wurde bei der Evaluation des Rechners, darauf geachtet einen Microcontroller einzusetzen, der möglichst viele Funktionen und Schnittstellen schon in Hardware implementiert hat. Je mehr Aufgaben von peripheren Bausteinen übernommen werden können, desto mehr ist die CPU entlastet. Die CPU kann die Hauptrechenzeit dazu benutzen, den Regleralgorithmus auszuführen.

Für diese Aufgabe wird ein Microcontroller MC68332 von Motorola eingesetzt. Die Hardware besteht aus einem Evaluationboard von Motorola, welches über den „Background Mode Port“ durch die Software HiCross programmiert werden kann. Ein Nachteil dieses Controllers ist das Fehlen von Analogeingängen. Es sind keine AD-Wandler auf diesem Chip integriert. Man musste also vorsehen, einen AD-Wandler auf einem Zusatzboard aufzubauen.

8.2. Aufgaben der Hardware

Aus der Problemstellung der Regelungsaufgabe ergeben sich verschiedene Aufgaben, die der Rechner zu bewältigen hat. Im Wesentlichen kann man eine Aufteilung in drei Gruppen machen.

- Einlesen der Eingangssignale
- Regelungsalgorithmus
- Ausgeben der Stellsignale

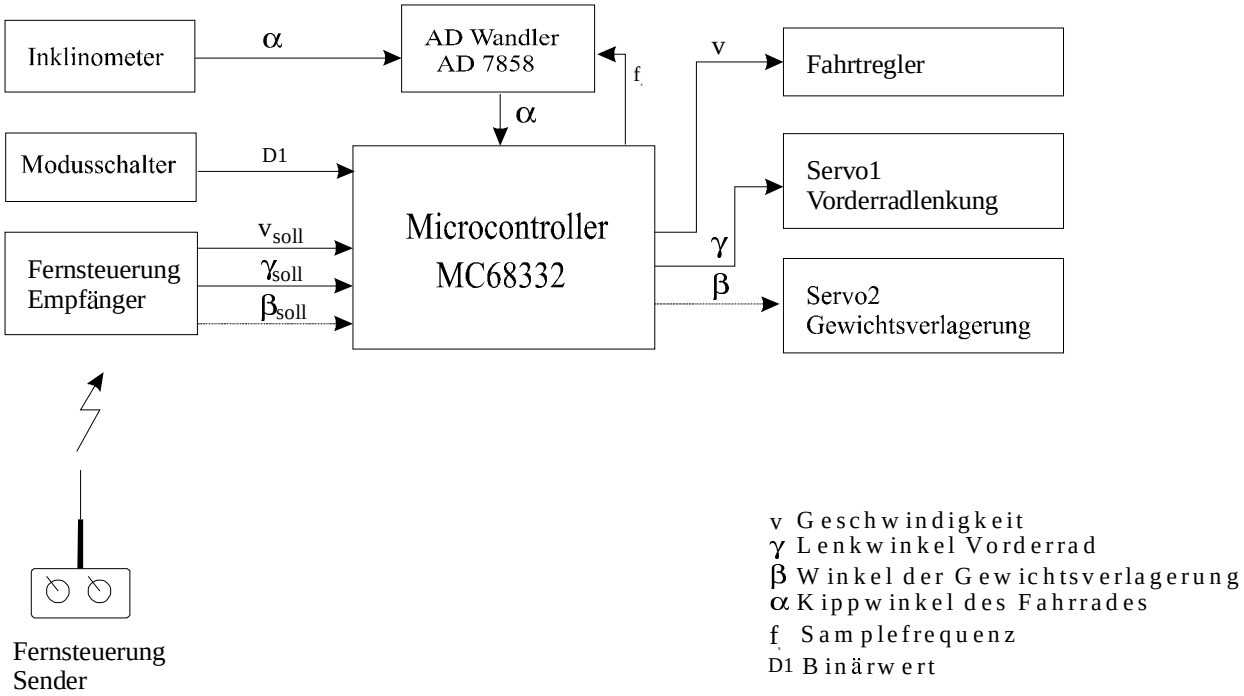
8.2.1. Eingangs-und Ausgangssignale

Das folgende Blockschaltbild zeigt die Signalpfade vom Eingang bis zu den Ausgängen. Über eine handelsübliche Fernsteuerung werden drei verschiedene Sollwerte angegeben: Sollgeschwindigkeit, Sollkipplage, Sollgewichtsverlagerung. Obwohl die Regelung nur die Kipplage als Input verwendet, werden alle Signale durch den Prozessor hindurchgeschlauft. So kann man in einer weiteren Phase leicht die anderen zwei Sollwerte in die Regelung einfliessen lassen. Man muss nur noch die Software abändern. Die Hardware ist so flexibler gegenüber Änderungen. Momentan braucht der Regler nur den Kippsollwert. Der Geschwindigkeitssollwert wird eingelesen und

sogleich am Regler vorbei auf den Fahrtregler gegeben. Das Einlesen des Sollwerts für die Gewichtsverlagerung ist für eine spätere Entwicklungsphase vorgesehen worden. Dieses Signal wird bei unserer Realisierung nicht verwendet.

Neben den Sollwerten müssen der Regelung auch die Istwerte der zu regelnden Grösse zur Verfügung gestellt werden. Es wird dazu verwenden dazu ein Inklinometer verwendet, der proportional zur Schräglage eine Spannung abgibt. Diese Spannung wird über den AD-Wandler in den Microcontroller eingelesen.

Stellsignale für den Fahrtregler, den Servo1 und den Servo2 werden direkt vom Controller ausgegeben. Der Microcontroller behält die volle Kontrolle über die Aktoren.



Blockschaltbild Hardware

8.2.2. Codierung der Signale und Bindung an die Hardware

Die untenstehende Tabelle zeigt die verschiedenen Schnittstellen zwischen den Schaltungsblöcken und ihren Signalen.

 © Copyright 1998 SA

Signal	Schnittstelle	Codierung	Bindung
α Kippwinkelistwert	Inklinometer -> AD-Wandler	Analog	ADC: + AIN0,- AIN1 Differential Input
α Kippwinkelistwert	AD-Wandler -> uC	Digital (16 Bit)	QSM: QSPI
vs Geschwindigkeit-sollwert	Empfänger -> uC	Pulsweitenmodulation	TPU: Kanal 0 Funktion PPWA
γs Kippwinkel-sollwert	Empfänger -> uC	Pulsweitenmodulation	TPU: Kanal 1 Funktion PPWA
βs Gewichtsver-lagerung	Empfänger -> uC	Pulsweitenmodulation	TPU: Kanal 2 Funktion PPWA
D1 Modus	Schalter -> uC	Binär	PORTE: Bit 7 (SIZ1)
v Geschwindigkeit	uC -> Fahrtregler	Pulsweitenmodulation	TPU: Kanal 8 Fuktion PWM
γ Lenkwinkel	uC -> Servo1	Pulsweitenmodulation	TPU: Kanal 9 Funktion PWM
β Gewichtsver-lagerung	uC -> Servo2	Pulsweitenmodulation	TPU: Kanla 10 Funktion PWM
fs Samplerate	uC -> AD-Wandler	Frequenz	TPU: Kanal 11 Funktion PWM

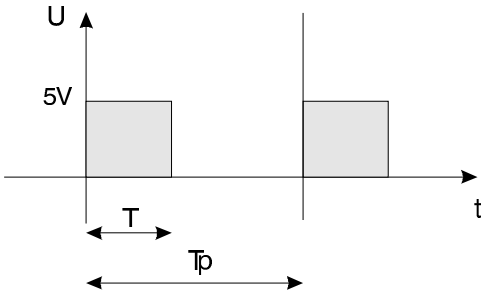
Tabelle Schnittstellen

Im Wesentlichen gibt es fünf verschiedene Codierungsarten, die verarbeitet werden müssen:

Analog: Der Inklinometer bildet den Bereich von -20° bis 20° ab auf eine Spannung von 0V bis 5V. (siehe Beschreibung Inklinometer)

Digital: Die serielle Schnittstelle sendet ein16 Bit Paket, wobei davon 12Bit Nutzdaten sind und 4Bit Steuerdaten. (siehe Datenblätter AD7858)

PWM: Alle Modellbaukomponenten in unserem System arbeiten mit PWM. Der Empfänger gibt alle 20ms einen Puls aus. Die Breite des Pulses sagt etwas aus über die Stellung des Steuerknüppels an der Fernsteuerung. Die Pulsbreite variiert im Bereich von 1.4 +/- 0.4 ms. Die Servos und der Fahrtregler können eine höhere Frequenz von Pulsen verkraften als die 50Hz des Empfängers.



T= Pulsbreite
Tp= Periode

Binär: Mit einem Schalter auf dem Zusatzboard wird durch einen TTL Pegel an Port E angegeben, in welchem Modus die Software arbeiten soll.

High: Modus Regler

Low: Modus Steuerung

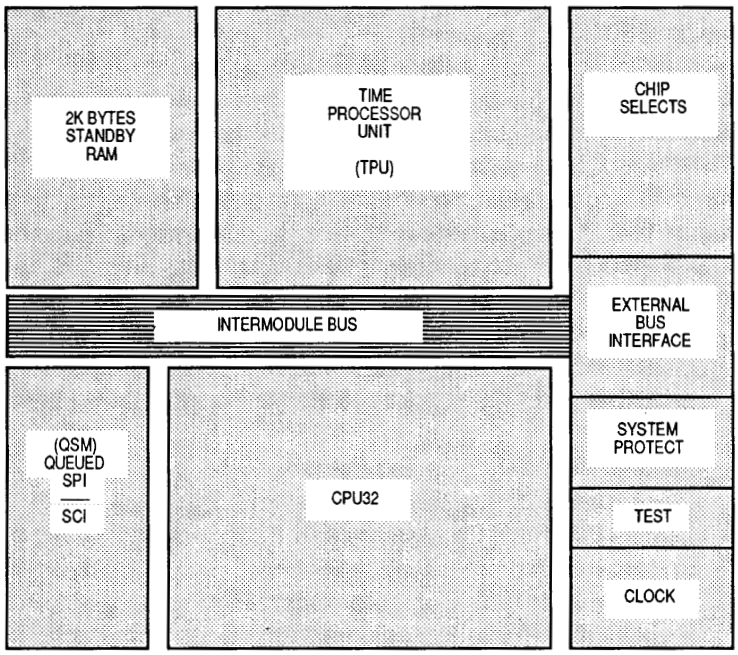
Frequenz: Der Microcontroller gibt Pulse an den AD-Wandler. Bei jeder steigenden Flanke startet der Chip eine neue Wandlung.

8.3. Der Microcontroller MC68332

Der MC68332 ist ein Mitglied von Motorolas 32Bit Microcontroller Familie. Er bietet leistungsstarke periphere Subsysteme. Diese können autonom ihre Funktionen ausführen.

Kommuniziert mit der CPU wird über einen intermodularen Bus. Das untenstehende Bild zeigt die verschieden Module, die der Microcontroller zur Verfügung stellt.

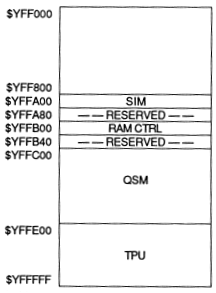
- 32-bit CPU Modul (CPU32)
- Timerfunktionen (TPU Time Processor Unit)
- Serielle Schnittstellen (QSM Queued Serial Modul)
- Modul zur Systemintegration (SIM System Integration Module)
- On-Chip 2 K Bytes Standby RAM



Blockdiagramm MC68332

Nach dem Aufstarten des Systems muss nun die CPU die verschiedenen Module konfigurieren. Dies geschieht über das interne RAM. Die CPU schreibt über den intermodularen Bus die Initialiserungsdaten an bestimmte, für die einzelne Module reservierte Speicherstellen. Die peripheren Bausteine holen sich die Daten ab und führen die gewünschten Funktionen aus. Ebenfalls werden die Resultate der TPU-Kanäle im RAM abgelegt.

Die Memory Map zeigt, wie die einzelnen Module auf den Speicher abgebildet werden.



Memory Map

 © Copyright 1998 SA

8.3.1. Time Processor Unit (TPU)

Die Time Processor Unit ist zuständig für alle Funktionen, die auf der Basis von Timern ausgeführt werden müssen. Es werden 16 Kanäle zur Verfügung gestellt, wobei jeder Kanal mit einem Pin verbunden ist. Dies sind die Pins TP0-TP15. In der Initialiserungsphase kann jedem Kanal eine Funktion zugeteilt werden.

Aus den folgenden Funktionen kann ausgewählt werden:

- Diskreter Input/Output (DIO)
- Input registrieren/Input Transitionen zählen (ITC)
- Output vergleichen (OC)
- Pulsbreitenmodulation (PWM)
- Synchronisierte Pulsbreitenmodulation (SPWM)
- Perioden Messungen (PMA)
- Pulsgenerator (PSP)
- Schrittmotoransteuerung (SM)
- Perioden/Pulsbreiten Messungen (PPWA)

Die TPU stellt zwei frei laufende 16-bit Timer (TCR1 und TCR2). Die Zeitbasis ergibt sich aus dem Systemclock, der je nach Wunsch heruntergeteilt werden kann. Als Basis für TCR2 kann auch ein externer Pin gewählt werden.

Zur Konfiguration der TPU geht man in zwei Schritten vor. Als erstes müssen die übergeordneten modulspezifischen Konfigurationregister beschrieben werden. Im zweiten Schritt werden dann die Register für die jeweils benötigten Kanäle beschrieben.

Bemerkung: Im Folgenden werden nur die für unsere Zwecke relevanten Registereinträge erläutert. Die restlichen Bits werden auf den Defaultwerten belassen. Für genauere Informationen kann man die Handbücher des MC68332 konsultieren.

<u>TPU Modul Konfigurationsregister</u>

TMCR $FFFE00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0							0						
TCR1 Prescaler									PSCK						

Die obigen Einstellung bedeuten, dass der Systemclock mit 32 dividiert wird. Der Clock auf dem System hat eine Frequenz von 16.77MHz. Damit existiert in Timer1 eine Auflösung von 1.908us. Die maximale Zeitspanne, die mit dieser Einstellung noch gemessen werden kann, beträgt somit 125ms. Wobei abhängig von der TPU-Funktion nicht immer die volle Zeitspanne genutzt werden kann.

16.77MHz/32= 1/ Auflösung [s]
Auflösung [s] * 2^{16} = $TCR1_{max}$

Kanalspezifische Kontrollregister

Kanal Funktionswahl Register
CFSR0 $FFFE0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0												
Kanal 15 Hilfsfunktion Diskreter Output				Kanal 14				Kanal 13				Kanal 12			

CFSR1 $FFFE0E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
Kanal 11 Samplerate PWM				Kanal 10 Gewichtsverl. PWM				Kanal 9 Lenkrad PWM				Kanal 8 Fahrtregler PWM			

CFSR3 $FFFE12

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Kanal 11 Geschwindigkeit				Kanal 10 Sollwert Gewichtsv.				Kanal 9 Sollwert Lenkung				Kanal 8 Sollwert Geschw.			

➔ In diesen Registern werden den einzelnen Kanälen Funktionen zugewiesen. Für jeden Kanal braucht es einen Eintrag von 4Bit in einem der 4 Register (CFSR0-CFSR3). Der Wert null bedeutet, dass derjenige Kanal keine Funktion ausübt.

Host Sequence Register
HSQR1 $FFFE16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								0	1	1	0	1	0	1	0
Kan7		Kan6		Kan5		Kan4		Kan3		Kan2		Kan1		Kan0	

➔ In diesem Register werden Funktionsmodi zugewiesen. So wird für Kanal3 der Modus „Pulse zählen“ eingestellt. Die Kanäle 0 bis 2 arbeiten mit einem 24Bit Akkumulator, um Pulsbreiten zu messen.

Host Service Register
HSRR0 $FFFE18

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 1	1 0							1	0	1	0	1	0	1	0
Kan15		Kan14		Kan13		Kan12		Kan11		Kan10		Kan9		Kan8	
Set Pin High Set Pin Low										Initialisierung					

HSRR1 $FFFE1A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								0	1	1	0	1	0	1	0
Kan7		Kan6		Kan5		Kan4		Kan3		Kan2		Kan1		Kan0	
										Initialisierung					

 © Copyright 1998 SA

➔ Die Kanäle 0 bis 3 und 8 bis 11 werden durch Setzen der entsprechenden Bits gezwungen, sich zu inititalisieren.

➔ Der Kanal 15 wird durch setzen der Bits auf 01 dazu gebracht, den Pin TP15 auf High zu setzen. Entsprechend wird der Pin bei Setzen der Bits auf 10 nach Low gezogen.

Kanal Priorität Register
CPRO $FFFE1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1							1	0	1	0	1	0	1	0
Kan15		Kan14		Kan13		Kan12		Kan11		Kan10		Kan9		Kan8	
										Kanal Priorität Middle					

CPR1 $FFFE1E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								0	1	1	0	1	0	1	0
Kan7		Kan6		Kan5		Kan4		Kan3		Kan2		Kan1		Kan0	
										Kanal Priorität Middle					

➔ Durch Setzen der Kanalpriorität wird der Kanal automatisch freigegeben. Sind die Bits auf Null gesetzt, so ist der Kanal gesperrt.

Als nächstes gilt es noch, für jeden Kanal funktionsspezifische Parameter einzustellen. Dies wird ebenfalls für jeden Kanal an einer spezieller Stelle im RAM gemacht.

Funktion	RAM Adresse	Parameter	Wichtige Einstellungen
PWM	$FFFFW0	Channel_Control	Timer 1 als Basis wählen
	$FFFFW2	PWMHI	High Time der Pulse
	$FFFFW6	PWMPER	Periodendauer
PMA	$FFFFW0	Channel_Control	Steigende Flanke erkennen; Timer 1 als Basis
	$FFFFW2	Max_Additional	Keine zusätzlichen Zähne
	$FFFFW6	Ratio	$0
		Tcr2_max_Value	$FF
	$FFFFWA	Period_Low_Word	Resultatwert
PPWA	$FFFFW0	Channel_Control	Steigende Flanke erkennen, Timer 1 als Basis
	$FFFFW2	Max_Cnt	$1 (Anzahl aufsummierter Pulsbreiten)
	$FFFFW8	Accum_Rate	$FF
	$FFFFWA	PPWA_LW	Resultat
W = Kanalnummer			

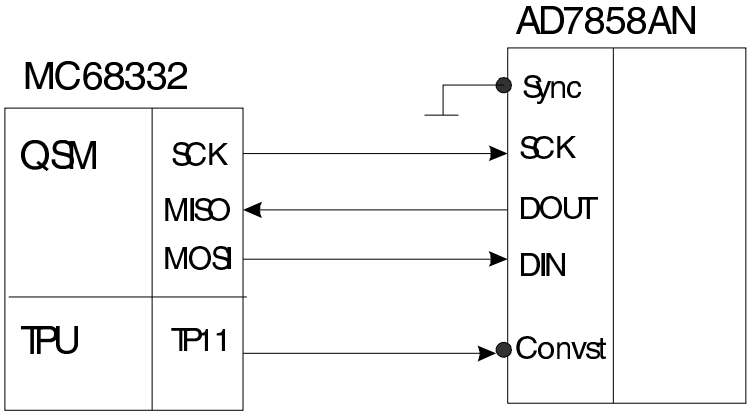
Für ausführliche Beschreibungen der Funktionen sind die Handbücher von Motorola zu konsultieren. Dort sind genauere Beschreibungen zu jeder Funktion vorhanden. In dieser Dokumentation war es nur das Ziel zu beschreiben, wie die Kanäle verwendet wurden.

8.3.2. Queued Serial Modul (QSM)

Um die Daten vom Wandler auf den Microcontroller zu bringen, wird eine synchrone serielle Schnittstelle verwendet. Das QSM stellt das QSPI Submodul zur Verfügung, welches wir verwendet haben, um die 12-Bit Daten zu übertragen. Dabei ist das Microcontroller Modul als Master und der AD-Wandler als Slave konfiguriert. Das QSPI Modul fragt also in regelmässigen Abständen die Werte ab und speichert sie in einem Puffer. Dieser Puffer kann als Ringpuffer konfiguriert werden

(Wraparound Mode). Somit werden andauernd Daten vom AD-Wandler übertragen. Die CPU holt sich diese Werte im Speicher ab.

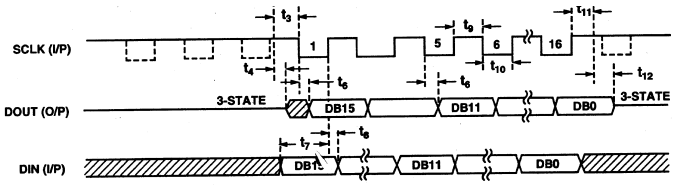
Die letzten 16 Übertragungen sind jeweils noch verfügbar und können für eine Mittelwertberechnung verwendet werden. Um dem AD-Wandler zwischen zwei seriellen Übertragungen Zeit zu geben, eine neue Wandlung durchzuführen, kann man einen Delay programmieren. Die Frequenz des seriellen Clocks ist ebenfalls programmierbar.

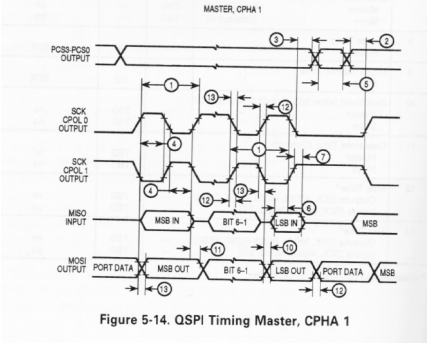


Master Slave

Anbindung des AD-Wandlers

Das obenstehende Bild zeigt die Anbindung des AD-Wandlers an die Schnittstelle des QSPI Moduls. Um zu gewährleisten, dass die Daten auch richtig übertragen werden, muss das QSPI Modul so konfiguriert werden, dass die Timing Spezifikation des AD-Wandlers eingehalten werden. Grundsätzlich funktioniert die serielle Uebertragung so, dass bei jeder steigenden Flanke ein Datenbit übertragen wird. Der Pin SCK des Controllers wird nun automatisch so gesteuert, dass er 16 steigende Flanken generiert und danach wieder eine bestimmte Zeit den Pin auf High belässt. Während den Flanken werden nun die Daten von DOUT nach MISO und von MOSI nach DIN übertragen.





Timing Diagram des AD7858 Timing Konfiguration des QSM

Für die Konfiguration des QSM müssen wiederum einige Register beschrieben werden. Dies sind die Pin Control Register und die QSPI Submodule Register

<u>QSM Pin Control Register</u>

QPAR $FFFC16

7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1
TXD	PCS3	PCS2	PCS1	PCS0 SS	SCK	MOSI	MISO

 © Copyright 1998 SA

➔ Dieses Register definiert ob die Pins als normale I/O Pins oder als QSPI Pins verwendet werden. (0 = I/O Pin; 1= QSPI Pin)

QDDR $FFFC17

7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	0
0	PCS3	PCS2	PCS1	PCS0 SS	0	MOSI	MISO

➔ Im Register QDDR wird festgelegt ob die Pins als Output oder als Input verwendet werden. (0 = Input; 1 = Output)

<u>QSPI Submodul Register</u>

Control Register 0
SPCR0 $FFFC18

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1
MSTR	WOMQ	BITS				CPOL	CPHA	SPBR							

MSTR: 1 = Master
0 = Slave

WOMQ: 1 = Wired OR Mode für Pins
0 = Normal Mode

BITS: 0000 = 16 Bits pro Übertragung
1000 bis 1111 = 8 bis 15 Bits pro Übertragung

CPOL: Inaktive Pegel des SCK
1 = High
0 = Low

CPHA: 1 = Der Wert wird während der zweiten Flanke übertragen
0 = Der Wert wird während der ersten Flanke übertragen

SPBR: SCK Frequenz = 16.78MHz/(2*SPBR)
493 kHz

Control Register 1
SPCR1 $FFFC1A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0
SPE				DSCKL							DTL				

SPE: 1 = QSPI enabled
0 = QSPI disabled

DSCKL: Verzögerung zwischen Chip Select setzen und Übertragung starten

Diese Verzögerung ist derzeit nicht relevant, da keine Chip Select Pins angesteuert werden

DTL: Verzögerung zwischen zwei Transfern

Verzögerung = ((32*DTL/ 16.78MHz)

Eingestellt auf 67.6 us

Control Register 2
SPCR2 $FFFC1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	1	0	0	1	0	0	0	0	0
SPIFIE	WREN	WRTO	0	ENDQP				0	0	0	0	NEWQP			

SPIFIE: 0 = Interrupts disabled

WREN: 1 = Wraparound Mode

WRTO: 0 = $0 Erste Relativadresse im Ringpuffer

ENDQP: 0000 bis 1111 Endadresse des Ringpuffers

NEWQP: Falls WRTO = 1 Startadresse des Ringpuffers

Statusregister
SPSR $FFFC1F

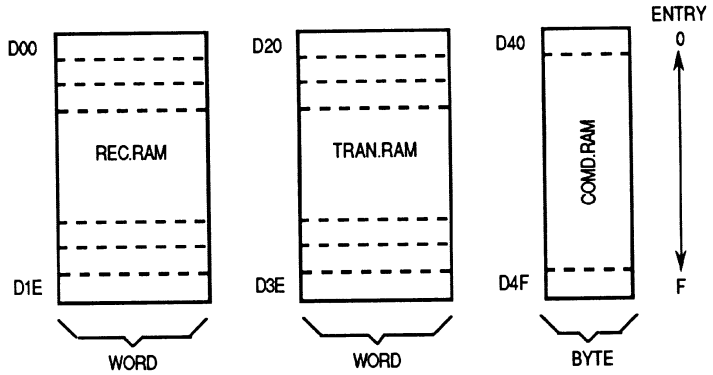
7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1
SPIF				CPTQP			

SPIF: 1 = Ist am Ende einer Schleife im Ringpuffer angelangt

CPTQP: Letzter beschriebener Adressplatz im Puffer

Um Daten zu senden und auch zu empfangen, stellt das QSPI Modul Speicherbereiche zur Verfügung. Daten werden gesendet, indem man Puffer „Tran.RAM“ beschreibt. Die serielle Schnittstelle sendet darauf einen Wert nach dem anderen an den AD-Wandler. Ist der Wraparound Mode eingeschaltet, so beginnt die Übertragung nach den 16 Übertragungen wieder am Anfang. Die empfangenen Daten werden im Puffer „Rec.RAM“ gespeichert.

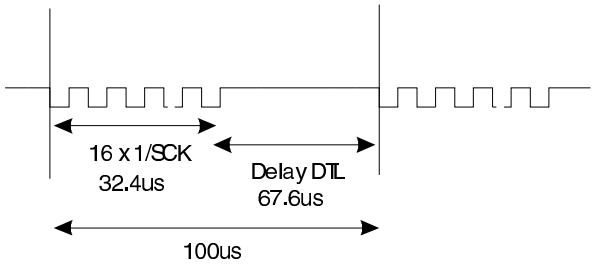
Über ein Control Byte kann jede einzelne Übertragung speziell konfiguriert werden. Diese Bytes werden ebenfalls in einen Puffer namens „Comd.RAM“ geschrieben.



QSPI RAM

 © Copyright 1998 SA

Die ganzen Konfigurationen in den verschiedenen Registern führen zu folgendem zeitlichen Ablauf der seriellen Übertragungen.



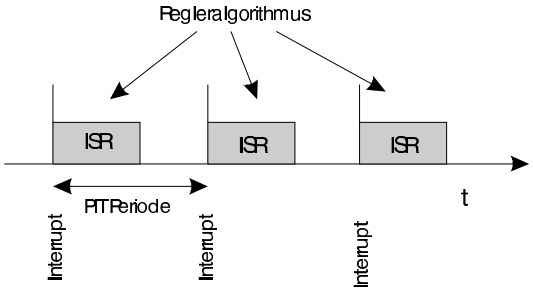
Wie in diesem Diagramm zu sehen ist, wird alle 0.1ms ein neuer Wert vom AD-Wandler an den Controller übergeben. Da wir im „Rec.RAM“ Puffer Platz für 16 Werte haben, können wir software-mässig einen Mittelwert über die Zeitspanne von 1.6ms bilden. Diese Mittelwertbildung ist zu vergleichen mit einem hardwaremässigen Tiefpassfilter.

8.3.3. System Integration Modul (SIM)

Das System Integration Modul des MC68332 besteht aus fünf verschieden Submodulen.

- System Konfiguration und Schutz
- Clock Synthesizer
- Chip Selects
- Externer Bus Interface
- System Test

Für das Ziel einen Regleralgorithmus zu implementieren, muss garantiert werden, dass zu fest vorbestimmten Zeitpunkten der Regleralgorithmus ausgeführt wird. Dazu bietet das Submodul „System Konfiguration und Schutz“ einen periodischen Interrupt Timer an. Dieser Timer kann so konfiguriert werden, dass in festgelegten zeitlichen Abständen Interrupts generiert werden. Durch Eintragen der Interrupt Service Routine in der Exception Tabelle wird die CPU dazu gebracht, immer wenn der Interrupt auftritt den Regleralgorithmus zu starten.



Periodic Interrupt Timer

Zur Konfiguration dieses periodischen Interrupt Timers müssen wiederum einige Register beschrieben werden. Daneben muss noch die Adresse der Interrupt Service Routine in der Exception Vektor Tabelle eingetragen werden.

Periodic Interrupt Timing Register
PITR $FFFA24

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1
								PTP	PITR6	PITR5	PITR4	PITR3	PITR2	PITR1	PITR0

PTP: 1 = Die Referenz-Clockfrequenz (32.768 kHz) wird mit 512 geteilt

0 = Die Refernz-Clockfrequenz (32.768 kHz) wird nicht geteilt

PITR0-7: Wert zur Berechnung Periodendauer

Periode = PITR-Wert* 4* Skalierung / 32.768 kHz

Skalierung = 512 (falls PTP = 1)

= 1 (falls PTP = 0)

Eingestellter Wert = 5ms

Periodic Interrupt Control Register
PICR $FFFA22

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
					PIRQL2	PIRQL1	PIRQL0	PIV7	PIV6	PIV5	PIV4	PIV3	PIV2	PIV1	PIV0

PIRQL0-2: Interrupt Request Level 1 - 7 wählen

Eingestellter Wert = 7

PIV0-7: Vektornummer in der Exception Vektor Tabelle

Eingestellter Wert = 64

In einem weiteren Schritt muss man noch die Adresse der ISR in der Exceptiontabelle speichern. Die Speicheradresse dieses Tabelleneintrages im RAM wird folgendermassen aus dem Vektor Basis Register und der Vektor Nummer berechnet.

Speicheradresse des Zeigers auf die ISR = VBR + (Vector Nummer * 4)

Table 5 Exception Vector Table

Vector Number (Decimal)	Vector Offset (Hexadecimal)	Assignment
0	0	Reset: Initial Stack Pointer
1	4	Reset: Initial Program Counter
2 - 15	8 - 3C	Various Errors and Exceptions
16 - 23	40 - 5C	Unassigned, Reserved
24	60	Spurious Interrupt
25	64	Level 1 Interrupt Autovector
26	68	Level 2 Interrupt Autovector
27	6C	Level 3 Interrupt Autovector
28	70	Level 4 Interrupt Autovector
29	74	Level 5 Interrupt Autovector
30	78	Level 6 Interrupt Autovector
31	7C	Level 7 Interrupt Autovector
32 - 47	80 - BC	Trap Instruction Vectors
48 - 63	C0 - FC	Unassigned/Reserved
64 - 255	100 - 3FC	User Defined Vectors

Exception Tabelle

Auf keinen Fall darf vergessen werden, die Ausmaskierung der Interrupts im Statusregister der CPU auszuschalten.

 © Copyright 1998 SA

8.4. Das Evaluationboard M68332EVS

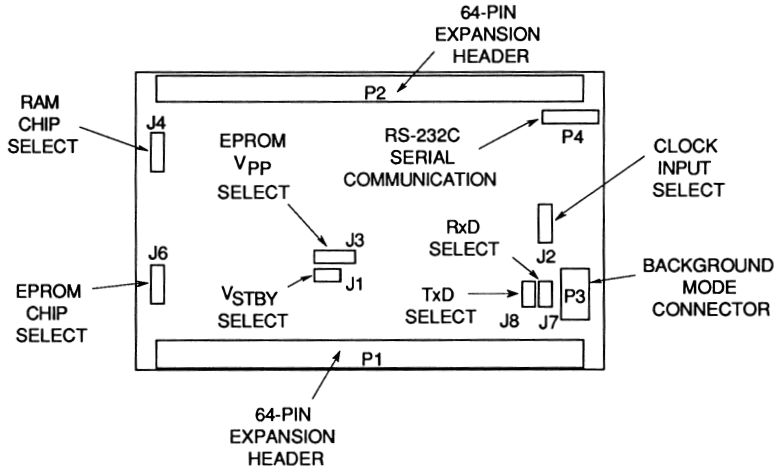
Um sofort mit dem programmieren der Hardware beginnen zu können, benutzten wir ein Evaluationboard von Motorola. So mussten wir nicht viel Zeit investieren in die Entwicklung eines eigenen Hardwareboards.

Das Evaluation System besteht aus zwei verschiedenen Boards.

- Business Card Computer (BCC)
- Platform Board (PFB)

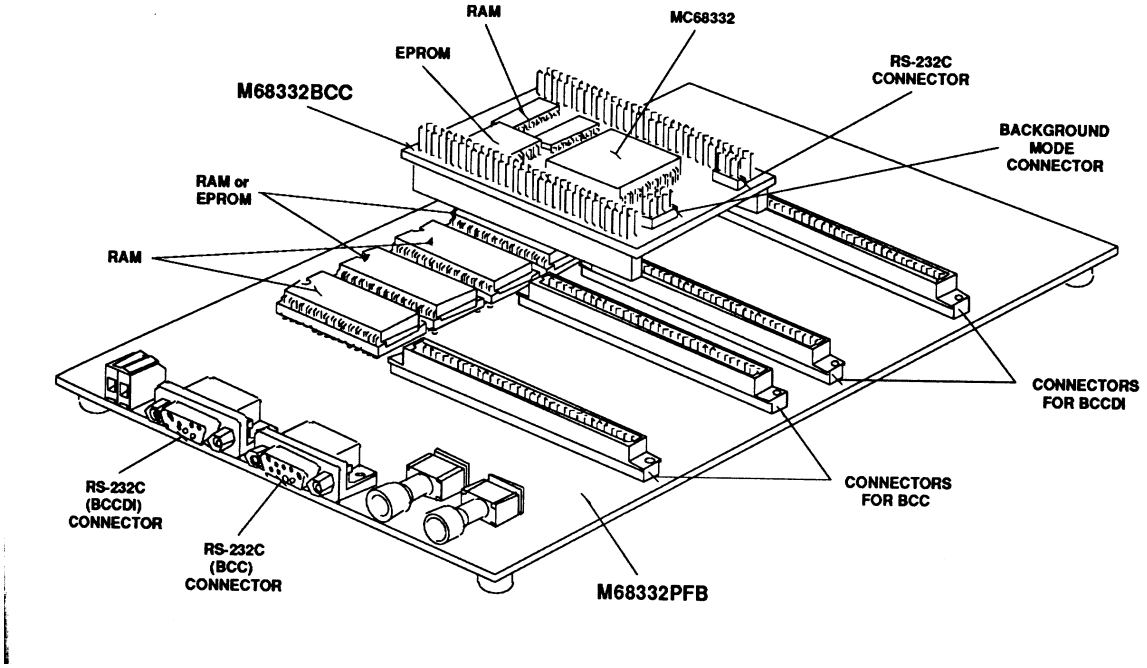
Auf dem BCC ist der eigentliche Zielrechner (MC68332) untergebracht. Dieses Board ist nur 5.7 x 8.9cm gross und somit sehr gut geeignet, um auf dem Motorrad plaziert zu werden. Des Weiteren besitzt dieses Board 64k x 16 bit EPROM, 32k x 16 bit RAM Speicher, eine RS-232C kompatible Schnittstelle für die Kommunikation mit einem Hostrechner und einen Background mode interface port. Über diesen Port kann ein Debugger auf die Daten im Controller zugreifen und diese kontrollieren. Über dieses Kabel wird auch der Maschinencode heruntergeladen.

Um möglichst alle Controller Pins für den Entwickler zugänglich zu machen, sind jeweils auf der Seite zwei Pinstecker montiert.



BCC

Auf der Grundplatine, dem PFB, gibt es weitere Bausteine, die für die Entwicklung hilfreich sein können. So ist auf diesem Board ein Stecker für die Spannungsversorgung montiert. Dieser fehlt auf dem BCC. Die Spannungsversorgung muss also beim Betrieb ohne Grundplatinen, über die Pinstecker zugeführt werden. Zusätzlich sind auf der Grundplatine noch Sockel montiert für RAM, EPROM Bausteine und für einen Coprozessor. Da diese Platine ziemlich gross ist (15.88 x 25.4 cm), ist es nicht möglich, die Grundplatine auch auf dem Motorrad zu plazieren. Sie diente also nur für die erste Testphase, in der wir für das kleine Board noch keine Spannungsversorgung hatten.



PFB

Die Pinbelegungen auf den Pfostenstecker sehen folgendermassen aus.

P1					P2				
GND	1	●	●	2 GND	GND	1	●	●	2 GND
VDD	3	●	●	4 VDD	VDD	3	●	●	4 VDD
DTROUT	5	●	●	6 A0	D1	5	●	●	6 D0
A1	7	●	●	8 A2	D3	7	●	●	8 D2
A3	9	●	●	10 A4	D5	9	●	●	10 D4
A5	11	●	●	12 A6	D7	11	●	●	12 D6
A7	13	●	●	14 A8	D9	13	●	●	14 D8
A9	15	●	●	16 A10	D11	15	●	●	16 D10
A11	17	●	●	18 A12	D13	17	●	●	18 D12
A13	19	●	●	20 A14	D15	19	●	●	20 D14
A15	21	●	●	22 A16	RXD DI	21	●	●	22 TXD DI
A17	23	●	●	24 A18	MODB DI	23	●	●	24 XMT 232
EPROM-W	25	●	●	26 EPROM-VPP	CSBOOT	25	●	●	26 RCV 232
13-24V DI	27	●	●	28 VSTBY	R/W	27	●	●	28 MODCK
13-24V DI	29	●	●	30 T2CLK	TSTME/TSC	29	●	●	30 A23/CS10
TP15	31	●	●	32 TP14	A22/CS9	31	●	●	32 A21/CS8
TP13	33	●	●	34 TP12	A20/CS7	33	●	●	34 A19/CS6
TP11	35	●	●	36 TP10	FC2/CS5	35	●	●	36 FC1/CS4
TP9	37	●	●	38 TP8	FC0/CS3	37	●	●	38 BGACK/CS2
TP7	39	●	●	40 TP6	BG/CS1	39	●	●	40 BR/CS0
TP5	41	●	●	42 TP4	IRQ1	41	●	●	42 IRQ2
TP3	43	●	●	44 TP2	IRQ3	43	●	●	44 IRQ4
TP1	45	●	●	46 TP0	IRQ5	45	●	●	46 IRQ6
MOSI	47	●	●	48 MISO	IRQ7	47	●	●	48 BERR
PCS0/SS	49	●	●	50 SCK	DSACK0	49	●	●	50 DSACK1
PCS2	51	●	●	52 PCS1	AVEC	51	●	●	52 RMC
TXD	53	●	●	54 PCS3	DS	53	●	●	54 AS
BKPT/DSCLK	55	●	●	56 RXD	SIZ0	55	●	●	56 SIZ1
RESET	57	●	●	58 FREEZE/QUOT	11RESET	57	●	●	58 CLKOUT
IPIPE/DSO	59	●	●	60 IFETCH/DSI	EXTAL	59	●	●	60 HALT
VDD	61	●	●	62 VDD	VDD	61	●	●	62 VDD
GND	63	●	●	64 GND	GND	63	●	●	64 GND

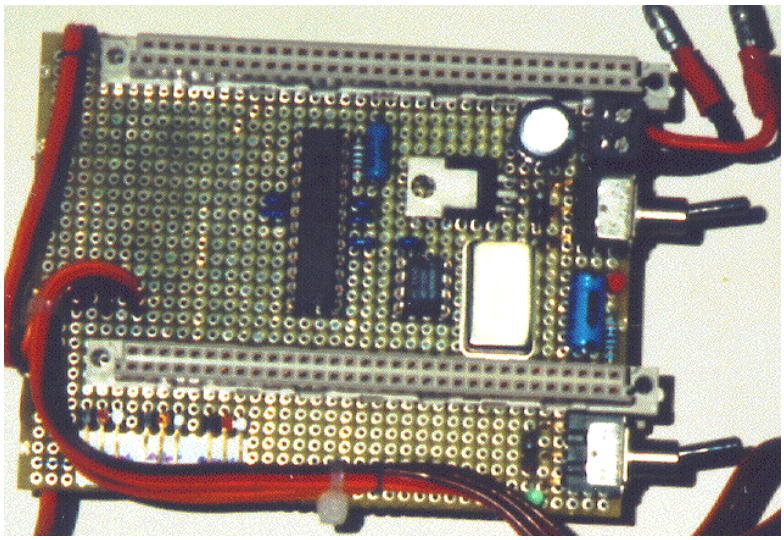
Pinbelegung

Detailiertere Informationen über das Promotional Evaluation System findet man im Benutzer Handbuch [M68332EVSP].

 © Copyright 1998 SA

8.5. Das Zusatzboard

Nun bestand die Aufgabe, unsere Signale vom Empfänger in den Prozessor reinzuschlaufen und auch wieder an die Servos auszugeben. Zusätzlich mussten noch eine Spannungsversorgung bereitgestellt und ein AD-Wandler an die serielle Schnittstelle des Prozessor angeschlossen werden. Zu diesem Zweck wurde in einem Sandwichaufbau eine zweite Platine bestückt. Diese wird einfach auf die Pinstecker des Evaluationboard gesteckt, um leichten Zugang zu den verschiedenen Pins des BCC und somit des MC68332 zu haben.

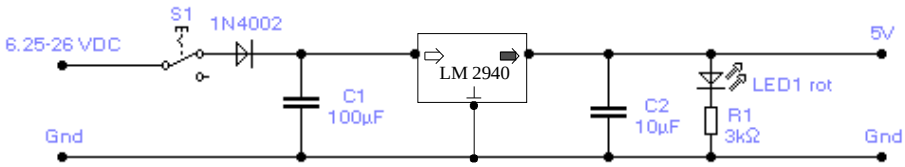


Zusatzplatine

8.5.1. Spannungsversorgung

Am Eingang der Spannungsversorung kann eine nicht stabilisierte Gleichspannung angelegt werden. Der Bereich liegt zwischen 6.25 und 26V. Dieser Spannungsbereich muss eingehalten werden, damit der Spannungsregler korrekt arbeitet und die stabilisierten 5V garantieren kann. Der Akku des Motorrades liefert eine Spannung von 7.2V und liegt somit gerade noch im Bereich.

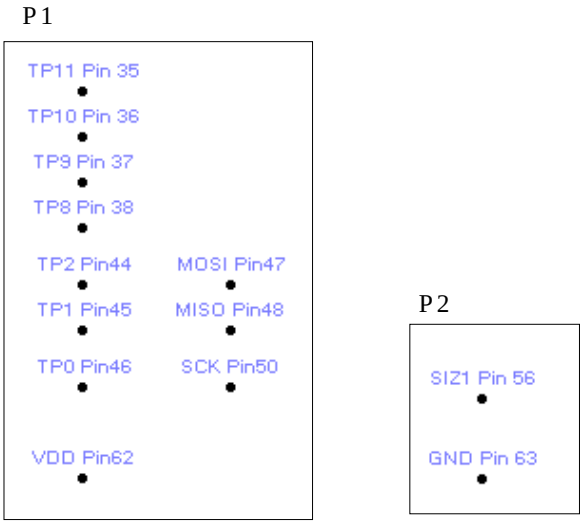
Mit dem Schalter S1 wird die ganze Spannungsversorgung für das Sandwichboard eingeschaltet. Angezeigt wird dies durch die rote LED.



Schaltplan Spannungsversorgung

8.5.2. Pinbelegung der Pfostenstecker

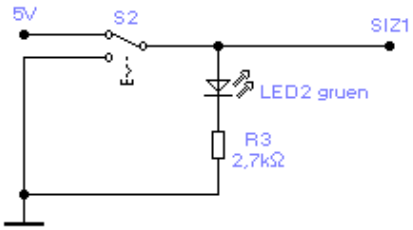
Die Pinbenennung auf dem Zusatzboard entsprechen den Pins auf dem Prozessorboard (BCC).



Pinbelegung

8.5.3. Modusumschaltung

Durch einen weiteren Schalter auf der Platine kann dem Prozessor ein Logikpegel zugänglich gemacht werden. Wir führen diesen Wert auf den Port E auf das Bit 7 (SIZ1 Pin). Der Pegel wird durch die grüne LED angezeigt. Diese Schalterstellung kann benutzt werden, um in der Software eine Fallunterscheidung zu machen. Unser Ziel war es, von aussen hardwaremässig vom Modus „Regelung“ auf den Modus „Steuerung“ umschalten zu können.



Schaltplan Modusumschaltung

In einer weiteren Phase wäre sinnvoll, diesen Schalter nicht auf dem Motorrad zu plazieren, sondern auf der Fernsteuerung. So kann man bei einer Freifahrt gewährleisten, jeder Zeit die direkte Kontrolle über das Motorrad zu erlangen.

8.5.4. Ein- und Ausschlaufen der Signale

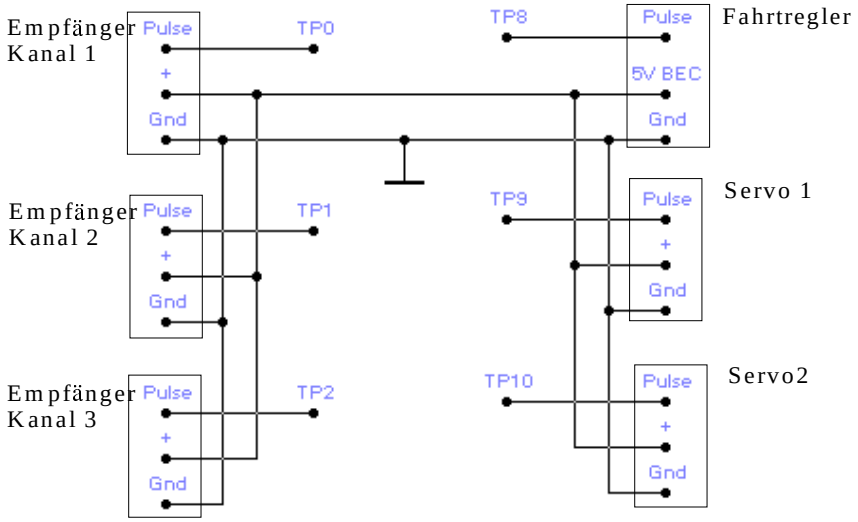
Das Ein- und Ausschlaufen der Signale geschieht über Modellbaustecker. Diese Stecker sind dreifarbig.

Die Zuordnung der Farben sieht folgendermassen aus.

- Braun = Gnd
- Rot = 5V
- Orange = Pulse

Der Empfänger und die Servos werden nicht über den Spannungsregler LM2940 gespiesen, sondern durch den Fahrtregler. Dieser liefert ebenfalls eine 5V Spannung. Um jedoch für den Prozessor eine stabile Spannung zu garantieren, haben wir die 5V des Fahrtreglers nur zur Speisung der Modellbaukomponenten verwendet.

 © Copyright 1998 SA



Schaltplan Signalpfade

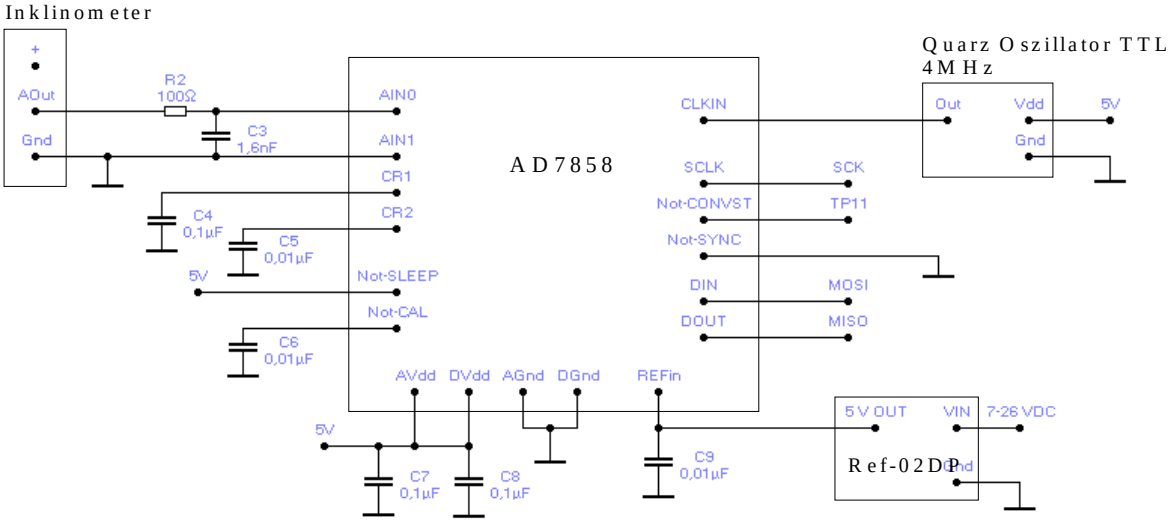
8.5.5. AD-Wandler

Zur Anbindung des Inklinometers wird ein AD-Wandler benötigt. Es wurde entschieden einen 8 Kanal 12-Bit Wandler mit einer seriellen Schnittstelle zu verwenden. Durch die serielle Anbindung des Wandlers an den Controller bleibt die Anzahl belegter Pins am Controller klein. Die verschiedenen Parallelports bleiben so frei für andere Zwecke. Die maximale Datenrate pro Kanal, falls alle 8 Kanäle belegt werden, beträgt 25kSPS. Diese Datenrate reicht für unsere Anwendung aus. Die Verwendlung eines 8 Kanal Wandlers ermöglicht in einer Ausbauphase das Einlesen weiterer Analogsignale

Für genauere Informationen über den Wandler konsultiere man im Anhang das Datenblatt des AD7858AN.

Durch Einsatz eines 12-Bit Wandlers, welcher einen Spannungsbereich von 0 bis 5V abdeckt, wird eine Auflösung von 1.2 mV erreicht. Bei dieser Grössenordnung ist man, gemäss Messungen, im Bereich des Rauschen. Aus diesem Grund wurde ein Tiefpass 1.Ordnung am Eingang des Wandlers plaziert. Die Eckfrequenz liegt bei 1MHz.

Als weitere Hilfsbausteine für den AD-Wandler wurde eine externe Referenzspannung von 5V und ein Quarz Oszillator (4MHz) angebracht. Der Inklinometer wurde an den differenziellen Input gehängt (AIN0 und AIN1).



Schaltplan AD-Wandler

9. Software

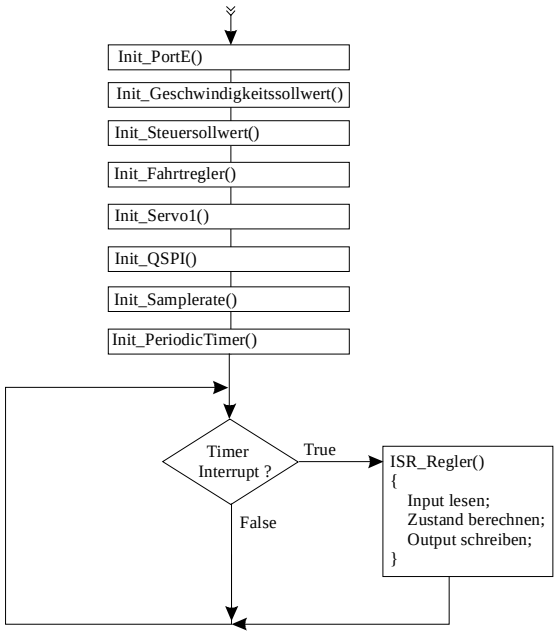
Um unseren Regleralgorithmus auf dem Zielrechner, dem MC68332, laufen zu lassen, müssen verschiedene Programmieraufgaben durchgeführt werden. Zum einen sind dies Initialisierungsroutinen für die verschiedenen Hardwaremodule. Diese Funktionen sind zuständig für das korrekte Einlesen der Reglereingangsgrössen und das Ausgeben der Regleroutputs. Des Weiteren muss natürlich der Regleralgorithmus implementiert werden. Da unser System nur 4.Ordnung ist, fällt diese Programmierung nicht allzu aufwendig aus.

Um die Programmierung des Controllers durchzuführenen wurde ein Cross-Compiler verwendet. Wir hatten also die Möglichkeit, all unsere Routinen in der Programmiersprache C zu formulieren. Dies beschleunigte den Entwicklungsprozess erheblich. Wird man später sehen, dass der Regleralgorithmus zu viel Zeit benötigt, um abgearbeitet zu werden, so kann man immer noch zum Assembler greifen, um den Code auf Geschwindigkeit zu optimieren.

Als Entwicklungsumgebung verwenden wir eine Software namens HiCross der Firma HiWare AG. Für nähere Informationen über dieses Tool konsultiere man das HiCross Handbuch.

9.1. Programmablauf

Wie bereits erwähnt, müssen beim Programmstart als erstes einige Initialisierungen durchgeführt werden. So müssen den Inputkanälen die entsprechenden Funktionen zugeteilt werden und der Interrupt Timer muss gestartet werden. Sind diese einmaligen Funktionen ausgeführt, führt das Programm in eine Endlosschleife. In dieser Schleife wartet das Programm, bis nach 5ms der Periodic Interrupt Timer einen Interrupt absetzt. Darauf führt der Prozessor die Interrupt Service Routine durch, in welcher unser Regleralgorithmus plaziert ist. Nach Beenden der Reglerroutine kehrt der Prozessor wieder in die Warteschlaufe zurück. Dort verbleibt er bis der nächste Interrupt generiert wird. Während der Wartezeit in der Endlosschlaufe könnten in einer späteren Entwicklungsphase Funktionen implementiert werden, die nicht zeitkritisch sind. So könnte man zum Beispiel gewisse Hilfsanzeigen mit neuen Werten versorgen.

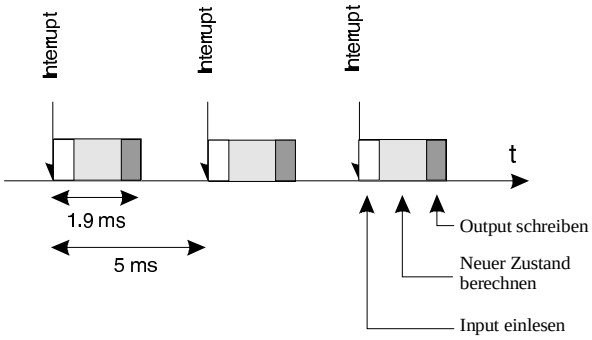


Schema Programmablauf

Mit diesem Programmablauf wird garantiert, dass die festgelegten Abtastzeiten eingehalten werden. Dies ist wichtig, weil der Parameter Abtastzeit in die diskreten Systemmatrizen des Reglers einfliessen. Es sollen also immer nach bestimmten Zeitpunkten die Inputs abgefragt, die neuen Zustände berechnet und wieder auf die Aktoren ausgegeben werden. Dabei muss beachtet

 © Copyright 1998 SA

werden, dass die Abtastzeit nicht kleiner ist als die Abarbeitungszeit des Regleralgorithmus. Es muss also der „Periodic Interrupt Timer“ dementsprechend eingestellt werden. Nach Ermitteln der Abarbeitungszeit, welche etwa 1.9ms beträgt, wurde die Abtastzeit als erster Ansatz auf 5ms festgesetzt.



Periodischer Ablauf

9.2. Einteilung der Software in Module

Um die Übersicht über die verschiedenen Initialisierungroutinen und Hilfsfunktionen nicht zu verlieren, wurde eine Einteilung in Module eingeführt. Je nach Zugehörigkeit zu einem bestimmten Hardwaremodul, wurde die Software zu bestimmen Dateien zugeteilt. So sind zum Beispiel alle TPU-Funktionen beieinander in derselben Datei implementiert. Zusätzlich gibt es noch eine Datei, die die Programmierung des Regelalgorithmus enthält. Im Gesamten haben wir folgende verschiedenen Dateien.

• main.c			
• Tpu_func.c	Tpu_func.h	Tpu_def.h	(Time Processor Unit)
• Qsm_func.c	Qsm_func.h	Qsm_def.h	(Queued Serial Modul)
• Sim_func.c	Qsm_func.h	Qsm_def.h	(System Integration Modul)
• Reg_func.c	Reg_func.h	Reg_def.h	(Regleralgorithmus)

Es war das Ziel, möglichst keine grossen Dateien zu haben und so die Lesbarkeit der Programme zu verbessern. Aus diesem Grund wurden alle Hardwareadressen und sonstigen hardware-abhängigen Definitionen ausgegliedert und in einer speziellen H-Datei gespeichert.

Diesen Dateien wurden die Namen mit der Endung „_def.h“ gegeben.

Im folgenden werden kurz die verschieden Routinen vorgestellt. Es geht darum, die Aufgabe der Funktion und deren Parameter zu erklären. Für genauere Informationen sind die Listings im Anhang beigelegt.

9.2.1. Tpu_func.c

```
//Kanal 0
void Init_Geschwindigkeitssollwert();
unsigned short Read_Geschwindigkeitssollwert();

//Kanal 1
void Init_Steuersollwert();
unsigned short Read_Steuersollwert();

//Kanal 2
void Init_Gewichtssollwert();
unsigned short Read_Gewichtssollwert();
```

Bemerkungen:

Dies sind die Funktionen, um den jeweiligen Kanal so zu konfigurieren, dass man die drei PWM-Signale vom Empfänger einlesen kann. Der Rückgabewert ist ein vorzeichenloser 16-Bit Wert.

Multipliziert mit der Timer1-Basis ergibt dies die Pulsweite, welche die Hebelstellung auf der Fernbedienung repräsentiert.

```
//Kanal 3
void Init_Geschwindigkeit();
unsigned short Read_Geschwindigkeit()
```

Bemerkungen:

Die Funktionen für Kanal 3 lesen Pulse von einem Geschwindigkeitsmessglied ein. Dieser Sensor ist auf unserem Motorrad zwar noch nicht eingebaut, doch die Funktionen stehen bereits zur Verfügung.

```
//Kanal 8
void Init_Fahrtregler();
Update_Fahrtregler(unsigned short Pulsbreite);
//Kanal 9
void Init_Servo1();
Update_Servo1(unsigned short Pulsbreite);
//Kanal 10
void Init_Servo2();
Update_Servo2(unsigned short Pulsbreite);
```

Bemerkungen:

Mit diesen drei Init-Funktionen werden der Fahrtregler, der Servo1 für die Lenkung und der Servo2 für die Gewichtsverlagerung eingerichtet. Die jeweiligen Update-Funktionen verlangen eine vorzeichenlose 16-Bit Zahl, um die Pulsbreite der Ausgangs PWM-Signale festzulegen. Die Ausgangswerte werden begrenzt. Man vergleicht die Werte mit den möglichen Maximal-und Minimalwerten. Dies ist so eingerichtet, dass die Servos an den Anschlägen halt machen.

```
//Kanal 11
void Init_SampleRate(unsigned short Periode);
void Update_Pulsbreite(unsigned short Pulsbreite);
void Init_Convst();
void Convst();
```

Bemerkungen:

Kanal 11 wird benutzt um Pulse an den AD-Wandler auszugeben. Mit jedem Pulse (High-Low-High Übergang) startet der AD7858 eine neue Wandlung. Es gibt nun zwei Möglichkeiten. Mit der Funktion Init_SampleRate() werden periodische Pulse ausgegeben. Die Funktion Init_Convst() initialisiert den Kanal 11 so, dass man mit der Funktion Convst() einzelne Startpulse ausgeben kann. Dies ist hauptsächlich für Tests nützlich.

```
//Kanal 15
void Init_DIO();
void Force_High();
void Force_Low();
```

Diesen Kanal verwendet man bei Testphasen. Um auf einfache Weise die Ausführungszeit des Regleralgorithmus zu messen, werden die Werte des TP15 Pins auf dem Oszilloskop dargestellt. Am Anfang der Routine wird der Pin mit Force_High() auf 5V gesetzt. Nach Beenden der Berechnungen setzt man den Wert wieder auf Low. Auf dem Oszilloskop liest man darauf die benötigte Zeit ab.

9.2.2. Qsm_func.c

```
void Reset_TransRam();
void Set_TransRam(unsigned short data, int pos);
```

 © Copyright 1998 SA

```
void Reset_ReceiveRam();
unsigned short Read_ReceiveRam(int pos);
```

```
void Reset_ContrRam();
void Set_ContrRam(unsigned short data,  int pos);
```

Bemerkungen:

Die oben aufgeführten Funktionen sind Hilfsroutinen für das Beschreiben und Lesen der drei Ringpuffer. Mit dem Parameter ‘data’ übergibt man einen Wert, der an die Stelle ‘pos’ im Ringpuffer geschrieben wird. Ist der Wert von ‘pos’ -1, so wird der ganze Puffer mit demselben ‘data’ beschrieben.

```
void Init_QSPI();
unsigned short Read_QSPI()
```

Bemerkungen:

Dies sind Funktionen, um die synchrone serielle Schnittstelle zu starten. Dabei wird der Lesepuffer als Ringpuffer konfiguriert. 16Bit Daten werden übertragen. Nach jeder Übertragung wird ein Delay abgewartet von 67.6us. Der serielle Clock wird mit 493kHz getaktet.

Beim Auslesen der Puffer wird über die letzten 16 Werte der Mittelwert gebildet.

9.2.3. Sim_func.c

```
void SetVBR();
void SetISRAdr(int VecNum, func_ptr func);
void Init_PeriodicTimer(unsigned int CountValue, func_ptr func)
```

Bemerkungen:

Die Hilfsfunktion SetVBR() setzt das Vektor Basis Register auf den Wert VBR_ADR, welcher in der Datei Sim_def.h definiert ist.

Mit der Funktion ‘SetISRAdr’ wird die Routine ‘func’ an der Position ‘VecNum’ in der Exception Tabelle eingetragen. ‘func_ptr’ ist ein Zeiger auf entsprechende Funktionen.

Die dritte Funktion ‘Init_PeriodicTimer’ automatisiert den ganzen Vorgang und verwendet auch die obigen Hilfsfunktionen. Als Parameter wird die Periodendauer und ein Zeiger auf die Service Routine übergeben.

9.2.4. Reg_func.c

```
void Init_PortE();
unsigned short Read_PortE();
```

Bemerkungen:

Um die Modusumschaltung mit dem Schalter 2 auf der Zusatzplatine erkennen zu können, wird mit diesen Funktionen der Port E konfiguriert und ausgelesen.

```
ISR_Regler();
```

Bemerkung:

Diese Funktion enthält den eigentlichen Regleralgorithmus.

10. Zusammenfassung

Das Ziel des Projektes "Autonomes Zweirad" ist es, ein freifahrendes Modell bereitzustellen, welches mit einem Regler ausgerüstet ist, um die stabile Gleichgewichtslage halten zu können. Das Halten der Gleichgewichtslage sollte auch gewährleistet sein, beim Befahren von verschiedenartigsten Trajektorien.

An welchem Punkt sind wir nun am Ende unserer Semesterarbeit angelangt? Welche Arbeiten wurden ausgeführt und in welchem Zustand kann das Motorrad einer weiterführenden Arbeit übergeben werden?

In einem ersten Schritt wurde eine Evaluation von verschiedenen Zweiradmodellen vorgenommen. Informationen zu diversen Ausführungsmöglichkeiten wurden gesammelt. Anhand dieser Grundlage entschied man sich, ein Modellbaumotorrad zu verwenden. Die handliche Grösse dieses Modells lässt es auch zu, im Hörsaal gewisse Demos vorzuführen.

Ebenfalls mussten Überlegungen gemacht werden über die Sensoren und Aktoren an unserem System. Die Messung des Lagesignals erfordert aufgrund der Dynamik eines Zweirades eine schnelle und exakte Aufnahme des momentanen Bewegungszustandes. Diese Information wird von einem Neigungssensor geliefert. Die Stellorgane bestehen aus Modellbaukomponenten. Zum Einen sind dies zwei Servos und zum Anderen ein Fahrtregler für die Ansteuerung des Motors. All diese Komponenten mussten im Motorrad untergebracht werden. Ebenfalls wurde eine Halterungsplattform für den Zielrechner aufgebaut.

Weitere Umbauarbeiten entstanden dadurch, dass man das ursprüngliche Fahrverhalten des Motorrades abgeändert hat. Das gekaufte Modell wurde durch Gewichtsverlagerung mit einem frei beweglichen Lenkrad gesteuert. Für den Regler war es aber notwendig, die Lenkung als Steuergrösse zu verwenden. Man baute also die Vorderradgabel so um, dass sie durch einen Servo kontrollierbar wurde.

Eine weitere Aufgabe bestand darin, das Motorrad in ein mathematisches Modell abzubilden. Ein Mehrkörpersystem, wie es ein Zweirad ist, führt auf ein Funktionaldifferentialgleichungssystem, welches nur mit speziellen DAE-Tools simuliert werden kann. Für die Reglerauslegung ist diese Modellierung zu kompliziert. Darum musste ein passendes Einkörpersystem gefunden werden, welches sich ähnlich einem Zweirad verhält. Als Grundmodell wurde dabei ein bewegtes invertiertes Pendel, welches einer Trajektorie mit gleichbleibender Geschwindigkeit folgt, verwendet. Das Inklinometer hat eine Einschwingzeit, welche in der Modellierung berücksichtigt wurde. Das Verhalten wurde aus der Sprungantwort auf ein System zweiter Ordnung übertragen. Daneben ist das Stellsignal sowohl in der Geschwindigkeit als auch in der Bewegung beschränkt. Gesamthaft ergibt sich ein System vierter Ordnung mit einem instabilen Pol. Die einzige Möglichkeit diese Strecke mit einem linearen Regler zu stabilisieren, ist ein Zustandsregler mit Beobachter.

Für die Implementierung des synthetisierten Reglers musste die Zielhardware bereitgestellt werden. Eingesetzt wurde ein Evaluation Board mit einem Microcontroller MC68332 von Motorola. Zusätzlich war noch ein Erweiterungsboard nötig, um die Versorgungsspannung für das Prozessorboard bereitzustellen und einen 8 Kanal 12-Bit AD-Wandler unterzubringen. Der AD-Wandler wandelt die analogen Signale des Neigungssensors um, so dass sie vom Prozessor verarbeitet werden können. Die übrigen Signale von der Fernsteuerung konnten leicht durch den Controller eingelesen werden. Das Prozessorboard mit zusätzlichem Versorgungsaufbau wurde im Steuerungsmodus ausgetestet und lief einwandfrei.

Die entwickelte Software stellt sämtliche Serviceroutinen bereit, um den Microcontroller für seine Aufgaben zu konfigurieren. Dabei wurde das Einlesen der Signale von der Fernsteuerung und der Neigungswerte vom AD-Wandler über die serielle Schnittstelle, sowie das Ausgeben der Stellgrössen auf die Aktoren berücksichtigt. Ebenfalls wurde ein Interrupt Timer eingerichtet, um zu garantieren, dass der Regleralgorithmus zu festgelegten Zeitpunkten gestartet wird. Die Grundstruktur des Regleralgorithmus steht bereit für den Test von weiteren Regelungs-konzepten.

Es besteht nun ein vollständiges Motorrad mit Prozessor und eigener Energiversorgung. Die einzelnen Subsysteme sind alle getestet und lauffähig. Leider konnte das Gesamtsystem nicht mehr getestet werden, so dass die Funktionstüchtigkeit des Regleralgorithmus unter realen Bedingungen nicht mehr verifiziert werden konnte.

 © Copyright 1998 SA

11. Ausblick

Die weitere Bearbeitung dieses Projektes soll die folgenden Schritte beinhalten:

Das umgebaute Motorrad konnte nicht im Regelungsmodus mit Prozessorplatine und Neigungssensor erprobt werden. Anhand von Tests mit dem bestehenden mechanischen System und dem entwickelten Regler soll eine Abschätzung der Fehlerquellen und Optimierungsmöglichkeiten gemacht werden.

Für den Reglerentwurf ist als weiteres Ziel die Umstellung von einem SISO- auf ein MIMO-System ins Auge zu fassen. Verschiedene Sensoren werden die benötigten Informationen liefern und mehrere Aktoren werden auf das Motorrad einwirken, um es zu stabilisieren. Als weiterer Sensor kommt in einem ersten Schritt ein Geschwindigkeitsmessglied zur Erfassung der Geschwindigkeit und deren Änderung hinzu. Zur Verbesserung der Information bezüglich des Kippens des Motorrades ist ein Gyrometer zur Erfassung der Winkelgeschwindigkeit einzusetzen. Einen weiteren Neigungssensor zu verwenden, verspricht nicht den gleichen Erfolg.

Die Regelung wird als weitere Ausgangsgrössen die Geschwindigkeit und die Gewichtsverlagerung verwenden, welche bereits hardwaremässig vorhanden sind. Die Wirkung der Auslenkung der Batterie unter dem Motorrad als Gewichtverlagerung muss in diesem Zusammenhang überprüft werden.

Die Modellierung als Einkörpersystem entspricht eigentlich nicht dem realen Wesen des Motorrades. Deshalb ist das bestehende Modell allenfalls durch eine Modellierung als vereinfachtes Vielkörpersystem [Get] zu ersetzen. Dazu sollten bei verschiedenen Instituten der Mechanik und der Physik weitere Informationen angefordert werden.

Die Identifikation des Inklinometers als System 2.Ordnung ist als einer der kritischen Punkte zu betrachten. Obschon eine verbesserte Identifikation sehr aufwendig werden könnte, wird man diesen Schritt nicht umgehen können.

Inwieweit die seitliche Auslenkung der Vorderachse zur Steuerung und Stabilisierung des Motorrades verwendet werden kann, hängt in erster Linie von der Machbarkeit der entsprechenden Modellierung ab. Dieser Schritt wird aber erst zu einem späteren Zeitpunkt realisierbar sein.

Das nichtlineare Verhalten des Motorrades lässt im weiteren auch einen Übergang zu nichtlinearen Reglerkonzepten zu.

Die Zielsetzung, ein lauffähiges Vorführmodell für eine Vorlesung zu entwickeln, muss immer im Mittelpunkt stehen bleiben.

12. Anhang

12.1. Literaturverzeichnis

[CPU32RM] *Central Processor Unit*, Referenzhandbuch, Motorola 1989

[Get] Neil Getz, *Control of Balance for a Nonlinear Nonholonomic Nonminimum Phase Model of a Bicycle*, American Control Conference, Baltimore, 1994

[Gra] *Graupner Modellbaukatalog*, 1997

[Hi-C] *Hi-Cross V2.7 Cross Development System*, Reference and User's Manual, 1994

[Mal] J. C. Maltbaek, *Dynamics in Engineering*, Ellis Horwood Limited, 1988

[MC68332UM] *Benutzerhandbuch MC68332*, Motorola 1990

[M68332EVSP] *User's Manual Promotional Evaluation System*, Motorola 1990

[Pai] Bruno Paillard, *Guidance Control and Trajectory Stabilization of a Bicycle*, University of Sherbrooke, Mechatronics Laboratory, 1997 http://www.gel.usherb.ca/mecatronique/educ/velo_an.html

[Pew] Pewatron AG, *Katalog Potentiometer Sensoren*, Wallisellen, 1997

[Ras] S. Neil Rasband, *Dynamics*, Malabar, Florida, 1991

[SIM32UM] *System Integration Module*, Benutzerhandbuch, Motorola 1989

[TPURM] *Time Processor Unit*, Referenzhandbuch, Motorola 1990

[Wis] Dirk von Wissel, *DAE Control of Dynamical Systems: Example of a Riderless Bicycle*, École des Mines de Paris, 1996

 © Copyright 1998 SA

12.2. Verwendete Variablen

α	Kippwinkel des Pendels bzw. Motorrades aus der Vertikalen (positiv nach rechts)
β	Fahrtrichtung des Hinterrades
γ	Winkel der Lenkung aus der Geradeausstellung (positiv nach rechts)
κ	Krümmung der Fahrkurve
φ_h, φ_v	Winkellage des Hinter- und Vorderrades
$\underline{\omega}_f$	Führungskreisfrequenz
A,B,C,D	Systemmatrizen für die Zustandraumdarstellung
F	Kraft
J	Trägheitsmoment des Pendels
J_h	Trägheitsmoment des Hinterrades
J_v	Trägheitsmoment des Vorderrades
R_v	Radius des Vorderrades
L	Drall
M	Moment
T	Zeitkonstante des Systems
$\underline{a}_f$	Zentripetalbeschleunigung
$\underline{a}_C$	Coriolisbeschleunigung
b	b-Koordinate binormal (senkrecht) auf den anderen Koordinaten
d	Dämpfung
g	Erdbeschleunigung
h	Abstand des Massepunktes vom Auflagepunkt des Pendels bzw. Höhe des Schwerpunktes über Boden beim senkrecht stehenden Motorrad
l	Radstand
m	Masse
n	n-Koordinate normal (senkrecht) auf der Trajektorie in der Auflageebene
r_h	Radius des Hinterrades
t	t-Koordinate tangential zur Trajektorie in der Auflageebene
v	Geschwindigkeit auf der Trajektorie
$\underline{v}_r$	Relativgeschwindigkeit
x, y	Ort des Auflagepunktes des Hinterrades

12.3. Wichtige Adressen

Autoren

Martin Neff
Rotbaldstrasse 4
CH-9532 Rickenbach
Tel. +41 71 923 70 46
email mneff@stud.ee.ethz.ch

Roland Odermatt
Aemättlistrasse 4
CH-6370 Stans
Tel. +41 41 610 78 80
email rodermat@stud.ee.ethz.ch

Alex Stucki
Hochstrasse 52
CH-8044 Zürich
Tel. +41 79 608 64 09
email uastucki@stud.ee.ethz.ch

Im folgenden sind sämtliche Adressen von Firmen aufgeführt, bei welchen wir Informationen und Produkte zu dieser Semesterarbeit angefordert haben.

Gyros

Gyration Inc.
12930 Saratoga Ave.
Bldg.C
Saratoga CA
email ebishop@gyration.com
Internet http://www.gyration.com
Ansprechperson Herr Ed Bishop ebishop@gyration.com

Ageltro (muRata Vertretung Schweiz)
Industrievertretungen und Beratung
Isenrietstrasse 19
CH-8617 Mönchaltdorf
Tel +41 1 948 13 14
Fax +41 1 948 17 69

Neigungssensoren

Pewatron AG
Hertistrasse 27
CH-8304 Wallisellen
Tel +41 1 830 29 44
Fax +41 1 830 51 57
email pewatron@access.ch
Internet http//www.access.ch/pewatron

Wyler AG
Neigungssysteme und -Instrumente
Im Hölderli
CH 8405 Winterthur
Tel +41 52 233 66 66

Crossbow Technology Inc.
41 E. Daggett Dr.
San Jose, CA 95134
Tel (408) 324 4830
Fax (408) 324 4840
email Info@xbow.com

 © Copyright 1998 SA

Internet http://www.xbow.com/index.htm
Ansprechperson Seth Randle seth@xbow.com

Contelec AG
Portstrasse 38
CH-2503 Biel
Tel +41 32 366 56 00
Ansprechperson Herr Schnorr

Geschwindigkeitssensor

HewlettPackard
Hauptsitz Schweiz
In der Luberzen 29
CH-8902 Urdorf ZH
Tel +41 1 735 71 11
Internet http://www.hp.com/HP-COMP/index.html

Elbatex
Handelsunternehmen für Elektronik-Bauteile
Hardstrasse 72
CH-5430 Wettingen
Tel +41 56 437 51 11
Fax +41 56 437 54 54
email info@elbatex.ch
Internet http:www.elbatex.ch

Basix AG
Hardturmstrasse 181
CH-8005 Zürich
Tel +41 1 276 11 11
Fax +41 1 276 12 34

EBV Electronic AG
Bernstrasse 394
CH-8953 Dietikon
Tel +41 1 745 61 61

Siemens Schweiz AG
Freilagerstrasse 40
CH-8047 Zürich
Tel +41 1 495 31 11
Fax +41 1 495 53 90
Internet http://www.siemens.ch/

Distrelec AG
Electronic Distributor
Grabenstrasse 6
CH-8606 Nänikon
Tel +41 1 944 99 11
Fax +41 1 944 99 88
email info@distrelec.com
Internet http://www.distrelec.ch

Baumer Electric AG
Hummelstrasse 17
CH-8500 Frauenfeld
Tel +41 52 728 11 22
Fax +41 52 728 11 44
Internet http:www.baumerelectric.ch

Litton
Precision Products
Gubelstrasse 28

CH-8050 Zürich
Tel +41 1 313 10 01
Fax +41 1 313 12 55

Honeywell AG
Hertistrasse 2
CH-8304 Wallisellen
Tel +41 1 839 25 25
Fax +41 1 831 03 14

Elektonik

Spörle Electronic
Cherstrasse 4, Postfach
CH-8152 Opfikon-Glattbrugg
Tel +41 1 874 62 62
Fax +41 1 874 62 00

Modellbau

Meili Modellbau
Zentrum Felsenburg
Thundorfstrasse 4
CH-8500 Frauenfeld
Tel +41 52 721 51 29
Fax +41 52 721 50 24
Ansprechperson Herr Meili

Graupner Schweiz
Ringstrasse 24
CH-8317 Tagelswangen
Tel +41 52 354 20 58
Ansprechperson Herr Lutz

12.4. Datenblätter

12.4.1. Modellbau

- Graupner, FM 314 8 Kanal Fernlenkset (6 Seiten) Seite 89-94
- Graupner, Bedienungsanleitung RC-Elektro-Motorrad (4 Seiten) Seite 95-98
- Robbe, Bedienungsanleitung Fahrtregler (2 Seiten) Seite 99-100
- Datenblatt Schnelladegerät (1 Seite) Seite 101
- Datenblätter Nickel Cadmium Akkumulatoren (2 Seiten) Seite 103-104

12.4.2. Sensoren

- Pewatron, Inklinometer PMP-S20T-SW2 (1 Seite) Seite 105

12.4.3. Elektronik Bauteile Zusatzboard

- Analog Devices, Datenblatt AD-Wandler AD7858/AD7858L (34 Seiten) Seite 107-140
- Analog Devices, Datenblatt Referenzspannung REF-02 (8 Seiten) Seite 141-148
- National Semiconductors, Datenblatt Spannungsregler LM2940 (18 Seiten) Seite 149-166

 © Copyright 1998 SA

12.5. MatLab-File für Regler und Simulation

Das M-File param.m enhält sämtliche Parameter und Berechnungen um alle Simulation auf Simulink durchzuführen. Für die Reglerimplementation wurde eine normierte Form von param.m verwendet. Bis auf wenige Änderungen entspricht das M-File normpara.m dem ursprünglichen File und wird hier deshalb nicht nochmals gedruckt.

Als weitere Hilfsmittel wurden vier weitere M-Files geschrieben, welche das Plotten der Resultate vereinfacht haben. Diese Files werden hier nicht abgedruckt, da sie auch auf der CD zur Semesterarbeit gespeichert sind. Die Namen der Files lauten:

plotink.m, plotmot.m, plotreg.m, plotpol.m

param.m

```
% Simulink-Modell des Motorrades
% ******************************

% Begonnen am 18.12.97
% Zuletzt bearbeitet am 05.02.98
%
% Semesterarbeit Autonomes Motorrad
% Martin Neff, Roland Odermatt, Alex Stucki


% Parameter des Modells
% *********************

% statische Parameter
% -------------------

% Konstanten
g=9.8067;        % Erdbeschleunigung

% Geometrie
l=0.29;          % Radstand (aus Katalog)
h=0.114;         % Hoehe des Schwerpunktes ueber Boden (gemessen
                 % 26.01.98)
rh=0.064;        % Radius des Hinterrades (aus Katalog)
rv=0.059;        % Radius des Vorderrades (aus Katalog)
rhf=0.044;       % Radius der Felge des Hinterrades (gemessen 19.01.98)
rvf=0.04;        % Radius der Felge des Vorderrades (gemessen 19.01.98)

% Masse
m=2.200;         % Gesamtgewicht (gewogen 26.1.98)
mh=0.205;        % Masse des Hinterrades (gewogen 16.1.98)
mv=0.160;        % Masse des Vorderrades (gewogen 16.1.98)

% Messung des Traegheitsmoments der Raeder
rg=0.008/2;      % halbe Dicke des Gewichts, welches ans Rad geklebt
                 % wurde
mg=0.05;         % Masse des Gewichts
% Vorderrad
pv=5;            % Anzahl Perioden der Schwingung am Vorderrad
rgv=rv+rg;       % Abstand des Mittelpunkts des Gewichts vom Drehpunkt
                 % des Vorderrrades
Tv1=4.25; Tv2=4.28; Tv3=4.30; Tv4=4.33;
                 % Messwerte Vorderrad fuer den Zeitverbrauch von pv
                 % Perioden (gemessen 19.01.98)
Tv=mean([Tv1; Tv2; Tv3; Tv4]);
                 % Mittelwert dieser Messwerte
Jv=mg*g*rgv/(pv*2*pi/Tv)^2-mg*rgv^2;
                 % Traegheitsmoment des Vorderrades (gemessen 19.01.98)
Jvmin=1/2*mv*rv^2;
                 % untere Beschraenkung des Traegheitsmoments des
                 % Vorderrades (als Zylinder)
Jvmax=1/2*mv*(rv^2+rvf^2);
                 % obere Beschraenkung des Traegheitsmoments des
```

```
                  % Vorderrades (als Hohlzylinder)
% Hinterrad
ph=3;             % Anzahl Perioden der Schwingung am Hinterrad
rgh=rh+rg;        % Abstand des Mittelpunkts des Gewichts vom Drehpunkt
                  % des Hinterrades
Th1=2.78; Th2=2.80; Th3=2.72; Th4=2.72;
                  % Messwerte Hinterrad fuer den Zeitverbrauch von ph
                  % Perioden (gemessen 19.01.98)
Th=mean([Th1; Th2; Th3; Th4]);
                  % Mittelwert dieser Messwerte
Jh=mg*g*rgh/(ph*2*pi/Th)^2-mg*rgh^2;
                  % Traegheitsmoment des Hinterrades (gemessen 19.01.98)
Jhmin=1/2*mh*rh^2;
                  % untere Beschraenkung des Traegheitsmoments des
                  % Hinterrades (als Zylinder)
Jhmax=1/2*mh*(rh^2+rhf^2);
                  % obere Beschraenkung des Traegheitsmoments des
                  % Hinterrades (als Hohlzylinder)

% Messung des Traegheitsmoments des Motorrades um die Kippachse
rf=0.12;          % Abstand des Angriffspunktes der Federwaage von der
                  % Drehachse
af=4*pi;          % Auslenkung des Motorrades aus der Nulllage [rad]
Ff=210/102;       % Kraft auf die Federwaage, umgerechnet von Gramm in
                  % Newton [N]
cf=Ff*rf/af;      % Federkonstante der Gummizuege
% Motorrad
pm=3;             % Anzahl Perioden der Schwingung
Tm1=13.58; Tm2=13.66; Tm3=13.66; Tm4=13.68;
                  % Messwerte Motorrad fuer den Zeitverbrauch von pm
                  % Perioden (gemessen 26.01.98)
Tm=mean([Tm1; Tm2; Tm3; Tm4]);
                  % Mittelwert dieser Messwerte
Js=cf/(pm*2*pi/Tm)^2;
                  % Traegheitsmoments des Rahmens um die Kippachse durch
                  % den Schwerpunkt (gemessen 26.01.98)
Jsmin=1/2*m*(h/2)^2;
                  % untere Beschraenkung des Traegheitsmoments des Rahmens
                  % (als Zylinder mit Radius halbe Schwerpunktshoehe)
Jsmax=1/2*m*h^2;  % obere Beschraenkung des Traegheitsmoments des Rahmens
                  % (als Zylinder mit Radius Schwerpunktshoehe)
J=m*h^2+Js;       % Traegheitsmoment des Motorades um die Kippachse durch
                  % die Auflagepunkte

% dynamische Parameter
% --------------------

% Motorrad
v=3;              % Geschwindigkeit des Motorrades
Lh=Jh*v/rh;       % Drall des Hinterrades
Lv=Jv*v/rv;       % Drall des Vorderrades
L=Lh+Lv;          % Gesamtdrall der Raeder

% Inklinometer
omei=140;         % Frequenz des Inklinometers (geschaetzt nach
                  % Sprungantwort)
di=0.25;          % Daempfung des Inklinometers (geschaetzt nach
                  % Sprungantwort)
Ui=5;             % Spannung am Inklinometer [V] (gemessen 26.01.98)
                  % Messpositionen des Inklinometers in Radiant (positiv
                  % fuer das Kippen nach rechts)
Mi=[-22 -20 -18 -16 -14 -12 -10 -8 -6 -4 -2
2 4 6 8 10 12 14 16 18 20]*pi/180;
                  % Messresultate an den Messpositionen abzueglich Offset
                  % (gemessen 26.01.98)
Ri=[2.93 2.88 2.85 2.81 2.775 2.74 2.70 2.67 2.63 2.58 2.55 2.46 2.42
2.38 2.34 2.30 2.26 2.23 2.18 2.14 2.11]-Ui/2;
Si=Ri./Mi;        % Verstaerkung des Inklinometers an den versch.
```

 © Copyright 1998 SA

```
                  % Messpunkten
ki=mean(Si);      % Verstaerkung des Inklinometers [V/rad]

% Servo
ks=1;             % Verstaerkung des Servos
vs=40/0.15*pi/180;
                  % Geschw. des Servos (0.15 Sekunden fuer 40 Grad) (aus
                  % Graupner Katalog)
as=18*pi/180;     % Anschlag des Servos bzw. der Lenkung


% Modellierung des Motorrades und der Sensoren / Aktoren
% *******************************************************

% Motorrad
T=sqrt(J/(m*g*h));
                  % Zeitkonstante
d=0.01;           % Daempfung
Am=1/T*[0 1; 1 -d];
b1=v*Jv/(J*rv);
b2=-v^2/(g*l)*(1-g*l/(m*h)*(Jh/rh+Jv/rv));
Bm=[b1/T; b2/T];
Cm=[1 0];
d1=-v^2/(l*g);
Dm=[d1];

numm=[d1 1/T*(d1*d+b1) 1/T^2*(d*b1+b2-d1)];
denm=[1 d/T -1/T^2];
%[numv, denv]=ss2tf(Am,Bm,Cm,Dm,1)
                  % zur Verifikation obiger Berechnungen

[zm, pm, km]=ss2zp(Am,Bm,Cm,Dm,1);
                  % Pole und Nullstellen des Motorrades

% Inklinometer
numi=ki;          % Zaehler der Utf des Inklinometers
deni=[1/omei^2 2*di/omei 1];
                  % Nenner der Utf des Inklinometers
[Ai,Bi,Ci,Di]=tf2ss(numi,deni);   % Zustandsraumdarstellung Inklinometer
                  % Sprung auf das Inklinometer
stpi=20;          % Sprunghoehe des Eingangs in Grad
Ti=[0; 0.1; 0.12; 0.5];
                  % Zeitvektor zum Eingangssignal U
Si=[0;0;stpi*pi/180;stpi*pi/180];
                  % Eingangssignal U

% Zustandsraumdarstellung der Gesamtstrecke (Motorrad und Inklinometer)
A=[Am zeros(size(Am,1),size(Ai,2)); Bi*Cm Ai];
B=[Bm; Bi*Dm];
C=[zeros(size(Cm)) Ci];
D=[0];

[z, p, k]=ss2zp(A,B,C,D,1);
                  % Pole und Nullstellen des Motorrades


% Reglerentwurf
% *************

% Zustands-Regler mit Beobachter und gedaempfter Ausgangsrueckfuehrung
pb=real([z(1) p(2) p(3) p(4)]);
                  % Basispole fuer die Polfestlegung
pbi=imag([z(1) p(2) p(3) p(4)]);
                  % Imaginaerteil der Basispole fuer die Polfestlegung
fg=4;             % Faktor fuer die Geschw. des Beobachters
pg=fg*pb+i*pbi;   % Pole zum Festlegen des Ausgangsrueckfuehrvektors
%pg=[fg*pb(1:2) pb(3:4)]+i*pbi;
g=place(A',C',pg)'
                  % Ausgangsrueckfuehrfaktoren
```

```
fk=2;            % Faktor fuer die Geschw. der Zustandsregelung
pk=[fk*pb(1:2) pb(3:4)]+i*pbi;
                 % Pole zum Festlegen des Zustandsrueckfuehrvektors
%pk=fk*pb+i*pbi;
k=place(A,B,pk)  % Zustandsrueckfuehrfaktoren, damit die Pole richtig
                 % liegen
k=[k(1) k(2) 0 0]

% Regler in Zustandsraumdarstellung
cr=[1 0 0 0];    % Ausgangsvektor fuer den Winkel der Schraeglage
yr=45*pi/180;    % Maximalwert der Schraeglage
rr=45*pi/180;    % Maximalwert des Regeleingangs
vr=-yr/rr/(cr*inv(A-B*k)*B);
                 % Vorfaktor fuer den Regeleingang
Ar=[A-g*C-B*k];
Br=[g vr*B];
Cr=[-k];
Dr=[D vr];

% Transformation in diskrete Zustandsraumdarstellung
Ts=0.005;
[Ad Bd]=c2d(Ar,Br,Ts);
Cd=Cr;
Dd=Dr;
```

12.6. C-Programme

12.6.1. main.c

```
#include "tpu_func.h"
#include "qsm_func.h"
#include "sim_func.h"
#include "reg_func.h"

void main()
{
      unsigned int i;
      Init_PortE();                        //Port E fuer Modusschalter
      Init_Geschwindigkeitssollwert();     //Kanal0
      Init_Steuersollwert();               //Kanal1
      Init_Fahrtregler();                  //Kanal8
      Init_Servo1();                       //Kanal9
      Init_SampleRate(54);                 //Kanal11
      Init_QSPI();                         //Serielle Schnittstelle
                                           //initialisieren

      Init_PeriodicTimer(0x0029, ISR_Regler);  //Periode auf 5ms setzen
      while(1)
      {
      };
}
```

12.6.2. Tpu_func.c

```
#include "tpu_def.h"

// Eingaenge
/*********************************************************************
Funktionen zur Handhabung des Kanal 0
Messung der Pulsbreite des PWM-Signals der Fernsteuerung (PPWA)
(Input = Geschwindikgeitssollwert)
*********************************************************************/
void Init_Geschwindigkeitssollwert()
{
      pram->chan0.p0= 0x0007; //TCR1 als Time Basis gesetzt, Steigende
```

 © Copyright 1998 SA

```
                                //Flanke erkennen
        pram->chan0.p1= 0x0100; //Anzahl Pulsweiten die aufsummiert werden
        pram->chan0.p4= 0xFF00; //Accum_Rate; alle 0.5ms wird Accum
                                //aufgefrischt

        //Kanalfunktion auf $F(PPWA) setzen
        cfsr3=cfsr3|0x000F;

        //Modus 2 24-Bit Pulsbreiten aufsummieren
        hsqr1=hsqr1|0x0002;

        //Host Service Request absetzen zur Initialisierung
        hsrr1=hsrr1|0x0002;

        //Kanalprioritaet auf High setzen
        cpr1=cpr1|0x0003;
}

unsigned short Read_Geschwindigkeitssollwert()
{
        return pram->chan0.p5;
}


/*********************************************************************
Funktionen zur Handhabung des Kanal 1
Messung der Pulsbreite des PWM-Signals der Fernsteuerung (PPWA)
(Input = Steuersollwert)
*********************************************************************/
void Init_Steuersollwert()
{
        pram->chan1.p0= 0x0007; //TCR1 als Time Basis gesetzt, Steigende
                                        //Flanke erkennen
        pram->chan1.p1= 0x0100; //Anzahl Pulsweiten die aufsummiert werden
        pram->chan1.p4= 0xFF00; //Accum_Rate; alle 0.5ms wird Accum
                                //aufgefrischt

        //Kanalfunktion auf $F(PPWA) setzen
        cfsr3=cfsr3|0x00F0;

        //Modus 2 24-Bit Pulsbreiten aufsummieren
        hsqr1=hsqr1|0x0008;

        //Host Service Request absetzen zur Initialisierung
        hsrr1=hsrr1|0x0008;

        //Kanalprioritaet auf High setzen
        cpr1=cpr1|0x000C;
}

unsigned short Read_Steuersollwert()
{
        return pram->chan1.p5;
}


/*********************************************************************
Funktionen zur Handhabung des Kanal 2
Messung der Pulsbreite des PWM-Signals der Fernsteuerung (PPWA)
(Input = Sollwert der Gewichtsverlagerung)
*********************************************************************/
void Init_Gewichtssollwert()
{
        pram->chan2.p0= 0x0007; //TCR1 als Time Basis gesetzt, Steigende
                                //Flanke erkennen
        pram->chan2.p1= 0x0100; // Anzahl Pulsweiten die aufsummiert werden
        pram->chan2.p4= 0xFF00; // Accum_Rate; alle 0.5ms wird Accum
                                //aufgefrischt
```

```
        //Kanalfunktion auf $F(PPWA) setzen
        cfsr3=cfsr3|0x0F00;

        //Modus 2 24-Bit Pulsbreiten aufsummieren
        hsqr1=hsqr1|0x0020;

        //Host Service Request absetzen zur Initialisierung
        hsrr1=hsrr1|0x0020;

        //Kanalprioritaet auf High setzen
        cpr1=cpr1|0x0030;
}

unsigned short Read_Gewichtssollwert()
{
        return pram->chan2.p5;
}


/*********************************************************************
Funktionen zur Handhabung des Kanal 3
Messung der Zeit zwischen 2 Input Transitions (PMA)
(Input = Geschwindigkeit)
*********************************************************************/

void Init_Geschwindigkeit()
{
        pram->chan3.p0=0x0006;
        pram->chan3.p1=0x00FF;
        pram->chan3.p3=0x00FF;

        //Kanalfunktion auf $B (PMA) setzen

cfsr3=cfsr3|0xB000;

        //Pulse zählen
        hsqr1=hsqr1|0x0040;

        //Host Service Request absetzen
        hsrr1=hsrr1|0x0040;

        //Kanalpriorität auf High setzen
        cpr1=cpr1|0x00C0;
}

unsigned short Read_Geschwindigkeit()
{
return pram->chan3.p5;
}


//Ausgaenge

/*********************************************************************
Funktionen zur Handhabung des Kanal 8
Generierung eines PWM Signals zur Ansteuerung des Fahrtreglers
*********************************************************************/

/*Initialisierung
Timer1 Basis ist auf 1.901us (16.8333MHz/32) gesetzt.
Periodendauer = 20ms   = 10521*1.901us
High Time     = 0ms
*/
void Init_Fahrtregler()
{
        pram->chan8.p0 = 0x0092;      //TCR1 als Time Basis gesetzt, Force
                                      //Pin Low
        pram->chan8.p2 = 0x0000;      // High Time 0ms setzen
```

 © Copyright 1998 SA

```
        pram->chan8.p3 = 0x2919;      //Periodendauer auf 20ms setzen

        //Kanalfunktion auf $9(PWM) setzen
        cfsr1=cfsr1|0x0009;

        //Host Service Request absetzen zur Initialisierung
        hsrr0=hsrr0|0x0002;

        //Kanalprioritaet auf High setzen
        cpr0=cpr0|0x0003;
}

/*Uebergabe einer neuen Pulsbreite fuer Fahrtregler an Kanal8
High Time
   0% (kein Ausschlag)= ? = ?*1.901us
 100% (+18° Ausschlag)= ? = ?*1.901us
*/
void Update_Fahrtregler(unsigned short Pulsbreite)
{
        if (Pulsbreite < 566) Pulsbreite=566;
        pram->chan8.p2 = Pulsbreite;              //High Time setzen
}


/*********************************************************************
Funktionen zur Handhabung des Kanal 9
Generierung eines PWM Signals zur Ansteuerung der Lenkung
*********************************************************************/

/*Initialisierung
Timer1 Basis ist auf 1.901us (16.8333MHz/32) gesetzt.
Periodendauer = 20ms   = 10521*1.901us
High Time     = 0ms
*/
void Init_Servo1()
{
        pram->chan9.p0 = 0x0092;      //TCR1 als Time Basis gesetzt, Force
                                      //Pin Low
        pram->chan9.p2 = 0x0000; // High Time 0ms setzen
        pram->chan9.p3 = 0x2919; // Periodendauer auf 20ms setzen

        //Kanalfusnktion auf $9(PWM) setzen mit bitweiser Oder-Verknuepfung
        cfsr1=cfsr1|0x0090;

        //Host Service Request absetzen zur Initialisierung
        hsrr0=hsrr0|0x0008;

        //Kanalprioritaet auf High setzen
        cpr0=cpr0|0x000C;
}

/*Uebergabe einer neuen Pulsbreite fuer Fahrtregler an Kanal8
High Time
   0% (kein Ausschlag)= 1.47ms = 775*1.901us
 100% (+18° Ausschlag)= 1.62ms = 850*1.901us
-100% (-18° Ausschlag)= 1.33ms = 700*1.901us
*/

void Update_Servo1(unsigned short Pulsbreite)
{
        //Begrenzung
        if (Pulsbreite>840) Pulsbreite=840;
        if (Pulsbreite<700) Pulsbreite=700;

        pram->chan9.p2 = Pulsbreite;        //High Time setzen
}
```

```
/*********************************************************************
Funktionen zur Handhabung des Kanal 10
Generierung eines PWM Signals zur Ansteuerung der Gewichtsverlagerung
*********************************************************************/

/*Initialisierung
Timer1 Basis ist auf 1.901us (16.8333MHz/32) gesetzt.
Periodendauer = 20ms   = 10521*1.901us
High Time     = 0ms
*/
void Init_Servo2()
{
      pram->chan10.p0 = 0x0092;     // TCR1 als Time Basis gesetzt, Force
                                    //Pin Low
      pram->chan10.p2 = 0x0000; // High Time 0ms setzen
      pram->chan10.p3 = 0x2919; // Periodendauer auf 20ms setzen

      //Kanalfunktion auf $9(PWM) setzen mit bitweiser Oder-Verknuepfung
      cfsr1=cfsr1|0x0900;

      //Host Service Request absetzen zur Initialisierung
      hsrr0=hsrr0|0x0020;

      //Kanalprioritaet auf High setzen
      cpr0=cpr0|0x0030;
}

/*Neue Pulsbreite
High Time
   0% (kein Ausschlag)= ?ms = ?*1.901us
 100% (?° Ausschlag)= ?ms = ?*1.901us
-100% (?° Ausschlag)= ?ms = ?*1.901us
*/
void Update_Servo2(unsigned short Pulsbreite)
{
      //Begrenzung
      if (Pulsbreite > 978) Pulsbreite=978;
      if (Pulsbreite < 620) Pulsbreite=620;

      pram->chan10.p2 = Pulsbreite;             //High Time setzen
}


/*********************************************************************
Funktionen zur Handhabung des Kanal 11
Generierung von Pulsen zur Ausloesung der AD Wandlungen des AD7858
*********************************************************************/

/*Initialisierung
Timer1 Basis ist auf 1.901us (16.8333MHz/32) gesetzt.
Periodendauer = 1/fs
*/

//Initialisierungsfunktion zur Ausgabe periodischer Conversion Start
Pulsen
void Init_SampleRate(unsigned short Periode)
{
      pram->chan11.p0 = 0x0092;          // TCR1 als Time Basis gesetzt,
                                         //Force Pin Low
      pram->chan11.p2 = Periode-10;      // High Time = Periodendauer
      pram->chan11.p3 = Periode;         // Periodendauer auf 20ms setzen

      //Kanalfunktion auf $9(PWM) setzen mit bitweiser Oder-Verknuepfung
      cfsr1=cfsr1|0x9000;

      //Host Service Request absetzen zur Initialisierung
      hsrr0=hsrr0|0x0080;
```

 © Copyright 1998 SA

```
        //Kanalprioritaet auf High setzen
        cpr0=cpr0|0x00C0;
}

//Sample Rate veraendern
//Ts=Pulsbreite*1.908us
void Update_Pulsbreite(unsigned short Pulsbreite)
{
        pram->chan11.p2 = Pulsbreite;       //High Time setzen
}

//Initialisierungsfunktion zur Ausgabe von einzelnen Conversion Start
Pulsen
void Init_Convst()
{
        cfsr1|=0x8000;
        cpr0|=0x00C0;
        hsrr0|=0x0040;
}

//Einzelner Conversion Start Puls ausgeben
void Convst()
{
        unsigned int i;
        hsrr0|=0x0080; //Force Low
        for (i=0;i<100;i++); //Wait
        hsrr0|=0x0040; //Force High
}


/*********************************************************************
Funktionen zur Handhabung des Kanal 15
Messung der Zeit fuer die Abarbeitung eines Algorithmus
TP15 Pin auf High am Anfang
TP15 Pin auf Low am Schluss
Messung mit Osziloskop
*********************************************************************/

void Init_DIO()
{
        cfsr0|=0x8000;
        cpr0|=0xC000;
        hsrr0|=0x8000;
}

void Force_High()
{
        hsrr0|=0x4000;
}

void Force_Low()
{
        hsrr0|=0x8000;
}
```

12.6.3. Qsm_func.c

```
#include "qsm_def.h"

/*********************************************************************
Funktionen zur Handhabung des AD Wandlers an der QSPI Schnittstelle
*********************************************************************/

//Setzt die Word zum Senden auf Null
void Reset_TransRam()
{
```

```
        int i;
        for (i=0;i<16;i++) *(WORD*)(0xfffd20+i*2) = 0x0000;
}

//Setzt das Word an der Stelle pos auf data
//Falls pos= -1 wird das ganze TransRam mit data beschrieben
void Set_TransRam(unsigned short data,int pos)
{
        if (pos==-1)
        {
        int i;
        for(i=0;i<16;i++) *(WORD*)(0xfffd20+i*2)= data;
        }
        else *(WORD*)(0xfffd20+2*pos)= data;
}

//Setzt das Ram fuer den Empfang von Daten auf Null
void Reset_ReceiveRam()
{
        int i;
        for(i=0;i<16;i++) *(WORD*)(0xfffd00+i*2) = 0x0000;
}

//Liest das empfangene Word an der Stelle pos im ReceiveRam
unsigned short Read_ReceiveRam(int pos)
{
        return *(WORD*)(0xfffd00+2*pos);
}

//Setzt das Control Ram auf Null
void Reset_ContrRam()
{
        int i;
        for (i=0;i<16;i++) *(BYTE*)(0xfffd40+i) = 0x0000;
}

//Setzt das Byte an der Stelle pos auf data
//Falls pos= -1 wird das ganze ContrRam mit data beschrieben
void Set_ContrRam(unsigned short data,int pos)
{
        if (pos==-1)
        {
                int i;
                for(i=0;i<16;i++) *(BYTE*)(0xfffd40+i)=data;
        }
        else *(BYTE*)(0xfffd40+pos) = data;
}


void Init_QSPI()
{
        Reset_TransRam();       //Keine Daten senden
        Set_ContrRam(113,-1);   //Control Bytes setzen  //112

        *QMCR |=0x0001;                 //Interrupt Prioritaet low

        //Pin Control Register
        *QPAR= 0x0f;                    //PCS0/SS, MOSI,MISO werden fuer QSPI
                                        //gebraucht
        *QDDR= 0x0e;                    //Datenrichtung Output:PCS0/SS,SCK,MOSI
                                        //Input:MISO
        //QSPI Submodul Register
        *SPCR0= 0x83FF;                 //QSPI Master, 16 Bit, BAUD=17 =>
                                        //baudiv=34, SCK= 494kHz @ 16.8MHz,
                                        //CPOL=1, CPHA=1
        *SPCR1= 0x44ff;                 //QSPI disabled,DSCKL= 4us,DTL=68.4 us
        *SPCR2= 0x6f00;                 //Interrupts disabled, Wraparound Mode
        //enable QSPI
```

 © Copyright 1998 SA

```
        *SPCR1|=0x8000;              //Bit15 setzen

        while(!(*SPSR & _SPIF));     // warten bis SPIF gesetzt
}

unsigned short Read_QSPI()
{
        //Mitteln ueber die letzten 16 Werte
        unsigned long Summe=0;
        unsigned short Temp;
        int i;
        for (i=0;i<16;i++)
        {
                Temp= *(WORD*)(0xfffd00+2*i); //Werte aus dem Datenpuffer
                                              //lesen
                Temp &=0x0FFF;                //Oberste 4 Bits ausmaskieren
                Summe=Summe + Temp ;          //Aufsummieren
        }
return Summe/16;                              //Mittelwert bilden
}
```

12.6.4. Sim_func.c

```
#include "sim_func.h"
#include "sim_def.h"

void SetVBR()
{
        asm
        {
                MOVE.L     #VBR_ADR,D0        //Vector Base Register auf
                                              //VBR_ADR setzen
                MOVEC      D0,VBR
        }
}

//Eintragen der Interrupt Service Routine Adresse in der Exception Vector
//Table
void SetISRAdr ( int VecNum, func_ptr func)
        asm
        {
                MOVEC    VBR,A0;              //Vector Base Register in A0
                MOVE.W   VecNum,D0;           //Vektor Nummer in D0
                ANDI.L   #0x000000FF,D0;      //Vektor Nummer im Bereich von
                                              //0..255 ausmaskieren
                ASL      #2,D0;               //Vektor Nummer *4
                ADDA.L   D0,A0;               //VBR und Vektor Offset addieren
                MOVE.L   func,A1;             //Adresse der ISR in A1 legen
                MOVE.L   A1,(A0);             //Adresse der ISR in der Tabelle
                                              //abspeichern (Adresse A0)
        }
}

void Init_PeriodicTimer(unsigned int CountValue, func_ptr func)
{
        SetVBR();                //VBR auf 0x000000 setzen
        SetISRAdr(0x40,func);    //Interrupt Service Routine (ISR) in der
                                 //Exception Tabelle eintragen
                                 //Vektor Nummer ist 64 (=$40) (Erste User
                                 //Defined Vector)
        pitr &= 0x0000;
        pitr |= CountValue;      //Periode 5ms
        picr &= 0x0000;
        picr |= 0x0640;          //Interrupt Level 6, Vector $40
        asm
        {       ANDI.W #0xf0ff,SR;//Interrupt Levels unter 6 ausblenden
                ORI.W  #0x0500,SR;
```

```
    }
    mcr|= 0x0004;          //Interrupt arbitration field auf 4 setzen
}
```

12.6.5. Reg_func.c

```
#include "reg_def.h"

void Init_PortE()
{
    pepar &= 0x7F;    //Pin SIZ1 als I/O Pin definieren
    ddre  &= 0x7F;    //Pin SIZ1 als Input definieren
}

unsigned short Read_PortE()
{
    unsigned short tok;
    tok=testporte;
    tok&=0x0080;
    if (tok==0x0080)
          return 1;
    else
          return 0;
}

void ISR_Regler()
#pragma TRAP_PROC
{
    const DimX = 4;
    const DimU = 2;
    const DimY = 1;
    const float A [DimX] [DimX] = {{?,?,?,?},
                                   {?,?,?,?},
                                   {?,?,?,?},
                                   {?,?,?,?}};
    const float B [DimX] [DimU] = {?,?},
                                         {?,?},
                                         {?,?},
                                         {?,?}};
    const float C [DimY] [DimX] = {{?,?}};
    const float D [DimY] [DimU] = {{?}};
    static float x [DimX] = {0,0,0,0};
    float x_alt [DimX] ={0,0,0,0};
    float y [DimY]= {0};
    float u [DimU]={0,0};
    float alpha,u1_soll;
    unsigned int y1,v_soll;
    int i,j;

    if (Read_PortE()==1)
    {

          //Regleralgorithmus

          //Einlesen der Inputs
          //Neigungssensor
          alpha= Read_QSPI();
          u[0]=(alpha/2048)-1;

          //Steuersollwert
          u1_soll=Read_Steuersollwert();
          if (u1_soll> 996) u1_soll=996;
          if (u1_soll<576) u1_soll=576;
          u[1]=(u1_soll-786)/210;

          //Geschwindigkeitssollwert wird durchgeschlauft
          v_soll=Read_Geschwindigkeitssollwert();
```

 © Copyright 1998 SA

```
                //Alter Zustand sichern und neuer Zustand auf Null setzen
                for (i=0;i<DimX;i++)
                {
                      x_alt[i]= x[i];
                      x[i]=0;
                }

                //Neuer Zustand berechnen
                for (i=0;i<DimX;i++)
                {     for (j=0;j<DimX;j++)
                      {
                            x[i]=x[i]+ A[i][j]*x_alt[j];
                      }
                      for (j=0;j<DimU;j++)
                      {
                            x[i]=x[i]+B[i][j]*u[j];
                      }
                }
                //Neuer Ausgang berechnen
                for (i=0;i<DimY;i++)
                {     for (j=0;j<DimX;j++)
                      {
                            y[i]= y[i]+C[i][j]*x_alt[j];
                      }
                      for (j=0;j<DimU;j++)
                      {
                            y[i]=y[i]+D[i][j]*u[j];
                      }
                }

                //Ausgeben der Stellsignale

                //Servo1
                y1=786+(y[0]*210);
                Update_Servo1(y1);

                //Fahrtregler
                Update_Fahrtregler(v_soll);
          };

          if (Read_PortE()==0)
          {

                //Steuerung
                Update_Fahrtregler(Read_Geschwindigkeitssollwert());
                u1_soll=Read_Steuersollwert();
                Update_Servo1(Read_Steuersollwert());

                for (i=0;i<DimX;i++)
                {
                      x[i]=0;
                }
          };
    }
```