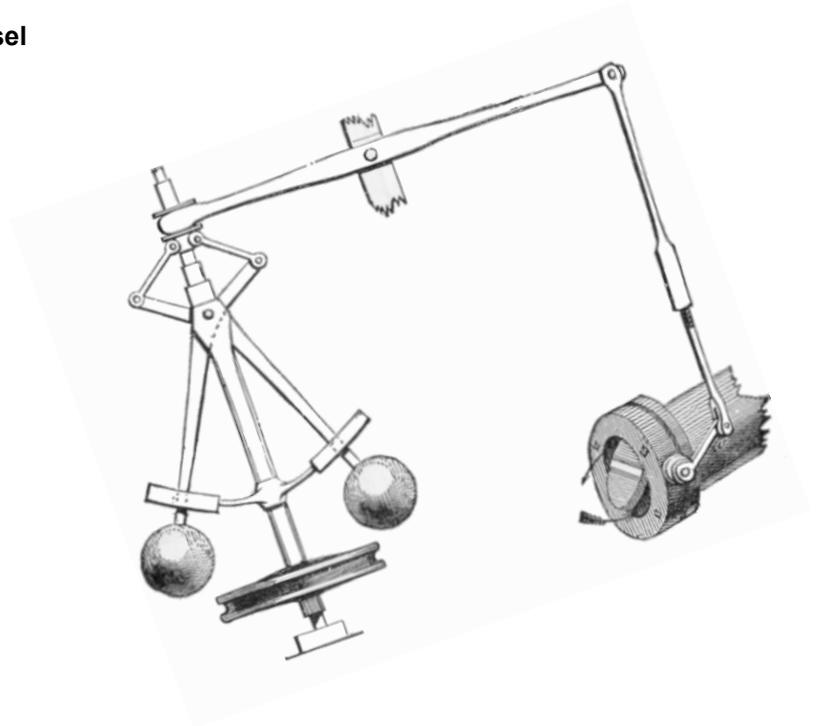
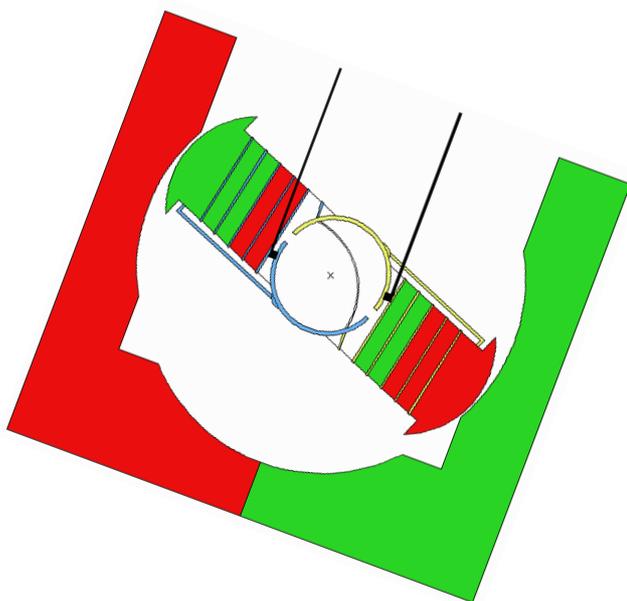


Berufsmaturitätsschule

Allgemeine Gewerbeschule Basel



Drehzahlregelung eines Elektromotors



Berufsmaturitätsarbeit
von
Christian Vögtli
und
Patrick Pfeifer

1	Abstract	2
2	Einleitung	3
2.1	Übersicht.....	3
2.2	Motivation.....	3
3	Regelungstechnik	4
3.1	Beispiele	4
3.1.1	Festwertregelung.....	4
3.1.2	Nachlaufregelung.....	5
3.2	Regelkreis	5
3.2.1	Darstellung.....	5
3.3	Übertragungsglieder	6
3.3.1	P-Glied.....	6
3.3.2	I-Glied.....	7
3.3.3	D-Glied.....	7
3.3.4	T ₁ -Glied.....	7
3.3.5	T ₂ -Glied.....	8
3.3.6	T _T -Glied.....	8
3.3.7	Verbindungen.....	8
3.3.8	Sprungantworten.....	8
3.4	Reglertypen	10
3.4.1	Analoge Regler.....	10
3.4.2	Digitalregler (Zeitdiskrete Regelung).....	10
4	Berechnung	11
4.1	Modellbildung	11
4.1.1	White-Box-Modellierung.....	11
4.1.2	Black-Box- Modellierung.....	11
4.1.3	Grey-Box-Modellierung.....	12
4.1.4	Laplace-Transformation.....	12
4.2	Computergestützte Analyse	12
4.3	Fuzzy-Logic	13
5	Praxis-Modelle	14
5.1	Modell des VSAS in Biel	14
5.1.1	Eigenschaften des Modells.....	14
5.1.2	Unsere Versuche.....	16
5.2	Eigenes Projekt	20
5.2.1	Ziel und Konzept.....	20
5.2.2	Material.....	20
5.2.3	Mechanik.....	20
5.2.4	Elektronik.....	21
5.2.5	Software.....	29
5.2.6	Schlussfolgerungen.....	34
6	Schluss	35
6.1	Reflexion	35
6.2	Ausblick	35
7	Authentizitätserklärung	36
8	Verzeichnisse und Indices	37
8.1	Bücher-Quellen	37
8.2	Internet-Quellen	38
8.3	Liste verwendeter Software	39
8.4	Abbildungsverzeichnis	40

1 Abstract

Im digitalen Zeitalter der Technik und der Globalisierung, in dem wir uns heute befinden, verblüfft uns die Welt immer wieder mit neuen technischen Raffinessen, die uns das Leben oft enorm vereinfachen. Es werden uns diverse Entscheidungen abgenommen, Prozesse automatisch in Gang gebracht und Abläufe geregelt, ohne dass sich ein Mensch darum kümmern muss. Wenigstens scheint es oft so, dass die Technik in gewisser Weise immer intelligenter wird. Wir wissen alle, dass es nicht so ist. Denn sämtliche so genannt intelligenten Systeme arbeiten einfach die Programme ab, mit denen sie gefüttert wurden. Die vorliegende Arbeit untersucht, wie ein solches Programm, das die Drehzahl eines Elektromotors konstant hält, zu Stande kommt. Das Faszinierende daran ist die schier beliebige grosse Komplexität, die mit dem Erreichen eines solch einfach klingenden Ziels verbunden sein kann.

2 Einleitung

2.1 Übersicht

Die Begriffe *Regelung* und *Steuerung* werden oft durcheinander gebracht. Nach DIN¹ 19226 ist die Regelung wie folgt definiert: „**Das Regeln, die Regelung, ist ein Vorgang, bei dem fortlaufend eine Größe, die Regelgröße (zu regelnde Größe), erfasst, mit einer anderen Größe, der Führungsgröße, verglichen und im Sinne einer Angleichung an die Führungsgröße beeinflusst wird.**“ Dieser Satz könnte, zugegebenermaßen, kaum trockener formuliert sein. Doch innerhalb der Norm erfüllt er damit seine Aufgabe: er ist allgemein gehalten. Was sagt er aber aus und wodurch unterscheidet sich nun die Regelung von der Steuerung?

Die *Regelungstechnik* hat im heutigen Alltag bereits in unzählige Bereiche Einzug gehalten und dieser Trend wird sich noch lange fortsetzen. Angefangen von der Klospülung, über die Heizung, zur Robotertechnik, hält sie heute auch immer mehr Einzug im Transport („Autopilot“) und komplexen vernetzten Systemen (Energienetz, Strassenetz, Internet). Man könnte natürlich so weit gehen, auch den Schulunterricht mit Feedback als regelungstechnisches Phänomen anzusehen. Wir bleiben aber bei der Technik. Im nächsten Kapitel finden sich dazu einige konkrete Beispiele, bei denen Regelungstechnik im Einsatz ist.

Anhand eines Versuchsmodells, welches in Kapitel 5 vorgestellt wird, realisieren wir eine digitale Drehzahlregelung mit einem universell einsetzbaren Mikroprozessor. Das Modell besteht aus einem alten Videorecordermotor als *Regelstrecke* und einem Inkrementdrehgeber² als *Messeinrichtung*. Die *Regeleinrichtung* umfasst den Mikroprozessor und den Schrittmotortreiber (die elektronische Schaltung zur Ansteuerung des Motors).

Die Begriffe, Abkürzungen und Symbole des Fachgebiets Regelungstechnik werden in der Literatur nicht ganz einheitlich verwendet. Wir halten uns in der vorliegenden Arbeit an die Konventionen aus unserem Elektrotechnik Fachbuch [FET02].

2.2 Motivation

Die gesamte Regelungstechnik ist ein sehr grosses Gebiet, auf welchem durch fortlaufende Forschungen nach wie vor Verbesserungen erzielt werden. Es wird an technischen Gewerbeschulen angeschnitten und ist an Fachhochschulen in einschlägigen Studiengängen eine grosse Herausforderung für die Studenten. Trotzdem ist, insbesondere im Bereich der digitalen Systeme, noch lange nicht alles vollständig erforscht und noch weniger dokumentiert. Uns faszinieren die immer schneller und präziser werdenden Bewegungsabläufe von Bearbeitungsmaschinen, wie z.B. CNC-Fräsmaschinen oder Fertigungsrobotern. Auch die Möglichkeiten, welche sich aus der Kopplung von Bildverarbeitender Software mit mechanischen Systemen ergeben, so zum Beispiel die automatische Abstandsregelung oder das automatische manövrieren von Fahrzeugen, sind faszinierend. Mit leistungsfähigen Mikrocomputern, intelligenter Software und sorgfältiger Anbindung der Mechanik wird es, unserer Ansicht nach, beispielsweise bald möglich sein, sein Auto per Knopfdruck automatisch rückwärts einparken zu lassen [SGPAR]. Selbstverständlich können wir in der zur Verfügung stehenden Zeit und mit unseren bescheidenen Grundlagenkenntnissen nicht alle interessanten Gebiete im Detail untersuchen. Aus diesem Grund konzentrieren wir uns auf die Drehzahlregelung.

¹ Deutsche Industrie Norm

² optisches System zur Erfassung des Drehwinkels (bzw. der Drehzahl)

3 Regelungstechnik

3.1 Beispiele

3.1.1 Festwertregelung

Das einfachste Beispiel einer Regelung mit festem Sollwert ist die **Toilettenspülung**. Dort soll immer der gleiche Füllstand herrschen. Wird die Spülung betätigt, leert sich der Spülkasten, womit auch der Schwimmer sinkt und gleichzeitig das Wasserventil öffnet. Ist die Toilette gespült, steigt der Wasserpegel im Spülkasten kontinuierlich an, bis das Wasser den Schwimmer wieder hochhebt. Dadurch wird das Ventil langsam geschlossen, bis kein Wasser mehr fließt. Egal, wie viel Wasser pro Spülung aus dem Spülkasten geleert wird, füllt sich dieser immer wieder auf das gleiche Niveau.

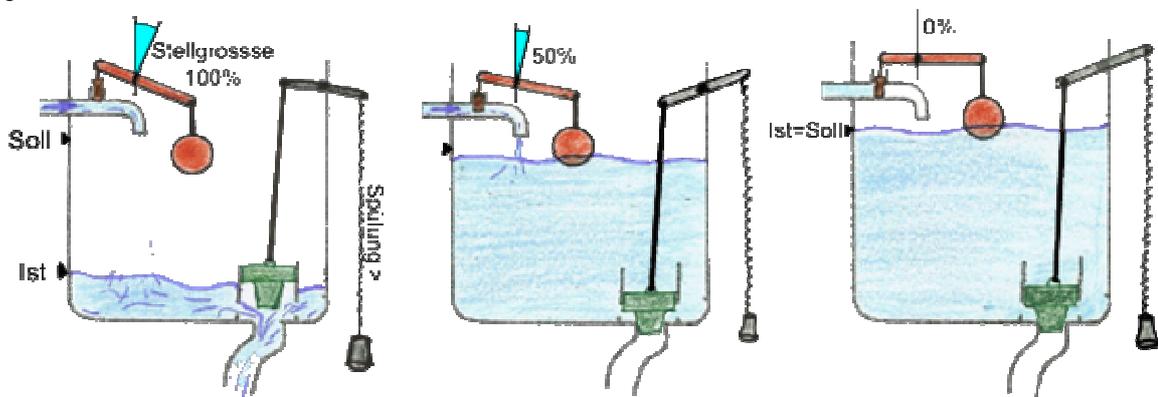


Abb. 3.1: Toilettenspülung

Das **Mahlwerk einer grossen Kornmühle** muss für jede Sorte Korn eine bestimmte Drehzahl haben. Die Belastung, durch mehr oder weniger Körner in der Mühle, ist aber sehr schwankend. Nun ist es Aufgabe der Regelung, die Betriebsspannung am Antriebsmotor entsprechend zu variieren, damit die Drehzahl bei Belastungsschwankungen konstant bleibt. Die Regelgrösse ist in diesem Falle die Drehzahl der Mühlesteine.

Die **Temperatur** in einem Wohnzimmer soll das ganze Jahr über konstant gehalten werden. Im Sommer ist die Heizung ausgeschaltet, da die Temperatur auch so den Sollwert erreicht. Sobald aber der Winter herein bricht, verliert der Innenraum durch die Wände und Fenster Wärmeenergie → die Temperatur sinkt. Folglich schaltet sich die Heizung ein, und die durch einen Gasbalg geregelten Ventile³ an den Heizkörpern öffnen sich, damit diese mit heissem Wasser durchströmt werden. Wird der Raum wärmer, schliessen sich die Ventile wieder, und lassen weniger heisses Wasser durch den Heizkörper fließen. Nach dem Lüften der Wohnung, ist die Innentemperatur gesunken, das Gasvolumen im Ventil wird durch die Abkühlung kleiner und öffnet das Ventil wieder. Somit fließt wieder mehr durch den Heizkörper.

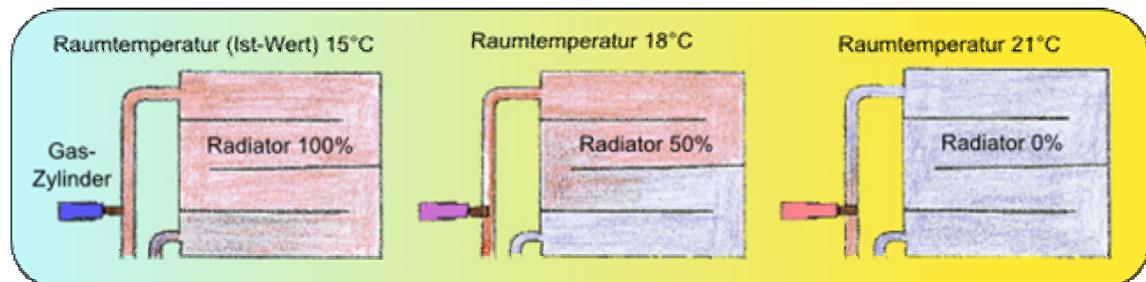


Abb. 3.2: Temperaturregelung

³ System „Danfos“

Eine Festwertregelung ist auch das Konstanthalten der **Geschwindigkeit** eines Autos im Autobahn-Verkehr. Auf gerader Strecke muss das Gaspedal, bei einem guten Auto, nur leicht betätigt werden, um eine Geschwindigkeit von 100km/h zu halten. Geht es plötzlich bergauf, so muss der Regler (Mensch oder Tempomat) mehr Gas geben. Führt der Strassenverlauf wieder bergab, so muss der Regler das Gas zurücknehmen, oder allenfalls bremsen, damit er die angestrebte Geschwindigkeit nicht überschreitet.

3.1.2 Nachlaufregelung

Befindet sich ein **Auto** im städtischen Verkehr, so ändert sich die Geschwindigkeit, als Sollwert, ständig. Dann spricht man von einer Folge- oder Nachlaufregelung. Die *Regelaufgabe* besteht darin, die Geschwindigkeit des Autos trotz Änderung des Strassenbelages oder des Gefälles möglichst schnell den Verhältnissen anzupassen: Rote Ampel = 0Km/h, Stau = 4Km/h, enge Passage = 20 km/h, Generell = 50km/h und so weiter.

Bei **Werkzeugmaschinen** benötigt die Positionierung des Fräskopfes oder des Drehstahls eine Regelung mit höchsten Ansprüchen. Das Ziel ist, während und vor allem zwischen Bearbeitungseingriffen in das Werkstück, hohe Geschwindigkeiten⁴ zu fahren, und gleichzeitig Positioniergenauigkeiten von wenigen μm ⁵ zu erreichen. Den Sollwert gibt das Programm vor, worauf die Positionierantriebe so angesteuert werden, dass sie ihn auch erreichen.

Ein weiteres Beispiel einer Nachlaufregelung, ist zum Beispiel die Regelung der Drehzahl grösserer **Handbohrmaschinen**. Die Drehzahl soll proportional zum Druck auf den „Pistolen-Knopf“, oder der Potentiometer-Einstellung sein. Bohrt man mit mittlerer Drehzahl ein grosses Loch in eine Metallschiene, so würde die Drehzahl ohne Regelung stark abgebremst. Die Regelung korrigiert dies so schnell wie möglich aus, damit am Werkzeug eine konstante Schneidgeschwindigkeit herrscht. Um die Drehzahl nach Bedarf noch etwas zu erhöhen, gibt man einfach mehr Gas, worauf die Drehzahl (bei einer guten Regelung) unabhängig von der Belastung proportional aufsteigt. Ist das Material durchbohrt, schwindet der Reibungswiderstand am Werkzeug sprunghaft. Da würde nun die Drehzahl einer nicht geregelten Bohrmaschine stark ansteigen, auch dies verhindert eine flinke Regelung fast unbemerkt.

3.2 Regelkreis

Ein Regelkreis ist ein *geschlossener Wirkungskreis*. Ein Istwert (die *Regelgrösse*) wird darin laufend mit einem Sollwert (der *Führungsgrösse*) verglichen und, abhängig vom Vergleichsergebnis, beeinflusst. Man spricht von einer *Rückkoppelung*, oder neudeutsch: *Feedback*. Äussere Störeinflüsse werden so, je nach Reglerart mehr oder weniger wirkungsvoll, kompensiert. Der laufende Soll-/Istwert-Vergleich, die Rückkoppelung und der dadurch entstehende geschlossene Wirkungskreis (Regelkreis), sind die zentralen Merkmale der Regelung. Bei einer *Steuerung* wird der Istwert nicht „kontrolliert“, es gibt keine Rückkoppelung und es besteht lediglich ein einfacher, offener Wirkungsweg.

Die Führungsgrösse hat bei vielen Regelungen einen bestimmten vorgegebenen Wert, der nur gelegentlich geändert wird, es müssen also vorwiegend Störungen ausgeregelt werden. In diesem Fall spricht man von einer *Festwertregelung*.

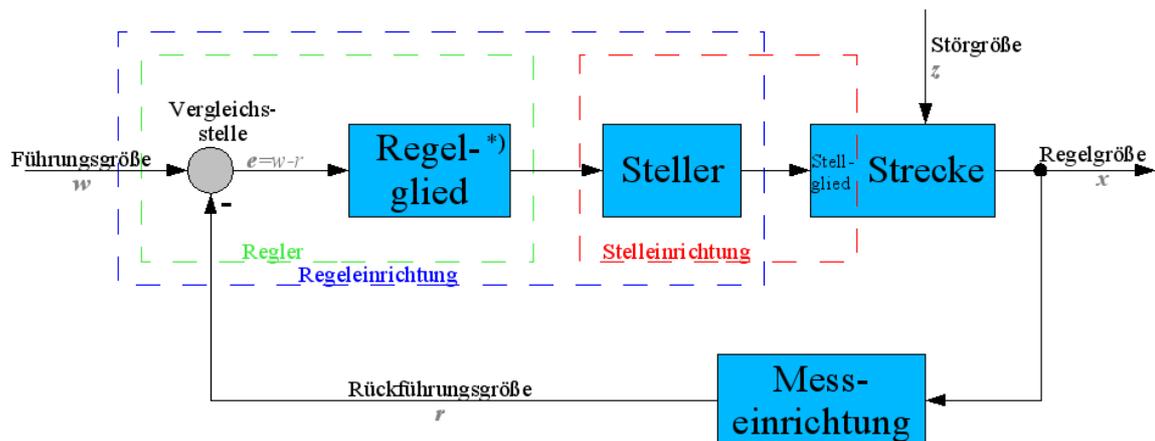
Wird eine Führungsgrösse dauernd verändert und ist es Aufgabe der Regelung, die Regelgrösse möglichst genau nachzuführen, so sprechen wir von einer *Folge- oder Nachlaufregelung*.

3.2.1 Darstellung

Zur Darstellungen von Regelkreisen werden die einzelnen Funktionselemente in Blockschaltbildern, wie Abb. 3.3, dargestellt. Die Darstellung eines Blocks gibt Aufschluss darüber, wie seine Ausgangsgrösse bezüglich der Eingangsgrösse reagiert. Die Darstellung wird in verschiedene Bereiche unterteilt (gestrichelt eingetragen), die sich teilweise überschneiden. Die Regeleinrichtung beinhaltet Vergleichler, Regelglied und Steller, der Regler nur Vergleichler und Regelglied, die Stelleinrichtung den Steller und das Stellglied. Die Informationsrichtung der Grössen, die in einem Blockschaltbild zirkulieren, wird mit Pfeilen angegeben. Diese Signalpfeile weisen vom verursachenden zum beeinflussten Element. Additionsglieder werden durch Kreise dargestellt, und summieren alle Eingangssignale mit Berücksichtigung des Vorzeichens. Bei Verzweigungen wird das Signal mehreren Elementen gleichzeitig zugeführt.

⁴ Heutiger Stand: bis zu 7m/s = 25km/h

⁵ Mikrometer (Millionstelmeter) $1\mu\text{m} = 0.001\text{mm}$



*) korrigiert das dynamische Verhalten

Abb. 3.3: Blockschaltbild eines Regelkreises [WPBSB]

Die Führungsgrösse (Sollwert) w ist z.B. die gewünschte Drehzahl. Um nun zu erreichen, dass sie trotz auftretenden Störungen konstant bleibt, muss die Regelgrösse x (tatsächliche Drehzahl / Istwert) laufend erfasst und mit der vorgegebenen Führungsgrösse verglichen werden. Der Vergleich von Ist- und Sollwert geschieht mittels Differenzbildung von Rückführ- und Führungsgrösse: $w - x = e$ (Stellgrösse).

Ein Regelkreis besteht im Wesentlichen aus zwei Teilbereichen: der Regelstrecke, die aus dem Stellglied und der Strecke besteht, und der Regeleinrichtung, die am Ausgang der Strecke misst, einen Vergleich mit dem Führungswert macht und nach den vorgegebenen Kriterien das Stellglied ansteuert.

Im Beispiel der Toilettenspülung aus der Einleitung, ist der Füllstand die zu regelnde Grösse x . Die Führungsgrösse w ist der, durch die mechanische Anordnung des Schwimmers vorgegebene, Höchst-Wasserstand. Die Rückführgrösse r ist der aktuelle Schwimmer- und damit Wasserstand. Der Schwimmer ist so konstruiert, dass das Ventil bei vollem Spülkasten geschlossen ist. Sobald der Spülkasten geleert wird, fällt der Schwimmer nach unten, womit er gleichzeitig das Ventil öffnet, welches hier das Stellglied ist. Die Stellgrösse ist jetzt maximal, da auch die Regeldifferenz $e (= w - r)$ maximal ist. Wird die Spülung nicht mehr betätigt, so steigt der Wasserstand, als Integralfunktion des Stellgliedes, wieder an. Ab einer bestimmten Höhe schwimmt der Schwimmer (die Messeinrichtung) auf dem Wasser auf und schliesst das Ventil mit zunehmendem Wasserspiegel, wodurch immer weniger Wasser fliesst und der Wasserspiegel auch immer langsamer steigt. Sobald der Wasserspiegel und der Schwimmer ihre Soll-Höhe erreicht haben, ist das Ventil ganz geschlossen, und es fliesst kein Wasser mehr ($x = w - r = 0$). Dargestellt in einem Diagramm ergibt der Istwertverlauf eine diagonale Linie, die oben mit einem Bogen in die Horizontale verläuft, ähnlich wie die Sprungantwort eines PT_1 -Gliedes (siehe Abb. 3.4).

3.3 Übertragungsglieder

Verschiedene Kombinationen der 6 Grund-Übertragungsglieder bestimmen das Verhalten jeder Regelstrecke und jedes Reglers. Sie sind als „Grundrechenoperationen“ der Regelungstechnik zu sehen.

3.3.1 P-Glied

Das *Proportionalglied* ist das einfachste Übertragungsglied, denn es ist lediglich ein Verstärker, wobei das Eingangssignal proportional zum Ausgangssignal verstärkt wird. Als Parameter des P-Glieds wird der *Proportionalbeiwert* K_p eingeführt, der den Verstärkungsfaktor zwischen dem Eingangswert x_e und dem Ausgangswert x_a angibt.

$$K_p = \frac{x_a}{x_e} \text{ respektive } x_a = x_e \cdot K_p$$

Betrachtet man zum Beispiel bei einem Transistor den Kollektorstrom I_C als Eingangsgrösse x_e und den Basisstrom I_B als Ausgangsgrösse x_a , folgt bei einer sprunghaften Änderung der

Eingangsgrösse die Ausgangsgrösse unverzögert und proportional der Eingangsgrösse. Der Verstärkungsfaktor K_p eines Transistors liegt durchschnittlich bei 100.

3.3.2 I-Glied

Das *Integralglied* ist da schon einiges komplexer. Bei diesem Glied ist die Geschwindigkeit der Änderung der Ausgangsgrösse Δx_a proportional zur Eingangsgrösse x_e . Umgekehrt heisst das: je grösser das Eingangssignal ist, desto schneller ändert sich die Ausgangsgrösse. Steht die Eingangsgrösse auf null, so bleibt die Ausgangsgrösse auf ihrem aktuellen Wert stehen.

Die Steigung des Ausgangssignals hängt nicht nur vom Festwert x_e am Eingang ab, sondern auch von der Integrationszeit T_I . Der Anfangswert der Integration ist x_{a0} .

$$x_a = x_{a0} + \Delta t \cdot \frac{x_e}{T_I}$$

Als Parameter für das Integralglied steht K_I und ersetzt dabei $\frac{1}{T_I}$ aus der Grund-Integralformel.

Damit heisst dann die neue Formel

$$x_a = x_{a0} + \Delta t \cdot K_I \cdot x_e$$

Mit Sinuskurven oder Anstiegsrampen als Eingangssignal wird dann die Berechnung ziemlich aufwändig. Ist der Eingangswert nicht konstant, so verändert sich die Steilheit des Integrals laufend. Ein Grafikrechner kann hier sehr hilfreich sein, der das Integral einer Funktion berechnen und den Graphen dazu zeichnen kann.

3.3.3 D-Glied

Das *Differentialglied* gibt an seinem Ausgang die Geschwindigkeit der Änderung der Eingangsgrösse wieder. Dies lässt sich mathematisch so formulieren:

$$x_a \approx \frac{\Delta x_e}{\Delta t}$$

Als Parameter des D-Gliedes wird der Differentialbeiwert K_D eingeführt. Dieser bestimmt, wie gross die Änderung am Eingang pro Zeit sein soll, um ein bestimmtes Ausgangssignal zu erreichen.

$$x_a = K_D \cdot \frac{\Delta x_e}{\Delta t}$$

Als Sprungantwort eines idealen D-Gliedes erhält man einen Nadelimpuls, der theoretisch unendlich gross ist, da die Änderung des Eingangssignals in (theoretisch) null Zeit erfolgt.

$$\text{„} \frac{x_e}{0} = \infty \text{“}$$

In der Praxis ist der Wert begrenzt durch die maximale Betriebsspannung der Regelungselektronik. Zur Untersuchung von D-Gliedern ist die Anregung mit einer Sprungfunktion nicht sehr sinnvoll, da sich der Nadelimpuls schlecht auswerten lässt. Nimmt man zur Analyse eine Anstiegsantwort auf, so wird der Zusammenhang mit der Formel deutlicher. Der Differentialbeiwert K_D lässt sich dann durch die Steilheit des Eingangssignals und durch die Höhe des Ausgangssignals bestimmen.

3.3.4 T_1 -Glied

Verzögerungsglieder erster Ordnung besitzen einen Energiespeicher. Ein anschauliches Beispiel aus der Elektronik wäre ein RC-Glied (Schaltung mit einem Widerstand und einem Kondensator). Oft wirken Schwungmassen, die bei allen rotierenden Teilen vorhanden sind, verzögernd. Die Sprungantwort kann nur in Kombination mit einem Proportionalglied aufgezeichnet werden. Sie entspricht dann der bekannten exponentiellen Aufladefunktion eines Kondensators. Die Formel dazu lautet:

$$x_a = K_p \cdot x_e \cdot \left(1 - e^{-\frac{t}{T_1}} \right)$$

Die Zeitkonstante T_1 entspricht der Zeit, nach der die Ausgangsgrösse 63% ihres Endwertes erreicht hat.

3.3.5 T_2 -Glieder

Die Funktion eines Verzögerungsgliedes zweiter Ordnung ergibt sich, wenn man zwei T_1 -Glieder hintereinander schaltet. Die Sprungantwort in Kombination mit einem P-Glied hat einen S-förmigen Verlauf, anfangs parabelförmig und am Ende wieder exponentiell.

$$x_a = K_p \cdot x_e \cdot \left(1 - \frac{T_1}{T_1 - T_2} e^{-\frac{t}{T_1}} + \frac{T_2}{T_1 - T_2} e^{-\frac{t}{T_2}} \right)$$

T_n -Glieder

Ein T_n -Glieder ergibt sich, wenn n T_1 -Glieder hintereinander geschaltet sind. Dies ist in der Praxis sehr häufig der Fall. Die Sensoren, welche die Regelgrösse messen, und die Messumformer, mit denen die Signale zu der Vergleichsstelle zurückgeführt werden, bringen immer Verzögerungen mit sich. Der Grad der Ordnung des Betrachteten T_n -Gliedes lässt sich oft gar nicht mehr explizit bestimmen, was aber nicht weiter schlimm ist, da sich die Charakteristik nur unwesentlich von der des T_2 -Gliedes unterscheidet und mit zunehmender Ordnung der eines T_T -Gliedes nähert.

3.3.6 T_T -Glieder

Das *Totzeit-Glieder* ist als Übertragung in einer Regelstrecke das problematischste Glied. Bei solchen Systemen ist es so, dass eine Änderung an der Stellgrösse keine sofortige Auswirkung auf die messbare Ausgangsgrösse hat. Bis nach Ablauf der sogenannten Totzeit ändert sich der Ausgangswert nicht, da die Regeleinheit in dieser Zeit auch keine Änderung am Istwert erfährt. So ist x_a stets um T_T zu x_e phasenverschoben.

3.3.7 Verbindungen

Die verschiedenen Grundtypen von Übertragungsgliedern sind in der Realität fast immer miteinander kombiniert. Oft ist die Strecke durch ein P-Glied dominiert. Man spricht dann von einer *Strecke mit Ausgleich*. Es gibt aber auch reine Integralstrecken, wie Tanks oder Aufzüge. Verzögerungsglieder sind immer mit anderen Grundgliedern kombiniert. Ein XT_0 -Glieder weist keine Verzögerung auf. T_0 wird lediglich der Vollständigkeit halber angefügt, damit das zeitliche Verhalten definiert ist.

3.3.8 Sprungantworten

Das dynamische Verhalten eines Übertragungsgliedes lässt sich dokumentieren indem man seine Sprungantwortfunktion aufzeichnet. Dies ist der Zeitabhängige Verlauf der Ausgangsgrösse x_a , wenn die Eingangsgrösse x_e zum Zeitpunkt $t=0$ von null auf eins springt. Man nennt diese Funktion *Einheitsprung* oder *Heavisidesche Sprungfunktion*:

$$y = f(x) = \begin{cases} 0 & \text{wenn } x \leq 0 \\ 1 & \text{wenn } x > 0 \end{cases}$$

Die folgende Tabelle gibt eine Übersicht über die Form der Graphen. Sie wurden mit der im nächsten Kapitel vorgestellten Software Scilab erstellt.

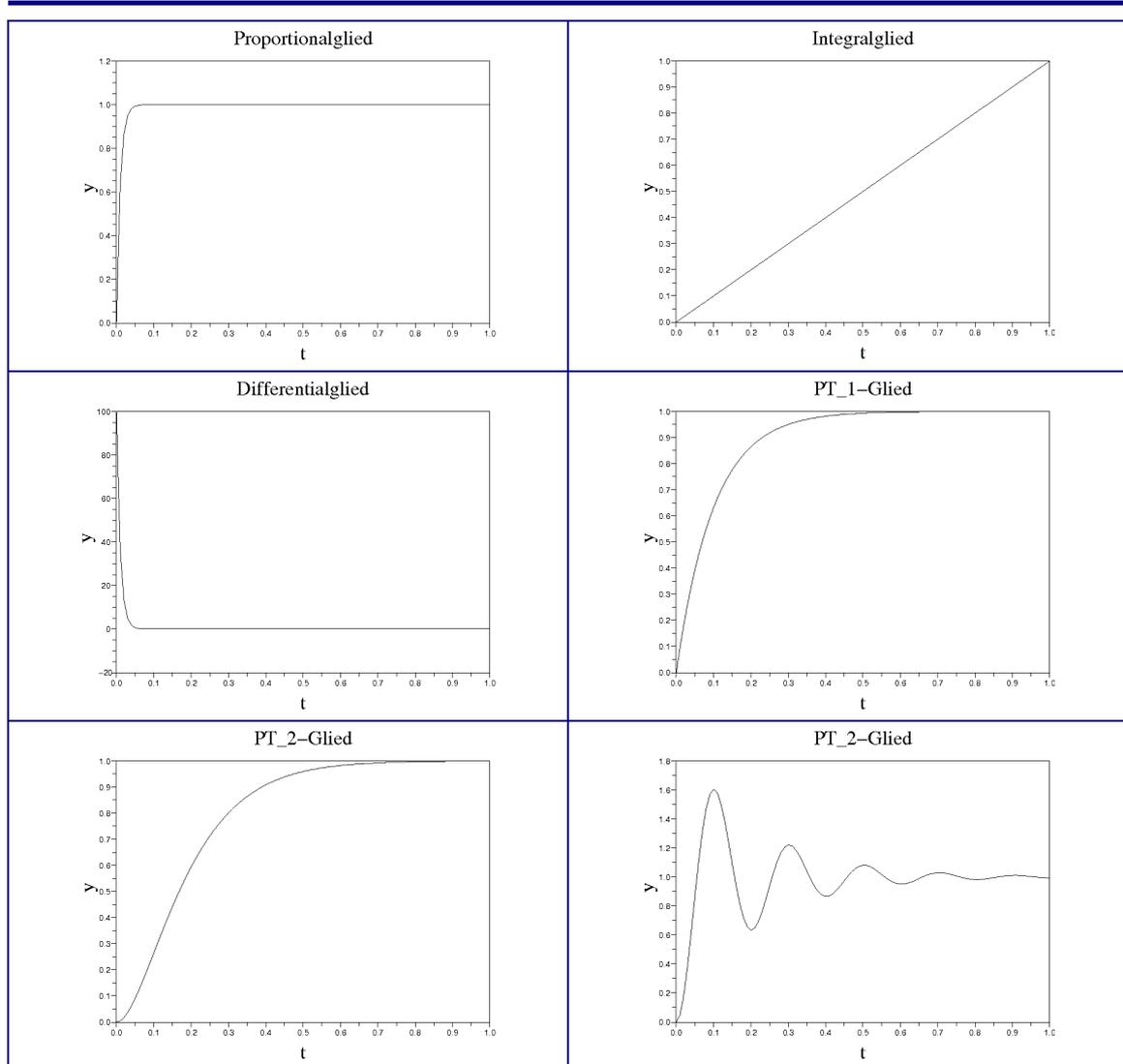


Abb. 3.4: Mit Scilab generierte Sprungantwortdiagramme

3.4 Reglertypen

3.4.1 Analoge Regler

Strom- und Spannungswerte werden oft elektronisch analog verarbeitet. Im Mittelpunkt steht der Operationsverstärker in seinen Grundschaltungen als invertierender oder nicht invertierender Verstärker, Subtrahierer, Integrierer, oder Differenzierer.

Daneben unterscheidet man zwischen stetigen und unstetigen Reglern. Bei stetigen Regeleinrichtungen, die im Normalfall mit analogen Regeleinrichtungen realisierbar sind, kann der Stellbereich sämtliche Werte der Stellgröße annehmen. Der Stellbereich bei OP-Schaltungen ist von dessen Betriebsspannung begrenzt.

Unstetige Regler können nur einige bestimmte feste Werte annehmen. Bei der Regelgröße treten Schwankungen auf: wenn die erforderliche feste Stellgröße nicht vorhanden ist, muss sie abwechselnd aus der nächst kleineren und der nächst grösseren zusammengesetzt werden. Am extremsten sind diese Schwankungen bei einem Zweipunkt-Regler zu erkennen, der nur EIN oder AUS als Stellgrösse annehmen kann. Er muss bei einem Sollwert von 50% der maximalen Aussteuergröße (also 50% EIN) gleichmässig ein- und ausschalten. Die Schaltfrequenz hängt von der Trägheit der Regelgröße und der Differenz des Ein- und Ausschaltpunktes, der Hysterese (Abb. 3.5), ab.

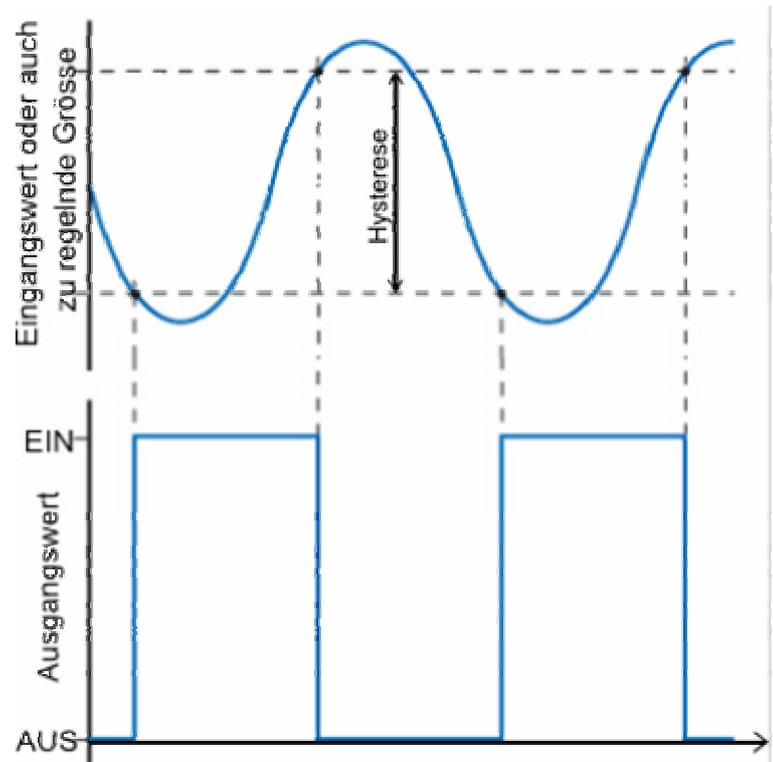


Abb. 3.5: Die Hysterese

3.4.2 Digitalregler (Zeitdiskrete Regelung)

Digitale Regler arbeiten, wie der Name schon verrät, mit digitalen Grössen. Soll eine analoge Grösse geregelt werden, ist eine Analog/Digital-Wandlung erforderlich, um das Eingangssignal als Zahl ins Programm einzulesen. Aus dem Eingangssignal berechnet ein Mikrocomputer anhand des gespeicherten Programms die Stellgröße, welche dann digital ausgegeben wird. Um dem analogen Stellglied eine brauchbare Stellgröße zu liefern, ist wiederum eine Digital/Analog-Wandlung erforderlich, die eine variable Spannung generiert.

Durch den immer einfacher und preiswerter werdenden Einsatz von Micro-Controllern gewinnt die digitale Regelung in der Praxis zunehmend an Bedeutung. Es ist dennoch unumgänglich, in der Theorie der analogen Regelungstechnik den Durchblick zu haben.

4 Berechnung

Analoge Reglersysteme lassen sich mit Hilfe von algebraischen Methoden sehr genau berechnen. Die dazu verwendete Mathematik ist von beachtlicher Komplexität und lässt sich nur durch intensive Auseinandersetzung mit den einzelnen Werkzeugen, Methoden *und* deren Grundlagen verstehen und anwenden. Digitale Systeme sind tendenziell noch komplizierter. Die grundlegenden Muster sind zwar dieselben wie bei den analogen Systemen und ein grosser Teil der Theorie lässt gut auf digitale Systeme übertragen, allerdings müssen noch einige weitere Faktoren beachtet werden. Durch die Nähe zum Computer und damit einer ganzen Palette von äusserst hoch entwickelten und hilfreichen Programmen, sind digitale Systeme im Endeffekt jedoch gleichwohl einfacher zu handhaben.

4.1 Modellbildung

Zur *Modellbildung* gibt es verschiedene Ansätze. Unterschieden wird im Fachjargon zwischen *White-* und *Black-Box-Modellierung*. Grundsätzlich sollte man sich immer zuerst einen Überblick über das zu berechnende System zu verschaffen. Anschliessend setzt man es in ein auf die Anwendung zugeschnittenes Modell um. Nicht alle Anwendungen erfordern den gleichen Grad an Präzision respektive Übereinstimmung des Modells mit der Realität.

Bevor man sich auf die Suche macht, sollte man sich darüber im Klaren sein, welcher Grundart das System angehört. Man unterscheidet zwischen *linearen* und *nichtlinearen* sowie *zeitinvarianten* oder *zeitvarianten* Systemen. Der einfachste Fall ist ein lineares, zeitinvariantes, kurz: *LZI-System*. Linear heisst, „dass zwischen Ein- und Ausgangsgrösse stets Proportionalität herrscht“⁶. Zeitinvariant ist ein System, wenn es sich unabhängig vom Zeitpunkt der Beobachtung immer gleich verhält. Wichtig ist diese Klassifizierung, um zu entscheiden, welche Art von Regler für das System überhaupt geeignet ist. Ist das System nicht linear, kann es unter Umständen *linearisiert* werden oder es müssen grundlegend andere, neue Wege beschritten werden – zum Beispiel mit Hilfe von Fuzzy-Logic (siehe Kapitel 4.4).

4.1.1 White-Box-Modellierung

Als *White-Box-Modellierung* bezeichnet man die *physikalische Modellbildung*. Das System wird genau untersucht und in Teilsysteme zerlegt. Solange, bis das Verhalten durch bekannte Naturgesetze (Energieerhaltungssatz, Ohmsches Gesetz, usw.) berechnet und exakt, mit Formeln, beschrieben werden kann. Häufig ist hier beispielsweise die Zerlegung eines Antriebssystems in drehmomentaufnehmende und drehmomentabgebende Systeme anzutreffen. Die Drehmomentabgabe eines Elektromotors ist abhängig vom Strom den er aufnimmt. Er ist somit ein Proportionalglied. Drehmomentaufnehmende Teilsysteme sind die Welle und andere rotierende Teile. Dies sind Integralglieder, da nach dem Impulserhaltungssatz von Newton⁷ alle Körper in ihrem Bewegungszustand verharren, solange keine äussere Kraft auf sie einwirkt. Die vom Motor zugeführte Energie wird in diesen Teilen also akkumuliert. Ausserdem ist die Reibung der Lager zu berücksichtigen, welche, sieht man einmal von der anfänglichen Haftreibung ab, konstant bleibt. Den grössten Teil des Drehmoments nimmt natürlich im Idealfall der angetriebene Teil selbst auf, der ganz unterschiedliche Eigenschaften haben kann, vor allem natürlich auch nicht vorhersehbare. Diese soll der Regler ja ausgleichen. Ein solches Modell kann extrem komplex und daher auch entsprechend aufwändig „herzustellen“ sein.

4.1.2 Black-Box- Modellierung

Den Problemen der White-Box-Modellierung versucht man entgegenzutreten durch die *Black-Box-Modellierung*, oder auch *Systemidentifikation*. Diese Methode eignet sich nur für LZI-Systeme. Indem man „ein mathematisches Systemmodell unter ausschließlicher Verwendung von Messwerten des Eingangs-/Ausgangsverhaltens“⁸ erstellt, und dabei die diesen Messwerten zugrunde liegenden physikalischen Vorgänge innerhalb des Systems vollständig ausser Acht lässt, erspart man sich enorm viel Aufwand. Die Schwierigkeit, eine zu den Messungen passende *Übertragungsfunktion* zu finden lässt sich heute mit Hilfe von leistungsfähigen Computern und

⁶ Zitat aus [WPLZI], Version von 14:35, 26. Okt. 2006

⁷ Sir Isaac Newton (1643-1727), Bedeutender englischer Physiker, Mathematiker, Astronom, Alchemist, Philosoph und Verwaltungsbeamter [WPNEW]

⁸ Zitat aus [WPRTK], Version von 13:38, 11. Nov. 2006

spezialisierte Software gut bewältigen. Was genau in der „Blackbox“ abläuft spielt letzten Endes oftmals ohnehin keine Rolle.

4.1.3 Grey-Box-Modellierung

Schliesslich lassen sich die beiden Methoden auch noch kombinieren. Dieser Ansatz wird *Grey-Box-Modellierung* genannt und ist wohl in den allermeisten Fällen der effizienteste. Man greift der Software dabei nämlich ein wenig unter die Arme, indem man aufgrund der Betrachtung eines rudimentären Modells vorhersagt, von welcher Art die *Übertragungsfunktion* des Systems sein wird.

4.1.4 Laplace-Transformation

Streng genommen ist die *Übertragungsfunktion* eines Systems immer die *Laplace-Transformation*⁹ des *Differentialgleichungssystems*, welches das System beschreibt. Ein Differentialgleichungssystem ist ein Mathematisches Gebilde, in dem Funktionsableitungen ($f'(x)$) und gewöhnliche Funktionen ($f(x)$) vorkommen. Damit zu rechnen ist äusserst kompliziert. Um die Handhabung dieser Gleichungen zu vereinfachen, werden sie mittels der Laplace-Transformation vom *Zeitbereich* in den *Frequenzbereich* übertragen. Es handelt sich um eine Integraltransformation und sie ist stark mit der Fourier-Transformation verwandt. Durch die Transformation fallen die Funktionen und abgeleiteten Funktionen aus den Gleichungen heraus und sie lassen sich algebraisch behandeln. Des Weiteren ist nach der Transformation die Eindeutigkeit der Funktion garantiert. Für ein System lassen sich nämlich beliebig viele verschiedene *Zustandsraumdarstellungen* (ZRD) finden und so ist es schwierig sie zu identifizieren und zu vergleichen. Alle ZRD führen aber, wenn man sie transformiert, zur gleichen Übertragungsfunktion.

4.2 Computergestützte Analyse

Das französische Forschungsinstitut INRIA (institut national de recherche en informatique et en automatique), welches den Ministerien für Industrie und Forschung angehört, forscht auf dem Gebiet der Informations- und Kommunikationswissenschaften und -technologien. Das Institut unterhält zu diesem Zweck auch verschiedene Softwarepakete, die es, grösstenteils, jedermann gratis zur Verfügung stellt. Scilab¹⁰ ist eine Software zur Durchführung von numerischen Berechnungen und mit dem bekannten MATLAB¹¹ vergleichbar. Die Software, welche für Linux und Windows Betriebssysteme zur Verfügung steht, liesse sich beispielsweise dazu verwenden, den digitalen Drehzahlregler, der im nächsten Kapitel vorgestellt wird, zu simulieren und zu optimieren.

Um mit Scilab vertraut zu werden und die Möglichkeiten auszuloten, entwickelten wir das unten aufgelistete Programm, das die Sprungantworten der meisten grundlegenden Übertragungsglieder der Regelungstechnik darstellt. Dazu wird jeweils mit dem Befehl `syslin()` ein entsprechendes System definiert und dann mit `csim()` das Verhalten bei einem Einheitssprung simuliert. Das Ergebnis (y) wird dann mit `plot2d()` in ein Diagramm übertragen, welches im zweiten Teil der selbst definierten Funktion `plotit()` noch angeschrieben und dann mit `xs2eps()` in einer Enhanced Postscript (.eps) Datei gespeichert wird. (Diese wurden noch zu Graphic Interchange Format (.gif) Dateien konvertiert, da dieses Format in Word besser dargestellt wird.)

```
n=0
s=%s
t=[0:0.01:1]
deff('u=input(t)', 'u=1')
//deff('u=input(t)', 'u=sin(t)')

clear plotit
function []=plotit(name, sys)
    xset("window",n)
    xbascc()
```

⁹ benannt nach Pierre-Simon Laplace (1749-1827), französischer Mathematiker und Astronom

¹⁰ © 1989-2006 INRIA ENPC; Scilab is a trademark of INRIA; www.scilab.org [SCIHP]

¹¹ © 1994-2006 The MathWorks, Inc.; MATLAB is a registered trademark of The MathWorks, Inc. [MATHP]

```

y=csim(input,t,sys)
plot2d(t,y)
//xgrid()
xtitle(name,"t","y")

a=get("current_axes")
xl=a.x_label
xl.font_size= 5
xl.font_style= 2
yl=a.y_label
yl.font_size= 5
yl.font_style= 2
t=a.title
t.font_size=8
t.font_style=2

xs2eps(n,name,1)

//disp(texprint(sys))

[n]=resume(n+1)
endfunction

plotit('Proportionalglied',syslin('c',1/(1+0.01*s)))
plotit('Integralglied',syslin('c',1/s))
plotit('Differentialglied',syslin('c',s/(1+0.01*s)))
//plotit('PT_t-Glied',syslin('c',1/(1+0.1*s)))
plotit('PT_1-Glied',syslin('c',1/(1+0.1*s)))
plotit('PT_2-Glied',syslin('c',1/((1+0.1*s)*(1+0.1*s))))
plotit('PT_2I-Glied',syslin('c',1/(1+0.001*s^2+0.01*s)))

```

Abb. 4.1: Programm für Scilab

Die von diesem Programm erzeugten Bilder (Proportionalglied, Integralglied, Differenzialglied, PT_1-Glied, PT_2-Glied und PT_2I-Glied) sind in Kapitel 3 bei den Beschreibungen der jeweiligen Übertragungsglieder abgedruckt.

4.3 Fuzzy-Logic

Hier wird nicht mehr im klassischen Sinne „gerechnet“, sondern eher „abgeschätzt“. „Fuzzy“ ist englisch und heisst „unscharf“. Die relativ neue Wissenschaft, erfunden 1965 von Lotfi A. Zadeh an der Kalifornischen Universität Berkeley, ist in jüngster Vergangenheit sehr erfolgreich unter anderem bei Haushaltsgeräten, aber auch in Fabriken eingesetzt worden. [GOFUZ] Besonders bei Systemen, die mit Menschen in Interaktion stehen kann diese Technologie ihre Stärken ausspielen.

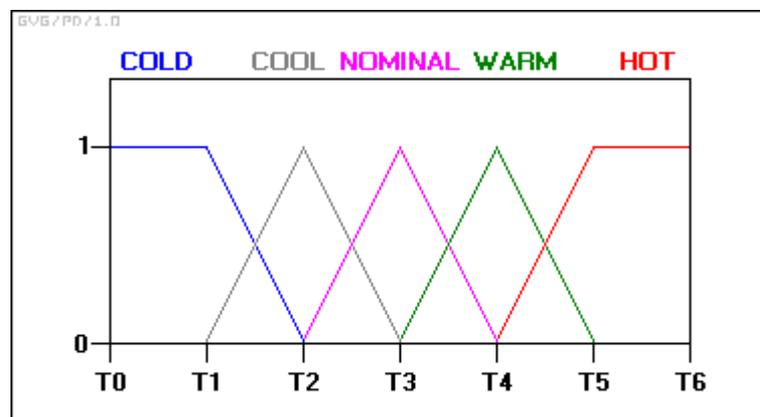


Abb. 4.2: Fuzzy-Logic

Die Abbildung veranschaulicht die Grundidee hinter Fuzzy-Logic, wie sie beispielsweise bei einer Heizungsregelung zur Anwendung kommen könnte. Die Temperatur, als reale Grösse, kann

ganz unterschiedliche Werte annehmen. Wir Menschen beschreiben sie mit Worten, die jeweils einen ganzen Temperaturbereich umfassen. „warm“, kann für Werte zwischen 20 und 30 Grad gelten. Aber 30 ist schon beinahe „heiss“. Für die einzelnen Bereiche bestimmt man nun einen Idealwert bei dem die jeweilige Bezeichnung voll zutrifft. Beispielsweiss ist es bei 25 Grad „warm“, bei 35 Grad „heiss“. Dazwischen, bei 30 Grad ist es „ein wenig heiss“, und „nicht mehr nur warm“. Die Fuzzy-Logik macht diese Ausdrücke für digitale Systeme zugänglich, indem sie die Bezeichnungen „kalt“, „kühl“, „normal“, „warm“, „heiss“ als Variablen betrachtet, die Werte zwischen null und eins annehmen können. 30 Grad ergäben dann 0.5 „warm“, 0.5 „heiss“. Eine Regelungsaufgabe kann so von Menschen sehr einfach beschrieben - „Wenn es kühl wird, die Heizung etwas aufdrehen“ - und von einer Maschine genau so ausgeführt werden – ein Programm wird dann die Leistung in Abhängigkeit der Variablen „kühl“ steuern. Der Vorteil besteht darin, dass die Regelung so stufenlos ist und die Heizung nicht bei 15 Grad abrupt anspringt, sondern schon vorher reagiert.

5 Praxis-Modelle

5.1 Modell des VSAS in Biel

Zu Beginn des vierten Lehrjahres besuchten wir einen ausserbetrieblichen Kurs des VSAS¹² in Biel, der sich vorwiegend der Messtechnik und den Grundlagen der Regelungstechnik widmete. Zur Veranschaulichung der verschiedenen Regelglieder stand dort ein Demonstrations-PID-Regelsystem zur Verfügung.

5.1.1 Eigenschaften des Modells...

An diesem Demonstrationssystem kann man die verschiedenen P-, I- und D- Anteile mittels Potentiometer einstellen, um so deren Wirkung beobachten zu können. Damit man den Regelverlauf gut erkennen kann, wurde dieses Regelsystem absichtlich sehr träge gemacht. Zwei grosse Panorama-Drehspulanzeigen zeigen Soll- und Istwert an und ein kleineres Drehspul-Instrument daneben die Aussteuerung des Motors, die Stellgrösse. Diese drei relevanten Grössen lassen sich auf der Rückseite des Gehäuses als analoge Spannung abgreifen. So kann der Verlauf mit einem Flachsreiber auf Endlospapier protokolliert werden.

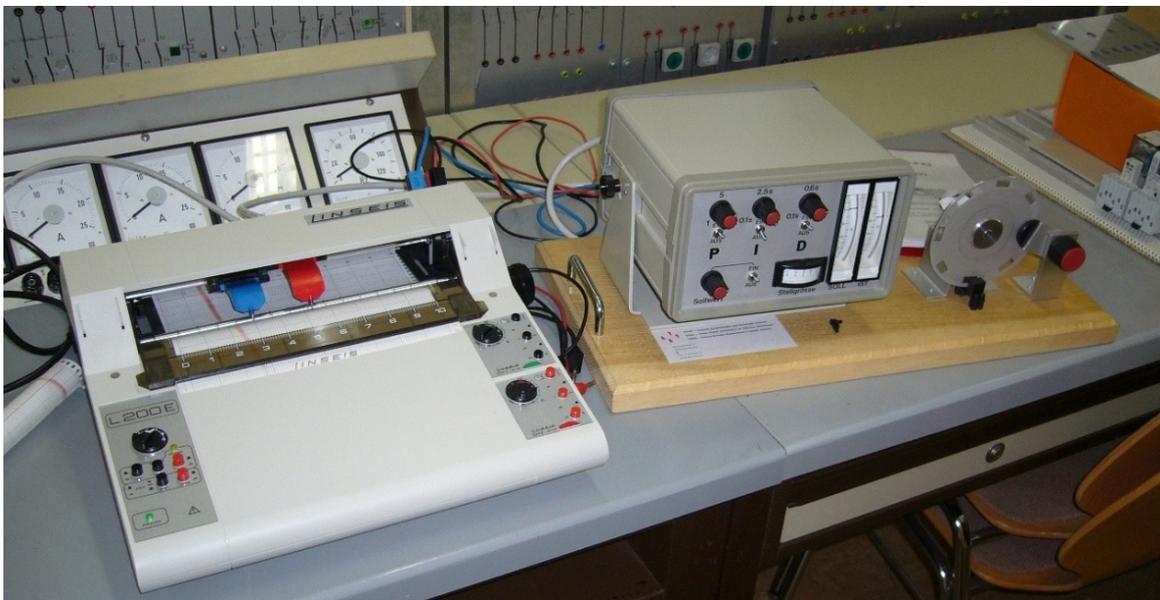


Abb. 5.1: Flachsreiber und Demonstrations- PID-Regelsystem

¹² Verband Schaltanlagen und Automatik Schweiz

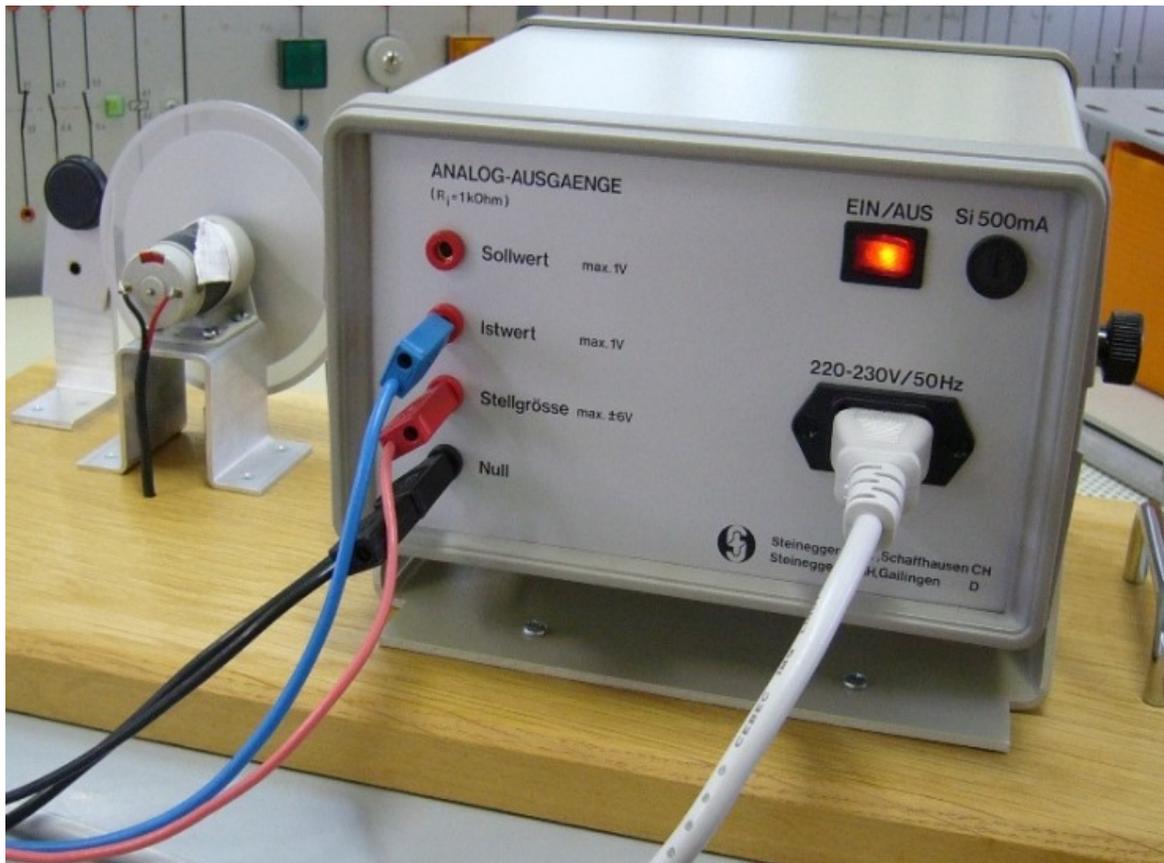


Abb. 5.2: Anschlüsse für die Analog-Ausgabe

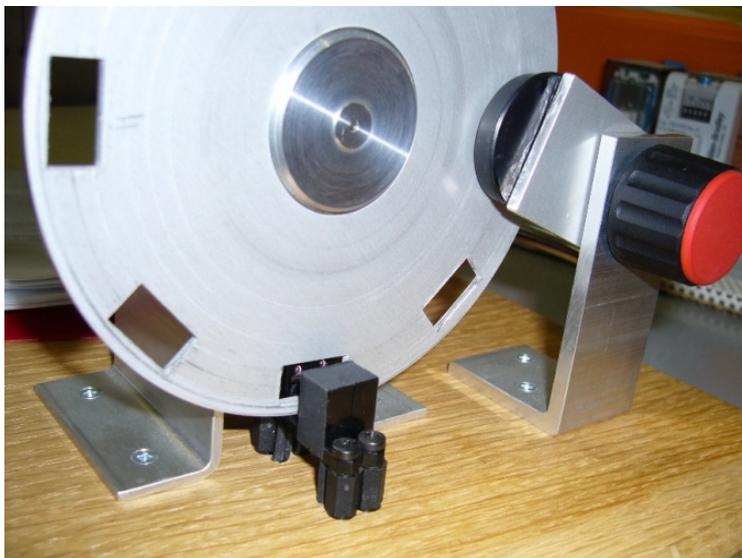


Abb. 5.3: Mehrzweck-Schwungscheibe

Als dynamische Last wirkt bei allen Beschleunigungs- und Verzögerungsvorgängen eine Aluminium-Schwungscheibe, die an der Motorwelle angebracht ist. Eine Wirbelstrombremse erzeugt, wenn zugeschaltet, eine statische Last. Die Drehzahl wird mit Hilfe einer zweikanaligen Lichtschranke erfasst, womit Angaben über Drehsinn (Impuls-Folge) und Geschwindigkeit (Impuls-Frequenz) der Scheibe gemacht werden können. Ein Umwandler formt diese Signale in eine zur Drehzahl proportionalen Spannung um, was dem Istwert x entspricht. Der Sollwert w kann mit einem

Potentiometer eingestellt werden, welches ebenfalls eine analoge Spannung ausgibt. Ein Vergleichs bildet und verstärkt die Differenz zwischen Soll- und Istwert ($w-x$) und gibt diese an das Regelglied weiter. (Abb. 5.4¹³) Der Proportional-, Integral- und Differentialregelanteil des Regelgliedes, der das Verhalten des Regelgliedes ausmacht, ist mit drei weiteren Potentiometern und Schaltern einstellbar. Diese Einstellungs-Aktoren sind auf dem Blockschaltbild nicht eingezeichnet, respektive sind im Kästchen „Regeleinrichtung“ enthalten.

¹³ Man beachte die Interpretationsunterschiede von Regelkreisen der verschiedensten Quellen

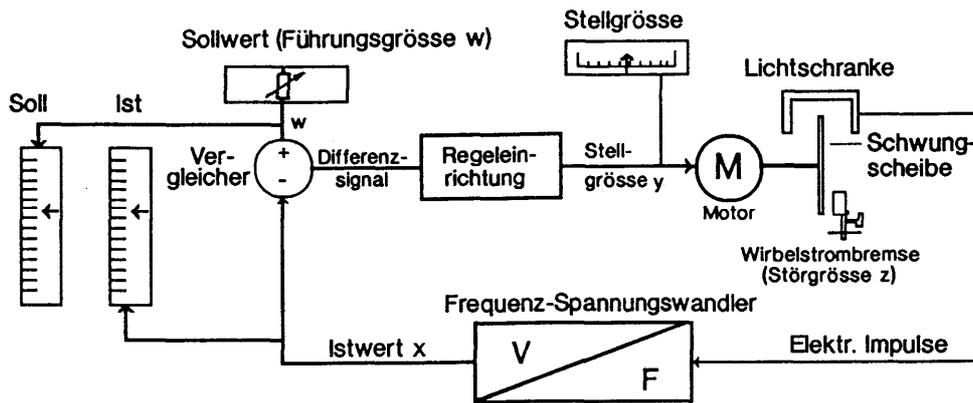


Abb. 5.4: Blockschema des Demo-Regelsystems



Abb. 5.5: Unser Arbeitsplatz in Biel

Auf obigem Bild ist der gesamte Versuchsaufbau mit Flachsreiber zu sehen. Die Y-t-Diagramme, die wir mit dem Flachsreiber aufgezeichnet haben, präsentieren wir im nächsten Abschnitt.

5.1.2 Unsere Versuche

Das Austüfteln einer guten Regelung ist nicht ganz einfach. Da ist es von Vorteil, zuerst das Verhalten der einzelnen Regelglieder zu analysieren. Wir zeichnen je eine gute Einstellung des P- und des I-Reglers auf.

Die Drehzahl (Ist-Wert) wird durch die blaue, und die Stellgrösse durch die rote Kurvenlinie gekennzeichnet. Die Stellgrösse wird immer durch die zweite rote, gerade Linie dargestellt. Um diese drei Grössen auf ein Papier zu zeichnen, haben wir es stets zweimal durch den zweikanaligen Schreiber laufen lassen. Da sich der Sollwert während den Regelungsabläufen nicht verändert, handelt es sich bei unseren Beobachtungsversuchen um Festwertregelungen.

Wir haben das System jedes Mal, nachdem sich die Drehzahl einigermaßen stabilisiert hatte, mit einer sprunghaften Belastung durch die Wirbelstrombremse beeinflusst. Dies machte sich bei allen Regelungsversuchen mit dem Abfallen der Drehzahl bemerkbar.

Proportional-Regler

40% P-Intensität
2s pro Häuschen

→ bleibende Regeldifferenz

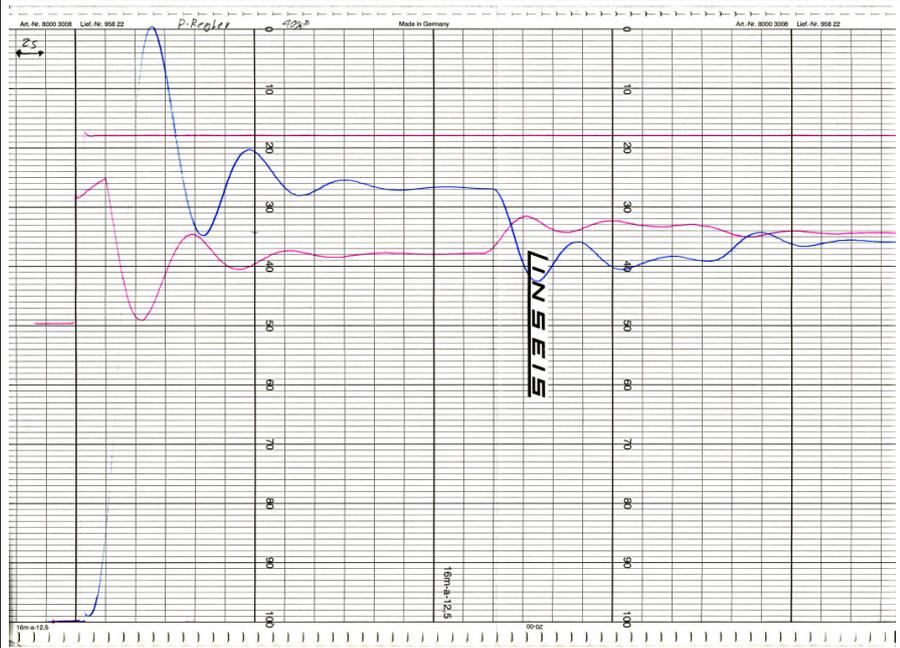


Abb. 5.6: P-Regler

Integral-Regler

100% I-Intensität; 10s pro Häuschen → keine bleibende Regeldifferenz

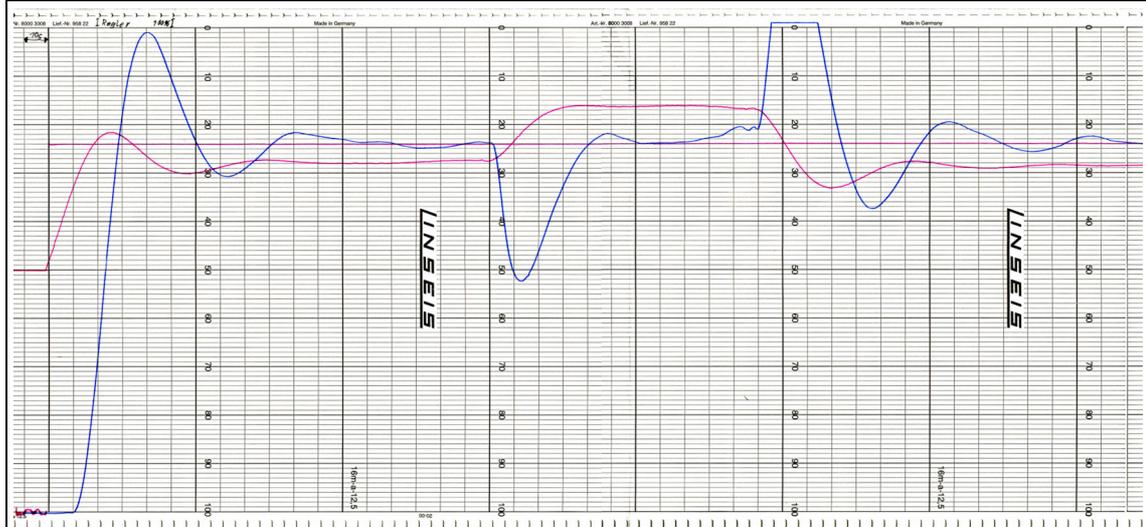


Abb. 5.7: I-Regler

Nachdem wir das Grundverhalten der beiden Regelglieder erkannt hatten, koppelten wir die Regelfunktionen in verschiedenen Kombinationen zusammen. Auch davon haben wir ein paar markante Beispiele auf Papier zeichnen lassen.

PI-Regler

20% P-Intensität
100% I-Intensität
2s pro Häuschen

→ flinke Regelung
ohne bleibende
Regeldifferenz

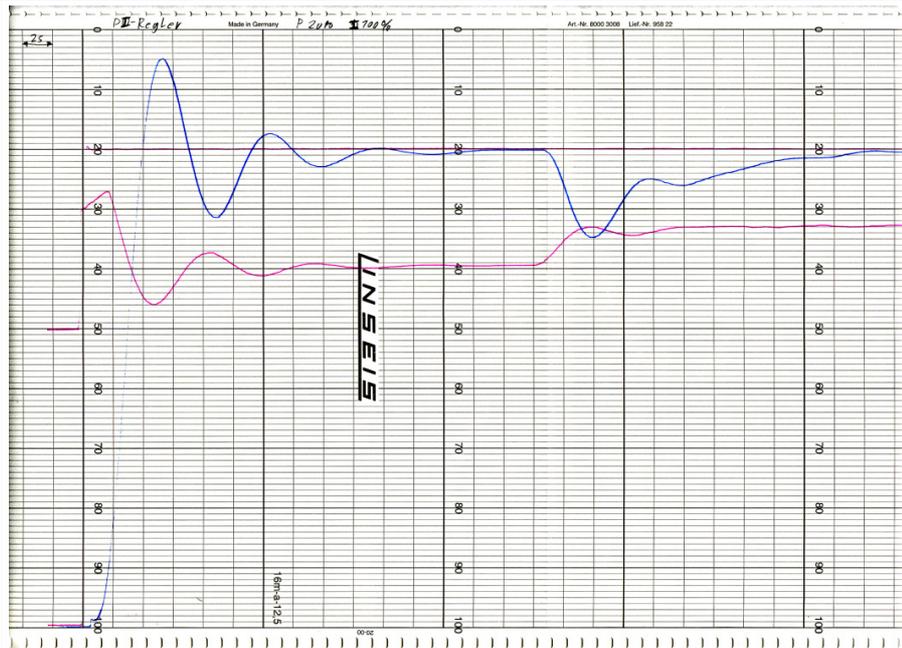


Abb. 5.8: PI-Regler

PD-Regler

60% P-Intensität
45% D-Intensität
2s pro Häuschen

→ flinke Regelung
mit bleibender
Regeldifferenz

(Der störende rote Strich
quer durch die Seite
markiert das kommende
Ende der Endlos-
Papierrolle, die leider
doch nicht ganz endlos
ist.)

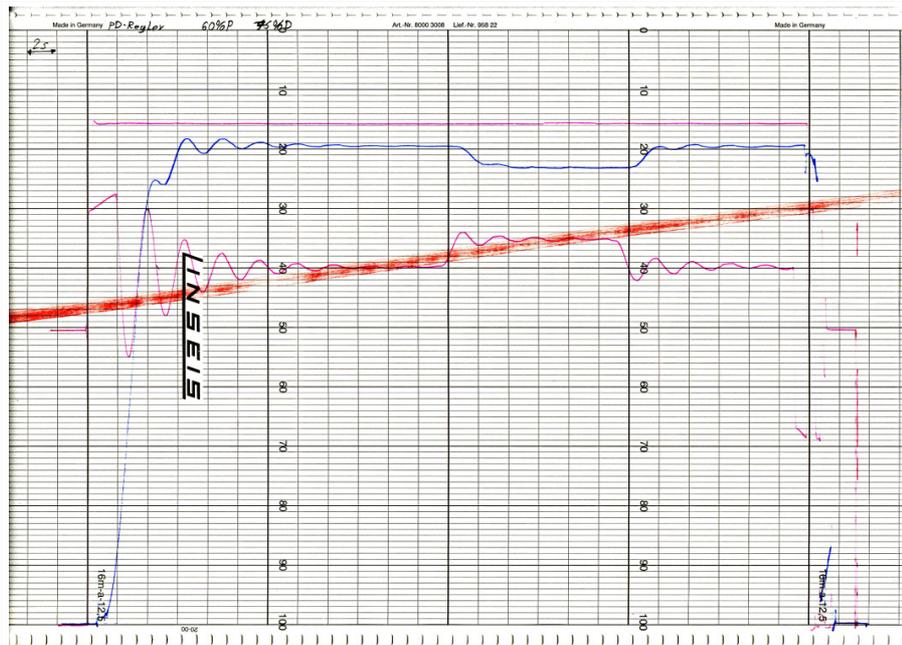


Abb. 5.9: PD-Regler

Durch das Verbinden zweier Regelglieder kann das Verhalten der Regelung stark verbessert werden, doch die richtigen Einstellungen sind maßgebend.

PID-Regler

20% P-Intensität; 100% I-Intensität; 100% D-Intensität; 2s pro Häuschen
 → Stabile Einstellungen, Sprungantwort überschwingt kaum

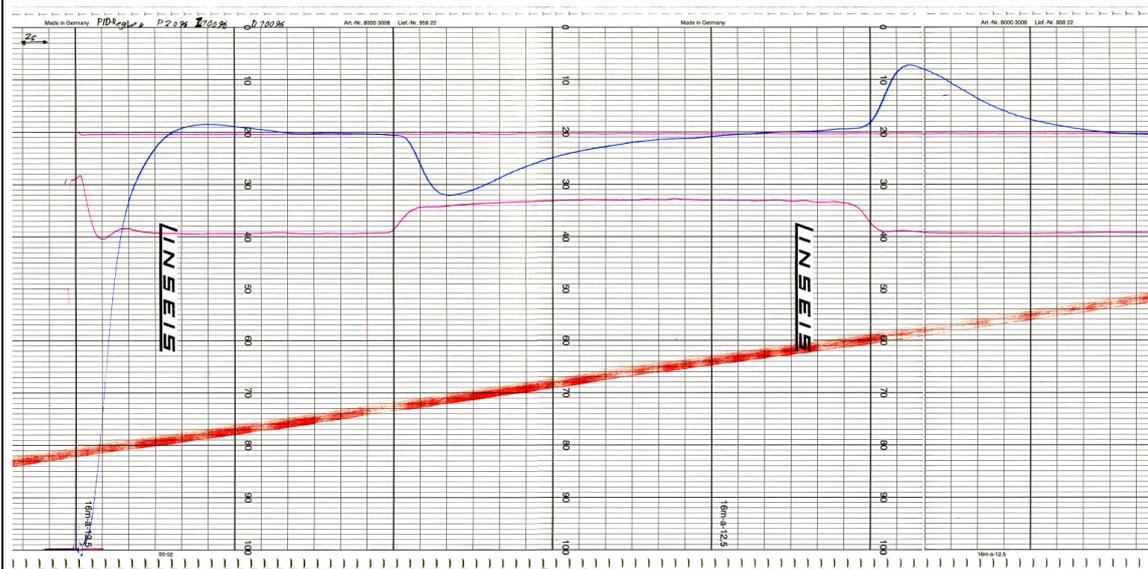


Abb. 5.10: PID-Regler 1

PID-Regler

25% P-Intensität
 25% I-Intensität
 50% D-Intensität
 2s pro Häuschen

→ Stabilität auf Störeinflüsse, Sprungantwort überschwingt etwas stärker

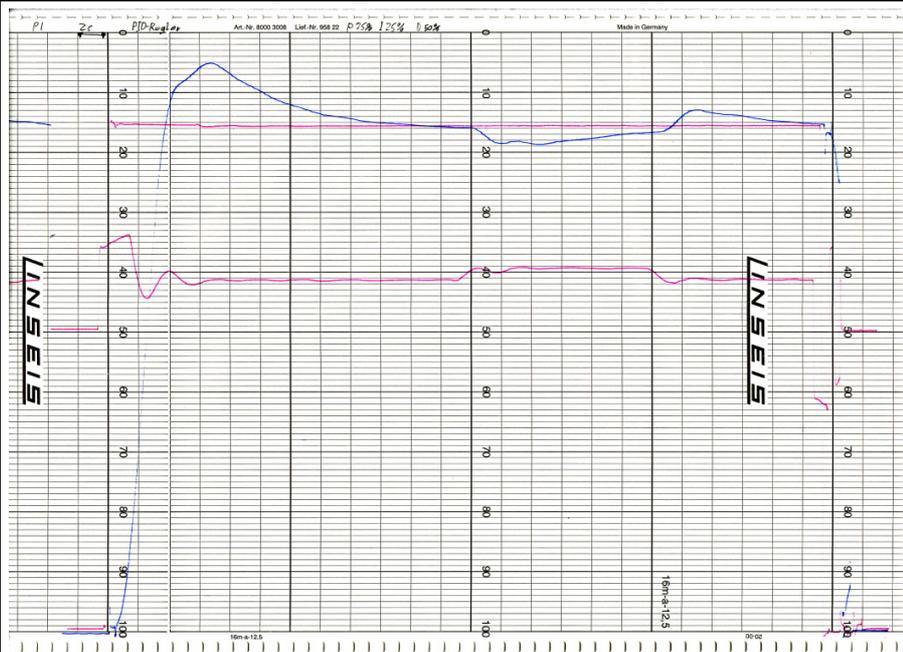


Abb. 5.11: PID-Regler 2

Mit diesem Demo-System ist es uns allerdings trotz erheblicher Anstrengungen nicht gelungen, eine perfekte Regelung hinzukriegen, was wohl systembedingte Gründe hat. In den Rückwärtszweig des Regelkreises wurde nämlich absichtlich eine grosse Verzögerung eingebaut, um die Auswirkungen der Verschiedenen Einstellgrößen auf das Verhalten deutlicher zu machen. Konkret wurde die Impulsglättungsschaltung, die die Signale des Drehgebers in eine Kontinuierliche Spannung umwandelt, künstlich träge gemacht. [STEIN]

5.2 Eigenes Projekt

Als Herausforderung haben wir einen eigenen Regelungs-Versuchsaufbau zusammengebaut, dessen Regellogik mit dem PC programmierbar ist.

5.2.1 Ziel und Konzept

Unser Ziel war, eine digitale Regeleinheit für eine Drehzahlregelung zu erstellen.

Als „Kopf“ der Regelung sollte der ATmega16 8-bit-Mikrocontroller vom Hersteller ATMEL dienen. Den Motor wollten wir mit einer Pulsweitenmodulation (PWM) über eine Gegentaktstufe ansteuern, damit man praktisch stufenlos beschleunigen und bremsen kann. Die Eingabe des Sollwertes sollte mittels Potentiometer möglich sein.

Zur Messung der Geschwindigkeit gibt es grundsätzlich zwei Möglichkeiten: Entweder mit einem Tachogenerator, dessen Spannung ungefähr proportional zur Geschwindigkeit ist (analoge Grösse) oder mit einer Lochscheibe und einer Lichtschranke, mit der direkt die Anzahl Umdrehungen pro Zeit gemessen werden kann (digitale Grösse). Wir entschieden uns für die digitale Variante, was sich zwar programmieretechnisch einiges aufwändiger darstellt, dafür jedoch genauer und störungsfreier ist. Gleichzeitig sollte noch ein zweiter Elektromotor angetrieben werden, der ein analoges Anzeigeeinstrument speist, oder als Brems-Generator benutzt werden kann.

5.2.2 Material

Als Antrieb verwendeten wir einen Elektromotor, der zuvor seinen Dienst in einem alten Videorecorder getan hatte. Dieser weist zwar eine eher niedrige Drehzahl, aber ein gutes Drehmoment auf, und ist, vor allem durch seine Grösse und den hohen Innenwiderstand, sehr belastbar. Nachdem uns die kleinen und feinen Rädchen und Schranken aus Computermäusen für unsere doch eher grobe Anwendung ungeeignet erschienen, fanden wir die passende Lichtschranke schliesslich in einem ausgemusterten Laserdrucker. Die Lochscheibe sowie die Wellen, Lagerungen und Halterungen entnahmen wir einem STOKYS¹⁴- Baukasten. Als Grundplatte benutzten wir ein massives, 22mm starkes Holzbrett, worauf wir alle mechanischen Teile befestigt haben. Den Mikrocontroller betreiben wir direkt auf dem Programmierboard, was ein schnelles Umprogrammieren und Kommunizieren ermöglicht. Die benötigten Bauteile für die weitere Elektronik, die um den Controller herum angeordnet wurden, fanden wir in C. Vögtlis Zimmer-Werkstatt oder haben sie im Elektronik-Versandhandel bestellt.

5.2.3 Mechanik

Anfänglich haben wir den Antriebsmotor frei beweglich aufgehängt, wodurch das Drehmoment, das der Motor abgab, direkt erkennbar wurde. Da dies die ganze Mechanik jedoch eher negativ beeinflusste, haben wir den Motor in der Folge starr befestigt.

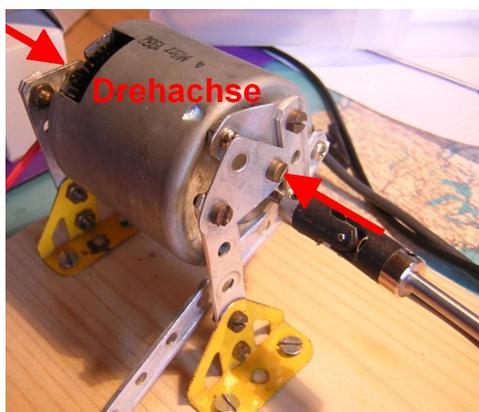


Abb. 5.12: Antrieb frei aufgehängt

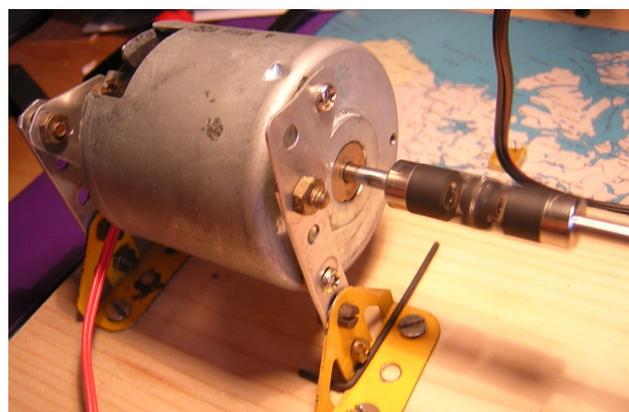


Abb. 5.13: Antrieb neu starr montiert

¹⁴ © STOKYS AG, Schweizer Metallbaukastensystem ähnlich wie Meccano.

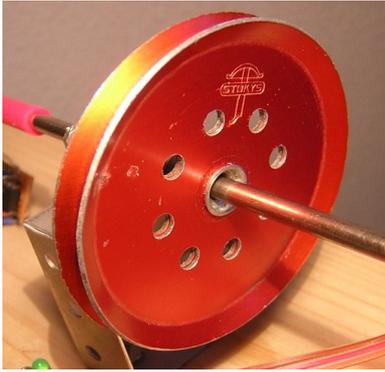


Abb. 5.14: Schwungrad

Die Lochscheibe aus dem STOKYS-Baukasten, drehten wir auf einer Drehbank bis knapp zu den Löchern ab, damit sie gut in die Lichtschranke passte. Anders als beim Demo-System verwenden wir hier nur eine einkanalige Lichtschranke, mit der sich der Drehsinn nicht berechnen lässt, sondern nur die Drehzahl (Impulsfrequenz), was aber vollkommen genügt. Als dynamische Lasten haben wir nebst all den Wellen ein Aluminiumrad und einen Elektromotor als „Generator“ an den Antriebsmotor gekoppelt. Sie wirken sich vor allem bei Beschleunigungs- und Verzögerungsvorgängen auf das Regelverhalten aus. Der Generator ist unbelastet auch eine dynamische Last, da seine rotierenden Teile auch träge sind. Sobald der Generator belastet wird, hat er eine bremsende Wirkung. Legt man eine Spannung an und führt Energie zu, wirkt er beschleunigend. So kann man dem System künstlich *Störgrößen* zuführen, welche dann ausgeregelt werden sollen.

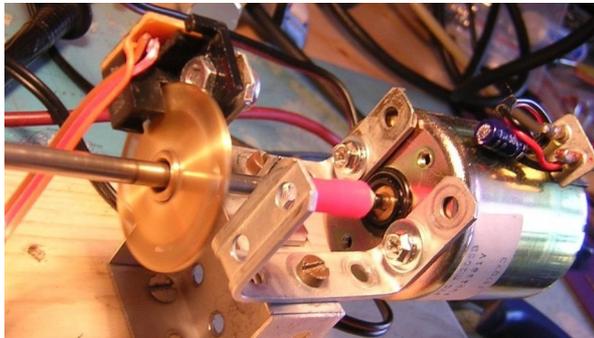


Abb. 5.16 Generator

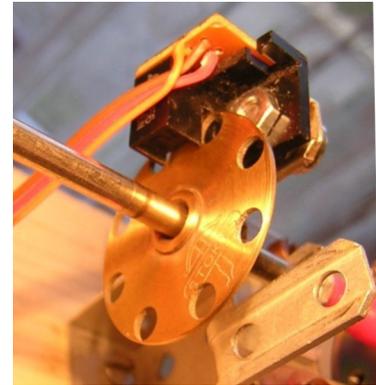


Abb. 5.15: Lochscheibe

5.2.4 Elektronik

Nachdem der mechanische Teil unsere Tests durch einfache Ansteuerung des Motors bestanden hatte, wendeten wir uns der Elektronik zu, die zwischen Mikrocontroller und Mechanik benötigt wird.

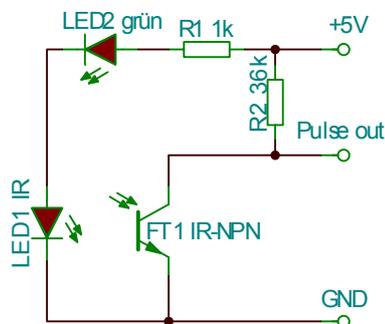


Abb. 5.17: LS-Schaltung

Geschwindigkeitserfassung

Die Grundlagen des Programmteils für die Geschwindigkeitsberechnung mit der optischen Geschwindigkeitsmessung hatten wir uns bereits erarbeitet. Nun bauten wir eine einfache elektronische Schaltung für die Pulsausgabe der Lichtschranke, welche sodann auch einwandfrei für hohe Frequenzen funktionierte.

Schaltungsbeschreibung¹⁵

Die Schaltung wird mit 5V versorgt. Dadurch leuchten LED1 (Lichtschrankensender) und LED2 (Betriebs-Anzeige). Wirft die LED1 das Infrarot-Licht durch ein Loch auf den Fototransistor FT1, so wird dieser leitend, und das elektrische Potential an „Pulse out“ sinkt zu GND (Ground, negativ) → low¹⁶. Im andern Fall wird

¹⁵ Alle abgedruckten Schaltpläne wurden mit TINA Pro (Student Version) erstellt [TIPRO]

¹⁶ low gebräuchliches Wort für logisch-0 in der Digitaltechnik.

FT1 nicht mehr belichtet, worauf dieser hochohmig wird, und das Potential an „Pulse out“ durch den nun niederohmigeren Widerstand R2 auf +5V gezogen wird → high¹⁷. Wir untersuchten das Ausgangssignal mit dem Kathodenstrahl-Oszilloskop (KO). Wie alle elektronischen Bauteile, hat auch dieser Fototransistor eine gewisse Trägheit, was frequenzabhängig zu folgenden Ausgangssignalen führt:

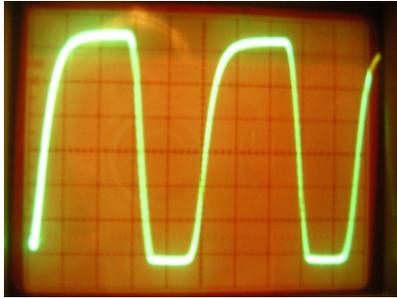


Abb. 5.18: out ca.30Hz

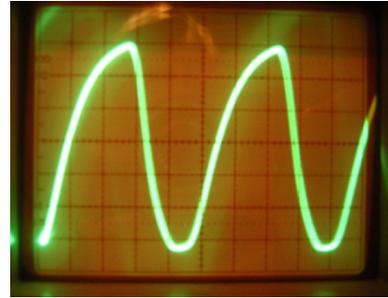


Abb. 5.19: out ca.170Hz

Idealerweise wäre das Signal exakt rechteckig, was aber durch Trägheit und fließendem Übergang von Belichtung und Schattierung des Fototransistors (der runden Löcher wegen) nicht möglich ist.

Motoransteuerung

Wie bereits angesprochen, wollten wir den Motor mit einem Pulsweitenmodulierten (PWM) Signal ansteuern. Dies ist eigentlich ein digitales Signal, welches für eine analoge Ausgangsgröße genutzt werden kann. Es kennt nur „1“ und „0“. Damit man aus diesen zwei Zuständen eine

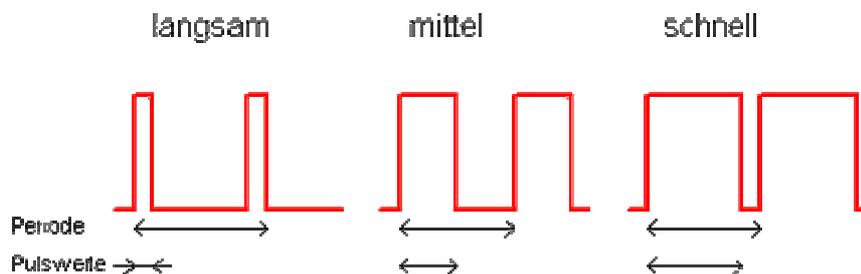


Abb. 5.20: Pulsweitenmodulation [RNPWM]

dazwischen liegende Größe simulieren kann, wird mit einer hohen konstanten Frequenz nach einem bestimmten Zeitmuster ein- und ausgeschaltet.

Ist während einer Periodendauer nur kurz eingeschaltet und lange ausgeschaltet (kurzer Puls), so ergibt sich eine kleine mittlere Leistung. Ist jedoch lange eingeschaltet und nur kurz ausgeschaltet (langer Puls), so ist die mittlere Leistung, welche in unserem Fall einen Motor antreibt, eher hoch.

Diese Art von Leistungssteuerung findet heutzutage immer häufiger Anwendung, da nicht nur die einfache Verarbeitung sehr vorteilhaft ist, sondern auch die geringen Verluste gegenüber analog (linear) angesteuerten Transistoren.

Um das PWM-Signal aus dem μC ¹⁸ am Motor zu testen, haben wir, mit einem entsprechenden Pin des Controllers, direkt einen Leistungs-MOSFET¹⁹ (mit Spannung ansteuerbarer Schalter) angeschlossen, der den Motor ansteuert.

¹⁷ **high** steht in der Digitaltechnik für logisch eins

¹⁸ Kürzel für Mikrocontroller

¹⁹ **M**etall-**O**xid-Halbleiter(**S**emiconductor)-**F**eld-**E**ffect-**T**ransistor

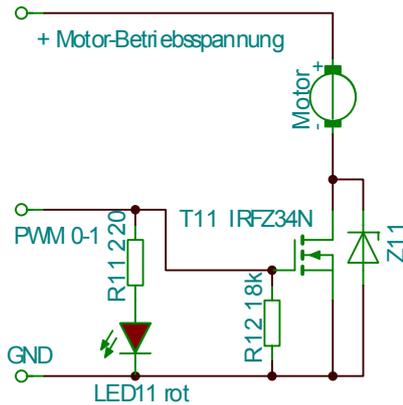


Abb. 5.21: Test-Treiber

(Spulen von Elektromotoren) entstehen, verursachten Einstreuungen in die Drehzahlmessung. Auf den folgenden KO-Diagrammen ist die Spannung am Motor visuell dargestellt. Die Aufnahmen sind so gedreht, dass die viertoberste Linie 0V entspricht (Motor Aus) und am oberen Rand die positive Betriebsspannung (Motor ein) angezeigt wird. In

Abb. 5.22 sehen wir den langen, negativen Impuls, der bei jedem PWM-Impulsende auftritt und die Störungen verursacht. Das Ergebnis ist nahe liegend: der Mikrocontroller zählt nebst den Impulsen der Lichtschranke auch die Störimpulse des PWM-Signals.

Die Nadelimpulse wurden zwar bereits durch die Z-Diode Z11 begrenzt, um T11 vor zu hohen und negativen Spannungen zu schützen, die ihn zerstören würden, doch dies reichte offensichtlich noch nicht aus, um den Rest der Schaltung störfrei zu halten.

Dieses Problem veranlasste uns zu einigen Versuchen, diese Nadelimpulse mit einem Kondensator zu dämpfen. Zuerst schlossen wir einen Kondensator mit einer Speicherkapazität von $4\mu\text{F}$ (4 Millionstel Farad²⁰) parallel zum Motor an. Dies machte sich dadurch bemerkbar, dass der Motor bei gleicher PWM-Impulslänge mehr Leistung hatte. Das hat damit zu tun, dass der Kondensator während eines Impulses genügend Energie speichert, dass er den Motor nach Ende des Impulses immer noch eine kurze Zeit antreiben kann. (Abb. 5.23) Der Nadelimpuls wurde zwar vollkommen vom Kondensator „verschluckt“, jedoch war die Motorleistung nicht mehr proportional zur PWM-Ansteuerung und somit auch nicht mehr linear.

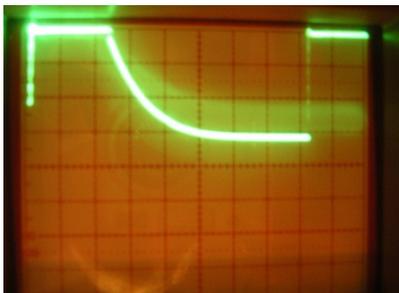
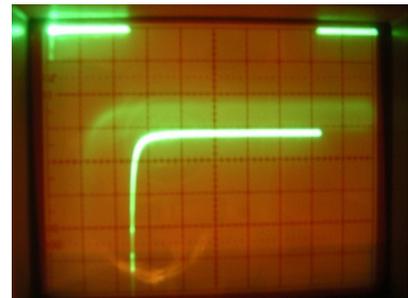
Abb. 5.23: Motor mit $4\mu\text{F}$ 

Abb. 5.22: Motor Kondensator

Abb. 5.24: Motor mit $1\mu\text{F}$

So griffen wir zum nächst kleineren $1\mu\text{F}$ -Kondensator, was schon ein ganz anderes Bild ergab (Abb. 5.24). Diese Kapazität macht sich auf eine Leistungsänderung kaum bemerkbar. Der Nadelimpuls ist nun nicht mehr „unendlich“ hoch. Er wird abgebremst, und die Spannung pendelt sich stabil auf die Leerlaufspannung des Motors ein (0V, wenn blockiert). Somit wurde zumindest

²⁰ Farad, Einheit von elektrischen Kapazitäten

die gesamte Elektronik vor den hohen Spannungsimpulsen geschont, aber leider war das Problem der Fehlzählung noch nicht gelöst.

Als möglichen Grund vermuteten wir, dass der Ausgang der einfachen Lichtschrankenschaltung durch seine Hochohmigkeit so leicht beeinflussbar ist, dass auf dieser Leitung die geringsten Störeinflüsse Probleme machen. Das wollten wir mit einem Schmitttrigger²¹ als Impedanzwandler, der direkt nach der Lichtschranke angeschlossen wird, ändern. Eine solche Schaltung würde den Ausgang der Lichtschranke einerseits etwas starrer machen und zu einem exakten Rechtecksignal führen. Und andererseits würden die extrem kurzen Anstiegs- und Abfallzeiten der Impulsflanken eines Rechtecksignals auch Störeinflüssen, die sich in der undefinierten Zone zwischen low und high auswirken, besser standhalten. Wir haben die Schaltung mit einem Operationsverstärker²² (OP1) und einigen weiteren notwendigen Kleinteilen ergänzt, welche die Funktion eines Schmitttriggers ergeben.

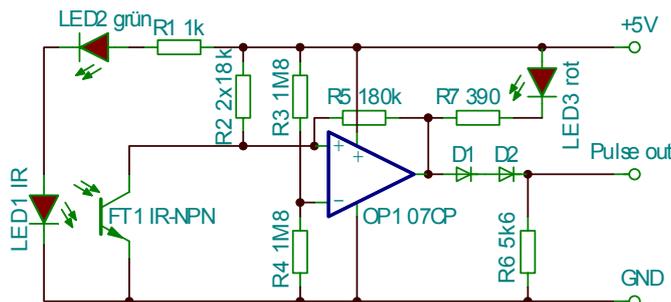


Abb. 5.25: Lichtschranken-Schmitttrigger

Schaltungsbeschreibung:

Die Schaltung wird immer noch mit 5V versorgt. Das Ausgangssignal der ursprünglichen Schaltung wurde nun an den „nicht invertierenden Eingang“ des OP's angeschlossen. Der Spannungsteiler aus R3 und R4 bestimmt die mittlere Kippspannung des OP-Ausgangs, während die Hysterese durch R5 bestimmt wird. Die LED3

leuchtet, wenn der Ausgang des OP's low ist, was dann der Fall ist, wenn der Fototransistor FT1 belichtet wird.

Wir haben diese Schaltung auf einem Elektronik-Steckbrett aufgebaut, um ihre Effektivität zu testen. Das obere Signal in Abb. 5.26 zeigt den Eingang, das untere den Ausgang des Schmitttriggers bei einer Frequenz von ca.170Hz. Diese Schaltung funktionierte einwandfrei, aber leider hatten wir das Problem damit noch nicht überlistet und behoben. Ist das Ausgangssignal der Schaltung high (FT1-Schatten), so ist deren Spannung stabil, ist es aber low (FT1-belichtet), so werden trotz des modifizierten OP-Ausgangs, immer noch Störimpulse gezählt.

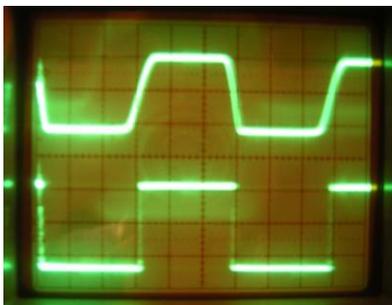


Abb. 5.26: Schmitttrigger

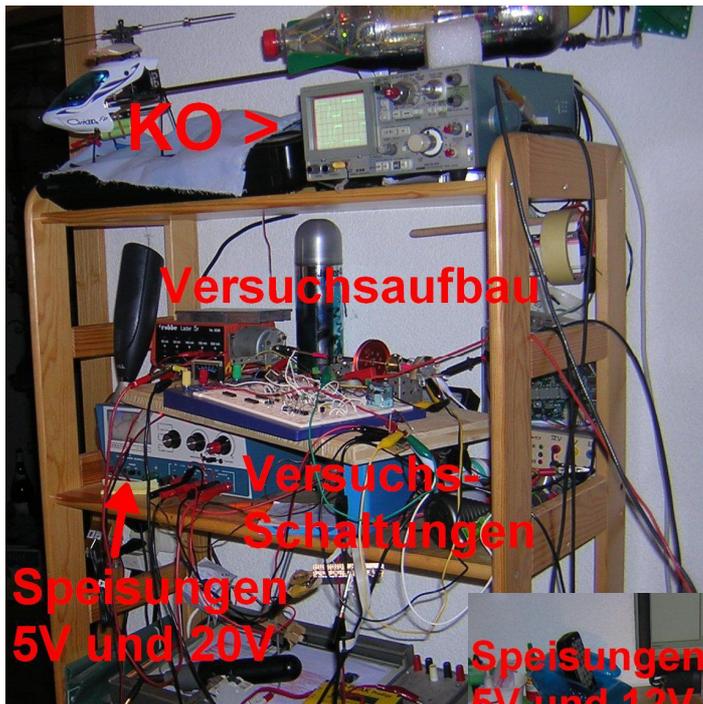
D1 und D2 bewirken am Ausgang „Pulse out“, bedingt durch den „Pull-down-Widerstand“ R6, eine um 1,4V tiefere Spannung, da der direkte Ausgang eines normalen OP's nicht bis an die untere Speissspannung herunter kommt. Damit wollten wir verhindern, dass die Störimpulse, die als positive Spannung in das optische Messsystem einstreuten, ausreichen würden, um vom μC als steigende Flanke erkannt zu werden. Zu unserer Enttäuschung hat auch das nichts

gebracht.

Als wir das gesamte Versuchsprojekt einmal aus der Vogelperspektive betrachteten und nur ein grosses Chaos erblicken konnten, entschieden wir uns, das Ganze nochmals neu und etwas zentraler aufzubauen. Die langen Verbindungsleitungen zwischen dem Programmierboard, das sich beim PC befand, und dem Versuchsaufbau im Regal, unter dem Oszilloskop, waren einfach zu Störanfällig.

²¹ Elektronischer Schwellwertshalter

²² Elektronischer Verstärker mit annähernd unendlichem Verstärkungsfaktor



So sah es zuvor aus: ziemlich einladend für jegliche Störungen. Es ist wohl üblich, dass solche Chaos-Versuche nicht auf Anhieb funktionieren können. Ist einmal ein Fehler vorhanden ist es einfacher, die Anordnung neu und nach Möglichkeit übersichtlicher aufzubauen, als lange nach der Ursache zu suchen.

Abb. 5.29: Chaos im Regal

Viel ordentlicher sieht es auf diesem Bild noch nicht aus, doch das ist es. Der Aufbau wird nun von einer zentralen Speisung versorgt und ist auf einer kleineren Arbeitsfläche stationiert.

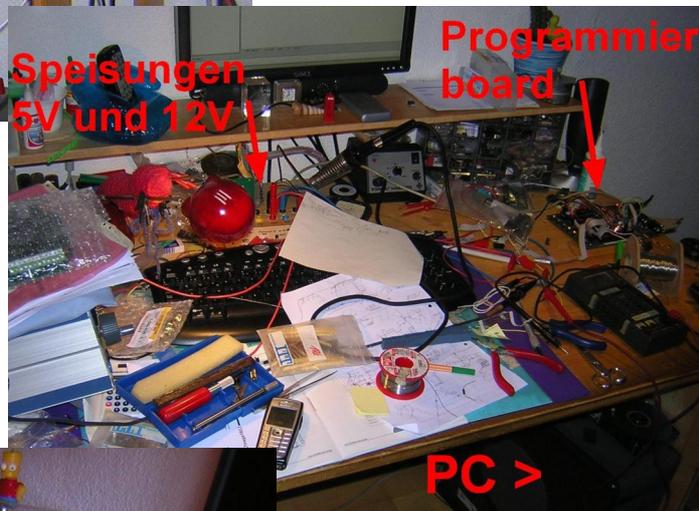


Abb. 5.27: Chaos auf dem Arbeitstisch



Den erhofften Erfolg brachte schliesslich diese Versuchsanordnung. Die Stromkreise beeinflussten sich nun nicht mehr gegenseitig, worauf wir auch die getestete Schmitttrigger-Schaltung nach einer groben Planung definitiv auf eine Hartpapier-Universal-Platine, auch „Veroboard“ genannt, löteten

Abb. 5.28: Neuaufbau

Auf dem Veroboard ist die Schaltung kompakter als zuvor beim Testaufbau auf dem Steckbrett.

Mit der beruhigenden Gewissheit, dass die PWM-Ansteuerung nun keine Störungen mehr verursacht, machen wir uns nun daran, die definitive Motor-Treiberschaltung zu entwerfen. Diese sollte den Motor stufenlos beschleunigen und bremsen können.

Eine einfache Schaltung, wäre die direkte Ansteuerung von zwei Transistoren mit je einem PWM-Ausgang des μ Controllers. Doch das würde einerseits zwei der drei Timer²³ brauchen, die wir eventuell noch für die

Geschwindigkeitsmessung benötigen und andererseits wäre zu befürchten, dass durch einen Softwarefehler beide Transistoren gleichzeitig angesteuert würden, was zu einem Kurzschluss der Betriebsspannung und damit zur Zerstörung der Transistoren führen würde. Deshalb haben wir uns für eine kompliziertere jedoch sicherere Variante entschieden. Diese kommt mit nur einem PWM-Ausgang aus, der mit einem weiteren Pin des Controllers auf einen der beiden Transistoren (Bremsen oder Beschleunigen) zugewiesen werden kann.

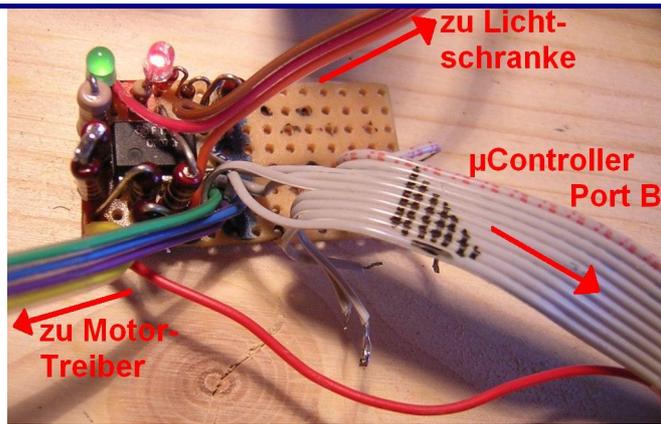


Abb. 5.30: Schmitttriggeraufbau

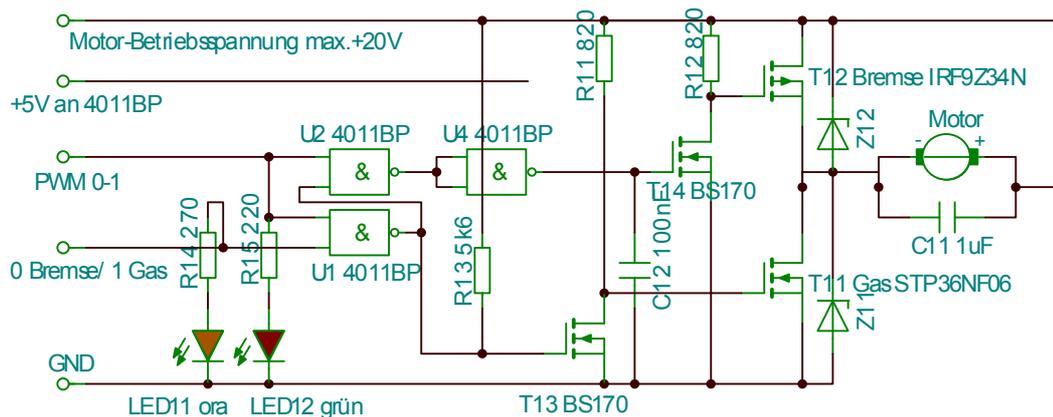


Abb. 5.31: Motortreiber

Schaltungsbeschreibung:

Im rechten Teil der Schaltung ist der Elektromotor, der mit einem Anschluss fix mit der positiven Betriebsspannung verbunden ist. Wird nun der MOSFET T11 angesteuert, so wird das Potential des anderen Motorenanschlusses nach GND gezogen, worauf über dem Motor die volle Betriebsspannung anliegt \rightarrow Beschleunigung. Wird T12 angesteuert, so wird der Motorenanschluss mit der positiven Betriebsspannung verbunden. Im zweiten Fall sind beide Anschlüsse des Motors auf dem gleichen Potential, was über dem Motor einen Kurzschluss bewirkt \rightarrow Bremswirkung. Die MOSFET's T13 und T14 ergeben mit den „pull-up-Widerständen“ R11 und R12 kleine Vorstufen, damit T11 und T12 mit der höheren Betriebsspannung angesteuert werden können. So sind sie in der Lage, grössere Ströme zu schalten.

Im Schaltplan ist auch gut ersichtlich, dass das Ansteuern beider MOSFET's (T11 und T12) gleichzeitig zu einem Kurzschluss der Betriebsspannung führen würde. Das wird mit dem vorderen Schaltplanteil mit den „NAND-Gattern“ verhindert, welche durch die Verbindungslogik das PWM-Signal abhängig vom Eingang „0 Bremse / 1 Gas“ an den entsprechenden MOSFET leiten. Der kleine Kondensator C12 verhindert ein gefährliches kleines „Problemchen“, das wir im

²³ Hardware-Teil in einem μ C für programmgelöste, zeitrelevante Anwendungen wie PWM-Ausgaben, Zeitmessungen oder Zeitintervalle für Programmaufrufe

nächsten Abschnitt noch schildern. Die LED's 11 und 12 visualisieren wiederum den Zustand der Eingangssignale.

Während den Testphasen der Schaltung gab es trotz des theoretisch einwandfreien Schaltplans noch ein weiteres Problem zu lösen: durch eine genaue Strommessung der Schaltung ohne Motor stellten wir im PWM-Gasbetrieb einen kleinen Stromfluss fest, welcher den MOSFET T12 stark erwärmte. Mit einer KO-Strommessung diagnostizierten wir kleine Stromimpulse mit der Taktfrequenz des PWM-Signals. Ein solcher Stromfluss kann nur entstehen, wenn kurzzeitig beide MOSFET's (T11 und T12) zeitgleich angesteuert werden. Da im Gasbetrieb der T11 regulär angesteuert werden soll, musste es also T12 sein, der irrtümlich „gezündet“ wurde. Eine Oszilloskopierung des Steueranschlusses (*engl.*: Gate) von T12 zeigte es offenkundig, der MOSFET wurde bei jedem Pulsbeginn des PWM für drei Mikrosekunden angesprochen. Das tönt nach wenig, ergibt aber im Moment des Impulses eine beträchtliche Peak-Leistung, was zu dieser Erwärmung führt. Im oberen Graphen (Kurvenlinie) von Abb. 5.32 ist dieser 3 μ s-Puls zu sehen. In der unteren Linie zeigen wir einen kurzen Abschnitt des PWM-Eingangssignals der Schaltung auf. Dies wirkt in dieser Zeitauflösung etwas zerknittert, was - wie wir später feststellten - auf eine negative Rückwirkung der Schaltung zurückzuführen ist.

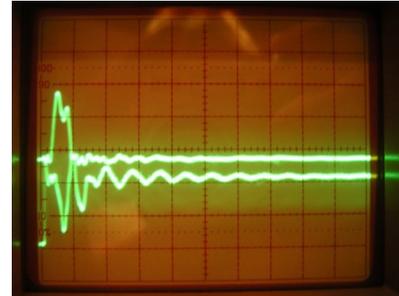


Abb. 5.32: Ansprechstörung

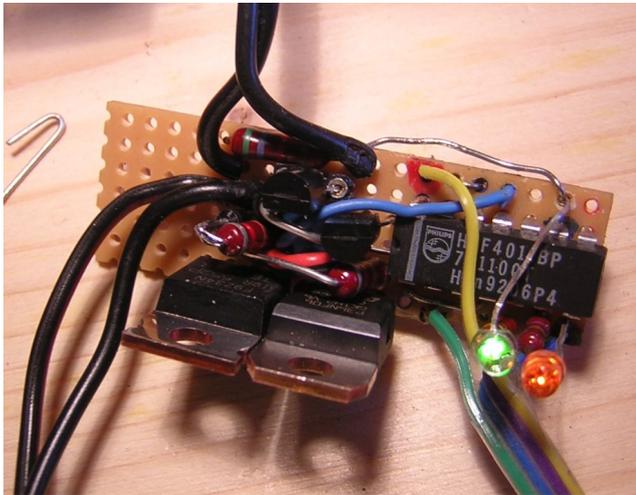


Abb. 5.33: Treiberelektronik

Nach zahlreichen Überlegungen, warum es zu dieser Ansprechstörung von T12 kam, führten wir die Ursache auf die unterschiedlichen Signallaufzeiten in den „NAND-Gatter-Logik“ zurück. Weitere Messungen bestätigten, dass das Gatter U2 im ersten Moment des low-to-high-Wechsels des PWM-Eingangs reagierte, bis das Ergebnis des zeitgleich angesteuerten Gatters U1 nach dessen Signallaufzeit an U2 angekommen war. Theoretisch hat dieser Problemimpuls also genau die Länge der Signallaufzeit eines NAND-Gatters.

Die Knacknuss lösten wir schliesslich durch den Kondensator C12, der im obigen Schaltplan (Abb. 5.31) bereits eingezeichnet ist. Dieser dämpft das

Signal am Ausgang des Gatters U4 stark genug, damit die Transistoren T14 und T12 nicht mehr darauf reagieren. Als positiver Nebeneffekt wurde die zerknitterte Anstiegsflanke des PWM-Eingangs etwas rechteckiger.

Einige weitere Messungen an der Treiber-Elektronik

Um die Geschwindigkeit der gesamten Brückenschaltung zu testen, haben wir für jede Zustandsänderung die Zeitverschiebung zwischen PWM-Eingang und Leistungsausgang gemessen. Damit am Ausgang keine rückwirkenden Induktionsspannungen des Motors auftreten, haben wir als Last einen ohmschen Widerstand angeschlossen. Die Messungen haben wir bei einer PWM-Grundfrequenz von ca. 1960Hz (0,5ms Periodendauer) durchgeführt, was aber keinen Einfluss auf die Reaktions-Messungen hat.

Die Oszillogramme sind jeweils in zwei Teile unterteilt. Der untere Graph stellt das Eingangssignal dar und der obere Graph zeigt das Ausgangssignal, das dann am Motor anliegen würde. In „normaler“ Zeitauflösung von einigen ms pro Bildbreite sehen alle Signale schön rechteckig aus, so wie sie schliesslich auch sein sollten. Bei einer Vergrößerung auf einige wenige μ s pro Bildbreite, zeigt sich aber schon ein ganz anderes Bild. Ideal wäre ein Sprung beider Graphen gleichzeitig von low nach high und umgekehrt. Die nachfolgenden Grafiken können mit der Zeitauflösung nicht miteinander verglichen werden, da wir diese auf den jeweiligen Verlauf angepasst haben. Wir messen jeweils die Reaktionszeit und die Anstiegszeit des Ausgangs-MOSFET's. Dabei wurde bei der Reaktionszeit die Zeit zwischen Eingangssignal-

Start und Ausgangssignal-Start gemessen. Für die Anstiegs- respektive Abfallzeit haben wir die Zeit von Anfang bis Ende der relevanten Signal-Änderung am MOSFET gemessen.

Da man diese Punkte bei solchen Verlaufsformen nicht genau bestimmen kann, haben wir jeweils einen ungefähren Wert festgelegt.

Gas-Ansteuerung

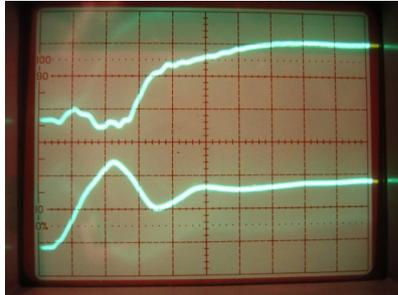


Abb. 5.34: Gas steigend
0.2 μ s-pro Häuschen

Reaktionszeit: 0.5 μ s
Anstiegszeit: 0.85 μ s
gesamte Ansprechzeit: 1.35 μ s

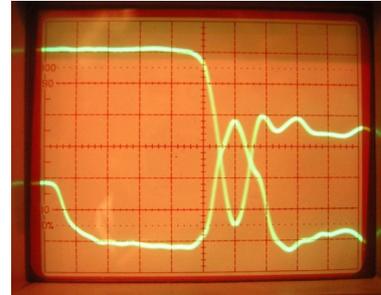


Abb. 5.35: Gas abfallend
0.05 μ s-pro Häuschen

Reaktionszeit: 0.2 μ s
Abfallzeit: 0.08 μ s
gesamte Abfallzeit: 0.28 μ s

Brems-Ansteuerung



Abb. 5.36: Bremse steigend
0.2 μ s-pro Häuschen

Reaktionszeit: 1 μ s
Anstiegszeit: 0.52 μ s
gesamte Ansprechzeit: 1.52 μ s

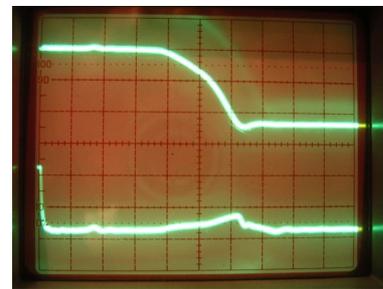


Abb. 5.37: Bremse abfallend
0.5 μ s-pro Häuschen

Reaktionszeit: 1.8 μ s
Abfallzeit: 1.25 μ s
gesamte Abfallzeit: 3.05 μ s

Wie aus den beiden sich deutlich unterscheidenden Abläufen zu erkennen ist, hat der Ausgang der Elektronik offensichtlich eine Rückwirkung auf den PWM-Eingang. Wir konnten uns dies zunächst nicht erklären, haben wir doch extra darauf geachtet, dass Steuer- und Leistungsstromkreis getrennt geführt werden. In Abb. 5.35 ist ein starker Nadelimpuls zu sehen. Die „lange“ Abfallzeit bei der Bremsung (Abb. 5.37) ist auf den Entstör-Kondensator C11 zurückzuführen.

Trotz diesen Schönheitsfehlern funktioniert die Elektronik einwandfrei, und die Reaktionszeiten sind im Vergleich zu der Periodenzeit des PWM-Signals (0,5ms bei 2kHz) sehr klein. Auch die Anstiegs- und Abfallzeiten der MOSFET's, welche hauptsächlich für deren Erwärmung verantwortlich sind, sind akzeptabel.

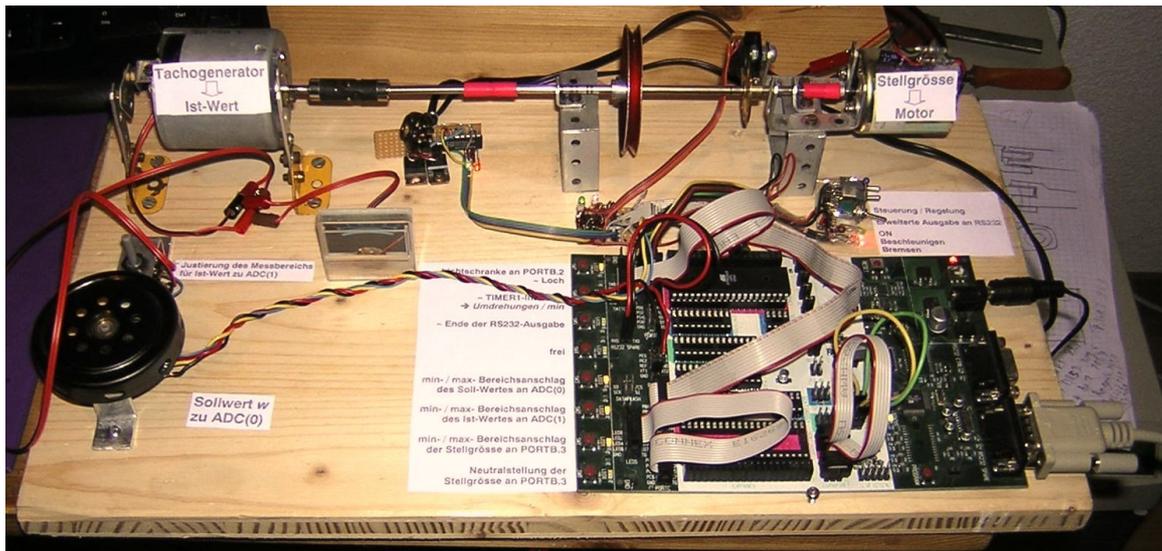


Abb. 5.38: fertiges Regler-Board (Generator und Antrieb absichtlich vertauscht)

5.2.5 Software

Ein Mikrocontroller ist ähnlich aufgebaut wie ein kompletter PC. Er hat verschiedene Speicher (RAM, Flash, EEPROM...), eine CPU, ein internes Bussystem und, abhängig vom Typ, noch viele weitere Features. Er benötigt lediglich 5V-Betriebsspannung, und hat einen geringen Stromverbrauch. Für den Betrieb wird auch sehr wenig Peripherie benötigt. Wir benutzen einen ATmega16 von ATMEL, den wir mit einer Taktfrequenz von 8MHz betreiben. Dieser hat vier Ports à je 8 Pins, womit mehr als genug Anschlüsse für Ein- oder Ausgänge vorhanden sind.



Abb. 5.39: ATmega16

Das Programm haben wir in der Programmiersprache C geschrieben und mit dem C-Compiler „CodeVisionAVR“ [CVAVR] in Assembler und anschliessend in ein HEX-file umgewandelt. Dieses gelangte danach über die RS232-Schnittstelle und das Programmierboard (STK500) in den Mikrocontroller.

Programmbeschreibung

Die folgende Beschreibung des Programms richtet sich nach dessen Quelltext und behandelt Wissenswertes, das aus den kurz gehaltenen Kommentaren nicht ersichtlich ist. Wichtig zu wissen ist, dass sich zwischen `/*` und `*/` oder nach `//` nur Kommentare befinden, die keinen Einfluss auf das Programm haben. Sie werden nicht kompiliert²⁴.

Mit der Preprocessor-Anweisung `#include <header.h>` werden benötigte Routine- und Makrodefinitionen hinzugefügt. Der Preprocessor „reinholt“ den Quellcode von den Kommentaren und fügt die Header-Dateien ein, bevor der Compiler den Code kompiliert. Mit der Analog-Digital-Umwandlung wird in unserem Fall die analoge Spannung vom Potentiometer (0 bis 5 Volt) in eine Digitalzahl mit einer Auflösung von 10bit (0-1023) umgewandelt.

Die globalen Variablen, die am Anfang definiert werden, haben im ganzen Programm Gültigkeit, lokale hingegen nur in einem Unterprogramm oder Programmabschnitt.

Die Funktion `do_step` wird von einem anderen Ort im Programm aufgerufen. Sie übermittelt mit dem Befehl `printf` aktuelle Werte verschiedener Variablen an den Computer, damit dort das Geschehen im Mikrocontroller in einem *Terminal-Fenster* beobachtet werden kann.

Die „External Interrupts“, von denen in unserem Chip maximal drei programmiert werden können, unterbrechen bei Betätigung eines vorgegebenen Eingangs (INT0-2) das Programm, um eine beliebige Funktion abzuarbeiten. Danach wird das Programm an der Unterbrechungsstelle fortgesetzt. Wir brauchen „INT2“ für das Zählen der Löcher der Lochscheibe. Liegt an INT2 eine steigende Flanke an, so wird die Interruptfunktion ausgeführt, welche die Variable „Zaehlerirq0“ um 1 erhöht. Gleichzeitig wechselt eine LED an Port C Pin7 ihren Zustand (von 1 nach 0 oder

²⁴ Übersetzung eines Quelltextes in ein ausführbares Maschinensprache-Programm

umgekehrt). Ausserdem haben wir noch einen „Timer-Interrupt“ programmiert, der nach jeder ¼ Sekunde periodisch ausgeführt wird. Die Funktion rechnet die gezählten Impulse der Lochscheibe in eine Geschwindigkeit (Impulse pro Zeit) um und setzt die Variable `Zaehlerirq0` wieder auf den Startwert 0. Der berechnete Wert wird in einer weiteren Variablen gespeichert, welche für das Reglerprogramm verwendet wird. Gleichzeitig wird jedes zweite Mal, also jede ½ Sekunde, die bereits beschriebene Funktion `„do_step()“` aufgerufen.

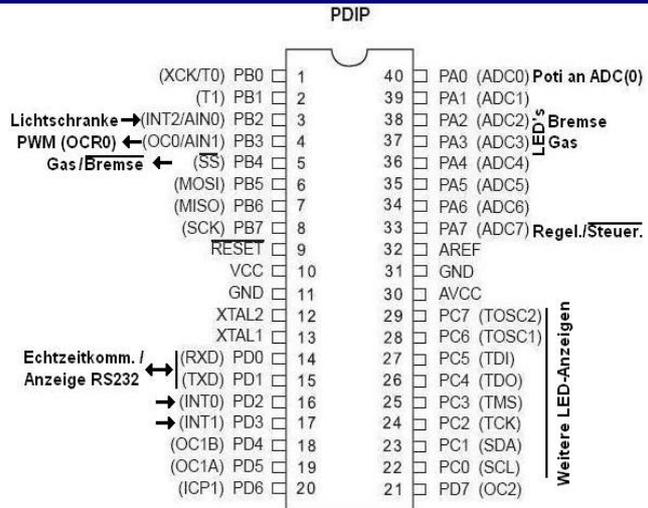


Abb. 5.40: Pinbelegung ATmega16

Nach den Initialisierungen folgt das eigentliche Programm. Mit einem Schalter an Port A, Pin7, lässt sich wählen, ob das Potentiometer direkt die Stellgröße des Motors beeinflusst (Steuerung), oder ob es für die Regelung des Motors den Sollwert festlegt.

Der vorletzte Programmabschnitt wandelt die Stellgröße aus dem Regelungsprogramm (zwischen 0 und 1023) in mehrere Ausgabewerte um. Einerseits wird daraus der Wert für die PWM-Ansteuerung (0-255) berechnet, andererseits werden weitere Bit's für die Treiberelektronik und die Anzeige der Aussteuerung bestimmt.

Die folgende Grafik veranschaulicht, welchen Wert die einzelnen Variablen (int...und PORT...) jeweils annehmen:

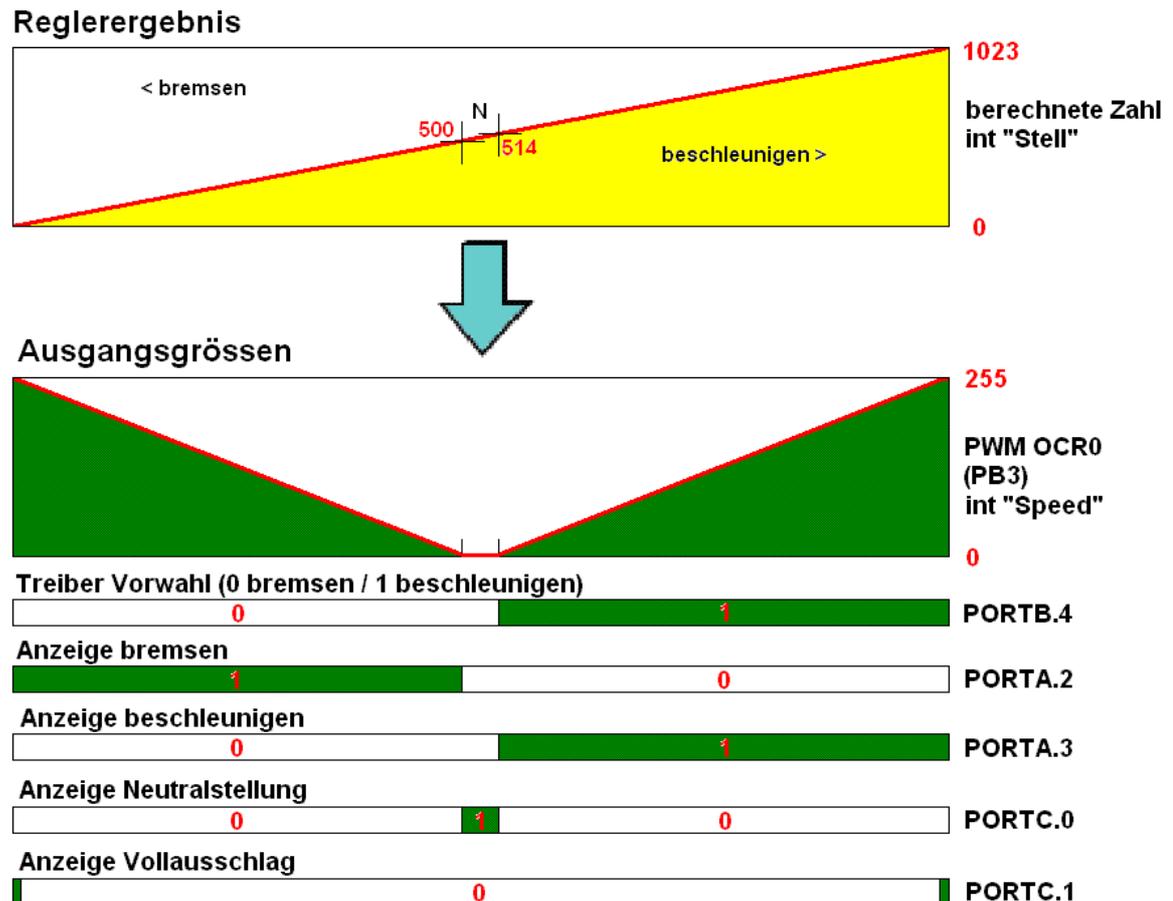


Abb. 5.41: Grafik der Ausgabewerte

Im letzten Programmteil wird noch der auszugebende PWM-Wert (zwischen 0 und 255), welcher bei der vorhergehenden Umwandlung unter der Variablen `Speed` gespeichert wurde, festgelegt. Der Teil des Programms innerhalb der `while`-Schleife (`while(1){[Programmteil]}`) wird wiederholt ausgeführt.

Im folgenden Fenster zeigen wir den kommentierten C-Code mit Ausnahme der Initialisierung für die Hardwarekonfigurationsregister, was einiges an Platz spart.

```

/*****
CodeVisionAVR V1.24.5 Standard
http://www.hpinfotech.com
e-mail:office@hpinfotech.com

Project : Digitaler Regler
Version : PWM 1.0
Date    : 17.12.2006
Author  : C.Vögtli P.Pfeifer
Company : BMS BMA
Comments: Regelungstechnik

Chip type      : ATmega16
Program type   : Application
Clock frequency : 8.000000 MHz
Memory model   : Small
External SRAM size : 0
Data Stack size : 256
*****/
#include <math.h>
#include <delay.h>
#include <stdio.h>
#include <mega16.h>

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;    // Start the AD conversion
    ADCSRA|=0x40;    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

// Declare your global variables here

//Eingabe Lochscheibe und Berechnungen
char *string="ins";
char ins[]="ups";

int soll=0;                //Sollgrösse (0-1023)
int p5=0;                 //für LED
int p6=0;                 //für LED
int p7=0;                 //für LED
int Zaehlerirq0 = 0;      //für LS-Interrupt
int Impulseprosekunde;
int Impulseprominute;
int Umdrehungenprominute;
int Markierungenproscheibe = 8;
int PropAnteil=0;
int Integralanteil=0;
int Differential=0;
int DiffAnteil=0;

```

```

//Ausgabe
int Stell=500;          //Stellgrösse (0-1023)
int Speed=0;           //PWM Ansteuerung des Motors (0-255)

void do_step(void)     //visuelle Ausgabe an RS232 für PC
{
    printf("Sollgr.: %d %s\r\n", soll);

    if(PINA.6==1)     //Weitere Daten nach Wahl mit Schalter
    {
        printf("Stellgr.: %d %s\r\n", Stell);
        printf("Prop-Anteil.: %d %s\r\n", PropAnteil);
        printf("Integralanteil: %d %s\r\n", Integralanteil);
//        printf("Diff-Anteil.: %d %s\r\n", DiffAnteil);
//        printf("Impulse/Sek: %d %s\r\n", Impulseprosekunde, ins);
    }
    printf("Umdreh./Min: %d\r\n", Umdrehungenprominute);
    p5=~p5;
    PORTC.5=p5;       //Blinkende Diode
}

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    if (soll > 0)      //Wenn Wert grösser 0, mache kleiner
        soll=(soll - 100);
    if (soll < 0)     //Wenn Wert zu klein wird, setze auf 0
        soll=0;
}

// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
    if (soll < 1024)  //Wenn Wert kleiner 1024, mache grösser
        soll=(soll + 100);
    if (soll > 1024)  //Wenn Wert zu gross wird, setze auf 1024
        soll=1024;
}

// 2 Liechtschranke-Interrupt - kommt einmal pro loch in der Scheibe
interrupt [EXT_INT2] void ext_int2_isr(void)
{
    Zaehlerirq0++;    //eins dazuzählen
    p7=~p7;
    PORTC.7=p7;      //LED Blinken
}

interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
    p6=~p6;
    PORTC.6=p6;      //AnfangsBlinken
    Impulseprosekunde = 4*Zaehlerirq0;
    Zaehlerirq0 = 0; //Lochscheibe auf Startwert
    Impulseprominute = Impulseprosekunde*60;
    Umdrehungenprominute = Impulseprominute/Markierungenproscheibe;

    //visuelle ausgabe an RS232 für PC (jedes 2.mal)
    if(p6==0)
        do_step();
}

void main(void)

```

```
{
```

Hier stünden einige Initialisierungsroutinen für die Hardware, in denen die richtigen Werte in die Konfigurationsregister ("Softwarejumper") der Hardware geschrieben werden. Diesen Teil haben wir weggelassen, da er aus lauter uns nichts sagenden Hexadezimalwerten besteht.

```
//DAS Programm
while (1)
{
  soll = read_adc(0);          //sollwert einlesen

  if (PINA.7==1)              //REGELUNG nach Poti
  {
    //Regel-Berechnungen

    //Proportionalregler
    PropAnteil=500+(soll-(Umdrehungenprominute));

    //Integralregelung mit festem Anstieg oder Abfall
    if (soll>(Umdrehungenprominute) & Integralanteil<550)
      Integralanteil++;          //Mehr Gas
    if (soll<(Umdrehungenprominute) & Integralanteil>-550)
      Integralanteil=Integralanteil-2; //weniger Gas

    //Differentialregler
    DiffAnteil=(soll-Differential); //Differenz letzter Wert
    Differential=soll;              //aktueller Wert speichern
    delay_ms(5);                   //Integral- und auch Differenzialzeit

    //Regler-Zusammenstellung
    if(1)
      Stell=PropAnteil;            //Proportional
    if(1)
      Stell=Stell+Integralanteil; //Integral dazuzählen
    if(0)
      Stell=Stell+(40*DiffAnteil); //Diff-Anteil dazuzählen
  }
  if (PINA.7==0)              //STEUERUNG per Poti
    Stell = soll;              //Direkt Ansteuerung der Stellgröße über
                              //Soll-Poti

  if(Stell<0)
    Stell=0;                   //Wertbegrenzung nicht kleiner als 0
  if(Stell>1023)
    Stell=1023;                //Wertbegrenzung nicht grösser als 1023

  //Für die PWM-Stellgröße |\_/|
  //PortA.2 und .3 sind Anzeigen Gas/Bremse
  //PortB.4 ist Treiber-Vorwahl PortC.0 ist Neutral
  //PortC.1 ist Vollausschlag
  if( ((Stell>0) && (Stell<=500)) ) // wenn bremsen
  {
    Speed = ( (int) (255- (0.51*(float)(Stell))) );
    PORTA.2=1; PORTA.3=0; PORTB.4=0; PORTC.0=0; PORTC.1=0;
  }
  if( ((Stell>=514) && (Stell<1023)) ) // wenn Gas geben
  {
    Speed = ( (int) (0.501*(float)((Stell)-514)) );
    PORTA.2=0; PORTA.3=1; PORTB.4=1; PORTC.0=0; PORTC.1=0;
  }
  if( ((Stell>500) && (Stell<514)) ) // wenn Speed aus
    // "Neutralstellung"
  {
    Speed = 0;
  }
}
```

```

    PORTA.2=0; PORTA.3=0; PORTB.4=0; PORTC.0=1; PORTC.1=0;
}
if(Stell==0 || Stell==1023)      // "Vollausschlag"
{
    Speed=255;
    PORTC.1=1;
}
// Set the OCR values
OCR0 = (int)(Speed); //Speed PWM (PB3)
OCR2 = (int)(Stell); //Könnte man zur Ansteuerung von
                    // "Verbrauchern und Einspeisern"
                    // des "Generators" brauchen oder
                    // für eine Analoganzeige für
                    // Regeldifferenz:
                    // (Soll-Umdrehungenprominute)
                    // OCR1-Timer wird bereits für
                    // Geschwindigkeitsmessung verwendet
} // end while (1)
} // end main()

```

Abb. 5.42: kommentierter C-Code

Wie dem Quellcode entnommen werden kann, haben wir einen Proportional-Integral-Regler programmiert. Die Regelung funktioniert zwar, jedoch nicht ganz so, wie wir es uns erhofft haben. Durch die niedrige Abtastrate der Geschwindigkeit (4Hz) ist das System instabil - es hört nie auf zu schwingen. Grund dafür ist, dass sich die Regeldifferenz bei konstanter Sollgrösse nur alle 250ms aktualisiert. So ist auch die Stellgrösse immer nur im ersten Moment nach der Aktualisierung korrekt und wirkt mit der Zeit zu stark oder zu schwach. So über- oder unterschwingt die Drehzahl ständig den Sollwert, und pendelt sich nie stabil ein. Durch ein Vergrössern der Integralzeit kann dieses Schwingen zwar sehr stark gedämpft werden, bringt aber eine extreme Trägheit ins System, da sich die Stellgrösse dann nur noch langsam ändern kann.

Würde man die Anzahl der Lochscheibenpulse in kürzeren Abständen abfragen und das Ergebnis mit einer entsprechend höheren Zahl multiplizieren, um wieder auf „Pulse pro Sekunde“ zu kommen, so wird der Wert immer ungenauer, da er jeweils nur ein Vielfaches des Multiplikators sein kann. Die Anzahl Umdrehungen pro Minute sind nur in 30er-Schritten berechenbar. Zählt der Controller ein Loch pro Abfrage (Viertel Sekunde) muss dies zu einer brauchbaren Einheit hochgerechnet werden: mal 4 (pro Sekunde) mal 60 (pro Minute) durch 8 (Löcher pro Umdrehung), das gäbe dann 30 Umdrehungen pro Minute. Bei einer Versorgungsspannung von 12V erreicht das System eine maximale Drehzahl von 930 U/min, was 31 Impulsen pro Abfrage entspricht. So können also lediglich 31 verschiedene Werte ausgerechnet werden.

Die Genauigkeit könnte in diesem Fall durch ein Lochrad mit mehr an Löchern verbessert werden, doch dann wäre unsere Lichtschranke bei hohen Geschwindigkeiten zu träge, da sich folglich auch eine höhere Lochfrequenz ergeben würde.

Eine andere Methode für die Geschwindigkeitsmessung mit derselben Lochscheibe wäre die Zeitmessung von einem bis zum nächsten Loch, doch dies würde im Stillstand nicht mehr funktionieren. So müsste man Idealerweise eine Mischung, respektive eine „Umschaltung“ der Messmethode programmieren.

5.2.6 Schlussfolgerungen

Grundsätzlich sind wir mit unserem Projekt sehr zufrieden, denn es läuft mit dem von uns erstellten Programm genau so, wie wir es uns vorgestellt haben. Was die Anordnung von elektronischen und anderen Bauteilen anbelangt, haben wir eine grundlegende Lehre daraus gezogen. Nun ist uns klar, dass sensible Signalerfassungen einer hohen Übertragungssicherheit und Störfreiheit bedürfen. Die Arbeit mit der Elektronik hat uns auch einiges an praktischer Erfahrung gebracht auf einem Gebiet, mit dem wir in unserem Beruf als Automatiker eher wenig in Berührung kommen.

6 Schluss

6.1 Reflexion

Wir haben am Anfang unserer Arbeit gewusst, dass sich die Regelungstechnik über ein weites Spektrum erstreckt. Dennoch hätten wir nicht gedacht, dass uns das Lernen und Verstehen des Inhalts so viel Zeit abverlangt. Doch unser Interesse schwand deshalb nicht. Im Gegenteil, wir vertieften uns sogar noch mehr in die Materie, als wir am Anfang geplant hatten. Im Fachunterricht, wo wir das Thema fast zeitgleich mit der BMA aufgriffen, konnten wir unsere Kollegen beim Studium der schwierigen Zusammenhänge gut beraten und teilweise sogar für die trockene Angelegenheit begeistern. Auch Diskussionen mit den Fachkräften inspirierten uns, unser Wissen zu erweitern und zu festigen.

Da es undenkbar gewesen wäre, die Arbeit von Hand direkt zu Papier zu bringen, sondern der „Umweg“ über den Computer, der Natürlich mehr Vor- als Nachteile mit sich bringt, vorgegeben war, mussten wir uns natürlich auch etwas mit Software-Fragen auseinander setzen. Die Zusammenarbeit wurde oft mittels des Online-Messaging-Systems MSN Messenger von Microsoft koordiniert und die gesamte Arbeit mittels der „Word“ Software selbiger Firma verfasst. Dabei zeigten sich Anfangs diverse Mängel. Viel versprechende Features wie das Zusammenführen mehrerer Dokumente in einem Zentralkokument oder die einfache Formatierung mittels Formatvorlagen sind wegen unzähliger Unzulänglichkeiten und Fehler im Programm schlicht nicht zu gebrauchen. Für eine Arbeit dieser Grösse, an der mehrere Personen zeitgleich arbeiten, wäre wohl das von Donald Ervin Knuth²⁵ entwickelte Textsatzprogramm „TeX“, bzw. ein darauf basierender „LaTeX“-Editor²⁶ wesentlich besser geeignet. Der Interessanteste Teil der BMA war denn auch die praktische Arbeit am eigenen Modell.

6.2 Ausblick

Für uns ist das Thema mit dem Abschluss der BMA noch nicht beendet, vielmehr wird es uns auf dem weiteren Ausbildungsweg beschäftigen. Wie wir an einem Besuchstag der FHNW selbst erfahren konnten, beschäftigen sich viele Projekte an Universitäten und Fachhochschulen mit den komplexen Problemen, die die Regeltechnik zu bieten hat. Womöglich bietet sich dem einen oder anderen von uns dereinst an einer Fachhochschule die Gelegenheit, bei einem weiteren Projekt im Umfeld der Regelungstechnik, an die hier geleistete Arbeit anzuknüpfen und das Modell noch zu perfektionieren.

²⁵ emeritierter Professor für Informatik an der Stanford University;
Homepage: <http://www-cs-faculty.stanford.edu/~knuth/>

²⁶ <http://www.latex-project.org/>

7 Authentizitätserklärung



Erziehungsdepartement des Kantons Basel-Stadt

Allgemeine Gewerbeschule Basel

▷ Berufsmaturitätsschule

Authentizitätserklärung BMA

Name / Vorname / Klasse: Pfeifer, Patrick, TBM 7b

Name / Vorname / Klasse: Vögtli, Christian, TBM 7b

Name / Vorname / Klasse: _____

Hiermit bestätige ich, die vorliegende Arbeit mit dem Titel

Drehzahlregelung eines Elektromotors

selbst verfasst zu haben. Informationen aus fremden Quellen sind stets durch die entsprechenden Angaben (Zitate, Quellenverzeichnis) gekennzeichnet.

Ort und Datum

Unterschrift Kandidatin / Kandidat

Ort und Datum

Unterschrift Kandidatin / Kandidat

Ort und Datum

Unterschrift Kandidatin / Kandidat

8 Verzeichnisse und Indices

8.1 Bücher-Quellen

- [FET02] Diverse Autoren: „Fachkunde Elektrotechnik“, Europa-Lehrmittel, 2002
- [GAS97] Gassmann: „Regelungstechnik“, Harri Deutsch, 1997
- [BUS99] Busch: „Elementare Regelungstechnik“, Vogel, 1999
- [REG00] Schleicher, Blasinger: „Regelungstechnik“, Jumo, 2000
- [STEIN] Steinegger & Co.: „Bedienungsanleitung Demonstrations-PID-Regelsystem“
- [HOL84] Holbrook: Laplace-Transformation, Vieweg & Sohn, 1984
- [NUS05] Nussbaumer: BSK Messen, Steuern, Regeln; VSAS, 2005

8.2 Internet-Quellen

- [WPKR] Wikipedia Artikel: „Regelkreis“
<http://de.wikipedia.org/wiki/Regelkreis>
- [WPRTK] Wikipedia Artikel: „Regelungstechnik“
<http://de.wikipedia.org/wiki/Regelungstechnik>
- [WPKPL] Wikipedia Artikel: „Komplexe Zahl“
http://de.wikipedia.org/wiki/Komplexe_Zahl
- [WPEUI] Wikipedia Artikel: „Eulersche Identität“
http://de.wikipedia.org/wiki/Eulersche_Identität
- [WPLAP] Wikipedia Artikel: „Laplace-Transformation“
<http://de.wikipedia.org/wiki/Laplace-Transformation>
- [WPÜTF] Wikipedia Artikel: „Übertragungsfunktion“
<http://de.wikipedia.org/wiki/Übertragungsfunktion>
- [WPPSL] Wikipedia Artikel: „Pierre-Simon Laplace“
http://de.wikipedia.org/wiki/Pierre-Simon_Laplace
- [WPNEW] Wikipedia Artikel: „Isaac Newton“
http://de.wikipedia.org/wiki/Isaac_Newton
- [WPTRA] Wikipedia Artikel: „Trägheit“
<http://de.wikipedia.org/wiki/Trägheit>
- [WPLZI] Wikipedia Artikel: „LZI-System“
<http://de.wikipedia.org/wiki/LZI-System>
- [WPBSB] Wikipedia Bild: „Regelblockschaltbild.png“
<http://de.wikipedia.org/wiki/Bild:Regelblockschaltbild.png>
- [SCHUM] Schumacher: „Regelungstechnik“ Vorlesungsunterlagen
http://www.fh-oow.de/fbi/we/al/pdf/vorlesungen/rt/rt_vl.pdf
- [RNPWM] RN-Wissen: „Pulsweitenmodulation“
<http://www.roboternetz.de/wissen/index.php/Pulsweitenmodulation>
- [KENLA] Kessler: „Nutzlosigkeit der Laplace-Transformation“
<http://www.home.hs-karlsruhe.de/~kero0001/Laplace/LaplaceNutzlos.html>
- [ERFUZ] Folien der Universität Erlangen zum Thema Fuzzy-Logik
<http://www8.informatik.uni-erlangen.de/IMMD8/Lectures/NEURO-FUZZY/ss2001/v20.ps.gz>
- [TARTS] Taeschler: „Regelungstechnik“ Skript
http://www.tfh-berlin.de/%7Emerkel/RT1/skript_RT1.pdf
- [BEDIP] Berger: Diplomarbeit
„Realisierung einer prototypischen Hardwarelösung für ein inverses Pendel“
http://www-user.tu-chemnitz.de/~beber/DA/Diplomarbeit_IP.pdf
- [GOFUZ] Goebel: „An Introduction To Fuzzy Control Systems“
<http://www.fags.org/docs/fuzzy/>
- [SCIHP] „Scilab Home Page“
<http://www.scilab.org/>
- [MATHP] „The MathWorks - MATLAB and Simulink for Technical Computing“
<http://www.mathworks.com/>
- [SGPAR] Schellinger: „Aufgaben beim automatischen Einparken“
<http://www.schellinger.de/robotik/parken/index.html>

8.3 Liste verwendeter Software

- [TIPRO] **Tina Pro (Student Version)**
Schaltplaneditor mit Testfunktionen
- [CVAVR] **CodeVision-AVR 1.24.5**
C-Compiler-Programm für ATMEL-Mikrocontroller
- [AVRST] **AVR Studio 4**
Assembler-Compiler und Tool für Programierboard
- [SCILB] **SciLab 4.0**
Berechnung und Erstellung von Kurvendiagrammen
- [PHSHP] **Adobe Photoshop CS 7**
Foto Bearbeitung
- [MSOPM] **MS Office Picture Manager**
Foto Bearbeitung
- [MSPNT] **MS Paint**
Bitmap Editor
- [INKSC] **Inkscape 0.44.1**
Vektor Programm
- [ACROW] **Adobe Acrobat Writer 6.0 Pro**
PDF-Reader und Editor
- [MSWRD] **MS Office Word**
Textverarbeitung
- [MSPPT] **MS Office Power Point**
Presentation
- [MSMSN] **MS MSN Messenger**
Echtzeit Kommunikation

8.4 Abbildungsverzeichnis

✚ Toilettenspülung	4
✚ Temperaturregelung	4
✚ Blockschaltbild eines Regelkreises [WPBSB]	6
✚ Mit Scilab generierte Sprungantwortdiagramme	9
✚ Die Hysterese	10
✚ Programm für Scilab	13
✚ Fuzzy-Logic	13
✚ Flachsreiber und Demonstrations- PID-Regelsystem	14
✚ Anschlüsse für die Analog-Ausgabe	15
✚ Blockschema des Demo-Regelsystems	16
✚ Unser Arbeitsplatz in Biel	16
✚ P-Regler	17
✚ I-Regler	17
✚ PI-Regler	18
✚ PD-Regler	18
✚ PID-Regler 1	19
✚ PID-Regler 2	19
✚ Antrieb frei aufgehängt	20
✚ Antrieb neu starr montiert	20
✚ out ca.30Hz	22
✚ out ca.170Hz	22
✚ Motor mit 4 μ F	23
✚ Motor mit 1 μ F	23
✚ Motortreiber	26
✚ Gas steigend	28
✚ Gas abfallend	28
✚ Bremse steigend	28
✚ Bremse abfallend	28
✚ fertiges Regler-Board	29
✚ Grafik der Ausgabewerte	30
✚ kommentierter C-Code	34