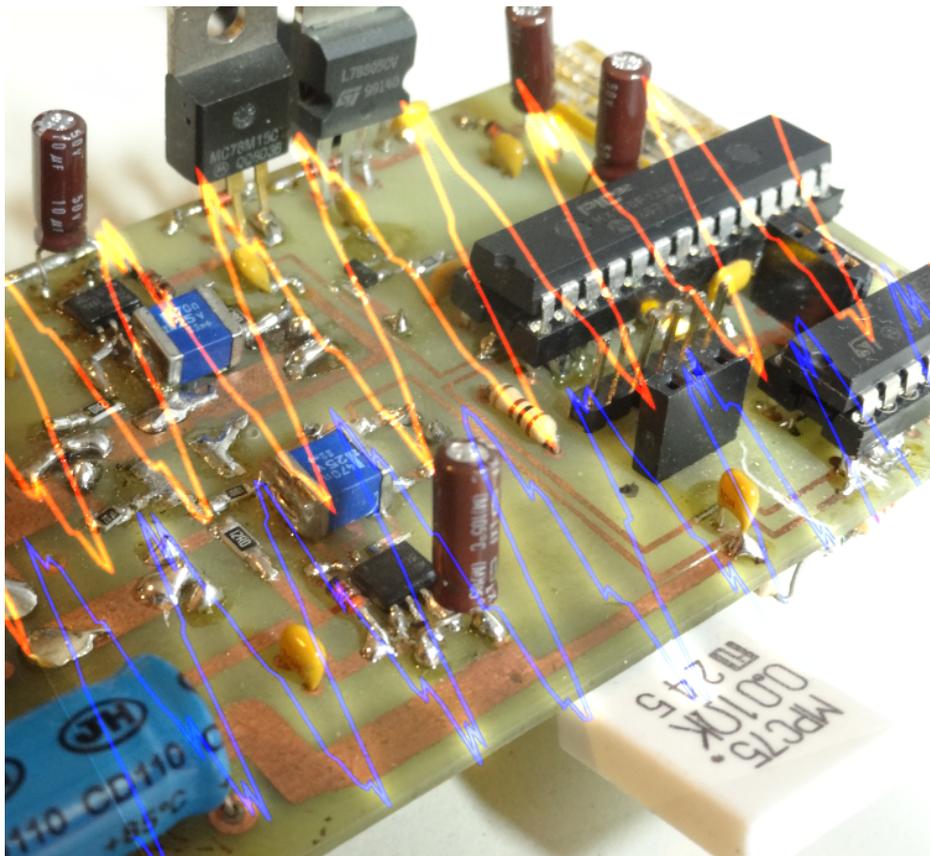

Digitale sensorlose Servoregelung für Gleichstrommaschinen



Wettbewerb „Jugend forscht“ 2014

Lucas Jürgens (18)

Arbeitsgemeinschaft „Jugend forscht“ des
Christian-Gymnasiums Hermannsburg
Leitung: StD Thomas Biedermann

Inhaltsverzeichnis

1	Allgemeines und Grundlagen	1
1.1	Einleitung	1
1.2	Versuchsmotor	1
1.3	Systemübersicht	2
2	Elektronik	3
2.1	Schaltung	3
2.2	Leistungsteil	4
2.3	Strommessung	5
3	Software	6
3.1	Datenübertragung	6
3.2	PC	7
3.3	Mikrocontroller	7
4	Regelungssystem	8
4.1	Drehzahlbestimmung	8
4.2	Positionsbestimmung	11
4.3	Regelungsalgorithmus	12
4.4	Positionierung	13
4.5	Regelverhalten	14
5	Ausblick	15
6	Literatur und Quellen	16
7	Danksagung	16

1 Allgemeines und Grundlagen

1.1 Einleitung

Elektromotoren ermöglichen die Umwandlung von elektrischer in mechanische Energie und finden in den verschiedensten Bereichen Anwendung. Ähnlich vielfältig ist auch das Angebot an Motortypen. Zu den bekanntesten gehören die Schrittmotoren (z.B. in Druckern, Scannern), Gleichstrommaschinen (mit Bürsten-Kommutator, z.B. bei Spielzeug, Scheibenwaschanlage) oder Bürstenlose Gleichstrommotoren („Brushless“, z.B. im Profi-Modellbau, Elektroautos), welche sich in Preis, Leistungsdichte, Drehzahlbereich oder Stell-Präzision unterscheiden. Besonders preisgünstig im niedrigeren Leistungsbereich sind die Bürsten-Gleichstrommotoren, die einfach mit Gleichspannung betrieben und über ein großen Drehzahlbereich stufenlos gesteuert werden können.

Für viele Anwendungen reicht es bereits aus, die Spannung fest einzustellen und den Motor so mit einer ungefähren Drehzahl zu betreiben. Für Anwendungen bei denen die Drehzahl genauer eingestellt werden muss, kommen dem Motortyp angepasste, elektronische Regelungen zum Einsatz, die die Drehzahl messen und die Motorspannung entsprechend nachregeln. Oftmals können solche Regelungen auch bestimmte Winkellagen einstellen. Solche Regelungen im Verbund mit einem Motor werden auch als Servo bezeichnet. Sie kommen dort zur Anwendung, wo definierte Drehwinkel eingestellt werden müssen, wie z.B. in der Industrie bei CNC-Maschinen, Lenkungen oder Robotern.

Zur Bestimmung der Drehzahl bzw. des Drehwinkels werden dabei Tachogeneratoren (induktiv), spezielle Inkremental-Drehgeber (meist optisch), Potentiometer oder Resolver (induktiv) verwendet. Die Genauigkeit der eingesetzten Sensoren bestimmt dabei die maximale Genauigkeit des Servos. Besonders bei kleineren Anwendungen übertrifft der Preis der Sensoren den der eigentlichen Motoren. Bei der Untersuchung eines Gleichstrom-Getriebemotors konnte ich feststellen, dass aus dessen Stromverlauf die Rotorposition ablesbar ist. Durch Ausnutzung dieser Tatsache könnte ein externer Sensor entfallen, was die Kosten eines kleineren Servosystems stark senken würde. Das mögliche Anwendungsgebiet wäre groß.

Mit diesem Projekt wurde ein Verfahren und ein Prototyp eines Motortreibers entwickelt, welcher es ermöglicht, den Drehwinkel einer bürsten-kommutierten Gleichstrommaschine ohne die Verwendung eines exter-

nen Sensors genau und reproduzierbar einzustellen.

1.2 Versuchsmotor

Zur Entwicklung des Treibers wird eine preisgünstige Motor-Getriebe-Kombination (Abbildung 1) der Firma *Doga* eingesetzt. Es handelt sich dabei um eine bürsten-kommutierte, permanent erregte Gleichstrommaschine, ausgelegt für eine Nennspannung von 24 V mit einem Nominalstrom von 2 A. Der Motor hat eine Nominaldrehzahl von 25 RPM, das Getriebe eine Untersetzung von 62:1. [4]

Abbildung 3 zeigt den schematischen Aufbau des Motors. Dieser besteht aus dem unbeweglichen Stator, der durch Permanentmagneten oder Erregerspulen ein Magnetfeld hervorruft, dessen Feldlinien den Rotor durchsetzen und senkrecht zu den Leitern der Ankerspule stehen. Durch den Kommutator werden die Ankerspulen entsprechend des Drehwinkels so gepolt, dass es durch die Lorentzkraft stets zu einer Kraftwirkung auf den Rotor kommt, welche sich positiv auf dessen Drehmoment auswirkt. In der Praxis werden meist mehrere Ankerspulen verwendet, die jeweils mit einem Lamellenpaar verbunden sind. Bei einer Umpolung der Ankerspulen durch den Kommutator kommt es zu mehr oder weniger stark ausgeprägten Stromeinbrüchen, die man bei Messung des Stroms beobachten kann.



Abbildung 1: Der verwendete Getriebemotor mit Anschlusskabel.

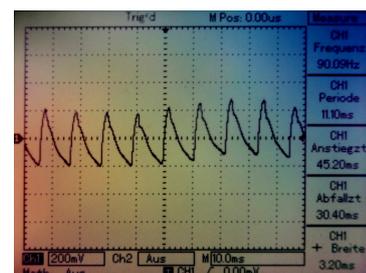


Abbildung 2: Stromripple des Motors bei 10 V Betriebsspannung. Y-Achse: 15 mA/DIV

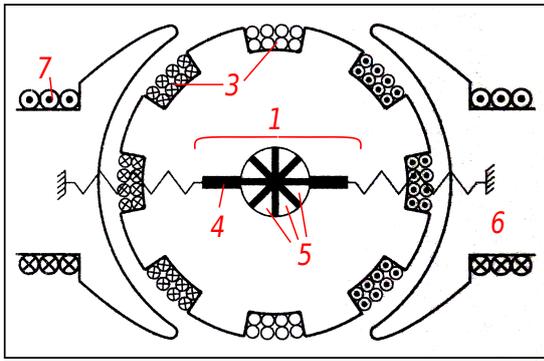


Abbildung 3: Aufbau einer Gleichstrommaschine, hier durch Erregerspule (7) erregt. Mit Kommutator (1), Ankerspulen (3) in Nuten, angepresste Kohlebürsten (4), 8 Lamellen (5) und Stator (6). Quelle: [8](Abb. 3.22), nachträglich beschriftet.

Abbildung 2 zeigt die Stromänderungen bei einer Betriebsspannung von 10 V. Der mittlere Motorstrom betrug 900 mA, der Stromripple ca. 30 mA *peak-to-peak* mit einer Frequenz von 90 Hz. Der Strom wurde über einen Messwiderstand mit einem Oszilloskop und AC-Kopplung gemessen. Die Stromschwingungen sind Drehzahlabhängig und betragen 3% bis 15% des mittleren Motorstroms. Bei sehr niedrigen Spannungen (<4 V) haben die Strompeaks eine andere Form. Bei dem verwendeten Motortyp steigt die Drehzahl annähernd linear mit

der Ankerspannung.

1.3 Systemübersicht

An das Gesamtsystem sind eine Vielzahl von Anforderungen gestellt. Der fertige Motortreiber muss zuverlässig und ausfallsicher konzipiert und auch ohne externen Computer lauffähig sein. Dafür wurde die Motorsteuerung in einem Mikrocontroller implementiert, der mit der zugehörigen Elektronik den Motortreiber bildet. Alle Regelkreisläufe laufen selbstständig innerhalb des Motortreibers und können somit nicht ohne Weiteres von Außen beeinflusst werden.

Um Regelbefehle von Außen entgegen nehmen zu können besitzt der Motortreiber eine USB-Schnittstelle. Über diese können mithilfe der entwickelten PC-Software Kommandos wie „Position xy anfahren“, „konstante Drehzahl halten“, „Stoppen“, sowie Parameter eingestellt werden. Zusätzlich wurde ein Softwareteil entwickelt der es ermöglicht, beliebige Regel-Variablen im Motortreiber in Echtzeit zu visualisieren, was die Entwicklung von Regelalgorithmen für den Treiber erleichtert.

Weitere Anforderungen an den Treiber sind ausreichende Stellgenauigkeit und eventuell Regelgeschwindigkeit, sowie Anpassbarkeit an verschiedene Motoren.

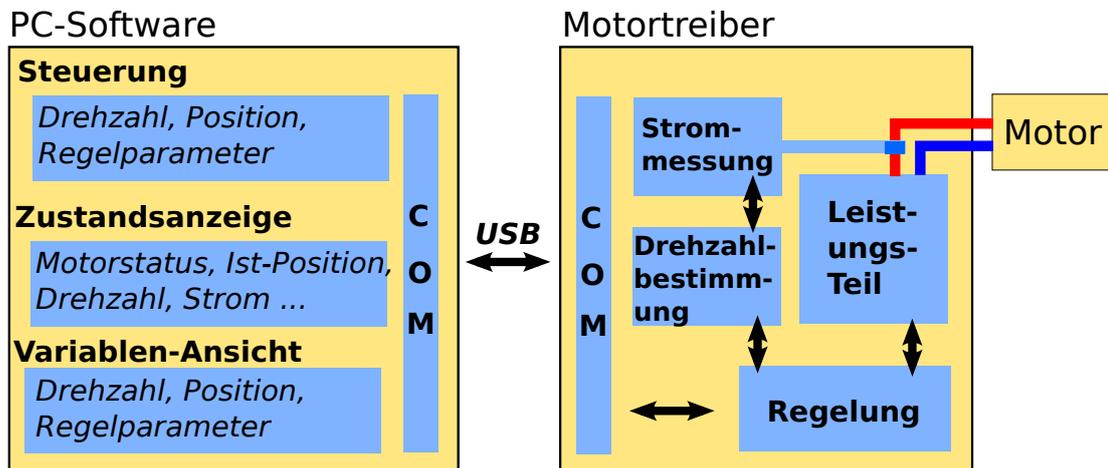


Abbildung 4: Übersicht der Komponenten des Motortreibers mit Treibersteuerung

Abbildung 4 zeigt schematisch den logischen Aufbau des Motortreibers mit angeschlossenem PC. Der Motortreiber erhält über den USB-Port Steuersignale, die weiter an die Software-Regelung gegeben werden. Die Regelung stellt mithilfe des Leistungsteils eine Mo-

torspannung ein. Der Motorstrom wird gemessen und daraus die Drehzahl-Information bestimmt, welche wiederum an die Regelung gegeben wird.

2 Elektronik

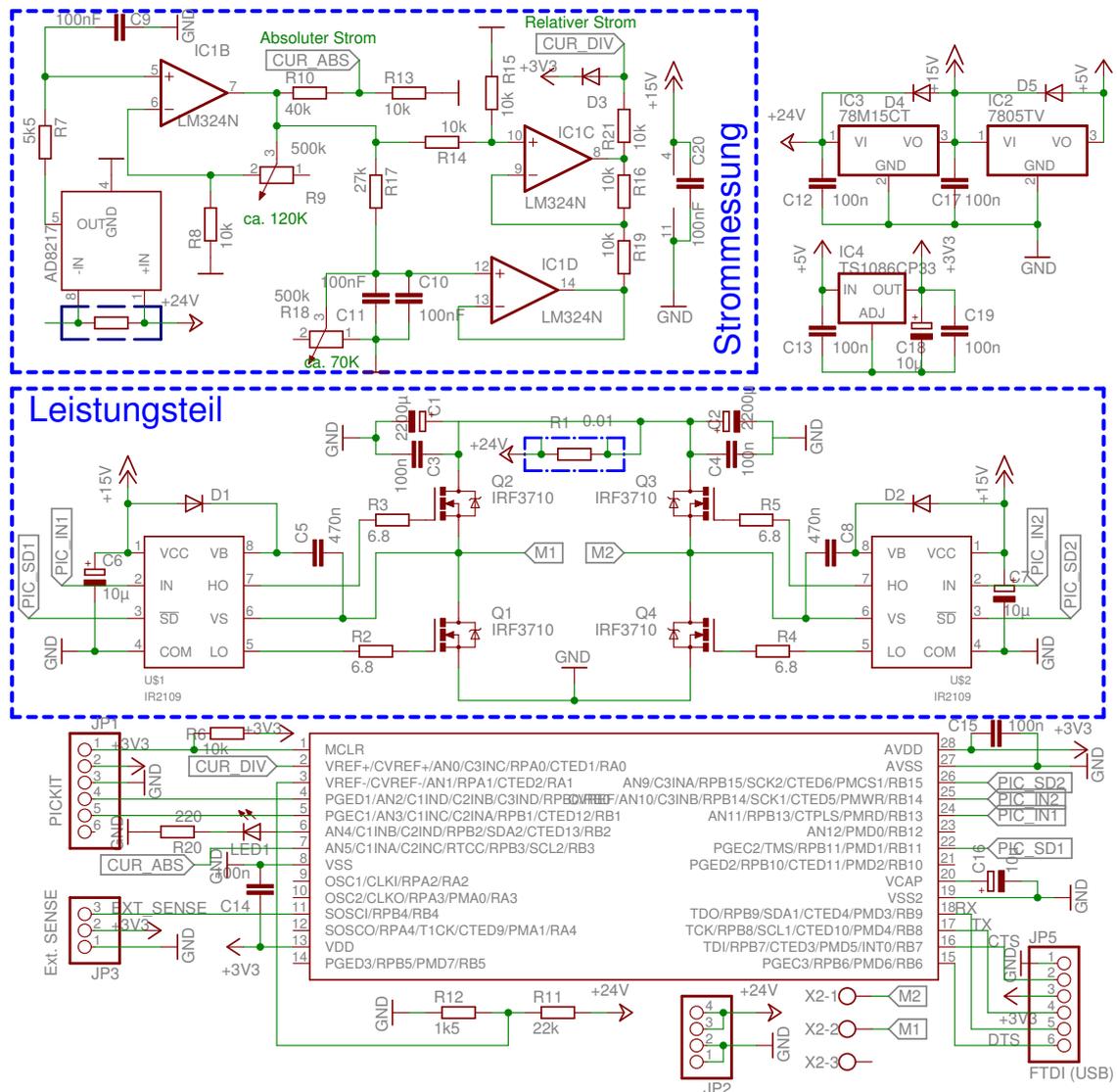


Abbildung 5: Vereinfachter Schaltplan der Motortreiber-Elektronik

2.1 Schaltung

Der Schaltplan zeigt die Motortreiber-Elektronik. Die markierten Bereiche *Leistungsteil* und *Strommessung* werden in den nachfolgenden Unterkapiteln (2.2, 2.3) beschrieben.

Die für den Betrieb der einzelnen Komponenten nötigen Spannungen (3,3 V, 5 V und 15 V) werden mithilfe von drei linearen Spannungsreglern (oben links im Schaltplan) aus der Eingangsspannung (hier 24 V) hergestellt. Die Schaltung ist ausgelegt für eine Betriebsspannung von 17 - 35 V. Die Spannungsversorgung muss dabei nicht unbedingt stabilisiert sein.

Im unteren Teil des Schaltplans ist der Mikrocontroller abgebildet, welcher mit einer Versorgungsspannung von 3,3 V betrieben wird. Direkt vor den Pins der Versor-

gungsspannung werden HF-Anteile durch 100 nF Kerkos abgeblockt. Diese haben sich bei ersten Versuchen mit dem Leistungsteil als zwingend notwendig erwiesen, da es bei einem Ein- und Ausschaltvorgang des Motors zu starken Einstreuungen in die Versorgungsspannung kam. Als Mikrocontroller wird der Typ *PIC32MX150F128B* der Firma *Microchip* im DIP24-Gehäuse verwendet. Dank einer Befehlsbreite von 32-bit und einer maximalen Taktfrequenz von 40 MHz können floating-point-Operationen, wie sie für die Regelungssoftware benötigt werden, besonders schnell durchgeführt werden. Er verfügt unter anderem über 21 I/O-Pins, sowie einem 10-bit Analog-zu-Digital-Wandler (ADC) mit 10 Kanälen, vier 16-bit Timern und einer UART-Schnittstelle. Er bietet 128 KByte Programm-, sowie 32 KByte Datenspeicher. [5]

Der Mikrocontroller ist mit verschiedenen Steckerbuchsen verbunden. Über den Verbinder "JP1, PICKIT 3,, kann der Treiber mit einem externen Programmer-Tool verbunden werden, mit dem der Mikrocontroller mit dem Betriebsprogramm geflasht wird. Über "JP3, 'Ext. Sense,, kann ein externer Positionssensor und über "JP5, FTDI,, das USB-zu-RS232-Modul angeschlossen werden. Eine LED an Pin 6 bietet eine Ausgabemöglichkeit. Über den Analog-zu-Digital-Wandler-Eingang Pin 3 wird über einen passenden Spannungsteiler die Versorgungsspannung gemessen.

2.2 Leistungsteil

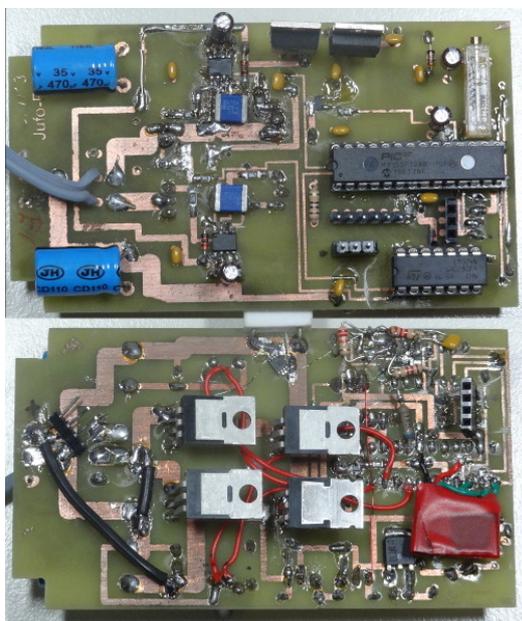


Abbildung 6: Elektronik des beschriebenen 1. Prototyps

Um die Drehzahl des Motors beeinflussen zu können, muss die Elektronik in der Lage sein, dessen Spannung einzustellen. Diese Aufgabe übernimmt der Leistungsteil der Elektronik (siehe Schaltplan, Abbildung 5). Hier wird die am Motor anliegende Spannung pulswertenmoduliert. Bei der Pulsweitenmodulation (PWM) wird die Spannung mit einer konstanten Frequenz zwischen 0 V und Betriebsspannung (V_b) geschaltet. Die Pulsweite - das Verhältnis aus Einschaltzeit zu Ausschaltzeit - wird nach Bedarf variiert. So kann die im zeitlichen Mittel an den Verbraucher abgegebene Leistung eingestellt werden. Die Frequenz der PWM muss dabei hoch genug sein, dass es durch die PWM allein nicht zu einer Drehzahländerung kommt. Die Spannung wird dann zum Teil durch die Induktivität der Motorwicklung geglättet und die Drehzahl durch die Rotorträgheit stabilisiert. Je hö-

her die PWM-Frequenz gewählt wird, desto größer sind allerdings auch Schaltverluste der Transistoren. Bei diesem Projekt hat sich eine Frequenz von 2,5 kHz bewährt.

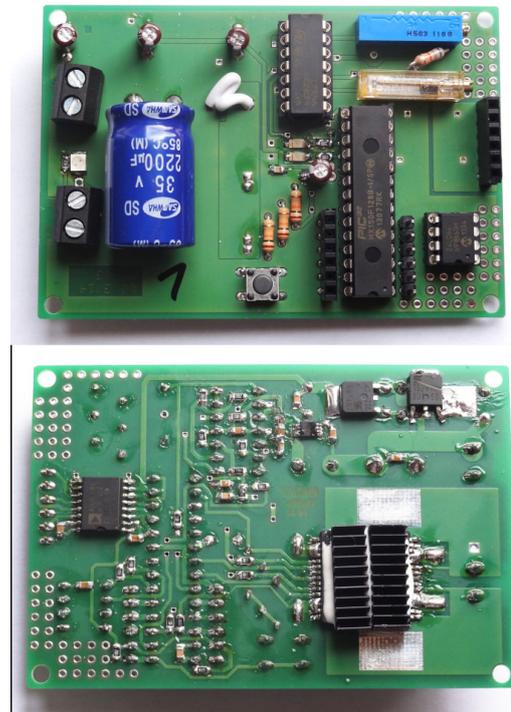


Abbildung 7: Elektronik des 2. Prototyps

Je nach Motor muss die Elektronik hohe Ströme leisten. Der Leistungsteil des Treibers wurde für Ströme bis zu 15 A ausgelegt. Zum Schalten der Spannung werden kostengünstige *N-Channel Mosfets* eingesetzt. Das an diesen Bauteilen anliegende Potential zwischen *source* und *gate* steuert den Stromfluss zwischen *drain* und *source*. Die am Motor anliegende Spannung muss für eine Drehrichtungsänderung in der Polarität geändert werden können, weshalb vier solcher Mosfets in einer H-Brücken-Konfiguration zum Einsatz kommen. Zwei Mosfets schalten dabei gegen V_b , zwei gegen GND. Im Betrieb sind zwei diagonale Mosfets aus jeweils einer Halbbrücke aktiv, wobei nur der V_b -Mosfet PWM-Moduliert wird. Das Umschalten der Mosfet-Gates zwischen den zwei Zuständen muss sehr schnell erfolgen. Befindet sich das Gate in einem Zwischenzustand kommt es zu hohen Verlusten im Innern, was das Bauteil zerstören kann, wie sich bei anfänglichen Versuchen auch recht schnell gezeigt hat. Aus diesem Grund werden zwei Halbbrückentreiber *IR2109* verwendet, welche das Gate mit internen Transistoren schnellstmöglich umladen. Der Treiber verhindert dazu durch eine eingebaute Totzeitschaltung den Schaltzustand, in dem beide Mosfets einer Halbbrücke aktiv sind. Es würde zum Kurzschluss kommen.

Für die V_b -Mosfets liefern die Treiber eine Spannung relativ zu deren Source-Anschluss, der nicht wie bei den GND-Mosfets auf GND liegt, sondern am Motor angeschlossen ist (High-Side Mosfet). Hierzu benötigen die Treiber einen *Bootstrap*-Kondensator und -Diode, deren Werte entsprechend Datenblatt dem verwendeten Mosfet angepasst werden müssen [1]. Die Treiber werden mit einer Spannung von 15 V betrieben und schalten mit dieser auch die Mosfets weit über deren Gate-Threshold hinaus. Die Steuereingänge der ICs werden direkt vom Mikrocontroller angesteuert, welcher das PWM-Signal generiert.

Die Spannung V_b wird pro Halbbrücke mit einem 2200 μF -Elko geglättet und HF-Anteile mit einem 100 nF Kerko gedämpft.

2.3 Strommessung

Für die Regelung muss der Strom, der den Motor durchfließt, bestimmt werden. Hierfür bieten sich verschiedene Verfahren an, wie zum Beispiel das Messen der Stärke des Magnetfeldes um einen Leiter zum Motor durch Hall-Sensoren. Bei diesem Projekt wird ein Messwiderstand verwendet, bei dem proportional zum Strom eine Spannung abfällt. Je größer der Widerstand, desto größer und somit besser messbar die abfallende Spannung, umso größer allerdings auch die Verlustleistung. Es wird ein Messwiderstand mit 30 m Ω verwendet.

$$U_m = R \cdot I_m = 30\text{m}\Omega \cdot 2\text{A} = 60\text{mV}$$

$$P_V = R \cdot I_m^2 = 120\text{mW}$$

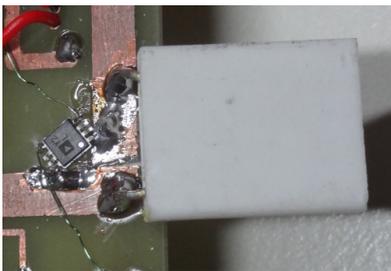


Abbildung 8: Differenzverstärker *AD8217* in SMD-Gehäuse direkt am Messwiderstand

Der Messwiderstand befindet sich zwischen positiver Versorgung der H-Brücke und V_b (R1, siehe Schaltplan, Abbildung 5), weshalb am Widerstand, unabhängig von der Polarität am Motor, eine positive Spannung abfällt. Um die sehr kleine Spannungsdifferenz messen zu können, wird ein spezieller Differenzverstärker „AD8217“ eingesetzt, welcher die Differenz 20-fach verstärkt und

diese als Potential bezogen auf GND am Ausgang liefert [3]. Durch die PWM entspricht die Ausgangsspannung nicht dem tatsächlichen Motorstrom und muss deshalb geglättet werden. Dies passiert analog durch ein nachfolgendes RC-Tiefpass (R7, C9). Dessen Grenzfrequenz ist zu 290 Hz bemessen, sodass der PWM-Anteil geglättet wird, aber schnelle Stromänderungen trotzdem festgestellt werden können. Das geglättete Signal wird anschließend durch einen als nicht-invertierenden Verstärker beschalteten Operationsverstärker verstärkt (IC1B). Dessen Ausgangsspannung U_m wird nach einem Spannungsteiler (CUR-ABS) vom Mikrocontroller digitalisiert und liefert den Wert des absoluten Motorstroms. Die Schaltung ist so bemessen, dass bei einem Strom von ca. 2 A der maximale Digitalwert gemessen wird, welcher einer Spannung von 3,3 V entspricht.

$$U_m = R_m \cdot I \cdot a_{AD8217} \cdot a_{IC1B} \cdot a_{Spannungsteiler}$$

$$a_{IC1B} = 1 + \frac{R_9}{R_8} [9]; a_{Spannungsteiler} = 1 \div \left(1 + \frac{R_{13}}{R_{10}}\right)$$

$$U_m = 30 \text{ m}\Omega \cdot 2 \text{ A} \cdot 20 \frac{\text{V}}{\text{A}} \cdot \left(1 + \frac{120\text{k}}{10\text{k}}\right) \div \left(1 + \frac{40\text{k}}{10\text{k}}\right) \approx 3,3 \text{ V}$$

Die Verstärkung muss dafür über das Potentiometer R9 angepasst werden. Mit einer ADC-Auflösung von 10-bit ergibt sich so eine maximale Messauflösung von $\Delta I_{abs} = 2 \text{ A} \div 2^{10} \approx 1,95 \text{ mA}$. Ist der absolute Strom sehr gering, wie es bei niedrigen Motordrehzahlen der Fall ist, können die Stromschwankungen nicht mehr mit ausreichender Auflösung digitalisiert werden. Um die Messgenauigkeit für Stromschwankungen zu erhöhen, wird mit dem Differenzverstärker (IC1C) eine Differenz zwischen dem verstärkten Strom und einem Spannungswert der knapp unterhalb der Stromschwankungen liegt gebildet. Letzterer wird von einem RC-Glied (R17, C10-11) mit niedriger Grenzfrequenz über ein Impedanzwandler (IC1D) geliefert. Über das einstellbare Potentiometer R18 kann der Spannungsoffset bezüglich des geglätteten Stromwertes nach unten angepasst werden. Die Differenz wird durch die Diode D3 auf 3,8 V begrenzt und vom Mikrocontroller als Eingangswert für die Drehzahlerkennung digitalisiert. Die Stromauflösung ergibt sich aus der Verstärkung des Verstärkers IC1B.

$$\Delta I_{div} = \Delta I_{abs} \div a_{Spannungsteiler} \approx 9,77 \text{ mA}$$

Aufgrund der hohen Verstärkung des Systems muss die Leitungs-Masche *Differenzverstärker - Messwiderstand* möglichst klein sein, um Einstreuungen zu vermeiden. Aus diesem Grund wurde das IC so mit einem sehr geringen Abstand zum Messwiderstand positioniert (siehe Abbildung 8).

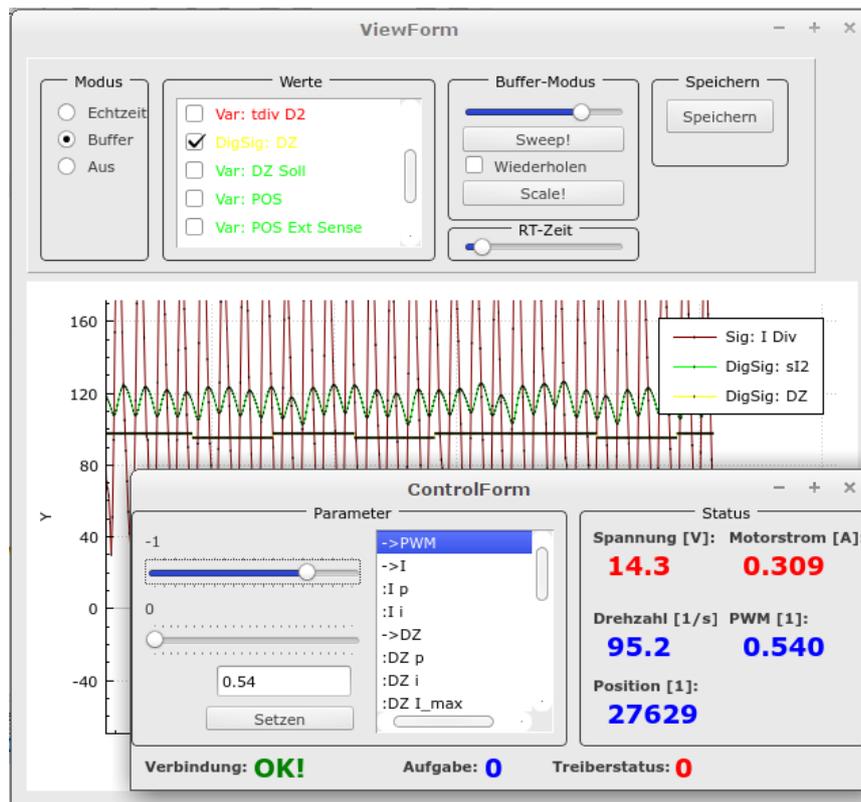


Abbildung 9: Screenshot der PC-Software. ViewForm: Ansicht der Regelvariablen in Echtzeit (X-Achse: Sekunden). ControlForm: Einstellen von Regelparametern, Ablesen des Treiberzustandes.

3 Software

3.1 Datenübertragung

Zur Steuerung des Motortreibers über den PC wird der USB-Port verwendet, welcher als moderne Schnittstelle von fast jedem Computer unterstützt wird. Die USB-Schnittstelle ist aufgrund ihres Aufbaus in Hardware und Software bei mittleren Kabellängen sehr fehlerresistent und ermöglicht dazu hohe Datenübertragungsraten. Auch können mehrere USB-Geräte über einen USB-Hub an einen PC angeschlossen werden. Der PC fungiert als Host, der die Koordination der Peripherie-Geräte übernimmt.

Der verwendete Mikrocontroller verfügt nicht selbst über eine USB-Schnittstelle, weshalb ein spezieller USB-zu-RS232-Adapter (IC: FT232R der Firma FTDI) verwendet wird. Der Mikrocontroller kommuniziert über zwei Datenleitungen (RX, TX) per serieller UART-Schnittstelle mit dem FTDI-Chip, der sich um die Abwicklung des USB-Protokolls kümmert. Bedingt durch die USB-Architektur können Daten nicht in jedem Moment übertragen werden. Möchte der FTDI-Chip ein USB-Datenpaket senden, setzt dieser ein USB-Request ab, erst nach Empfang einer Sendefreigabe darf das Paket gesendet werden. Möchte andersherum ein Pro-

gramm auf dem PC ein Datenpaket senden, wird dieses in eine Liste (USB-Scheduler) geschrieben und der USB-Treiber entscheidet, wann dieses abgeschickt wird. Dadurch kann es zu Latenzzeiten im ms-Bereich kommen. Damit ein Verlust von Datenpaketen aufgrund dieser Latenzen verhindert wird, werden zwei zusätzliche Datenleitungen zur Flusskontrolle zwischen Mikrocontroller und FTDI verwendet, über welche der Mikrocontroller eine Sendeanfrage signalisiert und der FTDI später mit einer Sendeerlaubnis antwortet. Diese Leitung wird auch dazu verwendet den USB-Sendebuffer des FTDIs direkt zu übertragen, was die Latenzzeit reduziert. [2]

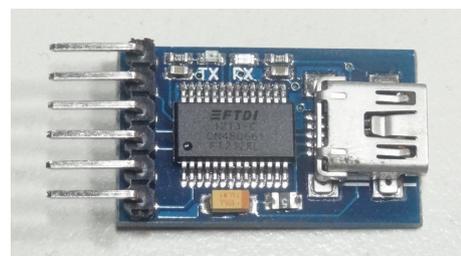


Abbildung 10: Verwendetes USB-Adapter-Modul mit FTDI-Chip

Auf der Seite des PCs kann der FTDI-Chip über einen speziellen USB-Treiber angesprochen werden, wo-

für eine dynamische Bibliothek des Herstellers geladen werden muss. Die Verwendung einfacher USB-zu-Seriell-Adapter, die als virtuelle COM-Ports angesprochen werden, hat sich als sehr fehleranfällig erwiesen. Oftmals sind einzelne Datenpakete nicht übertragen worden und die maximale Datenrate war gering. Mit dem FTDI-Chip wird eine Datenrate von 921600 Baud erreicht.

Zwischen Mikrocontroller und PC können Byte-Arrays bis zu einer Länge von 64 Byte übertragen werden. Es wurde ein einfaches Rahmenprotokoll implementiert, welches die Übertragung von Befehlen, Parameterwerten und gemessenen Daten vereinheitlicht. Hierzu verwendet es Start- und Stopp-Byte und ein Byte mit der Länge des Datenpakets. Pakete mit einer Länge größer als 64 Byte werden von der geschriebenen Software automatisch auf mehrere Pakete verteilt. Die Übertragung funktioniert so zuverlässig, dass keine besonderen Fehlerkorrekturmechanismen benötigt werden.

3.2 PC

Der Screenshot 9 zeigt die Benutzeroberfläche der entwickelten PC-Software. Das untere Fenster der PC-Software bietet dem Benutzer die Möglichkeit, Steuerbefehle an den Motortreiber zu senden und Systemparameter zu übertragen. Der zu ändernde Parameter wird in der Liste ausgewählt und durch einen Schieberegler oder mithilfe des Textfeldes geändert. Wichtige Zustandsinformationen (Regelmodus, Motorspannung, Strom) werden in Echtzeit angezeigt.

Im oberen Fenster können verschiedene Systemvariablen des Mikrocontrollers in Abhängigkeit von der Zeit dargestellt werden. Dies kann in zwei Modi geschehen. Im Echtzeit-Modus können mehrere Variablen gleichzeitig in Echtzeit dargestellt werden. Die Software fordert dazu pro Variable einen Messwert vom Motortreiber an. Nach Erhalt des Ergebnisses wird das Diagramm aktualisiert und die nächsten Werte angefordert. Die Samplingfrequenz liegt hier im Bereich von 100 Hz, wodurch die Visualisierung dem Benutzer flüssig erscheint. Sie ist maßgeblich durch die USB-Latenz bestimmt, denn die eigentliche Mess- und Übertragungsgeschwindigkeit ist weitaus höher. Dieser Modus erlaubt es beispielsweise, das Regelverhalten des Treibers bei sich ändernder Motorbelastung zu überprüfen. Die Entwicklung der Regelalgorithmen wird durch diese Funktion stark erleichtert. Im zweiten Modus, dem Buffer-Modus, fordert die Software die Aufnahme mehrerer Messpunkte einer Variablen an. In diesem Modus speichert der Mikrocontroller die Messergebnisse im internen Datenspeicher zwischen und überträgt diese erst nach Aufnahme von 512 Samples.

Dadurch kann die Samplingrate bis auf 1 kHz erhöht werden, was beispielsweise eine zeitauflösende Untersuchung des Motorstroms ermöglicht. Die aufgezeichneten Messdaten können zur weiteren Auswertung im universellen *.csv abgespeichert werden.

Mithilfe der Software können fast alle Variablen des Motortreibers als Gleitkommawert (float, 4 byte) gelesen bzw. gesetzt werden. Die entsprechenden Variablen sind dazu in einem File zusammengefasst, dass sowohl beim Kompilieren der Mikrocontroller- als auch PC-Software eingebunden wird. Die Mikrocontroller-Software wurde so programmiert, dass ein Sampling-Prozess die Regelabläufe nicht beeinflusst. Der Zugriff auf die Variablen erfolgt dabei als direkter Zugriff auf den betroffenen Speicherbereich.

Die PC-Software wurde mit einer Linux-Distribution unter C++ mit dem freien, plattformunabhängigen Qt5-Framework in der kostenlosen IDE NetBeans entwickelt. Das Framework bietet viele Funktionen, die das Entwickeln von graphischen Benutzeroberflächen vereinfachen, wie z.B. Threading-Funktionen, ein Signal-Slot-System oder Fenster-Objekte.

Das Programm wurde so entwickelt, dass der Motortreiber als Library in andere Software-Projekte eingebunden werden kann. Das Programm ist sowohl auf Linux- als auch Windows-Systemen lauffähig.

3.3 Mikrocontroller

Programmierung

Für die Entwicklung der Mikrocontroller-Software wird die *MPLAB X IDE*, zusammen mit dem *XC32-Compiler* des Chip-Herstellers verwendet. Die Entwicklungsumgebung ermöglicht die Kompilierung der Chip-Firmware aus C++-Code und die anschließende Übertragung auf den Chip mithilfe eines USB-Programmierinterfaces. Dieses Interface bietet auch Debugging-Funktionalitäten wie das Setzen von Breakpoints oder das direkte Auslesen des Datenspeichers.

Im Unterschied zu der Entwicklung von PC-Software kann über Register direkt auf Hardware-Funktionen des Mikrocontrollers, wie beispielsweise AD-Wandler, UART oder Timer zugegriffen werden. Die Register sind im Datenblatt des Mikrochips dokumentiert. Der Chip-Hersteller liefert hardware-optimierte DSP-Bibliotheken, welche floating-point-Rechenoperationen zeitoptimiert implementieren.

Für die Untersuchung von Code-Laufzeiten und der Häufigkeit von bestimmten Aufrufen hat sich das Umschalten eines Ports bei Routinen Ein- und Austritt als einfaches Mittel erwiesen. Mit dem Oszilloskop können die Zeiten

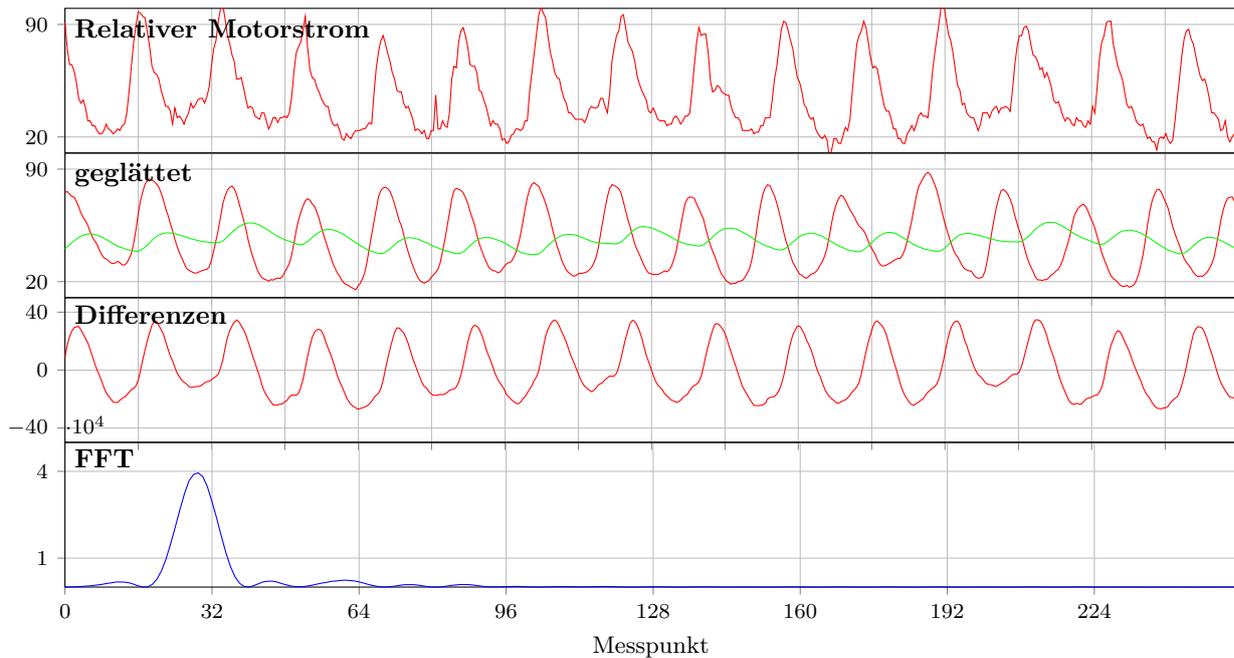


Abbildung 11: Datenaufbereitung Stromsignal (512 ms): 1. ADC-Messwerte (relativer Strom); 2. geglättet; 3. Differenzen der Glättungswerte; 4. FFT der Differenzen. Die Werte wurden auf der Seite des Treibers berechnet und mit der PC-Software im Sweep-Modus ausgelesen.

über die Pulsabstände am Digitalausgang genau ausgemessen werden.

Der Mikrocontroller wird etwas über dem spezifizierten Bereich mit einer Taktfrequenz von 48 MHz betrieben.

Interruptsteuerung

Der Mikrocontroller wird nicht mit einem dedizierten Betriebssystem betrieben, welches sich um die Verteilung der Rechen- und Speicherkapazitäten des Mikrocontrollers kümmern würde. Aus diesem Grund kann und muss von vornherein bestimmt werden, welche Routinen zu welchem Zeitpunkt und bei welchen Bedingungen ausgeführt werden.

Im Normalzustand befindet sich der Mikrocontroller in der Hauptprogrammschleife, in welcher bei Bedarf

rechen- und somit zeitintensive Funktionen ausgeführt werden. Um trotzdem zeitnah auf Ereignisse reagieren zu können, ermöglicht der Mikrocontroller das Setzen von Interrupts. Tritt eine dem Interrupt zugeordnete Bedingung ein, so findet ein direkter Sprung in den zugehörigen Interrupt-Code statt, in dem das Ereignis verarbeitet werden kann. Danach wird das Programm an der Aussprungstelle fortgeführt. Es gibt sieben Interrupt-Prioritäten, wobei ein Interrupt mit höherer Priorität einen mit niedrigerer unterbrechen kann. Allgemein sollten zeitkritische Interrupts eine höhere Priorität haben. Die Ausführdauer der Interruptroutine sollte deutlich geringer sein als die eigene Ausführperiode. Funktionen in der Hauptprogrammschleife können dort aus einem Interrupt durch Setzen von Flags ausgelöst werden.

4 Regelungssystem

4.1 Drehzahlbestimmung

Messverfahren

Die wichtigste Aufgabe des Motortreibers ist die Bestimmung der Drehzahl aus dem Verlauf des Motorstroms. Hierfür hat sich nach der Erprobung verschiedener Verfahren im gesamten Drehzahlbereich unter verschiedenen Regelbedingungen ein Verfahren als sehr zuverlässig erwiesen: Die Bestimmung der Drehzahl durch eine Frequenzanalyse des Motorstroms. Die Kommutatorwechsel

bei drehendem Motor bewirken eine periodische Schwingung im Stromverlauf. Die Frequenz dieser Schwingung entspricht der der Kommutatorwechsel und ist somit proportional zur Drehzahl.

Mithilfe der diskreten Fourier-Transformation wird das zeitbasierte Signal in ein frequenzbasiertes Spektrum transformiert. Dabei ist das empirisch gegebene Signal $f(x)$ durch die Fourier'sche Summe

$$g(x) = a_0/2 + \sum_{k=1}^N (a_k \cdot \cos(kx) + b_k \cdot \sin(kx)) \quad [6] \quad (\text{S.1008})$$

zu approximieren, sodass $g(x) \approx f(x)$. Die Koeffizienten a_0, a_k, b_k sollen dabei reell sein. N entspricht der Anzahl an Messpunkten. Es gilt nun diese Ansatzkoeffizienten zu bestimmen. Jeder Koeffizient a_k bzw. b_k gibt dann die Amplitude der zugehörigen Sinus- bzw. Cosinusschwingung der Kreisfrequenz k an, welche addiert die Ursprungsfunktion $f(x)$ annähern. Mithilfe der komplexen Beschreibung der $\sin(x)$ - und $\cos(x)$ - Funktion kann der Ausdruck in eine komplexe Form umgewandelt werden:

$$g(x) = \sum_{k=1}^N c_k \cdot e^{-ikx} \text{ mit } c_k = \frac{a_k - ib_k}{2}$$

Mithilfe der *schnellen Fourier-Transformation* (FFT) können für diese Gleichung dann die komplexen Koeffizienten c_k bestimmt werden. Dies ist ein schneller Algorithmus, der sich für den Spezialfall $N = 2^n$ anwenden lässt und relativ einfach programmiert werden kann.

Durch die Betragsbildung der komplexen Koeffizienten lässt sich ein Frequenzspektrum berechnen. Die Phaseninformation wird somit verworfen. Die dem Koeffizienten c_N zugehörige Frequenz entspricht dabei der Abtastfrequenz f_{ab} . Also gilt für die Frequenz eines Koeffizienten c_k :

$$f_k = \frac{k}{N} \cdot f_{ab} \quad (4.1)$$

Und für das Frequenzspektrum $a(x)$ somit:

$$a(x) = |c_x| \cdot f_x = \sqrt{\operatorname{Re}(c_x)^2 + \operatorname{Im}(c_x)^2} \cdot \frac{x}{N} \cdot f_{ab}$$

Da die Eingangswerte der FFT rein reell sind, muss die obere Hälfte des Spektrums verworfen werden. Die betroffenen Frequenzen sind Spiegelungen der unteren Hälfte, da diese aufgrund des Nyquist'schen Abtasttheorems nicht erfasst werden können. Das Theorem besagt sinngemäß, dass eine Frequenz nur dann ohne Artefakte erfasst werden kann, wenn mindestens mit der doppelten Frequenz abgetastet wird.

Niederfrequente Anteile, die womöglich einen höheren Betrag als die Frequenz der Drehzahl haben, sollen vernachlässigt werden. Nun kann die Drehzahl D bestimmt werden, indem die Stelle des Maximums von $a(x)$ ermittelt wird:

$$D = a^{-1}[\max(a(x))] \text{ mit } x = D_{\min} \dots k/2$$

Die Phasenlage des Drehzahlsignals entspricht dem Winkel der konjugiert Komplexen des der Drehzahl entsprechenden Koeffizienten:

$$\varphi = \arg\{c_D^*\} \quad (4.2)$$

Wird die FFT direkt auf den gemessenen Motorstrom angewendet, ist das resultierende Spektrum sehr unsauber, was vor allem an einem Gleichanteil und Anteilen höherer Frequenzen liegt, die sich als Artefakt im Spektrum abbilden. In der Praxis hat sich eine vorherige, digitale Glättung des Signals bewährt. Der geglättete Wert g wird dabei aus dem neuen Messwert m folgendermaßen berechnet:

$$g_{n+1} = (m - g_n) \cdot a$$

Je größer die Differenz zwischen Messwert und geglättetem Wert, desto stärker die Auswirkung auf den neuen Glättungswert. Je kleiner der Parameter a , desto „glatter“ der zeitliche Verlauf des geglätteten Wertes. Im Gegensatz zu einem gleitenden Durchschnitt hat dieses Verfahren den Vorteil, dass keine vergangenen Werte gespeichert werden müssen. Aktuellere Werte sind nach diesem Verfahren allerdings stets stärker gewichtet, als weiter zurück liegende.

Der relative Motorstrom wird mit zwei unterschiedlichen Glättungsfaktoren geglättet und die Differenz der sich ergebenden Signale gebildet, wodurch der Gleichanteil des Signals wegfällt. Auf die Differenzen wird dann die FFT angewendet. Die einzelnen Zwischenschritte des Verfahrens sind in Abbildung 11 veranschaulicht.

Messgenauigkeit

Die Frequenzauflösung der FFT wird umso größer, je höher die Anzahl N an Messpunkten bei gleichbleibender Abtastfrequenz ist (siehe Gleichung (4.1)). Mit der steigenden Anzahl an Messpunkten steigt jedoch auch die benötigte Gesamtmesszeit, was dann wiederum zu einer geringeren Zeitauflösung führt. Dieser Zusammenhang kann als eine Unschärfe der diskreten Fourier-Analyse betrachtet werden. Eine hohe Zeitauflösung ist für eine schnelle Regelung essentiell. Mit einem Trick kann man die Frequenzauflösung erhöhen, ohne die Zeitauflösung zu beeinträchtigen: Man erhöht zwar die Anzahl an Messpunkten die mit der FFT transformiert werden, allerdings ist nur eine kleinere Anzahl an Messpunkten mit tatsächlichen Messdaten gefüllt, der Rest enthält nur Nullen (Zero-Padding). Die Ungenauigkeit zeigt sich dann in der Peakbreite, wie man in Abbildung 12 erkennen kann.

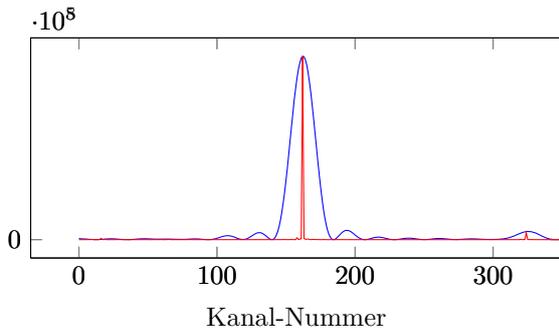


Abbildung 12: FFT ($N=2048$) des gleichen Stromverlaufs mit 90 (blau) und 2048 (rot) echten Samples. Die Frequenzauflösung ist bei 2048 Messpunkten wesentlich höher.

Die Genauigkeit ΔD der Drehzahlbestimmung mit diesem Verfahren ergibt sich aus der FFT-Kanalbreite Δf . D_0 gibt das Verhältnis zwischen dem mit der FFT-Verfahren errechneten relativen Frequenz k/N und der Drehzahl an. Dieses ist neben der Abtastrate abhängig von der Lamellenanzahl des Kommutators und der Getriebeuntersetzung und kann am besten empirisch bestimmt werden.

$$D(k) = D_0 \cdot \frac{k}{N}$$

$$\Delta D(k) = D(k) - D(k-1) = \frac{D_0}{N}$$

Zuverlässigkeit

Die Diagramme in Abbildung 13 zeigen die nach dem beschriebenen Verfahren bestimmte Drehzahl. Die Motorleistung wird dabei nach einem vorgegebenen Muster eingestellt. Bis 4,5 s werden kleine Stufen, weiter bis 6 s große Stufen, dann eine Rampe und ab 8 s eine konstante Leistung eingestellt. Mithilfe dieses Musters kann die Arbeitsfähigkeit der Drehzahlbestimmung bei verschiedenen Drehzahl-Änderungsraten und in verschiede-

nen Drehzahl-Bereichen eingeschätzt werden. Der Motor ist hierbei unbelastet, sodass die Drehzahl annähernd linear mit der Ankerspannung steigt.

Aus den Diagrammen kann man folgendes erkennen:

- Die Drehzahl wird bei einer Spannung unterhalb von 30% nicht zuverlässig erkannt (Sekunde 0-0,6 und 6-6,5).
- An den Sprungstellen der Stufen kommt es zu kurzen Einbrüchen der Drehzahl. Dies kann dadurch erklärt werden, dass es bei den vorgegebenen Spannungsänderungen zu schnellen Stromänderungen kommt, wie auch im oberen Diagramm zu erkennen ist. Diese Peaks weisen dabei Frequenzanteile auf, die im Bereich der Kommutator-schwingungen liegen. Sind die zugehörigen Amplituden höher, wird das falsche Maximum im Spektrum erkannt.
- Bei langsamen Änderungen, sowie bei einer konstanten Drehzahl, wird diese zuverlässig gemessen. Dabei kommt es nicht zu Abweichungen vom Mittelwert, die über einen FFT-Kanal hinaus gehen.
- Der bei der Drehzahl erkennbare Stufenverlauf hat eine Stufenbreite entsprechend der Samplingdauer für eine FFT-Berechnung.

Wie sich zeigte, liefert das Verfahren zur Drehzahlbestimmung bei niedrigen Drehzahlen kein sinnvolles Ergebnis. Dies liegt vorallem daran, dass der Motorstrom sehr gering ist und sich das Quantisierungsrauschen des ADCs stärker zeigt, als die Schwingungen bedingt durch den Kommutator. Dazu nehmen diese Schwingungen eine Form an, die zunehmend von einer Sinusschwingung abweicht.

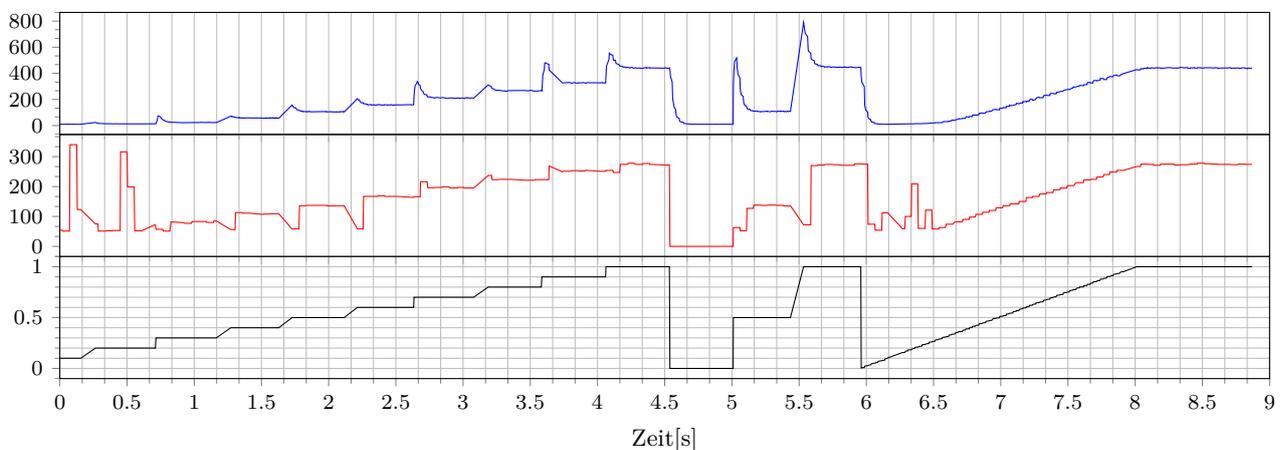


Abbildung 13: Geglätteter Motorstrom (blau, ADC-Wert), berechnete Drehzahl (rot, proportional zur FFT-Frequenz) bei vorgegebener Motorspannung (schwarz, 0-100%). Die Messwerte wurden mit der PC-Software im Echtzeit-Modus erfasst.

4.2 Positionsbestimmung

Integration der Drehzahl

Für den Betrieb als Servo muss der absolute Drehwinkel (im Folgenden Position) des Motors bekannt sein. Ohne die Verwendung eines externen Positionsgebers kann die aktuelle Position P aus der Drehzahl D gewonnen werden. Dazu wird diese nach der Zeit integriert.

$$P(t) = \int_0^t D(\xi) d\xi + P_0$$

Dieser Ausdruck kann aufgrund der stichprobenartigen Erfassung der Drehzahl nur durch Teilsummen angenähert werden:

$$P_n = \sum_{i=1}^n (D_i \cdot \Delta t) + P_0$$

Die Genauigkeit der momentanen Position hängt dabei von der Genauigkeit jedes in die Summe eingegangenen Wertes ab. Die momentane Position wird bei sich ändernder Drehzahl um so genauer bestimmt, je kleiner das Zeitintervall gewählt wird und je mehr Messpunkte so in die Summe einfließen. Es müsste dafür allerdings die Messzeit für die Samples die in eine FFT einfließen reduziert werden, was dann zu einer ungenaueren Drehzahl führen und sich schließlich negativ auf die Genauigkeit auswirken würde.

Unvermeidbar ist der Einfluss des Quantisierungsfehlers bei der Drehzahlbestimmung, wo es im schlimmsten Fall zu einem Fehler von einer Kanalbreite (ΔD) kommen kann. Der maximal zu erwartende relative Positionsfehler F ist abhängig von der Samplingfrequenz f_s und der Anzahl an FFT-Samples N :

$$F = \frac{\Delta D}{D_i} = \frac{1}{N} \cdot \frac{f_s}{D_i} \text{ mit } \Delta D = \frac{F_s}{N}$$

Wird die Samplingfrequenz an die maximal zu erwartende Drehzahl angepasst (unter Einhaltung des Abtasttheorems), ergibt sich für $N = 512$ der relative Fehler bei der halben maximalen Drehzahl zu:

$$F = \frac{4}{512} \approx 0,78\% \text{ mit } f_s = 2 \cdot D_{max} \text{ und } D_i = \frac{1}{2} \cdot D_{max}$$

Bei noch niedrigeren Drehzahlen wird der Fehler weitaus Größer, weshalb deswegen ein anderes Verfahren Anwendung findet.

Kontinuierliches Sampling

Die beschriebenen Fehler konnten reduziert werden, indem die Position über das direkte Auszählen der einzelnen

Stromperioden in einem Zeitintervall ermittelt wird. Der Strom muss dazu allerdings lückenlos erfasst werden, weshalb zwei Messwertspeicher benötigt werden. Während der eine Buffer mit neuen Messdaten gefüllt wird, wird der andere ausgewertet. Neben der Frequenz f wird bei der Auswertung aus der FFT auch die Phasenlage φ des Kommutatorsignals $s(t)$ erfasst, sodass die Anzahl an steigenden, positiven Halbwellen der Schwingungsperioden im Messintervall $[0, b)$ ermittelt werden kann.

$$s(t) = s_0 \cdot \sin(2\pi f \cdot t + \varphi)$$

$$2\pi \cdot n = 2\pi f \cdot t + \varphi \text{ mit } 0 \leq t < b$$

Das Verfahren wird umso resistenter gegen kurzzeitige Störungen im Motorstrom, je mehr Perioden im Messintervall liegen. Die Schwingungsinformation werden dann durch die korrekt gemessene Perioden geliefert. Eine hohe Periodenanzahl führt allerdings auch zu einer geringeren Toleranz bezüglich Drehzahländerungen, die sich innerhalb eines Messintervalls bemerkbar machen. Es kann dann zu zuviel oder nicht gezählten Perioden kommen.

In der Praxis hat sich eine Periodenanzahl von 8 bei 512 FFT-Eingangswerten mit 100 echten Samples (Zero-Padding) bewährt. Die Samplingrate wird dabei in Abhängigkeit von der aktuellen Drehzahl automatisch so angepasst, dass die Anzahl an Perioden ungefähr konstant ist und die Rate an Drehzahlinformation somit mit der Drehzahl steigt.

Durch geringe Fehler bei der Bestimmung der Phasenlage des Schwingungsausschnitts durch die FFT könnte es dazu kommen, dass eine Schwingungsperiode doppelt bzw. garnicht gezählt wird, wenn diese in der Nähe der Nahtstelle zwischen zwei Messintervallen liegt. Um dies zu verhindern wird bei der Berechnung der Schwingungsanzahl zusätzlich überprüft, ob innerhalb der letzten halben Periodendauer vor der ersten gezählten Periode im aktuellen Intervall eine Periode im vorherigen Intervall gezählt wurde. In diesem Fall wird diese erste Periode nicht gezählt, da sie eben höchstwahrscheinlich schon vom vorherigen Intervall erfasst wurde. Dagegen wird eine Periode zusätzlich gezählt, wenn 1 1/2 Periodendauer vor der ersten Periode keine Periode im vorherigen Intervall gezählt wurde.

Bei der Berechnung der Periodenanzahl werden darüber hinaus nur die Perioden gezählt, zu deren Zeitpunkt der absolute Motorstrom einen einstellbaren Grenzwert überschreitet. Auf diese Weise wird ein falsches Zählen während des Anlauf- und Stoppvorgangs reduziert, bei dem sich die Drehzahl noch innerhalb eines Messinter-

valls stark ändert. Zusätzlich wird zur Berechnung die Drehzahl-Information des nachfolgenden bzw. vorhergehenden Messintervalls verwendet, da die betroffenen Messintervalle nur zum Teil Schwingungen enthalten und die entsprechende Frequenzanalyse damit stärker fehlerbehaftet ist.

Zeitliche Kenndaten

Das beschriebene Verfahren zur Drehzahl- und Positionsbestimmung wurde zeitoptimiert im Mikrocontroller implementiert. Die Auswertung eines Messintervalls dauert dabei 4 ms. Die Samplingfrequenz beträgt bis zu 5 kHz. Die Dauer eines Samplingvorgangs, bei dem der Motorstrom aufbereitet und gespeichert wird beträgt 30 μ s. Die maximale Rate an Drehzahldaten beträgt bei 100 Strommesspunkten pro Messintervall 50 Hz.

Externe Sensoren

Bei vielen Anwendungen muss der Servodrehwinkel in Bezug zum Systemwinkel gesetzt werden (zum Beispiel Mittelstellung bei einer Lenkung, Start- und Endposition bei einer Scheibenwischanlage, etc.), sodass ein externer Sensor eingesetzt werden muss. Ein solcher Sensor muss nicht über eine Winkelaufösung verfügen, es reicht ein Taster oder eine Lichtschranke. Wird dieser dazu beispielsweise an einem Zahnrad angebracht wird ein regelmäßiges Signal generiert, was eine Positionsinformation mit sehr hoher absoluter Genauigkeit liefert. Der Motortreiber verfügt über einen Digitaleingang und kann solche Signale externer Sensoren verarbeiten. Mit einem Parameter muss dabei die Positionsänderung pro Sensorauslösung angegeben werden. Bei jeder Auslösung wird dann die Position korrigiert.

4.3 Regelungsalgorithmus

Drehzahlregelung

Die Aufgabe des Regelkreises ist es, die Regelgröße Drehzahl unabhängig von äußeren Störgrößen (mechanische Belastung, Spannungsänderungen) innerhalb eines gewissen Toleranzbereiches zu halten. Die Regelgröße kann durch die Ankerspannung beeinflusst und durch die Drehzahlbestimmung gemessen werden. Der Zusammenhang zwischen eingestellter Ankerspannung und der resultierenden Drehzahl ist durch das technische System vorgegeben und durch Verzögerungen und Nicht-Linearität gekennzeichnet. Der Regler soll diesen Zusammenhang möglichst passend beschreiben und den Motor-

strom so regeln, dass die Regelgröße die Soll-Drehzahl (Führungsgröße) möglichst schnell und stabil erreicht. Als Regler kommt ein PI-Regler zur Anwendung, welcher als Eingangsgröße die Regeldifferenz $r(t)$, also die Differenz aus Soll- und Ist-Drehzahl, entgegen nimmt. Die Stellgröße Motorspannung $y(t)$ wird dabei aus einem Proportionalanteil $P(t)$ und einem Integralanteil $I(t)$ berechnet:

$$y(t) = P(t) + I(t)$$

Der Proportionalanteil (Proportionalglied) hat bei einer Änderung der Regeldifferenz einen sofortigen Einfluss auf die Stellgröße, da dessen Ausgangswert das K_P -fache des Eingangswertes annimmt.

$$P(t) = K_P \cdot r(t)$$

Der Integralanteil (Integrierglied) ist proportional zum Integral der Eingangsgröße. Der Einfluss auf die Stellgröße kann über den Parameter K_I eingestellt werden.

$$I(t) = K_I \cdot \int_0^t h(\xi) d\xi [7]$$

Nur mit dem Integralteil könnte bereits eine Regelung realisiert werden: Ist die Ist-Drehzahl null und die Soll-Drehzahl wird auf einen Wert eingestellt, so beginnt der Wert des Integrals und somit auch die Stellgröße zu steigen. Ändert sich die Ist-Drehzahl nicht, so würde sich ein Rampenverlauf zeigen. Nach einer gewisse Zeit beginnt jedoch die Ist-Drehzahl zu steigen, die Regeldifferenz wird immer kleiner und das Integral steigt immer langsamer. Nach sehr langer Zeit ist die Regeldifferenz dann null und die Soll-Drehzahl ist erreicht. Je größer der Integralparameter, desto schneller läuft dieser Prozess ab. Ist dieser jedoch zu hoch kommt es zu einem Überschwingen, „da die Antwort Drehzahl-Antwort des Motors zu spät kommt“. Der Integralparameter sollte so eingestellt werden, dass es gerade nicht zu einem Überschwingen kommt. Die geforderte Reaktionsgeschwindigkeit kann dann über den Proportionalanteil eingestellt werden.

Für den Integralanteil wird der Regelfehler jeweils addiert, der sich so ergebene Wert nähert das Integral an. Dieser ist nach oben und unten so beschränkt, dass es nicht zu einer Übersteuerung der Stellgröße kommen kann.

Die maximale Regelgeschwindigkeit ist begrenzt durch die Dauer der Drehzahlbestimmung.

Regelparameter

Allgemein sollten die Regelparameter wie oben beschrieben so eingestellt werden, dass der Sollwert möglichst schnell und stabil erreicht wird. Bei diesem Motortreiber hat die Stellgröße allerdings einen direkten Einfluss auf die Messung der Drehzahl: Ändert sich der Strom zu schnell, liefert die Drehzahlbestimmung keine sinnvollen Werte. Deshalb wurden die Regelparameter bewusst so gewählt, dass die Regelung den Strom langsam steigen lässt. Aus diesem Grund müssen die Parameter nicht besonders optimiert werden und von einer genaueren mathematischen Beschreibung des Regelmodells wird deshalb abgesehen.

Anlaufregelung

Wie sich in Kapitel 4.1 zeigte, liefert die Drehzahlbestimmung unterhalb einer gewissen Drehzahl D_{on} sporadische Werte. Würde man die Regelung bei Stillstand aktivieren, müsste diese den betroffenen Drehzahlbereich durchschreiten. Aufgrund der falschen Regeldifferenzen würde es zu einer chaotischen Reaktion kommen, wodurch die Drehzahl dann gar nicht mehr bestimmt werden kann.

Um diesen Fall zu vermeiden, muss der betroffene Drehzahlbereich von der Regelung „übersprungen“ werden. Zur Minimierung der Zeit im kritischen Bereich, wird die Motorspannung beim Anlaufvorgang direkt auf die der kritischen Drehzahl ungefähr entsprechenden gesetzt. Parallel dazu wird der Betrag der Motorstromänderung untersucht. Unterschreitet dieser einen gewissen Wert, ab dem die Drehzahlbestimmung verlässlich funktioniert, wird der Regelkreis aktiviert.

Zusätzliche Mechanismen

Üblicherweise verfügen Servoregelungen über Mechanismen zur Drehmomentbegrenzung. Da sich das Drehmoment bei dem betrachteten Motortyp proportional zum Motorstrom verhält, kann diese durch eine Strombegrenzung realisiert werden. Dazu wird bei Ausführung der Regelroutine zusätzlich der Betrag des gemessenen Motorstroms überprüft. Überschreitet dieser den des Drehmomentes entsprechenden Strom, wird die obere Begrenzung der Regeldifferenzsumme des Integriergliedes reduziert. Fällt der Strom zurück, wird die Begrenzung wieder auf den ursprünglichen Wert zurückgesetzt.

Kommt es zu einem besonders schnellen Anstieg des Stroms, wie es beispielsweise der Fall wäre, wenn ein Fremdkörper in die Antriebsmechanik gelangt, so bewirkt eine zusätzliche Funktion die automatische, sofortige Abschaltung der Motorspannung und ein Zurückfahren des Motors.

4.4 Positionierung

Positionierungsfahrt

Mit einem Servo müssen oft größere Massen beschleunigt werden. Bei einer sprunghaften Beschleunigung käme es aufgrund der Trägheitskraft zu ungünstig hohen Spitzenkräften in der Mechanik. Um dies zu vermeiden, muss die Beschleunigung bei der Einstellung einer neuen Soll-Position beschränkt werden. Hierfür wird vor der Fahrt eine Funktion bestimmt, die die Soll-Drehzahl in Abhängigkeit von der aktuellen Position vorgibt. Die Positionsregelung ist somit eine Regelung, die auf der Drehzahlregelung aufbaut.

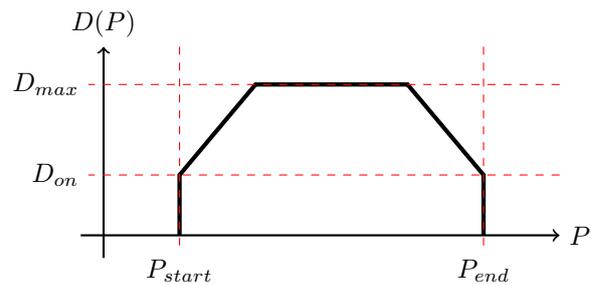


Abbildung 14: Diagramm des Drehzahlverlaufs bei anfahren einer neuen Soll-Position

Das Diagramm 14 zeigt die implementierte Drehzahlrampe. Zu Beginn des Positionierungsprozesses entspricht der eingestellte Drehwinkel der Position P_{start} . Es wird die kritische Drehzahl D_{on} eingestellt und der Motortreiber startet mit einem Anlaufvorgang. Mit zunehmender Position steigt die vorgegebene Drehzahl linear, bis sie die Maximaldrehzahl D_{max} erreicht. Bevor die Endposition P_{end} erreicht wird, sinkt die Drehzahl wieder linear, sodass bei Erreichen der Endposition erneut D_{on} eingestellt ist. An diesem Punkt wird die Motorleistung direkt auf Null gesetzt. Die Steigung der Rampen entspricht dabei der eingestellten Maximalbeschleunigung. Auch D_{max} kann als Parameter eingestellt werden.

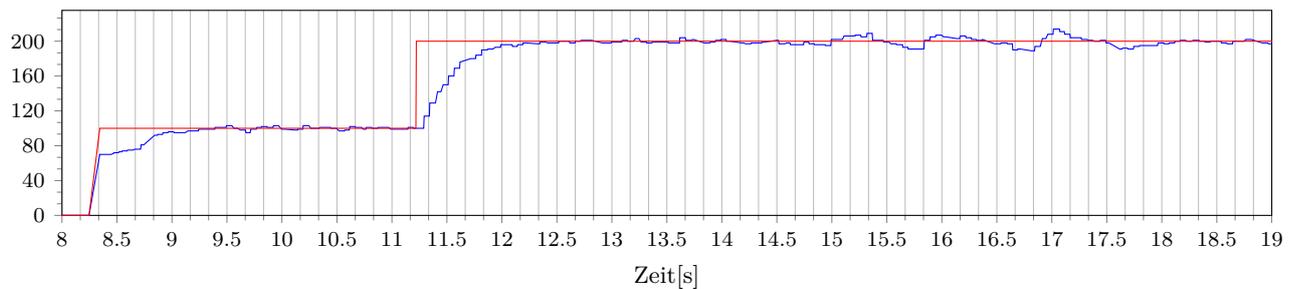


Abbildung 15: Sprungantwort der Drehzahlregelung. Gemessene Drehzahl (blau) bei Änderung der Soll-Drehzahl (rot). Y-Achse: Einheit proportional zur Drehzahl.

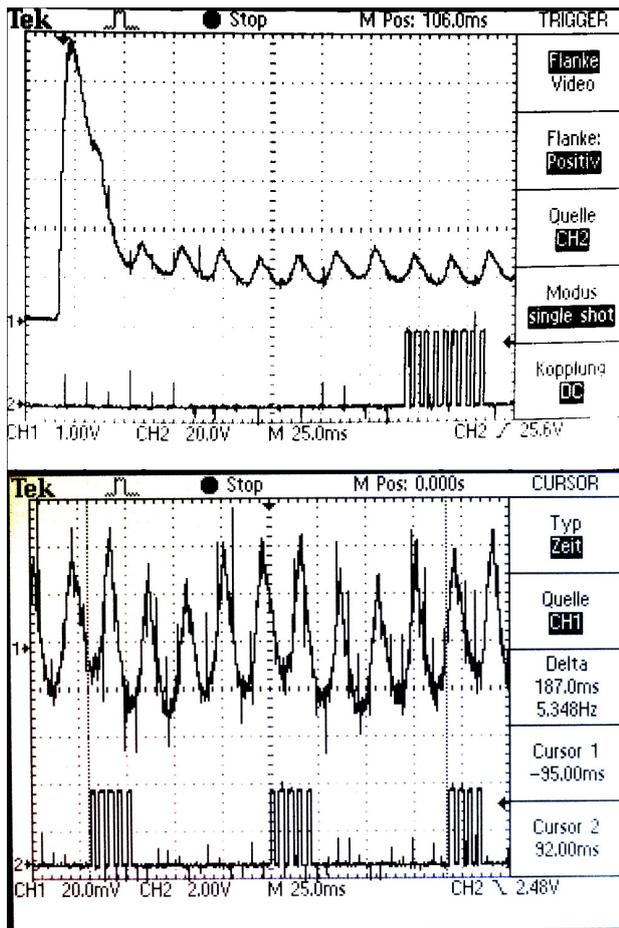


Abbildung 16: Zählverhalten des Motortreibers beim Motoranlauf und im laufenden Betrieb. Jeweils 1. Graph: Analog gefilterter Motorstrom, 2. Graph: Pulssignal des Mikrocontrollers.

Positioniergenauigkeit in der Praxis

Der Fehler bei der Bestimmung der Position des Motors wurde mithilfe eines externen Sensors, der jede Umdrehung der Servoachse exakt bestimmt, ausgemessen. Dazu wurde bei einem Sensorsignal die berechnete Positionsänderung zum vorherigen Signal gespeichert. Die Messung wurde mit 300 Umdrehungen bei unterschiedlichen Drehzahlen durchgeführt. Oberhalb der kritischen Drehzahl liegt die Standardabweichung bei 4 Schwingungs-

perioden vom Mittelwert von 620 Schwingungsperioden pro Umdrehung - unabhängig von der Drehzahl. Der gemessene Fehler ist damit zu begründen, dass die Positionsänderung durch die Einteilung des Messsignals in Zeitintervalle nicht auf die Schwingungsperiode genau, sondern auf ein ganzzahliges Vielfaches bestimmt wird. Deshalb gilt dieser Fehler nur für die Genauigkeit einer Positionsbestimmung an sich und wird nicht akkumuliert, da die „echte“ Position bekannt ist. Daher ist auch nach einer höheren Zahl an Umdrehungen der gleiche absolute Fehler zu erwarten.

Während des Betriebs des Motors konnten keine Fehler beim Auszählen der Schwingungsperioden festgestellt werden. Lediglich beim Anlauf und beim Stopp des Motors kommt es aufgrund des starken Stromgradienten zu Fehlern bei der Schwingungs-Erfassung die sich bei mehreren Positionierungsfahrten akkumulieren können.

Die Screenshots 16 zeigen das Zählverhalten des Motortreibers. Über einen digitalen Ausgang des Mikrocontrollers wird die Anzahl der im letzten Messintervall gezählten Schwingungsperioden über kurze Pulse angezeigt. Die jeweils ersten Pulse zeigen die Grenzen des Intervalls an. Im laufenden Betrieb kommt es dabei nicht zu Zählfehlern.

4.5 Regelverhalten

Abbildung 15 zeigt die gemessene Drehzahl des Motors bei einer sprunghaften Änderung der Soll-Drehzahl. Bei 8,3 s führt der Treiber einen Anlaufvorgang aus, wie sich an dem flachen Verlauf der Drehzahl von 8,3 s bis 8,8 s erkennen lässt. Nach ca. 0,6 s ist die Soll-Drehzahl (100) erreicht. Ab 11,2 s wird die Soll-Drehzahl sprunghaft erhöht, welche nach ca. 1 s erreicht wird. Danach schwankt die Ist-Drehzahl mit einer Abweichung von höchstens 2% um den Soll-Wert. Ab 14,5 s wird der Motor wechselnd stark belastet, wodurch die Ist-Drehzahl kurz sinkt, bis eine Nachregelung stattfindet. Bei Sinken der Belastung kommt es dann zu einem Anstieg der Drehzahl. Eine höhere Regelgeschwindigkeit würde zu weniger stark aus-

geprägten Regelfehlern führen, was allerdings auch in einem hohen Stromgradient resultiert und die Drehzahlbestimmung stärker beeinflusst.

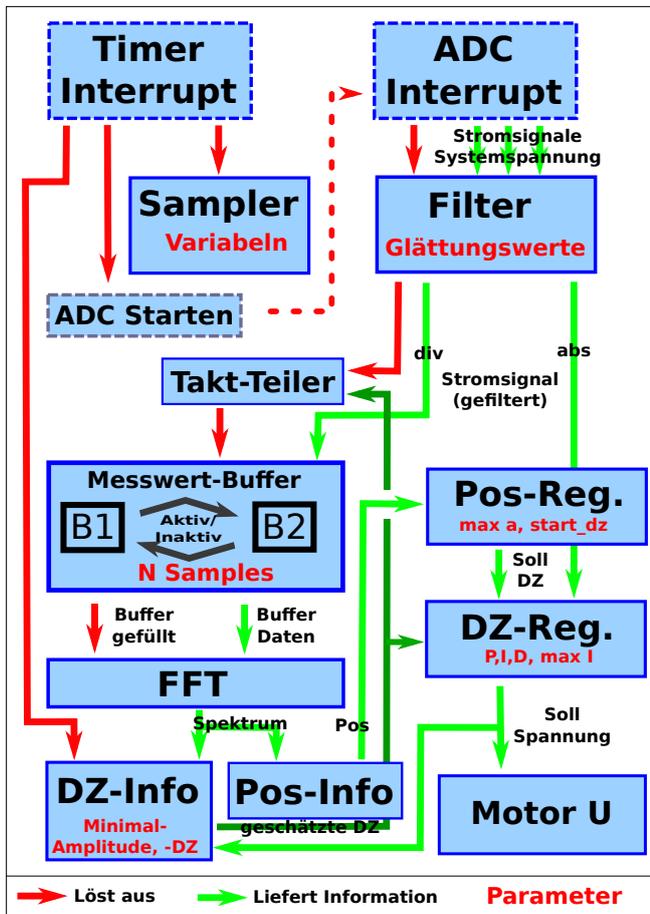


Abbildung 17: Schematische Darstellung der Module des Regelalgorithmus mit einstellbaren Parametern, dem Informationsfluss und der Verarbeitungsreihenfolge.

5 Ausblick

Anwendungsmöglichkeiten

Eine Anwendung des Motortreibers wäre überall da möglich, wo keine besonders hohen Regelgeschwindigkeiten von Nöten sind. Ein Einsatz wäre dann sinnvoll, wenn die Gesamtkosten besonders gering sein sollen. Auch denkbar wäre eine Anwendung als redundantes, einen Drehgeber ergänzendes Messsystem. Mit diesem könnte man beispielsweise auch Schlupf oder einen Defekt in der Antriebsmechanik feststellen, was sich in einer Inkonsistenz der gemessenen Drehzahlen zeigen würde.

Weitere Untersuchungen

Der Prototyp sollte in einigen Bereichen noch weiter untersucht und entwickelt werden. Dies betrifft vor allem die Genauigkeit und Verlässlichkeit der Werte aus der

Drehzahlbestimmung. Ein großes Problem stellt dabei die kritische Drehzahl dar, die nicht unterschritten werden darf. Könnte die Drehzahl bis zum Stillstand gemessen werden, wäre eine kontinuierliche Regelung in diesem Bereich möglich und es käme nicht zu einer Beschleunigungsspitze bei Motorstart. Dadurch wäre zudem die Positionsberechnung deutlich genauer.

Im Moment wird die Drehzahl bestimmt, indem die Stelle des größten Frequenzanteils ermittelt wird, welche durch die Auflösung der FFT beschränkt ist. Der entsprechende Peak im Spektrum könnte stattdessen mit einer geeigneten Funktion angenähert werden, sodass die Stelle des Maximums auch stetig zwischen zwei Kanälen bestimmt werden kann.

Wenn die Genauigkeit des Prinzips ausgereizt ist, sollte das System bezüglich der Regelfehler, Positionsfehler und des Regelverhaltens noch genauer charakterisiert werden. Dazu sollte auch ein Vergleich mit den Messergebnissen eines genauen Drehgebers, wie er bei professionellen Servoantrieben zum Einsatz kommt, durchgeführt werden. Dazu sollte auch die Funktion des Treibers bei der Verwendung anderer Motoren getestet werden.

Die Software könnte um eine Funktion erweitert werden, die es ermöglicht die optimalen Regelparameter automatisch zu bestimmen. Diese müsste die Parameter variieren und aus der Drehzahlantwort die Güte der Konfiguration bestimmen und anpassen.

Das Regelverhalten könnte eventuell durch die Anwendung eines anderen Regelverfahrens verbessert werden. Denkbar wäre eine Regelung basierend auf einem Zustandsschätzer.

Für den universellen Einsatz des Motortreibers wäre eine Erweiterung um einige wichtige Schnittstellen wie zum Beispiel RS232, Profibus oder CAN sinnvoll.

Fazit

Abschließend kann man sagen, dass die Regelung nur anhand des Motorstroms mit dem entwickelten Verfahren überraschend gut und die Positionsregelung sehr genau funktioniert. Vermutlich wird sich das System noch weiter verbessern lassen.

Durch Einsatz des Treibers könnten bestehende Antriebe nachträglich, ohne Eingriff in den Aufbau, mit einer Drehzahlregelung ausgestattet werden, da nur die Elektronik ersetzt werden muss. Durch das sensorlose Prinzip ist das Gesamtantriebssystem sicherer, da ein Ausfall durch Defekte an Sensoren ausgeschlossen werden kann. Gerade mit kleineren Motoren lässt sich so sehr einfach ein sehr günstiger Servo realisieren, wodurch sich ganz neue Anwendungsgebiete ergeben.

6 Literatur und Quellen

- [1] *Datenblatt: International Rectifier IR2109 (PD60163-U)*, September 2004.
- [2] *Application Note: FTDI Chip, AN232B-04 Data Throughput, Latency and Handshaking*, Oktober 2006.
- [3] *Datenblatt: Analog Devices AD8217*, März 2011.
- [4] *Datenblatt: DOGA Motor 111*, Juni 2011.
- [5] *Datenblatt: Microchip PIC32MX1XX/2XX*, Februar 2012.
- [6] Bronstein, Semendjajew, Musiol, and Mülig. *Taschenbuch der Mathematik*. Verlag Europa-Lehrmittel, 2013.
- [7] Lunze. *Regelungstechnik 1*. Springer-Verlag, 2013.
- [8] D. Schröder. *Elektrische Antriebe - Grundlagen*. Springer-Verlag, 2007.
- [9] Ch. Schenk U. Tietze. *Halbleiter-Schaltungstechnik*. Springer-Verlag, 1991.

Alle Abbildungen, Tabellen und Diagramme wurden, sofern in der Beschreibung nicht anders angegeben, selbst angefertigt. Für Grafiken wurde die Software *Inkscape v0.48* und *Gimp v2.8*, für den Textsatz *Latex*, für Diagramme und Plots die Latex-Erweiterung *PGF and TikZ* eingesetzt.

7 Danksagung

Auch wenn mein Jufo-Projekt in diesem Jahr überwiegend Zuhause entstanden ist, bedanke ich mich recht herzlich bei meinem Betreuungslehrer Herrn Thomas Biedermann für die Unterstützung durch Denkanstöße und das zur Verfügung stellen der Werkstatt und der Komponenten. Vielen Dank auch an seine Frau Susanne, die immer für unser leibliches Wohl gesorgt hat. Auch bei den anderen AG-Teilnehmern möchte ich mich bedanken für die tolle Zeit mit Euch. Danke!