



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Patrick Stahl

Simulation und Implementierung einer
asynchronen Point to Point Steuerung

Patrick Stahl

Simulation und Implementierung einer
asynchronen Point to Point Steuerung

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Gustav Vaupel
Zweitgutachter : Prof. Dr. -Ing. Michael Röther

Industrieller Betreuer : Dr. -Ing. Enrico Dressler

Abgegeben am 4. August 2014

Patrick Stahl

Thema der Bachelorthesis

Simulation und Implementierung einer asynchronen Point to Point Steuerung

Stichworte

Ruckbegrenztes Ansteuern eines Synchronmotors, Simulation einer Kaskadenregelung mit Motormodell, Trapez- und Sinoideninterpolation, mathematische Definition des Rucks

Kurzzusammenfassung

Die Arbeit behandelt die Steuerung von BLDC- und PMSM-Motoren mit Schwerpunkt der ruckbegrenzten Beschleunigung während des Positioniervorgangs. Zu Beginn werden die verwendeten Arten an Motoren, Sensoren, Regelungsverfahren und Interpolationen vorgestellt. Darauf folgt die mathematische Gegenüberstellung des Trapezgeschwindigkeits- und des sinoiden Beschleunigungsprofils, die Betrachtung der Simulation mit einer Kaskadenregelung und die Implementierung auf realer Hardware. Anschließend stellt eine ausführliche Evaluierung die Auswirkung des Rucks beider Interpolationsarten anhand verschiedener Szenarien dar. Die Arbeit endet mit einer Zusammenfassung der erzielten Ergebnisse und einem Ausblick auf weiterführende Arbeiten.

Patrick Stahl

Title of the paper

Simulation and implementation of an asynchronous point to point control

Keywords

limited jerk start of a synchronous motor, simulation of cascade control with a motor model, trapezoidal and sinoidal interpolation, mathematic definition of the jerk

Abstract

The thesis deals with the control of BLDC- and PMSM motors with focus on limited jerk acceleration during positioning process. First of all the used forms of motors, sensors, control system processes, and interpolations are presented. Up next, the mathematical comparison of trapezoidal-velocity-profile and sinoidal-acceleration-profile, the consideration of the simulation with a cascade control and the implementation on an actual hardware. After that a detailed evaluation represents the effect of the jerk on both interpolation forms based on different scenarios. Finally the thesis ends with a summary of the attained results and a perspective on further theses.

Vorwort

Zu Beginn möchte ich mich bei meinen betreuenden Professoren Prof. Dr. Gustav Vaupel und Prof. Dr. -Ing. Michael Röther bedanken, die mich während meiner Thesis begleitet haben. Die Umsetzung und die Bearbeitung der Bachelorarbeit wurde mir in der Firma TRINAMIC Motion Control GmbH & Co. KG ermöglicht. Alle nötigen Hilfsmittel standen mir jederzeit zur Verfügung. Während meiner Ausbildungszeit und dem Studium der Elektrotechnik, habe ich ein besonderes Interesse für die Regelungstechnik entwickelt, die nun auch das Thema meiner Bachelorarbeit darstellt. Seit dem Jahr 2012 arbeite ich als Werkstudent bei der Firma TRINAMIC in Hamburg, bei der ich auch mein 20-wöchiges Hauptpraktikum absolviert habe. Die Arbeit macht mir großen Spaß und verstärkt mein Interesse für die Elektrotechnik.

Mein Engagement detaillierteres Wissen über die Techniken der Ingenieurskunst zu erlangen, wurde mit zunehmender Verantwortung belohnt. So konnte ich das erlernte Wissen gut transferieren und praktisch anwenden. Ihren Fokus legt die Firma TRINAMIC auf das Gebiet der „Regelung und Steuerung von kleinen Motoren“, welches auch bei mir großes Interesse für die Regelungstechnik und deren Umsetzung an Motoren geweckt hat. Passend dazu schlug mir mein industrieller Betreuer Dr. -Ing. Enrico Dressler das Thema „Simulation und Implementierung einer asynchronen Point to Point Steuerung“ vor. Bei dem ich mich an dieser Stelle sehr für die Unterstützung bedanken möchte. Nach kurzer Einarbeitung habe ich mich dazu entschlossen, das genannte Thema als Bachelorarbeit auszuarbeiten.

Schließlich möchte ich mich bei meiner Familie für die Unterstützung und den seelischen Beistand bedanken.

Patrick Stahl

Inhaltsverzeichnis

Abkürzungs- und Symbolverzeichnis	III
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VII
1. Einführung	1
1.1. Einleitung	1
1.2. Gliederung und Vorgehensweise	2
1.3. TRINAMIC Motion Control GmbH & Co. KG	2
2. Grundlagen	3
2.1. Motorarten	3
2.1.1. Der bürstenlose Gleichstrommotor	6
2.1.2. Der permanent erregte Synchronmotor	8
2.1.3. Lagemessung des Rotors	9
2.1.4. Motormodell	13
2.2. Kaskadenregelung	15
2.2.1. Reglerauslegung	17
2.2.2. Betragsoptimum	18
2.2.3. Symmetrisches Optimum	18
2.2.4. Zusammenfassung	19
2.3. Asynchrone Point to Point Steuerung	20
2.3.1. Trapezinterpolation	21
2.3.2. Sinoideninterpolation	23
2.3.3. Zusammenfassung	24
2.4. Der Ruck	26
3. Simulation und Implementierung	29
3.1. Simulation	29
3.1.1. Simulationsprogramm	30
3.1.2. Numerische Berechnung der Geschwindigkeitsprofile	30
3.1.3. Kaskadenregelung	31
3.1.4. Darstellung der Simulationsergebnisse	40

3.2. Implementierung	43
3.2.1. Hardware	43
3.2.2. Software	45
3.2.3. Erweiterung der Regelstruktur um das Sinoidenprofil	47
4. Evaluierung	49
4.1. Vorgehensweise der Testreihe	49
4.1.1. Parametereinstellungen	50
4.1.2. Experimentelle Reglerparameterbestimmung	51
4.2. Testablauf	52
4.3. Vergleich der Profile	53
4.3.1. Trapezinterpolation	54
4.3.2. Sinoideninterpolation	55
4.3.3. Zusammenhang von Beschleunigung und Strom	56
4.4. Validierung	57
4.5. Praktische Beispiele	59
5. Zusammenfassung und Ausblick	60
5.1. Zusammenfassung	60
5.2. Ausblick	61
Literaturverzeichnis	63
A. Anhang	i
A.1. Messung der Gegen-EMK	i
A.2. Datenblatt des Simulationsmotors	iii
A.3. Datenblatt des Testmotors	v

Abkürzungs- und Symbolverzeichnis

Abkürzungsverzeichnis

BLDC Brushless DC Motor

Gegen-EMK Gegen-Elektro Motorische Kraft

IC Integrated Curciuts

IGBT insulated-gate bipolar transistor

MOSFET Metal Oxide Semiconductor Field-Effect Transistor

PMSM Permanent Magnet Synchronous Motor

PTP Point to Point

PWM Pulsweitenmodulation

rpm rotation per minute

SMD Surface-Mounted Device

TMCL Trinamic Motion Control Language

USB Universal Serial Bus

Lateinische Symbole

a_{\max}	rpm/sek	maximale Beschleunigung
F	N	Kraft
$G_{0,P}(s)$	–	Übertragungsfunktion offener Positionsregelkreis
$G_{0,V}(s)$	–	Übertragungsfunktion offener Geschwindigkeitsregelkreis
$G_{RI}(s)$	–	Übertragungsfunktion Stromregler
$G_{RP}(s)$	–	Übertragungsfunktion Lageregler
$G_{RV}(s)$	–	Übertragungsfunktion Geschwindigkeitsregler
$G_{SI}(s)$	–	Übertragungsfunktion Regelstrecke
$G_{WI}(s)$	–	Führungsübertragungsfunktion Stromregelung
$G_{WP}(s)$	–	Führungsübertragungsfunktion Positionsregelung
$G_{WV}(s)$	–	Führungsübertragungsfunktion Geschwindigkeitsregelung
I_S	A	Phasenstrom
i_{str}	A	Idealer Phasenstrom
J	Nm m ²	Trägheitsmoment
k_{EMK}	V _{rms} /krpm	Gegen-EMK Konstante
K_{IRI}	sek ⁻¹	I-Anteil des Stromreglers
K_{IRV}	sek ⁻¹	I-Anteil Geschwindigkeitsregler
K_{PRI}	–	P-Anteil des Stromreglers
K_{PRP}	–	P-Anteil des Lageregler
K_{PRV}	–	P-Anteil des Geschwindigkeitsreglers
k_T	Nm/A	Drehmomentkonstante
l	m	Länge
L_S	H	Induktivität von Phase zu Phase
M	Nm	Drehmoment
P	–	Polpaarzahl im Rotor
q _{st}	Ink	Startposition

qz	Ink	Zielposition
r	rpm/sek ²	Ruck
R _S	Ω	Widerstand von Phase zu Phase
s _e	Ink	Betragsstrecke
t _b	sek	Beschleunigungszeit
t _e	sek	Zeitpunkt bis Ziellage
T _{I-reg}	sek	Zeitkonstante Stromregelung
T _{Leistung}	sek	Zeitkonstante Leistungsteil
T _{NRI}	sek	Nachstellzeit des Stromreglers
T _{NRV}	sec	Nachstellzeit Geschwindigkeitsregler
T _S	sek	Zeitkonstante Induktivität/ohmschen Widerstand
t _{Step}	sek	Schrittweite
t _v	sek	Einsatz der Bremszeit
T _{V-reg}	sek	Zeitkonstante Geschwindigkeitsregelung
u _{EMK}	V	Spannung der Gegen-EMK
u _{ind}	V	Induzierte Phasenspannung
u _k	V	Phasenspannung
v _{max}	rpm	maximale Geschwindigkeit
v _{Ramp}	rpm	Sollwert Geschwindigkeitsgenerator
v _s	rpm	Sollgeschwindigkeit Lageregler
V _S	Ω ⁻¹	Reziprokwert Phasenwiderstand

Griechische Symbole

α	sek ⁻²	Winkelbeschleunigung
θ	rad	mechanischer Drehwinkel
θ _{el}	rad	elektrischer Drehwinkel
ω	sek ⁻¹	Winkelgeschwindigkeit

Abbildungsverzeichnis

2.1.	Anordnung der MOSFETs an einem BLDC	4
2.2.	Spannungs- und Stromverlauf bei der Blockkommutierung	5
2.3.	Spannungs- und Stromverlauf bei der Sinuskommütierung	5
2.4.	Stator und Rotor eines Synchronmotors	6
2.5.	Anordnung der Hall-Sensoren	10
2.6.	PMSM mit Encoder	11
2.7.	Aufbau eines Encoders	11
2.8.	Beispiel einer Kaskadenregelstruktur	16
2.9.	Theoretisches Trapezprofil	22
2.10.	Theoretisches Sinoidenprofil	24
2.11.	Zeitverlauf des Rucks bei der Trapezinterpolation	26
2.12.	Zeitverlauf des Ruck bei Sinoidverlauf	28
3.1.	Stromregelung im Laplace-Bereich	32
3.2.	Geschwindigkeitsregelkreis im Laplace-Bereich	35
3.3.	Positionsregelkreis im Laplace-Bereich	38
3.4.	Simulation der Trapezinterpolation	42
3.5.	Simulation der Sinoideninterpolation	42
3.6.	TMC603-Eval Board	44
3.7.	Programmer von ST-Link	44
3.8.	Messstand des Drehmoments	44
4.1.	<i>TMCL-IDE</i> mit Parametereinstellungen	50
4.2.	Evaluierung der Trapezinterpolation	54
4.3.	Evaluierung der Sinoidinterpolation	55
4.4.	Strom/Beschleunigung bei Trapezinterpolation	56
4.5.	Strom/Beschleunigung bei Sinoideninterpolation	56
4.6.	Drehmomentverlauf bei unterschiedlichen Interpolationsarten	58
A.1.	Gemessener Spannungsverlauf der Gegen-EMK eines BLDCs	i
A.2.	Gemessener Spannungsverlauf der Gegen-EMK eines PMSMs	ii
A.3.	Datenblatt des BLDC Simulationsmotors	iii
A.4.	Datenblatt des BLDC Testmotors	v

Tabellenverzeichnis

2.1. Auflösungen der Sensoren eines acht-poligen Motors	12
2.2. Vergleich der Interpolationszeiten	25
3.1. Parametereinstellungen	46
3.2. Bei der Evaluierung verwendete Software	46
4.1. Experimentell bestimmte Reglerparameter	52
4.2. Vergleich der Zeiten der Trapezinterpolation	54
4.3. Vergleich der Zeiten der Sinoideninterpolation	55

1. Einführung

1.1. Einleitung

In der heutigen Zeit wird großer Wert auf Genauigkeit und Präzision von steuer- und regelbaren Motoren gelegt. Die Industrie wendet solche Verfahren für Fließbänder, Punktschweißroboter oder Wasserstrahlschneider an. Diese müssen in einer fest definierten Umgebung eine Position oder vorgegebene Koordinaten genau anfahren. Die Positionssteuerung ist somit in sehr vielen Einsatzgebieten zu finden. Für ein exaktes Anfahren sind daher alle physikalischen Störungen zu vermeiden und Umgebungsbedingungen einzubeziehen.

Ein wichtiger Faktor, der bei der Umsetzung in der Bachelorarbeit erfasst wird, ist der Ruck des Motors. Dieser macht sich beim Starten oder Bremsen eines Motors abrupt bemerkbar. Ein bestimmter mathematischer Ansatz kann diesen begrenzen und ein sanftes Anlaufen des Motors bezwecken. Um den physikalischen Schritt umsetzen zu können, wird auf ein bestimmtes Modul der Firma TRINAMIC zurückgegriffen. Das Modul verfügt über ein breites Spektrum an Funktionen, die einen Motor ansteuern können. Dieses wird im Rahmen der Bachelorarbeit um die neue Funktion erweitert und anschließend mit dem vorherigen Zustand, ohne Ruckbegrenzung, verglichen.

In der Praxis findet das Verfahren große Anwendung. Insbesondere der Transport von Behältern die mit Flüssigkeit gefüllt sind, wo es gilt Überschwappen oder Vermischung zu vermeiden. Zu Beginn der Arbeit ist eine Simulation eines einfachen Motormodells zu erstellen und auf ihre spezifischen Vor- und Nachteile mathematisch, sowie regelungstechnisch zu untersuchen. Vorteil: Durch die simulative und mathematische Sichtweise ist das Verhalten am konkreten Motor gut vorhersehbar und mögliche Unstimmigkeiten lassen sich korrigieren. Ebenfalls dienen die Vorberechnungen dazu, vor Beginn der praktischen Umsetzung eventuell auftretende Fehler erkennen und beheben zu können. Aus dem Titel der Bachelorarbeit ist ersichtlich, dass im ersten Schritt eine Simulation erstellt und diese im zweiten auf eine bestimmte Steuereinheit implementiert werden soll.

Zusammenfassend ist eine Regelung, die ein sanftes Anlaufen eines Motors bezwecken soll, mit bestimmten mathematischen Eigenschaften zu erstellen, zu testen und alle nötigen Informationen zu dokumentieren.

1.2. Gliederung und Vorgehensweise

Die Bachelorarbeit ist in fünf Kapitel eingeteilt, bestehend aus Einleitung, Grundlagen, Simulation und Implementierung, Evaluierung und einer Zusammenfassung mit Ausblick auf mögliche Zukunftsentwicklungen. Das Grundlagenkapitel beschäftigt sich mit den theoretischen physikalischen Eigenschaften von Motoren, Sensoren und dem Aufbau der Regelung. Ebenfalls werden die Fragen: „Was ist eine asynchrone Point to Point Steuerung?“ und „Warum wird sie eingesetzt?“ beantwortet. Damit ist das grundlegende Wissen für die Simulation und Implementierung bereitgelegt. Diese Kapitel beschäftigten sich im Anschluss mit der konkreten Simulation der Regelung, die später als Implementierung auf einer realen Hardware zur Ansteuerung eines Motors umgesetzt wird. Zusätzlich werden alle Hardware- und Softwarekomponenten, die für die Regelung eines Motors nötig sind, erläutert und beschrieben. Die Evaluierung testet die neuen programmierten Funktionen, die in die Software eingebettet sind, auf ihre jeweilige korrekte Ausführung. Im weiteren Verlauf ist die Erweiterung der gewonnenen Zusatzfunktion mit der bestehenden zu vergleichen. Nach der Umsetzung und Durchführung der Testreihe verdeutlichen einige Beispiele das praktische Einsatzgebiet. Abschließend gibt das Kapitel Zusammenfassung und Ausblick einen Eindruck, wie eine Positionssteuerung noch umgesetzt werden kann.

1.3. TRINAMIC Motion Control GmbH & Co. KG

Die Firma TRINAMIC Motion Control GmbH & Co. KG ist ein kleines Unternehmen mit Sitz in Hamburg Eimsbüttel. TRINAMIC hat sich auf Integrated Curciuts (IC) und auf Module für Motoren und deren Steuerung spezialisiert. Die Anwendungsgebiete der Produktpalette reichen von der Biotechnologie über die Labor-Automatisierung bis hin zum Umgang mit der Halbleitertechnik, die alle überwiegend im industriellen Bereich eingesetzt werden. Mit der eigenen Programmiersprache Trinamic Motion Control Language (TMCL) kann der Verbraucher mit wenigen Zeilen Quellcode einen Motor mit den gewünschten und vorhandenen Parametereinstellungen antreiben. Ebenfalls wird sehr viel Wert auf energieeffiziente Motorsteuerung gelegt, um dem heutigen Stand der Technik gerecht zu werden. Das Unternehmen beschäftigt vorwiegend Ingenieure, die im technischen sowie im kaufmännischen Fachbereich tätig sind. Außenstellen befinden sich in Singapur und Nord Amerika, um auch das globale Interesse an den Produkte von TRINAMIC vor Ort zu vertreten. Dabei blickt das Unternehmen auf 20 Jahre Kompetenzen und Erfahrungen zurück. Ebenfalls sind sie nach dem ISO9001(Qualitätsmanagement) und ISO13485 (Qualitätsmanagementsysteme im Medizinproduktbereich) zertifiziert, was einen gewissen Qualitätsstandard vertritt.

2. Grundlagen

Die in den Grundlagen genannten Themen, wie die Wahl des Motors und der Sensoren, der Regelung und die theoretischen Ansätze spielen für die spätere Umsetzung eine wichtige Rolle. Sie ebnet den Weg für das Verständnis der einzelnen elektronischen Komponenten und fassen alle wichtigen und nötigen Informationen zusammen. Zuerst werden die Motoren mit den jeweiligen angepassten Sensor beschrieben, um sie später in der Simulation als ein Motormodell beschreiben zu können. Darauf aufbauend werden zwei verschiedene Geschwindigkeitsprofile anhand der Ruckverläufe betrachtet.

2.1. Motorarten

Im täglichen Leben stoßen wir in den verschiedensten Bereichen auf Motoren, die in erster Linie für uns nicht ersichtlich sind. Ob es der Vibrationsalarm im Mobiltelefon, der Motor im Küchenmixer, die Wasserpumpe in der Waschmaschine oder der Lüfter in einem Laptop ist. Damit der Motor zum Laufen gebracht werden kann, bedarf es meist einer analogen oder digitalen Recheneinheit, die gewünschte Drehzahlen oder das Bloße fahren eines Motors regeln. Als Einstieg ist der grobe Aufbau eines Motors und seine Steuerung dargestellt.

Ein Motor ist eine elektrische Antriebsmaschine, die üblicherweise aus einem festen Stator und einem rotierenden Rotor aufgebaut ist. Der Stator besteht aus einer festen und robusten Metallummantelung, in der sich drei um 120° angeordnete Spulen, die aus einem reinen, isolierten, aufgewickelten Kupferdraht (Phase) bestehen, befinden. Die Anzahl der Spulen kann je nach Bauform und Motorart variabel sein. Der bewegliche Teil wird allerdings abhängig vom Motortyp unterschiedlich ausgeführt. Zum einen kann der Rotor aus einem Weicheisenkern mit und ohne Magnet, zum anderen aus einem reinen Permanentmagnet bestehen. Abhängig von der Spulenzahl (Polpaarzahl) ist ein solcher Permanentmagnet mit mehreren magnetischen Polen ausgestattet. Eine Rotation der Welle kann man dadurch erreichen, indem das elektromagnetische Statorfeld in seiner Ausbreitungsrichtung gegenüber dem magnetischen Rotorfeld verschoben wird. Um einen Wechsel des Statorfeldes zu erreichen, wird z. B. bei Gleichstrommotoren oder Asynchronmotoren je mit Kohlebürsten die Kommutierung (Ansteuerung der einzelnen Phasen) mechanisch ausgeführt. Es gibt Motorarten, bei denen die Kommutierung nicht mechanisch, sondern elektrisch stattfindet. Der Rotor verfügt über keine elektrische Verbindung zu der Steuereinheit.

In dieser Arbeit wird auf die Ansteuerung von einem Brushless DC Motor (**BLDC**) und einem Permanent Magnet Synchronous Motor (**PMSM**) eingegangen, welche die genannten magnetischen und rotorischen Eigenschaften aufweisen. Diese Art von Motoren können nicht einfach an eine Wechsel- oder Gleichspannung angeschlossen werden, sondern brauchen zur Ansteuerung einen speziellen elektronischen Aufbau. Dabei besitzt jede Spule einen elektrischen Anschluss, genannt Phase. In einem Drei-Phasen-Wicklungssystem sind, wie der Name sagt, drei Spulen mit einem gemeinsamen Sternpunkt mit jeweils einem Phasensanschluss aus dem Motorgehäuse geführt.

Um die genannten Motoren betreiben zu können, muss jede Phase separat angesteuert werden. Dies geschieht mit elektronischen Schaltern. Hier findet entweder der Metal Oxide Semiconductor Field-Effect Transistor (**MOSFET**) oder der insulated-gate bipolar transistor (**IGBT**) seinen Einsatz. Dabei handelt es sich um Halbleiterelemente, welche durch Anlegen einer Spannung einen Stromkreis schließen oder öffnen können. Abbildung 2.1 zeigt die typische Verdrahtung und Ansteuerung eines **BLDC** welche äquivalent zu dem eines **PMSM** ist.

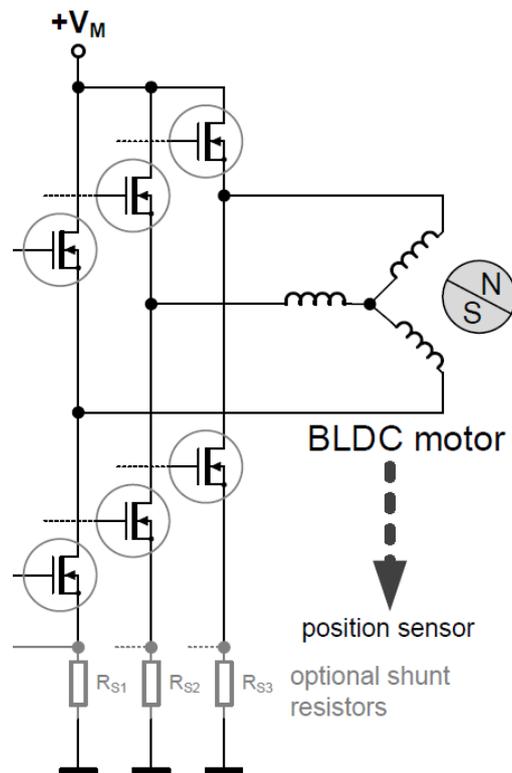


Abbildung 2.1.: Anordnung der **MOSFETs** an einem **BLDC** [1, S. 5]

Die Ansteuerung der Leistungsschalter geschieht über Pulsweitenmodulation (PWM), welche ein spezielles Signal darstellt und über ein Impuls-Pause-Verhältnis definiert ist. Ein Stromfluss durch eine Spule ist dann gewährleistet, wenn immer einer der oberen und entsprechend der Phasenführung einer der unteren MOSFET angesteuert wird. So ergibt sich je nach Lage ein anderes Bestromungsmuster und folglich unterschiedliche Ansteuerungen der Leistungsschalter. Bei diesem Verfahren spricht man von der Kommutierung. Mit anderen Worten wird der Stromfluss durch die Phasen elektrisch mit einer Recheneinheit gesteuert. Als Resultat zeigt sich bei einem BLDC ein blockförmiger und bei einem PMSM ein sinusförmiger Stromverlauf. Abbildung 2.2 und 2.3 legen die genannten Verläufe dar:

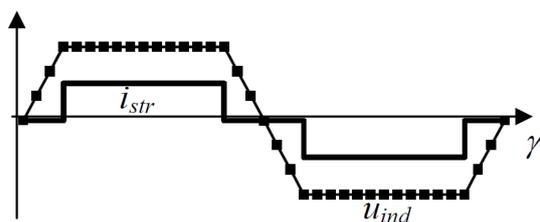


Abbildung 2.2.: Spannungs- und Stromverlauf bei der Blockkommutierung [2, S. 13]

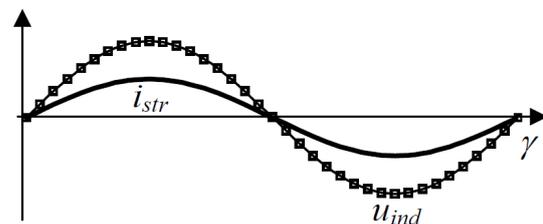


Abbildung 2.3.: Spannungs- und Stromverlauf bei der Sinuskommutierung [2, S. 13]

u_{ind} ist die induzierte Spannung an den Motorphasen und i_{str} ist der daraus resultierende, eingepreßte Strom.

Bei der Blockkommutierung werden stets nur zwei der sechs Leistungsschalter zyklisch geschlossen und führen jeweils immer nur durch eine Phase den Strom. Das stellt einen großen Vorteil gegenüber der Sinuskommutierung dar. Bei dieser Art der Kommutierung werden alle drei Motorleiter mit jeweils phasenverschobenen Spannungen durchgeschaltet. Damit nicht im Fehlerfall zwei obere und zwei untere Leistungsschalter geschlossen werden, sind zusätzliche Sicherheitsmaßnahmen sehr wichtig, um Kurzschlüsse in den Phasen zu vermeiden. Die eingerichteten Sicherheitsmaßnahmen bedeuten einen größeren Rechenaufwand im Vergleich zur Blockkommutierung.

Die genauere Funktionsweise der Ansteuerung bedarf keiner weiteren Vertiefung, da dieser Teil bereits von der Firma TRINAMIC realisiert und funktional übernommen wurde. Besprechung der Kommutierung und Funktionsweise der PWM Signale in [3].

2.1.1. Der bürstenlose Gleichstrommotor

Für den Versuchsaufbau kann ein bürstenloser Gleichstrommotor **BLDC**, auch elektronisch blockkommutierender Motor genannt, angeschlossen werden. Diese Motorart ist für einen Drehmomentbereich von 0.5 Nm bis 200 Nm ausgelegt, wobei die untere Grenze für Kleinantriebe noch kleiner sein kann. Für eine exakte Positionierung ist diese Motorart jedoch nicht geeignet.

Vorteile für die Verwendung eines **BLDCs** [4]: bürstenlos: verschleißfrei und mechanisch länger haltbar; geringe Stromwärmeverluste: der Rotor hat keine Spulen; wartungsarm: keine Verschleißteile bis auf Lager vorhanden; kurzzeitige Überlasten: ohne Probleme fahrbar; geringes Trägheitsmoment: der Rotationsteil besteht nur aus der Welle und einem Permanentmagneten. Nachteile eines **BLDCs**: hoher Steueraufwand: für Motor-Betrieb ist ein aufwändiger, elektronischer Aufbau nötig; Sensoren: Messung der Rotorstellung für die **PWM**.

Der Motor besteht aus einem Stator, indem sich in einem Drei-Phasen-Wicklungssystem drei um 120° angeordnete Spulen befinden. Und einem Rotor, der ein Permanentmagnet ist (siehe Abbildung 2.4). Die im Stator angeordneten Spulen rufen durch einen Stromfluss ein Magnetfeld hervor, die den Permanentmagnet durchströmen und so eine Kraftwirkung des Motors gemäß der Rechten-Hand-Regel ausüben. Dabei sollte das Magnetfeld des Läufers gegenüber dem Ständer phasenverschoben sein, um eine Drehbewegung sicherzustellen.



Abbildung 2.4.: Stator und Rotor eines Synchronmotors

Ein Stromfluss durch einen Leiter ruft ein elektromagnetisches Feld hervor, welches aus der Kraft wiederum ein Drehmoment erzeugt. Der mathematische Zusammenhang zwischen Drehmoment und Phasenstrom ist:

$$M = k_T \cdot i_{str} \quad (2.1)$$

k_T ist dabei die Drehmomentkonstante, welche vom Hersteller vorgegeben wird. i_{str} ist der Phasenstrom, der proportional zum Drehmoment ist [3]. Wegen der blockförmigen Stromverläufe entstehen bei dem BLDC große Drehmomentwelligkeiten [5]. Von insgesamt drei sind immer zwei Phasen stromführend und eine Phase stromlos. Aufgrund der Stranginduktivitäten entsteht eine Stromanstiegszeit, oder auch Kommutierungszeit, die sich auf das Drehmoment negativ ausübt [5]. Mit einem anderen Verfahren, der Sinuskommutierung kann das Drehmoment nahezu konstant gehalten werden, was im Kapitel 2.1.2 *Der permanent erregte Synchronmotor* beschrieben wird.

Begründet im mechanischen Aufbau kann nur eine bestimmte Anzahl von Rotorpositionen angefahren werden. Aus der Anordnung (Abbildung 2.1) der Spulen, können jeweils um 60° verschobene Winkel angefahren werden. Mit einem anderen Verdrahtungsverfahren und einer anderen Anordnung der Spulen, lassen sich mehr als drei der üblichen Polpaare realisieren. Allgemein hat ein Polpaar zwei gegenüberliegende Wicklungen, die in Abhängigkeit der Stromflussrichtung einen Nord- oder Südpol haben. Nun werden die Wicklungen, damit eine Winkeldifferenz zwischen Rotor und Stator besteht, zyklisch nacheinander bestromt, um eine Drehbewegung herzustellen (siehe Bestromungsmuster im Datenblatt des Motors A.3). Dadurch ist ein kleinerer Verfahrenswinkel möglich. Später wird dieser Winkel über einen Positionsgeber aufgenommen und der Steuereinheit zugeführt.

Die Umschaltung der Phasenströme und somit die Bewegung des magnetischen Feldes passiert mit den in Abbildung 2.1 beschriebenen MOSFETs. Geringere Schaltverluste und kleinere Schaltzeiten bieten einen großen Vorteil gegenüber den IGBTs [2]. Der BLDC-Motor wird blockförmig angesteuert, man spricht von einer Blockkommutierung, dargestellt in Abbildung 2.2. Diese bekommt über Hall-Sensoren (Abbildung 2.5) die Rückmeldung zu welchem Zeitpunkt eine bestimmte Phase ansteuerbar ist. Der Motor braucht gegenüber anderen elektrischen Maschinen wie z.B. Asynchronmotoren einen Sensor. Weiter gibt es noch Möglichkeiten einer sensorlosen Ansteuerung, welche die induzierte Gegen-Elektromotorische Kraft (Gegen-EMK)-Spannung an der je nach Kommutierungsmuster offenen Phase oder die hochfrequenten Anteile der Spannungen und Ströme messen und auswerten [2]. Nähere Informationen sind in Quelle [6] zu finden.

Das elektromagnetische Feld des Stators verändert sich zyklisch, sodass der Rotor stets in Bewegung bleibt und eine magnetische Abstoßung zwischen beiden Komponenten herrscht. Das rotierende Magnetfeld des Rotors induziert wiederum eine Spannung die Gegen-EMK genannt wird und der Quellspannung entgegenwirkt. Diese macht sich als Störung auf den Motor bemerkbar, wodurch die Phasenspannungen in ihren Amplituden reduziert werden. Bei dem BLDC ergibt sich damit ein markanter, trapezförmiger Spannungsverlauf. Die Messungen mit einem BLDC stützen die theoretischen Verläufe nicht ganz. Dazu wurde ein Motor an einem gemeinsamen Sternpunkt, über jeweils angeschlossene Widerstände, verbunden und unter rotieren der Welle die induzierte Spannung an einem Strang gemessen. Es zeigt sich eine Abweichung der idealen Verläufe. In Abbildung A.1 ist der gemessene Verlauf dargestellt.

2.1.2. Der permanent erregte Synchronmotor

Der **PMSM** ist ein sinuskommutierender Motor, der einen identischen Drehmomentbereich wie der **BLDC** hat. Vor- und Nachteile gibt es bei beiden genannten Motoren. Jedoch hat der **PMSM** aufgrund seines mechanischen Aufbaus einen besseren Rundlauf und wird häufig zusammen mit einem hochauflösenden Lagegeber für die Positionssteuerung eingesetzt [3]. Zusätzlich sind die Verluste im Motor geringer, da der Rotor aus einem Permanentmagnet besteht und hier keine Kupferverluste im Rotor auftreten können. Diese Eigenschaft trifft auf beide erwähnten Motoren zu.

Der **PMSM** ist eine Weiterentwicklung des **BLDCs**. Der typische Stromverlauf bei der Sinuskommütierung, der in Abbildung 2.3 zu sehen ist, kann entweder über eine sinusförmige Aufteilung der Spulen im Stator oder über eine sinusförmige Induktionsverteilung im Luftspalt erreicht werden [2]. Bei einem Drei-Phasen-Wicklungssystem befinden sich auch hier drei um 120° angeordnete Spulen, welche aufgrund der beschriebenen Anordnung keine trapezförmige, sondern eine sinusförmige **Gegen-EMK** an den Phasen hervorrufen. Der Spannungsverlauf der **Gegen-EMK**, welche mit demselben Verfahren wie beim **BLDC** gemessen wurde, zeigt dementsprechend eine zeitliche Sinusverteilung auf. Hier decken sich die theoretischen mit den gemessenen Verläufen komplett. Abbildung A.2 stellt den erwähnten Spannungsverlauf dar.

Die Anzahl der Polpaare ist mit dem bürstenlosen Gleichstrommotor identisch und kann ebenfalls durch ein anderes Verdrahtungsmuster erhöht werden. Man spricht hier auch von einem Synchronmotor, da das magnetische Feld des Stators identisch mit der Drehbewegung des Rotors läuft. Bei der Gruppe der Asynchronmotoren besteht hier eine Geschwindigkeitsdifferenz zwischen dem Magnetfeld des Stators und des Rotors, der mit dem Begriff Schlupf verknüpft ist. Das Drehmoment setzt sich mit der gleichen mathematischen Formel 2.1 zusammen, welche den proportionalen Zusammenhang zwischen dem Drehmoment und dem Phasenstrom beschreibt. Die Drehmomentwelligkeiten fallen hier bei hohen Drehzahlen und der Sinuskommütierung weg. Das Drehmoment zeigt einen zeitlich konstanten Verlauf. Letztlich hat der **PMSM** bei identischer Dimensionierung der Mechanik im Gegensatz zum **BLDC** eine größere Kraftumsetzung auf die Welle, was durch ein konstantes Drehmoment erreicht wird.

In der Antriebstechnik gibt es bei der Sinuskommütierung keine Signaltabelle, sondern eine Betrachtung über Raumzeiger. Jeder Phase bekommt eine Koordinate zugeteilt. Entsprechend der mechanischen Verteilung der Spulen von 120° fällt die grafische Aufteilung der Raumzeiger im gleichen Maße aus.

Ausgehend davon, dass die Wicklungen oder auch Wicklungsachsen nicht um ihre Lage verschoben werden, ändert sich nur die Amplitude des Raumzeigers aber die Ausrichtung bleibt konstant [4]. Das heißt, dass z. B. die induzierte Spannung an einer Spule nur um ihre Höhe und nicht um ihre Phase, bei der Betrachtung über Raumzeiger, variieren. Mathematisch werden die Zeiger über ein Vektorkonstrukt gebildet, näheres in Quelle [7].

Der Motor sollte nicht mit Hall-Sensoren, sondern mit einem Encoder bestückt sein, der auf das Gehäuse des Motors montiert wird. Dieser Aufbau wird häufig für die Positionssteuerung eingesetzt, da der Motor im Idealfall keine Gleichlaufschwankungen aufweist [2]. Da die Sinuskommutierung alle drei Phasen gleichzeitig bestromt, muss eine aufwändige Rotorpositionsmessung erfolgen. Die jeweiligen Kommutierungsgruppen müssen aufeinander abgestimmt sein, um keinen Kurzschluss zu verursachen. Aus diesem Grund ist eine hochauflösende Messung der Rotorposition unerlässlich. Die genaue Analyse der Funktionsweise siehe Kapitel 2.1.3 *Der Encoder*.

2.1.3. Lagemessung des Rotors

Die Messung der Lage des Rotors ist für die Kommutierung unverzichtbar. Aus der aktuellen Rotorlage wird das angelegte Magnetfeld des Stators nach der Sinus- bzw. Blockkommutierung berechnet. Zusätzlich werden die Beschleunigung und Geschwindigkeit aus der Information der Lage umgerechnet und der Regelstruktur als Istzustand zugeführt. Es gibt dabei einen starken Unterschied welcher Sensor hinsichtlich der Auflösung und der Genauigkeit Verwendung findet. Die Regelung des Motors ist auf die Funktionsweise mit Kommutierungsgebern (auch Hall-Sensoren) für **BLDCs** und die Verwendung eines Inkrementalwertgebers, Encoder genannt, für **PMSM** ausgelegt. Es ist zwingend erforderlich den Motor mit den angegebenen Sensoren zu bestücken. Je nach Kommutierungsart ist ein entsprechender Sensor auszuwählen, der die Informationen der Rotorlage an die Steuereinheit überträgt. Der Hall-Sensor sollte für die Blockkommutierung und der Encoder für die Sinuskommutierung eingesetzt werden, damit die Berechnungen zum Ansteuern der elektronischen Schalter fehlerfrei ablaufen. Es besteht allerdings auch die Möglichkeit einen **PMSM** mit Hall-Sensoren zu betreiben. Dabei lassen sich die Winkel zwischen den einzelnen Rotorpositionen interpolieren und erreichen dabei einen akzeptablen Motorrundlauf. Die Messtechnik bietet eine große Zahl anderer Lagegeber, unterteilt in zwei Hauptgruppen: In die absoluten (darunter fallen die Hall-Sensoren) und die inkrementellen Lagegeber (der verwendete Encoder). Beide haben weitere, zahlreiche Untergruppen.

Der Hall-Sensor

Der Hall-Sensor (Kommutierungsgeber) ist unerlässlich für die Messeinrichtung der Blockkommutierung. Die drei Sensoren sind bei BLDCs jeweils um 120° am Motorgehäuse befestigt und umschließen den rotierenden Permanentmagnet. Durch Auswertung und Messung des veränderlichen Magnetfeldes, kann eine Hall-Spannung (nach Edwin Hall) gemessen werden. Ein vertikal gerichtetes Magnetfeld - von den außen liegenden Spulen erzeugt - drängt die Elektronen in eine bestimmte Richtung und es entsteht eine Elektronendifferenz. Diese Differenz bewirkt einen Potentialunterschied, die man als Hall-Spannung misst [8].

Allgemein handelt es sich hier um relativ robuste und grob arbeitende Sensoren, welche ein binäres Signal senden, wenn ein magnetischer Pol an dem Hall-Sensor vorbeiläuft. Genauere Auflösung thematisiert Kapitel 2.1.3 *Vergleich der Auflösung der Sensoren*. Eine typische Signaltabelle der Hall-Signale ist in dem Datenblatt A.3 des BLDCs verzeichnet. Abbildung 2.5 zeigt die Messeinrichtung mit den Hall-Sensoren, welche sich im Inneren des Motors befinden. Zusammen mit dem restlichen Aufbau aus Abbildung 2.4 ist der Motor damit mechanisch komplett beschrieben.

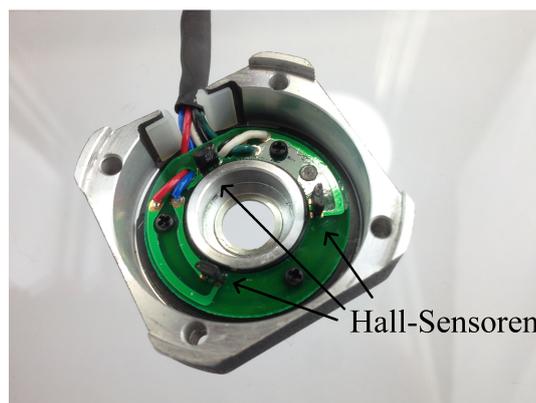


Abbildung 2.5.: Anordnung der Hall-Sensoren

Der Encoder

Der Encoder ist ein inkrementeller Lagegeber (Sensor), der in der digitalen Messwertfassung Anwendung findet und im Gegensatz zu den Hall-Sensoren nicht auf ein induktives Messverfahren setzt. Der Encoder gibt eine fest definierte Anzahl an Impulsen pro Umdrehung an, sodass aus der Information der Lage auch die Geschwindigkeit erfasst werden kann.

Analoge Tachogeneratoren werden immer häufiger durch Encoder ersetzt, weil der Lagegeber sehr schnell und ohne großen Verdrahtungsaufwand angeschlossen und die Signale direkt ausgewertet werden können. Bei der Installation ist stets darauf zu achten, dass ein Motor mit zwei austretenden Wellen eingesetzt wird, um auf der einen Seite die Lage zu erfassen und auf der anderen Seite eine Last betreiben zu können. Der in dem Aufbau verwendete Encoder hat eine Auflösung von 4000 Impulsen pro Umdrehung. Er gehört zu den niedrig auflösenden Messwerterfassungssystemen. Gängige Auflösungen erstrecken sich von 500 bis 32768 Impulsen pro Umdrehung bei einem relativ akzeptablen Preis (ca. 100 Euro), wobei die Grenze nach oben hin noch sehr viel höher sein kann.

Ein großer Nachteil ist, dass bei Abschaltung des Lagegebers die Information bezüglich der Lage verloren geht. Nicht mehr erkannt und gespeichert wird das Bewegen der Welle. Abhilfe schafft nur die Verwendung eines absoluten Encoders. Dieser wird über die Welle, welche zusätzlich mit einem Magnet zu bestücken ist, platziert und wertet das magnetische Feld direkt aus. Bei dem verwendeten ABN-Encoder kann die aktuelle Lage nur gespeichert werden, wenn nach dem Einschalten des Inkrementalwertgebers der Offset zwischen der Rotorposition und dem Statorfeld gemessen wird. Dieser Offset ist dann in der Berechnung des aktuellen Kommutierungswinkels anhand des Encoder-Signals zu berücksichtigen.

Der grobe Aufbau besteht aus einer Lichtquelle, einer Scheibe mit Teilstrichen (bis zu 10 μm Abstand) und Photozellen [4]. Die Scheibe befindet sich an einer mechanischen Befestigung an der rotierenden Welle des Motors. Nach der Lichtquelle passiert das Licht ein fest fixiertes Abtastgitter. Befinden sich Scheibe (mit den Teilstrichen) und Abtastgitter in einer Reihe, gelangt das Licht auf die Photozellen. Die Rotation beleuchtet oder beschattet die Photozellen. Die Änderung des Lichts wird in elektrische Energie in Form von Rechtecksignalen umgewandelt und der Steuereinheit des Moduls zugeführt [4]. Abbildung 2.6 und 2.7 zeigt den mechanischen Aufbau eines Encoder mit einem PMSM.

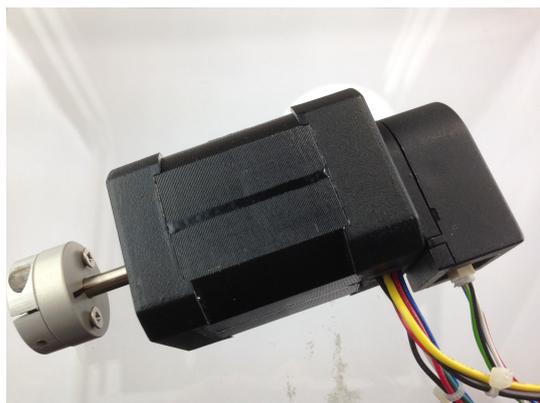


Abbildung 2.6.: PMSM mit Encoder



Abbildung 2.7.: Aufbau eines Encoders

Vergleich der Auflösung der Sensoren

Für die qualitative Bewertung und um einen Unterschied, bezogen auf die Genauigkeit bei der Positionierung zu erhalten, spielt die Anzahl der Pole des Motors eine große Rolle. Das nachfolgende Beispiel soll den Unterschied der Auflösung eines Encoders und eines Hall-Sensors deutlich machen: Es wird ein Motor mit acht Polen verwendet, damit ergeben sich vier Polpaare. Daraus folgt, dass eine mechanische Umdrehung vier elektrische Umdrehungen oder vier zeitliche Perioden des Motorstroms bedeuten. Betrachtet man den Encoder der mit einer Auflösung von 4000 Schritten pro mechanischer Umdrehung definiert ist, ergeben sich 1000 Schritte pro elektrischer Umdrehung.

Das Hall-Sensormesssystem besitzt drei um 120° angeordnete Sensoren, woraus sich sechs verschiedene binäre Zustände mit mindestens einem und höchstens zwei High-Zuständen zusammensetzen. Mit der Information bezüglich der Polpaare ergibt sich mit den sechs möglichen Zuständen eine Auflösung von 24 Schritten pro mechanische Umdrehung. Die Tabelle 2.1 hebt den Unterschied deutlich hervor:

Tabelle 2.1.: Auflösungen der Sensoren eines acht-poligen Motors

	Hall-Sensor	Encoder
Schritte je elektrische Periode	6	1000
Schritte je mechanische Umdrehung	24	4000

Im direkten Vergleich, bezogen auf die Auflösung, kann der Encoder punkten. Der verwendete Inkrementalgeber gibt 4000 Impulse pro Umdrehung aus. Auf 360° gesehen kann der Motor in $360^\circ / 4000$ Inkremente = $0,09^\circ$ Schritten betrieben werden - eine relativ gute Positionierung und ein guter Rundlauf des Motors werden erreicht. Der Hall-Sensor sendet 24 Impulse pro Umdrehung bei einem Laufwinkel von $360^\circ / 24$ Impulse = 15° aus, womit keine genaue Positionierung und kein gleichmäßiger Rundlauf des Motors stattfindet.

2.1.4. Motormodell

Aufbau, Funktionsweise und Kommutierungsart wurden für beide Motoren beschrieben. Der bisher vernachlässigte mathematische Zusammenhang soll daher jetzt genauer untersucht werden.

Die in den Motorphasen vorhandenen ohmschen und induktiven Verbraucher verursachen jeweils einen Spannungsabfall. Der Rotor wiederum induziert eine Gegenspannung, hier U_{EMK} , die der Quellspannung entgegenwirkt. Die Phasenspannung die sich aus den vorherigen Überlegungen zusammensetzt, wird wie folgt definiert [9]:

$$u_k = R_S \cdot I_S + L_S \cdot \frac{dI_S}{dt} + U_{EMK} \quad (2.2)$$

Wie bereits erwähnt ist die Spannung U_{EMK} eine negative Begleiterscheinung eines Motors und wird mit Formel 2.3 beschrieben [9]:

$$U_{EMK} = \omega \cdot k_{EMK} \cdot f(\theta_{el}) \quad (2.3)$$

Die Konstante k_{EMK} , dem Datenblatt A.3 zu entnehmen, beschreibt die Höhe der Gegen-EMK in Abhängigkeit des elektrischen Drehwinkels θ_{el} . Mit dem Produkt der Winkelgeschwindigkeit der Welle und der genannten Variablen ergibt sich die Spannung U_{EMK} . Bei einem BLDC resultiert mit Formel 2.3 ein trapezförmiger und bei dem PMSM ein sinusförmiger U_{EMK} -Spannungsverlauf.

Bei der Betrachtung der Auflösung der Sensoren wurde zwischen elektrischen Umdrehungen und mechanischen Umdrehungen unterschieden. Bei der Berechnung der Spannung U_{EMK} wurde der Begriff des elektrischen Drehwinkels θ_{el} eingeführt. Um jetzt einen Zusammenhang zwischen den elektrischen und dem mechanischen Drehwinkel zu schließen zeigt Formel 2.4 diesen an [9]:

$$\frac{d\theta_{el}}{dt} = \frac{d\theta}{dt} \cdot P = \omega \cdot P \quad (2.4)$$

Die zeitliche Ableitung des elektrischen Drehwinkels ist gleichzusetzen mit dem Produkt der Polpaare im Rotor und der Ableitung des mechanischen Drehwinkels. Aus dem Produkt der Winkelgeschwindigkeit und der Polpaarzahl stellt sich ein linearer Zusammenhang zu dem elektrischen Drehfeld dar. In der Antriebstechnik wird die übliche Angabe der Geschwindigkeit eines Motors meist in einer Drehzahl ausgedrückt [9].

$$\omega = \frac{v}{l} \quad (2.5)$$

Die Winkelgeschwindigkeit ω setzt sich somit aus der Geschwindigkeit v , z. B. in Umdrehungen pro Minute (rotation per minute ([rpm](#))) und dem Radius l zusammen.

Für eine regelungstechnische Betrachtung reichen die vorherigen Definitionen nicht aus, da sie nur den elektrotechnischen Zusammenhang deutlich machen. Um eine mathematische Beziehung und Beschreibung der einzelnen Motorelemente in einer Simulation zusammenfassen zu können, werden diese durch den Bildbereich näher beschrieben. In Folge dessen lässt sich in der Regelungstechnik ein logischer und praktischer Zusammenhang zwischen den Motorelementen und der angewandten Mathematik herstellen. Ebenfalls entfallen komplizierte und meist unverständliche mathematische Ausdrücke, wie Differentialgleichungen. Motorelemente lassen sich sehr gut und kompakt im Laplace-Bereich darstellen.

In den nachfolgenden Kapiteln wird dieser Ansatz verwendet. Mit dem Datenblatt eines [BLDCs](#) im Anhang [A.3](#) und den Formeln aus [\[3\]](#), [\[4\]](#) und [\[10\]](#) wird der spezifische Aufbau in den Laplace-Bereich umgerechnet.

2.2. Kaskadenregelung

In der Antriebstechnik kommen bestimmte Regelungen zum Einsatz um vorgegebene Geschwindigkeits- oder Beschleunigungsvorgaben am Motor umzusetzen. Beim Transport von z. B. Flaschen auf Förderbändern kann ein Motor direkt über eine Spannungsvorgabe (Drehmomentvorgabe) angesteuert bzw. betrieben werden. Um jedoch auf Änderungen von z. B. zu vielen Flaschen auf dem Förderband zu reagieren, kann es sein, dass der Motor bei aktueller Spannungsvorgabe die gewünschte Drehzahl nicht halten kann. Und jetzt kommt die Regelung ins Spiel. Mit ihrer Hilfe werden zeitliche Änderungen erfasst und in Form von z. B. einer Spannungserhöhung so reagiert, dass die gewünschte Drehzahl erreicht wird.

Der einfache Regelkreis setzt sich aus Sollwertvorgabe, Istwertrückführung (bzw. Messwerterfassung), Regler und Regelstrecke zusammen. Die Regelungstechnik bietet eine breite Palette an verschiedenen Regelungsformen. Darunter fallen die Kaskadenregelung, Zustandsraumregelung und Fuzzy-Regelung. Bei der Betrachtung im Zustandsraum, oder auch Regelung durch Zustandsrückführung genannt, handelt es sich um ein Mehrgrößenregelsystem. Man spricht von einer Zustandsraumregelung, wenn eine Stellgröße mehrere Ausgangsgrößen beeinflussen kann. Die Fuzzy-Regelung beschreibt das Verhalten eines Reglers in linguistischer Form [11]. Einsatzort dieser Regelung: Wenn Sensoren unscharfe Größen liefern und sich nicht über High-Low-Logik beschreiben lassen. Die Kaskadenregelung ist eine der am meisten eingesetzten Regelungen in der Antriebstechnik. Sie besteht im Wesentlichen nicht nur aus einer Regelschleife, sondern aus drei ineinander geschachtelten Schleifen. Dabei stellt die [...] "Stellgröße des überlagerten Regelkreises die Führungsgröße für den unterlagerten Regelkreis dar"[10][757 f]. Die einfachere Handhabung und gute Umsetzbarkeit ebnet den Weg für die Verwendung einer Kaskadenregelung, bestehend aus Positions-, Geschwindigkeits- und Stromregelung, der vorliegenden Bachelorarbeit bereit. Der Vorteil bei der Wahl einer Kaskadenstruktur ist die Ausregelung von Störungen, die in den jeweiligen Regelschleifen auftreten und dementsprechend minimiert werden, bevor diese an die nächste Regelstruktur weitergegeben werden. Daraus folgt, dass der unterlagerte Regelkreis immer schneller als der überlagerte Regelkreis sein muss. Ebenfalls kann eine Begrenzung der Sollwerte in den entsprechenden Regelkreisen vorgenommen werden, die den Motor und die elektrischen Komponenten schützen [4].

In der verwendeten Kaskadenregelung besteht die innerste Regelschleife aus einem Stromregler, um eine schnelle Erhöhung des Motormoments und damit eine schnellere Auswirkung auf das dynamische Verhalten des Motors zu erreichen [10]. Ebenfalls übergibt der Stromregler umgerechnete Spannungswerte an den Leistungsteil, die über ein **PWM** Signal die **MOSFETs** und damit die Kommutierung des Motors steuern. Der übergeordnete Regelkreis, also auch die Sollwertvorgabe für den Stromregler ist die Geschwindigkeitsregelung. Diese erhält direkte Geschwindigkeitssollwerte von dem überlagerten Positionsregelkreis. Abhängig von der momentanen Position wird eine Geschwindigkeit gemäß der gewählten Interpolation umgesetzt.

Die Lageregelung ist die letzte Ebene der Kaskadenregelung. Der Positionsregler erhält eine beliebige Zielposition, vergleicht diese mit der aktuellen Position und übergibt eine Zielgeschwindigkeit an den Drehzahlregelkreis, woraus eine rotarische Geschwindigkeit, die Winkelgeschwindigkeit umgesetzt wird [12].

Im einfachen Regelkreis findet eine Rückführung oder Messung des Ausgangszustandes statt. Im Kaskadenregler hingegen gibt es drei Messungen der Ausgangszustände. Der Stromwert wird über einen Shunt-Widerstand gemessen. Dieser ist meist ein kleiner ohmscher Widerstand an dem eine Spannung abfällt, die über eine definierte Skalierung die Stromwerte umrechnet. Für den Lageregler und Geschwindigkeitsregler werden die Istwerte über den Encoder oder über den Hall-Sensor erfasst. Zusätzlich lassen sich in den Kaskadenreglern noch andere Rückführungen, die das Verhalten der Regelung beeinflussen, integriert. Darunter fällt z. B. die **Gegen-EMK**, die der Versorgungsspannung der Spulen im Stator entgegenwirkt, sowie das Lastmoment. Mehr hierzu in Kapitel 3.1 *Simulation*. Ein beispielhafter Aufbau und die Verschachtelung der Schleifen sind in Abbildung 2.8 dargestellt.

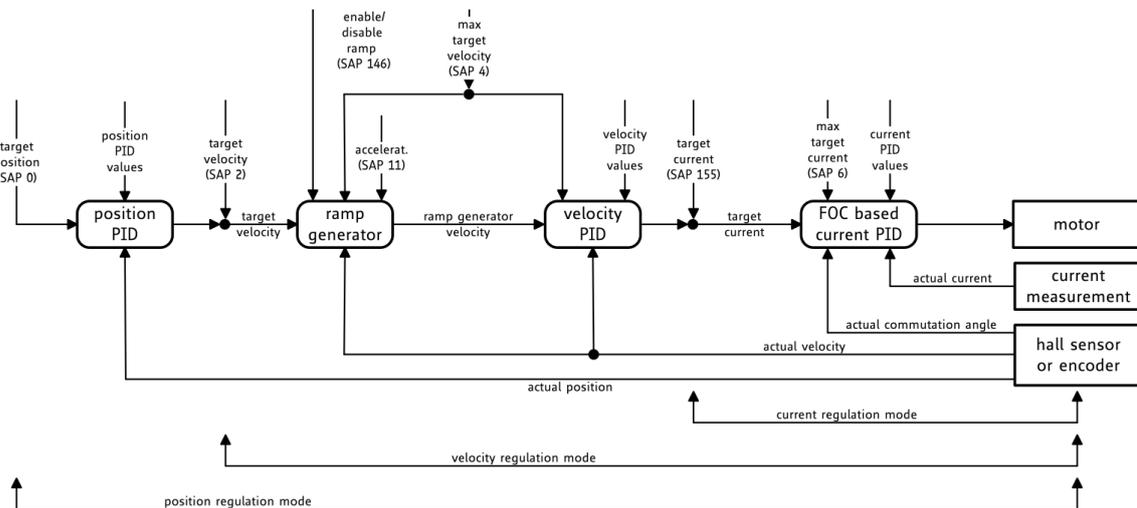


Abbildung 2.8.: Beispiel einer Kaskadenregelstruktur [13]

Die Abbildung zeigt den groben Aufbau der verwendeten Kaskadenregelung mit einigen zusätzlichen Elementen bestehend aus den drei Reglern, dem Rampengenerator, dem Motor und den Messwerterfassungssystemen. Die Messeinheiten, u. a. die Strommessung und der Hall-Sensor oder Encoder liefern in Echtzeit die aktuellen physikalischen Größen und führen sie der Regelstruktur zu. Die Regler und der Rampengenerator stellen die Hauptkomponenten der Regelung dar. Über eine Software, die in Kapitel 3.2.2 *Software* näher beschrieben wird, lassen sich alle relevanten physikalischen Größen eines Positionsprozesses anzeigen und den zeitlichen Verlauf darstellen.

2.2.1. Reglerauslegung

In der Regelungstechnik gibt es verschiedene Reglerarten, die einen Einfluss auf das gesamte elektrische System haben. Das heißt, dass für jede Regelstrecke ein gewisser Regler ausgewählt werden muss, damit die Ausgangsgröße sinnvolle und stabile Werte annehmen kann. Da in der Kaskadenregelstruktur auf verschiedene Sollwerte reagiert werden muss, ist es wichtig dementsprechend unterschiedliche Regler einzusetzen, um das gewünschte Resultat zu erreichen.

Für den Stromregelkreis wird daher ein Verfahren nach dem Betragsoptimum, im Kapitel 2.2.2 *Betragsoptimum* beschrieben, gewählt, damit der Istwert für einen großen Frequenzbereich dem Sollwert folgen kann [3]. Umgesetzt mit einem PI-Regler der mathematisch im Laplace Bereich wie folgt lautet:

$$G_{RI}(s) = \frac{K_{PRI} \cdot (1 + T_{NRI} \cdot s)}{T_{NRI} \cdot s} \quad (2.6)$$

Der Faktor K_{PR} stellt dabei den Proportionalwert oder auch P-Anteil des Reglers dar, der kontinuierlich die Differenz von Soll- und Istwert mit dem K_{PR} -Wert multipliziert. Eine weitere wichtige Größe ist die Nachstellzeit T_{NR} , die die größte Zeitkonstante in dem Regelkreis kompensiert. Für eine Simulation wird daher die größte Zeitkonstante ermittelt und dementsprechend die Nachstellzeit angepasst - schnellerer Regelkreis und gesenkte Ordnung dessen. In den meisten Anwendungen oder Programmen wird jedoch nicht die Nachstellzeit sondern der Verstärkungsfaktor angegeben:

$$K_{IRI} = \frac{K_{PRI}}{T_{NRI}} \quad (2.7)$$

Allgemein lässt sich die Wahl eines PI-Reglers damit begründen, dass das Regelverhalten sehr schnell und genau ist. Dabei vereint: die positiven Eigenschaften eines P-Reglers und eines I-Reglers. Der P-Regler ist ein sehr leicht umsetzbarer Proportionalregler, der zwar eine bleibende Regeldifferenz aufweist, jedoch dafür sehr schnell ist. Der I-Regler hat die positive Eigenschaft, dass er sehr genau, jedoch langsam ist.

Der ebenfalls für den Geschwindigkeitsregelkreis eingesetzte PI-Regler, eingestellt nach dem symmetrischen Optimum, wird im Kapitel 2.2.3 *Symmetrisches Optimum* beschrieben. Analog zu dem PI-Regler beim Stromregelkreis erhalten wir:

$$G_{RV}(s) = \frac{K_{PRV} \cdot (1 + T_{NRV} \cdot s)}{T_{NRV} \cdot s} \quad (2.8)$$

Zuletzt wird für den Lageregler ein nach dem Betragsoptimum eingestellter P-Regler verwendet. Dieser besitzt keinen I-Anteil, da sich in der Lageregelung bereits ein Integrator befindet und sonst die Regelung zum Schwingen angeregt und instabil werden würde. Er ist wie folgt beschrieben [12]:

$$G_{RP}(s) = K_{PRP} \quad (2.9)$$

Im Folgenden werden die Parametereinstellungen der Regler nach dem Betragsoptimum bzw. dem symmetrischen Optimum begründet und auf deren genaue Funktion eingegangen.

2.2.2. Betragsoptimum

Der Strom- und Positionsregler wird nach dem Betragsoptimum eingestellt. Das heißt, dass die Grundstrukturen der Regler erhalten bleiben und nur die Parameter gesondert berechnet werden müssen. Nach Führungsverhalten optimierte Regler, weisen ein Überschwingen (4,3 Prozent des Sollwerts) in der Sprungantwort auf [14]. Damit sich diese Bedingung erfüllt, muss der Frequenzgang für einen großen Bereich $|F(j\omega)| = 1$ sein. Große Stromspitzen, große Drehmomentänderungen und das dynamische Verhalten des Motors sollen so nicht zu stark angeregt werden. Bei dem Positioniervorgang wäre ein Überschwingen ebenfalls unvorteilhaft, da die Position angefahren und nicht durch ein Überschwingen erreicht werden soll. Als Beispiel dient ein Schweißroboter, der in einer abgegrenzten Kabine Schweißnähte in einem sehr beschränkten Umfeld versiegeln muss. Würden die Positionen bei einer Positionsansteuerung mit zu großen Überschwingen angefahren, könnte der Roboterarm an das zu bearbeitende Werkstück stoßen, es möglicherweise beschädigen oder selbst davon Schaden tragen. Deswegen wird der Positionsregler nach dem Betragsoptimum eingestellt. Bei dem Stromregler wird daher T_{NRI} gleich der größten Zeitkonstante gewählt, um diese zu kompensieren und das Regelsystem entsprechend zu beschleunigen.

2.2.3. Symmetrisches Optimum

Das Symmetrische Optimum kommt immer dann zum Einsatz, wenn der Regler ein PI-Regler und in der Strecke ein I-Glied, oder auch Integrierer, vorhanden ist [12]. Der Geschwindigkeitsregler wird daher nach dem Verfahren des symmetrischen Optimums eingestellt. Viele Störeinflüsse wie Reibung, Unwucht der Welle und die anzutreibende Last beeinflussen die Drehzahl des Motors [3].

Deswegen wird auf ein optimales Störverhalten sehr großen Wert gelegt. Die Verwendung nach dem Betragsoptimum würde zu einer sehr langsamen Störgrößenunterdrückung führen [12]. Für einen breiten Frequenzbereich werden die Störgrößen durch die Regelparameter-einstellung nach dem symmetrischen Optimum unterdrückt. Betrachtet man die Führungssprungantwort ist ein Überschwingen von bis zu 43 Prozent sichtbar [14]. Die Zeitkonstante T_{NRV} darf aus Stabilitätsgründen nicht die größte Zeitkonstante kompensieren. Die spätere Betrachtung der Regelstrecke des Geschwindigkeitsregelkreises zeigt IT1-Verhalten. Eine Kompensation der T1-Zeitkonstante, welche die größte Zeitkonstante ist, bringt durch Kürzen dieser den Regelkreis zum Schwingen. Setzt sich eine Strecke aus zwei I-Gliedern zusammen, zeigt der Regelkreis instabiles Verhalten. An einem konkreten Motor macht sich dies durch ein unkontrolliertes Zucken und Summen bemerkbar.

2.2.4. Zusammenfassung

Mit den genannten Grundlagen zur Kaskadenregelstruktur und mit einem regelungstechnischen Grundwissen, kann eine Simulation von einem Motormodell erstellt und genau wie die mechanischen Komponenten in die Simulation eingebettet werden. In der späteren Evaluierung spielen die genannten theoretischen Grundlagen eine wichtige Rolle. In Kapitel 4.1.2 *Experimentellen Parameterbestimmung* muss der Regler stets an das Verfahren nach dem Betragsoptimum und nach dem Symmetrischen Optimum für die jeweilige Strecke angepasst werden. Dies bedarf noch einiger Überlegungen, u. a. wie sich eine mechanische Komponente, wie z. B. der Rotor oder eine Strommessung mathematisch beschreiben lassen. Die genauere Dimensionierung der einzelnen Regler werden in Kapitel 3.1.3 *Kaskadenregelung* mathematisch näher beschrieben und liefern damit konkrete Werte für eine Simulation.

Bevor sich die Simulation umsetzen lässt, bedarf es dem Geschwindigkeitsgenerator besondere Achtung. Dieser übergibt die Geschwindigkeitssollwerte an den Geschwindigkeitsregelkreis. Damit das gesamte Motorsystem nicht mit zu großen Belastungen, wie Stromspitzen, aufgeschaltet wird, müssen die Geschwindigkeitswerte gesondert über eine Interpolation berechnet werden.

2.3. Asynchrone Point to Point Steuerung

In der Steuerung und Regelung von positionsgesteuerten Motoren werden Startpunkte erfasst, und Zielpunkte in Form von Koordinaten, indem das System arbeitet, angefahren. Dabei gibt z. B. der Anwender einer Software zur Steuerung eines Roboterarms eine gewünschte Endposition vor, die selbstständig und mit einer vorher definierten Genauigkeit angefahren werden soll [15]. Da ein Roboterarm mehrere Achsen und damit mehrere Motoren zum Steuern im Raum benötigt, sind diese auf unterschiedliche Art und Weise steuerbar. Es gibt dabei verschiedene Bahnsteuerungen u. a. die synchrone und die asynchrone Point to Point (PTP) Steuerung. Von einer synchronen Steuerung spricht man, wenn alle beteiligten Motoren, die zum Bewegen eines Roboterarms eingesetzt werden, zeitlich zum gleichen Zeitpunkt zum Stehen kommen. Damit sind alle Bewegungselemente voneinander abhängig. Bei einer asynchronen Steuerung tritt genau das Gegenteil auf, indem alle Motoren und damit alle Achsen vollständig unabhängig voneinander angesteuert werden und zu möglichen unterschiedlichen Zeiten zum Stillstand kommen [15].

Die Mechanik besteht meist aus längeren Hebeln, hier die Länge der Roboterarme, welche in Abhängigkeit von der Geschwindigkeit und Beschleunigung unterschiedliche Kräfte an den Achsen verursachen oder sogar das gesamte mechanische System zum Schwingen anregen. Dies lässt sich nur schwer mathematisch beschreiben, da es sich meist um nicht-lineare Zustände handelt, die unvorhersehbar auftreten können. Aus diesem Grund weißt eine PTP-Steuerung keine großen Beschleunigungs- und Geschwindigkeitssprünge auf und die Grenzen der Sprünge liegen immer im möglichen Grenzbereich oder Verfahrbereich einer ausführenden Bewegungseinheit, die aus mehreren Achsen besteht. Dabei werden die benötigten Geschwindigkeitswerte in einer Steuereinheit während einer Bewegung immer wieder neu berechnet und auf aktuelle Änderungen bzw. Belastungen, die ein Motor erfahren kann, eingegangen.

Bei der Programmierung der Steuerung konzentriert sich diese Bachelorarbeit nur auf die Verwendung eines Motors und damit einer Achse. Die maximale Beschleunigung und die Geschwindigkeit ergeben sich aus der anzufahrenden Zielposition und zyklusweise in einer Recheneinheit dem Motor zugeführt. Diese Berechnung heißt auch Interpolation. Bedeutet, ein Motor wird durch Wahl einer Interpolationsart mit unterschiedlichen Geschwindigkeits- und Beschleunigungsvorgaben betrieben. Dabei sind zwei unterschiedliche Interpolationsarten besonders hervorzuheben. Zum einen das Rampen- bzw. Trapezprofil und zum anderen das Sinoidenprofil. Das Trapezprofil wird auf den Modulen von TRINAMIC bereits umgesetzt und im Rahmen dieser Arbeit um das Sinoidenprofil erweitert.

Der Begriff Profil ist für den Verlauf einer Beschleunigung, Geschwindigkeit und Wegstrecke die während der Start- und Endposition eines Motors auftritt, definiert. Das heißt: Es können bei unterschiedlichen Interpolationsarten verschiedene Geschwindigkeits- und Beschleunigungsverteilungen zwischen Start- und Endlage auftreten.

Die Bezeichnung der Wegstrecke bezieht sich auf Umdrehungen der Motorwelle. Wird z.B. von einer Position von zehn gesprochen, sind damit auch zehn Umdrehungen der Welle gemeint. Die beiden folgenden Kapitel, [2.3.1 Trapezinterpolation](#) und [2.3.2 Sinoideninterpolation](#) beschäftigen sich mit den erwähnten Interpolationsarten und klären die mathematische Definition.

2.3.1. Trapezinterpolation

Bevor sich das konkrete Profil dargestellt, müssen zuerst alle verwendeten Variablen aufgeführt und erklärt werden. Das Ansteuern eines Zielpunktes passiert in drei aufgeteilten Segmenten. Das Beschleunigungs- und Bremssegment und einem mit konstanter Geschwindigkeit. Die Beschleunigungszeit wird mit t_b , die Zeit bis zum Einsatz des Bremsens mit t_v und die Zeit bis zum Treffen des Zielpunktes mit t_e beschrieben [15].

Das Rampenprofil ist eine Interpolationsart um von einem Startpunkt qst zu einem Endpunkt qz zu gelangen. In einem Modul werden Geschwindigkeitswerte berechnet und dem Motor als Spannung zugeführt, der die neuen Vorgaben umsetzt. Für die Interpolation zwischen zwei Punkten braucht man lediglich vier Parameter, die über eine Software an das Modul übertragen werden. Darunter fallen die maximale Geschwindigkeit in rpm , die maximale Beschleunigung in $\frac{rpm}{sek}$, der Startpunkt und der Zielpunkt. Mit diesen Parametern werden folgende Formeln definiert [15]:

$$s_e = |qz - qst| \quad (2.10)$$

$$t_b = \frac{v_{max}}{a_{max}} \quad (2.11)$$

$$t_e = \frac{s_e}{v_{max}} + t_b \quad (2.12)$$

$$t_v = t_e - t_b \quad (2.13)$$

Mit Hilfe der vier Parameter werden die geforderten Randbedingungen berechnet, die für die Trapezinterpolation erforderlich sind. Bei den folgenden Betrachtungen wird von einer Anfangsgeschwindigkeit von 0 rpm und bei Erreichen des Zielpunktes eine Endgeschwindigkeit von ebenfalls 0 rpm ausgegangen.

In [Abbildung 2.9](#) sind alle wichtigen und relevanten Verläufe dargestellt, die das Trapezprofil definieren. Dabei ist der mathematische Hintergrund zu beachten, der die Geschwindigkeit als erste Ableitung der Strecke und die Beschleunigung als erneute Ableitung definiert ($a = \dot{v} = \ddot{s}$). Über die Beschleunigung wird das gesamte Verhalten des Geschwindigkeits- und des Streckenprofils bestimmt.

Der Begriff Weg oder Strecke ist auf die Umdrehung des Motors bezogen. Abhängig von der Auflösung der Sensoren, die die Lage erfassen, kann somit eine Umdrehung hoch- oder

niedrigauflösend durchfahren werden. Folgende Simulation hat eine Auflösung von 60. Das heißt, eine Umdrehung eines Motors wird mit einer Positionsänderung von 60 angegeben. Somit ergibt sich hier eine Position von 15000, bei 250 mechanische Motorumdrehungen.

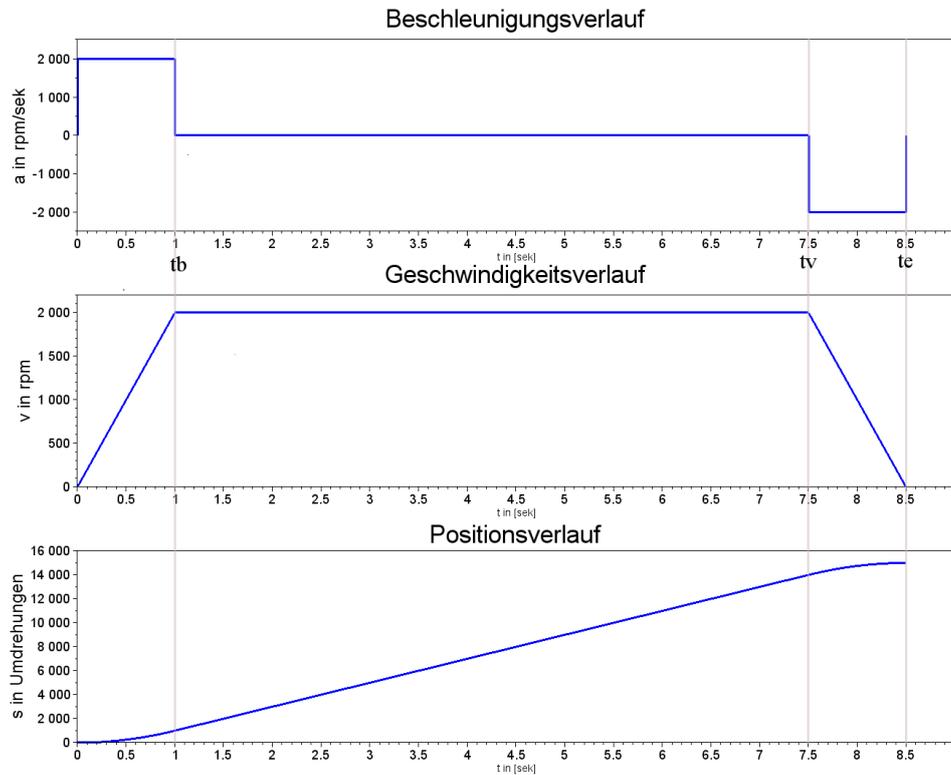


Abbildung 2.9.: Theoretisches Trapezprofil

Bei der Trapezinterpolation wird a_{max} bis zur Zeit t_b , hier 1 sek und zwischen den Zeiten t_v bei 7.5 sek und t_e bei 8.5 sek konstant angelegt, woraus sich eine positive Steigung und bei der negativen Beschleunigungsphase eine konstante Steigung bei dem Geschwindigkeitsprofil ergibt. Zwischen den Zeitpunkten t_b und t_v ist die Beschleunigung gleich Null, das Geschwindigkeitsprofil in diesem Bereich wird mit konstanter Geschwindigkeit durchfahren. Genau in dieser Zeitspanne zeigt sich ein linearer Anstieg des Weges. Auf eine fest definierte Position gesehen, ist die Zeit in der konstant gefahren wird im Vergleich zur Sinoideninterpolation größer. Später wird auf diese auftretende Diskrepanz eingegangen. Mit Sicht auf ein sanftes Anfahren entstehen bei dem dargestellten Beschleunigungsverlauf Probleme, die den Motor negativ beeinflussen. Die Sprünge während der Beschleunigung und des Bremsens führen zu einem ruckhaften Anfahren des Motors. Die Sinoideninterpolation zeichnet sich hier durch die Änderung der Beschleunigungsform positiv in einer Ruckbegrenzung aus.

2.3.2. Sinoideninterpolation

Für das Sinoidenprofil können dieselben Indizes, wie beim Trapezprofil verwendet werden, da der sinoide Verlauf der gleichen mathematischen Umgebung gehorchen muss. Der größte Unterschied bezieht sich auf den Verlauf der Beschleunigung, die nicht mehr konstant in der Beschleunigungsphase und Bremsphase, sondern zeitlich abhängig ist. Das heißt, dass bei der Trapezinterpolation die Beschleunigung sprunghaft aufgeschaltet wird und sich daraus ein unbegrenzter Ruck des Motors, der in Kapitel 2.4 *Der Ruck* beschrieben wird, entsteht [15]. Die zeitabhängige Beschleunigungsfunktion ist bei der Sinoideninterpolation, in Formel 2.14 erklärt, eine quadrierte Sinusfunktion, die ihren maximalen Beschleunigungspunkt bei $a_{max} = 2000 \text{ rpm/sek}$ hat, ist in Abbildung 2.10 zu sehen.

$$a(t) = a_{max} \cdot \sin^2\left(\frac{\pi}{t_b} \cdot t\right) \quad (2.14)$$

Mit der genannten Formel wird schrittweise das Geschwindigkeitsprofil berechnet, das den nachstehenden Gesetzmäßigkeiten folgt [15]:

$$s_e = | qz - qst | \quad (2.15)$$

$$t_b = \frac{2 \cdot v_{max}}{a_{max}} \quad (2.16)$$

$$t_e = \frac{s_e}{v_{max}} + t_b \quad (2.17)$$

$$t_v = t_e - t_b \quad (2.18)$$

Die Zeiten gelten für dieselben drei Segmente, die bei der Trapezinterpolation genannt wurden. Damit unterscheidet sich die Berechnung der Zeiten nur in einem Punkt und zwar in t_b (Formel 2.11 und 2.16). Daraus ergibt sich jeweils bei identischen $v_{max} = 2000 \text{ rpm}$ und $a_{max} = 2000 \text{ rpm/sek}$ eine Beschleunigungs- und Bremszeit von 2 sek.

Da der Anstieg der Beschleunigung nun nicht mehr sprunghaft ist, sondern einer Sinusfunktion gehorcht, wird ein sanfterer Anstieg/Abfall bei dem Geschwindigkeitsprofil mit dem identischen mathematischen Zusammenhang zum Trapezprofil ($a = \dot{v} = \ddot{s}$) bezweckt. Dieser sanfte Anstieg macht sich auch bei dem Streckenverlauf bemerkbar, wo der Übergang vom Anfangspunkt bzw. Endpunkt sachter in den linearen Wegverlauf übergeht.

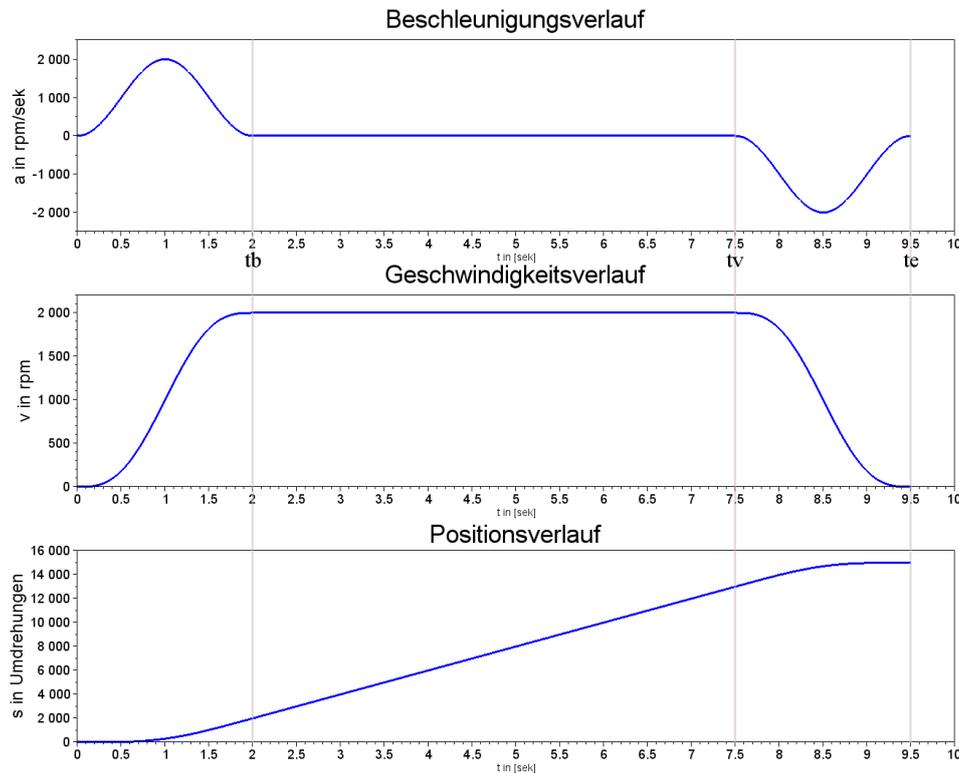


Abbildung 2.10.: Theoretisches Sinoidenprofil

Das Verhältnis der konstant linearen Wegstrecke zu der Summe aus dem Beschleunigungs- und Bremssegment ist jetzt kleiner als das genannte Verhältnis bei der Trapezinterpolation. Identische Translationsbedingungen zeigen damit einen zeitlichen Unterschied zu dem Trapezprofil auf. Eine Erhöhung der maximalen Beschleunigung stellt das Gleichgewicht, bezogen auf die Zeit die bei gleicher Position erreicht werden soll, wieder her. Welche Auswirkungen dies aber auf den Ruck hat, wird in Kapitel 2.4 *Der Ruck* geklärt. Ebenfalls lässt sich aufgrund der Beschleunigungsform eine Minderung der Belastung der mechanischen Komponenten eines Motors erahnen.

2.3.3. Zusammenfassung

Um einen direkten Vergleich von Sinoiden- und Trapezinterpolation zu erhalten, werden im Folgenden die einzelnen markanten Stellen zum Anfahren eines Endpunktes angesprochen, die die jeweilige Interpolationsart hervorheben.

Bei dem Rampenprofil wird wie bereits erwähnt, die Beschleunigung sprunghaft aufgeschaltet, wobei die Beschleunigung bei dem sinoiden Verlauf zeitlich abhängig und nur für einen kurzen Zeitraum bei der Beschleunigungs- und Bremsphase anliegt.

Daraus ergibt sich eine längere Zeit, die bei der Beschleunigung und beim Bremsen des Sinoidenprofils benötigt wird. Das lässt sich auch aus der Formel 2.16 herleiten. Ebenfalls hat eine Verschiebung der Maxima und Minima von a_{max} stattgefunden. Der Übergang von dem Anfangspunkt zum linearen Anstieg ist zusätzlich deutlich langsamer gegenüber dem Rampenprofil und somit werden die zeitlichen Translationsänderungen reduziert, die die mechanischen Komponenten vor zu starker Beanspruchung schützen.

Die Positionsdivergenz von 15000 Umdrehungen, bei 250 mechanischen Motorumdrehungen ist bei beiden Interpolationsarten identisch. Betrachtet man die Zeiten bis zum Erreichen der Zielposition, ist diese mit dem Trapezprofil schneller erreicht worden. Da bei dem Sinoidenprofil die Beschleunigung in zeitlich gestreckter Form auftritt, wird die Position um genau eine Sekunde später angefahren. Die Tabelle 2.2 listet alle markanten Zeiten und Translationsparameter auf.

Tabelle 2.2.: Vergleich der Interpolationszeiten

Interpolationsart	Trapez	Sinoid
a_{max} in rpm/sek	2000	2000
v_{max} in rpm	2000	2000
se in Positionen	15000	15000
t_b in sek	1	2
t_v in sek	7,5	7,5
t_e in sek	8,5	9,5
t_{konstV} in sek	6,5	5,5

Die Zeit t_{konstV} der konstanten Geschwindigkeit ist bei der Trapezinterpolation um eine Sekunde kleiner als bei dem Sinoidenprofil. Daraus und aus dem Beschleunigungsverlauf ergeben sich jeweils unterschiedliche Beschleunigungs- und Bremszeiten. Eine Anpassung der Beschleunigungs- bzw. Bremszeit ist dann nur über einen Parameter, dem Beschleunigungswert, möglich. In dem konkreten Fall, in Abbildung 2.10, muss die Beschleunigung verdoppelt werden, um eine identische Endzeit t_e zu erreichen. In der Testphase, d. h. während der Ansteuerung des Motors mit dem neuen Profil, muss eine Anpassung der maximalen Beschleunigung bei der Sinoideninterpolation vorgenommen werden - nur dann ist ein fairer Vergleich möglich. Die beiden theoretischen Interpolationsverläufe sind auf eine einheitliche Zielposition genormt und daraus erklärt sich die auftretende Diskrepanz der unterschiedlichen Endzeiten t_e .

Wie bereits erwähnt, macht sich die neue Interpolation mit einem verteilten Beschleunigungsverlauf bemerkbar, womit die Sinoideninterpolation gegenüber der Trapezinterpolation eine deutliche Ruckminderung ermöglicht. Welche Auswirkungen das auf den Ruck und wie dieser physikalisch definiert ist, stellt das folgende Kapitel klar.

2.4. Der Ruck

Wird ein Motor abrupt beschleunigt, so folgt der Rotor dem magnetischen Feld des Stators. Die Welle erfährt dabei eine ruckhafte Bewegung. Das Hauptziel dieser Arbeit ist es, eine Lösung zu finden, die genau dieser ruckhaften Bewegung entgegenwirkt, sodass der Motor ruckbegrenzt angesteuert werden kann. Mathematisch gesehen ist der Ruck die dritte Ableitung vom Weg und die erste Ableitung der Beschleunigung $\ddot{s} = r = \frac{da}{dt}$ [4].

Der Beschleunigungsverlauf bei der Trapezinterpolation hat während dem Beschleunigen einen positiven und beim Bremsen einen negativen konstanten Wert. Leitet man diesen zeitlich ab, erhält man zwei Impulse pro konstant anliegender Beschleunigung mit jeweils positiven und negativen Vorzeichen. Dies gilt auch in umgekehrter Weise für das Bremssegment. Aus Sicht der Signal- und Systemtheorie wird u. a. das Beschleunigungssegment wie folgt dargestellt [16]:

$$rect(t) = \left(\frac{t - t_b}{T}\right) - \left(\frac{t - t_v}{T}\right) \quad (2.19)$$

Die Konstante T steht für die Zeit t_b , sodass zum Zeitpunkt $t = 0$ die maximale Beschleunigung anliegt. Wird nun $rect(t)$ zeitlich abgeleitet entsteht der Ausdruck [16]:

$$rect'(t) = \delta(t) - \delta(t - t_b) - \delta(t - t_v) + \delta(t - t_e) \quad (2.20)$$

Abbildung 2.11 stellt damit die zeitliche Ableitung des vorher definierten Beschleunigungsverlaufes des Trapezprofils (Abbildung 2.9) dar.

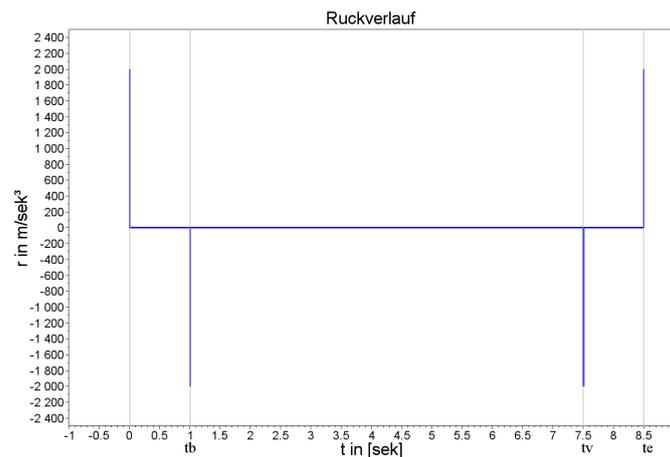


Abbildung 2.11.: Zeitverlauf des Rucks bei der Trapezinterpolation

Die Höhe des Energieimpulses wurde in der Abbildung 2.11 auf einen bestimmten Wert begrenzt, um diesen hier darstellen zu können. Gemäß der Definition eines Dirac $\delta(t)$ ist die Höhe abhängig von der Energieimpulsbreite [16]. Bei einer gegen unendlich kleiner strebenden Impulsbreite ist die Höhe, oder die Amplitude, unendlich hoch. Zum Zeitpunkt $t = 0$ strebt die Beschleunigung einen konstanten Wert an und hat damit eine unendliche Steigung - der Dirac $\delta(t)$ entsteht. Der Verlauf der Beschleunigung zwischen $t = 0$ und t_b ist konstant und hat somit eine Steigung von Null. In der praktischen Umsetzung ist der Energieimpuls nicht unendlich. Durch die mechanischen und elektrischen Komponenten, welche zusätzliche Verluste aufweisen, stellt sich ein begrenzter Dirac-Impuls ein. Die Gültigkeit des Energieerhaltungsgesetzes besteht somit zu jeder Zeit. Das Auftreten bei der Trapezinterpolation eines Impulses beansprucht das gesamte mechanische System stark.

Die Impulse treten nur zu einem sehr kleinen Zeitpunkt auf und um dies zu vermeiden wird das Sinoidenprofil verwendet, wo sich der Ruck auf eine größere Fläche verteilt und die Energie in begrenzter Form an das mechanische System abgegeben werden kann. Mit dem gegebenen mathematischen Beschleunigungsverlauf kann dies umgesetzt werden.

$$a(t) = a_{max} \cdot \sin^2\left(\frac{\pi}{t_b} \cdot t\right) \quad [15] \quad (2.21)$$

$$a'(t) = \frac{\pi}{t_b} \cdot a_{max} \cdot \sin\left(2 \cdot \frac{\pi}{t_b} \cdot t\right) \quad (2.22)$$

$$t_b = \frac{2 \cdot v_{max}}{a_{max}} \quad (2.23)$$

$$r(t) = \frac{a_{max}^2 \cdot \pi}{2 \cdot v_{max}} \cdot \sin\left(\frac{a_{max} \cdot \pi}{v_{max}} \cdot t\right) \quad (2.24)$$

Für den Bremsvorgang ergibt sich folgender Ausdruck, wobei t_v der Zeitpunkt ist, ab dem gebremst werden muss:

$$r(t) = \frac{a_{max}^2 \cdot \pi}{2 \cdot v_{max}} \cdot \sin\left(\frac{a_{max} \cdot \pi}{v_{max}} \cdot (t - 2t_v)\right) \quad (2.25)$$

Der Verlauf in Abbildung 2.12 zeigt grafisch an, wie sich der Ruck bei einem Positionsvorgang von 15000 Positionen bei einer maximalen Beschleunigung von $a_{max} = 2000 \text{ rpm/sek}$ und einer Geschwindigkeit von $v_{max} = 2000 \text{ rpm}$ zeitlich verteilt.

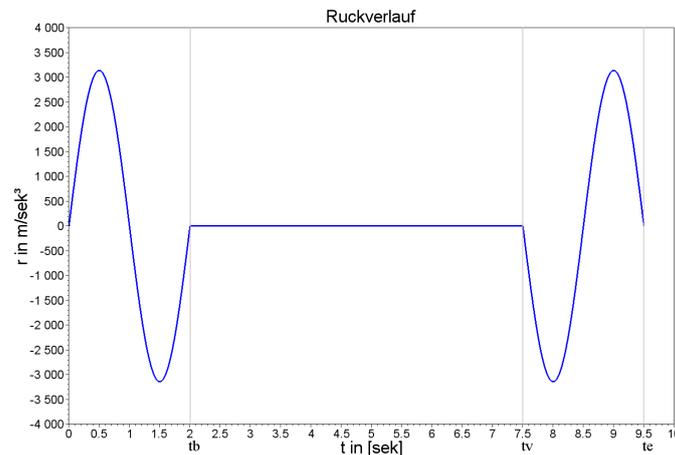


Abbildung 2.12.: Zeitverlauf des Ruck bei Sinoidverlauf

Der Sinoidenverlauf des Rucks zeigt, dass das Flächenintegral zeitlich verteilter ist und so eine zeitliche Verschiebung der Minima und Maxima stattgefunden hat. Der Übergang der Extremwerte verhält sich nach einer Sinusfunktion. Zum Anfangs- und Endpunkt besteht ein Ruck von Null. Zusätzlich wird deutlich mehr Zeit für den Beschleunigungs- und Bremsvorgang benötigt. Der Hauptgrund warum der Ruck bei dem Trapezprofil so stark zu beobachten ist, liegt darin, dass zu Beginn des Beschleunigungsvorganges die gesamte Energie des Rucks sofort verbraucht wird und dadurch keine zeitliche Verteilung auftritt. Bei dem Ruckverlauf der Sinoideninterpolation ist dies wesentlich sanfter und begrenzter verteilt. Im Gegensatz zu der Trapezinterpolation tritt kein Energieimpuls auf und der Ruck ist durch die Amplitude von Formel 2.24 begrenzt.

Zusammenfassend hat eine Verbesserung der Flächenverteilung des Rucks stattgefunden, um möglichst eine verteilte Energieabgabe an das mechanische und elektrische System zu gewährleisten. Durch die konstante Beschleunigung in der Beschleunigungs- und Bremsphase der Trapezinterpolation, führt die sprunghafte Änderung des Rucks zur Anregung z. B. der Robotermechanik und reduziert damit die Lebensdauer eines Motors. Bei der Sinoideninterpolation tritt dies nur im geringeren Maße auf [15]. Die bloße Streckung der Beschleunigung auf einen großen Zeitbereich, mit Hilfe eines kleinen Beschleunigungswertes, würde beim Trapezprofil nicht zu einer Ruckminderung beitragen. Der Energieimpuls würde am Anfang der Beschleunigung und am Ende des Bremsvorgangs trotzdem auftreten.

3. Simulation und Implementierung

In der Industrie, besonders in der Antriebstechnik, findet eine Überprüfung der theoretischen Algorithmen typischerweise durch Simulationen statt. Aus dem simulativen Umfeld lassen sich präzise Aussagen über die Umsetzbarkeit und Genauigkeit treffen. Nach der Simulation erfolgt die Implementierung, d.h. die Programmierung der Theorie, als Software für einen Mikrocontroller.

Der Grund warum vor der Implementierung eine Simulation erstellt werden sollte, ist zum einen mögliche Fehler vorher abfangen zu können und zum anderen der Kostenfaktor, der damit verbunden ist. Aus der Sicht des Projektmanagements sollte diese Reihenfolge stets eingehalten und durchgeführt werden, um spätere Unklarheiten aus dem Weg zu räumen und das Projekt in einem vorher festgelegten Zeitrahmen zu halten. Es kommt jedoch oft vor, dass die genannte Reihenfolge nicht beachtet wird und es später, im Endstadium eines Projektes, zu Komplikationen und damit zum Verzug des möglichen Fertigstellungstermins kommen kann. So wird die genannte Reihenfolge in den beiden folgenden Kapiteln eingehalten und jeweils beschrieben.

3.1. Simulation

Die im folgenden Kapitel verwendeten theoretischen Kenntnisse bauen auf dem Grundlagenkapitel 2 *Grundlagen* auf und sind unerlässlich für das vollständige Verständnis eines Motormodells. Hierzu sind alle mathematischen Berechnungen im Bildbereich dargestellt und beschreiben vollständig das einfache Motormodell, welches für die spätere Simulation der Kaskadenregelung benötigt wird.

Im Einzelnen finden Berechnungen der Regler in der Kaskadenregelung statt. Dazu werden die Rolle und die Funktionsweise des Geschwindigkeitsgenerators vorgestellt und beschrieben.

3.1.1. Simulationsprogramm

Es gibt verschiedene Simulationsprogramme die sich für die Bachelorarbeit eignen und die die Regelung simulieren können. *FreeMat*, *Scilab* und *GNU Octave* stehen dem Anwender kostenlos zur Verfügung. *Matlab* ist dabei das umfangreichste und damit auch das teuerste. Aus Kostengründen fällt dieses Simulationsprogramm aber weg. Zum Schluss ist die Wahl auf *Scilab* gefallen, da es vom Umfang und von der Bedienung stark an *Matlab* anknüpft.

Scilab ist ein numerisches, mathematisches Programm. Über eine einfache Syntax lassen sich mathematische Ausdrücke formulieren und als Ausgabe anzeigen. Voraussetzungen der Kenntnisse über die Programmiersprache C werden nicht benötigt. In der linearen Algebra sind solche Programme sehr beliebt, da sich z. B. Matrizen oder Vektoren mit vielen Elementen grafisch anzeigen und berechnen lassen.

Das Motormodell und die Kaskadenregelung könnten als Syntax in *Scilab* beschrieben werden, was jedoch zu sehr komplizierten und unübersichtlichen Ausdrücken führen würde. Um den Umgang und die Verständlichkeit zu steigern, wird ein Unterprogramm von *Scilab* verwendet. Dieses heißt *Xcos* und ist eine Simulationsumgebung mit der sich analoge, sowie digitale Signale und Systeme darstellen lassen. Diese werden in Form von Blockschaltbildern zusammengefasst und bilden eine übersichtliche Darstellung des simulativen Umfeldes. Die generelle Vorgehensweise einer kompletten Simulation setzt sich aus den folgenden vier Punkten zusammen:

1. **Scilab:** Auflistung aller Parameter, die der Simulation übergeben werden.
2. **Stack laden:** Variablen werden in *Scilab* mit den Werten in den Speicher geladen.
3. **Xcos:** Simulation wird gestartet.
4. **Scilab:** Erneutes Ausführen der Syntax erzeugt die grafische Darstellung der Simulation.

Als Übersetzer, der die Syntax in Maschinensprache umsetzt, wird der *Mingw Compiler* verwendet. Mit seiner Hilfe sind die Befehle in *Scilab* und *Xcos* zu übersetzen und auf Fehler, wie z. B. nicht verwendete Variablen, zu überprüfen.

3.1.2. Numerische Berechnung der Geschwindigkeitsprofile

Die in Kapitel 2 *Grundlagen* dargestellten Abbildungen (2.9, 2.10, 2.11, 2.12) zur Erklärung der Geschwindigkeitsprofile und des Rucks wurden mit *Scilab* generiert. Dazu wurde keine Simulation erstellt, sondern die Ausgaben nur mit Hilfe eines Algorithmus, angepasst an die jeweilige Anforderung, als reiner Quellcode generiert.

Um ein besseres Verständnis zu bekommen und den Umgang mit den Formeln (2.15 bis 2.18) zur asynchronen PTP-Steuerung genauer zu verstehen, wurden die Berechnungen in einer Art Vorsimulation getestet. Diese Vorgehensweise ermöglicht eventuell auftretende Fehler schneller zu finden und zu korrigieren.

3.1.3. Kaskadenregelung

Ein Motor besteht, wie in Abbildung 2.4 dargestellt, aus mehreren mechanischen Komponenten, wie z.B. Rotor, Stator, Spulen, Welle, Lager und dem Gehäuse. Diese einzelnen Komponenten sind im jeweiligen Regelkreis, in dem sie das Regelverhalten definieren und einen Einfluss haben, in der Simulation im Laplace-Bereich dargestellt. Die Kaskadenstruktur ist eine dreifach ineinander geschachtelte Regelstruktur. Im Folgenden von innen nach außen mit der jeweiligen Übersetzung von der mechanischen in die simulative Umgebung beschrieben. Nachstehende Simulationen basieren auf dem Datenblatt (siehe Abbildung A.3) eines PMSM. Dieser wird im Datenblatt zwar als BLDC beschrieben, aber bei genauerer Betrachtung der Gegen-EMK nimmt diese keine trapezförmige, sondern eine sinusförmige Form an. Für die Simulationen gilt somit die Betrachtung eines PMSM.

Die in der Simulation verwendeten Funktionsblöcke, mit dem Namen „To Workspace“, haben für die eigentliche Simulation keine regelungstechnische Bedeutung. Sie sind lediglich zur Darstellung der Ausgangsgrößen in die Simulation eingebunden.

Stromregelkreis

Der Stromregelkreis bestimmt den Stromfluss durch die Spulen und somit das gesamte Verhalten des Motors. Eine große Erhöhung des Stroms führt automatisch zu einer Zunahme des Drehmoments. Der behandelte Regelkreis ist direkt an den aus sechs MOSFETs bestehenden Leistungsteil gebunden und stellt die sensibelste, elektronisch beeinflussbare Größe dar.

Der Geschwindigkeitsregelkreis liefert als überlagerter Regelkreis den Stromsollwert, der in Abbildung 3.1 zu sehen ist. Anschließend wird der Sollwert mit dem Istwert verglichen und in Form einer Soll-/Istwert-Differenz an den Stromregler übergeben. Dieser liefert ein PWM-Signal abhängig von der Position des Rotors und gibt dem Leistungsteil eine Steuerungsspannung vor. Infolgedessen werden bei einem BLDC jeweils zwei der drei Spulen über zwei MOSFET und bei einem PMSM alle drei Leiter phasenverschoben bestromt. Das Statorfeld ändert seine Polung, welche der Polung des Rotors entgegenwirkt, womit eine Bewegung der Rotorwelle umgesetzt wird. Dies geschieht mit dem Modul *TMC603-Eval* der Firma TRINAMIC mit einer Frequenz von 10kHz aus der sich eine Zeitkonstante $T_{Leistung} = 100 \mu\text{sek}$ ergibt.

Der Leistungsteil wird über ein ständiges Impuls-Pause-Signal betrieben und ist als Totzeitglied zu betrachten. Insgesamt zeigt die Regelstrecke der Stromregelung Tiefpassverhalten, wodurch das Totzeitglied näherungsweise mit einem T1-Glied zu beschreiben ist [10].

Nach Ansteuerung der Leistungsschalter liegt eine Spannung an den jeweiligen Spulen an, welche durch die spezifische **Gegen-EMK** rotationsabhängig gegen die induzierte Spannung an den Statorspulen wirkt. Eine Rückführung der Istgeschwindigkeit wird hierzu in den Stromregelkreis aufgenommen und ist mit der **Gegen-EMK-Konstante** aus dem Datenblatt des Motors zu multiplizieren. Anschließend wird die Rotationsspannung von der Spannung an den jeweiligen Spulen abgezogen und erzeugt durch den ohmschen Widerstand $V_S = 1/R_S = 0.67 \Omega^{-1}$, welcher in dem Datenblatt A.3 des Motors aufgeführt ist, einen Stromfluss I_S . Eine Kupferspule besitzt immer einen ohmschen und einen induktiven Widerstand. Wird nun ein Strom durch die Spulen geleitet, erreicht dieser nicht sofort seinen Maximalwert, sondern zeigt einen exponentiellen Zeitverlauf. Aufgrund der Magnetisierung der Spulen tritt zusätzlich eine Verzögerung des Stromverlaufs auf. Zusammenfassend kann der Stromverlauf mit der Eigenschaft eines T1-Glied verknüpft werden und lässt sich mit der Zeitkonstante $T_S = L_S/R_S = 1.4 \text{ msek}$ beschreiben.

Nach der Darstellung der Motorwicklung wird der Stromwert dem Stromregelkreis vorne wieder zugeführt, wo erneut ein Soll-/Istwert-Vergleich stattfindet.

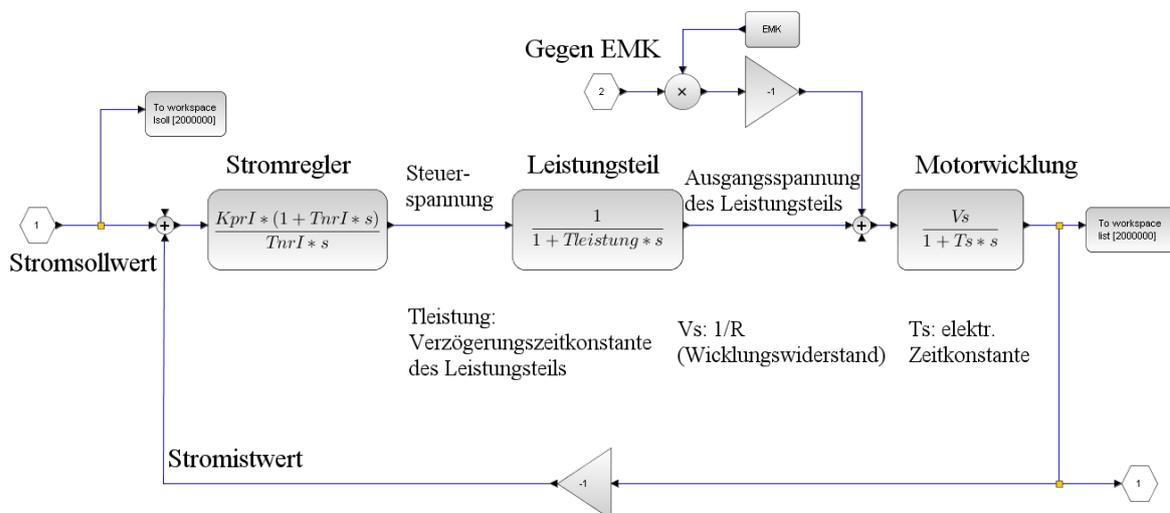


Abbildung 3.1.: Stromregelung im Laplace-Bereich

Aus den genannten stromabhängigen Komponenten kann mit Hilfe der Daten aus dem Datenblatt eine Auslegung des Stromreglers vollzogen werden. Hierzu werden die Proportionalanteile K_{PRI} und T_{NRI} bzw. K_{IRI} berechnet.

Ein nach dem Betragsoptimum eingestellter Stromregler zeigt somit ein Führungsübertragungsverhalten, dass durch Vorgabe eines Sollwerts den Istwert schnell erreicht. Dies wird durch Wahl der Zeitkonstante T_{NRI} gleich der größten Zeitkonstante (hier T_S) umgesetzt. Somit folgt:

$$T_{NRI} = T_S \quad (3.1)$$

Mit der genannten Vereinbarung senkt sich die Ordnung des Regelkreises und lässt eine einfachere Rechnungen zu. Der Proportionalitätsfaktor K_{PRI} setzt sich wie folgt zusammen:

Übertragungsfunktion des PI-Reglers:

$$G_{RI}(s) = \frac{K_{PRI} \cdot (1 + T_{NRI} \cdot s)}{T_{NRI} \cdot s} \quad (3.2)$$

Regelstrecke der Stromregelung:

$$G_{SI}(s) = \frac{V_S}{(1 + T_{Leistung} \cdot s) \cdot (1 + T_S \cdot s)} \quad (3.3)$$

Führungsübertragungsfunktion:

$$G_{WI}(s) = \frac{G_{RI}(s) \cdot G_{SI}(s)}{1 + G_{RI}(s) \cdot G_{SI}(s)} \quad (3.4)$$

Mit $T_{NRI} = T_S$ ergibt sich:

$$G_{WI}(s) = \frac{K_{PRI} \cdot V_S}{K_{PRI} \cdot V_S + T_{NRI} \cdot s \cdot (1 + T_{Leistung} \cdot s)}$$

$$G_{WI}(s) = \frac{K_{PRI} \cdot V_S}{K_{PRI} \cdot V_S + T_{NRI} \cdot s + T_{Leistung} \cdot T_{NRI} \cdot s^2}$$

$$G_{WI}(s) = \frac{1}{1 + \frac{T_{NRI}}{V_S \cdot K_{PRI}} \cdot s + \frac{T_{NRI} \cdot T_{Leistung}}{V_S \cdot K_{PRI}} \cdot s^2} \quad (3.5)$$

Koeffizientenvergleich wird mit $\vartheta = \frac{1}{\sqrt{2}}$ durchgeführt, um ein gutes Führungsübertragungsverhalten zu erreichen:

$$\frac{1}{1 + 2 \cdot \vartheta \cdot T \cdot s + T^2 \cdot s^2} \quad (3.6)$$

$$2 \cdot \vartheta \cdot T \cdot s = \frac{T_{NRI}}{V_S \cdot K_{PRI}} \cdot s \quad (3.7)$$

$$T^2 \cdot s^2 = \frac{T_{NRI} \cdot T_{Leistung}}{V_S \cdot K_{PRI}} \cdot s^2 \quad (3.8)$$

$$4 \cdot \vartheta^2 \cdot T^2 = \frac{T_{NRI}^2}{V_S^2 \cdot K_{PRI}^2} \quad (3.9)$$

Das Einsetzen von 3.8 in 3.9 ergibt:

$$4 \cdot \vartheta^2 \cdot \frac{T_{NRI} \cdot T_{Leistung}}{V_S \cdot K_{PRI}} = \frac{T_{NRI}^2}{V_S^2 \cdot K_{PRI}^2} \quad (3.10)$$

$$K_{PRI} = \frac{T_{NRI}}{2 \cdot V_S \cdot T_{Leistung}} \quad [3] \quad (3.11)$$

$$= \frac{1.4 \text{ msek}}{2 \cdot 0.67 \Omega^{-1} \cdot 100 \mu\text{sek}} = 10.448 \quad (3.12)$$

Aus dem Proportionalitätsfaktor folgt nun mit Formel 2.7 die Berechnung des Verstärkungsfaktors K_{IRI} :

$$K_{IRI} = \frac{K_{PRI}}{T_{NRI}} = \frac{10.448}{1.4 \text{ msek}} = 7462.86 \text{ sek}^{-1} \quad (3.13)$$

Damit sind alle relevanten Parameter des Stromregelkreises bestimmt worden und sind später in *Scilab* als feste Konstante einzubinden.

Geschwindigkeitsregelkreis

Der Geschwindigkeitsregelkreis ist der überlagerte Regelkreis des Stromregelkreises und bekommt seine Geschwindigkeitssollwerte von dem Positionsregler, der wiederum die überlagerte Instanz bildet. Der Geschwindigkeitsregler liefert nach dem Soll-/Istwert-Vergleich einen Steuerstrom an die Stromregelung. Anschließend wird der Stromwert durch eine Begrenzung geleitet, die den Strom auf den minimalen und maximalen zulässigen Wert begrenzt. Nach der Begrenzung erfolgt die Umrechnung der Stromwerte, siehe Strom-Drehmoment-Verhalten in Formel 2.1, mit Hilfe der Drehmomentkonstante $k_T = 0.035 \text{ Nm/A}$ in ein Drehmoment. Auf die Verwendung einer lastabhängigen Drehmomentkennlinie verzichtet man, da der eingestellte Regler nach dem symmetrischen Optimum zu realisieren ist und dieser Störungsverhalten gut kompensieren kann. Daher ist bei der Simulation ein konstantes Lastmoment im Einsatz und kein variables.

Der Rotor setzt eine Geschwindigkeitsvorgabe in eine Umdrehung des Motors um. Die Anzahl der Umdrehungen nimmt mit der Zeit zu, sodass der Rotor oder die drehende Mechanik als I-Glied angesehen werden kann. Voraussetzung: Der Motor dreht in eine Richtung. Mathematisch betrachtet nimmt mit steigender Zeit das Integral (Fläche) unter dem Weg (Umdrehung) zu. Der Rotor stellt einen Integrator (I-Glied) dar. Der Motor führt rotarische Umdrehungen aus, die bei der Bewegung der Masse (z. B. die Welle) ein Trägheitsmoment hervorrufen [17]. Die Konstante J stellt dieses Verhalten dar und ist aus dem Datenblatt eines Motors zu entnehmen.

Zum Schluss wird mittels einer Regelrückführung der Geschwindigkeitswert dem Anfang der Regeleinrichtung zugeführt und bildet damit einen in sich abgeschlossenen Regelkreis.

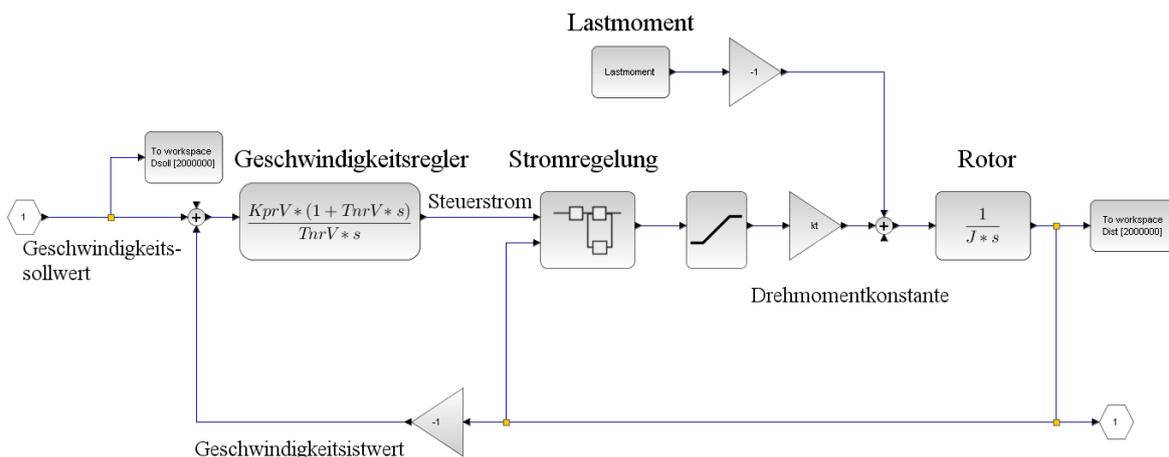


Abbildung 3.2.: Geschwindigkeitsregelkreis im Laplace-Bereich

Ein nach dem symmetrischen Optimum eingestellter Regler darf nicht die größte Zeitkonstante kompensieren, da sonst der Regelkreis instabil werden würde. Die Kompensation würde zum Kürzen des T1-Glieds führen und es blieben zwei Integratoren übrig. Der Regelkreis neigt dann zum Schwingen, mit dem Hintergrundwissen, dass in einem Regelkreis nie zwei Integratoren vorhanden sein dürfen. Bevor mit den Berechnungen der Parameter der Regler begonnen werden kann, muss die Stromregelung zuvor zu einem gemeinsamen Übertragungsglied zusammengefasst werden. Die Formel 3.11 wird wie folgt umgestellt:

$$2 \cdot T_{Leistung} = \frac{T_{NRI}}{V_S \cdot K_{PRI}} \quad (3.14)$$

Das berechnete Verhältnis eingesetzt in die Formel 3.5:

$$G_{WI}(s) = \frac{1}{1 + 2 \cdot T_{Leistung} \cdot s + 2 \cdot T_{Leistung}^2 \cdot s^2} \quad (3.15)$$

Die Sprungantwort eines PT1-Glieds und die eines PT2-Glieds mit der Dämpfung $\vartheta = 0.7$ zeigen einen ähnlichen Zeitverlauf, womit folgende Vereinfachung durchzuführen ist [4]:

$$G_{WI,vereinfacht} \approx \frac{1}{1 + 2 \cdot T_{Leistung} \cdot s} = \frac{1}{1 + T_{I-reg} \cdot s} \quad (3.16)$$

Somit hat $T_{I-reg} = 2 \cdot T_{Leistung} = 200 \mu\text{sek}$. Durch die gewonnene, einfachere Schreibweise ist die Ordnung des Geschwindigkeitsregelkreises niedriger.

Für die Berechnungen der Zeitkonstante T_{NRV} dient folgender Ansatz:

$$T_{NRV} = \alpha^2 \cdot T_{I-reg} \quad [10] \quad (3.17)$$

Mit $\alpha = 2$ ist der Betrag für einen großen Frequenzbereich = 1 [10] und ergibt mit $T_{I-reg} = 200 \mu\text{sek}$:

$$T_{NRV} = 2^2 \cdot 200 \mu\text{sek} = 800 \mu\text{sek} \quad (3.18)$$

Für die Bestimmung des P-Faktors K_{PRV} muss zuerst der offene Geschwindigkeitsregelkreis aufgestellt werden, um anschließend den Betragsgang (siehe 3.21) zu bilden:

$$G_{0,V} = \frac{G_{RV}(s) \cdot k_T}{(1 + T_{I-reg} \cdot s) \cdot J \cdot s} \quad (3.19)$$

Mit 2.8 ergibt sich:

$$G_{0,V} = \frac{K_{PRV} \cdot (1 + T_{NRV} \cdot s) \cdot k_T}{T_{NRV} \cdot s \cdot (1 + T_{I-reg} \cdot s) \cdot J \cdot s} \quad (3.20)$$

$$|G_{0,V}| = \frac{k_T \cdot \sqrt{1 + T_{NRV}^2 \cdot \omega^2} \cdot K_{PRV}}{J \cdot T_{NRV} \cdot \omega^2 \cdot \sqrt{1 + T_{I-reg}^2 \cdot \omega^2}} = 1 \quad (3.21)$$

Mit $\omega = \frac{1}{\sqrt{T_{NRV} \cdot T_{I-reg}}}$ [10], $\alpha = 2$ [10], Formel 3.17 und $J = 2.4 \cdot 10^{-6} \text{ Kg} \cdot \text{m}^2$ (aus dem Datenblatt des Motors) leitet sich folgender Ausdruck ab:

$$K_{PRV} = \frac{J}{2 \cdot k_T \cdot T_{I-reg}} = \frac{2.4 \cdot 10^{-6} \text{ Kg} \cdot \text{m}^2}{2 \cdot 0.035 \frac{\text{Nm}}{\text{A}} \cdot 200 \mu\text{sek}} = 0.171 \quad [3], [4] \quad (3.22)$$

Der Verstärkungsfaktor K_{IRV} wird damit zu:

$$K_{IRV} = \frac{K_{PRV}}{T_{NRV}} = \frac{0.171}{800 \mu\text{sek}} = 213.75 \text{ sek}^{-1} \quad (3.23)$$

Die Parameter des Geschwindigkeitsregelkreises können nun in *Scilab* als feste Konstante vereinbart und der Regelstruktur übergeben werden.

Positionsregelkreis

Der Positionsregelkreis stellt die letzte Instanz der Kaskadenregelung dar. Die Regelung mit dem jeweiligen Beschleunigungsverlauf wird auf die in den Produkten von TRINAMIC angewandte Regelung und Programmierumgebung angepasst. Dadurch ist der Programmieraufwand im Anschluss der Simulation erheblich kleiner und die komplette Regelung lässt sich kompakt als eine Funktionseinheit beschreiben und interpretieren. Der Positionsregelkreis besteht aus der Parameterübergabe, einem Geschwindigkeitsgenerator, der unterlagerten Geschwindigkeitsregelung, der unterlagerten Stromregelung, einer Sollwertbegrenzung und aus einem Integrator, der die Geschwindigkeitswerte aufintegriert ($\int v dt = s$).

Die Parameterübergabe ist ein wichtiger Schritt, um zu bestimmen in welchem Bereich sich die Regelung befinden soll. Darunter fallen die maximale Geschwindigkeit/Beschleunigung, die Auflösungsdarstellung (hier entspricht eine Umdrehung 60 Positionen), sowie Startposition (*actual position*) und Zielposition (*target position*).

Diese Parameter müssen auch in der Software vom Anwender eingegeben werden und definieren das Umfeld der Regelung. Falls die Zielposition sprunghaft auf die Regelung gegeben würde, hätte es negative Auswirkungen auf den Motor. Im ersten Moment kann der Strom sehr stark ansteigen und daraus eine schnelle Beschleunigung des Motors umsetzen. Zusätzlich verursacht die plötzliche Stromerhöhung einen starken Ruck.

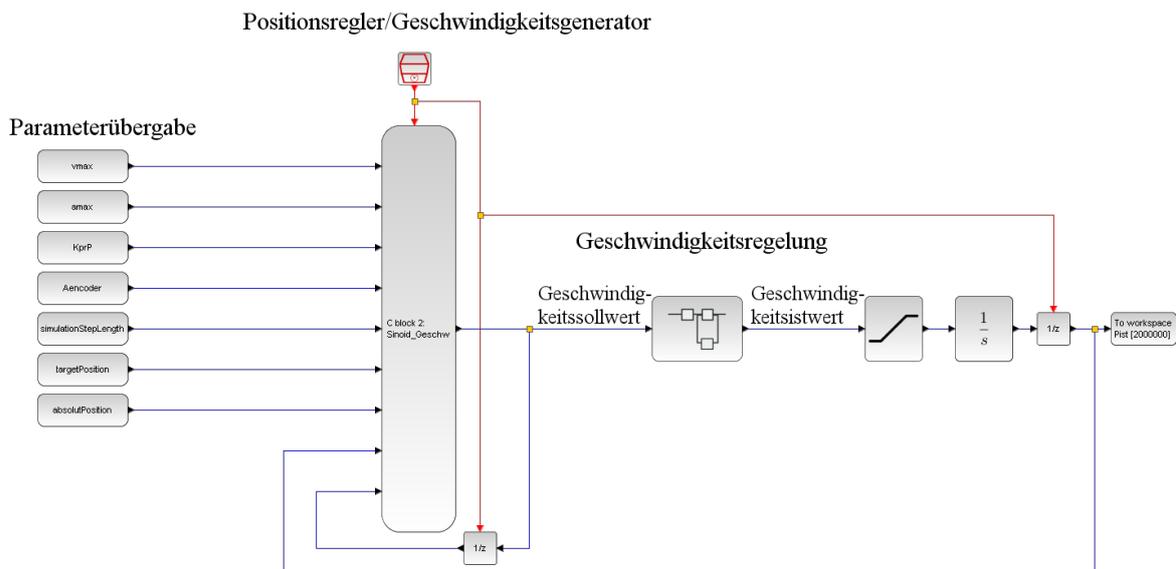


Abbildung 3.3.: Positionsregelkreis im Laplace-Bereich

Zu Beginn hat der P-Regler keinen Einfluss auf die Regelung. Das ist auch ersichtlich aus der Berechnung des K_{PRP} -Faktors (Proportionalwert des P-Reglers), welcher in den nachfolgenden Berechnungen hergeleitet wird. Der genannte Faktor setzt sich u. a. aus der Differenz der aktuellen und der gewünschten Position zusammen und würde zu Beginn der Motoransteuerung, mit maximalem Ausschlag, die komplette Regelung beeinflussen. Der K_{PRP} -Faktor ist erst beim Erreichen der Position aktiv und sorgt später dafür, dass die Welle die geforderte Position einhält. Der Geschwindigkeitsgenerator stellt eine wichtige Komponente, während der K_{PRP} -Faktor inaktiv ist, dar. Dieser interpoliert zwischen zwei Positionswerten einen resultierenden Geschwindigkeitsverlauf. Das heißt: Der Regelung werden kleine Geschwindigkeitsänderungen übergeben, woraus der Integrator im Positionsregelkreis Umdrehungen bzw. einen Weg umsetzt. Einerseits wird dadurch das dynamische Verhalten der Regelung in Grenzen gehalten und andererseits ein Überschwingen beim Erreichen der Position vermieden. Der Geschwindigkeitsgenerator besteht aus einem Funktionsblock, der in der Programmiersprache C ein Algorithmus implementiert, welcher später auf das Modul *TMC603-Eval* programmiert werden kann. Infolgedessen erhält man eine sofortige Überprüfung über die Funktionsweise des Algorithmus.

In der Abbildung 3.3 sind zwei identische Funktionsblöcke zu sehen, die in die Rückführung des C-Blocks integriert sind. Der diskret arbeitende Funktionsblock (C-Block) muss die verwendeten Messwerte der Regelstrecke zyklusweise übergeben bekommen, um bei einer definierten Simulationsschrittweite die Messwerte abarbeiten zu können. Diese Aufgabe übernehmen die Funktionsblöcke $1/z$.

Die Dimensionierung des K_{PRP} -Faktors benötigt die Zusammenfassung der Geschwindigkeitsregelung zu einem Übertragungsglied.

Die Bildung der Führungsübertragungsfunktion mit $G_{0,V}(s)$ (Formel 3.20) ergibt:

$$G_{WV}(s) = \frac{G_{0,V}(s)}{1 + G_{0,V}(s)} \quad [4] \quad (3.24)$$

$$G_{WV}(s) = \frac{K_{PRV} \cdot (1 + T_{NRV} \cdot s) \cdot k_T}{T_{NRV} \cdot s \cdot (1 + T_{I-reg} \cdot s) \cdot J \cdot s} \cdot \frac{1}{1 + \frac{K_{PRV} \cdot (1 + T_{NRV} \cdot s) \cdot k_T}{T_{NRV} \cdot s \cdot (1 + T_{I-reg} \cdot s) \cdot J \cdot s}} \quad (3.25)$$

$$G_{WV}(s) = \frac{K_{PRV} \cdot k_T \cdot (1 + T_{NRV} \cdot s)}{k_T \cdot K_{PRV} \cdot (1 + T_{NRV} \cdot s) + T_{NRV} \cdot J \cdot s^2 + T_{NRV} \cdot J \cdot T_{I-reg} \cdot s^3}$$

Mit $K_{PRV} = \frac{J}{2 \cdot k_T \cdot T_{I-reg}}$ und $T_{NRV} = 4 \cdot T_{I-reg}$ ergibt sich:

$$G_{WV}(s) = \frac{1 + 4 \cdot T_{I-reg} \cdot s}{1 + 4 \cdot T_{I-reg} \cdot s + 4 \cdot T_{I-reg}^2 \cdot s^2 + 8 \cdot T_{I-reg}^3 \cdot s^3} \quad [4] \quad (3.26)$$

Der Ausdruck scheint im ersten Moment sehr kompliziert, lässt sich allerdings weiter vereinfachen. Die Zeitkonstante T_{I-reg}^3 hat einen sehr kleinen Wert, der keinen nennenswerten Einfluss auf die Beschreibung des Geschwindigkeitsregelkreises hat [4]. Vergleicht man die Führungsübertragungsfunktion 3.26 und $G_{WV}(s)$ mit folgender vereinfachten Formel,

$$G_{WV}(s) \approx \frac{1}{1 + 4 \cdot T_{I-reg} \cdot s} = \frac{1}{1 + T_{V-reg} \cdot s} \quad (3.27)$$

so ist nur ein geringer Unterschied zu erkennen. Gleichung 3.26 zeigt lediglich ab -20 dB eine größere Steigung aufgrund der höheren Ordnung [4]. Folglich hat die Zeitkonstante einen Wert von $T_{V-reg} = 4 \cdot T_{I-reg} = 4 \cdot 200 \mu\text{sek} = 800 \mu\text{sek}$.

Mit der Abbildung 3.3 und der vereinfachten Übertragungsfunktion des Geschwindigkeitsregelkreises ergibt sich folgende Zusammenfassung des Positionsregelkreises. Zuerst wird dabei die Übertragungsfunktion des offenen Regelkreises und die darauf aufbauende Führungsübertragungsfunktion aufgestellt:

$$G_{0,P}(s) = \frac{K_{PRP}}{(1 + T_{V-reg} \cdot s) \cdot s} \quad (3.28)$$

$$G_{WP}(s) = \frac{G_{0,P}(s)}{1 + G_{0,P}(s)} \quad (3.29)$$

$$G_{WP}(s) = \frac{1}{1 + \frac{1}{K_{PRP}} \cdot s + \frac{T_{V-reg}}{K_{PRP}} \cdot s^2} \quad (3.30)$$

Mit Koeffizientenvergleich aus Gleichung 3.6 und mit $\vartheta = \frac{1}{\sqrt{2}}$ ergibt sich:

$$K_{PRP} = \frac{1}{2 \cdot T_{V-reg}} = \frac{1}{2 \cdot 800 \mu\text{sek}} = 625 \quad [3], [4] \quad (3.31)$$

Der Proportionalitätsfaktor beträgt $K_{PRP} = 625$ und setzt die Regeldifferenz zu Null.

3.1.4. Darstellung der Simulationsergebnisse

Die Simulation der Kaskadenregelung beinhaltet drei ineinander geschachtelte einzelne Regelkreise, die chronologisch von der Stromregelung über die Geschwindigkeitsregelung zu der Positionsregelung aufgeteilt und in Abbildung 3.4 und 3.5 dargestellt sind. Die Positions- und Geschwindigkeitsverläufe der beiden genannten Abbildungen zeigen bei $v_{max} = 2000 \text{ rpm}$ und $a_{max} = 2000 \text{ rpm/sek}$ eine Übereinstimmung mit den theoretisch berechneten Interpolationsverläufen, die in dem Kapitel 2.3 *Asynchrone Point to Point Steuerung* aufgeführt sind. Die einzige Ausnahme ist, dass der Stromregelkreis keinen direkten Beschleunigungswert, sondern einen Strom liefert. Bei den behandelten Synchronmotoren besteht ein direkter proportionaler Zusammenhang zwischen der Beschleunigung und dem Strom [12]. Das heißt, der Stromverlauf kann eine direkte Aussage über den Beschleunigungsverlauf liefern.

Der Stromverlauf ist ein Äquivalent zu dem Beschleunigungsverlauf und macht den Zusammenhang mit den folgenden Ausdrücken deutlich:

$$M = J \cdot \frac{d\omega}{dt} = J \cdot \alpha \quad [3], [17] \quad (3.32)$$

Dabei ist M das Drehmoment, J das Trägheitsmoment, ω die Winkelgeschwindigkeit und α die Winkelbeschleunigung. Der Strom I , der durch einen Leiter fließt, bildet ein magnetisches Feld H . Zu diesem Feld steht senkrecht die Kraft F und bewirkt eine Drehung des Motors. Aus der Kraft F und dem Hebel l der Welle ergibt sich:

$$M = F \cdot l \quad [3], [17] \quad (3.33)$$

Die Proportionalität zwischen dem Strom I und dem Drehmoment M ist hier zu erkennen.

$$M \sim I \quad (3.34)$$

Mit Formel 3.32 und dem proportionalen Zusammenhang stellt sich dar:

$$J \cdot \alpha \sim I \quad (3.35)$$

Aus der Formel lässt sich somit ableiten, dass bei konstant angesehenem Trägheitsmoment J der Strom I proportional zur Winkelbeschleunigung ist. So hat der Strom einen direkten Einfluss auf die Winkelbeschleunigung bzw. die Beschleunigung des Positionsprofil und umgekehrt. Mit Hilfe des Datenblatts A.3 fließt bei Last ein Strom von 1.79 A. Dieser spiegelt sich in den ersten Abbildungen 3.4 und 3.5 als Offset wider.

Der Regelkreis ist so konzipiert, dass sich mehrere Interpolationsarten in die Simulation eingebettet lassen. Die berechneten Regel- und Streckenparameter gelten jeweils für die Trapez- und die Sinoideninterpolation. Lediglich der Geschwindigkeitsgenerator muss um die Funktion der neuen Interpolation angepasst werden (siehe Abbildung 3.3). Einzige Änderung hierzu geschieht im C-Code in dem Funktionsbaustein C-Block.

Die zuerst betrachtete Simulation, ist die Trapezinterpolation. Die Zeitverläufe decken sich mit den vorausberechneten, theoretischen Verläufen. Der Stromregelkreis gehört zu der innersten Regelschleife einer Kaskadenregelung und hat somit die kleinste Zeitkonstante. Da die Periodendauer der Stromregelung am kleinsten ist, bildet sich hier eine Hüllkurve im Stromverlauf aus.

Abbildung 3.4 zeigt die erwähnte Interpolationsart mit der bekannten Einteilung der Translationszeitabschnitte (t_b , t_v und t_e).

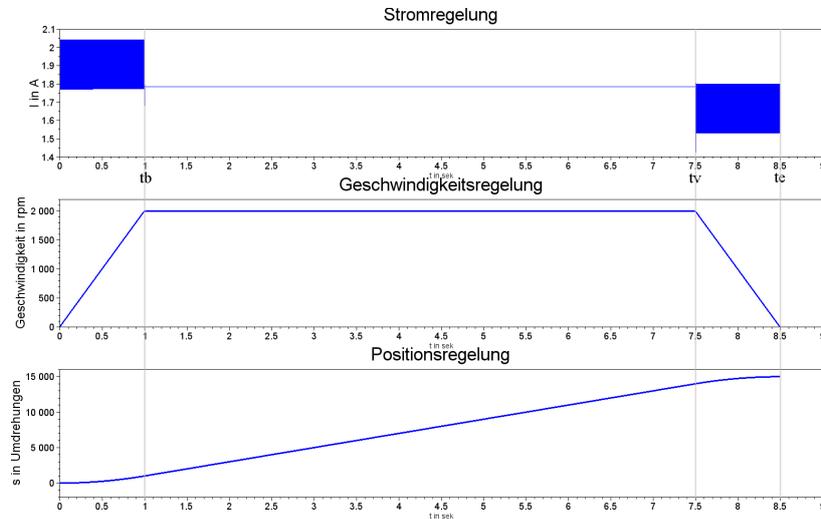


Abbildung 3.4.: Simulation der Trapezinterpolation

In Abbildung 3.5 ist der sinoide Verlauf aufgezeichnet.

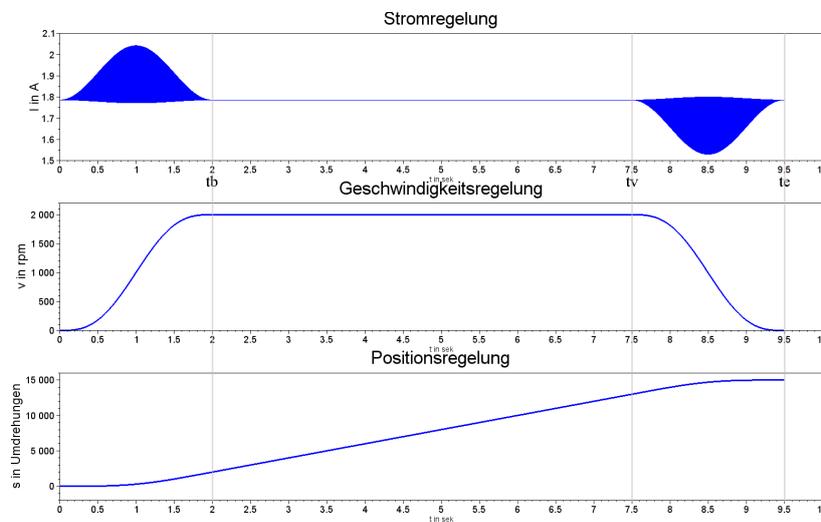


Abbildung 3.5.: Simulation der Sinoideninterpolation

Das Strom- und Beschleunigungsverhältnis trifft auch auf die Sinoideninterpolation zu. Das Erreichen des Zielpunktes zeigt sich mit einer Zeitverzögerung von einer Sekunde, anders als bei der Trapezinterpolation. Auch hier ist der Stromverlauf als eine Hüllkurve dargestellt.

Zuerst beschrieben wurden die theoretischen Profile. Diese liefern mit dem Simulationsergebnis eine Aussage über eine mögliche reelle Umsetzung. Die simulierten Verläufe der beiden Interpolationsarten stimmen mit den theoretischen Verläufen sehr gut überein. Die Simulationen führen jeweils eine Positionsdifferenz von 15000 an, welche sich ebenfalls mit den markanten Zeitabschnitten t_b , t_v und t_e unterteilen lassen.

Abschließend festzustellen: Beide Interpolationsarten arbeiten sehr genau, nur das bei ersterer der Ruck nicht in begrenzter Form auftritt. Die Simulationen legen den Weg für die Implementierung bereit und liefern einen definierten Startpunkt zur Programmierung.

3.2. Implementierung

Nach der Simulation der Algorithmen erfolgt die konkrete Umsetzung zu einem Quellcode. Dieser Schritt wird als Implementierung bezeichnet und ermöglicht die Umsetzung der theoretischen Überlegungen in ein reales System. Anschließend können die Simulationsergebnisse mit den gemessenen Werten überprüft und auf Unstimmigkeiten untersucht werden. Für die Module von TRINAMIC u. a. das *TMC603-Eval* findet die Programmierung in der Programmiersprache C statt. Diese Sprache bietet ein breites Spektrum von verschiedenen Bibliotheken, die das Programmieren erleichtern (z. B. *math.h*, *stdio.h*, *string.h* und *stdlib.h*). Durch die hardwarenahe Programmierung kann schnell eine Übersetzung der gedachten Aufgabe in eine reale Funktion erfolgen, was z. B. das Betreiben eines Motors bedeuten kann.

Die Programmiersprache C ist eine der wichtigsten Sprachen für eingebettete Systeme. Die elektronisch kommutierenden Motoren müssen über eine Regeleinheit - in einem sehr kleinen Zeitbereich - separat angesteuert werden. So müssen z. B. die Leistungsschalter zur PWM im Kiloherz-Bereich arbeiten. Andere Sprachen wie *Java* würden den Rechenaufwand auf einem Microcontroller nicht schaffen oder die benötigten harten Echtzeiteigenschaften nicht einhalten können.

3.2.1. Hardware

Für die Umsetzung der Simulation und die spätere Implementierung bedarf es einiger Hardwarekomponenten. Dazu gehören das Modul *TMC603-Eval* [13], ein Netzgerät, ein Motor (BLDC oder PMSM) mit Lagerfassung des Rotors und ein Datenübertragungskabel mit einem Programmer, um eine Verbindung von dem PC zum Modul herzustellen. Weiterhin wird zur Messung des Drehmoments ein spezieller mechanischer Aufbau genutzt. Das einstellbare Netzgerät liefert eine Spannung von 24 Volt und versorgt das in Abbildung 3.6 dargestellte Modul.

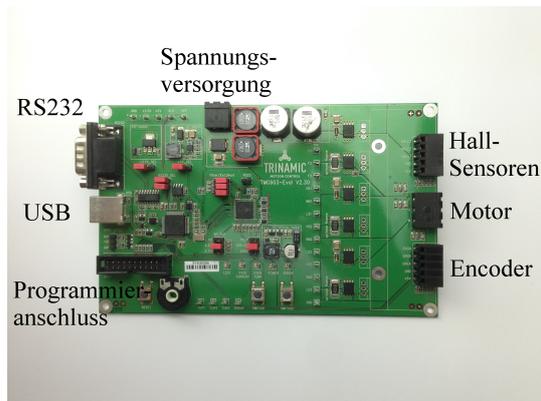


Abbildung 3.6.: TMC603-Eval Board



Abbildung 3.7.: Programmer von ST-Link

Über eine Universal Serial Bus (USB)-Schnittstelle sind PC-Software und Eval-Board verbunden. Ist diese erfolgreich, wird eine virtuelle, serielle Schnittstelle durch *Microsoft Windows* eingerichtet und der Datenaustausch beginnt. Das Überspielen der Firmware funktioniert am schnellsten über den 10-Pin-Programmierschluss, der mit in [Abbildung 3.7](#) dargestellten *ST-Link Programmer* verbunden ist. Dieser ist wiederum über eine USB-Schnittstelle am PC angeschlossen. Der verwendete Motor ist ein „BLDC“, nach den Messungen der *Gegen-EMK* eigentlich ein *PMSM*, von der Firma *Nanotec*. Die Übertragung des übersetzten Quellcodes geschieht mit dem Tool *ST-LINK Utility* [19] und sendet den Maschinencode auf den Mikrocontroller des Moduls. Für die spätere Validierung wird ein mechanischer Aufbau verwendet, mit dem man das Drehmoment messen kann. Dieser besteht aus vier wesentlichen Komponenten die in [Abbildung 3.8](#) zu sehen sind.



Abbildung 3.8.: Messstand des Drehmoments

Auf der rechten Seite befindet sich ein Motor, hier ein **PMSM**, mit Encoder. Auf der linken Seite ist eine elektrische Bremse montiert, die über eine externe Spannung die Bremskraft variabel erhöhen kann. Das eigentlichen Herzstück der Messanlage ist der Dehmomentmesser von der Firma *Sensor Technology* mit einem Messbereich bis maximal 4 Nm. Über eine Welle werden die genannten Komponenten miteinander verbunden. Der verwendete Sensor misst die Torsion der Achse an der Seite des Motors gegenüber der Achse auf der Seite der elektrischen Bremse. Die Stärke der Verdrehung wird in Form eines Drehmoments umgerechnet und als Messwertetabelle ausgegeben.

3.2.2. Software

Alle Simulationen wurden mit der Simulationssoftware *Scilab* durchgeführt, welche einen ähnlichen Aufbau und Funktionsumfang wie *Matlab* hat. Diese Software steht dem Anwender kostenfrei zur Verfügung¹. Die Simulationsoberfläche, in der die Funktionsbausteine, wie die Kaskadenregelung (in Kapitel 3.1.3 *Simulation* verwendet) aufgebaut sind, heißt *Xcos*. Das Unterprogramm ist mit dem bekannten *Matlab/Simulink* zu vergleichen. In *Scilab* können, als Syntax, numerische Berechnungen sehr leicht und schnell umgesetzt werden, die mit einem intern arbeitenden Compiler die Eingaben in Maschinensprache umsetzen. Möchte man allerdings ein möglichst reales System simulieren, ist die Simulationsoberfläche *Xcos* die bessere Wahl. Diese benötigt allerdings einen zusätzlichen Compiler, der die Funktionsblöcke, wie z. B. die Übertragungsfunktion eines Reglers oder den C-Block, auf Fehler überprüft und anschließend in Computersprache übersetzt.

Das genannte Eval-Board hat einen umfassenden Quellcode, der folgende Funktionen unterstützt: Motor-Ansteuerung, -Regelung, -Positionierung, Strommessung und Auswertung der Encodersignale. Dieses Grundgerüst wird komplett übernommen und um das Sinoidenprofil erweitert. Der Quellcode wird über die Entwicklungsumgebung *Eclipse*² geöffnet. In dieser wird der neue C-Code in den bestehenden Quellcode integriert. Als Compiler kommt der *G++ Lite* zum Einsatz, der den geschriebenen C-Code in Maschinensprache kompiliert. Um ein reales System, hier ein Motor, in einem virtuellen Umfeld zu simulieren, sind bestimmte Parametereinstellungen nötig. Darunter fallen die Länge der finalen Integrationszeit und die Art des Gleichungslösers.

¹www.scilab.org

²www.eclipse.org

Tabelle 3.1 zeigt die Einstellungen, die bis auf die finale Integrationszeit nicht zu verändern sind.

Tabelle 3.1.: Parametereinstellungen

Finale Integrationszeit	9.5E00
Echt-Zeit-Skalierung	0.0E00
Absolute Toleranz des Integrators	1.0E-10
Ralative Toleranz des Integrators	1.0E-10
Zeit-Toleranz	1.0E-12
Maximales Zeitintervall der Intgration	1.00001E05
Gleichungslöser	Sundials/CVODE - BDF - NEWTON
Maximale Schrittweite	0.0E00

Die Übertragung des Quellcodes geschieht mit dem Übertragungsprogramm *STM32 ST-LINK Utility*. Hier muss der in *Eclipse* generierte und kompilierte Quellcode erfasst und auf das gewünschte Modul überspielt werden. Aufbauend erfolgt nach dem Neustart des Moduls der Start der Software *TMCL-IDE*. Der Anwender steuert den Motor an verschiedene Positionen bei z. B. maximal erlaubten Motorströmen und maximal einstellbarer Beschleunigung und Geschwindigkeit. Dieses Programm gehört ebenfalls zu der Produktpalette von TRINAMIC und ist ein leicht zu bedienendes Tool mit einer übersichtlichen Struktur.

Die Tabelle 3.2 listet alle verwendeten Programme, mit Angaben der Versionsnummer und der Wortbreite auf.

Tabelle 3.2.: Bei der Evaluierung verwendete Software

Betriebssystem	Windows XP	32 bit
Simulationssoftware	Scilab 5.4.1	32 bit
Compiler 1	LCC-Win Compiler	32 bit
Entwicklungsumgebung	Eclipse IDE for C/C++ Developers	32 bit
Compiler 2	G++ Lite 2009q3-68	32 bit
Code-Übertragungsprogramm	STM32 ST-LINK Utility v2.4.0	32 bit
Modulsteuerprogramm	TMCL-IDE Version 3.0	32 bit

3.2.3. Erweiterung der Regelstruktur um das Sinoidenprofil

Die Erweiterung der Regelstruktur geschieht in einem kleinen Teil des Quellcodes, wo die Funktion des Sinoidenprofils implementiert werden soll. Dabei ist stets darauf zu achten, dass der neue Code den vorherigen Funktionszustand nicht ändert. Durch die numerische und spätere Simulation des Geschwindigkeitsgenerators, kann der Algorithmus relativ schnell angepasst und mit der zusätzlichen Funktion ausprobiert werden. Bei der Übermittlung des Berechnungsverfahrens ist zusätzlich auf verwendete Datentypen zu achten. In *Scilab* oder speziell in *Xcos* ist keine genauere Datentypangabe nötig. In *Xcos* ist ausschließlich in dem C-Block des Geschwindigkeitsgenerators die richtige Wahl der Datentypen wichtig.

Die Berechnungen des Geschwindigkeitsprofils werden in dem Datentyp *double* (Datenlänge von 8 Byte) auf dem Mikrocontroller vorgenommen. Damit die Berechnungen auch in den Grenzbereichen, wie am maximalen Geschwindigkeitspunkt, genau funktionieren, muss Rundungsfehlern große Beachtung geschenkt werden. Hierzu ein Beispiel: Für die Berechnung der Geschwindigkeitssollwerte ist ein Datenbereich *double* definiert. Da die Recheneinheit keine Fließkommazahlen ausgeben kann wird nur die Zahl vor dem Komma ausgegeben. Die Werte des Geschwindigkeitsgenerators müssen somit zyklusweise gerundet werden.

Eine hilfreiche Funktion, die eine grafische oder numerische Ausgabe in der *TMCL-IDE* zulässt, ist die Debug-Funktion. Mit ihr lassen sich zusätzlich zu den bestehenden Parametern, die aus dem Modul ausgelesen werden können, neu definierte Variablen ausgeben. Generell lassen sich Zwischenwerte der Berechnungen visualisieren oder den Ursprung des Fehlers deutlich machen.

Der während dieser Arbeit entwickelte Quellcode ist in zwei Hauptfunktionen unterteilt. Zum einen in den Geschwindigkeitsmodus und zum anderen in den Positioniermodus. Der erste setzt über eine gewünschte Geschwindigkeitsvorgabe diese um. Die Vorgabe einer neuen Geschwindigkeit oder stoppen des Motors, geschieht stets mit einem sinoiden Beschleunigungsverlauf, sodass theoretisch und praktisch der Ruck nur in begrenzter Form auftreten kann. In diesem Modus sind nur zwei der dreischichtigen Ebenen der Kaskadenregelung aktiv. Der Positionsregelkreis ist dabei nicht aktiv und der Regelstruktur werden Geschwindigkeitssollwerte vorgegeben. Der zweite Modus ist der Positioniermodus und besteht aus der Umsetzung der kompletten Kaskadenregelung mit allen enthaltenen simulierten Komponenten.

Zu Beginn der Positionsregelung müssen die maximale Geschwindigkeit, die maximale Beschleunigung sowie der Startpunkt und die Zielposition vorgegeben werden. Zum Schluss ausgewählt wird das Positionserfassungssystem. Damit stehen zwei Messverfahren zur Verfügung. Zum einen der Encoder und zum anderen der Hall-Sensor. Aus diesem Messsystem definieren sich wie viele Impulse oder Inkremente der Motor pro Umdrehung durchführt. Aus der Wahl der Lageerfassung und der Berechnungen der Geschwindigkeitssollwerte reduziert sich der Formelbestand auf das Wesentliche. Die erste Berechnung bezieht sich auf die Rotorerfassung und konkret auf den Weg der zum Beschleunigen/Bremsen benötigt wird. Die zweite Berechnung auf die Geschwindigkeitssollwerte:

Allgemein:

$$s(t_b) = \frac{1}{4} \cdot a_{max} \cdot t_b^2 = \frac{v_{max}^2}{a_{max}} \quad [15] \quad (3.36)$$

Hall-Sensor:

$$s(t_b) = \frac{v_{max}^2 \cdot 6 \cdot motorpole/2}{60 \cdot a_{max}} \quad (3.37)$$

Encoder:

$$s(t_b) = \frac{v_{max}^2 \cdot encodersteps}{60 \cdot a_{max}} \quad (3.38)$$

Geschwindigkeitssollwert:

$$v_{Ramp} = a_{max} \cdot \sin^2 \left(\frac{\pi \cdot a_{max}}{2 \cdot v_{max}} \cdot t_{Step} \right) \cdot t_{Step} \quad (3.39)$$

v_{Ramp} ist solange mit dem vorherigen Wert zu addieren bis die maximale oder die minimale Geschwindigkeit erreicht ist. Nach erfolgreicher Umsetzung stellt die reibungslose Überführung von dem Positioniermodus in den Geschwindigkeitsmodus und umgekehrt einen wichtigen Programmierschritt dar. Das heißt, es muss zu jedem Zeitpunkt ein Wechsel in den anderen Modus erfolgen.

4. Evaluierung

Nach erfolgreichem Abschluss der Simulation mit der darauffolgenden Implementierung ist nun der Weg für die Testphase bereitgelegt und es können die theoretischen Überlegungen an einem konkreten Motor getestet werden. Es findet damit eine Überprüfung der theoretischen Zusammenhänge statt und gibt einen Ausblick auf die Integrationsfähigkeit in die Produktpalette von TRINAMIC. Die Evaluierung soll ebenfalls dazu dienen, um einen Vergleich von Simulation und Testphase zu erhalten und auf ihre korrekte Ausführung zu erproben.

Im folgenden Kapitel wird dazu die Vorgehensweise der Testreihe erläutert, um später einen präzisen und qualitativen Vergleich zwischen Theorie und Praxis aufstellen zu können.

4.1. Vorgehensweise der Testreihe

Der Vorgang der Testreihe ist ein wichtiger Schritt der Evaluierung. Dabei stellt sich die Frage: Wie muss ich vorgehen, um eine gute Gegenüberstellung der Theorie mit der Praxis zu erreichen?

Die Testreihe beginnt mit der Trapezinterpolation, dann folgt die Sinoideninterpolation und zum Schluss werden die beiden Interpolationsarten miteinander verglichen und geben das endgültige Ergebnis der Ruckbegrenzung an. Im ersten Schritt werden die jeweiligen Profile anhand der markanten Beschleunigungs-, Brems- und Konstantfahrverläufen näher betrachtet. In der Zusammenfassung sind die Profile mit einer anderen Skalierung dargestellt und machen den Zusammenhang von Strom und Ruck deutlich. Ebenfalls findet eine Validierung statt. Diese soll zeigen, wie sich die neue Interpolation mit Hilfe einer Drehmomentkennlinie auf den Motor ausübt. Zum Schluss werden praktische Beispiele und der eigene Testaufbau zur Untersuchung des Rucks erläutert.

Die Wahl der Parametereinstellungen, die in der *TMCL-IDE* vorgenommen werden müssen, sind im Folgenden beschrieben.

4.1.1. Parametereinstellungen

Das Programm *TMCL-IDE* ist die Benutzeroberfläche von dem Eval-Board, indem alle Parametereinstellungen vorgenommen werden. Abbildung 4.1 zeigt einen kleinen Ausschnitt von der Benutzeroberfläche. Dazu kurz beschrieben: Die einzelnen Einstellungen und ihre Auswirkungen auf die Ansteuerung eines Motors.

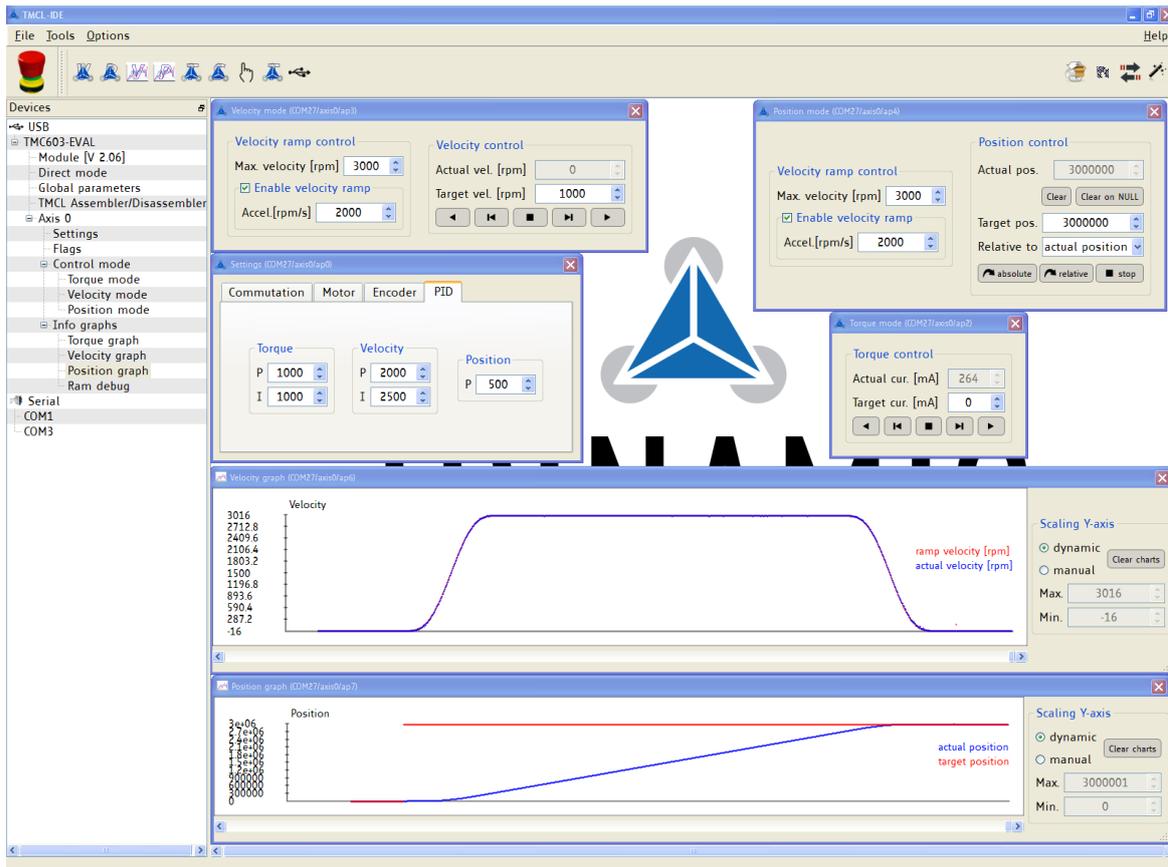


Abbildung 4.1.: *TMCL-IDE* mit Parametereinstellungen

In dem Funktionsabschnitt *Settings* aufgeführt sind die Auflösung des Encoders, die Anzahl der Motorpolpaare und die Einstellungen für die Regler der Kaskadenregelung. Die beiden erst genannten müssen für beide Betriebsmodi, den Geschwindigkeitsmodus und Positionsmodus, korrekte Einstellungen bekommen, um die Regelung möglichst genau durchführen zu können. Die Einstellungen für den Geschwindigkeitsmodus befinden sich in der Darstellung im oberen linken Bereich. Hier sind die maximale Geschwindigkeit und die Beschleunigung anzugeben, die Motorprofile definieren. Die gewünschte Geschwindigkeit wird im Bereich *target velocity* angegeben. Die Änderungen in der Software haben einen direkten Einfluss auf den Motor.

Im rechten Bereich der Abbildung 4.1 sind die Einstellungen für die Positionsregelung aufgezeigt. In Abhängigkeit von den maximalen Translationsparametern wird hier die aktuelle Position aufgenommen und die Positionszielvorgabe im Bereich *target position* eingegeben. Nun wird das Ziel relativ von der aktuellen Position oder als absolute Zielposition angefahren.

Die bei der Simulation bestimmten Reglerparameter können nicht bei der Evaluierung direkt umgesetzt werden. Eingabewerte in der *TMCL-IDE* lassen sich nicht als Fließkommazahl eingeben. Zusätzlich sind die Parameter mit einer anderen Skalierung im Quellcode dargestellt (I-Anteile werden durch 32768 und die P-Anteile durch 65568 geteilt und erst dann der Regelung zugeführt). Aus diesem Grund sind die Parameter für die Testreihe experimentell herauszufinden - für jeden Motortypen eine separate Bestimmung. Der Einfluss der Last und des Trägheitsmoments spielen dabei auch eine große Rolle. Spezielle Anpassungen brauchen beispielsweise die Parameter des Geschwindigkeitsreglers.

4.1.2. Experimentelle Reglerparameterbestimmung

In der Regelungstechnik sind Regelkreise über den Frequenzbereich (Reglersynthese) sehr genau definiert. Eine zusätzliche Art der Betrachtung ist die Auswertung der Sprungantworten, welche über einen Sollwert dem Regelkreis zugeführt und anschließend zeitlich betrachtet werden können. Das Programm *TMCL-IDE* stellt diese Betrachtung als grafische Ausgabe zur Verfügung. Dabei wird der Motor angefahren und gleichzeitig kann eine direkte Auswertung der Sprungantwort erfolgen. Somit können alle Reglerparameter selektiv von innen nach außen, gemäß der Kaskadenregelung, bestimmt werden. Bei der direkten Nutzung der unterlagerten Regelkreise müssen die Regler des überlagerten Regelkreises ausgeschaltet sein. So wird eine optimale Möglichkeit zum Einstellen bereitgelegt.

Beim weiteren Verfahren sollten alle Interpolationsarten, darunter fällt der Rampengenerator, ausgeschaltet sein. Messung und Darstellung geschieht mit dem *RAM Debug Mode* in der *TMCL-IDE*, in der die jeweiligen physikalischen Größen beobachtet und die Sprungantworten dargestellt werden können. Zu Beginn müssen alle Parameter der drei Regler auf null stehen. Die Stromregelung ist jetzt ausgewählt und die Motorwelle bei einer kleinen Stromvorgabe blockiert. Den P-Anteil solange erhöhen bis ca. 60 Prozent des maximalen Stroms erreicht sind. Unter Betrachtung der Sprungantwort muss der I-Anteil ebenfalls kontinuierlich erhöht werden bis sich ein Überschwingen von ca. 4.3% (nach dem Betragsoptimum 2.2.2) einstellt. Damit ist der Stromregler eingestellt und die neu bestimmten Werte sind abzuspeichern und für den Geschwindigkeitsregelkreis nicht mehr auf Null zu setzen. Für die Parameterbestimmung der Geschwindigkeitsregelung ist wie bei der Stromregelung vorzugehen, nur das man jetzt die Welle nicht mehr blockieren muss. Die Sprungantwort ist auf ein Überschwingen von 43.4% (nach dem symmetrischen Optimum 2.2.3) einzustellen.

Dabei ist der Motor über eine Geschwindigkeitsvorgabe anzusteuern. Die letzte Instanz ist der Positionsregelkreis. Dabei ist der P-Anteil solange zu erhöhen bis kein Überschwingen und keine bleibende Regeldifferenz mehr zu sehen sind. Die Tabelle 4.1 zeigt die Parameter, die bei den Testreihen verwendet wurden:

Tabelle 4.1.: Experimentell bestimmte Reglerparameter

	Strom-	Geschwindigkeits-	Positionsregelung
P-Anteil	1000	1400	100
I-Anteil	600	3000	—

Mit den genannten Parametern ist eine funktionsfähige Motorregelung möglich. Die Vorgehensweise der Einstellungen ist nach den Standards von TRINAMIC abgearbeitet worden. Nachzuvollziehen im Datenblatt [13].

4.2. Testablauf

Die *TMCL-IDE*-Software ist ein Programm, mit dem das Modul über eine Schnittstelle mit dem PC verbunden ist. Hier über eine [USB-Verbindung](#), die die Vorgabewerte direkt an den Motorcontroller übergibt. Um zu dem Positionsmodus zu gelangen, sind vorerst bestimmte Abläufe und die damit verknüpften Parametereingaben chronologisch abzuarbeiten. Zu Beginn sind alle Einstellungen wie im Kapitel 4.1.2 *Experimentelle Reglerparameterbestimmung* erwähnt, das Lageerfassungssystem sowie die Auflösung einzugeben. Dann erfolgt der Wechsel in den Geschwindigkeitsmodus, indem über eine Geschwindigkeitsvorgabe der Motor anzusteuern und nach dem beschleunigen zu stoppen ist. Einmalig nötiger Vorgang nach einschalten des Moduls, damit die Encoder-Initialisierung durchgeführt wird. Die optimale sinoide Positionierung setzt eine Startgeschwindigkeit von $v_{Ramp} = 0$ voraus. Somit sind die im Kapitel 2.3.2 *Sinoideninterpolation* vereinbarten Formeln 2.15 bis 2.18 gültig.

Allgemein sind die maximale Beschleunigung a_{max} und die maximale Geschwindigkeit v_{max} während eines Beschleunigungsvorgangs nicht zu ändern. Falls eine Änderung doch erfolgt, sorgt das während der Beschleunigungsphase für eine andere Skalierung der Translationsdefinitionen in der Regelung. Eine sinoide Geschwindigkeitsverteilung ist somit nicht gewährleistet. Ein weiterer wichtiger Punkt ist das Verhältnis zwischen der Beschleunigung und der Geschwindigkeit. Ist die maximale Beschleunigung gegenüber der maximalen Geschwindigkeit um ein vielfaches größer, ist der Beschleunigungsvorgang sehr schnell abgeschlossen und die Regelung gibt nur sehr wenige Geschwindigkeitszwischenwerte vor. Der Motor wird demnach abrupt angesteuert. Deshalb sollte das maximale Verhältnis nicht größer als $a_{max}/v_{max} = 5$ ausfallen.

Zusammenfassend sollte beim Fahren des Motors auf eine Parameteränderung verzichtet und alle Parameter im Rahmen der umsetzbaren physikalischen Eigenschaften dimensioniert werden.

4.3. Vergleich der Profile

Um einen direkten Vergleich der Messwerte aus der Simulation mit denen aus der Evaluierung darzustellen, verlaufen die Zeitbereiche der beiden Interpolationsarten nahezu identisch. So lassen sich am besten Qualitätsunterschiede sichtbar machen. In der Simulation steht ein Zahlenwert z. B. 5 für fünf Umdrehungen eines Motors. Da in der praktischen Umsetzung eine Auflösung von nur einem Signal bei der Positionierung pro Umdrehung nicht ausreichen würde, wird auf die Verwendung eines Encoders gesetzt. Dieser hat eine Auflösung von 4000 Inkrementen pro Umdrehung (siehe Kapitel: 2.1.3 *Der Encoder*). Aufgrund identischer Skalierung der Zeiten für die jeweiligen Interpolationarten, müssen die Zielpositionen separat berechnet werden. Folgende Berechnungen stellen den Zusammenhang der Zielposition und der Interpolationart dar.

Für die Trapezinterpolation:

$$\text{target position} = \frac{v_{max} \cdot \text{encoder steps}}{60} \cdot \left(\frac{v_{max}}{a_{max}} + (t_v - t_b) \right) \quad (4.1)$$

Mit $v_{max} = 2000 \text{ rpm}$, $a_{max} = 2000 \text{ rpm/sek}$, $\text{encoder steps} = 4000$, $t_b = 1 \text{ sek}$ und $t_v = 7.5 \text{ sek}$ ergibt sich eine Position von 1000000 Inkrementen.

Für die Sinoideninterpolation:

$$\text{target position} = \frac{v_{max} \cdot \text{encoder steps}}{60} \cdot \left(\frac{2 \cdot v_{max}}{a_{max}} + (t_v - t_b) \right) \quad (4.2)$$

Mit $v_{max} = 2000 \text{ rpm}$, $a_{max} = 2000 \text{ rpm/sek}$, $\text{encoder steps} = 4000$, $t_b = 2 \text{ sek}$ und $t_v = 7.5 \text{ sek}$ errechnet sich ebenfalls ein Weg von 1000000 Inkrementen.

Es ergeben sich zwei identisch anzufahrende Positionen, welche im Folgenden zuerst mit der Trapezinterpolation und anschließend mit der Sinoideninterpolation zu vergleichen sind. Direkt von dem Eval-Board ausgegebene Messwerte, als CSV-Datei (Tabelle), werden von *Scilab* eingelesen und anschließend grafisch dargestellt. Zusätzlich ist die Messwertaufnahme mit kleinen Schwankungen behaftet, da sich die physikalischen Größen intern über Sensoren erfassen lassen. Dies ist bei der Darstellung der Geschwindigkeitsprofile und der Strommessung sehr stark zu beobachten. Dabei müssen die Werte der Strommessung gesondert numerisch behandelt werden, die anschließend den Zusammenhang von Beschleunigung und Phasenströme des Motors klar machen.

4.3.1. Trapezinterpolation

Aus den gewonnenen Daten der Positionierung mit der Trapezinterpolation ist das Ziel bei 1000000 Inkrementen (entspricht 250 Umdrehungen des Motors) in einer Zeit von $t_e = 8.55$ sek erreicht. Dies entspricht auch dem erwarteten Zeitbereich und liefert das in Abbildung 4.2 dargestellte Ergebnis.

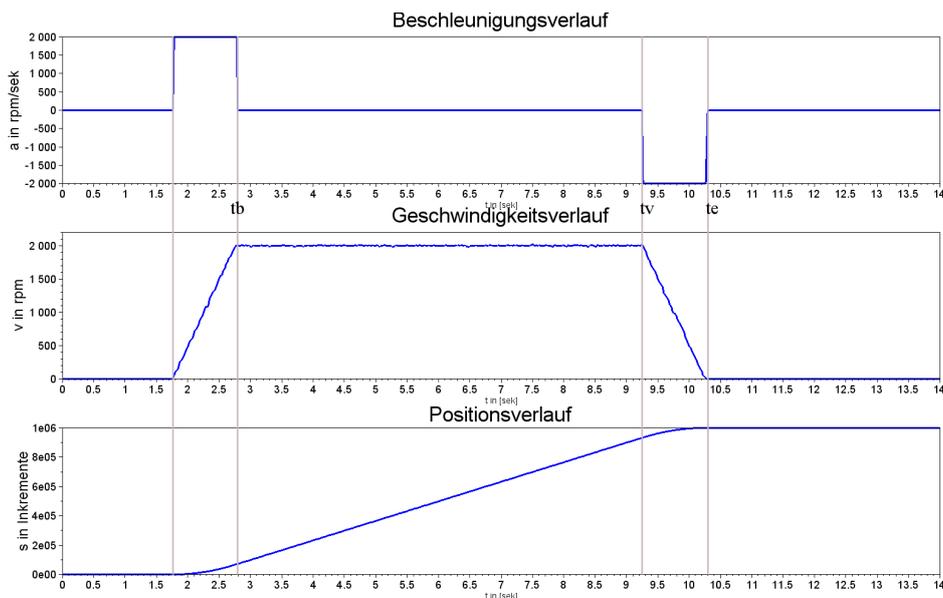


Abbildung 4.2.: Evaluierung der Trapezinterpolation

Tabelle 4.2 zeigt die einzelnen Zeitabschnitte der theoretischen und der gemessenen Zeiten und erbringt ein sehr genaues Ergebnis.

Tabelle 4.2.: Vergleich der Zeiten der Trapezinterpolation

Zeiten	theoretische	gemessene
t_b in sek	1	1.05
t_v in sek	7.5	7.5
t_e in sek	8.5	8.55

Die kleinen Zeitdifferenzen zwischen den aufgenommenen und den theoretischen Messwerten sind auf Ables- und Quantisierungsfehler sowie auf Verzögerungen durch die [USB-Schnittstelle](#) zurückzuführen und befinden sich in einem akzeptablen Bereich.

Die negative Eigenschaft des starken Rucks bleibt bei dieser Interpolation allerdings bestehen und wird im Folgenden durch die Weiterentwicklung, die Sinoideninterpolation, reduziert.

4.3.2. Sinoideninterpolation

Begründet aus der Idee der ruckfreien Ansteuerung eines Motors, erfolgt die Weiterentwicklung der Trapezinterpolation, die Sinoideninterpolation. Durch den verteilten Beschleunigungsverlauf erreicht der Motor ein ruckbegrenztes Anfahren (Ruckverlauf siehe Kapitel 2.4). Aus den theoretischen Vorüberlegungen wurde im ersten Schritt eine Simulation erstellt, welche eine Überprüfung der Umsetzbarkeit liefert. Darauf aufbauend wird der simulierte Algorithmus auf das Modul *TMC603-Eval* programmiert und liefert schließlich alle relevanten Messdaten, die für die qualitative Bewertung notwendig sind. Die aus den Messdaten gewonnene Abbildung 4.3 zeigt alle markanten Zeitverläufe.

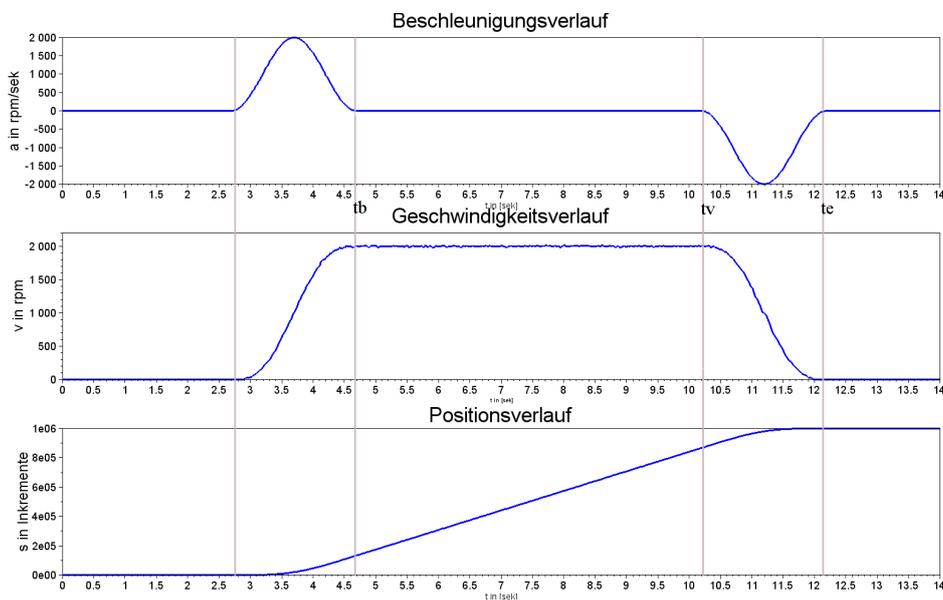


Abbildung 4.3.: Evaluierung der Sinoidinterpolation

Der Vergleich von theoretischen und gemessenen Zeiten der Sinoideninterpolation liefert ein genaues Ergebnis. Tabelle 4.3 zeigt den Zusammenhang der gemessenen Zeiten:

Tabelle 4.3.: Vergleich der Zeiten der Sinoideninterpolation

Zeiten	theoretische	gemessene
t_b in sek	2	1.95
t_v in sek	7.5	7.5
t_e in sek	9.5	9.4

Alle Zeiten liegen im erwarteten Messbereich. Die Abweichungen sind auf Ablese- und Quantisierungsfehler zurückzuführen und befinden sich in einem akzeptablen Bereich.

Auch hier wird die Position von 1000000 Inkrementen (250 Umdrehungen des Motors) in der Zeit $t_e \approx 9.5$ sek erreicht. Im Vergleich zu der Trapezinterpolation ist der Positionsverlauf im Beschleunigungs- und Bremsverlauf weiter gestreckt und jeweils eine Sekunde länger. Dadurch ist der Konstantfahrbetrieb um eine Sekunde kürzer. Der sinouide Beschleunigungsverlauf sorgt für einen sachteren Geschwindigkeitsverlauf.

Zusammenfassend ist diese Interpolationsart ebenfalls sehr genau, mit der einzigen Einschränkung, dass bei identischen Translationsbedingungen die Sinoideninterpolation mehr Zeit benötigt, um die Zielposition zu erreichen. Eine Erhöhung der Beschleunigung verkürzt die Zeit zum Erreichen der Position und ist zusätzlich ruckbegrenzter als die Trapezinterpolation. Ein kompromissloser Wechsel der Interpolationsart bei der Positionierung ist möglich.

4.3.3. Zusammenhang von Beschleunigung und Strom

Der Zusammenhang der Beschleunigung und des Stroms wurde in den beiden vorherigen Abbildungen 4.2 und 4.3 nicht vertieft. Um einen klar sichtbaren Vergleich der Ströme während der Beschleunigungs- und Bremsphase deutlich zu machen, sind die Ströme während des Beschleunigungssegments in Abbildung 4.4 und 4.5 dargestellt.

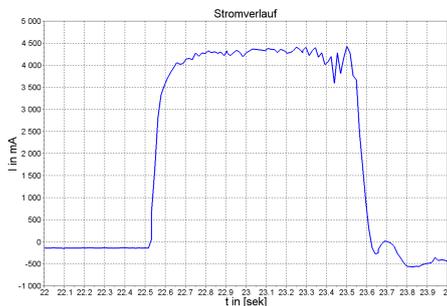


Abbildung 4.4.: Strom/Beschleunigung bei Trapezinterpolation

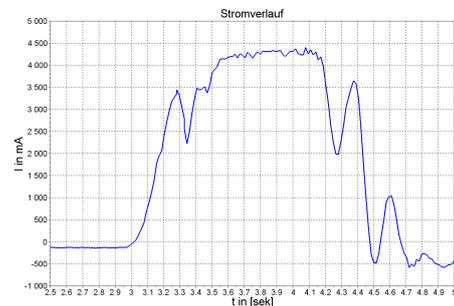


Abbildung 4.5.: Strom/Beschleunigung bei Sinoideninterpolation

Die Ströme wurden messtechnisch über einen *Shunt*-Widerstand erfasst. Der dargestellte Stromverlauf setzt sich aus dem Betrag der drei Phasenströme zusammen:

$$I_{abs} = \sqrt{I_a^2 + I_b^2 + I_c^2} \quad (4.3)$$

Das Verfahren der Strommessung ist sehr ungenau, da die aufkommenden Skalierungs- und Quantisierungsfehler die Messwerte stark verauschen.

Die Stromverläufe kommen, trotz der Verauschung, den jeweiligen theoretischen Interpolationsarten sehr nahe. Der Stromverlauf des Trapezprofils hat eine Zeitkonstante $t_b \approx 1$ sek. Die Steigung zum Start der Positionierung liegt nahezu bei Eins. Mit wenigen Abweichungen folgt der Strom den theoretischen Vorgaben. Der Stromverlauf der Sinoideninterpolation hat eine Zeitkonstante $t_b \approx 2$ sek während der positiven Beschleunigungsphase. Die zeitliche Streckung der Beschleunigung stimmt somit überein. Jedoch weicht der Verlauf von den theoretischen Erwartungen ab. Dies ist auf Quantisierungsfehler und auf den Phasenstrom, welcher im oberen Bereich durch das *Eval-Board* begrenzt wird, zurückzuführen.

Die theoretischen Zeitkonstanten des Beschleunigungssegments stimmen mit den gemessenen überein. Die variable Beschleunigungsverteilung bewirkt eine langsam ansteigende Energieabgabe an das mechanische System (siehe Ruckverteilung in Kapitel 2.4). Bei Anlegen einer konstanten Beschleunigung tritt eine direkte Energieabgabe an den Motor ein, die sich durch einen starken, ruckhaften Start bemerkbar macht. Dies ist auch durch einen größeren Einschaltstrom oder Startstrom ersichtlich. Durch die Weiterentwicklung der Trapezinterpolation zu der Sinoideninterpolation ist der abrupte Ruck nahezu verschwunden, was auch die Tests an verschiedenen **BLDC**- und **PMSM**-Motoren zeigten.

Den Ruckunterschied zwischen beiden Interpolationsarten kann man mit folgendem Versuch sehr gut deutlich machen: Ein Motor wird freischwingend an einem Seil, das ca. 50 cm lang ist, an einem Punkt befestigt und mit den beiden unterschiedlichen Interpolationsarten nacheinander angesteuert. Der Aufbau stellt ein sehr sensibles und schwingungsfähiges System dar, das den Ruck gut sichtbar für das menschliche Auge macht. Mit den gesammelten Erfahrungen aus diesem Versuchsaufbau ist ein ruckbegrenztes Hochfahren eines Motors möglich und wurde erfolgreich auf dem Modul *TMC603-Eval* umgesetzt.

4.4. Validierung

Die konkrete Validierung, wie sich der Wechsel von der Trapez- zur Sinoideninterpolation äußert, hat in den vorangegangenen Kapitel noch nicht stattgefunden. Um einen genauen und qualitativen Unterschied sichtbar zu machen, wurden im Folgenden die Drehmomentkennlinien während der Positionierung aufgenommen. Die Messung der mechanischen Kraftübertragung gibt an, was in Wirklichkeit an eine anzutreibende Last übergeben wird und wie sich dabei die einzelnen Interpolationsarten verhalten. Der Weg der zum ruckbegrenzten Anfahren eines Motors benötigt wird, wurde ausführlich angesprochen und beschrieben. Diesen heißt es nun über die Messung des Drehmoments zu bestätigen.

Der mechanische Aufbau zum Messen des Drehmoments stellte TRINAMIC zur Verfügung. Der Drehmomentsensor hat einen Messbereich von 4 Nm. Der in den Simulationen verwendete Motor (A.3) hat ein sehr kleines Nennmoment von 62.5 mNm. Dies würde als Messrauschen interpretiert und der Motor kann daher hier keinen Einsatz mehr finden.

Durch den Wechsel des Motors zu einem mit gleicher Bauform aber mit höherer Leistung, findet eine größere Kraftübertragung auf die Welle statt. Somit ist das Drehmoment größer gegenüber dem Motor mit kleinerer Leistung und wird nicht mehr als Messrauschen wahrgenommen. Der neue Motor hat ein Nennmoment von 250 mNm (siehe Datenblatt A.4). Der angeschlossene Encoder hat nun eine Auflösung von 2000 Inkrementen pro Umdrehung. Für den präzisen Vergleich sind alle Translationsparameter so angepasst worden, dass die Position von 1000000 (500 Umdrehungen) erreicht wird. Bei der Trapezinterpolation ist eine Beschleunigung $a_{max} = 500$ rpm/sek bei einer maximalen Geschwindigkeit von $v_{max} = 2000$ rpm einzustellen. Die Formel zu der Beschleunigungs- und Bremszeit bei der Sinoideninterpolation ist fest mit der doppelten maximalen Geschwindigkeit dividiert durch die maximale Beschleunigung definiert (siehe Formel 2.16). Damit das Ziel mit den jeweiligen Interpolationsarten zum gleichen Zeitpunkt erreicht wird, muss $a_{max} = 1000$ rpm/sek bei $v_{max} = 2000$ rpm eingestellt werden. Bei den genannten Parametereinstellungen ergibt sich folgender Drehmomentverlauf:

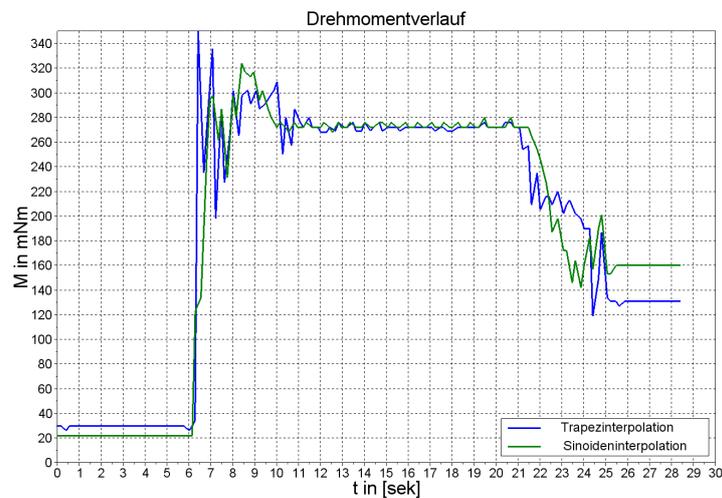


Abbildung 4.6.: Drehmomentverlauf bei unterschiedlichen Interpolationsarten

Die Drehmomentverläufe der beiden Interpolationsarten weisen eine ähnliche Verteilung auf. Für die analytische Bewertung werden die Verläufe zu den drei markanten Zeitpunkten näher beschrieben.

In der Beschleunigungsphase macht sich bei dem Trapezprofil ein schneller Anstieg zum konstanten Drehmoment bemerkbar. Hingegen ist der Anstieg bei der Sinoideninterpolation etwas langsamer und zeichnet sich mit weniger Überschwingen aus. Die Auswirkung des Rucks übt sich durch Drehmomentwelligkeiten aus, welche bei dem Trapezprofil besonders stark zu beobachten sind.

Mit Hilfe der neuen Interpolation wird das System weniger zum Schwingen angeregt und reduziert die Drehmomentwelligkeit. Zusammen mit der Wahl eines **PMSM** mit Sinuskommütierung und des quadrierten sinoiden Beschleunigungsverlaufs, wird eine sanftere Positionierung erreicht.

Während der konstanten Fahrt, bedeutet die Beschleunigung ist gleich Null, stellt sich ein konstantes Drehmoment ein. Die kleinen Welligkeiten sind auch hier auf Messungenauigkeiten und auf eine leichte Unwucht im Versuchsaufbau zurückzuführen. Das Drehmoment in der Bremsphase verhält sich bei beiden Interpolationen nahezu identisch. Der mechanische Versuchsaufbau, u. a. die Welle, hat gegenüber dem Motor eine größere Masse. Diese verhält sich dem Motor entsprechend träge und verursacht dadurch einen langsameren Abfall des Drehmoments. Gegen Ende der Positionierung bleibt ein konstantes Drehmoment erhalten. Der Positionsregler gibt irgendwann eine Position von Null vor, womit der P-Anteil des Geschwindigkeitsreglers auch den Wert Null hat. Dessen I-Anteil ist nur gerade so groß, um die Geschwindigkeit auf Null zu halten und verursacht so ein konstant anhaltendes Drehmoment. Ebenfalls würde eine Positionsdifferenz den Geschwindigkeitsregler, dessen I-Anteil und die Zielvorgabe an dem Stromregler eine Kraftänderung auf der Welle bewirken.

4.5. Praktische Beispiele

In der Industrie findet das ruckbegrenzte Ansteuern eines Motors einen sehr großen und vielfältigen Anwendung. Die Sinoideninterpolation kann dabei in den verschiedensten Bereichen eingesetzt werden. Die Pharmaindustrie transportiert beispielsweise auf Fließbändern komplett gefüllte Behälter, die unter keinen Umständen überschwappen dürfen. Bei einer bestehenden Anlage mit einer Steuereinheit muss lediglich der Rampengenerator modifiziert werden, der die neue Funktion des ruckbegrenzten Anfahrens umsetzen kann. Ein weiteres Einsatzgebiet ist der Surface-Mounted Device (**SMD**)-Bestückungsautomat. Dieser ist ein Automat, der Platinen mit den jeweiligen, geforderten Bauteilen, welche sich auf einer Rolle befinden, bestückt. Der Rolle wird ein Bauteil entnommen und anschließend auf der Platine platziert. Damit dieser Prozess möglichst präzise und nahezu ruckfrei durchgeführt werden kann, findet die Sinoideninterpolation hier Einsatz.

5. Zusammenfassung und Ausblick

Das Ziel dieser Bachelorarbeit war die Erweiterung einer Positionsregelung für **BLDC**- und **PMSM**-Motoren, mit dem Schwerpunkt einer ruckbegrenzten Beschleunigung. Zu Beginn wurden alle nötigen Grundlagen, wie Motortypen, Sensorik, Regelungsverfahren und Interpolationen vorgestellt. Die Regelung eines einfachen Motormodells wurde mit einem Kaskadenregler, bestehend aus einem Strom-, Geschwindigkeits- und Positionsregler simuliert. Eine sinoide Beschleunigungsform machte ein ruckbegrenzt Positionieren möglich und konnte anschließend auf einer realen Hardware implementiert werden. Als Referenz diente das Trapezprofil was im mathematischen Vergleich zum sinoidem Profil stand. Mit Hilfe einer ausführlichen Evaluierung, bestehend aus einer Betrachtung der Stromverläufe und Auswertung der Drehmomentkennlinien für beide Interpolationsarten, konnte das ruckbegrenzte Anfahren bewiesen werden.

5.1. Zusammenfassung

Ein dynamisches System wie der Motor, kann durch äußere Einflüsse z.B. durch eine zu große Last, sehr schnell unerwünschtes Verhalten zeigen. Dabei spielt die zu übertragene Kraft oder das Drehmoment des Motors eine große Rolle. Die Kraft oder das Drehmoment zeigen einen proportionalen Zusammenhang zum fließenden Strom in den Motorphasen. Eine abrupte Erhöhung des Stroms führt somit unausweichlich zu einem starken Ruck. Dieser auftretende Ruck lässt sich nur mit einem bestimmten Verfahren begrenzen, indem der Beschleunigungsverlauf eine spezielle Form bekommt.

In der Arbeit wurden zwei verschiedene Profile betrachtet: Zum einen der trapezförmige Geschwindigkeits- und zum anderen der sinoide Beschleunigungsverlauf. Bei der Trapezinterpolation hat dieser während der Start- und Bremsphase eine konstante Beschleunigung, so dass der Ruck sofort ungehindert seine gesamte Energie an das mechanische System überträgt. Aus diesem Ansatz wurde die Sinoideninterpolation entwickelt, welche den Ruck in begrenzter Form an einen Motor abgibt. Mit dem mathematisch definierten Umfeld wurde eine Simulation der Motorregelung (Kaskadenregelung) mit integriertem Geschwindigkeitsgenerator erstellt. Die theoretischen und simulierten Erwartungen deckten sich zu fast 100 Prozent und stellten den Weg für die anschließende Implementierung des Algorithmus bereit.

Der Vergleich der beiden unterschiedlichen Interpolationsarten machte einen qualitativen Unterschied, bezogen auf den Ruck, erkennbar. Bei identischen Translationsbedingungen hat die neu entwickelte Sinoideninterpolation eine längere Zeit zum Positionieren benötigt. Mit einer Verdopplung der Beschleunigung gegenüber der Trapezinterpolation kann die Zeit eingehalten werden, ohne die negative Eigenschaft des unbegrenzten Rucks zu erlangen. Dies ließ sich auch aus den praktischen Versuchen an einem Motor deutlich machen.

Zusammenfassend ist die Sinoideninterpolation die bessere Wahl und zeigt bei sehr sensiblen mechanischen Systemen besseres Positionierungsverhalten. Auch sichtbar anhand der aufgenommenen Drehmomentkennlinien. Für die Industrie spielt die neu entwickelte Interpolation, speziell für präzises Arbeiten eine große Rolle. Diese lässt sich mit einfachen Mitteln, wie z. B. der Anpassung der Geschwindigkeitssollwerte realisieren, ohne zusätzliche Kosten zu verursachen. In der heutigen Zeit spielen die Energieeffizienz und der wirtschaftliche Faktor eine stetig wachsende Rolle. Mit Hilfe der neuen Interpolation, lassen sich Produktionsabschnitte z. B. in der Automatisierungstechnik besser optimieren und der Ausschuss wird aufgrund der genaueren Positionierung reduziert.

Abschließend konnte im Rahmen der Bachelorarbeit die Sinoideninterpolation erfolgreich umgesetzt und ein Motor ruckbegrenzt zu einer beliebigen Position angesteuert werden. Dabei hat die Sinoideninterpolation einen großen Vorteil gegenüber der Trapezinterpolation und bereichert die Produktpalette von TRINAMIC.

5.2. Ausblick

Die Regelungstechnik bedient sich Positionssteuerungen, die der Industrie und uns im alltäglichen Leben begegnen. Die gesamte Ausarbeitung hat sich u. a. mit dem Ruck beschäftigt, der die eigentliche physikalische Hürde darstellt. Nach den theoretischen Überlegungen und aus dem Vergleich der beiden Interpolationsarten, lässt sich der Ruck nicht komplett über die Wahl der Antriebstechnik reduzieren. Die Energie, die zum Beschleunigen und Bremsen aufgebracht werden muss, kann nicht einfach über andere physikalische Komponenten reduziert werden. Bei der Trapezinterpolation wurde die komplette Energie in Form des Rucks, beim Anfahren und Erreichen der Geschwindigkeit sofort an das mechanische System übergeben. Die Weiterentwicklung (Sinoideninterpolation) ging den Schritt, den Ruck zeitlich während einer Beschleunigungs- und Bremsphase zu strecken und so die Energie in begrenzter Weise an das mechanische System zu übertragen. Mit anderen Beschleunigungsformen kann diese zeitliche Abgabe gestreckt oder gestaucht werden.

Mögliche Beschleunigungsformen sind die Sägezahnverteilung, die Kombination aus Sinoidenverteilung und dem Sägezahnverlauf, die S-Rampe und der trapezförmige Beschleunigungsverlauf. Die Sägezahnverteilung verfügt bei geringer Drehzahl über ein großes Drehmoment, im höheren Drehzahlbereich stellt sich geringerer Drehmomentbedarf ein [20]. Im ersten Moment ist die Beschleunigung am Motor maximal und nimmt bis zum Erreichen der maximalen Geschwindigkeit ab. Das verursacht einen sprunghaften Beschleunigungsanstieg [20]. Die Kombination aus beiden Beschleunigungsformen (Sägezahn- und Sinoidenverlauf) verändert den Geschwindigkeitsverlauf im oberen Bereich, sodass dieser sachter, von der maximalen Geschwindigkeit ausgehend, angefahren wird. Auch hier wird der Ruck in begrenzter Form an den Motor mit einer Verschiebung der Ruckmaxima und -minima auf der Zeitachse übergeben. Bei der S-Rampe kann man von einer Weiterentwicklung der Sinoideninterpolation sprechen, da diese einen noch leichteren Übergang zu den Maximal- und Minimalpunkten im Ruckverlauf hat. Hier findet eine noch größere Reduzierung des Rucks statt. Der trapezförmige Beschleunigungsverlauf geht bei der Reduzierung des Rucks erneut einen Schritt zurück. Die zeitliche Ableitung ist begrenzt und zeitlicher gestreckt als das Rampenprofil. Allgemein sollte je nach benötigter Genauigkeit eine Auswahl auf die jeweilige Beschleunigungsform getroffen werden.

Eine weitere Reduzierung des Rucks kann sich aus der Beschleunigungsform höherer Ordnung ableiten. Damit lassen sich Motoren nahezu ruckfrei ansteuern und so die Positionierung noch genauer machen. Die genannten Interpolationen ließen sich sehr gut auf die Module von TRINAMIC umsetzen und um die oben aufgeführten Beschleunigungsverläufe erweitern. Damit könnte die Produktpalette ohne Erstellen von neuen Hardwarekomponenten erweitert werden.

Literaturverzeichnis

- [1] Trinamic Motion Control. *TMC603A*. URL <http://www.trinamic.com/products/integrated-circuits/bldc-driver/tmc603> [online; Stand 30. Juni 2014].
- [2] Leandro Guillermo Cravero. *Entwurf, Auslegung und Betriebsverhalten von dauermagneterregten bürstenlosen Motoren kleiner Leistung*. Dissertation, Fakultät für Elektrotechnik und Informationstechnik der Technische Universität Ilmenau, 2005. URL <http://d-nb.info/98059832X> [online; Stand 30. Juni 2014].
- [3] Jens Weidauer. *Elektrische Antriebstechnik*. ISBN: 978-3-89578-431-6. Siemens, 2013.
- [4] Uwe Probst. *Servoantriebe in der Automatisierungstechnik*. ISBN: 978-3-8348-0927-8. Vieweg+ Teubner, 2011.
- [5] Heiko Hembach. *Systematischer Vergleich von BLDC-Motorkonzepten mit Anwendung auf nass laufende Wasserpumpen kleiner Leistung*. Dissertation, Universität der Bundeswehr München, Fakultät für Elektrotechnik und Informationstechnik, 2007. URL <http://137.193.200.7:8081/node?id=86268> [online; Stand 30. Juni 2014].
- [6] Woo-Young Ahn, Jang-Mok Kim, and Byung-Moon Han. Sensorless control algorithm of bldc motors with current model. *Industrial Technology (ICIT), 2013 IEEE International Conference on*, 5:1648 – 1652, Februar 2013. URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6505920&queryText%3DBLDC+current+model> [online; Stand 30. Juni 2014].
- [7] Tzu-Yu Liu, Cheng-Hu Chen, Wen-Chun Chi, and Ming-Yang Cheng. Study on the current control loop of pmsm based on sinusoidal commutation. *Industrial Technology (ICIT), 2013 IEEE International Conference on*, 6:670–675, 2011. URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6147324&queryText%3Dstudy+on+the+current+Control+loop> [online; Stand 30. Juni 2014].

- [8] Prof. Dr. H.-Ch. Mertins and MSc. M. Gilbert. Hallspannung. URL https://www.fh-muenster.de/physiklabor/downloads/Pr_PhII_FK03_Halleffekt_06.pdf [online; Stand 30. Juni 2014].
- [9] Jens Richter. *Systemmodellierung eines bürstenlosen Gleichstrommotors mit Drehzahlregelung*. Diplomarbeit, HTW Berlin, 2009. URL <http://elib.dlr.de/59606/> [online; Stand 30. Juni 2014].
- [10] Prof. Dr.-Ing. Holger Lutz and Prof. Dr.-Ing. Wolfgang Wendt. *Taschenbuch der Regelungstechnik*. ISBN: 978-3-8171-1895-3. Verlag Harri Deutsch, 2012.
- [11] Rolf Isermann. *Mechatronische Systeme*. ISBN: 978-3-8348-0927-8. Springer Berlin Heidelberg, 2008.
- [12] Hans Groß, Jens Hamann, and Georg Wiegärtner. *Elektrische Vorschubantriebe in der Automstisierungstechnik*. ISBN: 3-89578-278-5. Sie, 2006.
- [13] Trinamic Motion Control. *TMC603-EvalV2.30*. URL <http://www.trinamic.com/products/integrated-circuits/evalboards/tmc603-eval> [online; Stand 30. Juni 2014].
- [14] Prof. Dr.-Ing. Vaupel. *Formelsammlung le v12*, 2013.
- [15] Prof. Dr.-Ing. Wolfgang Weber. *Industrieroboter*. ISBN: 978-3-446-41031-2. Hanser, 2009.
- [16] Martin Werner. *Signale und Systeme*. ISBN: 978-3-8348-0233-0. Vieweg+ Teubner, 2008.
- [17] Horst Kuchling. *Taschen der Physik*. ISBN: 978-3-446-42457-9. Hanser, 2011.
- [18] Nanotec. *DB42S03*. URL <http://de.nanotec.com/fileadmin/files/Datenblaetter/BLDC/DB42/DB42S03.pdf> [online; Stand 30. Juni 2014].
- [19] ST-LINK. *ST-LINK Programmer*. URL http://www.st.com/st-web-ui/static/active/en/resource/technical/document/data_brief/DM00027105.pdf [online; Stand 30. Juni 2014].
- [20] Friedrich Wilhelm Garbrecht, Hans-Jürgen Schaad, and Rolf Lehmann. *Workshop der professionellen Antriebstechnik*. ISBN: 3-7723-4332-5. Franzis Verlag GmbH, 1996.
- [21] Nanotec. *DB42C01*. URL <http://de.nanotec.com/fileadmin/files/Datenblaetter/BLDC/DB42/DB42C01.pdf> [online; Stand 02. August 2014].

A. Anhang

A.1. Messung der Gegen-EMK

Die Messung der **Gegen-EMK** stellt einen wichtigen Schritt zur mathematischen und analytischen Auswertung über das Verhalten eines Motors dar. Hierzu wurde ein Sternpunkt mit den Motorphasen mit jeweils einem Widerstand erzeugt und der Spannungsverlauf mit einem Oszilloskop gemessen. Während der Messung wird über einen zweiten Motor die Welle zum Rotieren gebracht. Die Drehzahl des externen Motor beträgt 2100 rpm. Abhängig von der Umdrehungsfrequenz des extern angeschlossenen Motors fallen Amplitude und Frequenz, aufgrund des Verhältnisses der externen und der Geschwindigkeit des gemessenen Motors, unterschiedlich aus. Zusätzlich spielt der mechanische Aufbau und die Anzahl der Polpaare eine große Rolle.

Die Abbildung [A.1](#) zeigt die Messung aus dem Modellbau mit einem **BLDC**. Der Motor kommt aus dem Modellflugzeugebereich und hat eine maximale Drehzahl von 13000 rpm. Der theoretische Fall eines trapezförmigen Spannungsverlaufes ist hier nicht eingetreten. Von einem reinen sinusförmigen Spannungsverlauf kann hier allerdings auch nicht gesprochen werden, da der Verlauf sehr verzerrt ist.

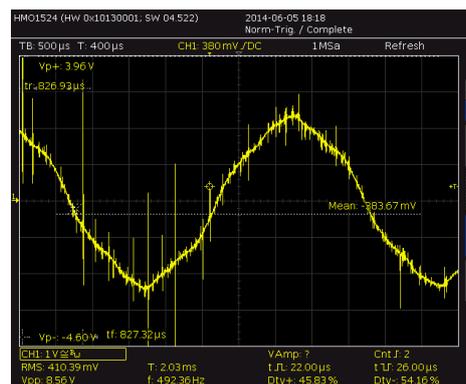


Abbildung A.1.: Gemessener Spannungsverlauf der **Gegen-EMK** eines **BLDCs**

Folgende Abbildung A.2 zeigt einen PMSM von der Firma Anaheim Automation und hat eine maximale Drehzahl von 4000 rpm mit dem erwarteten sinusförmigen Spannungsverlauf. Der Motor benötigt zur genauen Positionierung einen Encoder und sollte im Idealfall über eine Sinuskommutierung angesteuert werden.

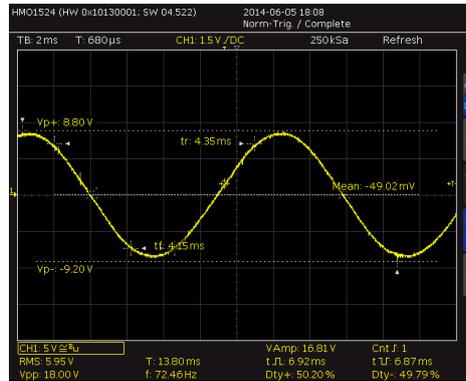


Abbildung A.2.: Gemessener Spannungsverlauf der Gegen-EMK eines PMSMs

A.2. Datenblatt des Simulationsmotors

Bei Messung der **Gegen-EMK** stellt sich bei dem ausgeschriebenen **BLDC** ein sinusförmiger Spannungsverlauf ein. Für einen guten Rundlauf ist eine Sinuskommutierung unerlässlich. Folgende Abbildung **A.3** zeigt das Datenblatt des verwendeten und simulierten Motors von der Firma Nanotec.

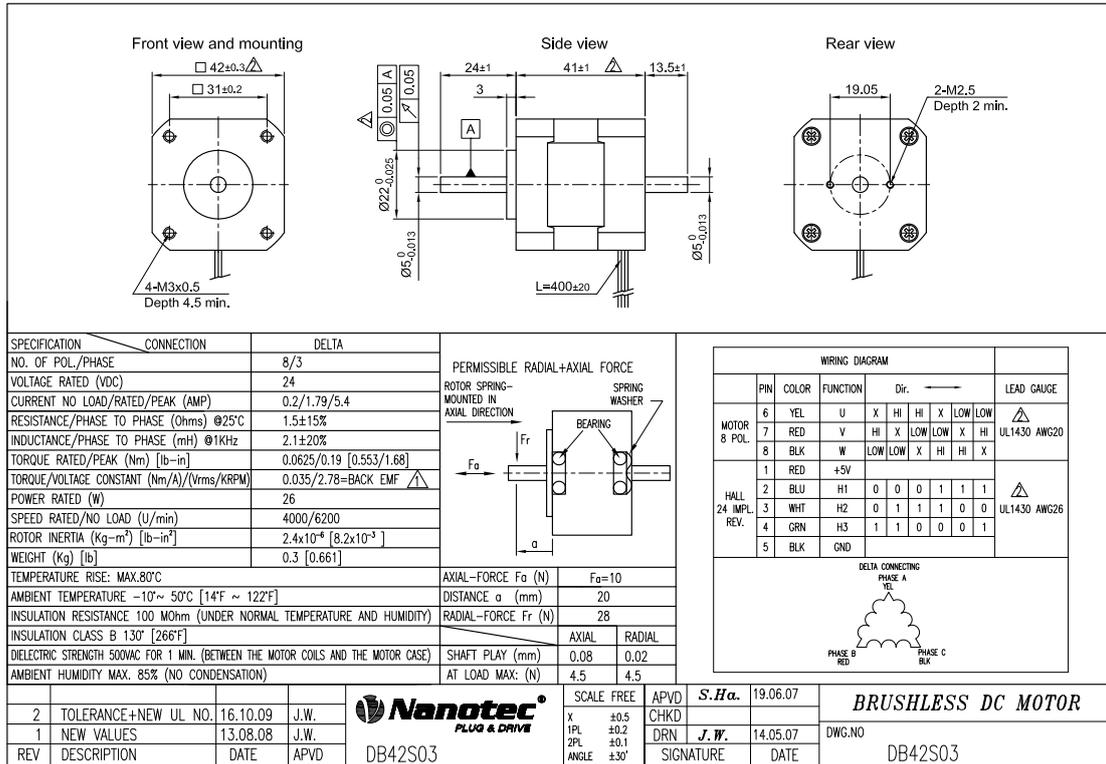


Abbildung A.3.: Datenblatt des BLDC Motors von der Firma Nanotec (Bez.: DB42S03) [18]

Alle wichtigen Informationen, welche für die Simulation und den Zusammenhang zwischen dem theoretischen und dem praktischen Betrieb eines Motors dienen, sind hier gesondert hervorgehoben.

Polpaare: der Motor verfügt über acht

Spannungsversorgung: mit 24 V Gleichstrom

Laststrom: sollte im Normalbetrieb 1,79 A nicht überschreiten

Spitzenstrom: bei Überhöhung der Grenze kann der Motors zerstört werden

Phasenwiderstand: zwischen zwei Phasen

Phaseninduktivität: zwischen zwei Phasen

Drehmoment: ausgelegt für einen kleinen Bereich

Drehmomentkonstante: dient zur Umrechnung von einem Phasenstrom zum Drehmoment

BACK EMF: drehzahlabhängige Spannung (**Gegen-EMK**), die als Störgröße auftritt

maximale Drehzahl: wird von der Regelung begrenzt

Trägheitsmoment des Motor: tritt bei der Rotation der Welle auf

A.3. Datenblatt des Testmotors

Eine Messung der Gegen-EMK liefert ebenfalls einen sinusförmigen Spannungsverlauf. Die Drehmomentmessung wurde mit dem Motor auf Abbildung A.4 durchgeführt, da dieser ein größeres Drehmoment besitzt und dadurch der Messbereich des Drehmomentsensors besser ausgenutzt werden kann.

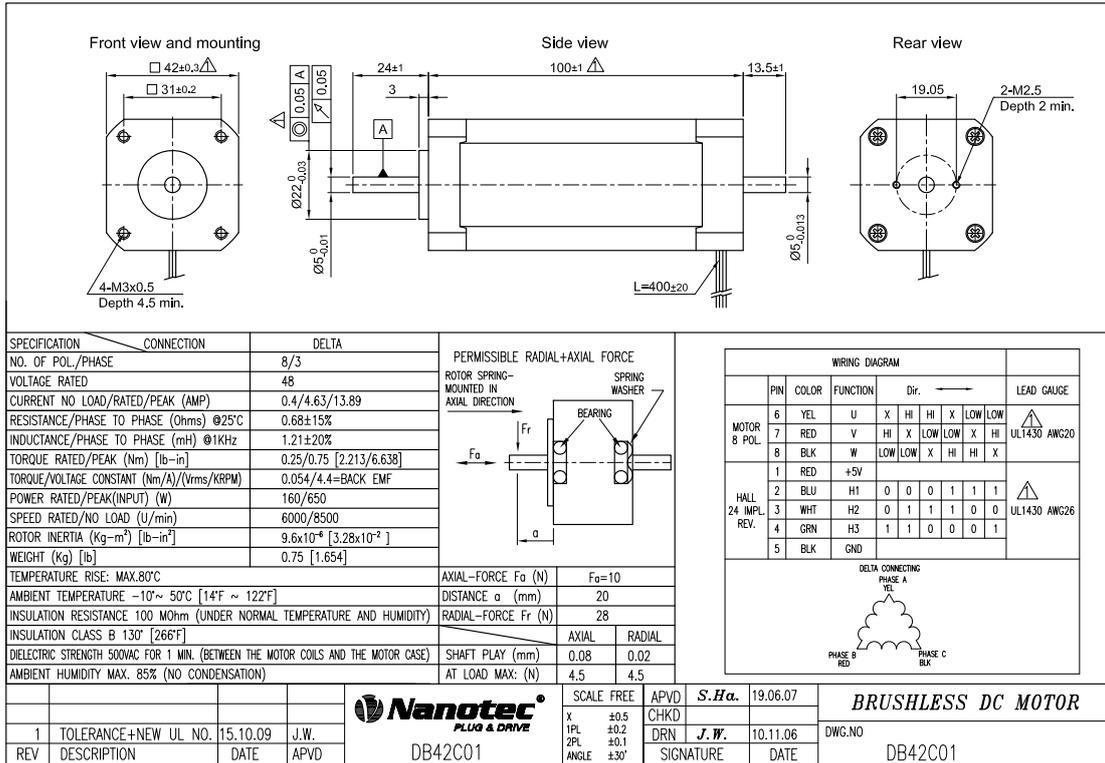


Abbildung A.4.: Datenblatt des BLDC Testmotors von der Firma Nanotec (Bez.: DB42C01) [21]

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 4. August 2014

Ort, Datum

Unterschrift