

BACHELORARBEIT

# Maya

Laser-Zielvorrichtung für ein bewegliches Ziel

ausgeführt zur Erlangung des akademischen Grades  
eines Bachelors unter der Leitung von

Univ.Ass. Dipl.-Ing. Dr.techn. Friedrich Bauer

am

**Institut für Computertechnik (E384)**  
der Technischen Universität Wien

durch

Andreas Gruber  
Matr.Nr. 1425111  
1180 Wien, Martinstraße 1 / 16

15. Januar 2017

---

## **Kurzfassung**

Die Motivation für diese Bachelorarbeit entspringt aus der Lehrveranstaltung „Mikrocomputer Labor“, aus dem Curriculum „Elektrotechnik und Informationselektronik“, bei welchen Studierende verschiedene Aufgabenstellungen mit Mikrocontrollern programmieren. Um dieses Labor interessant zu gestalten, ist es wichtig, immer wieder neue Hardware-Aufbauten für das Labor zu entwerfen.

Zu den Kriterien für den Prototyp zählt, dass er sowohl mechanisch als auch elektrisch tolerant gegenüber Fehlbedienungen sein muss, da dieser über einen längeren Zeitraum eingesetzt werden soll. Es ist natürlich wichtig, dass er für Studierende ansprechend und interessant ist und eine fächerübergreifende Aufgabe für diese darstellt.

Als Aufgabenstellung hat sich aus den gegebenen Kriterien ein beweglicher Laserdetektor und eine steuerbare Spiegelablenkeinheit, welche einen Laser positioniert, ergeben. Der Laserdetektor ist ein 4-Quadrantendetektor, welcher aus vier separaten Fotodioden zusammengesetzt ist. Der gesamte Laserdetektor ist auf einer drehbar gelagerten Scheibe montiert und wird über Schleifringe mit dem Mikrocontroller verbunden. Als Rotationsencoder fungieren zwei Lichtschranken und eine, mit einem Strichcode bedruckte, Folie .

Die Schwerpunkte dieser Arbeit liegen in der Mechanik, Elektronik und der Software, welche für diesen Aufbau notwendig ist, sowie einer Referenzlösung für das Labor.

## **Abstract**

Short version of your thesis ...

**Danksagung (optional)**

–Danke!–

# Inhaltsverzeichnis

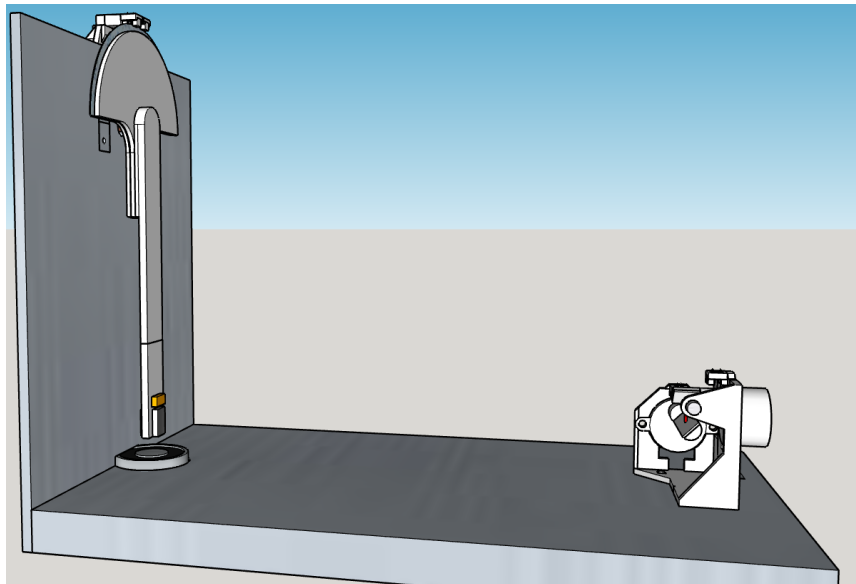
<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Aufbau</b>	<b>3</b>
2.1	Spiegelablenkeinheit . . . . .	3
2.1.1	Berechnung der Winkel . . . . .	4
2.1.2	Marken für Endpositionen . . . . .	4
2.2	Inkrementalgeber . . . . .	5
2.2.1	Funktionsweise . . . . .	5
2.2.2	Betrachtung durch endliche Automaten . . . . .	6
2.2.3	Fehlerquellen . . . . .	7
2.2.4	Nulldurchgangsdetektion mit großer Schrittmarke . . . . .	7
2.2.5	Beispielscheibe für Inkrementalgeber . . . . .	8
<b>3</b>	<b>Elektronik</b>	<b>10</b>
3.1	Schrittmotoren . . . . .	10
3.1.1	Funktionsweise . . . . .	10
3.1.2	Eingesetzte Schrittmotoren . . . . .	10
3.1.3	Ansteuerung . . . . .	11
3.1.4	H-Brücke . . . . .	12
3.1.5	Vollschrittbetrieb . . . . .	12
3.1.6	Halbschrittbetrieb . . . . .	13
3.1.7	Mikroschrittbetrieb . . . . .	13
3.2	Auswertung der Gabellichtschranken . . . . .	14
3.3	Laser-Detektion . . . . .	15
3.3.1	Transimpedanzverstärker . . . . .	15
3.3.2	Transiente Vorgänge . . . . .	16
3.3.3	Quadrantendiode . . . . .	18
3.3.4	Auswertung . . . . .	18
3.3.5	Verbesserte Verfahren . . . . .	18
<b>4</b>	<b>Software</b>	<b>20</b>
4.1	Controller . . . . .	20
4.2	Schrittmotoren . . . . .	21
4.2.1	Ablauf . . . . .	21
4.2.2	Timing . . . . .	22

4.3	Quadratur Encoder . . . . .	23
4.4	Auswertung der Fotodioden . . . . .	24
4.4.1	Ablauf . . . . .	24
4.4.2	Timing . . . . .	24
4.5	Implementierung einer Regelung . . . . .	25
4.5.1	Charakteristik einer Fotodiode . . . . .	25
4.5.2	Charakteristik aller Fotodioden . . . . .	26
4.5.3	Entwurf eines Dreipunktreglers . . . . .	26
4.5.4	Implementierung eines Dreipunktreglers . . . . .	28
<b>5</b>	<b>Zusammenfassung</b>	<b>30</b>
	<b>Wissenschaftliche Literatur</b>	<b>31</b>
	<b>Internet Referenzen</b>	<b>31</b>

# Abkürzungen

FSM	Finite State Machine, Endlicher Automat
FPU	Floatingpoint Unit, Gleitkommaeinheit
IIR	Infinity Impulse Response, Unendliche Impulsantwort
ISR	Interrupt Service Routine
OPV	Operationsverstärker
VV-OPV	Spannungsgesteuerter Operationsverstärker
PSSG	Phasenselektiven Synchrongleichrichter
PUSD	Phasenunabhängigen Synchrodemodulator
TIA	Transimpedance Amplifier, Transimpedanzverstärker

# 1 Einleitung



**Abbildung 1.1:** 3D-Modell einer Spiegelablenkeinheit und einem Pendel.

In der Lehrveranstaltung „Mikrocomputer Labor“ sollen Studierende den Umgang mit Mikrocontrollern auf spielerische Art und Weise erlernen. Diese Aufbauten beinhalten meistens verschiedene Sensoren, Aktuatoren, Eingabe- und Ausgabegeräte. Da für diese Lehrveranstaltung immer wieder neue Aufbauten mit Mikrocontrollern benötigt werden, wurde ich von einem Betreuer des Labors beauftragt, einen neuen Prototypen zu entwerfen. Die Vorgaben für diesen Prototyp sind, dass eine Laserablenkeinheit enthalten ist, welche auf ein Objekt gerichtet ist. Wichtige Eigenschaften eines solchen Laboraufbaus sind, dass er optisch ansprechend ist und einer Fehlbedienung mechanisch und elektrisch über einen langen Zeitraum standhält.

Auch an die Aufgabenstellung, welche durch den Prototypen gestellt wird, richten sich verschiedene Anforderungen. Dabei ist es wichtig, dass diese leicht erklärbar und gut vorstellbar ist. Sie darf weder über- noch unterfordern und es müssen verschiedene Teile aus verschiedenen Disziplinen vorhanden sein, um die Aufgabe interessant zu machen. So können Teilaufgaben sowohl aus den Gebieten der Mathematik, Informatik, Regelungstechnik als auch aus anderen entspringen. Einige Eigenschaften sind mit der Vorgabe der Spiegelablenkeinheit schon erfüllt, auf die weiteren gilt es besonders Acht zu geben.

Als Ziel für den zu steuernden Laserpunkt kommen sowohl feststehende Objekte als auch bewegliche infrage. Um die Dynamik zu erhöhen und somit die Aufgabe interessanter zu machen, ist die Auswahl auf ein schwingendes oder sich drehendes Objekt gefallen. Die erste Realisierung wurde dabei mit einem Pendel bewerkstelligt. Ein Beispiel für diesen Aufbau ist in Abbildung 1.1 zu sehen. Um die Zielgenauigkeit zu detektieren, besitzt das Pendel einen Detektor für den Laserstrahl. Dies macht die Aufgabe sehr interessant, da mit dieser Information auch eine Regelung und nicht nur eine Steuerung realisierbar ist.

Eine Regelung eines Lasers auf ein Ziel ist eine sehr interessante Aufgabe, vor allem deswegen, da dies kein Standardalgorithmus ist, welcher in der Literatur vollständig vorhanden ist. Um einen solchen Prototypen aufzubauen und die nötige Software zu erarbeiten, sind verschiedene Probleme zu lösen. Die Methodik, wie in dieser Arbeit auf die verschiedenen Teile eingegangen wird, gliedert sich in jeweils einen Abschnitt über den Aufbau, die Elektronik und einen abschließenden für die Software.



## 2 Aufbau

Dieses Kapitel befasst sich mit dem Aufbau, welcher alle Sensoren und Aktuatoren miteinander verbindet. Die erste Überlegung war, den beweglichen Teil als Pendel auszuführen. Diese Idee ist zwar gut, allerdings ist das Pendel durch die Reibung von den Schleifringen für die Elektronik zu stark gebremst worden. Dadurch musste mit dem Magneten, welcher das Pendel bei jedem Nulldurchgang beschleunigte, eine relativ große Kraft übertragen werden. Das Problem an dieser Lösung war, dass das Pendel nicht mehr frei schwingt, sondern deutliche Unregelmäßigkeiten beobachtet werden können. Als Lösung wurde der Entwurf mit dem Pendel überarbeitet und es wurde stattdessen eine Scheibe verwendet. Der Antrieb erfolgt durch einen Schrittmotor, dadurch kann eine vorgegebene Drehgeschwindigkeit eingestellt werden. Ein weiterer Vorteil ist, dass die vorgegebene Geschwindigkeit sehr konstant gehalten werden kann. Dieser Aufbau wird in Abbildung 2.1 dargestellt. In dieser 3D-Zeichnung kann man auch die Lichtschranken für den Drehgeber, die Schrittmotoren für die Spiegel und deren Lichtschranken erkennen.

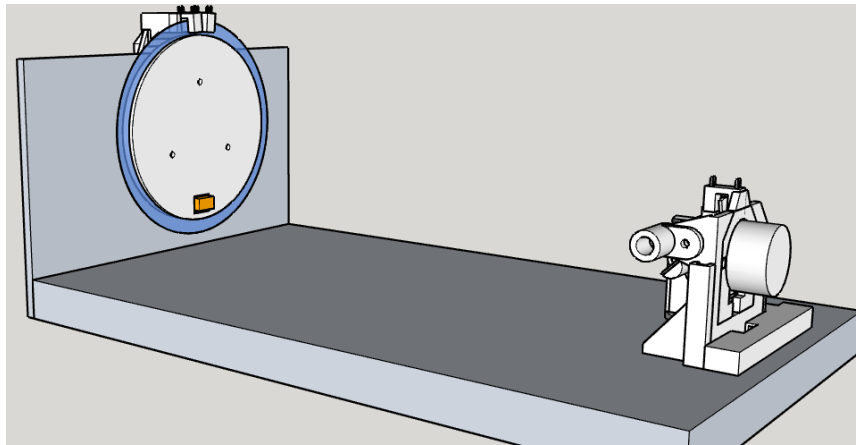
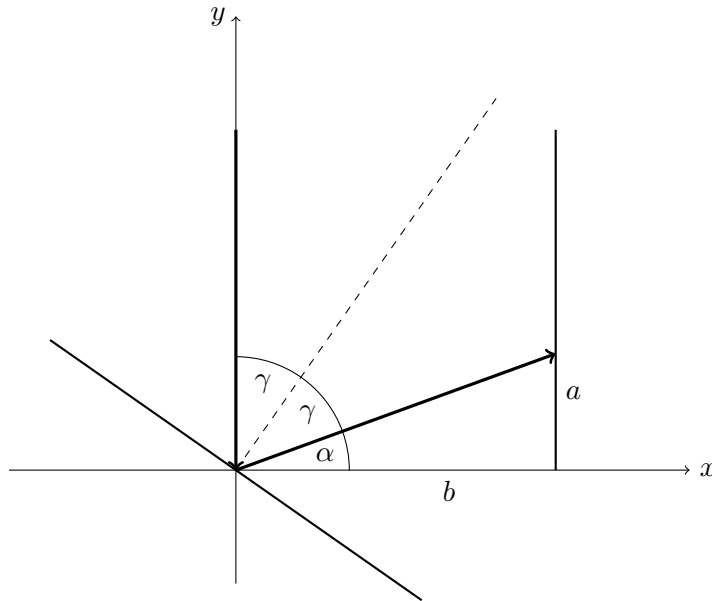


Abbildung 2.1: 3D-Modell des Prototypen.

### 2.1 Spiegelablenkeinheit

Die Berechnung der Winkel für die Spiegel wird für die Studenten, welche mit diesem Modell arbeiten werden, ein Kernpunkt. Natürlich müssen die maximalen Winkel, welche mit eingeschaltetem Laser erreicht werden dürfen, bereits vorher bekannt sein. Im folgenden wird näher darauf eingegangen.

### 2.1.1 Berechnung der Winkel



**Abbildung 2.2:** Winkel bei der Laserablenkeinheit.

In Abbildung 2.2 sind die Winkel der Spiegelablenkeinheit dargestellt. Hierbei sind die Laserstrahlen als dicke Linien und die Normale auf den Spiegel als strichlierte Linie dargestellt. Dabei bezeichnet  $\gamma$  den Winkel, welcher die Normale des Spiegels mit dem einfallenden Laserstrahl einschließt.  $\alpha$  bezeichnet den einschließenden Winkel zwischen dem Ausfallstrahl und der Grundplatte. Diese Winkel können wie folgt berechnet werden.

$$\alpha = \arctan(a/b) \quad (2.1)$$

$$\gamma = \frac{90^\circ - \alpha}{2} \quad (2.2)$$

Um zu berechnen wo der Laserstrahl auf die Wand projiziert wird, muss nur auf  $a$  umgeformt werden.

$$a = b \cdot \tan(90^\circ - 2 \cdot \gamma) \quad (2.3)$$

Diese Überlegungen gelten sowohl für die horizontale als auch die vertikale Richtung.

### 2.1.2 Marken für Endpositionen

Die Überlegungen, welche Winkel eingestellt werden müssen, sind nicht nur für die Positionierung des Lasers wichtig, sondern auch für die Berechnung der Bereichsmarken. Diese Marken geben an, ob bei dieser Winkeleinstellung der Laserstrahl gefahrlos aktiviert werden kann. Dies ist nur dann der Fall, wenn der Laserpunkt auf der Rückwand des Prototypen projiziert wird. Dazu wird im Modell eine Markierung auf den Motoren angebracht, welche bei der maximalen Auslenkung einen Lichtschranken unterbricht.

Sei  $a$  in dieser Überlegung die maximal erlaubte Auslenkung auf der Projektionsfläche, welche gefahrlos eingestellt werden darf. Zusätzlich nehmen wir an, dass die Lichtschranken entlang

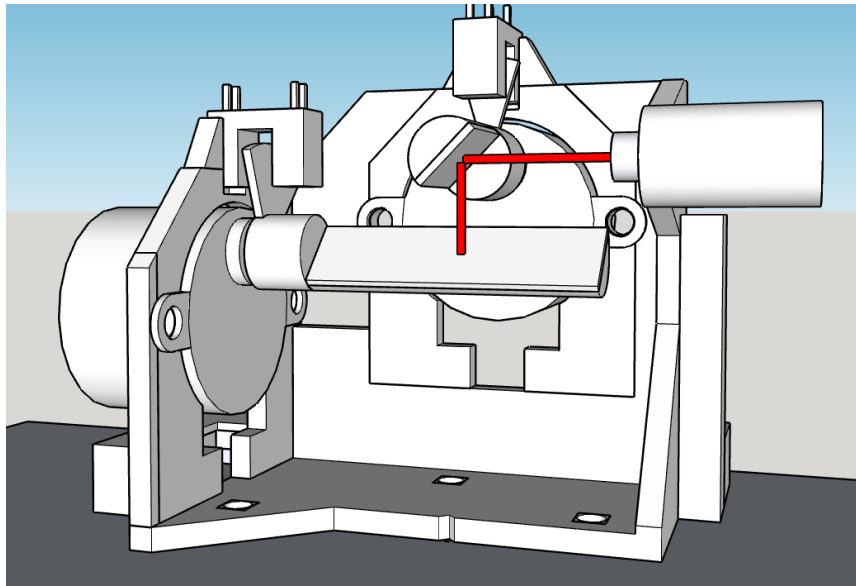


Abbildung 2.3: 3D-Modell der Spiegelablenkeinheit.

des einfallenden Lasers positioniert sind. Das bedeutet, dass der Laser abgeschaltet wird, sobald eine gedachte Marke in Abbildung 2.2 die y Achse überschreitet. Diese Marke ist winkelfest mit der strichlierten Normale und dem Spiegel verbunden. Unter diesen Voraussetzungen gibt  $\gamma$  den Winkel an, mit welchem die Marke konstruiert werden muss. Dieser Winkel reicht von dem Ende der Marke bis zur Normale des Spiegels. Das 3D-Modell der fertigen Konstruktion ist in Abbildung 2.3 ersichtlich.

## 2.2 Inkrementalgeber

### 2.2.1 Funktionsweise

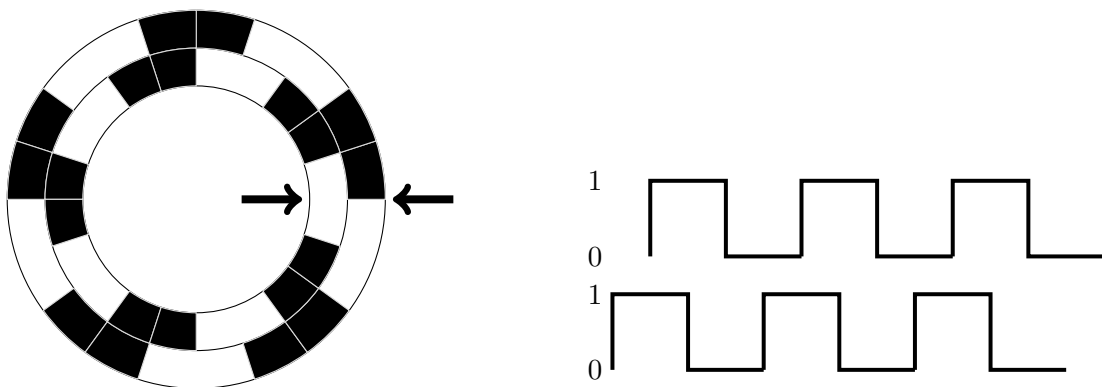


Abbildung 2.4: Funktionsweise eines Drehgebers.

Ziel eines Drehgebers ist es, die Position der Scheibe zu ermitteln. Je nach Aufgabenstellung gibt es dazu verschiedene Arten von Drehgebern. Grundsätzlich wird zwischen Absolutgebern und Inkrementalgebern unterschieden. Erstere haben auf der Scheibe, welche in  $n$  Segmente unterteilt

ist, genau  $\log_2(n)$  Spuren mit einem Binärcode. Jede Spur wird mit einem separaten Sensor ausgewertet. Dadurch kann immer ausgelesen werden, in welchem Segment sich die Leseköpfe der Sensoren befinden.

Nachteilig an einen Absolutgeber ist, dass eine hohe Anzahl an Sensoren und auch an Spuren auf der Scheibe benötigt wird. Dadurch ist er für dieses Projekt nicht von Bedeutung, da für eine annehmbare Schrittauflösung zu viele Sensoren verwendet werden müssten.

Bei einem Inkrementalgeber kann man die Anzahl der Sensoren reduzieren, indem man nur zwei Bahnen verwendet, welche je ein Rechtecksignal produzieren. Die beiden Rechtecksignale sind zueinander eine Viertelperiode versetzt. Eine andere Variante, welche noch einfacher realisierbar ist, ist nur eine Spur zu verwenden und stattdessen die Sensoren um eine Viertelperiode voneinander versetzt anzuordnen. Eine Illustration einer solchen Abnehmerkonstellation ist in Abbildung 2.4 dargestellt. [ES12, 333]

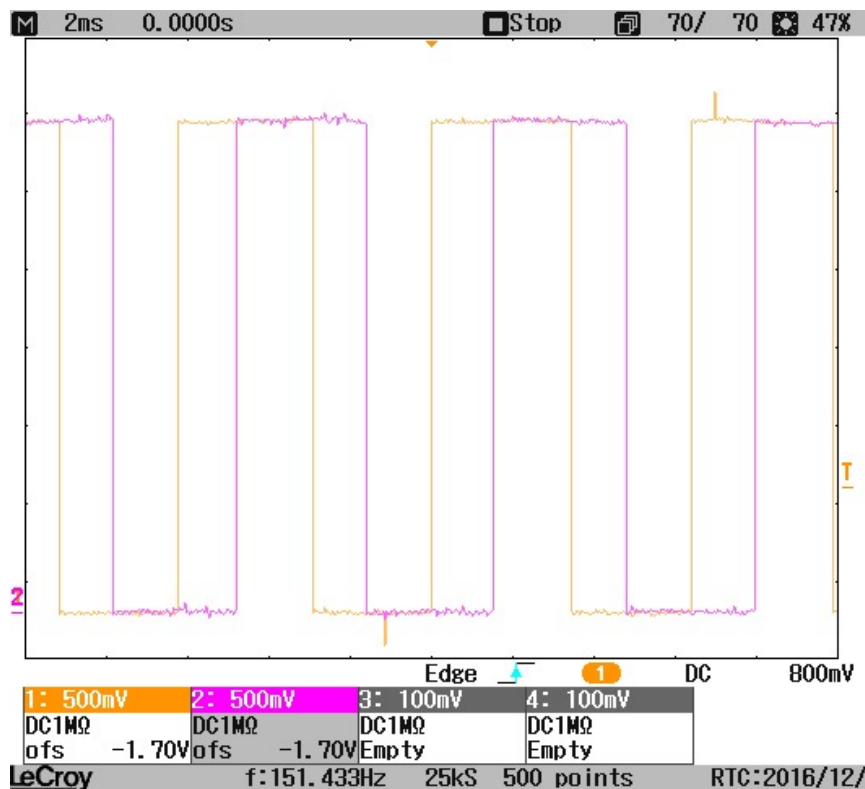


Abbildung 2.5: Aufnahme des Quadratursignals der drehbaren Scheibe.

Das Ausgangssignal hat nun die Form eines 2-Bit-Gray-Codes. Wenn nun das Signal ausgewertet wird, kann bei jeder Flanke die Richtung des Schrittes ermittelt werden. In Abbildung 2.5 ist eine Aufnahme, des von diesem Prototypen produzierten Signals, abgebildet.

## 2.2.2 Betrachtung durch endliche Automaten

Eine einfache Betrachtungsweise, welche sehr elegant in Software implementiert werden kann, ist die Darstellung durch einen endlichen Automaten, auch Zustandsmaschine oder Finite-State-Machine (FSM) genannt. Dabei gibt es vier Zustände und jeder stellt eine Kombination der

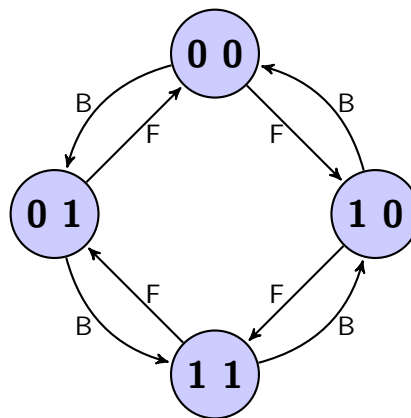


Abbildung 2.6: FSM eines Drehgebers.

beiden Eingänge dar. Ein Zustandswechsel wird durch einen logischen Wechsel eines Einganges ausgelöst und gibt als Ausgabe entweder einen Vorwärts- oder einen Rückwärtsschritt zurück. Um die Gesamtposition zu errechnen müssen nur alle Schritte vorzeichenrichtig gezählt werden.

Eine FSM, welcher den geschilderten Sachverhalt implementiert, ist in Abbildung 2.6 dargestellt. Damit die abgebildete Graphik eine gültige FSM wird, müssen noch Anfangszustand und gültige Endzustände eingezeichnet werden. Der Anfangszustand ergibt sich jedoch daraus, wie die Encoderscheibe bei der Initialisierung des Controllers steht und kann deswegen noch nicht eingezeichnet werden. Der Endzustand ist der letzte eingeleseene Zustand. [GG03, 51]

### 2.2.3 Fehlerquellen

Bei dieser Variante gibt es allerdings bestimmte Eigenschaften, welche bei der Implementierung berücksichtigt werden müssen. Da dies nur eine inkrementelle Positionierung ist, ist der Startpunkt nicht durch das Sensorsignal ermittelbar. Weiteres werden einmalige Fehler aufaddiert und können nur mit dem Rotationsencoder nicht mehr korrigiert werden. Da bei Inkrementalgebern mit hoher Schrittauflösung sehr viele Wechsel von Flanken verarbeitet werden müssen, können bei hohen Drehgeschwindigkeiten Schritte verloren gehen. Selbst eine sehr effiziente Implementierung mit Interrupts ist nicht für jeden Inkrementalgeber ausreichend. Abhilfe schafft in diesem Fall nur die Verwendung eines Mikrocontrollers mit einem eingebauten Quadraturencoder oder einer speziellen zusätzlichen Hardware.

### 2.2.4 Nulldurchgangsdetektion mit großer Schrittmarke

Um ohne zusätzliche Sensoren den Nulldurchgang detektieren zu können, können in die Encoderscheibe zusätzliche Information eingefügt werden. Eine Variante ist, dass ein einziger Puls mit dreifacher Länge eingefügt wird. Dadurch ist ein Nulldurchgang erkennbar. Allerdings verliert man dadurch die Möglichkeit, schnelle Richtungswechsel detektieren zu können. Nimmt man nun an, dass die Anzahl an Richtungswechsel sehr klein im Vergleich zu den Umdrehungen sind, kann man durch diese Zusatzinformation sowohl die Drehrichtung als auch die Nulldurchgänge erkennen. Für diesen Prototypen wurde eine Marke mit der dreifachen Länge verwendet. Dies ist gleich einem ausgefüllten Zwischenraum der Encoder-Scheibe.

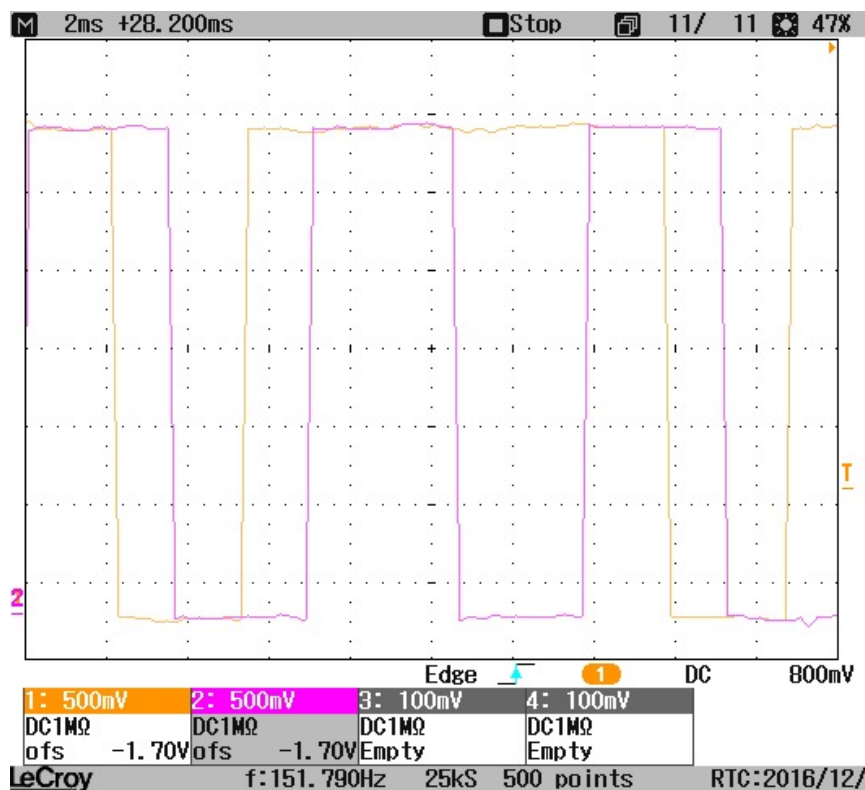


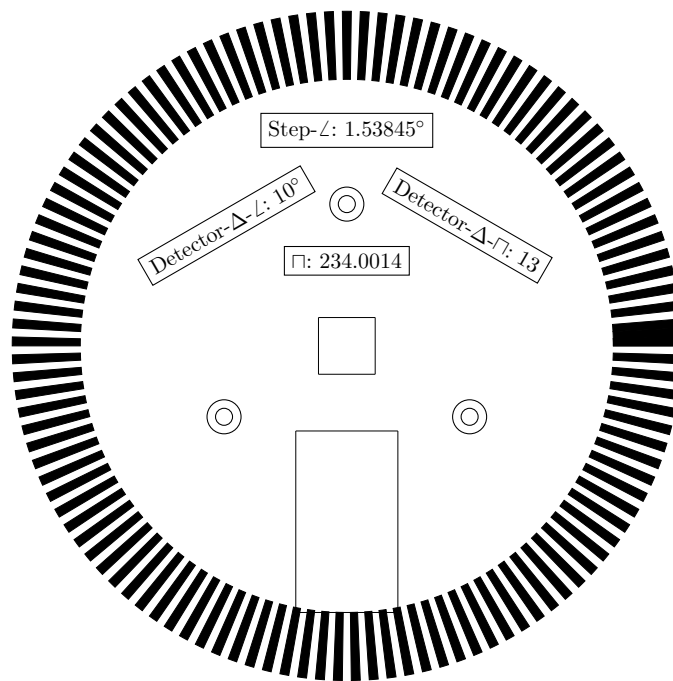
Abbildung 2.7: Aufnahme des Quadratursignals der drehbaren Scheibe.

Eine Aufnahme des Signals bei einer großen Marke ist in Abbildung 2.7 abgebildet. Wenn diese Schrittkombination mit der in Kapitel 2.2.2 erläuterten FSM abgearbeitet wird, kann die Marke dadurch erkannt werden, dass genau ein Schritt in die falsche Richtung generiert wird. Dies zu erkennen ist trivial. Wichtig ist noch anzumerken, dass diese Marke doppelt auftritt, für jeden Sensor einmal. Als Nullpunkt darf aber nur eine der beiden Stellen verwendet werden. Für eine korrekte Funktionsweise ist zusätzlich ein ausreichend großer Schrittabstand zwischen den beiden aufnehmenden Sensoren nötig.

Nun müssen allerdings die Schritte, welche durch dieses neue Muster verloren gegangen sind, durch virtuelle Schritte ersetzt werden. Bei jeder Detektion einer großen Marke wird nun ein Rückwärtsschritt statt einem Vorwärtsschritt detektiert und es gehen aufgrund der fehlenden Flanken zwei Pulse verloren. Daher müssen bei jeder der beiden Detektionen des breiten Impulses vier Impulse addiert werden. Nur bei einem der beiden darf die aktuelle Schrittzahl auf Null gesetzt werden.

### 2.2.5 Beispielscheibe für Inkrementalgeber

Die Scheibe für diesen Prototypen wurde mit Latex und Tikz gezeichnet. Dabei wurde zum Testen ein Template erstellt, welches beliebig konfiguriert werden kann. In Abbildung 2.8 ist die verwendete Scheibe dargestellt. Die Aufschriften in der Mitte charakterisieren die Scheiben und geben Informationen für die in der Software einzustellenden Parameter. Der oberste Wert ist der Winkel zwischen zwei Impulsen, danach folgen der Winkel und die Pulse zwischen den beiden



**Abbildung 2.8:** Scheibe für den Rotationsencoder.

Detektoren und der unterste Wert gibt die Anzahl der Impulse pro Umdrehung und Sensor an. Für die Darstellung in dieser Arbeit wurde die Grafik noch um den Faktor 0,75 verkleinert.

## 3 Elektronik

### 3.1 Schrittmotoren

Für diesen Prototypen kommen drei Schrittmotoren zum Einsatz. Einer der Motoren wird als Antrieb der Ziel-Scheibe verwendet und die anderen beiden als Aktuator zum präzisen Positionieren. Dadurch sind Schrittmotoren sehr interessant für diese Arbeit und werden im folgenden näher erläutert.

#### 3.1.1 Funktionsweise

Schrittmotoren sind Synchronmotoren, welche eine Umdrehung in mehrere Schritte unterteilen. Diese Schrittunterteilung wird durch Einkerbungen am Stator und am Rotor erreicht. Eine übliche Schrittweite ist, um ein Beispiel zu nennen,  $1,8^\circ$  pro Schritt. Dadurch eignen sich diese Motoren sehr gut zur sensorlosen Positionierung. Vorsicht ist jedoch bei zu hohen Drehmomenten oder zu schneller Ansteuerung geboten, da dadurch ein Schrittverlust auftreten kann, welcher sich als Fehler in der Positionierung widerspiegelt.

Schrittmotoren können wiederum in verschiedene Klassen unterteilt werden. Die hier relevanten Typen sind der Reluktanz-, der Permanentmagnet- und der Hybrid-Schrittmotor. Bei dem Permanentmagnet-Schrittmotor ist der Rotor magnetisiert. Dabei sind die Zähne abwechselnd als Nord- und Südpole ausgeführt. Bei dem Reluktanz-Schrittmotor besteht der Rotor aus einem weichmagnetischen Material. Daher muss der gesamte magnetische Fluss durch den in den Windungen fließenden Strom erzeugt werden.

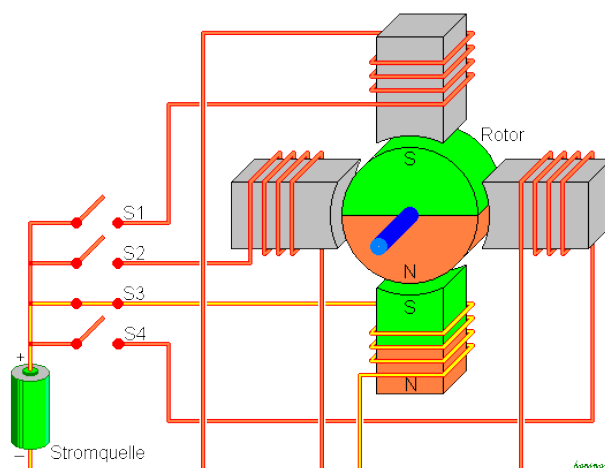
Der Hybrid-Schrittmotor ist eine Mischform, bei der versucht wird, die positiven Eigenschaften zu vereinen. Dabei wird eine höhere Genauigkeit und mehr Drehmoment erzielt. Der Rotor wird aus einem Permanentmagneten, welcher in axialer Richtung magnetisiert ist, und gezahnten Weicheisenkränzen aufgebaut. [MWa01, 28]

In Abbildung 3.1 ist der Aufbau eines Permanentmagnet-Schrittmotors mit einer Schrittweite von  $90^\circ$  dargestellt.

#### 3.1.2 Eingesetzte Schrittmotoren

In diesem Prototypen werden zwei Modelle von Schrittmotoren verwendet. Der erste Schrittmotor ist jener, welcher zum Antrieb der Scheibe mit den Fotodetektoren verwendet wird. Dieser





**Abbildung 3.1:** Aufbau eines Schrittmotors mit unipolarer Beschaltung. Quelle: Wikipedia [Wik17b]

benötigt 24 Schritte pro Umdrehung und hat somit einen Schrittwinkel von  $15^\circ$ . Bei diesem Schrittmotor ist eine Mittelpunktanzapfung der Spulen vorhanden und somit wäre ein unipolarer Betrieb möglich. Da bei diesem Schrittmotor jedoch eine, für einen Schrittmotor, hohe Geschwindigkeit erreicht werden soll, wird eine bipolare Beschaltung verwendet.

Der in der Laserablenkeinheit verwendete Schrittmotor ist ein 28BYJ-48 mit einer unipolaren Beschaltung. Dieser benötigt 64 Schritte pro Umdrehung und besitzt somit einen Schrittwinkel von  $5,625^\circ$ . Zusätzlich ist in diesem Motor ein Getriebe mit einem Übersetzungsverhältnis von 64:1 eingebaut. Nach dem Getriebe hat dieser Motor einen effektiven Schrittwinkel von  $0,0879^\circ$ . Dies ist bereits eine feine Unterteilung einer Umdrehung, welche für die Anwendung gerade ausreichend ist.

Zur Verdeutlichung wird mit Gleichung 2.3 der minimalen Abstand zweier möglicher Positionen eines Laserpunktes berechnet. Diese maximale Auflösung kann genau dann erreicht werden, wenn der Laser senkrecht zur Grundplatte ausgestrahlt wird. Dafür wird angenommen, dass die Projektionsfläche 20 cm entfernt ist.

$$a = 20 \text{ cm} \cdot (\tan(90^\circ - 2 \cdot (45^\circ + 0,0879^\circ)) - \tan(90^\circ - 2 \cdot (45^\circ))) \quad (3.1)$$

$$a = 20 \text{ cm} \cdot \tan(2 \cdot 0,0879^\circ) = 0,61 \text{ mm} \quad (3.2)$$

Mit dieser Auflösung lässt sich der Laserpunkt genau positionieren. Eine noch feinere Unterteilung einer Umdrehung würde keinen Mehrwert bringen. Der limitierende Faktor liegt, bei einer solch feinen Unterteilung, bei dem Spiel der Zahnräder und der Lagerung des Getriebes. Bei den verwendeten Motoren ist sowohl die Lagerung als auch das Getriebe aus Kunststoffteilen gefertigt. Möchte man die erzielbare Genauigkeit verbessern, dann kann man dies dadurch erreichen, dass auf ein spielarmes Metallgetriebe umgestiegen wird.

### 3.1.3 Ansteuerung

Für die Ansteuerung eines Schrittmotors sind zwei Techniken verbreitet. Diese sind die unipolare und die bipolare Beschaltung.

Bei der unipolaren Ansteuerung ist für jeder Windung eine Mittelpunktzapfung vorhanden, welche zusätzlich aus dem Motor herausgeführt wird. Dadurch ist es möglich diesen Mittelpunkt auf Versorgungsspannung zu legen und mit einem Schalter eine der beiden Hälften zu aktivieren. Der Vorteil dabei ist, dass nur ein schließender Schalter verwendet werden muss. Es ist also keine Richtungsumkehrung des Stromes notwendig. Der gravierende Nachteil ist, dass nur die Hälfte der Spule verwendet wird und somit auch nur das halbe Drehmoment generiert wird. Der in Abbildung 3.1 dargestellte Schrittmotor ist unipolar beschaltet.

Bei der bipolaren Beschaltung werden die Mittelpunkte nicht aus dem Motor herausgeführt oder nicht verwendet. Dafür muss allerdings die Stromrichtung vorgegeben werden können. Es ist somit möglich einen Schrittmotor, welcher für eine unipolare Ansteuerung vorgesehen ist, mit einer bipolaren Ansteuerung zu betreiben. Umgekehrt ist dies im allgemeinen nicht der Fall, da bei manchen Modellen die Mittelpunkte der Spulen nicht herausgeführt werden.

### 3.1.4 H-Brücke

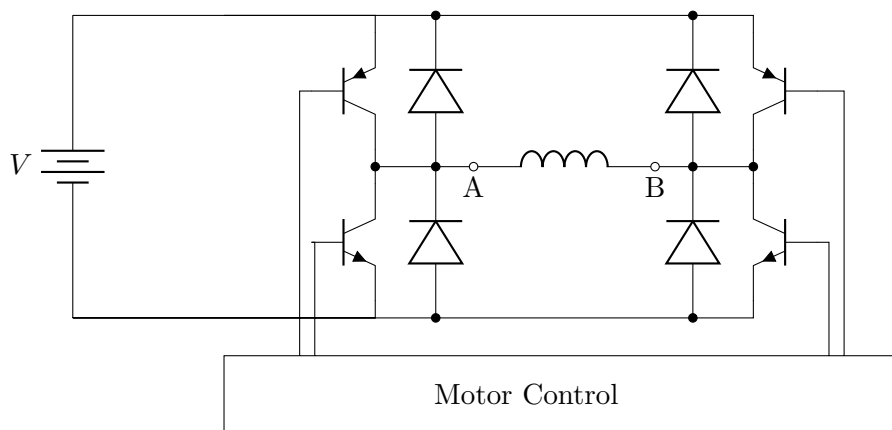


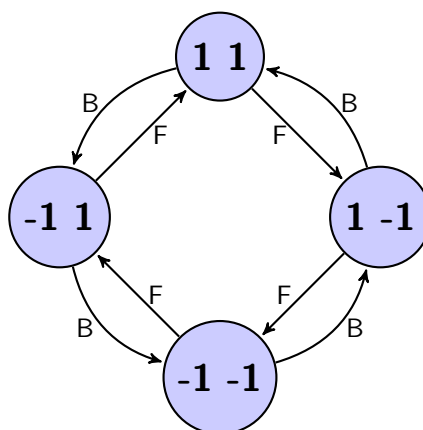
Abbildung 3.2: Aufbau einer H-Brücke für eine Windung

Um eine Windung eines Schrittmotors sowohl in Vorwärts- als auch in Rückwärtsrichtung betreiben zu können, wird eine so genannte H-Brücke benötigt. Bei dieser Schaltung werden für beide Anschlüsse der Windung sowohl ein Leistungsschalter zur Versorgung als auch einer zur Masse vorgesehen. Die Art der Leistungsschalter ist dabei unbedeutend und es können sowohl Bipolartransistoren, MOSFETs als auch andere Schalter wie Relais verwendet werden. Da es sich bei dem zu schaltenden Aktuator um eine Induktivität handelt, sind zusätzlich Freilaufdioden von der Masse und zu der Versorgungsspannung erforderlich. Die Freilaufdioden schützen die Leistungsschalter vor der in der Induktivität gespeicherten Energie.

Dieser Aufbau ist mit Bipolartransistoren in Abbildung 3.2 dargestellt. Die dargestellte H-Brücke ist in diesem Fall sowohl mit NPN- als auch mit PNP-Transistoren aufgebaut. Es ist auch möglich die H-Brücke nur mit npn Transistoren aufzubauen, allerdings benötigt man dazu eine hohe Steuerspannung. In diesem Prototypen kommt ein L298 als H-Brücke zum Einsatz.

### 3.1.5 Vollschrittbetrieb

Für die Ansteuerung gibt es verschiedene Varianten. Dabei ist es allerdings nicht von Bedeutung, ob die Beschaltung unipolar oder bipolar erfolgt und ist somit unabhängig davon.



**Abbildung 3.3:** FSM eines Schrittmotors mit Vollschritte.

Bei der einfachsten Variante werden immer beide Windungen betrieben und nur das Vorzeichen des Stromflusses gewechselt. Diese Ansteuerung wird Vollschrittbetrieb bezeichnet und kann mit einer FSM sehr elegant beschrieben werden. Eine mögliche FSM ist in [Abbildung 3.3](#) dargestellt. Hier wird als Zustand die Richtungen der Ströme verwendet, wobei 1 für einen Strom in Bezugsrichtung, -1 für einen Strom gegen Bezugsrichtung und 0 für eine stromfreie Windung steht. Ein Vergleich der [Abbildung 3.3](#) mit der [Abbildung 2.6](#) verdeutlicht, dass es sich bei dem generierten Signal zur Ansteuerung um ein Quadratur-Signal handelt.

Der Vollständigkeit halber ist anzumerken, dass ein Vollschrittbetrieb auch dadurch erreicht werden kann, dass immer nur eine Spule bestromt wird. Diese Art von Vollschrittbetrieb wird dadurch gekennzeichnet, dass weniger Leistung benötigt wird aber auch ein kleineres Drehmoment erzeugt wird. Außerdem sind die Schritte um einen Halbschritt versetzt. [[Ger12](#), 142]

### 3.1.6 Halbschrittbetrieb

Die Genauigkeit dieser Ansteuerung lässt sich nun verdoppeln, indem nicht nur die Richtungen der Ströme als Zustände verwendet werden, sondern auch erlaubt wird, maximal eine Spule stromfrei zu schalten. Dies ist also eine Kombination der beiden Varianten für den Vollschrittbetrieb und wird als Halbschrittbetrieb bezeichnet. Eine FSM für die Halbschrittansteuerung ist in [Abbildung 3.4](#) dargestellt.

Die doppelte Genauigkeit ist hier dadurch erkennbar, dass acht statt vier Zustände benötigt werden. Eine negative Begleiteigenschaft ist ein Verlust an Drehmoment, wenn mit dem gleichen Strom angesteuert wird. Dies lässt sich dadurch erklären, dass die Hälfte der Schritte nur mit halben Strom durchgeführt werden. [[Hag15](#), 105]

### 3.1.7 Mikroschrittbetrieb

Das Prinzip, einen Schritt zu unterteilen, lässt sich nun beliebig oft anwenden. Dazu wird zwischen je zwei Zuständen ein neuer Zustand eingefügt, bei welchem nicht der maximale Strom verwendet wird, sondern nur ein Bruchteil davon. Die Stromregelung kann in diesem Fall mit einer PWM durchgeführt werden. Da sich der Anker bei jedem Schaltvorgang bewegt, können dadurch

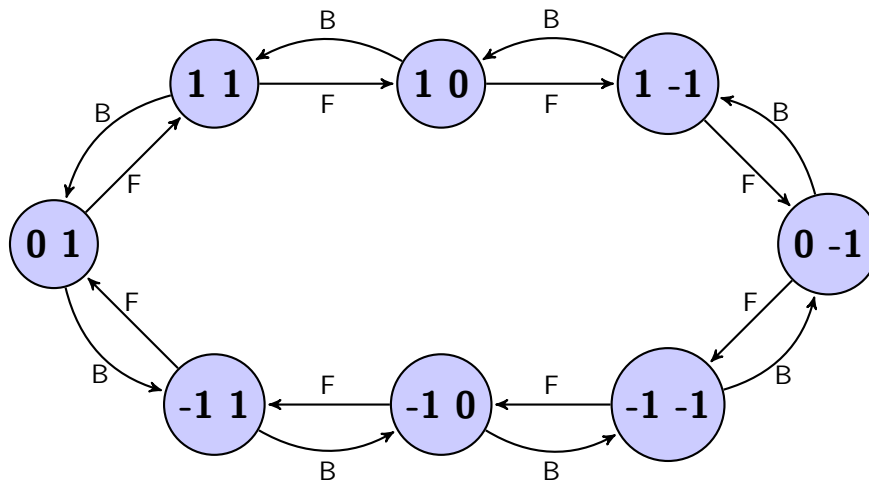


Abbildung 3.4: FSM eines Schrittmotors mit Halbschritte.

Geräusche entstehen. Deshalb sollte die Frequenz hoch genug sein, um von dem menschlichen Ohr nicht mehr wahrgenommen zu werden.

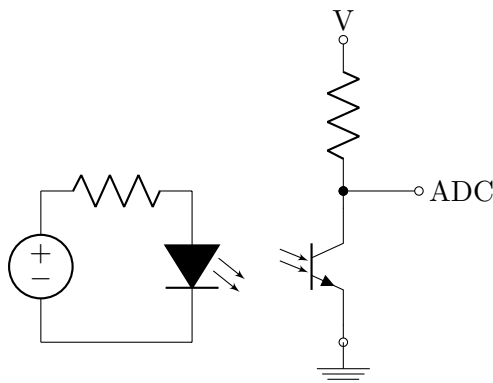
Dieses Prinzip kann beliebig oft durchgeführt werden, bis die Ansteuersignale einem Sinus und einem Kosinus entsprechen. Ab einer gewissen Schrittteilung wird jedoch keine höhere Genauigkeit mehr erzielt, da die Position Unregelmäßigkeiten aufweist, welche größer als die Schrittweite ist, der so genannte Winkelfehler. Diese Grenze ist stark abhängig von dem Typ des Schrittmotors, seiner Bauform, seine Verarbeitungsqualität, des Anwendungsbereichs und weiteren Faktoren. Der Winkelfehler wird unter anderem durch Belastung, Haftreibung und ungenaue Phasenströme verursacht. Neben der erhöhten Genauigkeit können dadurch auch Drehmomentrippel reduziert werden. Drehmomentrippel, auch Drehmomentschwankungen bezeichnet, sind Unregelmäßigkeiten im Verlauf des Drehmoments. Diese sind meistens unerwünscht, da durch diese Drehschwingungen entstehen können. Dadurch werden Mechanische Teile wie Getriebe mehr als nötig belastet und es kann zu einer erhöhten Geräuscentwicklung kommen. [MWa01, 30]

Die Grenze für den verwendeten Getriebe-Schrittmotor liegt bei dem Halbschrittbetrieb. Eine Schrittteilung bringt in diesem Fall nur eine ruhigere Umdrehung mit weniger Drehmomentrippel, diese sind hier jedoch ohnehin nicht von besonderer Bedeutung, da kein Drehmoment übertragen werden muss. Eine genauere Positionierung ist, wie bereits erläutert, wegen dem Spiel des Getriebes auch nicht möglich. Bei den für den Antrieb verwendeten Motor ist die Grenze für eine sinnvolle Schrittteilung bereits bei dem Viertelschrittbetrieb erreicht. Da hier die genaue Positionierung nicht erforderlich ist, wird auch ein Halbschrittbetrieb verwendet. Dies reduziert die Drehmomentrippel im Vergleich zu einem Vollschrittbetrieb und stellt mehr Drehmoment zur Verfügung als ein Viertelschrittbetrieb.

## 3.2 Auswertung der Gabellichtschranken

Wenn ein Lichtsignal ausgewertet werden soll, gibt es neben den Fotowiderständen hauptsächlich die Auswahl, ob Fototransistoren oder Fotodioden verwendet werden sollen. Dabei sind Fotodioden sowohl schnell als auch linear. Im Vergleich weisen Fotodioden Grenzfrequenzen bis zu

10 MHz und Fototransistoren bis zu 300 kHz auf. [UT02, 1199] Die Ansteuerung mit Fotodioden ist aber aufwändiger als jene mit Fototransistoren. Deshalb werden bei Lichtschranken oft Fototransistoren verwendet.



**Abbildung 3.5:** Einfache Auswertung eines Fototransistors.

Für diesen Anwendungszweck ist die Schaltzeit in den meisten Anwendungsfällen ausreichend und Linearität nicht gefordert. Um einen Fototransistor auszuwerten genügt, als einfachste Variante, ein Spannungsteiler mit einem Widerstand, wie in Abbildung 3.5 abgebildet. Bei dieser Art der Ansteuerung reduziert sich die Grenzfrequenz in den einstelligen kHz Bereich. Dies ist jedoch für einen Inkrementalgeber, wie er hier verwendet wird, ausreichend.

Das Signal der Fototransistoren ist ein analoges Signal, welches je nach Geschwindigkeit und Qualität der Encoderscheiben einem Rechtecksignal oder auch einem Dreiecksignal ähneln kann. Dies gilt jedoch nur unter guten Bedingungen. Ist beispielsweise die Encoderscheibe verschmutzt oder durch die Montage beschädigt worden, wird dadurch die Signalforn verzerrt. Mit einem Mikrocontroller ist jedoch ein Rechtecksignal einfacher auszuwerten. Um das Signal der Fotodioden digital an den Mikrocontroller zu liefern, ist eine zusätzliche Hardware notwendig.

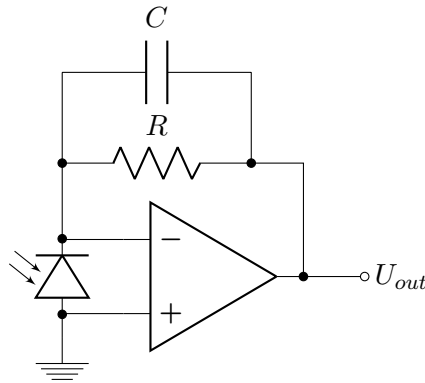
Diese Aufgabe kann durch einen so genannten Schmitt-Trigger erfolgen. Dieser digitalisiert das Signal nicht nur, er vermeidet mit einer Hysterese auch, dass durch das Rauschen beim Durchgang der Schaltschwelle ein Prellen entsteht. [ES12, 307]

### 3.3 Laser-Detektion

Die Detektion des Laserstrahles ist eine Kernaufgabe bei dieser Arbeit. Dabei sind mehrere Komponenten notwendig. Grundsätzlich lässt sich das System in den Laser, den Detektor und seine Ansteuerelektronik unterteilen. Weil der Laser für ein einfaches Auswertungsverfahren nur ein und ausgeschaltet werden muss, genügt eine Emitterschaltung um den Laser anzusteuern. Bei dem Messen des Fotostromes einer Diode wird eine Schaltung benötigt, welche den Strom verstärkt.

#### 3.3.1 Transimpedanzverstärker

Fotodioden liefern nur einen kleinen Strom im Bereich von wenigen Mikroampere. Um mit einem Mikrocontroller diesen Strom präzise messen zu können, ist ein Strom-Spannungs-Wandler



**Abbildung 3.6:** Transimpedanzverstärker.

notwendig. Ein so genannter Transimpedanzverstärker (TIA) lässt sich, wie in [Abbildung 3.6](#) dargestellt, mit einem herkömmlichen spannungsgesteuerten Operationsverstärker (VV-OPV) realisieren. In dieser Beschaltung arbeitet Photodiode wie wenn sie kurzgeschlossen wäre. Es fällt über die Diode keine Spannung ab.

Eine alternative Beschaltung ist, die Fotodiode nicht gegen Masse sondern gegen eine Vorspannung zu schalten. Bei dieser Beschaltung würde nicht die Kathode am invertierenden Eingang anliegen, sondern die Anode. Die Fotodiode wird in dieser Beschaltung in einem anderen Quadranten der Kennlinie betrieben und kann eine höhere Grenzfrequenz erreichen. [[Web97](#), 80] [[Goe99](#), 89]

Da eine Gegenkopplung vorliegt, sind die Eingänge des OPV mit einem virtuellen Kurzschluss verbunden. Deshalb ist der Strom durch den Widerstand gleich dem Fotostrom der kurzgeschlossenen Diode. Die Spannung am Ausgang ist nun die negative Spannung des Widerstandes, da sich die Richtung des Bezugssystemes geändert hat. Somit ergibt sich die Ausgangsspannung zu:

$$U_a = R * I_R = -R * I_E \quad (3.3)$$

Mit dem Widerstand  $R$  lässt sich nun die Spannung am Ausgang einstellen. In der Praxis wird dieser allerdings experimentell bestimmt, da der einfallende Fotostrom nur unzureichend bekannt ist. Für diesen Aufbau hat sich ein Widerstand von  $R = 100 \text{ k}\Omega$  als optimal erwiesen.

Da die Fotodiode eine parallele Kapazität hat, welche durch die Raumladungszone entsteht, wird eine Phasennacheilung in der Gegenkopplung erzeugt. Dies kann die Schaltung instabil machen. Um dem entgegenzuwirken wird ein Kondensator parallel zum Widerstand geschaltet. Die Größe der Kapazität ist zwar berechenbar, allerdings ist dafür der Wert des Sperrschichtwiderstandes erforderlich. Dieser wird in den Datenblättern nur selten angegeben. Auch im Fall der verwendeten BPW34 ist dieser im Datenblatt nicht vorhanden. Deshalb muss die Kapazität, welche im Bereich von wenigen pF liegt, experimentell bestimmt werden. Für manche Anwendungsfälle sind die vorhandenen Streukapazitäten bereits ausreichend. Auch bei dem Aufbau dieses Prototypen sind die Streukapazitäten bereits groß genug, um eventuelle Schwingungen in akzeptabler Zeit abklingen zu lassen.

### 3.3.2 Transiente Vorgänge

Bei einem sprunghaftigen Einschalten des Lasers müssen zuerst die vorhandenen Kapazitäten geladen werden, bevor die korrekte Ausgangsspannung abgreifbar ist. Da die Fotodiode im Kurzschluss betrieben wird, muss die Sperrschichtkapazität nicht geladen werden. Somit bleibt die

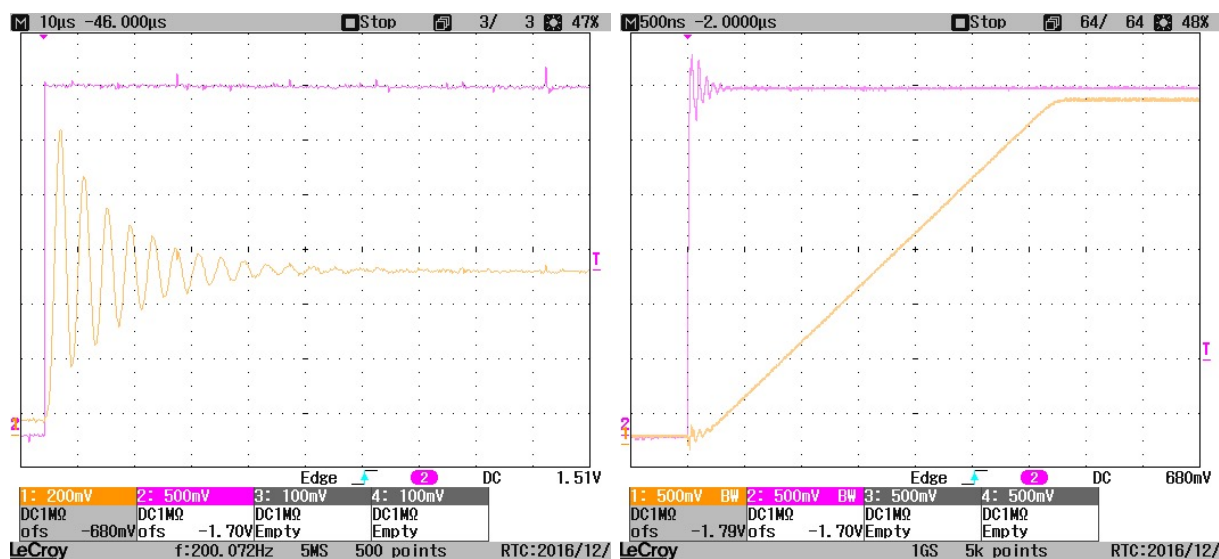


Abbildung 3.7: Abrupter Lichtanstieg bei Transimpedanzverstärker.

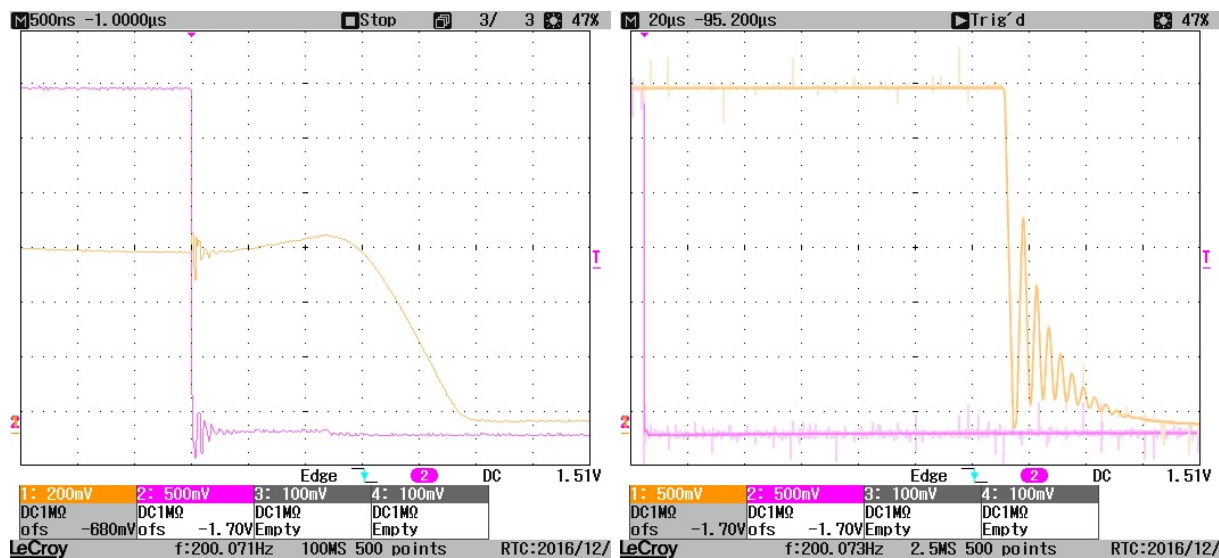
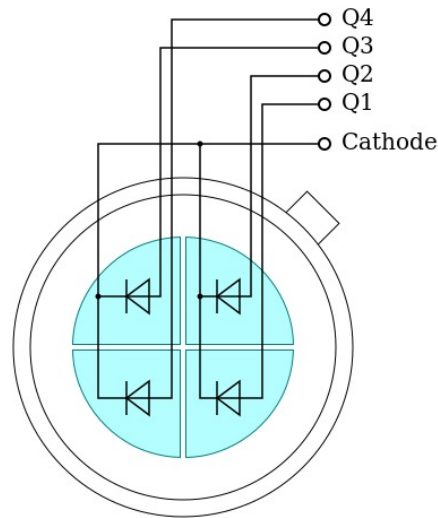


Abbildung 3.8: Abrupter Lichtabfall bei Transimpedanzverstärkers.

Kapazität des TIA übrig. Dieser Vorgang kann in Abbildung 3.7 beobachtet werden. Im linken Bild fällt nur ein kleiner Teil des Laserlichtes auf die Sensorfläche. Es ist auch gut ersichtlich, dass bevor sich ein stationärer Endzustand einstellt, die Ausgangsspannung schwingt. Nach etwa  $50 \mu\text{s}$  kann der Ausgangswert als stationär angenommen werden. Im rechten Bild ist der Laser direkt auf die Fotodiode ausgerichtet. In diesem Fall wird der TIA in die Sättigung gebracht und die Zeit, welche für die transienten Vorgänge benötigt wird, liegt unter  $4 \mu\text{s}$ . Diese Werte sind wichtig, und müssen später in der Software berücksichtigt werden.

Auch bei den Abschaltvorgängen existieren wieder transiente Vorgänge. Diese sind in Abbildung 3.8 abgebildet. Die Zeiten betragen hier  $2,5 \mu\text{s}$  und  $180 \mu\text{s}$ . Interessant ist, dass bei starken Einstrahlung der Einschaltvorgang schneller und der Abschaltvorgang langsamer abläuft. Diese lange Totzeit ist darauf zurückzuführen, dass sich im Fall einer Sättigung eine Sperrschichtkapazität aufbauen kann, da die Diode nicht mehr im Spannungsnullpunkt betrieben wird.

### 3.3.3 Quadrantendiode



**Abbildung 3.9:** Quadrantenfotodiode. Quelle: Wikipedia [Wik17a]

Um nun die Lage eines Laserpunktes bestimmen zu können, werden mehrere Fotodioden benötigt. Dazu gibt es verschiedene Möglichkeiten diese Fotodioden anzuordnen. Die bekannteste Variante ist die Quadrantendiode. Dabei werden vier Fotodioden mit einem kleinen Abstand zwischen den Sensorflächen nebeneinander positioniert. In Abbildung 3.9 ist eine Quadrantendiode schematisch dargestellt. Diese können fertig in einem Gehäuse verbaut gekauft werden. Auch Anordnungen mit mehr als vier Dioden sind erhältlich, sogenannte sedimentierte Fotodioden.

Bei dieser Arbeit werden vier einzelne BPW34 Dioden verwendet. Für die Schaltung dieses Prototypen kommt eine gemeinsame Anode zum Einsatz, da damit ein OPV mit einer einseitigen Versorgung ausreicht und keine Vorspannung eingestellt werden muss.

### 3.3.4 Auswertung

Um mit der bereits vorgestellten Hardware zu detektieren, wie viel von dem Laserlicht auf die Fläche der Fotodiode fällt, ist ein Algorithmus zur Auswertung notwendig. Zur Vereinfachung wird angenommen, dass die Ausgangsspannung linear proportional zu dem einfallenden Lichtstrom ist. Dies ist in der ersten Näherung, wie wir gezeigt haben, auch erfüllt. Dadurch kann die gemessene Intensität bei eingeschaltetem Laserstrahl von der bei ausgeschaltetem Laserstrahl subtrahiert werden. Dabei ist allerdings wichtig, dass sich mögliche Störungen langsamer ändern, als die zeitliche Entfernung der Messpunkte. Da nach dem Einschalten des Lasers, das Ausgangssignal nicht sofort auf den Endwert springt, müssen die in Kapitel 3.3.2 vermessenen Einschaltvorgänge abgewartet werden.

### 3.3.5 Verbesserte Verfahren

Eine Verbesserung zu dem bereits vorgestellten Verfahren zur Auswertung der Ausrichtung eines Laserstrahls wäre, den Laserstrahl zu modulieren. Anstatt nur die auftreffende Intensität mit der



Dunkelintensität zu vergleichen, wird ein sinusartiger Lichtstrom erzeugt. Im Empfänger wird nun mit einem phasenselektiven Synchrongleichrichter (PSSG) oder einem phasenunabhängigen Synchrodemodulator (PUSD) das Signal auf einen bestimmten Frequenzbereich überprüft. Beide Verfahren stellen schmalbandige Bandpassfilter dar. Damit lassen sich alle Störungen, welche nicht in der Messfrequenz oder deren Oberschwingungen moduliert sind, unterdrücken. [ES12, 210].

Dazu ist es allerdings notwendig, den Laser auf einem Arbeitspunkt zu modulieren, was mit einer steuerbare Stromquelle realisiert werden kann. Diese wird mit einem Frequenzgenerator gespeist. Der Frequenzgenerator ist zum Beispiel durch den DAC des Microcontrollers realisierbar. Neben dem schaltungstechnisch erhöhten Aufwand wird eine zusätzliche Software auf dem Mikrocontroller benötigt. Der wichtigste Nachteil ist allerdings die erhöhte Rechenzeit. Diese wird benötigt, um das Ausgangssignal zu generieren und eine ausreichend hohe Abtastrate zu erreichen. Aufgrund der beschränkten Rechenleistung auf dem Mikrocontroller kamen diese Verfahren nicht zum Einsatz.

## 4 Software

Dieses Kapitel gibt einen detaillierten Einblick in die Software des Prototypen.

### 4.1 Controller

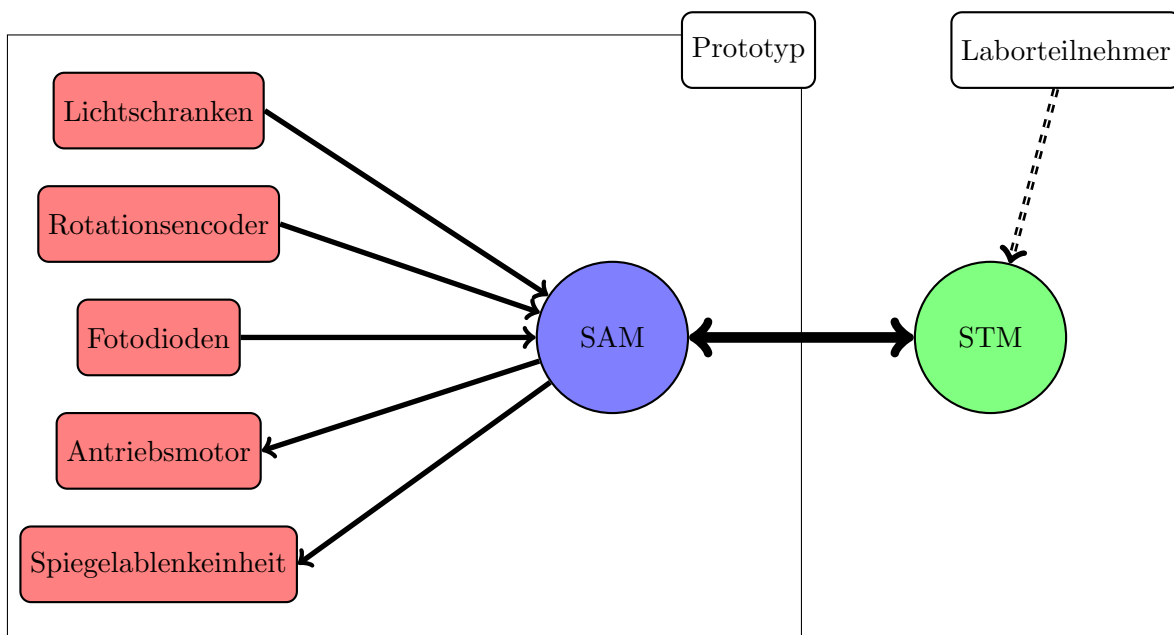


Abbildung 4.1: Zusammenspiel der Komponenten

Bei diesem Prototypen existieren zwei verschiedene Controller, auf denen Software ausgeführt wird.

Der erste ist ein SAM3X8E von Atmel mit einer 32-Bit ARM Cortex-M3 Architektur. Dieser ist auf dem Entwicklungsboard mit dem Namen „Arduino Due“ verbaut. Dieser Controller stellt für die Speicherung des Codes 512 KByte Flash und für Variablen 96 KByte SRAM zur Verfügung. Auf diesem soll die Verwaltung der Hardware ablaufen. Die Taktrate beträgt bei diesem Controller 84 MHz.

Der zweite Controller ist ein Entwicklungsboard von ST mit den Namen „NUCLEO-F334R8“. Der verbaute Mikrocontroller ist ein STM32F334R8 mit einer 32-Bit ARM Cortex-M4 Architektur. Der wesentliche Unterschied zwischen einer M3- und einer M4-Architektur liegt dabei in der zusätzlichen Gleitkommaeinheit (FPU). Eine FPU reduziert die Dauer von Operationen mit Gleitkommazahlen erheblich. Als Speicher stehen bei dem STM Controller 64 Kbyte Flash und 12 KByte SRAM zur Verfügung. Die Taktfrequenz liegt bei 72 MHz. Dieser Controller ist nur mit dem SAM3X8E verbunden und wird verwendet um den ausführbaren Code der Studierenden abzuarbeiten. Dabei soll ein Teil der Regelung von den Studenten auf dem STM32 implementiert werden.

Der Vergleich der Controller zeigt, dass der STM32 nur eine geringfügig langsamere Taktfrequenz aufweist, dafür aber eine FPU besitzt, welche einen erheblichen Geschwindigkeitsvorteil bringt. Der STM32 weist zusätzlich noch deutlich weniger Speicher auf, dies ist jedoch für die vorgesehene Aufgabe keine Einschränkung. Zusammenfassend kann behauptet werden, wenn die Regelung neben den zusätzlichen Aufgaben auf dem SAM3x8E lauffähig ist, sollte die Implementierung auf dem STM32 auch kein Problem darstellen.

Um die Zusammenhänge der Komponenten deutlich zu machen, wurden die Controller und ihre Verknüpfungen in Abbildung 4.1 grafisch dargestellt. Der SAM wird als Controller zur Verwaltung der Hardware und als Schnittstelle zu den Studenten verwendet. Der STM32 wird von den Studenten verwendet und ist nur mit dem SAM verbunden. Im Folgenden wird vor allem der Code für den SAM erläutert. Dieser enthält neben der Verwaltung der Hardware auch eine Musterlösung und somit den Teil, welcher im Labor auf den zweiten Controller ausgeführt wird. Dadurch ist die gesamte Logik, zumindest für diesen Controller, implementiert. Somit werden Ausschnitte aus dem Code in der für den SAM Controller üblichen Notation dargestellt. Diese Notation umfasst die gebräuchlichen Typen und Befehle, welche für Arduino beziehungsweise für AVR Mikrocontroller üblich sind. Dazu zählen auch die Integer-Typen, wie *uint16\_t*, welche in der *avr-libc* definiert sind. In dieser Form sind sie auf einem STM32 standardmäßig nicht definiert.

## 4.2 Schrittmotoren

### 4.2.1 Ablauf

Für die Schrittmotoren wird in diesem Prototypen ein Halbschrittbetrieb verwendet. Eine noch feinere Unterteilung ist, wie in Kapitel 3.1.7 erwähnt, nur begrenzt sinnvoll. Wie bereits erläutert, können die Schrittmotoren als FSM dargestellt und behandelt werden. Die FSM für den Halbschrittbetrieb ist in Abbildung 3.4 dargestellt.

Eine elegante Variante dies zu implementieren ist mittels eines Arrays. Dazu wird zuerst die FSM als Array dargestellt.

```
const uint8_t fsm[stateCnt][pinCnt] = {
    { 1,0, 0,0 },
    { 1,0, 1,0 },
    { 0,0, 1,0 },
    { 0,1, 1,0 },
    { 0,1, 0,0 },
    { 0,1, 0,1 },
    { 0,0, 0,1 },
    { 1,0, 0,1 }
};
```

Dabei stellen die beiden Pärchen des Arrays je eine Spule des Motors dar. Vereinfacht erklärt symbolisiert eine Zahl die normierte Spannung an einem Anschluss der Spule. Somit steht 1,0 für einen Strom in Bezugsrichtung, 0,1 für einen Strom gegen die Bezugsrichtung und 0,0 oder 1,1 für eine stromfreie Spule. 1,1 sollte jedoch nicht verwendet werden, da dies bei unipolar beschalteten Motoren zu hohen Verlusten führt.

Nun muss von einer Timer-Interrupt-Routine die aktuelle Schrittposition auf die Ausgänge des Mikrocontrollers übernommen werden. Die Timer-Interrupt-Routine wird so eingestellt, dass sie periodisch mit der maximal erlaubten Schrittgeschwindigkeit aufgerufen wird. Dies kann mit folgendem Code erreicht werden:

```
int8_t state = position % stateCnt;
for (uint8_t i = 0; i < pinCnt; i++) {
    digitalWrite(pins[i], fsm[state][i]);
}
```

Dadurch ergibt sich eine asynchrone Ansteuerung der Schrittmotoren, welche unabhängig von der Geschwindigkeit des Hauptprogramms, die Motoren im richtigen Zeitpunkt ansteuert. Dies ist deshalb besonders wichtig, da eine zu schnelle oder unregelmäßige Ansteuerung einen Schrittverlust bedeuten kann.

Vollständigkeitshalber sei hier noch erwähnt, dass der Code für einen Mikroschrittbetrieb zwar als FSM darstellbar ist, dies aber nicht zweckmäßig ist. Eleganter ist es, mit Sinus und Kosinus zu arbeiten. Da in diesem Fall der Mikroschrittbetrieb ohnehin nicht benötigt wird und die Berechnung von Gleitkommazahlen ohne FPU zeitaufwendig ist, wird bei dem Prototypen diese Methode nicht verwendet.

### 4.2.2 Timing

Für den Code, welcher in Interrupts ausgeführt wird, ist es von besonderer Bedeutung eine schnelle Verarbeitung zu erzielen. Wenn in einem Interrupt viel Zeit benötigt wird, können dadurch andere Interrupts am Ausführen gehindert werden. Dies ist unter allen Umständen zu vermeiden.

Die Frequenz, mit welcher der Interrupt aufgerufen wird, ist fest vorgegeben durch die Schrittgeschwindigkeit. Der einzige Weg, den Tastgrad der Ausführungszeiten zu variieren ist somit die Rechenzeit. Das bedeutet, dass die benötigte Rechenzeit kurz genug sein muss, um den restlichen Code nicht zu beeinflussen.

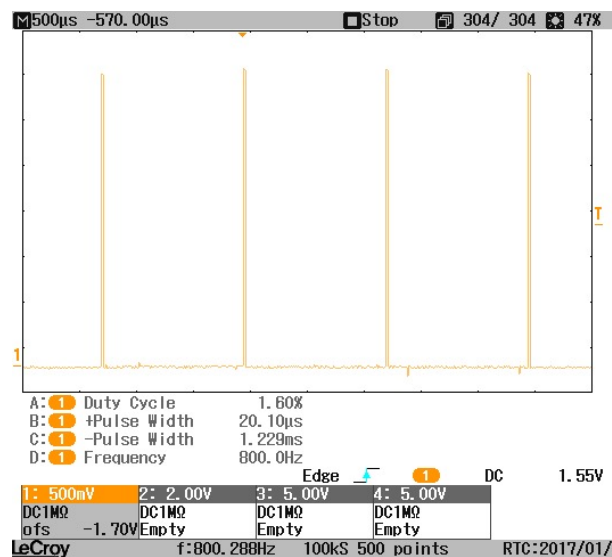


Abbildung 4.2: Dauer der Interrupts für Schrittmotoren.

Um diese Anforderung zu überprüfen, sind in Abbildung 4.2 die Zeiten aufgenommen, in denen die Interrupt-Routine aktiv war. Dabei ist der gemessene Wert auf 3,3 V wenn der Interrupt aktiv ist und 0 V wenn er nicht aktiv ist. Der gemessene Tastgrad beträgt 1,6 %. Das bedeutet, dass der Interrupt nur 1,6 % der gesamten Rechenzeit in Anspruch nimmt. Dieser Wert ist gut vertretbar und es kann angenommen werden, dass der restliche Code unbeeinflusst abgearbeitet werden kann.

### 4.3 Quadratur Encoder

Die Implementierung des Quadratur Encoders funktioniert ähnlich wie die Implementierung der Schrittmotoren. Auch hier muss der bereits in Kapitel 2.2.2 betrachtete Automat implementiert werden. Dieser kann wieder in einem Array dargestellt werden. Hier werden jedoch nicht die Zustände als Werte der Felder verwendet, sondern die Schrittweite. Die Indizes des Arrays stellen dabei die Zustände dar.

```
const int8_t encoderFsm[4][4] = {
    { 0, 1, -1, 0},
    {-1, 0, 0, 1},
    { 1, 0, 0, -1},
    { 0, -1, 1, 0}
};
```

Dabei hat die erste Ziffer des Zustandes die Wertigkeit 2 und die zweite Ziffer die Wertigkeit 1. Die Zustandsnamen stellen sozusagen den Binärkode der Indizes dar. Wird nun von dem Zustand 10 auf den Zustand 11 gewechselt, werden die Indizes 2 und 3 abgefragt und das Ergebnis ist die Schrittweite -1.

Eine wichtige Eigenschaft dieser Matrix ist, dass sie schiefsymmetrisch ist. Das bedeutet, dass die transponierte Matrix negativ ist. Somit liefert eine Umkehrung der Reihenfolge der Zustände ein negiertes Ergebnis. Ein ungültiger Übergang, so wie Übergänge von einem Zustand auf sich

selbst, liefert in dieser Variante 0. Als Beispiel kann der Übergang von 01 auf 10 mit den Indizes 1 und 2 betrachtet werden.

Da jedoch Eingangsdaten und nicht Ausgangsdaten behandelt werden, funktioniert hier die Implementierung anders als die Implementierung des Schrittmotors in Kapitel 4.2.1. Zuerst werden die Eingangsdaten benötigt, welche mit *ina* und *inb* gekennzeichnet sind. Zusätzlich ist auch der letzte Zustand von Bedeutung. Auch dieser wird mit den zugehörigen Eingangsdaten dargestellt, welche mit *lasta* und *lastb* benannt sind. Der nötige Code, um die Position zu berechnen, kann folgendermaßen geschrieben werden.

```
pos += encoderFsm[(ina << 1) + inb][(lasta << 1) + lastb];
```

Die Erkennung der großen Marke ist mit den nun berechneten Werten trivial. Wie in Kapitel 2.2.4 bereits erläutert wurde, müssen dazu nur einzelne Schritte detektiert werden, welche ein anderes Vorzeichen aufweisen. Auch eine Fehlerdetektion ist möglich, wenn in der Matrix spezielle Werte für einen Fehler eingetragen werden. Ein Fehler entsteht, wenn ein Interrupt am Ausführen gehindert wird. Dies passiert bei einer zu hohen Drehgeschwindigkeit. Der hier präsentierte Code wird im optimalen Fall direkt von der Interrupt-Service-Routine (ISR) aufgerufen, welche für die verwendeten Eingangs-Pins zuständig und auf Pegelwechsel programmiert ist.

## 4.4 Auswertung der Fotodioden

### 4.4.1 Ablauf

Genau wie bei Schrittmotoren, ist es auch bei der Auswertung der Fotodioden sinnvoll, Interrupts zu verwenden. Für diesen Prototypen wurde die in Kapitel 3.3.4 erläuterte Methodik verwendet.

Das Messen findet in zwei verschiedenen Methoden statt. Bei der ersten wird zuerst der Wert ermittelt, welchen die Fotodioden bei ausgeschalteten Laser zurückgeben. Dieser Wert kann nun aufgrund der Linearität in der zweiten Methode dazu verwendet werden, langsame Störungen, wie das Tageslicht, durch Subtraktion zu eliminieren.

Diese beiden Methoden müssen abwechselnd mit Pausen dazwischen ausgeführt werden. Am besten wird dazu ein Interrupt verwendet. Dazu muss die Periodendauer des Interrupts bei jeder Abarbeitung umgeschaltet werden. Die Zeiten werden im folgenden als Messzeit und Wartezeit bezeichnet. Die Messzeit ist jene, welche zwischen den beiden Methoden verstreicht. Die Wartezeit ist die Zeit, welche zwischen der zweiten Methode und der ersten der nächsten Messung verstreicht.

### 4.4.2 Timing

Da auch bei den Interrupts für die Fotodioden nicht zu viel Zeit verschwendet werden darf, ist besonders die Einstellung der Messfrequenz wichtig. Bei den Schrittmotoren war die Messfrequenz gegeben, anders ist dies bei der Auswertung der Fotodioden. Hier kann die Ausführungszeit für jede der beiden Unterfunktionen als konstant angenommen werden. Um nun den Tastgrad auf einen vertretbaren Wert zu bringen, darf die Summe von Messzeit und Wartezeit zwischen den Messungen nicht zu niedrig liegen. Somit muss die Messfrequenz so gewählt werden, dass ein ungestörter Programmablauf der anderen Programmteile gewährleistet werden kann. Die Messfrequenz muss

aber ausreichend hoch sein, um die Regelung nicht negativ zu beeinflussen. Für diesen Prototypen wurde eine Messfrequenz von 200 Hz empirisch ermittelt.

Der einzige Parameter, welcher nach den bisherigen Festlegungen noch gewählt werden kann, ist die Aufteilung in Messzeit und Wartezeit. Die Zeitdauern, welche für diese Wartezeiten mindestens eingehalten werden müssen, wurden bereits im Kapitel 3.3.2 ausgemessen. Um eine gültige Messung zu gewährleisten, dürfen diese Zeiten auf keinen Fall unterschritten werden.

Solange die notwendigen Mindestzeiten eingehalten werden, verändert sich dadurch auch die Qualität der Messung nicht. Das Verhältnis dieser Werte gibt jedoch die, von menschliche Auge wahrgenommene, Helligkeit des Laserpunktes an. Ist die Messzeit auf den minimal erlaubten Wert eingestellt, ist der Laserpunkt unter Tageslicht kaum mehr wahrnehmbar. Bei einer minimalen Wartezeit ergibt sich die maximale Helligkeit des Lasers. Somit kann als Nebeneffekt dieser Regelung die gewünschte Helligkeit eingestellt werden. Das Timing der Interrupts ist auch in diesem Fall wieder von besonderem Interesse und wird zur Überprüfung in Abbildung 4.3 dargestellt. Der Tastgrad ist mit 8,41 % höher als bei den Schrittmotoren aber immer noch vertretbar. In dieser Abbildung ist auch ersichtlich, dass die Methode mit unterschiedlichen Wartezeiten aufgerufen wird. Für diese Aufnahme wurde der Laser auf eine geringe Leuchtkraft eingestellt. Zwischen zwei nahen Peaks ist also der Zeitraum in denen der Laser leuchtet.

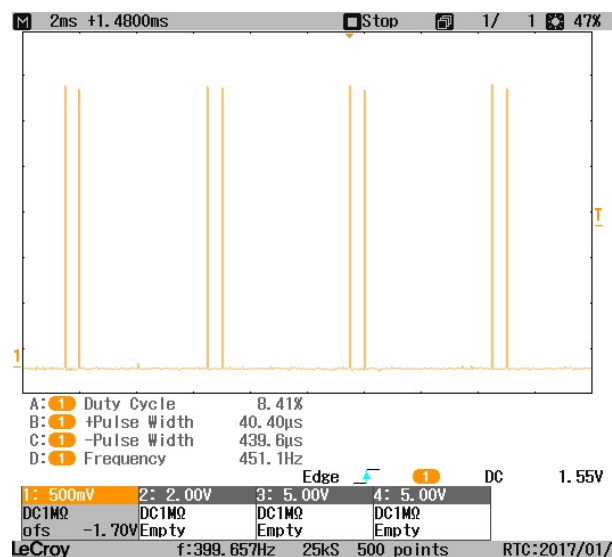
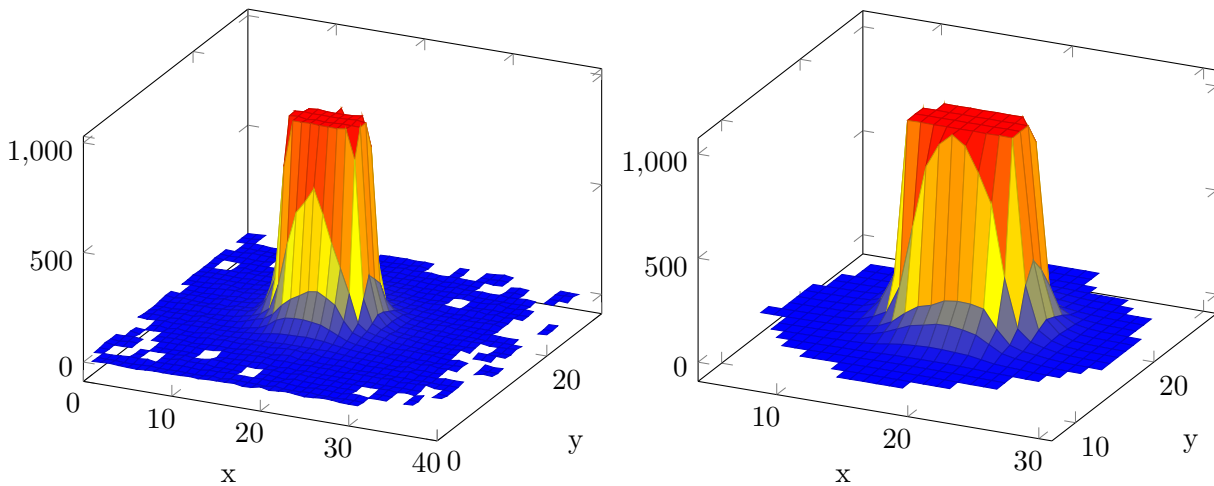


Abbildung 4.3: Timing der Interrupts für die Auswertung der Fotodioden.

## 4.5 Implementierung einer Regelung

### 4.5.1 Charakteristik einer Fotodiode

Eine wichtige Grundlage beim Aufbau einer Regelung sind die zur Verfügung stehenden Informationen. Um sich ein Bild von den möglichen Eingangsdaten zu machen, wurde mit dem Laserstrahl im Halbschrittbetrieb jeder mögliche Punkt auf den Fotodioden abgetastet. Die Referenzspannung des ADC hat bei der Messung 3.3 V betragen und es wurde mit 10 Bit Auflösung gemessen. Somit entspricht ein digitaler Wert von 1 einer Spannung von 3,22 mV



**Abbildung 4.4:** Ausgabewerte einer Fotodiode mit und ohne Störeinflüsse, welche im normalen Betrieb auftreten.

Die Daten einer einzelnen Fotodiode wurden in Abbildung 4.4 dargestellt. Die Grafik gibt dabei detailliert die Ausgangswerte, in Abhängigkeit zu der Schrittposition der beiden Schrittmotoren, an. Um dabei die Übersichtlichkeit zu erhalten, wurden nur Datenpunkte angegeben, welche einen Wert von mindestens 5 aufweisen, dies entspricht  $16,1 \text{ mV}$ .

Eine wichtige Anmerkung ist, dass die rechte Grafik auf einer Messung unter optimalen Bedingungen basiert. Dabei wurde versucht die Fremdlichtintensität auf ein Minimum zu reduzieren. Weiteres wurde für einen Aufnahmepunkt über 100 Messungen gemittelt. Um eine Vorstellung davon zu bekommen, wie diese Einflüsse die Messung beeinflussen können, wurde zum Vergleich das linke Bild unter normalen Bedingungen aufgenommen. Dabei ist deutlich sichtbar, dass die Werte gewissen Schwankungen unterliegen, die Qualität jedoch ausreichend ist, um damit eine Regelung zu implementieren.

#### 4.5.2 Charakteristik aller Fotodioden

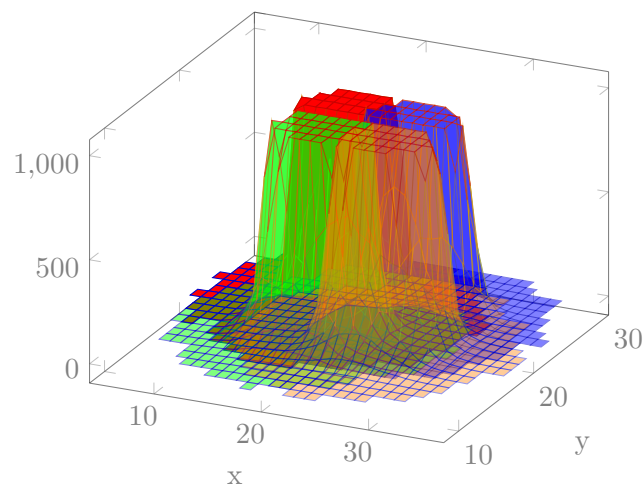
Abbildung 4.4 zeigt neben den bereits diskutierten Eigenschaften auch, dass die Fotodiode bei direkter Bestrahlung sättigt. Dies ist für eine Regelung durchaus ein Problem und könnte behoben werden, indem der, in Kapitel 3.3.1 besprochene Widerstand, durch einen Kleineren ersetzt wird. Dadurch verringert sich jedoch auch die Empfindlichkeit am Rand des Messbereiches.

Wenn nun alle vier Fotodioden in ein Bild eingearbeitet werden, sieht man, dass sich die Sättigungsbereiche der Dioden kaum überlappen. Dies wird in Abbildung 4.5 dargestellt. Selbst wenn man sich nun auf einen Punkt befindet, bei welchem zwei Dioden sättigen, so befindet man sich bereits im Messbereich der anderen beiden Dioden. Dadurch kann im gemeinsamen Messbereich der Dioden, an den meisten Positionen der Standort bestimmt werden.

#### 4.5.3 Entwurf eines Dreipunktreglers

Mit den bereits diskutierten Werten werden nun zwei Regler entworfen. Ein Regler ist zuständig für den Schrittmotor, welcher den Laser horizontal ablenkt, der andere regelt die vertikale Ab-



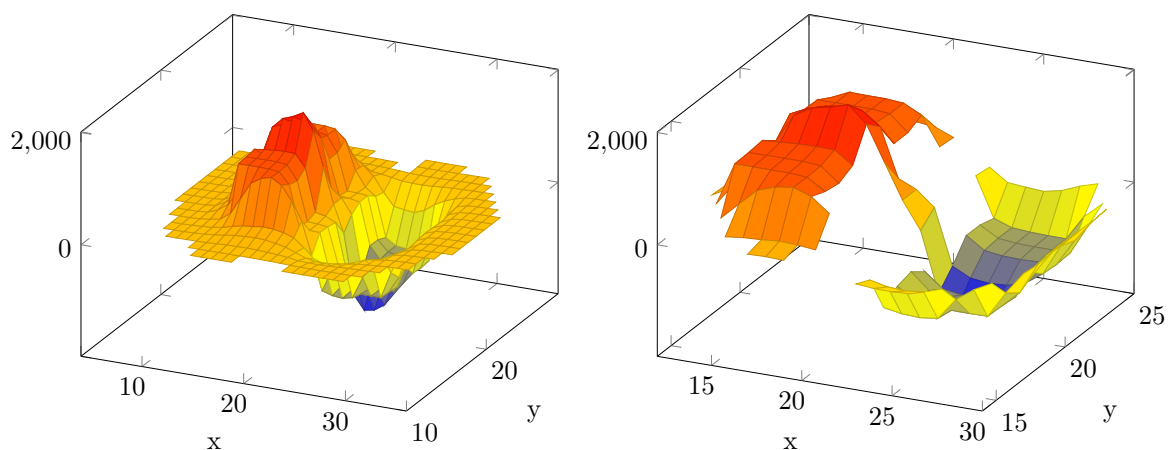


**Abbildung 4.5:** Ausgabewerte der Fotodioden in einer gemeinsamen Grafik.

lenkung. Um die Diagramme vergleichbar zu machen, zeigen sie in diesem Abschnitt immer den Regler, welcher die lange Seite der Dioden regelt. Diese Achse wird als x Achse bezeichnet.

Bei dem Entwurf des Reglers könnte im Folgenden berücksichtigt werden, dass die Fotodioden rechteckig sind. Würde diese Information jedoch die Regelparameter mitbestimmen, so müsste im folgenden für jede Achse ein Regler mit verschiedenen Parametern verwendet werden. Das soll hier für die erste Betrachtung jedoch vermieden werden. Da nun beide Regler äquivalent zueinander sind, ist es ausreichend einen zu entwerfen.

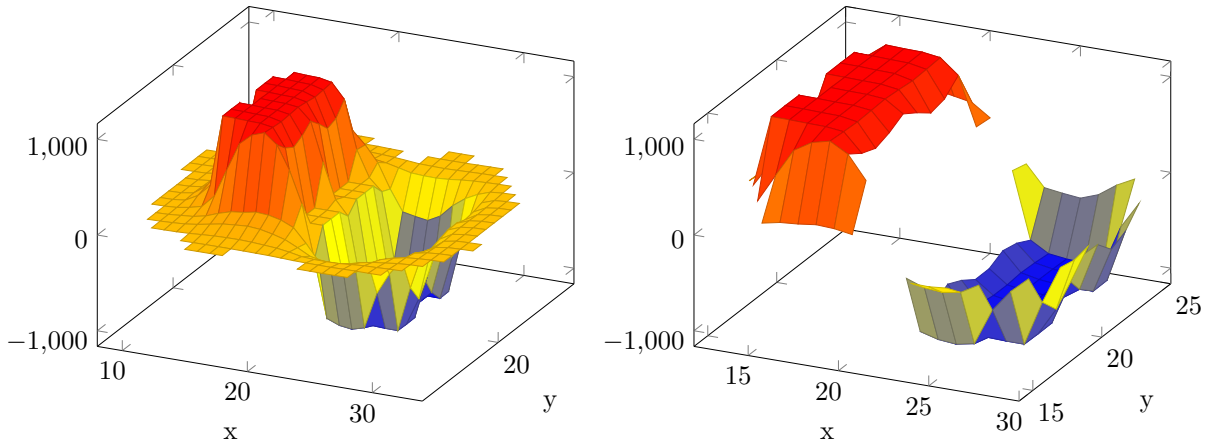
Das Ergebnis dieses Entwurfes soll eine Entscheidungsstrategie sein, welche entscheidet, ob der Motor in eine Bezugsrichtung drehen, gegen die Bezugsrichtung drehen oder still stehen soll. Dafür gibt es verschiedene Strategien die gemessenen Rohdaten aufzubereiten. So könnte zuerst die Position des Laserpunktes ermittelt werden, um mit dieser Information die Regelung durchzuführen. Da die Position aber nichtlinear von den Eingangsdaten abhängig ist, ist es einfacher, direkt mit den Eingangsdaten der Dioden zu arbeiten.



**Abbildung 4.6:** Dreipunktregler basierend auf addierten Messwerten.

Ein erster Ansatz für die Auswertung der Daten ist, die Werte auf der y Achse zu summieren und auf der x Achse zu subtrahieren. Diese Verknüpfung der Eingangsdaten ist in [Abbildung 4.6](#)

im linken Bild dargestellt. Mit den so gewonnenen Daten kann nun ein Dreipunktregler gespeist werden, welcher mit zwei Schaltschwellen entscheidet, wie der Motor angesteuert werden sollen. Die dabei entstehende Schaltcharakteristik ist in Abbildung 4.6 im rechten Bild dargestellt. Dabei wurden nur jene Bereiche gezeichnet, welche über oder unter möglichen Schwellwerten liegen. Anders formuliert sind die Punkte des rechten Bildes die Teilmenge der Punkte des linken Bildes, bei der eine Interaktion des Motors erforderlich ist.



**Abbildung 4.7:** Dreipunktregler basierend auf Maximum Werte.

Eine alternative Verknüpfung ist, die y Achse nicht zu summieren, sondern den Maximalwert zu verwenden. Dadurch können Überhöhungen, welche beim Übergang von einer Fotodiode auf eine andere entstehen, vermieden werden. Die Eingangsdaten und der Schaltbereich dieser Variante ist in Abbildung 4.7 dargestellt. Dabei sind die Unterschiede zur ersten Variante gut erkennbar. Diese Unterschiede befinden sich hauptsächlich bei den Grenzbereichen der Fotodioden.

#### 4.5.4 Implementierung eines Dreipunktreglers

Im folgenden wird die Implementierung des bereits besprochenen Dreipunktreglers gezeigt. Dabei wird die erste Variante erläutert, bei der die Eingangswerte additiv verknüpft werden.

Es ist wichtig, dass der Regler mit einer konstanten Frequenz periodisch aufgerufen wird. Um eine konstante Frequenz zu gewährleisten, könnte der Regler mit einem Timer-Interrupt ausgeführt werden. Da der Regler allerdings viele Gleitkommaoperationen benötigt und deshalb eine lange Verarbeitungszeit benötigt wird, ist dies nicht zu empfehlen. Stattdessen wird der Regler in der Hauptschleife abgearbeitet und es wird versucht die anderen Aufgaben in Interrupts zu verlagern. Um dabei eine konstante Abtastfrequenz zu erhalten, wird nach jedem Zyklus solange gewartet, bis der reziproke Wert der Update-Frequenz erreicht ist.

Diese Implementierung beinhaltet drei Teilaufgaben. Die erste ist das Errechnen und Mitteln des Wertes für den Dreipunktregler. Zur Mittlung wird ein Infinity Impulse Response (IIR) Tiefpassfilter erster Ordnung verwendet. Der Vorteil dieses IIR Filter ist seine einfache Implementierung. Dies kann mit folgendem Code erreicht werden.

```

horizontal = diodeLU + diodeLD - diodeRU - diodeRD;
vertical = diodeLU + diodeRU - diodeLD - diodeRD;
lowHor = (lowHor*filterParam + horizontal) / (filterParam + 1);
lowVer = (lowVer*filterParam + vertical) / (filterParam + 1);

```

Eine schnellere und präzisere Reaktion des Reglers kann dadurch erreicht werden, dass der Filter häufiger ausgeführt wird als der Regler. Dies sorgt für eine genauere Messung und reduziert das Rauschen und sonstige unkorrelierte Messfehler.

Die zweite Teilaufgabe ist die Berücksichtigung der Drehung der Encoderscheibe. Dazu wird, mit dem vom Drehgeber generierten Winkel, das Koordinatensystem der Fotodioden in das der Spiegelablenkeinheit umgerechnet. Dies kann, da die Drehung nur um die z Achse verläuft, mit einer Multiplikation mit Sinus und Kosinus erfolgen.

```

double h = cos(encoderAngle);
double v = sin(encoderAngle);

```

Die letzte Aufgabe besteht darin, zu entscheiden ob, eine der Schaltschwellen erreicht wurde. Wenn dies der Fall ist, kann ein Koordinaten transformierter Schritt ausgeführt werden.

```

if (lowVer > triggerLevel) {
    stepperPosV += cosV;
    stepperPosH -= sinV;
} else if (lowVer < -triggerLevel) {
    stepperPosV -= cosV;
    stepperPosH += sinV;
}
if (lowHor > triggerLevel) {
    stepperPosH += cosV;
    stepperPosV += sinV;
} else if (lowHor < -triggerLevel) {
    stepperPosH -= cosV;
    stepperPosV -= sinV;
}

```

Hier ist es auch möglich, nicht nur einen Schritt sondern mehrere zu machen. Ein Schritt bezeichnet dabei die kleinste verwendete Unterteilung. Bei dem Prototypen ist dies ein Halbschritt für den Motor.

Dabei sind zwei Punkte zu beachten, damit der Regler gut funktionieren kann. Zum ersten müssen die Schritte gerundet werden, bevor sie dem Schrittmotor als Eingangssignal gegeben werden. Dies ist nachvollziehbar, da immer nur ein vollständiger Schritt ausgeführt werden kann, wenn nicht eine weitere Schrittteilung vorgenommen wird.

Der zweite Punkt ist, dass nicht mehr Schritte von der Regelung generiert werden dürfen, als der Motor wirklich umsetzen kann. Dies ist für die Regelung von großer Bedeutung, da diese dadurch instabil werden kann, wenn die Motoren asynchron angesteuert werden. Hat der Regler eine höhere Abtastrate als die Schrittfrequenz der Motoren, dann generiert der Regler Schritte ohne dass die Motoren die letzte Position überhaupt erreicht haben. Erreicht der Motor danach die Soll-Position, generiert der Regler keinen Schritt mehr, die Motoren haben aber noch nicht abgearbeitete Schritte. Dies führt zu einem Überschwingen oder zu einem instabilen Regler.

## 5 Zusammenfassung

In dieser Arbeit wurde ein Prototyp vorgestellt, der in Zukunft in der Lehrveranstaltung „Mikrocomputer Labor“ verwendet werden soll. Die Aufgaben, welche von den Studierenden gelöst werden müssen, sind Teilaufgaben der Regelung einer Spiegelablenkeinheit auf eine Scheibe mit einem Detektor.

Bei Arbeiten, welche einen repräsentativen Charakter haben, gibt es neben den technischen Eigenschaften auch gewisse Voraussetzungen, welche das Design der Arbeit betreffen. Dadurch treten Probleme auf, mit denen normalerweise nicht gerechnet werden können. Das am Anfang geplante Pendel war technisch gesehen voll funktionsfähig, jedoch war dieses optisch nicht ansprechend. Unregelmäßige Schwingeeigenschaften und eine zu instabile Konstruktionen waren die größte Probleme, weshalb das Pendel durch eine Scheibe ersetzt wurde.

Viele Schwierigkeiten dieser Arbeit sind bei der Erkennung des Lasers gelegen. Dafür musste nicht nur ein geeignetes Verfahren gesucht werden, sondern auch die Aufbereitung der so gewonnenen Informationen ist ein interessanter Aufgabenbereich. In diesem Teil der Arbeit gibt es durchaus noch Herausforderungen wie das Implementieren eines PSSG oder eines PUSD.

Eine weiterer interessanter Themenbereich, welcher in dieser Arbeit nur kurz angeschnitten wurde, ist die verwendete Regelung. Der Übergang von einem Dreipunktregler zu einem Zustandsregler oder anderen Verfahren, würde dabei eine merkbare Verbesserung der Regeleigenschaften versprechen.

# Wissenschaftliche Literatur

- [ES12] ELMAR SCHRÜFER, Bernhard Z.: *Elektrische Messtechnik*. 10. Hanser, 2012. – ISBN 978-3-446-43079-2
- [Ger12] GERKE, Prof. Dr.-Ing. W.: *Elektrische Maschinen und Aktoren*. Oldenbourg Verlag München, 2012. – ISBN 3-540-00730-X
- [GG03] GERHARD GOOS, Jan van Leeuwen Andy M. Tyrrell Pauline C. Haddow Jim T.: *Evolvable Systems: From Biology to Hardware: 5th International Conference, ICES 2003, Trondheim, Norway, March 17-20, 2003. Proceedings*. 5. Springer-Verlag Berlin Heidelberg New York, 2003. – ISBN 978-3-486-71265-0
- [Goe99] GOERTH, Professor Dipl.-Ing. J.: *Bauelemente und Grundsaltungen*. 1. B. G. Teubner Stuttgart - Leipzig, 1999. – ISBN 978-3-519-06258-5, 978-3-322-87191-6
- [Hag15] HAGL, Rainer: *Elektrische Antriebstechnik*. 2. Carl Hanser Verlag München, 2015. – ISBN 978-3-446-44270-2
- [MWa01] H. MANFRED WECK (AUTH.), Prof. Dr.-Ing. Dr.-Ing. E.: *Werkzeugmaschinen Fertigungssysteme 3: Mechatronische Systeme: Vorschubantriebe und Prozessdiagnose*. 5. Springer Berlin Heidelberg, 2001. – ISBN 978-3-662-10917-5, 978-3-662-10916-8
- [UT02] ULRICH TIETZE, E. G.: *Halbleiter-Schaltungstechnik*. 12. Springer, 2002. – ISBN 3-540-42849-6
- [Web97] WEBSTER, John G.: *Design of pulse oximeters*. Taylor & Francis Group, LLC, 1997 (Medical science series). – ISBN 0-7503-0467-7
- [Wik17a] WIKIPEDIA. *Quadrantendiode*. 2017
- [Wik17b] WIKIPEDIA. *Schrittmotor*. 2017

# Abbildungsverzeichnis

1.1	3D-Modell einer Spiegelablenkeinheit und einem Pendel. . . . .	1
2.1	3D-Modell des Prototypen. . . . .	3
2.2	Winkel bei der Laserablenkeinheit. . . . .	4
2.3	3D-Modell der Spiegelablenkeinheit. . . . .	5
2.4	Funktionsweise eines Drehgebers. . . . .	5
2.5	Aufnahme des Quadratursignals der drehbaren Scheibe. . . . .	6
2.6	FSM eines Drehgebers. . . . .	7
2.7	Aufnahme des Quadratursignals der drehbaren Scheibe. . . . .	8
2.8	Scheibe für den Rotationsencoder. . . . .	9
3.1	Aufbau eines Schrittmotors mit unipolarer Beschaltung. Quelle: Wikipedia [Wik17b]	11
3.2	Aufbau einer H-Brücke für eine Windung . . . . .	12
3.3	FSM eines Schrittmotors mit Vollschrte. . . . .	13
3.4	FSM eines Schrittmotors mit Halbschritte. . . . .	14
3.5	Einfache Auswertung eines Fototransistors. . . . .	15
3.6	Transimpedanzverstärker. . . . .	16
3.7	Abrupter Lichtanstieg bei Transimpedanzverstärker. . . . .	17
3.8	Abrupter Lichtabfall bei Transimpedanzverstärkers. . . . .	17
3.9	Quadrantenfotodiode. Quelle: Wikipedia [Wik17a]	18
4.1	Zusammenspiel der Komponenten . . . . .	20
4.2	Dauer der Interrupts für Schrittmotoren. . . . .	23
4.3	Timing der Interrupts für die Auswertung der Fotodioden. . . . .	25

4.4	Ausgabewerte einer Fotodiode mit und ohne Störeinflüsse, welche im normalen Betrieb auftreten. . . . .	26
4.5	Ausgabewerte der Fotodioden in einer gemeinsamen Grafik. . . . .	27
4.6	Dreipunktregler basierend auf addierten Messwerten. . . . .	27
4.7	Dreipunktregler basierend auf Maximum Werte. . . . .	28

## Erklärung

*Hiermit erkläre ich, dass die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt wurde. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.*

*Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.*

Wien, am 15. Januar 2017

---

Andreas Gruber