

Mikroprozessorgesteuerte Fasereinkopplung

Bachelorarbeit
zur Erlangung des akademischen Grades
Bachelor of Science

vorgelegt von
Simon Johannes Stephan Reichel

Rheinische Friedrich-Wilhelms-Universität Bonn
Mathematisch-Naturwissenschaftliche Fakultät
Institut für angewandte Physik
Arbeitsgruppe Meschede
August 2014

1. Gutachter : Professor Dr. Meschede

2. Gutachter : Dr. Alberti

Datum des Einreichens der Arbeit: 01.08.2014

Inhaltsverzeichnis

Inhaltsverzeichnis	III
1 Einleitung	1
2 Theoretische Grundlagen	2
2.1 Fasereinkopplung	2
2.1.1 Glasfaserkabel	2
2.1.2 Einkopplung	3
2.1.3 Manuelle Optimierung	4
2.1.4 Kommerzielles System	5
2.2 Optimierung	6
2.2.1 Bergsteigeralgorithmus	6
2.2.2 PID-Regler	7
2.3 Bewegungssystem der Spiegel	7
2.3.1 Motoren mit Pulsweitenmodulation	7
2.3.2 Steppermotoren	8
2.3.3 Piezos	8
3 Experiment	9
3.1 Aufbau	9
3.1.1 Erklärte Skizze	9
3.1.2 Bauteile	10
3.2 Ein Schritt	11
3.2.1 Rauschen und Mittelung	11
3.2.2 Wartezeit nach einem Schritt	13
3.2.2.1 Richtungswechsel	15
3.2.3 Algorithmus „Ein Schritt“	17
3.3 Scan mit einem Motor	18
3.3.1 1D-Optimierung	20
3.3.2 Messung Algorithmus “Scan“	22
3.4 Scan über eine Achse	23
3.4.1 Messung Achsenscan	24
3.5 Scan im Zusammenspiel von zwei Achsen	25
3.5.1 Algorithmus	25
4 Ausblick und Fazit	29
4.1 Verbesserungsmöglichkeiten	29
4.1.1 Verbesserung der Laufzeit	29
4.1.2 Verbesserungen im Aufbau	29

INHALTSVERZEICHNIS

4.1.3	Verbesserung der Algorithmik	30
4.2	Fazit	30
A	Fotos	V
A.1	Fotos vom Aufbau	V
A.2	Einzelne Bauteile	VII
A.2.1	Spiegelhalterung	VII
A.2.2	Halterung Motoren	VIII
A.2.3	Photodiode	IX
A.2.4	mbed	X
B	Verwendete Software	XIII
B.1	Latex	XIII
B.2	Compiler	XIII
B.3	Graphiken	XIII
B.4	Schaltplan	XIII
	Quellenverzeichnis	XIV

Kapitel 1

Einleitung

Immer komplexer werdende Experimente erfordern ein hohes Maß an Feineinstellungen und routinemäßiger Wiedereinstellung vieler Geräte im Labor. Viele dieser Arbeiten können dabei prinzipiell automatisiert und von Computern bzw. Mikrocontrollern übernommen werden.

Ein Routineproblem in den Laboren der Arbeitsgruppe Meschede ist die Einkopplung eines Laserstrahls in ein Glasfaserkabel. Dabei wird die Intensität nach der Faser über vier Stellschrauben optimiert. Erfahrungsgemäß ist die Einkopplung aber nicht über Wochen stabil und somit muss sie wiederholt optimiert werden.

Diese Arbeit bringt dem Laboranten keine weiteren wissenschaftliche Erkenntnisse. Deswegen soll in dieser Bachelorarbeit ein System erarbeitet und analysiert werden, das automatisch einen Laserstrahl in ein polarisationserhaltendes Einmoden-Glasfaserkabel einkoppelt.

Kapitel 2

Theoretische Grundlagen

Zunächst sind hier Fasereinkopplung und einige Optimierungsalgorithmen im Allgemeinen dargestellt, um eine theoretische Grundlage für den späteren Aufbau zu geben.

2.1 Fasereinkopplung

Glasfaserkabel sind für die Quantenoptiklabore unerlässlich.

In ihnen kann ein Laserstrahl leicht von einem Ort zu einem anderen geführt werden. Bei der Einkopplung kommt es dabei jedoch zu ungewollten Verlusten. Diese zu minimieren ist Idee der Bachelorarbeit und um sie zu verstehen, sind Glasfaserkabel und die Einkopplungsverluste in diesem Kapitel erklärt.

2.1.1 Glasfaserkabel

Nach dem SNELLIUSschen Brechungsgesetz $n_1 \sin(\theta_1) = n_2 \sin(\theta_2)$ existiert das Phänomen der Totalreflexion: Bricht Licht aus einem optisch dichteren Medium unter einem großen Winkel an einer Grenzfläche mit einem weniger dichtem Medium, wird alles Licht reflektiert, da wenn $\frac{n_1}{n_2} \sin \theta_1 > 1$ folgt, dass keine Lösungen für θ_2 existieren.

Dies macht man sich bei Lichtwellenleitern wie Glasfaserkabeln zu nutze. Durch die Totalreflexion „fängt“ man Licht in einem Kabel und führt es an einen beliebigen Ort. Betrachten wir die Einkopplung schematisch:

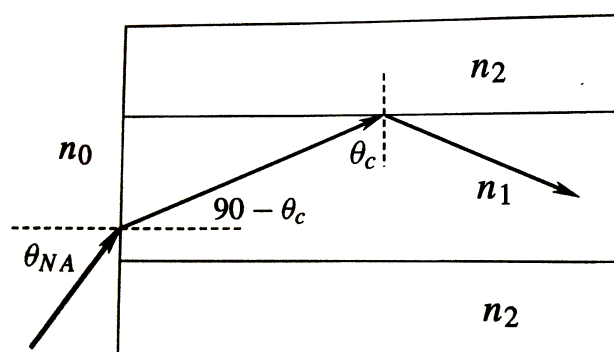


Abbildung 2.1: Einkopplung eines Strahls in ein Glasfaserkabel. Bearbeitete Abbildung aus [1, S. 52]
Aus diesem Schema lässt sich mit bekannten optischen Dichten der Glasfaser der minimale Winkel des geführten Lichtes für die Lichtleitung unter Vernachlässigung des Krümmungswinkels der Faser berechnen. Daraus ergibt sich ein minimaler Akzeptanzwinkel θ_{NA} des Lichtes

vor der Faser. Führt man nun einen Gaußstrahl auf das Kabel, wird alles Licht außerhalb des Akzeptanzwinkels abgeschnitten.

Damit sollte die Intensität von Winkel und Position jeweils gaußförmig abhängig sein. Da einer Disposition durch einen Winkel $\neq 90^\circ$ entgegengewirkt werden kann, ist zudem ein Korrelationsterm zu erwarten.

Um die Reflexionen der Grenzfläche aus dem Aufbau abzulenken, ist die Grenzfläche typischerweise um einige Grad zur Faser verkippt. Die resultierenden Terme sind konstant und damit für die Optimierung unwichtig und werden vernachlässigt. Natürlich kann ein Glasfaserkabel je nach Struktur viele Moden führen, in unserem späteren Aufbau sind allerdings nur Monomodenfasern verbaut.

Desweiteren werden im Aufbau polarisationserhaltende Fasern verwendet. Polarisationserhaltung wird dadurch gewährleistet, dass jede Faser mehr als einen Kern hat, in dem sie das Licht führt. Die Polarisation spielt für die resultierende Intensität keine Rolle und wird damit auch vernachlässigt.

2.1.2 Einkopplung

Koppelt man einen Laserstrahl in ein Glasfaserkabel kommt es zu Verlusten. Gehen wir von einem gaußförmigen Laserstrahl aus, erhalten wir nach [1, S. 116 und 117]:

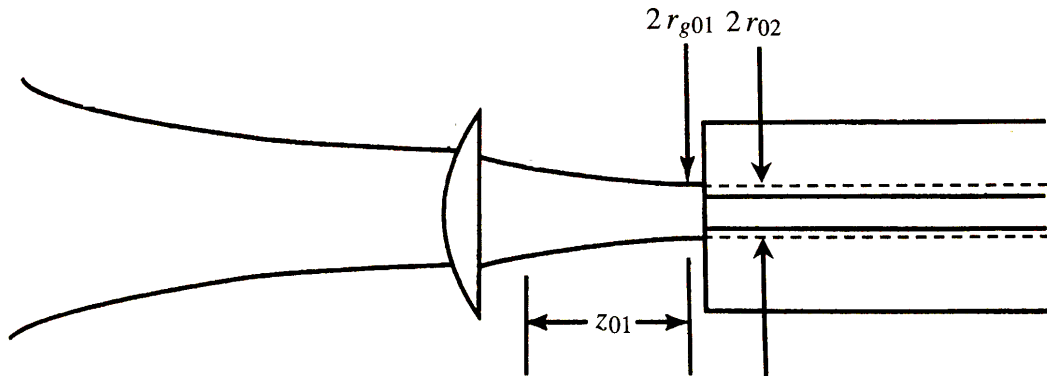


Abbildung 2.2: Einkopplung eines Gaußstrahls in ein Glasfaserkabel bearbeitet aus [1, S. 116]

Für die Einkopplungseffizienz η gilt nach [1, S. 117]:

$$\eta = \eta_l \exp \left(-\eta_l \left\{ \frac{x_0^2}{2} \left(\frac{1}{r_{g01}^2} + \frac{1}{r_{02}^2} \right) + \frac{\pi^2 \theta^2}{2\lambda^2} [r_{g01}^2(Z) + r_{02}^2] - \frac{x_0 \theta Z}{r_{g01}^2} \right\} \right) \quad (2.1)$$

Zeichen	Bedeutung	ungefähre Größenordnung ¹
η	Einkopplungseffizienz	
x_0	Transversale Abstand zwischen den Achsen	< 0,1 mm
r_{g01}	Gaußer Strahlradius an Taille	0,1 mm
r_{02}	Strahlradius in Glasfaserkabel	0,1 mm
θ	Winkel zwischen Strahlachsen (in Radiant)	wenige Grad $\rightarrow$ 0 Grad
λ	Wellenlänge des Lasers	853 nm (infrarot)
Z	Der Abstand des Ende der Glasfaser zum Fokus	< 0,01 mm $\rightarrow$ 0 mm
η_l	Longitudinale Effizienz	< 1 $\rightarrow$ 1

Tabelle 2.1: Erklärung der Formelzeichen Einkopplungseffizienz

In dieser Formel sehen wir zunächst die erwartete gaußförmige Abhängigkeit ($\exp(-cx^2)$) von jeweils Achsen- und Winkeldisposition. Zudem ist der Korrelationsterm linear im Exponenten ($\exp(+c \cdot \theta x_0)$).

Die Kollimation ist optimierbar, wird im späteren Aufbau aber konstant gehalten, so dass r_{g01}, r_{02}, λ und Z Konstanten sind. Die longitudinale Effizienz η_l ist von [1, S. 117] auch gegeben.

$$\eta_l = \frac{4r_{g01}^2 r_{02}^2}{(r_{g01}^2 + r_{02}^2) + \lambda^2 Z^2 / \pi^2}$$

Für $Z \rightarrow 0$ und $r_{g01} \cong r_{02}$ ist $\eta_l \lesssim 1$. Für eine Abweichung der Strahlradien um ca. 10% liegt die longitudinale Effizienz η_l bei ca. 0,83. Auch Dispositionen in der Kollimation, wiedergespiegelt durch Z , verschlechtern sie. Die Einkopplungseffizienz η ist immer schlechter als η_l , da $\eta = \eta_l \cdot \exp(-\eta_l \cdot x)$. Mit der Exponentialfunktion immer kleiner 1, da η_l positiv (zwischen 0 und 1) und in x alle Terme positiv sind, denn bei großen Z unterdrückt η_l das Vorzeichen des letzten Terms und der Exponent geht gegen 0.

Das Glasfaserkabel wird allerdings nicht nur im Querschnitt, sondern flächig vom Strahl getroffen. Somit gilt es in zwei Dimensionen jeweils θ und x_0 zu optimieren. Dies bedarf vier Freiheitsgrade. Die Justierung wird einfacher, je unabhängiger diese sind. Zwei Spiegel im Strahlengang mit jeweils verstellbaren x- und y-Positionen erfüllen diese Anforderung.

Die Laborerfahrung zeigt, dass die Positionen gewissen Drifts unterliegen, die eine erneute Optimierung nötig machen. Dadurch verschlechtert sich jede durchgeführte Optimierung auf Zeitskalen von Tagen bis Wochen merklich und der Laborant muss die Einkopplung wiederholen.

2.1.3 Manuelle Optimierung

Bei der manuellen Justage wird typischerweise der Algorithmus „beam walk“ verwendet. Eine Analyse dazu findet man bei [2].

Der verwendete Aufbau ist hier dargestellt:

¹Die Größenordnungen außer der Wellenlänge des Lasers sind grob abgeschätzt. Als Anhalt diene das Datenblatt des Kollimators http://www.sukhamburg.com/products/Fiber_collimators_60FC-_.html

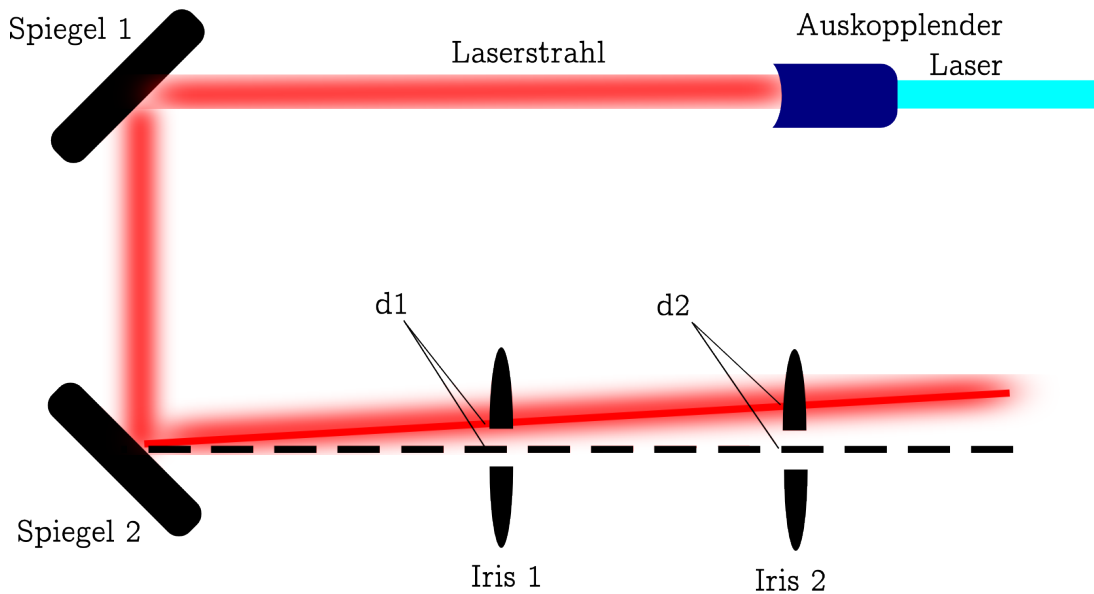


Abbildung 2.3: Skizze für den Aufbau „Beamwalk“ vgl [2]

Optimiert abwechselnd Spiegel 1 den Strahl auf $d_1 = 0$ und dann Spiegel 2 auf $d_2 = 0$, konvergiert nach mehreren Iterationen sowohl d_1 als auch d_2 gemeinsam gegen 0.

Nach [2] lässt sich die Konvergenz beschleunigen, wenn statt der Optimierung auf $d_i = 0$ kleine Übersteuerungen in einer Achse zugelassen werden, da die Größen korreliert sind.

2.1.4 Kommerzielles System

Es gibt kommerzielle Systeme, die die Optimierung der Einkopplung übernehmen. Beispielsweise genannt sei hier „Fiberlock“ von TEM-Messtechnik:

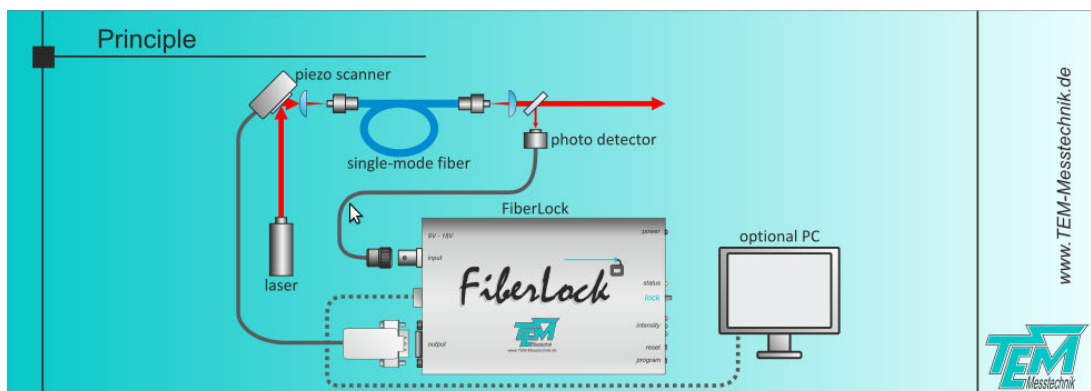


Abbildung 2.4: Skizze für den Aufbau „Fiberlock“ von vgl [3, S. 1]

Bei diesem System werden die Spiegel mit Piezoelementen gesteuert und auch die Kollimation des Strahls optimiert. Dadurch, dass die Piezoelemente so schnell sind, gibt es weitere Funktionen wie 3D-Ansichten des Strahls und Rauschunterdrückung („Noise Eater Mode“).

Bewerkstelligt wird dies durch PID-Algorithmen.
Kosten nach [4] sind:

Komponenten	Beschreibung	Einzelpreis
FiberLock	Lock Elektronik mit Input- & Outputverstärkern	1200.- €
BeamScanLv	2D Piezo Strahlscanner mit großem Radius	800.- €
BeamScanOneInch	2D Piezo Spiegelscanner für 1“ Spiegel	1000.- €
PDA-S	Optional! Photodiodenverstärker	500.- €
Photodiode		
Weitere Bauteile	Strahlteiler etc.	

Tabelle 2.2: Kosten TEM-System zusammengestellt von [4]

Damit liegt der Aufbau bei mehr als 2000.- € pro zu koppelndem Laser. Die Motivation der Bachelorarbeit ist es, ein günstiges System zu entwickeln.

2.2 Optimierung

Standardverfahren zur Optimierung von Gaußkurven wie das Gradientenverfahren und ähnliche erfordern eine sehr genaue Positionsänderungen auf der X-Achse. Im später erklärten Aufbau werden aber Motoren verwendet, deren Positionsänderungen aus im Experiment analysierten Gründen nicht sehr genau vorhersagbar sind.

Somit entfallen alle Algorithmen, die kompliziertere Bewegungen auf der X-Achse erfordern, als nur einen Schritt in eine Richtung zu gehen.

2.2.1 Bergsteigeralgorithmus

Algorithmus nach [5]:

- Es wird in eine Richtung gegangen.
- Wird der Wert signifikant schlechter, wird die Richtung umgekehrt.
- Nach $x \geq 2$ Richtungswechseln terminiert der Algorithmus.

Der Algorithmus ist gut geeignet für Funktionen mit einem Maximum das in beiden Richtungen monoton fallend ist. Bei Nebenmaxima konvergiert der Algorithmus möglicherweise auf dieses. Wir erwarten Gaußverteilungen nach Formel 2.1, somit sind Nebenmaxima nicht zu erwarten.

Je nach Signifikanz terminiert der Algorithmus bei einem Wert bis zu einigen Sigma unter dem Maximum. Dies ist insbesondere beim 4-dimensionalen Optimieren zu beachten.

Erweiterungen des Algorithmus sind unter anderem:

Sinnflutalgorithmus: Schlechtere Werte können akzeptiert werden, wenn sie eine Schwelle nicht unterschreiten. Die Schwelle steigt.

Simuliertes Auskühlen: Schlechtere Werte können mit einer abnehmenden Wahrscheinlichkeit angenommen werden.

Schwellenakzeptanz: Das Maß der Verschlechterung fließt in die Wahrscheinlichkeit mit ein.

2.2.2 PID-Regler

Mit Hilfe von Regelungstechnik kann man die Intensität nicht nur optimieren, sondern auch stabilisieren. Für den laufenden Betrieb wäre es für die Kopplung praktisch, wenn sie nach der Optimierung stabilisiert wäre, statt die Optimierung zu wiederholen.

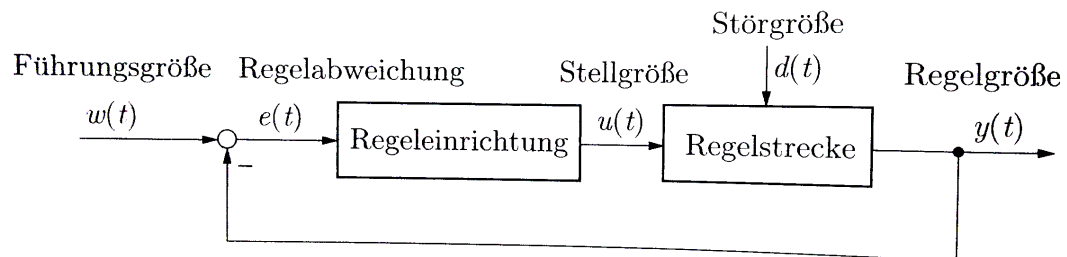


Abbildung 2.5: Grundstruktur eines Regelkreises [6, S. 4]

Hierbei wird die Regelgröße (für uns: die Intensität) kontinuierlich gemessen und nach Regelabweichung (bei uns: Abweichung vom Maximum) über die Regeleinrichtung (Mikrocontroller) eine Stellgröße (Spiegelposition) verändert. Die Regelstrecke beschreibt das Stellverhalten und das Störverhalten der Regelung. Die Regelung wirkt der Abweichung entgegen.

Beliebt und auch in der kommerziellen Version von TEM-Messtechnik umgesetzt ist das Konzept von PID-Reglern. Hierbei wird auf Abweichungen sowohl **p**roportional, **i**ntegrierend als auch **d**ifferenzierend reagiert.

Bei einfachen Motoren gilt es eine nicht ganz triviale Abhängigkeit zwischen Spannung am Motor und der Veränderung der Spiegelposition zu finden. Im späteren Experiment sind dazu viele nicht herausrechenbare Probleme beschrieben (Winkelabhängigkeit, Hysterese etc). Dadurch konnte in diesem Aufbau kein PID-Algorithmus implementiert werden.

2.3 Bewegungssystem der Spiegel

Der Neigungswinkel der Spiegel setzt sich in Winkel- und Positions-Disposition um. Er kann von Hand per Stellschraube oder durch einen Motor mit Koppler bewegt werden. Für Mikrocontroller gibt es verschiedene Systeme, um die Spiegel zu bewegen, die unten aufgezählt sind.

2.3.1 Motoren mit Pulsweitenmodulation

Zunächst kann ein einfacher Motor die Stellschraube bewegen.

Bei Motoren mit Pulsweitenmodulation wird die Bewegungsgeschwindigkeit verringert, indem nicht die Spannung reduziert wird, sondern die Spannung über viele Takte nur einige wenige Anteile des Taktes ausgeschaltet wird.

2.3.2 Steppermotoren

Es gibt Motoren, die sich nicht in einem kontinuierlichen Bereich bewegen, sondern in gewissen Schritten. Dadurch lässt sich die aktuelle Position des Motors besser bestimmen. Dadurch ist die x-Achse kalibriert.

2.3.3 Piezos

Elektrische Spannung kann in mechanische umgesetzt werden. Die Position der Spiegel ist damit linear von der anliegenden Spannung abhängig. Dies ist viel genauer als einfache Motoren und praktisch für die Implementierung von PIDs, da von diesen eine Stellgröße, die proportional zur Änderung ist, gefordert wird.

Eine Piezozelle kann auch ein Stellsystem bewegen. Damit kann die Reichweite, die sonst von der Ausdehnung der Zelle begrenzt ist, erweitert werden.

Motoren mit diesem Prinzip sind kommerziell zu erwerben. Der Preis von günstigen Piezos ist allerdings viel höher als der von einfachen Motoren.

Kapitel 3

Experiment

3.1 Aufbau

Um die Einkopplung zu automatisieren, ist es erforderlich, einen Aufbau zu finden, an dem die Einkopplung optimiert und vom Mikroprozessor gesteuert werden kann.

3.1.1 Erklärte Skizze

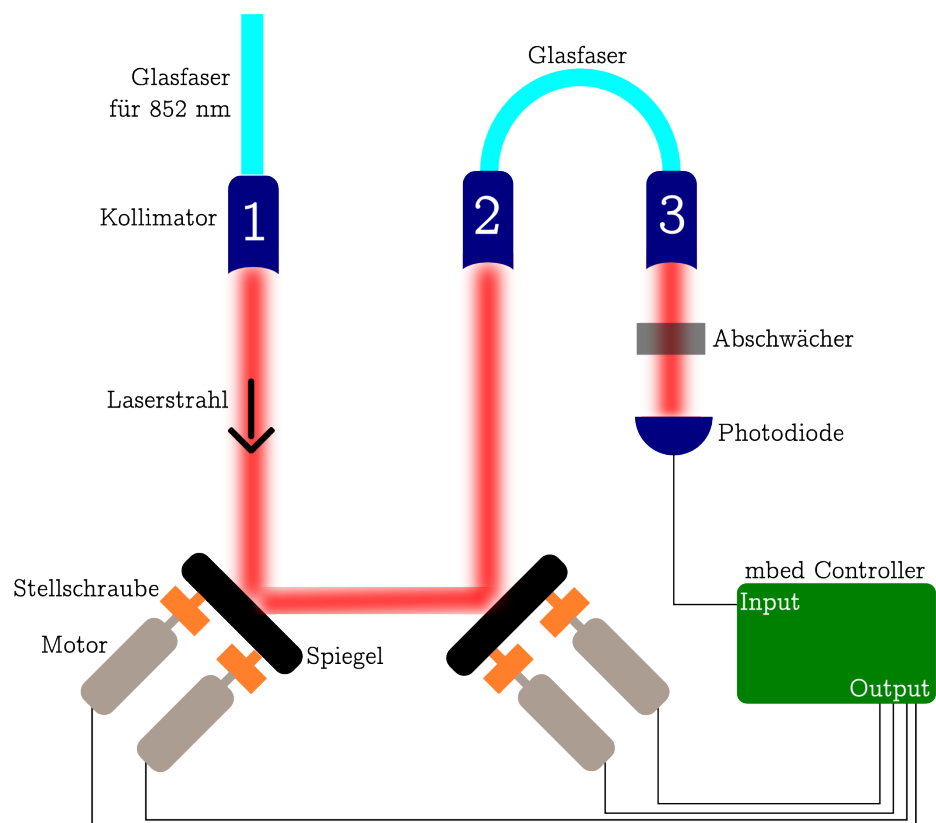


Abbildung 3.1: Skizzierte automatisierte Fasereinkopplung¹

¹Ein entsprechendes Foto befindet sich im Anhang A.1

Ein Laserstrahl, bereitgestellt durch Kollimator 1, soll in Kollimator 2 eingekoppelt werden. Für die in der Theorie geforderten Freiheitsgrade stehen zwei Spiegel im Strahlengang. Die Position der Spiegel wird über einen mbed²-Mikrocontroller via vier Motoren gesteuert. Über H-Brücken kann der mbed die Motoren in jede Richtung drehen und durch Kurzschließen der Pole auch anhalten, statt sie nur ausfahren zu lassen.³

Der Controller kann die Intensität nach dem Auskoppeln aus Kollimator 3 über eine Photodiode auslesen. Die Photodiode hat einen linearen Bereich von 0 V bis 3 V und ist direkt an den Analogeingang des Mikrocontroller mbed angeschlossen, der eine Auflösung von 0 V bis 3,3 V hat.

Der Laser hat einige (ca. 10) mW Intensität. Die Photodiode hat eine Sensitivität von 0,6 V / μ W. Um also im linearen Bereich der Photodiode zu arbeiten, müssen Abschwächer in den Strahl gestellt werden.

3.1.2 Bauteile

Die verwendeten Bauteile sind hier tabellarisch gelistet. Fotos dazu befinden sich im Anhang A.2.1.

Name	Typbezeichnung
Laser	Laser IFL #001 im DQSIM-Labor
Spiegel	von Thorlabs, 1" Broadband Dielectric Mirror, 750-1100 nm
Spiegelhalterung	von Radient Dyes Laser
Kollimator	Kirchhof 60FC-4-A11
Mikrocontroller	mbed NXP LPC 1768
Board des Mikrocontroller	Hergestellt von Stefan Brakhane ⁴
Motor + Getriebe	Getriebemotor PGM-25DC12/22
Photodiode und Gehäuse	Gehäuse hergestellt von der Werkstatt des IAP ⁵
Halterung Motoren	Hergestellt von der Werkstatt des IAP ⁶
Power Supply	Laboratory Power Supply EA-PS 3016-20B
Oszilloskop	Agilent Technologies InfiniiVision MSO-X 2024A

Tabelle 3.1: Bauteile

²Ein mbed ist die verwendete Entwicklungsplattform mit einem Mikroprozessor. Weitere Informationen unter <https://mbed.org/>

³Das dazugehörige Board wurde von Stefan Brakhane entworfen und gebaut. Die Schaltpläne befinden sich im Anhang A.8

⁴Siehe dazu Anhang A.2.4

⁵Siehe dazu Anhang A.2.3

⁶Siehe dazu Anhang A.2.1

3.2 Ein Schritt

Zunächst gilt es zu bestimmen, welche Bewegungen der Motoren sich hervorrufen und messen lassen. Gibt man einem Motor kontinuierlich Spannung, bewegt er sich für eine Regelung zu schnell.

In einem einfachen Test wurde das Intensitätsprofil für einen Motor abgefahren. Bei Spannungen von ca. 6 V bis 9 V war die Bewegung zu schnell, um das Profil vom Mikrokontroller aufzulösen. Unter einem Schwellwert von ca. 6 V bewegte sich der Motor gar nicht. Für eine maschinelle Verarbeitung ist eine hohe Auflösung gewünscht.

Um dies zu erreichen, kann man den Motoren kurze Spannungspulse (10 ms bis 50 ms bei 9 V, variiert aber je nach Motor) geben, ihn direkt danach anhalten und dann messen.

3.2.1 Rauschen und Mittelung

Um spätere Messwerte beurteilen zu können, benötigt man eine Abschätzung des Signalrauschens. Dieses setzt sich aus Intensitätsrauschen des Laserstrahls und Rauschen in der Elektronik und Datenverarbeitung des mbed zusammen. Um eine größtmögliche Zeitauflösung zu erhalten, wurden 1000 Intensitätswerte der Photodiode am mbed zuerst abgespeichert und danach ausgelesen.

Um das Streulicht zu reduzieren, wurde eine Abdeckung aus Pappe gebastelt, die über Auskoppler und Photodiode gestellt werden kann. In unserer Abbildung 3.2 sind die Daten einmal mit und einmal ohne diese Streulichtabschirmung aufgetragen. Die Mittelung und Bestimmung der Varianz ist nach der χ^2 -Methode.

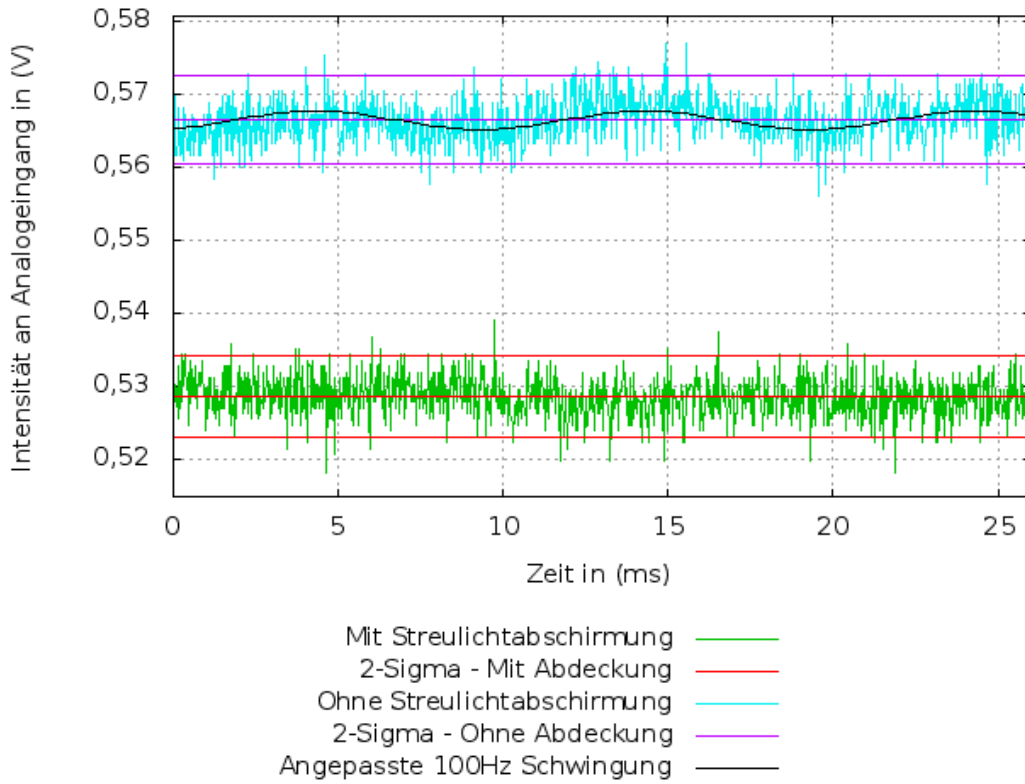


Abbildung 3.2: Rauschbestimmung⁷

Man sieht, dass die Streulichtabschirmung die Intensität insgesamt reduziert. Zu erkennen ist davon auch das Intensitätsschwanken der Deckenbeleuchtung bei 100 Hz. Diese stört die Messung, so dass in den weiteren Messungen die Abschirmung weiter verwendet wurde.

Der Speicher des Mikrokontroller ist begrenzt und jeder Speicheraufruf kostet Zeit. Um eine hohe Auflösung zu erhalten, sollten möglichst wenig Werte gespeichert werden.

Einen Algorithmus, der die Messwerte mittelt und dazu die Standardvarianz ausgibt ohne die Dateien speichern zu müssen, stellt [7] vor. Definiert man x_k als den k -ten Messwert und:

$$M_k = \begin{cases} x_1 & k = 1 \\ M_{k-1} + \frac{x_k - M_{k-1}}{k} & k = 2, \dots, n \end{cases}$$

$$Q_k = \begin{cases} 0 & k = 1 \\ Q_{k-1} + \frac{(k-1)(x_k - M_{k-1})^2}{k} & k = 2, \dots, n \end{cases}$$

ist für jedes n M_n der Mittelwert aus allen Werten davor und $s_n^2 = Q_n / (n - 1)$ die entsprechende Stichprobenvarianz.

⁷Aufgenommen am mbed nach Aufbau 3.1

Nimmt man bei jedem Auslesen 1000 Werte und verarbeiten sie nach diesem Algorithmus, dauert das mit dem mbed ca 0,1 s. Damit ist gewährleistet, dass mindestens über eine Periode der Schwankung der Intensität der Deckenbeleuchtung gemittelt wird.

Die so ermittelte Standardabweichung gibt einen zuverlässigen Wert zur weiteren Berechnung an. Deswegen wurden in den folgenden Schritten alle Werte so bestimmt. Um die Fehlergrenzen zu verkleinern, könnte man darüber hinaus die eingelesenen Werte noch gaußgewichten. Dies ist nicht realisiert, da jede Minimalbewegung der Motoren in der Nähe des Intensitätsmaximums die Werte weit über die Fehlergrenzen hinaus verändert.

3.2.2 Wartezeit nach einem Schritt

Um einen Motor zu stoppen, gibt es für den mbed die Möglichkeit, die Pole des Motors kurzzuschließen, statt ihn nur ausfahren zu lassen. Dennoch hält der Motor aufgrund seiner Trägheit und der des Getriebes nicht instantan an. Um die Verzögerungsdauer nach einem Spannungspuls an einem Motor zu bestimmen, wird mit der maximalen Einleserate des mbed (ca. 8 Werte pro 10 ms) die Intensität gemessen, indem für alle zehn eingelesenen Werte ein Mittelwert ausgegeben wird.

Dazu wird zunächst die Transmission optimiert und dannach wird die zu messende Motorstellung so verdreht, dass alle Messwerte sich auf einer Flanke der Intensität befinden. Die Änderungen sind somit gut ablesbar.

Alle 0,3 s gibt der mbed einen Spannungspuls an dem Motor. Während dieses Pulses wird die Intensität nicht gemessen. Die Messwerte dazu sind hier aufgetragen:

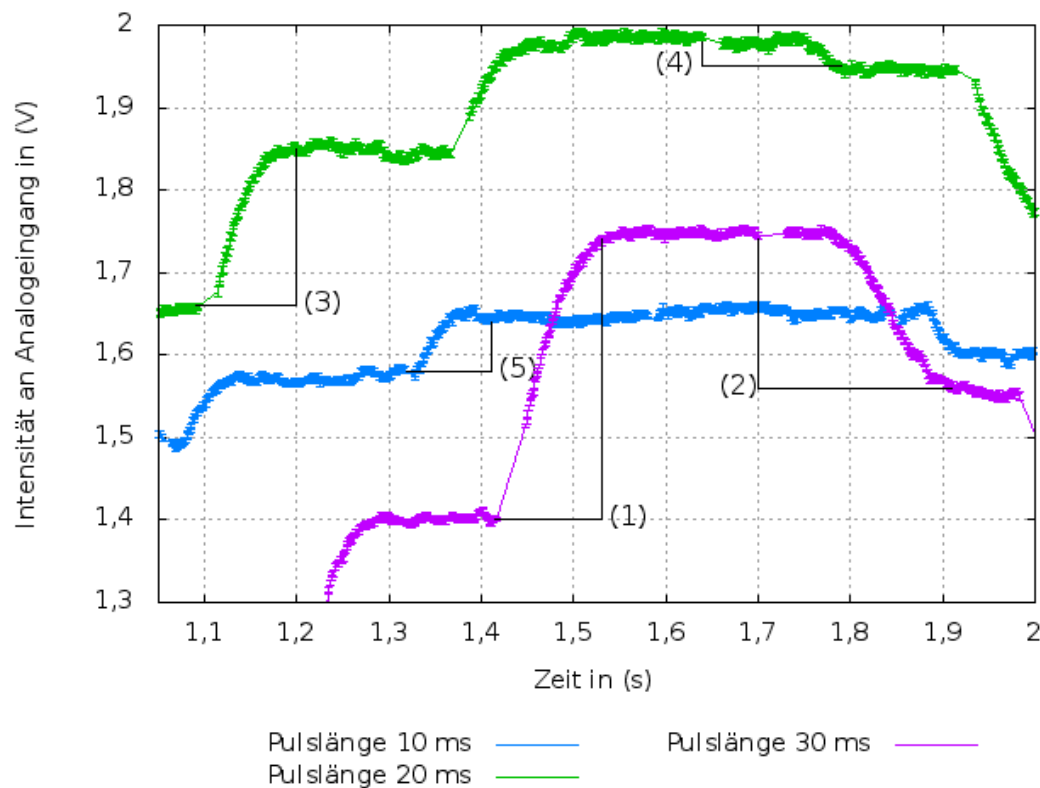


Abbildung 3.3: Wartezeitbestimmung mit verschiedenen Pulslängen⁸

Aus dieser Messung 3.3 lassen sich die Änderungen abschätzen:

Pulsdauer	Nummer	Änderungsdauer
ms		ms
30	(1)	120
30	(2)	210
20	(3)	90
20	(4)	110
10	(5)	90

Dies zeigt, dass man für die verwendeten Pulsdauern mindestens 100 ms warten muss. Bei (2) und (4) fand zudem ein Richtungswechsel des Motors statt, sodass es aufgrund des Spiels des Getriebes Zeit dauert, bis sich der Puls in einer Intensitätsänderung widerspiegelt. Damit ist es nach einem Puls nicht ohne weiteres möglich, einfach so lange auf Werte zu warten, bis sie sich nicht mehr signifikant ändern.

⁸Aufgenommen am mbed nach Aufbau 3.1

3.2.2.1 Richtungswechsel

Im Folgenden werden Richtungswechsel genauer betrachtet. Dazu wird die Intensität von einer Seite auf eine bestimmte Höhe (ca. 1,3 V) gefahren. Nach einer Sekunde wird ein Puls in die andere Richtung gegeben. Die Intensitäten wurden kontinuierlich einzeln ausgegeben.

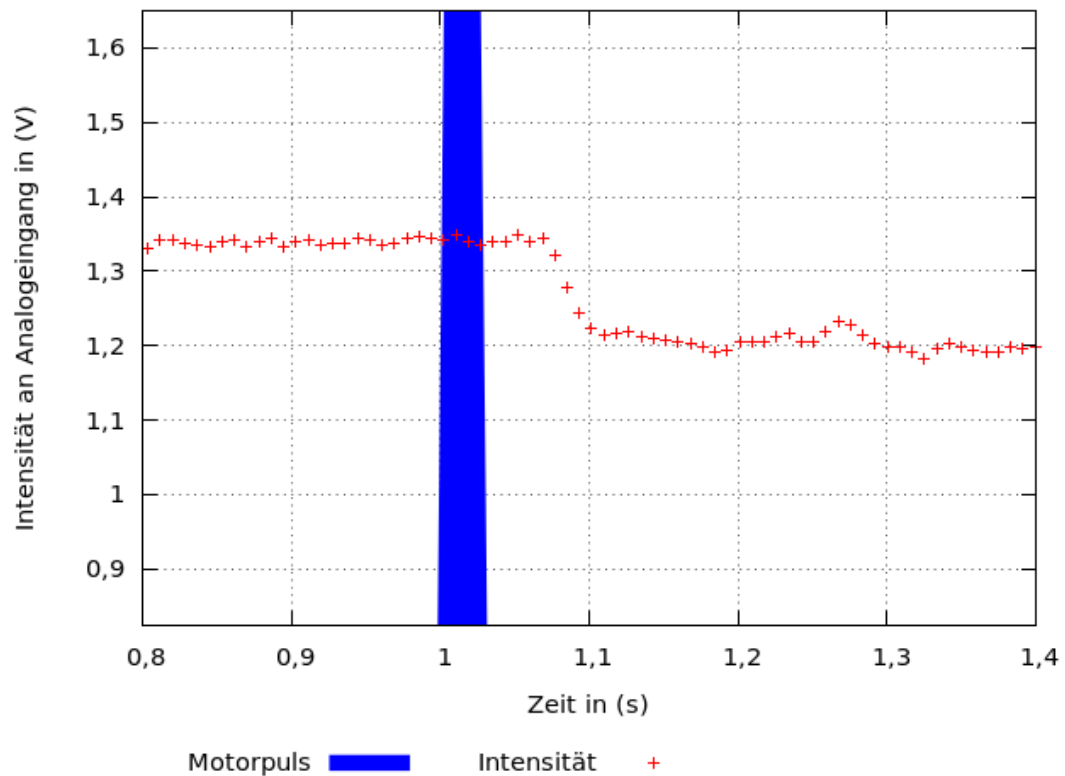


Abbildung 3.4: Wartezeitbestimmung mit Richtungswechsel bei 20 ms Puls ⁹

⁹Aufgenommen am mbed nach Aufbau 3.1

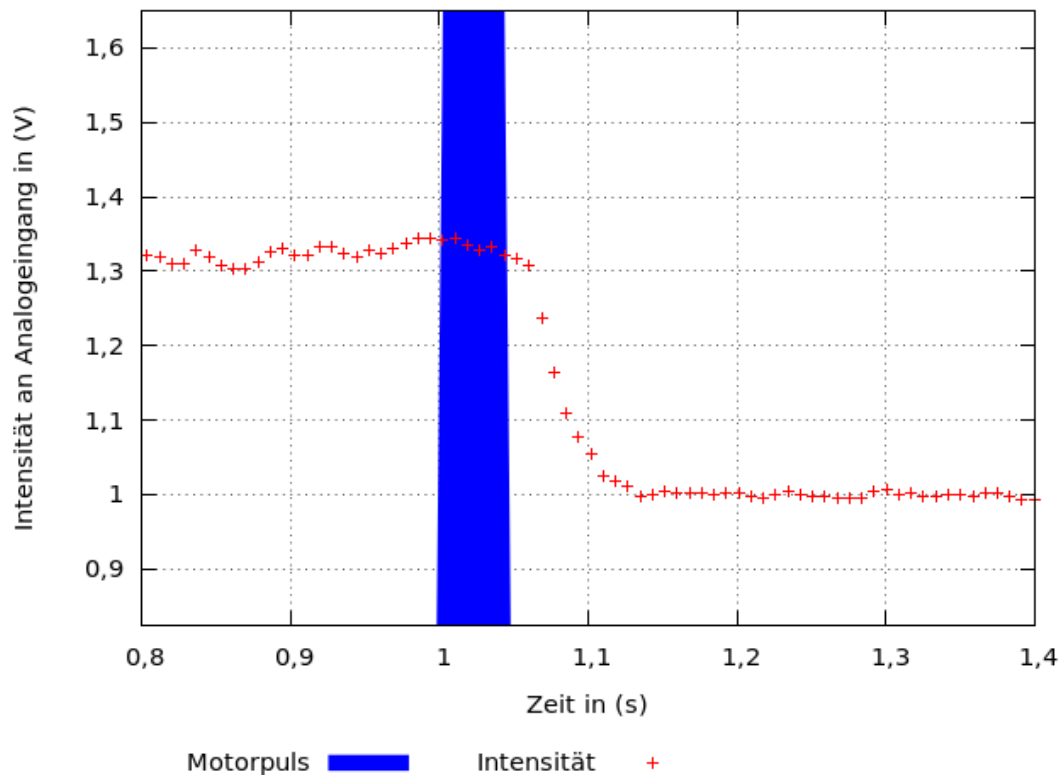


Abbildung 3.5: Wartezeitbestimmung mit Richtungswechsel bei 40 ms Puls ¹⁰

Auch hier lässt sich ablesen, dass immer mindestens 100 ms gewartet werden muss, bis ein Messwert eingelesen werden kann. Um Fällen wie in Tabelle 3.2.2 Label (2) gerecht zu werden, wird im Folgenden jeder Wert erst nach der zehnfachen Pulsdauer und mindestens 100 ms später eingelesen.

Für viele Algorithmen ist es nötig, zu einem vorher angenommenen Punkt zurückzukehren. „Verwirft“ man einen Wert, soll die Apparatur die vorherige Ausgangsstellung wieder einnehmen. Durch das Spiel des Getriebes und dadurch, dass die Motoren eine gewisse Vorzugsrichtung haben, bringt ein entgegengesetzter Puls sie aber nicht in die Anfangsposition.

Um das zu sehen, wird nach einer Optimierung ein großer Schritt mit einem Motor auf die Flanke gemacht. Aus dieser Position heraus werden ständig Pulse in abwechselnde Richtung gegeben:

¹⁰Aufgenommen am mbed nach Aufbau 3.1

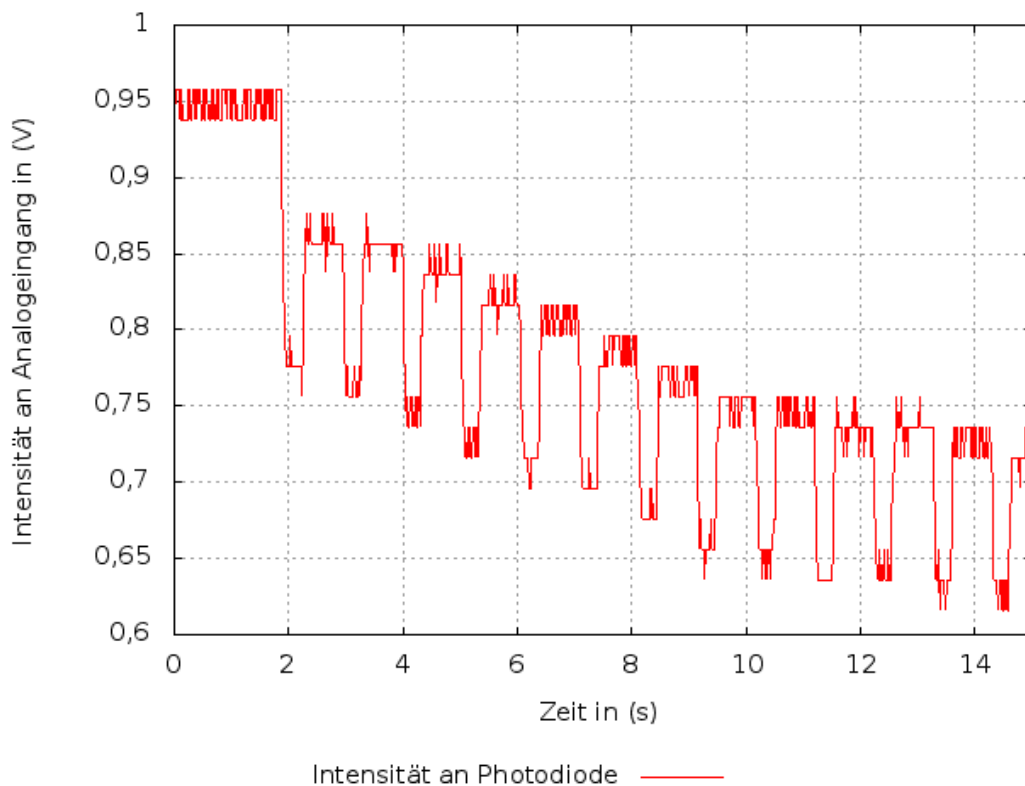


Abbildung 3.6: Pulse in abwechselnde Richtung ¹¹

Man sieht in dieser Messung 3.6, dass nach dem ersten Richtungswechsel (bei ca. 2,5 s) nicht wieder der Ursprungswert von ca 0,95 V angenommen wird, sondern nur 0,88 V. Dies ist auch für andere Pulslängen und Wartezeiten so.

Zusätzlich ist in diesem Beispiel die momentane Vorzugsrichtung anhand des Drifts nach vielen Richtungswechseln erkennen. Einflüsse wie diese Vorzugsrichtung und leichter Bewegung bei bestimmten Winkeln durch Asymetrien der Schraubenkopplung verhindern, dass der mbed bei Richtungswechseln einfach die Pulsänge um einen Faktor verändert, um der Hysterse entgegenzuwirken.

3.2.3 Algorithmus „Ein Schritt“

Da das Verhalten der Motoren weder leicht vorhersagbar noch ihre Position direkt messbar ist, ist es am erfolgsversprechensten, nur kleine Änderungen der Stellschraubenwinkel hervorzurufen. Nach jedem dieser Schritte kann neu entschieden werden, wie der Algorithmus sich weiter verhalten soll. Ein Schritt besteht nun aus folgenden Teilen:

¹¹ Aufgenommen am Oszilloskop nach Aufbau 3.1

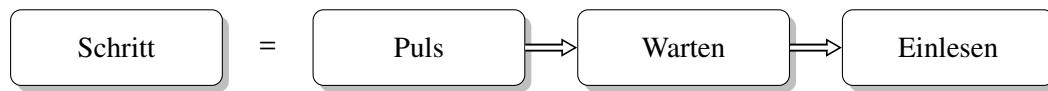


Abbildung 3.7: Darstellung zur Erklärung des Algorithmus „Ein Schritt“

Der Motor erhält einen Spannungspuls von 10 ms bis zu 30 ms, denn in diesem Bereich ist eine Bewegung gewährleistet, aber die Auflösung der abzufahrenden Gaußkurve noch groß genug, um ein Maximum nicht durch einen großen Schritt zu überspringen. So ein Spannungspuls entspricht nur einiger weniger Grad Drehung der Schrauben und ist die unterste Grenze der möglichen Bewegung der Motoren.

Entsprechend Kapitel 3.2.2.1 muss vor dem Einlesen gewartet werden, bis man eine neue Intensität bestimmen kann. Die Zeit entspricht dem Zehnfachen der Pulsdauer und mindestens 100 ms.

Danach werden 1000 Werte über ca. 100 ms eingelesen und gemittelt.

Gibt der mbed die Daten auch noch an einen Computer aus, dauert ein Schritt ca. 0,3 s bis 0,5 s.

3.3 Scan mit einem Motor

In dieser Schrittweise lassen sich also die Intensitätsprofile abfahren. Dazu wurde die Intensität in Aufbau 3.1 zunächst maximiert. Ein Motor fährt nun in Spannungspulsen vom Maximum in ein festgelegtes Minimum. Dort wechselt er die Richtung und fährt das Profil von Minimum zu Minimum ab:

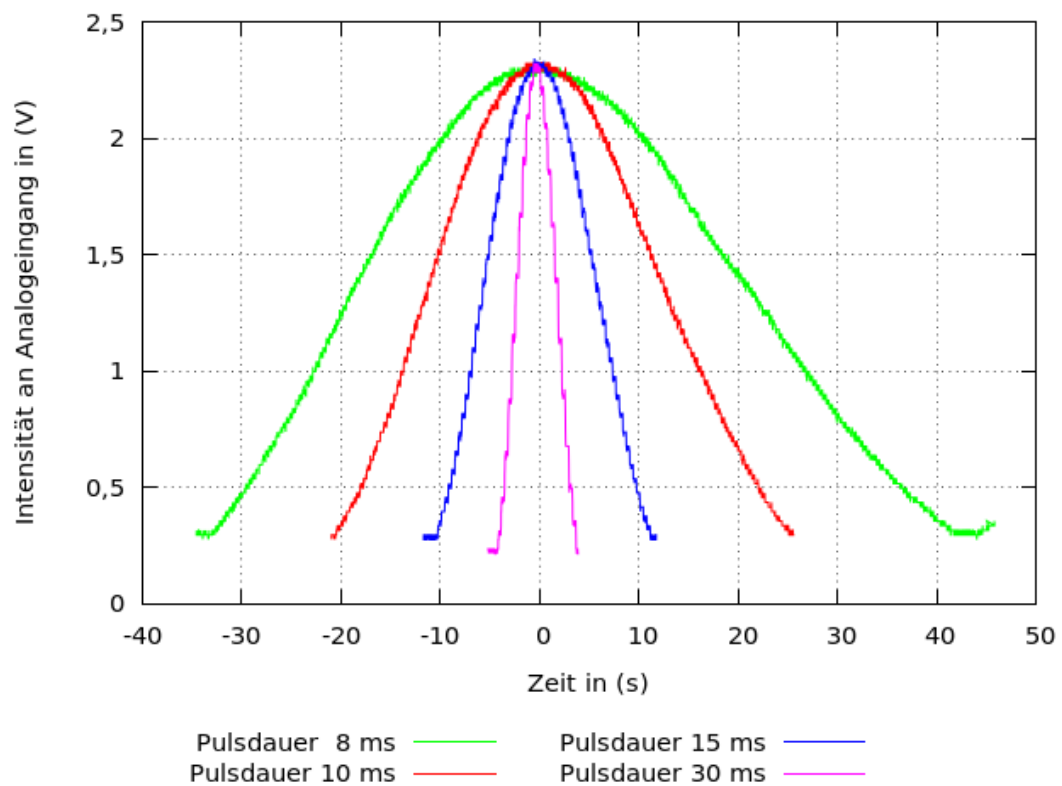


Abbildung 3.8: Intensitätsprofil in einer Dimension¹²

Bei Pulsweiten unter 10 ms kann nicht immer eine Bewegung gewährleistet werden. Für eine hohe Auflösung wählen wir also 10 ms und beachten auch die Wertbereiche der Photodiode. Es gibt einen variierenden Offset, der mit Streulichtabschirmung meist bei 0,1 V bis 0,3 V liegt.¹³

¹²Aufgenommen am Oszilloskop nach Aufbau 3.1

¹³Zur Erinnerung: Der lineare Bereich der Photodiode ist bei 0 V bis 3,3 V

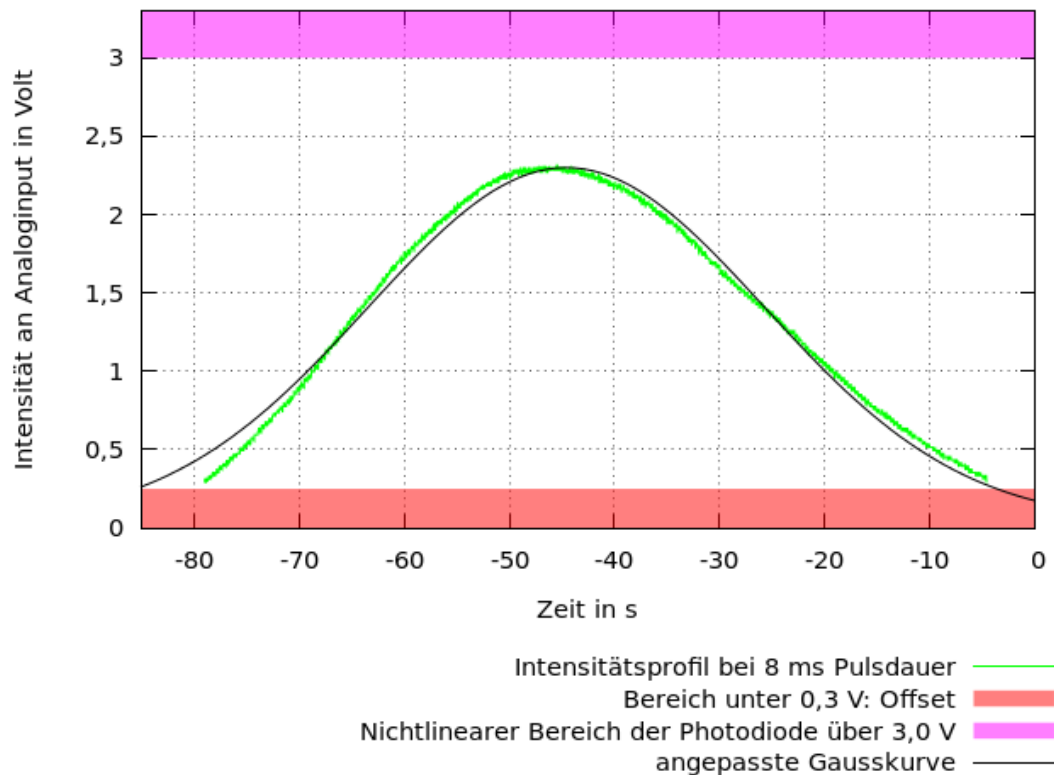


Abbildung 3.9: Intensitätsprofil mit Gaußanpassung und Bereichen der Photodiode¹⁴

Das Profil entspricht einer Gaußkurve (vgl Formel 2.1) mit

$$Gauss(x, \mu, \sigma, V, y) = \frac{V}{\sigma \sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2} \left(\frac{x - \mu}{\sigma}\right)^2\right) + y$$

$$\sigma = 18,85 \pm 0,06$$

$$\mu = -44,569 \pm 0,017$$

$$y = 0,0367 \pm 0,0071$$

$$V = 106,93 \pm 0,63^{15}$$

Die leichte Verzerrung entsteht aus dem Korrelationsterm der Größen in Formel 2.1.

3.3.1 1D-Optimierung

Es gilt zunächst, für eine Dimension ein Optimum zu finden. Diese Optimierung lässt sich auch in Algorithmen wie „Beamwalk“ (vergleiche Kapitel 2.1.3) einsetzen, da für „Beamwalk“ immer abwechselnd eine Dimension optimiert wird.

¹⁴Aufgenommen am Oszilloskop nach Aufbau 3.1

¹⁵Die Werte wurden in Anpassung der Daten an die genannte Formel von gnuplot berechnet

Als einfachstes wird zuerst der „Bergsteigeralgorithmus“ (vergleiche Kapitel 2.2.1) in den Aufbau 3.1 implementiert. Dazu wird zunächst über das Optimum hinweggefahren, um sicher zu sein, ein lokales Optimum zu haben. Wird auf dem Rückweg das letzte Optimum wieder erreicht, terminiert der Algorithmus.

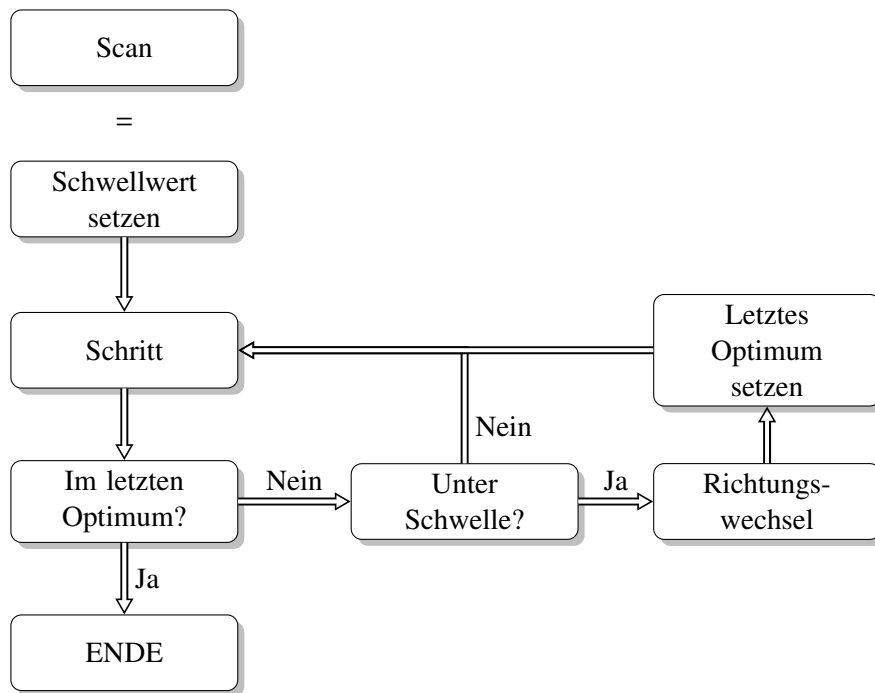


Abbildung 3.10: Darstellung zur Erklärung des Algorithmus „Scan“. Die Erklärung findet sich im folgenden Text

Erklärung: Zu Beginn wird ein Schritt ausgeführt, in dem die Richtung der positiven Flanke bestimmt wird und ein Schwellwert gesetzt wird, bei dessen Unterschreitung später die Richtung gewechselt wird.

Daraufhin wird die positive Flanke abgefahren und der jeweils größte Wert als lokales Optimum gespeichert. Fällt die Flanke wieder, dies ist der Fall, wenn mehrere (fünf) Messwerte hintereinander eine abnehmende Intensität aufweisen, oder wird der Schwellwert unterschritten, wird das letzte Optimum als Wert gesetzt und die Richtung gewechselt.

Wird bei neuem Abfahren das letzte Optimum erreicht, ist das Optimum sicher ein Optimum und der Algorithmus terminiert.

Wird es aufgrund eines unerwarteten Effektes nicht erreicht, fährt er weiter bis er wieder die Richtung wechselt, setzt das nun letzte bekannte Optimum und fährt an dem Punkt fort, an dem er das letzte Optimum sucht. Dadurch ist der Algorithmus gegenüber Störungen stabil, die z.B. dadurch entstehen, dass die Streulichtabschirmung nicht perfekt ist und ein sich in der Nähe bewegendes Laborant die Intensität auf der Photodiode ändert.

3.3.2 Messung Algorithmus „Scan“

Beispielhaft sei hier ein Scan vorgeführt:

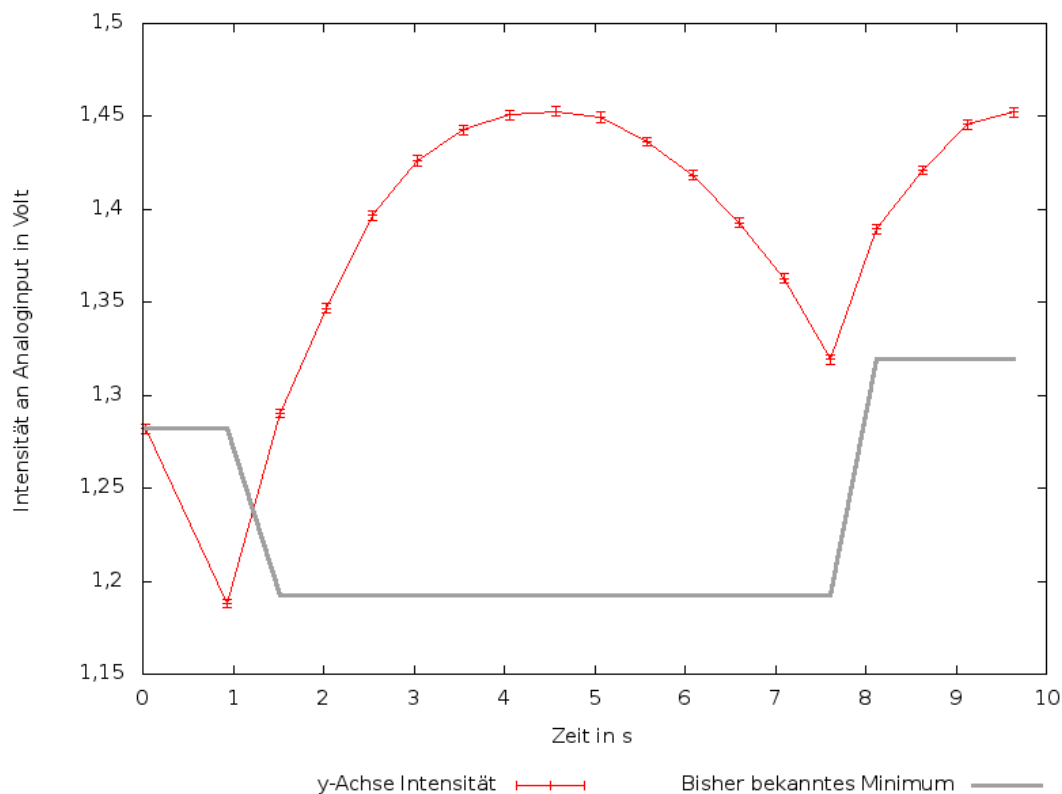


Abbildung 3.11: Eine Messung des Algorithmus „Scan“

¹⁵ Aufgenommen am mbed nach Aufbau 3.1

Die Messung 3.11 wird betrachtet: Im ersten Schritt (Sekunde 0 zu Sekunde 1) werden Schwellwert und Richtung der Flanke bestimmt.

Danach wird einmal die Gaußkurve abgefahren (Sekunde 1 bis Sekunde 7,5). Nach fünf Messwerten mit fallender Intensität (Sekunde 5,5 bis Sekunde 7,5) wird der Schwellwert neu gesetzt und die Richtung gewechselt. Hier ist nach insgesamt 19 Schritten das vorherige Optimum erreicht und der Algorithmus terminiert.

Die Abschätzung von ca. 0,5 s pro Schritt bestätigt sich hier noch, und dies bei einer geringen Pulsdauer von 10 ms. Die meiste Zeit pro Schritt geht hier bei der Kommunikation zwischen mbed und Computer verloren.

Die Laufzeit von 10 s für einen Algorithmusdurchlauf kann dabei durchaus als typisch angesehen werden, denn es werden mindestens fünf Schritte ins Negative gemacht, die wieder rückgängig gemacht werden müssen. Zusammen mit den Schritten davor sind es im Durchschnitt 20 Schritte beim wiederholten Aufrufen des Algorithmus „Scan“ in der Nähe des Optimum.¹⁶

Damit braucht „Scan“ im Durchschnitt 10 s. Kann ein Schritt auf 0,1 s reduziert werden (unterste Grenze im Kapitel 3.2.3) und reduziert man die Schritte ins Negative auf drei, lässt sich die Laufzeit auf mindestens eine Sekunde abschätzen.

3.4 Scan über eine Achse

Um eine Achse zu optimieren, muss der Korrelationsterm minimiert werden. Dazu reicht es nicht, die Motoren einfach abzuwechseln, denn die Verbesserungen sind so gering, dass der Algorithmus in jedem Fall zu früh terminiert. Deswegen wird einer der Motoren als Raster, der andere als Scanmotor benutzt.

Der Scanmotor scant nach jedem Schritt des Rasters nach 3.3.1. Das Raster an sich ist auch eine Gaußkurve. Den Algorithmus „Scan“ kann man also auch auf der großen Zeitskala verwenden, um eine Achse vollständig zu optimieren. Erweitert wird der Algorithmus dadurch, dass nach jedem Schritt des Rastermotors der Scanmotor scant.

¹⁶Um dies zu bestimmen wurde die Zeit pro Durchlauf des noch folgenden Algorithmus „Scan über zwei Achsen“ gemittelt (400 Scans)

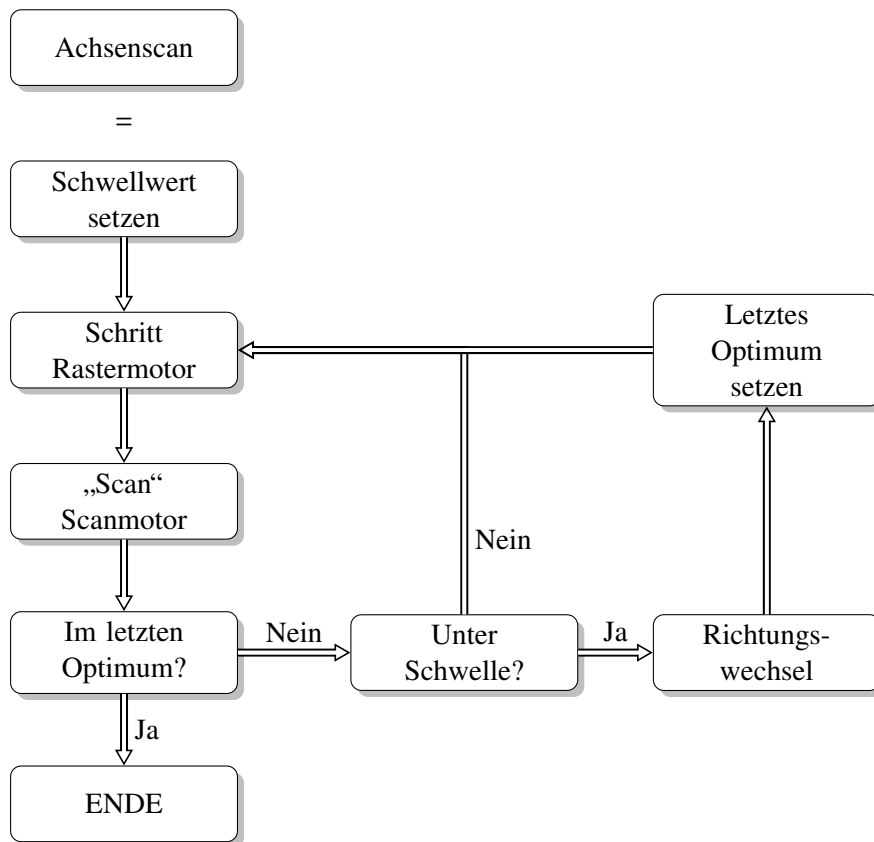


Abbildung 3.12: Darstellung des Algorithmus „Achsen-scan“. Er entspricht „Scan“, nur dass nach jedem Schritt ein Scan des anderen Motors der Achse durchgeführt wird

3.4.1 Messung Achsen-scan

Der Rastermotor kann viel größere Schritte machen, da der Scanmotor seine Disposition ausgleicht. Realisiert wurde dies mit einer bis zu fünffachen Schrittweite. Dabei wurde beachtet, dass Motor (1) (siehe Foto im Anhang A.2) sich von sich aus in den kurzen Pulsbereichen ca. 3x so schnell wie die anderen Motoren dreht.

Zu beachten ist weiterhin, die Schritte dennoch nicht zu groß zu wählen, um nicht aus dem betrachtbaren Bereich der Photodiode herauszufahren.

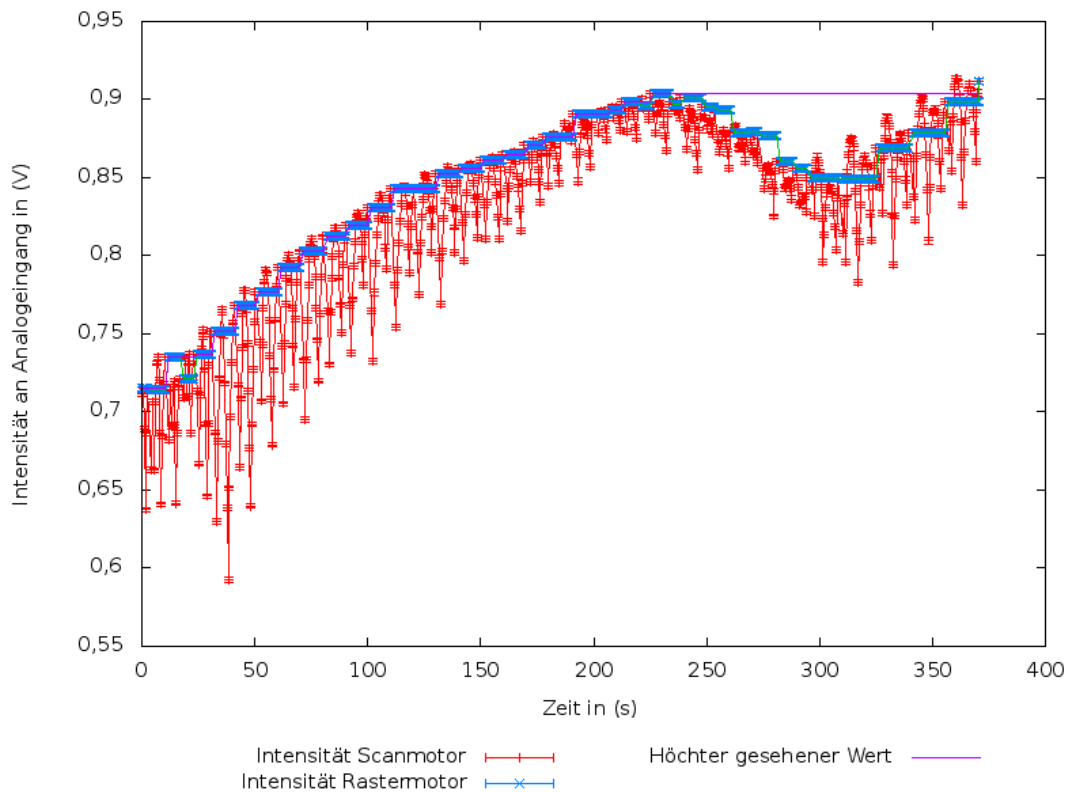


Abbildung 3.13: Eine Messung des Algorithmus „Achsen-scan“

Zu sehen ist in Messung 3.13 die Implementierung des Algorithmus „Achsen-scan“ 3.4. Blau ist der Wert, den der Rastermotor für seine Bewegung kennt. Seine Bewegung entspricht dem Algorithmus „Scan“ 3.3.1. Zwischen jedem Schritt des Rastermotors vollführt der Scanmotor einen ganzen „Scan“, aufgetragen in roten Linien und Punkten. Zur Veranschaulichung ist das bisher gefundene Maximum in lila auch aufgezeichnet. Ab Sekunde 300 ist gut zu sehen, dass jeder Durchlauf des Scanmotors den Referenzwert des Rastermotors erhöht.

3.5 Scan im Zusammenspiel von zwei Achsen

Die Optimierung wird nun durch die Optimierung beider Achsen abgeschlossen.

3.5.1 Algorithmus

Im Idealfall sind die Achsen unabhängig und es lassen sich die Achsen optimieren, ohne dass sie sich beeinflussen. Dies ist im Labor nicht der Fall, weil die Bewegung der Spiegel nicht senkrecht zur Strahlachse stattfindet. Deswegen werden die Achsen abwechselnd optimiert, bis

¹⁶Aufgenommen am mbed nach Aufbau 3.1

keine signifikante Verbesserung mehr eintritt.

Aus praktischen Gründen empfiehlt es sich, zunächst jeweils einen Motor jeder Achse zu optimieren, um einen guten Startwert zu erhalten und die Achse mit dem größeren Verbesserungspotential auszuwählen.

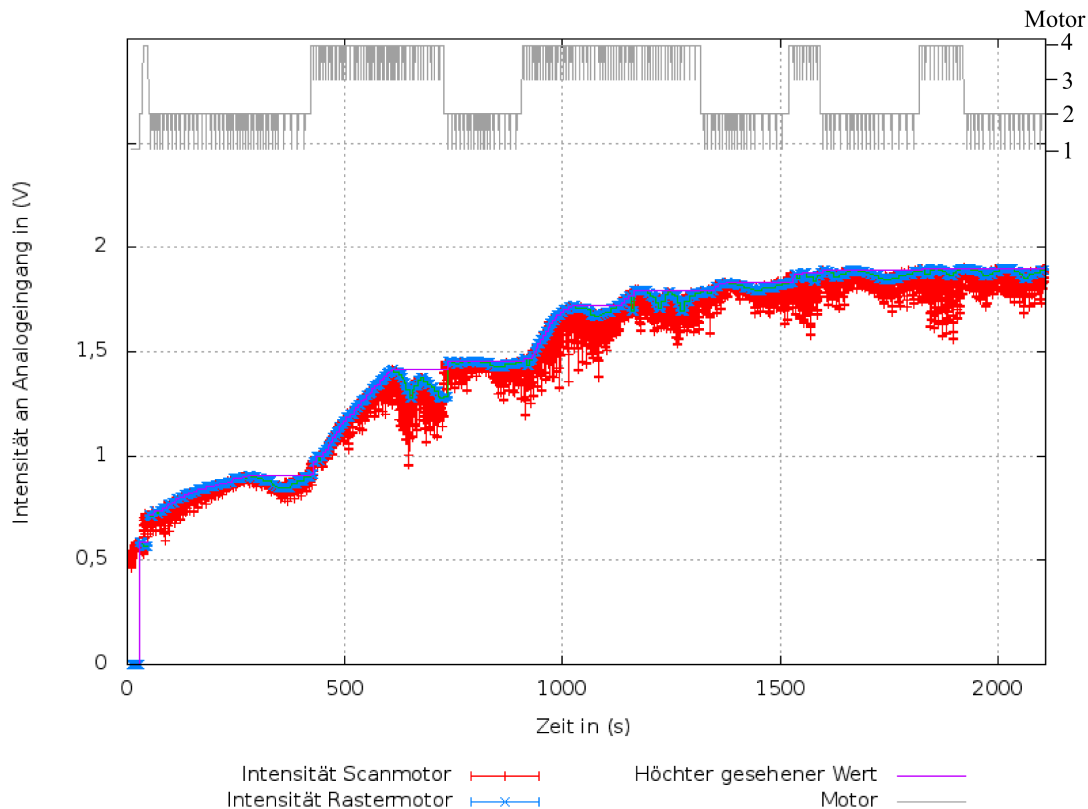


Abbildung 3.14: Eine Messung des Algorithmus „Scan im Zusammenspiel von zwei Achsen“

Zu sehen ist hier wieder der Intensitätsverlauf über die Laufzeit in rot. In blau ist wieder die Intensität nach der Optimierung des Scanmotors, also die Intensität nach der sich der Rastermotor bewegt. Die obere Linie symbolisiert, welcher Motor sich gerade bewegt.

Dadurch lässt sich der Algorithmus gut ablesen:

Zuerst führen alle Motoren den Algorithmus „Scan“ einmalig durch, um von einem guten Startwert zu beginnen.

In den ersten 400 Sekunden wird die erste Achse optimiert, danach bis Sekunde 750 die zweite. Mindestens einer der Motoren zeigte manchmal Intensitätssprünge, die vermutlich durch ein Rutschen von Kappe gegen Stellschraube oder Getriebeispiel ausgelöst werden.

Dies ist in diesem Algorithmus kein Problem, so dass nach mehrmaligem Wechsel der Achsen keine weitere Verbesserung stattfindet und der Algorithmus zum Maximum konvergiert.

Die Laufzeit von 2100 Sekunden (35 Minuten) ist dabei recht lang, was unter anderem daran

¹⁶Aufgenommen am mbed nach Aufbau 3.1

liegt, dass (bewusst) sehr weit vom Maximum entfernt gestartet wurde. Über Nacht fanden 130 Messungen statt, bei denen nach der Maximierung die Motoren zufällig gedreht werden, bis ein Schwellwert (0,3 V) unterschritten wurde. Danach wurde die Maximierung neu gestartet. Mit dieser Konfiguration lässt sich schnell maximieren, da der Faktor der Beeinflussung der Achsen durch nur wenige Drehungen der Motoren nicht stark verschlechtert wird. Startet man im Labor die Optimierung, so ist in den meisten Fällen die Konfiguration ähnlich, da das Setup nur leicht aus seinem Maximum gedriftet ist.

In dieser Langzeitmessung traten ein bis sechs Achsenwechsel auf und eine Achsenoptimierung dauerte im Schnitt $275 \text{ s} \pm 25 \text{ s}$. Damit ist mit mindestens 10 Minuten bis maximal ca. 30 Minuten zu rechnen.

Der Wert nach jeder Maximierung ist hier aufgetragen:

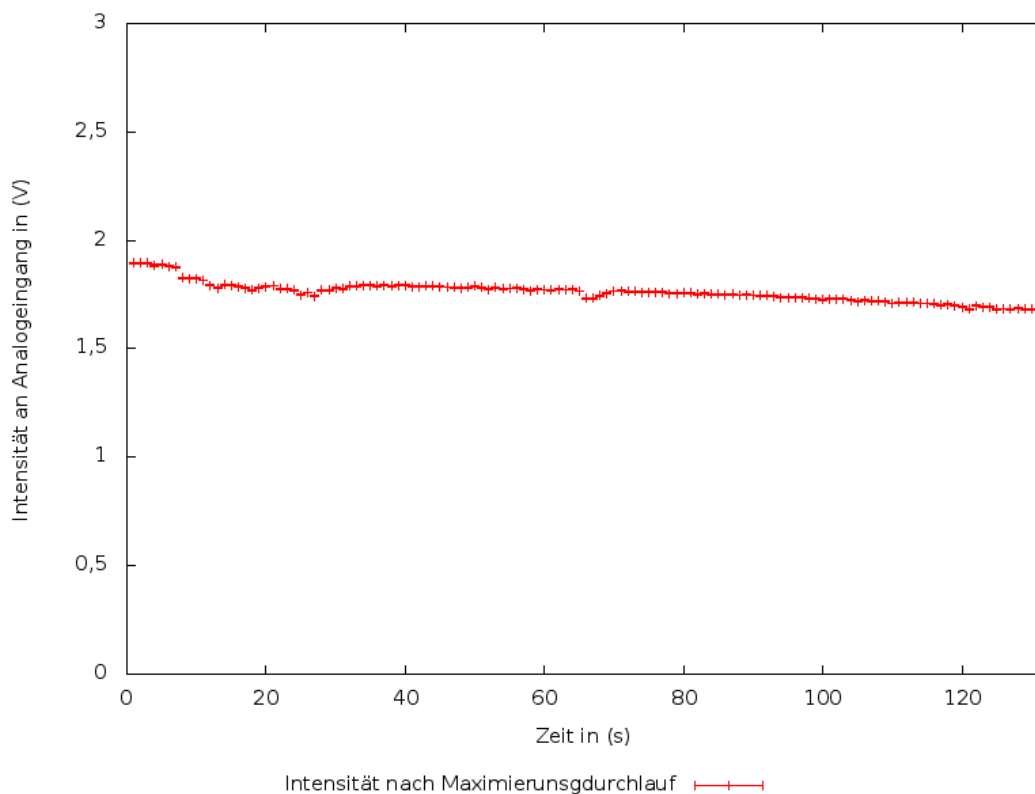


Abbildung 3.15: Wert nach Maximierung in 130 Messungen

Zu sehen ist, dass immer eine Optimierung erfolgt. Am Anfang schwankt dieser Wert noch leicht, da hier die Deckenbeleuchtung noch angeschaltet war und so Einflüsse wie sich vor der Diode bewegend Laboranten den absoluten Wert nach der Maximierung verändern.

Der absolute Wert sank kontinuierlich über alle Messungen leicht. Dies lag vermutlich an einem Intensitätsdrift des Lasers selbst. Auch mit der Hand ließen sich vor und nach den 130

¹⁶Aufgenommen am mbed nach Aufbau 3.1

Messungen keine weitere Verbesserung herbeiführen.

Sollte die Geschwindigkeit nach der Feineinstellung des Aufbaus immer noch nicht genügen, gilt es einen anderen Aufbau, z.B. mit Piezos zu wählen.

Kapitel 4

Ausblick und Fazit

4.1 Verbesserungsmöglichkeiten

4.1.1 Verbesserung der Laufzeit

Einige Feineinstellungen können jeden Schritt beschleunigen. Unter anderem kann die Ausgabe der Daten erst später (oder auch gar nicht) erfolgen. Auch die Mittelung der Werte kann auf einen kürzeren Zeitraum reduziert werden, wenn die Deckenbeleuchtung genügend abgeschirmt ist. An Tabelle 3.2.2 lässt sich die unterste Grenze der Laufzeit klar ablesen: Selbst mit geschicktesten Vorhersagen lässt sich die Wartezeit nach einen Schritt nicht unter die Größenordnung von 0,1 s reduzieren. Damit kann in dem Setup mit mindestens 0,1 s pro Schritt gerechnet werden.

Durch andere Algorithmik ließen sich auch noch einige Schritte sparen, dies ist unten (4.1.3) diskutiert.

4.1.2 Verbesserungen im Aufbau

In einem weiteren Automatisierungsschritt könnte der mbed die Abschwächung über einen motorisierten Abschwächerarm steuern. Auch die Kollimation der Einkopplung (vergleiche Aufbau 3.1 Kollimator 2) könnte prinzipiell vom mbed gesteuert werden. Aufgrund der zusätzlichen Komplexität wurde darauf im Rahmen der Arbeit verzichtet.

Andere Motoren könnten die Nicht-Kalibriertheit der X-Achse beheben, wie z.B. Stepper- oder Piezomotoren. Damit wären auch andere Algorithmen und das Implementieren eines PID-Regler zum Halten des Maximum möglich.

Sieht man die langen Wartezeiten als ausreichend an, kann man es bei diesem Aufbau belassen. Es empfiehlt sich allerdings ein Programm zur automatischen Feineinstellung der Motoren bzw. eine Schrittweitensteuerung einzubauen. Mit Schrittweitensteuerungen im Aufbau wurden gute Erfahrungen während der Bachelorarbeit gesammelt. Sie ist eigentlich nötig, da die Kopplungen Motor ↔ Schraube sich oft für bestimmte Winkel verklemmen.

Im eigentlichen Experiment wurden die Motoren so gut eingestellt, dass eine Schrittweitensteuerung nicht unbedingt nötig war. Um die Auswertung zu vereinfachen wurde sie deswegen nicht verwendet.

4.1.3 Verbesserung der Algorithmik

Einige Werte, wie z.B. für einen Richtungswechsel fünf Schritte ins Negative zu gehen, sind frei gewählt. Statt nur andere Werte zu testen, könnte man hier einen Algorithmus des Typ „Simuliertes Auskühlen“ verwenden: Die Wahrscheinlichkeit für einen Richtungswechsel hängt vom Maß der Verschlechterung ab.

Erhält man eine bessere Auflösung bzw. Kalibrierung der X-Achse, könnte man für die Optimierung noch einen Effekt benutzen: Anhand der Verzerrung der Gaußkurve (vergleiche Kapitel 3.9) ließe sich die Korrelation bestimmen und damit die Schrittweite des Rastermotors einstellen.

4.2 Fazit

Die Automatisierung der Einkopplung eines Laserstrahl in ein Glasfaserkabel konnte verwirklicht werden. Dadurch, dass die Änderungswinkel an der unteren Grenze der Auflösung der Apparatur sind, ist die Laufzeit sehr hoch. Anhand der Arbeit und der mit ihr gemachten Erfahrung können die Quantenoptiklabore entscheiden, wie sie eine Automatisierung implementieren wollen.

Anhang A

Fotos

A.1 Fotos vom Aufbau

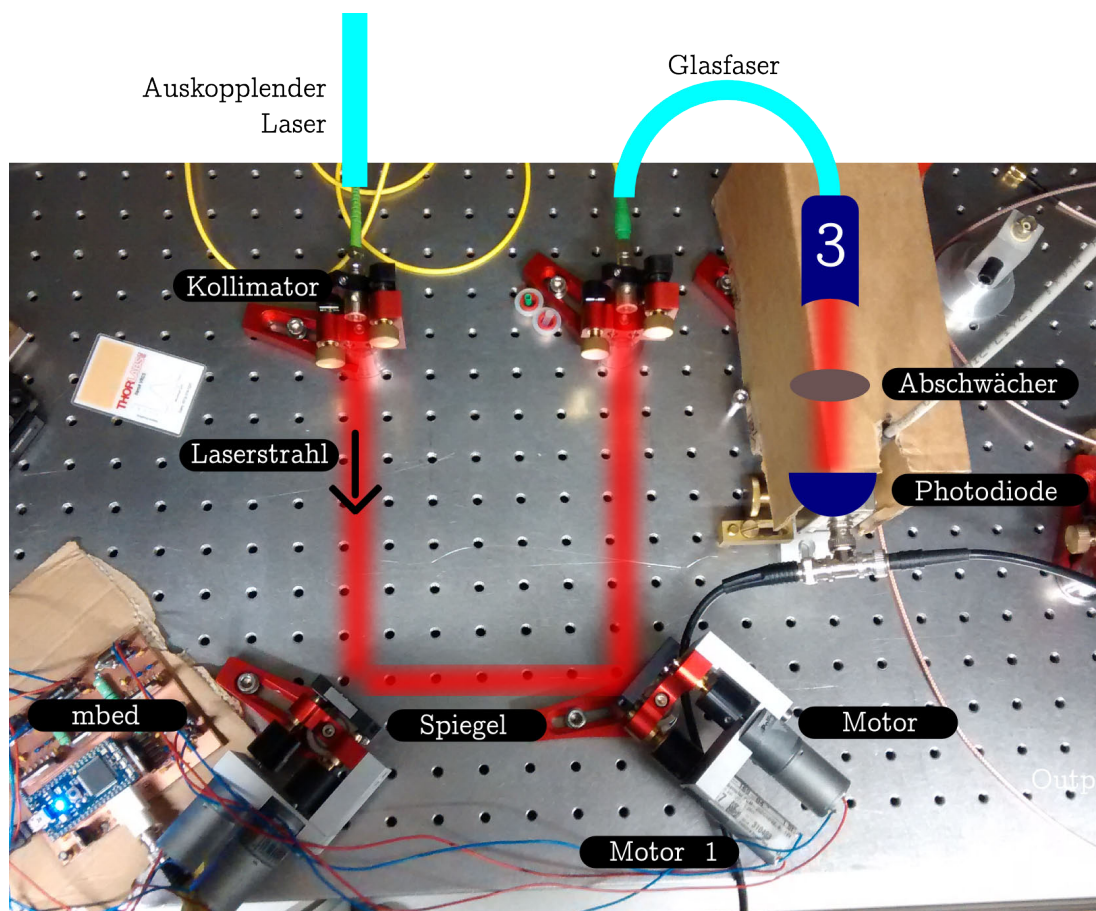


Abbildung A.1: Bearbeitet, um die Funktion der Bauteile im Aufbau zu beschreiben¹

¹Selbst aufgenommen am 04.06.2014, bearbeitet mit Inkscape

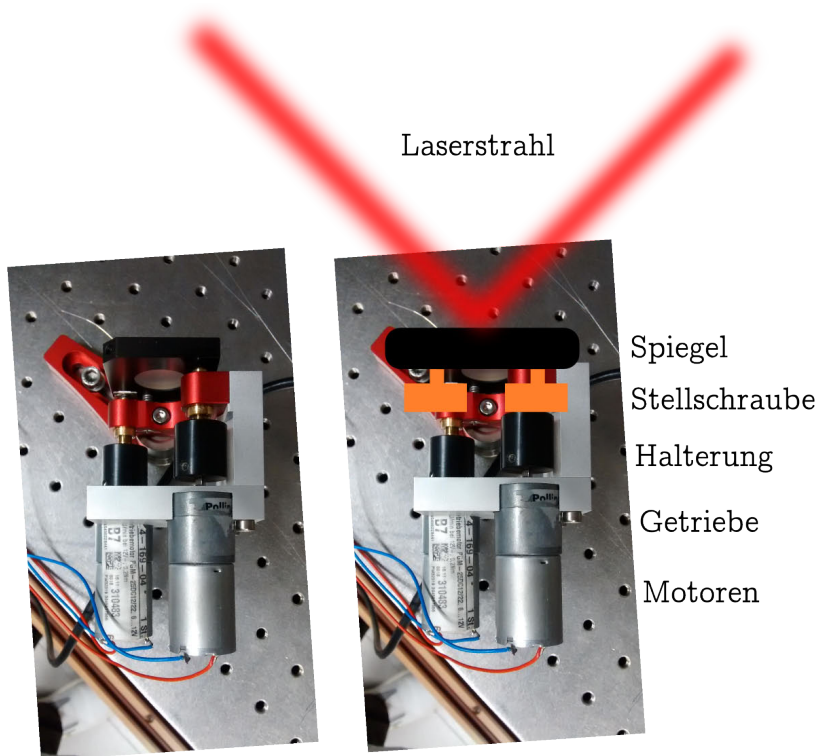


Abbildung A.2: Bearbeitetes Foto der Spiegelhalterung²

²Selbst aufgenommen am 04.06.2014, bearbeitet mit Inkscape

A.2 Einzelne Bauteile

A.2.1 Spiegelhalterung

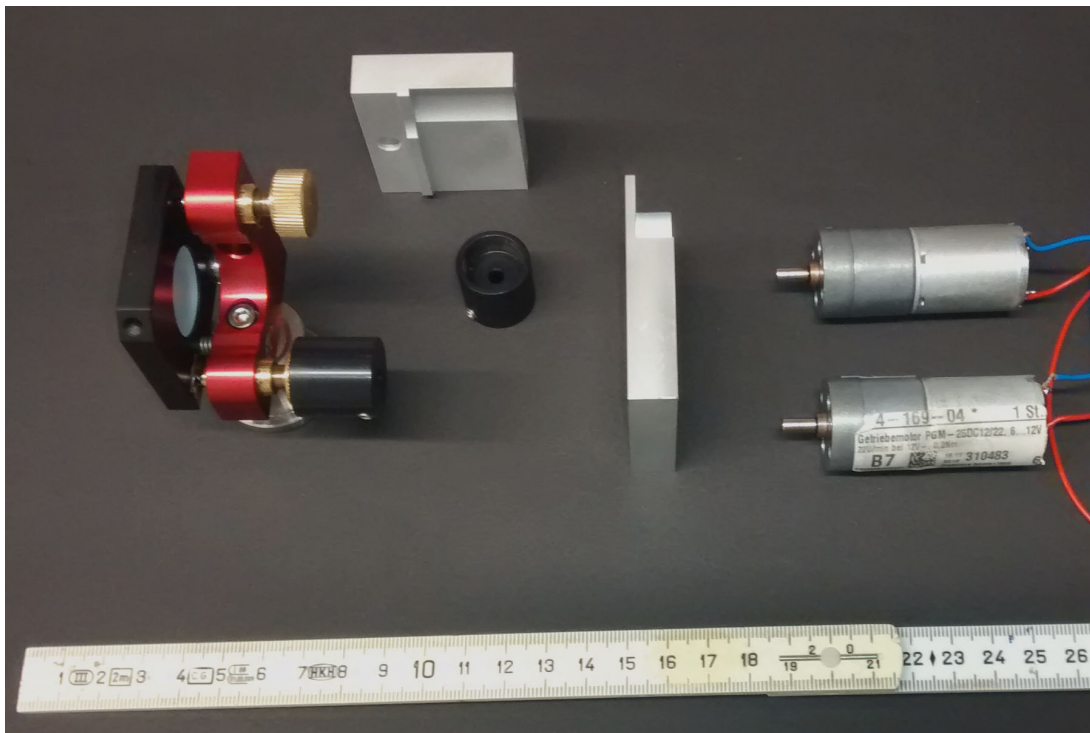


Abbildung A.3: Explosionsansicht der Spiegelhalterung³

³Selbst aufgenommen am 15.07.2014

A.2.2 Halterung Motoren

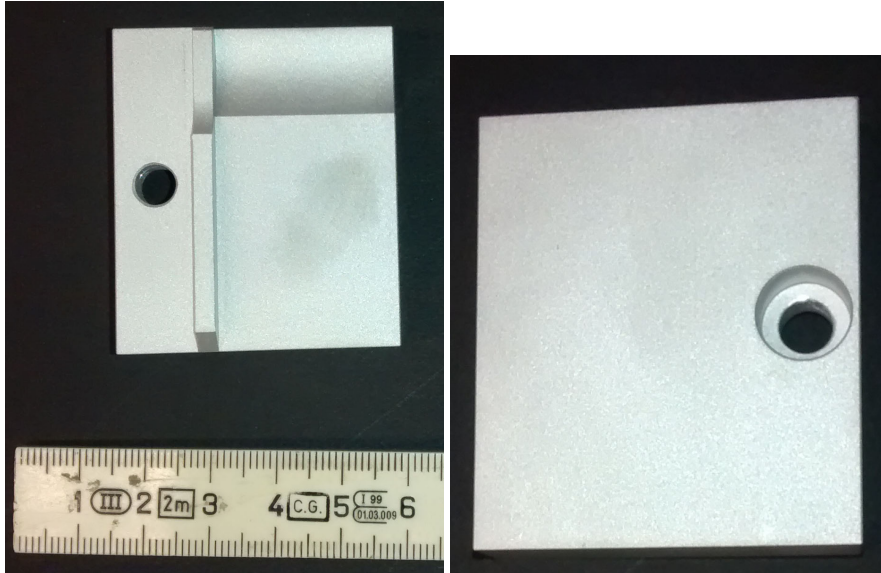


Abbildung A.4: Erster Teil der Halterung der Motoren, hergestellt in der Werkstatt, Entwurf von Stefan Brakhane

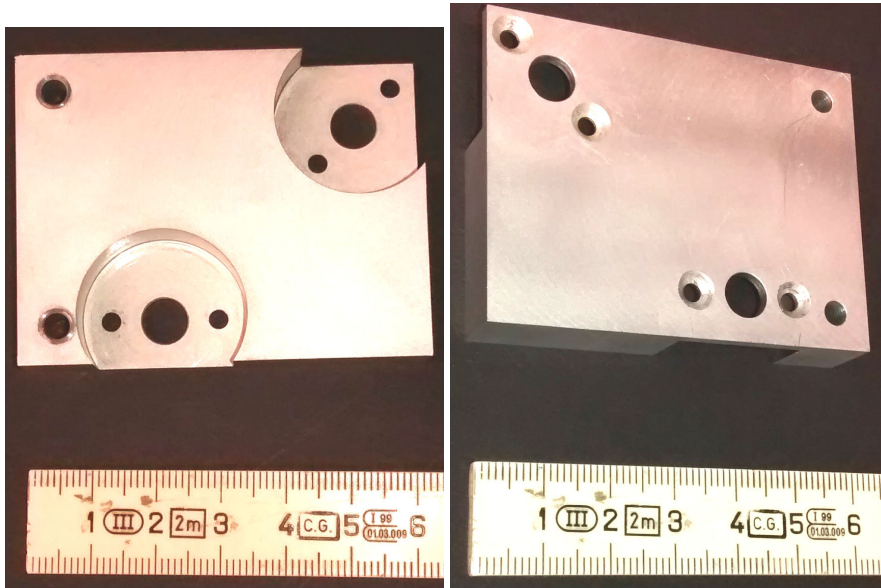


Abbildung A.5: Zweiter Teil der Halterung der Motoren, hergestellt in der Werkstatt, Entwurf von Stefan Brakhane ⁴

⁴Fotos selbst aufgenommen am 15.07.2014

A.2.3 Photodiode

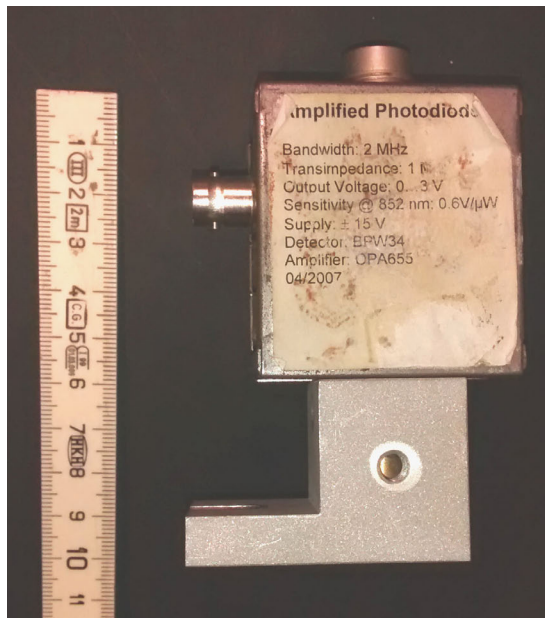


Abbildung A.6: Foto von der Photodiode⁵

Relevante Daten:

Ausgangsspannung	0 bis 3 V
Sensitivität	0,6 V / μ W bei 852 nm
Bandweite	2 Mhz

⁵Selbst aufgenommen am 30.07.2014

A.2.4 mbed

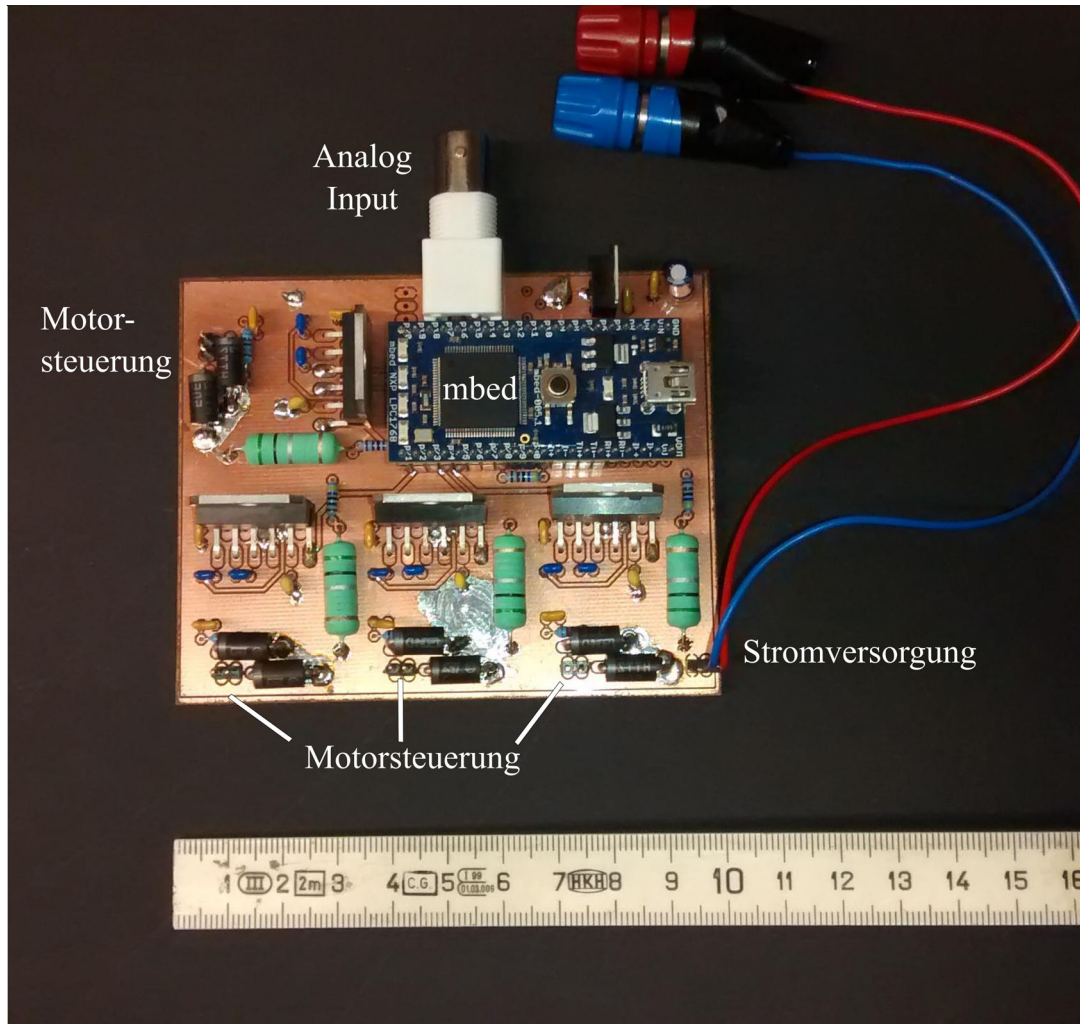


Abbildung A.7: Foto vom mbed-Controller und der verwendeten Platine⁶

Der Schaltplan des Board, entworfen von Stefan Brakhane vom Institut für angewandte Physik der Uni Bonn:

⁶Selbst aufgenommen am 15.07.2014

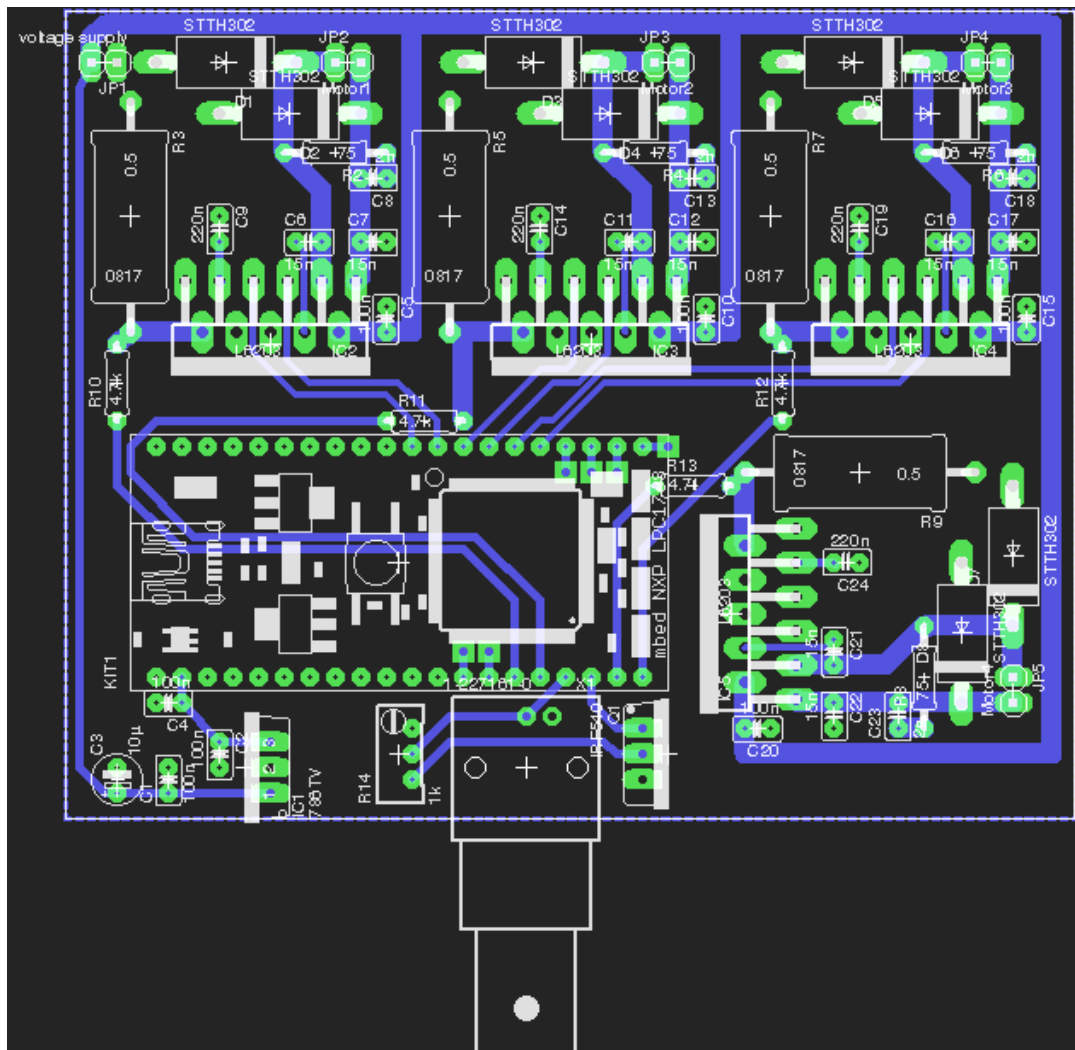


Abbildung A.8: Schaltplan aus der Eagle Datei der verwendeten Platine

Anhang A. Fotos

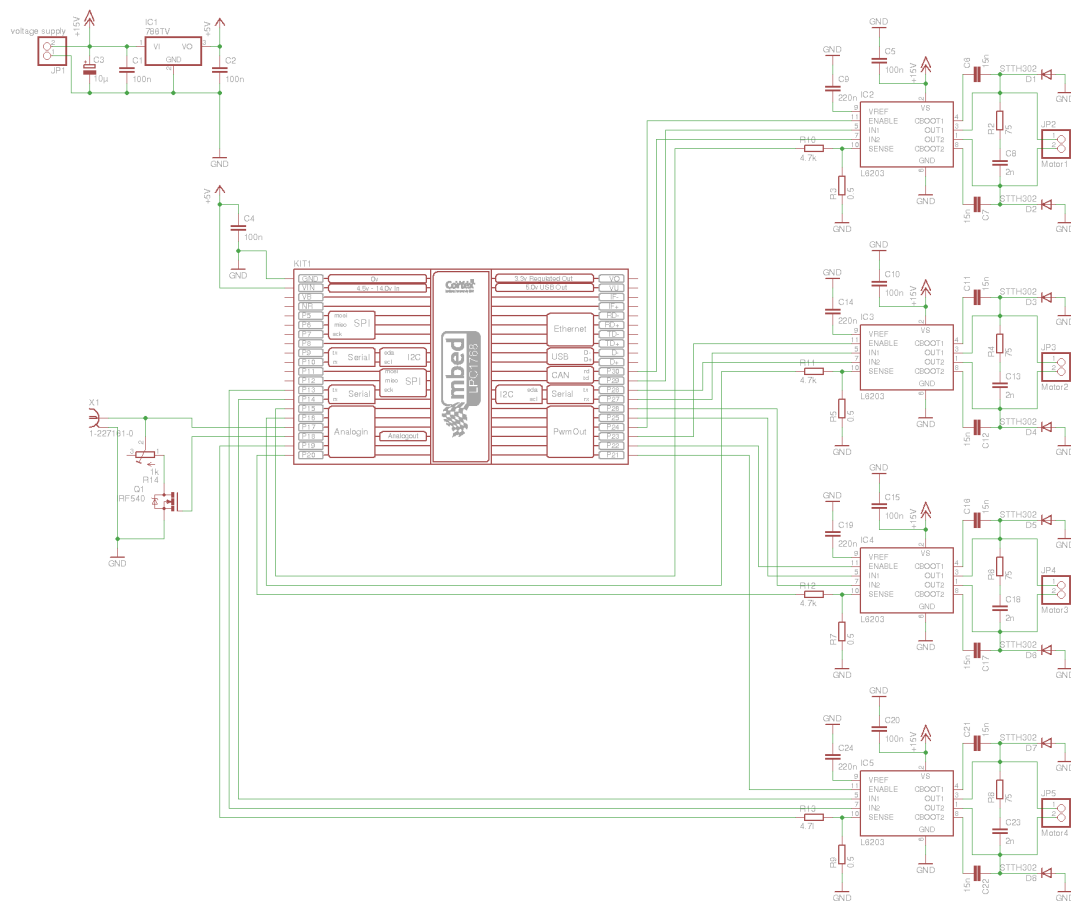


Abbildung A.9: Zweiter Teil des Schaltplan aus der Eagle Datei der verwendeten Platine

Anhang B

Verwendete Software

B.1 Latex

Als Entwicklungsumgebung für Latex diente Kile, als Vorlage für diese Bachelorarbeit diente <http://www.e1.physik.tu-dortmund.de/cms/Medienpool/Bachelor/BachelorArbeit.zip> (Heruntergeladen am 11.06.2014).

B.2 Compiler

Die Programme wurden in der Programmiersprache C++ geschrieben und über den Webcompiler der mbed Platform compiliert. <https://mbed.org/compiler/#nav:/>;

B.3 Graphiken

Zum Erstellen der Aufbauskizzen diente Inkscape. Zum grafischen Veranschaulichung der Daten diente gnuplot.

B.4 Schaltplan

Es existieren Schaltpläne im Format der Software Eagle für das Board, über das der mbed die Motoren steuert. Die Platine wurde für diese Bachelorarbeit verwendet und nicht weiter bearbeitet, so dass ihr Design nicht Teil dieser Arbeit ist.

Quellenverzeichnis

- [1] John A. Buck. *Fundamentals of Optical Fibers*. John Wiley and Sons, New York, 2004. 2. Auflage.
- [2] Simone Agha and Daniel Minkin. Understanding Walking the Beam. <http://laser.physics.sunysb.edu/~simone/mini-project/>, Juli 2007. Stony Brook Laser Teaching Center.
- [3] Fiberlock. http://www.tem-messtechnik.de/index_htm_files/FiberLock_Digital_Web.pdf. Aufgerufen am 25.06.2014.
- [4] Fiberlock. <http://www.tem-messtechnik.de/fiberlock.htm>. Aufgerufen am 25.06.2014.
- [5] Bergsteigeralgorithmus. <http://de.wikipedia.org/wiki/Bergsteigeralgorithmus>. Aufgerufen am 29.07.2014.
- [6] Lunze. *Regelungstechnik 1*. Springer, Berlin Heidelberg New York, 2008. 7., neu bearbeitete Ausgabe.
- [7] Mark Hoemmen. Computing the standard deviation efficiently. <http://www.cs.berkeley.edu/~mhoemmen/cs194/Tutorials/variance.pdf>, August 2007. Aufgerufen am 11.05.2014.

Eidesstattliche Versicherung

Ich versichere, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate kenntlich gemacht habe.

Bonn, den

Unterschrift