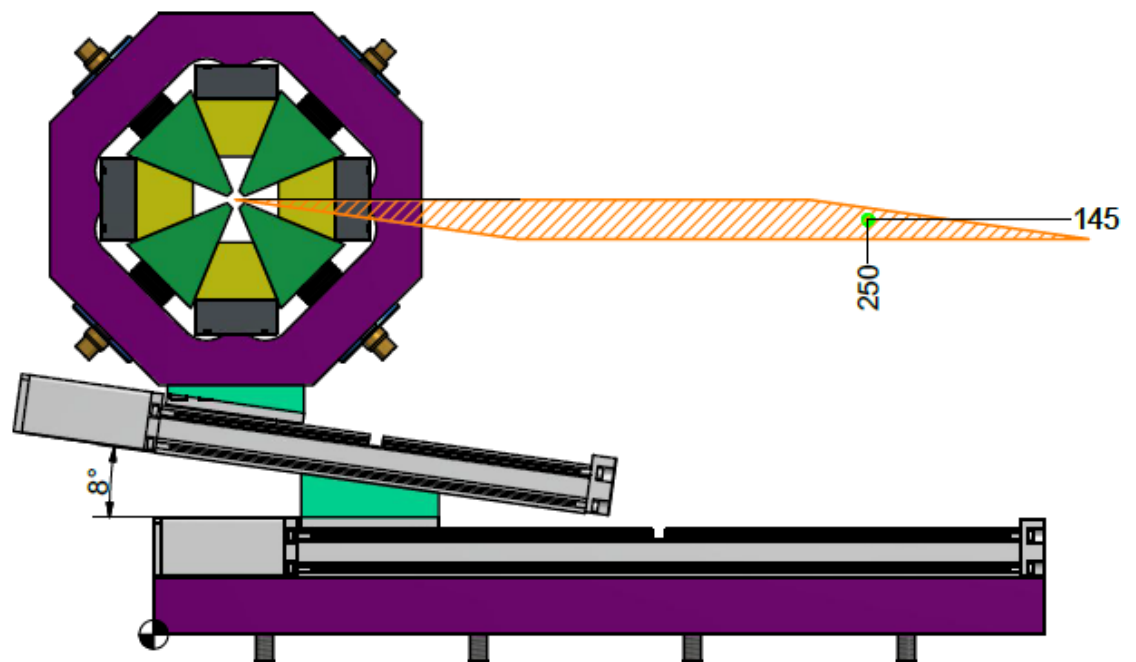


Individuelle Praktische Arbeit 2017

Dominique Zehnder



Quadrupole Positionierungssystem für das Projekt ACHIP

Dominique Zehnder
Paul Scherrer Institut
Advanced Instrumentation Group

Fachvorgesetzter: Rasmus Ischebeck
Betreuer: Eugenio Ferrari
Lehrmeister: Tomislav Rebac
Experte: Martin Klöckner
Co-Experte: Thomas Mettler

13. Februar – 09. März 2017

Inhaltsverzeichnis

1	Einführung.....	3
1.1	Aufgabenstellung.....	3
1.2	Zeitplan.....	4
1.3	Stand der Arbeit.....	5
1.3.1	Konstruktion.....	5
1.3.2	Vakuumkammer.....	7
1.3.3	Testkammer.....	7
2	Linearstages.....	8
2.1	Anforderungen.....	8
2.2	Technische Daten.....	8
2.3	Ansteuerung.....	9
2.3.1	Elektrischer Anschluss.....	9
2.3.2	Software.....	10
3	Optik.....	11
3.1	Kamera.....	11
3.2	Objektiv.....	11
3.3	Software.....	12
4	Messung.....	13
4.1	Prinzip.....	13
4.2	Aufbau.....	15
4.2.1	Kammer & Pumpe.....	15
4.2.2	Optische Aufbau.....	16
4.3	Software.....	18
4.3.1	ZaberContoll.py.....	18
4.3.2	GUI_Main.py.....	18
4.3.3	Imageanalysing.py.....	19
4.3.4	Histogramm.py.....	20
4.3.5	Zaber.py.....	20
4.3.6	XY_POS.py.....	20
4.3.7	Utilities.py.....	21
4.3.8	Anwendung.....	21
4.4	Messungen.....	22

4.4.1	Stabilitätsmessung	22
4.4.2	Home-Messung	22
4.4.3	Target-Messung	22
4.4.4	Target-Target-Messung.....	22
4.4.5	Relativ-Messung.....	22
4.5	Auswertung.....	23
4.5.1	Masssystem.....	23
4.5.2	Home- Messungen.....	24
4.5.3	Stabilitäts-Messung.....	25
4.5.4	Target-Messungen	25
4.5.5	Relativ-Messungen.....	26
4.5.6	Konklusion.....	27
5	Auswertung der Arbeit.....	28
5.1	Software.....	28
5.2	Aufbau & Messungen.....	29
5.3	Probleme& Fehlerquellen.....	30
5.4	Ausblick	30
5.5	Schlussfazit.....	31
5.6	Danksagung.....	31
6	Anhang	32

1 Einführung

1.1 Aufgabenstellung

Die Gruppe Fortgeschrittene Instrumentierung ist Kollaborationspartner des Projekt ACHIP. Das Ziel des Projektes ist es mikroskopische Beschleuniger Strukturen aus Silizium Wafern herzustellen. Diese speziellen Strukturen können durch ein angelegtes Laserlicht Elektronen beschleunigen. Das langfristige Ziel ist es solche Strukturen am SwissFEL–Beschleuniger zu testen. Momentan wird eine Vakuumkammer vorbereitet. Eine Anforderung an den Aufbau ist, dass das Experiment aus dem Strahl entfernt werden kann.

Um die Elektronen auf die kleine Struktur zu fokussieren werden Quadrupol Magnete verwendet. Um eine genau Fokussierung zu erreichen, müssen die Magnete sehr exakt positioniert werden. Für die Positionierung sind zwei Linearstages geplant. Die Stages wurden mir zu Beginn des Praktikums übergeben. Die Aufgabe ist es, die Linearstages auf ihre Genauigkeit zu testen.

Für den Test müssen die Linearstages in Betrieb genommen werden. Um die Motoren zu steuern wird eine Software entwickelt. Die Software soll die Position des Magneten kontrollieren können. Dafür muss eine Formel zur Berechnung der Teilkomponenten erarbeitet werden.

Die Reproduzierbarkeit der Motoren, kann mit einem Mikroskop gemessen werden. Als Referenz wird eine Markierung am Magneten Angebracht. Anhand der Pixelposition kann ein Fehler berechnet werden. Es sollen verschiedene Bewegungsmuster gemessen werden und eine statistische Auswertung erstellt werden.

1.2 Zeitplan

Der Zeitplan wurde bereits weit im Voraus erstellt. Zu diesem Zeitpunkt waren noch nicht alle Einzelheiten definiert. Daher wurde zu Beginn der Arbeit ein neuer Zeitplan erstellt.

IPA Zeitplanung

Dominique
Zehnder
Version 2.1

Mo	Di	Mi	Do	Fr
1. Tag 13.02.2017	2. Tag 14.02.17	3. Tag 15.02.17	4. Tag 16.02.17	5. Tag 17.02.17
Vorbereitung	Vorbereitung	Vorbereitung	Vorbereitung	Vorbereitung
Planung + Dokumentation Motoren testen (Mitgeliefertes Programm)	Software zum Ansteuern der Motoren(Python)	Motor Steuerung Kamera Inbetriebnahme Python	Bildauswertung mit Python	1. Besuch Experte Bildauswertung Adapterkabel
Berufsmaturität 20.02.17	6. Tag 21.02.17	7. Tag 22.02.17	8. Tag 23.02.17	Berufsschule 24.02.17
	Vorbereitung	Umsetzen	Umsetzung	
	Zusammenbau Stages Aufbau Vakuumkammer	Mikroskop Ausrichten	Test Messungen Provisorische Datenauswertung	
Berufsmaturität 27.02.17	9. Tag 28.02.17	10. Tag 01.03.17	11. Tag 02.03.17	Berufsschule 03.03.17
	Umsetzung	Umsetzung	Dokumentation	
	Messungen 2. Besuch Experte	Messungen	Messungen Datenauswertung	
Berufsmaturität 06.03.17	12. Tag 07.03.17	13. Tag 08.03.17	14. Tag 09.03.17	Berufsschule 10.03.17
	Umsetzung	Dokumentation	Präsentation	
	Datenauswertung Dokumentation + Vortrag vorbereiten	Dokumentation Vortrag vorbereiten	Vortrag fertigstellen 3. Besuch Experte + Vortrag	

Dokumentation erfolgt, wenn nicht explizit erwähnt, parallel zur Arbeit

1.3 Stand der Arbeit

Ich habe im Vorfeld der Individuellen Praktischen Arbeit bereits an diesem Projekt gearbeitet. In diesem Kapitel zeige ich den Stand zu Beginn der IPA auf. Die Arbeiten im Vorfeld beschränkten sich auf die Konstruktion und Fertigung der mechanischen Komponenten.

1.3.1 Konstruktion

Das Ziel des Systems ist es einen Quadrupole Magneten zu positionieren. Im Vorfeld der Arbeit habe ich den Quadrupol konstruiert und Gefertigt. Für diesen Prototypen verwendeten ich noch keine magnetischen Materialien. Das Gewicht sowie die Dimensionen sind dem schlussendlichen Quadrupole sehr ähnlich. Dieser Prototyp wird während der IPA zum Messen gebraucht. Das Gewicht des Quadrupols ist 6.25kg.

Alle Komponenten wurden so konstruiert, dass diese im Hochvakuum einsatzfähig sind. Dazu sind alle Löcher entlüftet. Dies geschieht mittels aufgebohrten Schrauben oder Durchgangslöcher. Als Material wurde wo möglich Aluminium verwendet. Dies hat den Vorteil, dass die Konstruktion leicht bleibt und es wenig Ausgast.

Der Magnet muss in zwei Achsen bewegt werden. Besonders in vertikaler Richtung müsste eine „klassische“ XY-Konstruktion das gesamte Gewicht heben. Um dies zu verhindern wurde eine Anordnung der Stages gewählt, bei welcher dies verhindert wird. Als allgemeines Model wurde folgende Skizze gebraucht (Abb. 1). Um die Winkel zu dimensionieren, wurde die X,Y Position in einer Formel Ausgedrückt. Die Abmasse der Vakuumkammer und die Strahl Höhe waren bekannt. Durch eine Parameterstudie konnten die Winkel gewählt werden. Dies ergab für den Prototypen $\alpha=0$ und $\beta=8$. (Abb. 2)

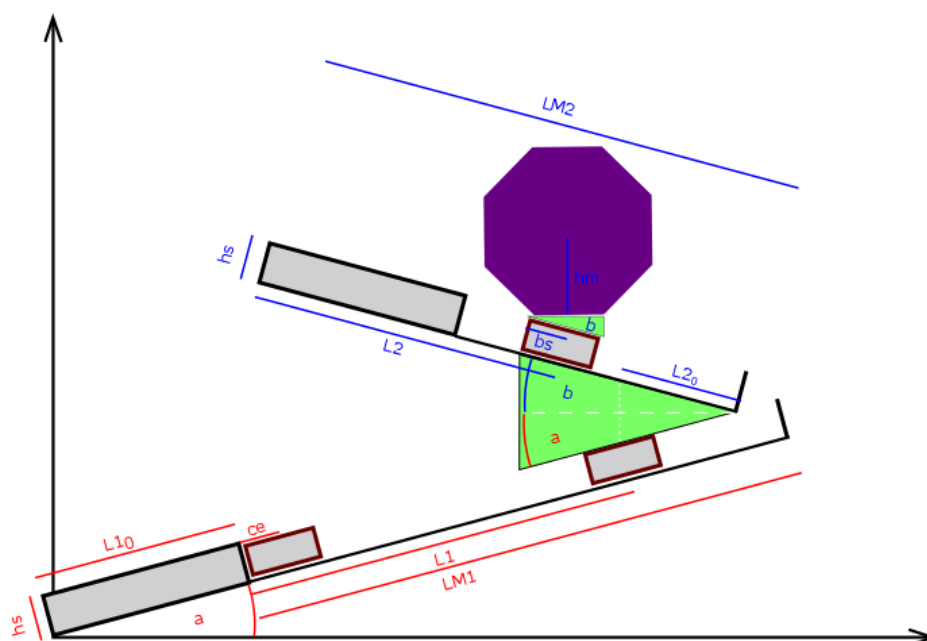


Abb. 1 Aufbau Positionierungssystem

Die X/Y Position in Abhängigkeit der Stage Position ergibt sich durch folgende Formel:

$$x = \cos(\alpha) * (L10 + l1) - \sin(\alpha) * hs + \sin(\beta) * (L20 * \tan\beta) + \cos(\beta) * (L20 - LM2 + l2 + ce) + \sin(\beta) * hs$$

$$y = \sin(\alpha) * (L10 + l1) + \cos(\alpha) * hs + \tan(\alpha) * \cos(\beta) * L20 + \sin(\beta) * (LM2 - l2 - ce) + \cos(\beta) * hs + \sin(\beta) * bs + hc + hm$$

Umgekehrt muss ein beliebiger X/Y Punkt angefahren werden können. Dazu werden die benötigten Positionen durch die Umkehrfunktionen der obigen Funktionen berechnet. (Siehe XY_POS.py)

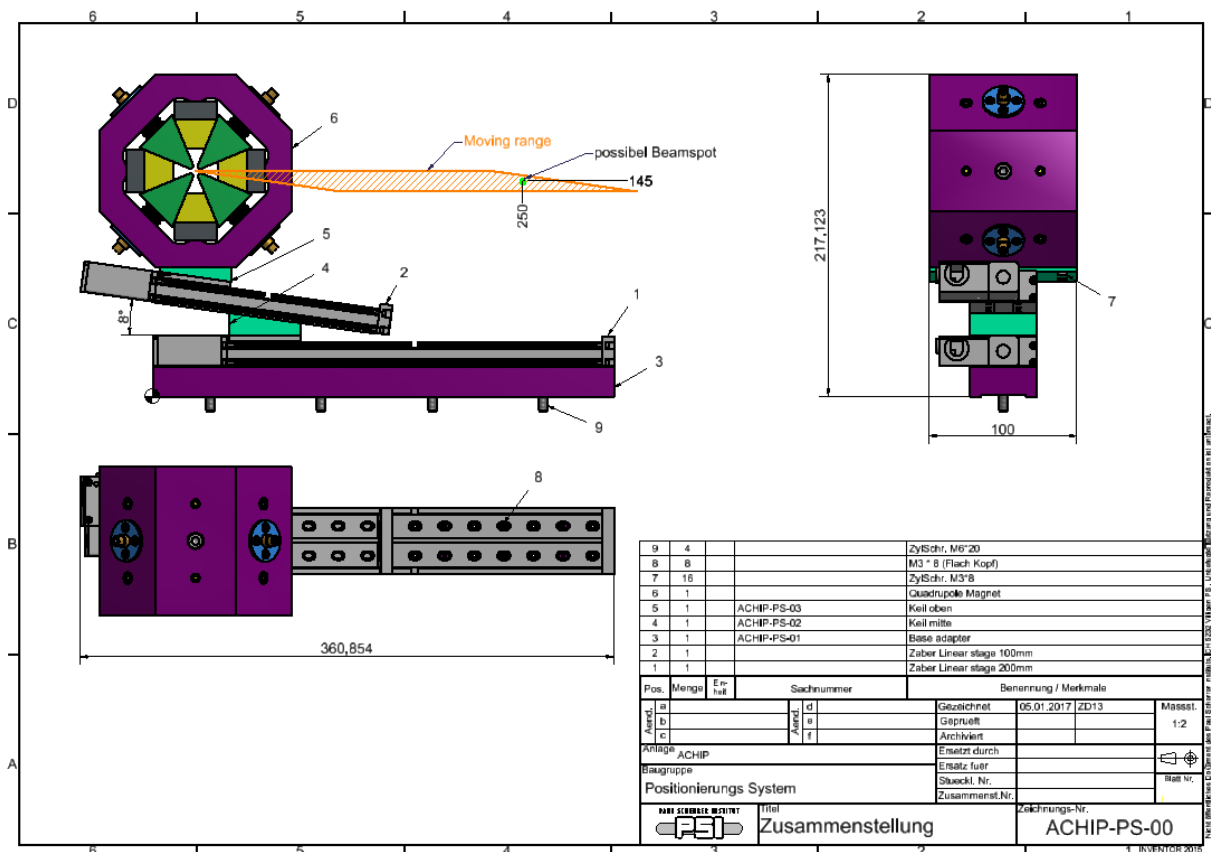


Abb. 2 Übersicht Positionierungssystem

1.3.2 Vakuumkammer

Das ganze System wird in seinem fertigen Zustand in einer Vakuumkammer eingebaut, welche sich im SwissFEL Strahlgang befindet. Total werden sechs Quadrupole verbaut. Je drei vor und nach dem Sample. Durch die Geometrie der Kammer wurde vor allem die Höhe des Systems stark eingeschränkt. Der horizontale Verfahrweg ergab sich dadurch, dass die Magnete zeitweise komplett aus dem Strahl bewegt werden müssen. (Abb. 3)

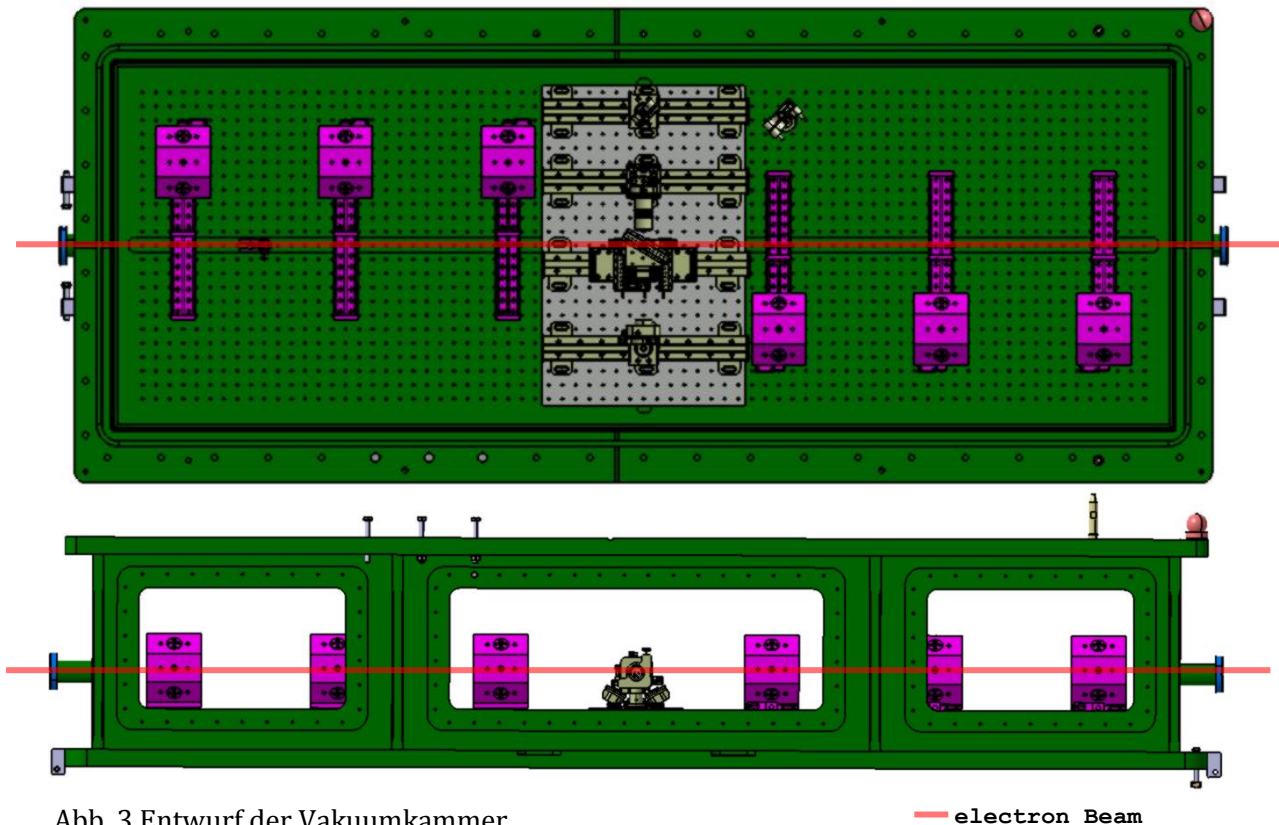


Abb. 3 Entwurf der Vakuumkammer

— electron Beam

1.3.3 Testkammer

Um das System zu testen, möchte ich die Bedingungen möglichst gut nachstellen. Daher sollte der Test in einer Vakuumkammer durchgeführt werden. Die zukünftige Vakuumkammer ist erst in Planung und noch nicht fertiggestellt. Für meinen Aufbau wollte ich vor allem die Kühlung der Motoren testen. Deshalb spielt der Enddruck nicht eine zu grosse Rolle. Die Vakuumkammer musste zusätzlich ein Fenster enthalten, um ein optisches Messverfahren anzuwenden. Das letzte Kriterium war die Grösse. In die Kammer musste der komplette Aufbau mit Quadrupol Magnet passen.

Schlussendlich entschied ich mich für eine Probekammer von Ferrovac. Diese hatte den Vorteil, dass die Wände aus Plexiglas gefertigt sind. Dadurch habe ich eine grosse Flexibilität im Aufbauen des Mikroskops. Mit dieser Kammer wird der Enddruck der späteren Kammer nicht annähernd erreicht. Die Kühlung durch Konduktion und Konvektion ist jedoch schon deutlich geringer als bei einem Test in Atmosphäre.

2 Linearstages

2.1 Anforderungen

Die Anforderungen welche an das Positionierungssystem gestellt sind ergeben sich hauptsächlich durch die Quadrupole Magnete. Diese müssen mit einer bestimmten Genauigkeit positioniert werden. Die Nötigen Werte wurden von der Magnetgruppe berechnet. Im Folgenden ist eine Aufstellung mit den wichtigsten Punkten.

Positionierung:	genauer als 1 μ m
Reproduzierbarkeit:	\sim 1 μ m
Arbeitsgewicht:	\sim 7kg
Horizontale Distanz:	min. 150mm
Vertikale Distanz:	\sim 6mm
Endswitches:	für Position ausserhalb des Beams(0,0)
Encoder:	Nicht verlangt, wäre aber gut
Vakuum:	Hoch Vakuum ($1E^{-6}$ mbar)

2.2 Technische Daten

Als Linearstages werden zwei Stages der Firma Zaber verwendet. Die Stages unterscheiden sich lediglich in ihrer Verfahrdistanz, ansonsten sind sie gleich spezifiziert. Wie aus dem Datenblatt zu entnehmen ist, kann die Stage die Anforderungen grösstenteils erfüllen. Die Restlichen Punkte werden im Rahmen dieser Arbeit überprüft.

T-LSM100A-SV2 / T-LSM200A-SV2	
Specification	Value
Microstep Size (Default Resolution)	0.047625 μ m
Built-in Controller	Yes
Travel Range	101.6 / 203.2 mm
Accuracy (unidirectional)	61 μ m
Repeatability	< 1 μ m
Backlash	< 3 μ m
Maximum Speed	8 mm/s
Encoder Type	None
Communication Interface	RS-232
Maximum Centered Load	100 N
Maximum Cantilever Load	300 N-cm
Vertical Runout	< 8 μ m
Horizontal Runout	< 12 μ m
Linear Motion Per Motor Rev	0.6096 mm
Motor Steps Per Rev	200
Default Resolution	1/64 of a step
Limit or Home Sensing	Magnetic hall sensor
Vacuum Compatible	High vacuum (10^{-6} Torr)



Abb. 5 Linearstage

*Dies ist ein Auszug aus dem Datenblatt, komplette Spezifikation im Anhang

2.3 Ansteuerung

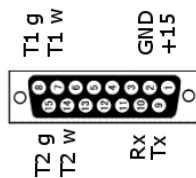
Die Stages haben bereits einen Controller eingebaut. Die Ansteuerung gestaltet sich dadurch sehr einfach. Mit einem RS232 zu USB Adapter kann ich die Stages mit dem Computer verbinden. Durch einen Speziellen Adapter kann ich die Speisespannung von +15V einkoppeln. Dadurch besteht das Kabel aus lediglich 4 Leitern. Ein weiterer Vorteil ist, dass sich mehrere Stages in Serie schalten lassen (Abb. 7). Dadurch kann man beide Stages über das gleiche Kabel betreiben.

2.3.1 Elektrischer Anschluss

Um die Stages in der Vakuumkammer zu betreiben braucht es eine Vakuumdurchführung. Um eine möglichst grosse Flexibilität zu erreichen, wurde die Vakuumkammer mit einem D-Sub 15 Pin Stecker ausgerüstet. Da der an den Motoren Verbaute min Din6 Stecker am PSI kein Standard ist, entschied ich mich den Stecker direkt an den Motor zu Löten. Gleichzeitig befestigte ich zwei Thermoelemente am Stecker um die Temperatur zu überwachen. Auf der Aussenseite verfügt der Stecker über ein RS232 Stecker, zwei Thermoelementstecker sowie den Anschluss für das Powersupply. Die Belegung ist wie Abb. 6 realisiert.

Um die Stage mit dem Computer zu verbinden wird ein USB zu RS232 Converter verwendet. Dieser wird als virtueller COM-Port angezeigt und ermöglicht es die Stages von einer beliebigen Software aus anzusteuern.

Outside



Vacuum

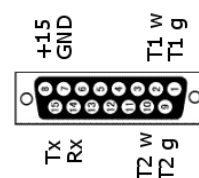


Abb. 6 Pinbelegung

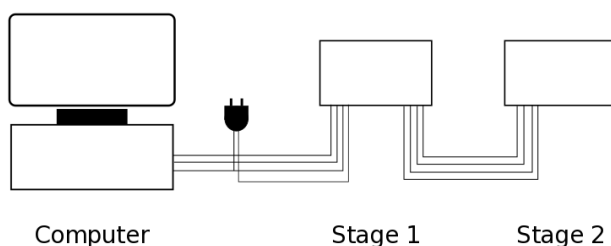


Abb. 7 Anschluss der Stages

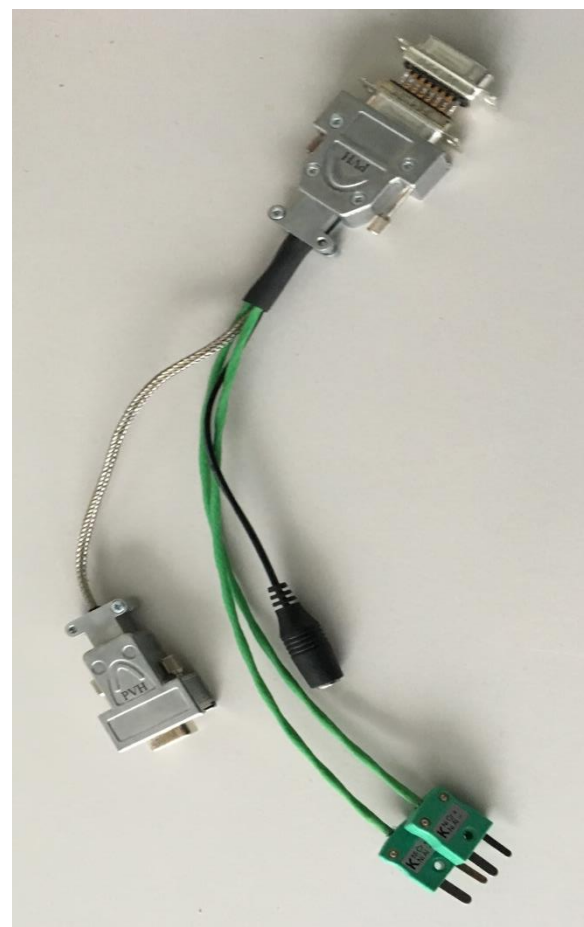


Abb. 8 Adapter für Vakuumdurchführung

2.3.2 Software

Mit den Stages wird ein Tool mitgeliefert um die Motoren zu bewegen. Dieses eignet sich jedoch nicht für meinen Aufbau, da die Bewegung mit der Kamera synchronisiert werden muss. Ich schrieb daher eine Software in Python.

Für die Serielle Kommunikation wird das Python-Package Serial verwendet. Mit diesem wird eine Verbindung zum COM-Port des Controllers aufgebaut. Bevor die Stages verwendet werden, müssen diese nummeriert und gehomet werden. Danach stehen alle Befehle des Zaber Binary Protokolls zur Verfügung. Alle Funktionen zur Kommunikation mit den Zabermotoren habe ich in einem Python Modul zusammengefasst.

Im verwendeten Protokoll besteht jeder Befehl aus 6 Bytes. Die ersten beiden Bytes sind Adresse sowie die Kommandonummer. Die restlichen Bytes sind die Daten bzw. die Parameter. Die Adresse '0' spricht alle Controller gleichzeitig an.

Das Modul enthält die wichtigsten Kommandos als Funktion. Mit der Funktion send hat man die Möglichkeit jeden möglichen Befehl zu schicken. Folgende Funktionen sind implementiert:

```

send( port, device, command, data=0 )           #sendet Kommando an Controller
receive_raw( port )                             #empfängt Daten des Controller und gibt Bytes zurück
receive( port )                                 #empfängt Daten und gibt diese geparkt zurück
wait( port, size=6 )                           #wartet auf (eine) Antwort(en)
success( port )                                #überprüft ob ein Error gesendet wurde
getpos( port, device )                         #gibt aktuelle Position zurück [stepps]
get_xy( port )                                 #gibt aktuelle Position zurück x/y [mm]
move_a( port, device, target )                 #bewegt Stage an abs Position [stepps]
move_r( port, device, distance )               #bewegt Stage an rel Position [stepps]
move_xy( port, x, y )                          #bewegt Stage an abs Position x/y [mm]
to_step( mm )                                  #rechnet mm zu stepps um
to_mm( step )                                  #rechnet stepps zu mm um
    
```

Die genaue Funktion der Befehle kann aus dem Modul entnommen werden(siehe Anhang).

Das Package braucht für eine einwandfreie Funktion folgende Packages:

`Serial, Struct, Colorama`

Zusätzlich wird folgendes Modul gebraucht:

`XY_POS`

3 Optik

Um die Position zu bestimmen wird die Markierung mit einem Mikroskop erfasst. Für diesen Aufbau wird eine Kamera, ein Objektiv sowie eine Beleuchtungsoptik gebraucht. Alle optischen Komponenten waren im Labor vorhanden.

3.1 Kamera

Als Kamera wird eine Basler acA4600-10uc verwendet (Abb. 9). Diese zeichnet sich durch eine hohe Pixeldichte aus und ist daher ideal für meine Anwendung. Die Kamera wird über einen USB 3.0 Stecker gespeist und ausgelesen.

Basler acA4600-10uc	
Specification	Value
Auflösung (H x V Pixel)	4608 px x 3288 px
Pixelgröße horizontal/vertikal	1,4 μm x 1,4 μm
Bildrate	10 fps
Mono/Farbe	Farbe
Schnittstelle	USB 3.0
Video Output Format	Mono 8, YCbCr 422_8, Bayer BG 8, Bayer BG 12, Bayer BG 12 Packed
Pixel Bit Depth	12 bits
Stromversorgung	Via USB 3.0 Interface
Sensor Type	CMOS
Sensorgröße (mm)	6,45 mm x 4,63 mm



Abb. 9 acA4600

*Dies ist ein Auszug aus dem Datenblatt, komplette Spezifikation im Anhang

3.2 Objektiv

Als Objektiv verwendete ich ein Nikon Objektiv (Abb. 10), welches eine zehnfache Vergrößerung hat.

Nikon 10X/0.25 160/- WD5.6	
Specification	Value
Magnification	10x
Working distance	5.6mm
Tube Length	160mm
Numerical Aperture	0.25



Abb. 10 Objektiv

3.3 Software

Die Ansteuerung der Kamera realisiere ich über ein Python Skript. Dieses verwendet für die Kameraverbindung das Package 'pypylon' Dieses ist ein Wrapper um eine C-Bibliothek welche von Basler herausgegeben wird. Durch diesen Zwischenschritt über C wird die Ansteuerung langsam. Bei einem Test mass ich eine Verzögerung von ca. einer Sekunde. Dies ist für meine Anwendung jedoch kein Problem, da ich jeweils nur ein Bild brauche, welches nicht Zeitkritisch ist.

Es können nur Graustufenbilder mit der verwendeten Kamera ausgelesen werden, da das Package die restlichen Formate nicht unterstützt. Dies macht in meinem Fall das Auswerten deutlich leichter, da ich so nur mit einem zweidimensionalen Array arbeiten muss.

Zum Auslesen der Bilder werden folgende Befehle verwendet:

```
pypylon.factory.find_devices()           #zeigt alle verfügbaren Geräte an
pypylon.factory.create_device(available_cameras[-1]) #erzeugt Kamer Objekt
cam.open()                               #Oeffnet verbindung zu Kamera
cam.close()                              #Schlisst Verbindung zu Kamera
cam.properties['ExposureTime'] = 500.0    #Stellt Belichtungszeit ein
cam.properties['PixelFormat'] = 'Mono8'   #Bildformat Grayscale
cam.grab_image()                         #Ein Bild aus der Kamera lesen
cam => Variablen und Objekte
```

Um 'pypylon' zu verwenden muss die PylonSDK von Basler installiert sein. 'Pypylon' ist ein open Source Projekt und kann nur über Git installiert werden. Die Funktionalität ist nicht so ausgeprägt wie dies mit der C Library möglich ist.

4 Messung

Das Ziel ist es die Genauigkeit des Positionierungssystems zu testen. Dabei ist die Absolute Genauigkeit nicht wirklich relevant. Viel wichtiger ist es die Position zuverlässig reproduzieren zu können. Für das Messverfahren bedeutet dies, dass es genügt die Position an der Endposition zu messen. Diese Einschränkung macht den Aufbau etwas einfacher. Dennoch wird eine hohe Auflösung im Sub-Mikrometer Bereich verlangt, um die angestrebten Genauigkeiten zu überprüfen. Die Messung muss in der Lage sein in zwei Dimensionen gleichzeitig zu messen.

Da eine Statistische Auswertung mit vielen Messwerten notwendig ist um eine Aussage zu der Genauigkeit zu treffen, muss das Messverfahren automatisiert werden können. Dabei sollte jedoch eine gewisse Flexibilität vorhanden sein, um verschiedene Bewegungsmuster erfassen zu können.

4.1 Prinzip

Schlussendlich entschied ich mich für ein optisches Messverfahren. Dieses kann alle Anforderungen abdecken und bietet viel Flexibilität. Ein weiterer Vorteil ist, dass die Benötigte Infrastruktur grösstenteils vorhanden ist.

Die Kernidee ist es die Positionsmessung mit einem Mikroskop durchzuführen. Dafür wird ein eine Kamera und ein passendes Objektiv fest installiert. Durch die Vergrösserung des Objektivs wird die höchst mögliche Auflösung bestimmt. Die Stages werden direkt vor dem Mikroskop Aufbau platziert. An dem beweglichen Teil der Stage wird eine Markierung angebracht (Abb. 11). Beim Start der Messung sollte die Markierung im Zentrum des CCD Sensors sein. Mit einem Algorithmus wird die Position der Markierung bestimmt. Nun kann die Stage bewegt werden und die Position des Kreuzes kann verfolgt werden. Über die Anzahl Pixel kann dies in eine absolute Grosse umgerechnet werden.

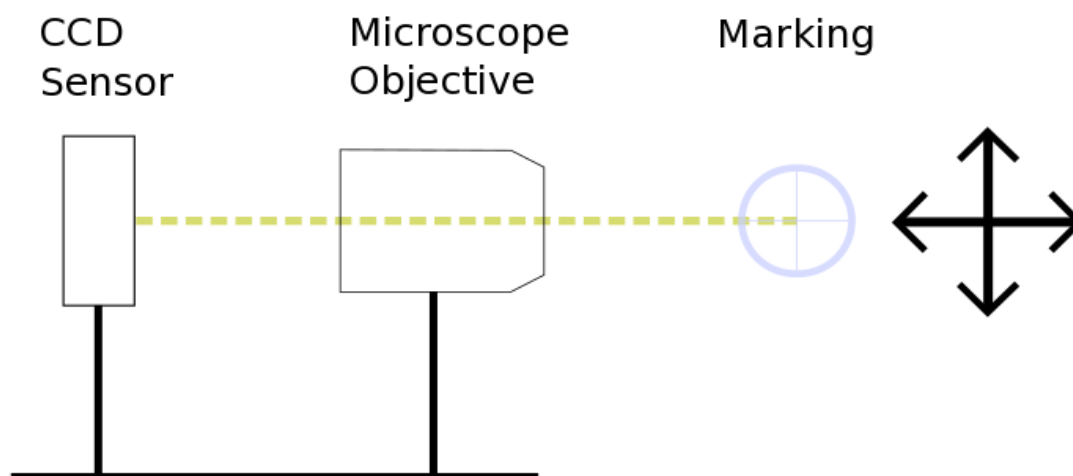


Abb. 11 Messprinzip

Um die Auflösung des Masssystem zu berechnen muss die Pixeldichte der Kamera sowie die Vergrößerung des Objektivs bekannt sein. Dies sind beides Werte, welche im Datenblatt angegeben sind. ($l_p = 1.4 \mu m$ $m = 10$)

$$\Delta l = \frac{l_p}{m} = 0.14 \mu m$$

Damit die Vergrößerung des Objektivs mit dem Angegebenen Wert übereinstimmt, muss das Verhältnis von Objekt zu Objektiv und Kamera den vorgegebenen Werten entsprechen. Diese Distanzen werden durch das Objektiv bestimmt. Beim verwendeten Objektiv ist die Working Distance (Distanz zum Objekt) 5.6 mm. Die Tubelength (Distanz zum Sensor) ist 160mm. Durch diese fixen Längen wird definiert, dass das Objektiv in der Vakuumkammer platziert wird, die Kamera jedoch ausserhalb der Kammer sein muss.

Damit die Tubelength von 160mm stimmt, muss zusätzlich berücksichtigt werden, dass der Lichtstrahl durch die Vakuumkammer geführt wird (Abb. 12). Denn die Ausbreitungsgeschwindigkeit des Lichts ist in Acrylglas langsamer als in Luft ($\eta_p = 1.49$ $d = 35mm$). Die äquivalente Strecke in Luft berechnet sich nach den untenstehenden Formeln.

$$d * \eta = l_c = 52.2$$

$$l_o + l_v + l_c = 160$$

$$l_o + l_v = 107.8$$

Die Unterschiede zwischen Vakuum und Luft sind so klein, dass diese beim Aufbauen vernachlässigt werden können. Die feine Einstellungen der Distanzen erfolgt zuletzt mit Linear Stages, wodurch diese kleinen Fehler korrigiert werden können.

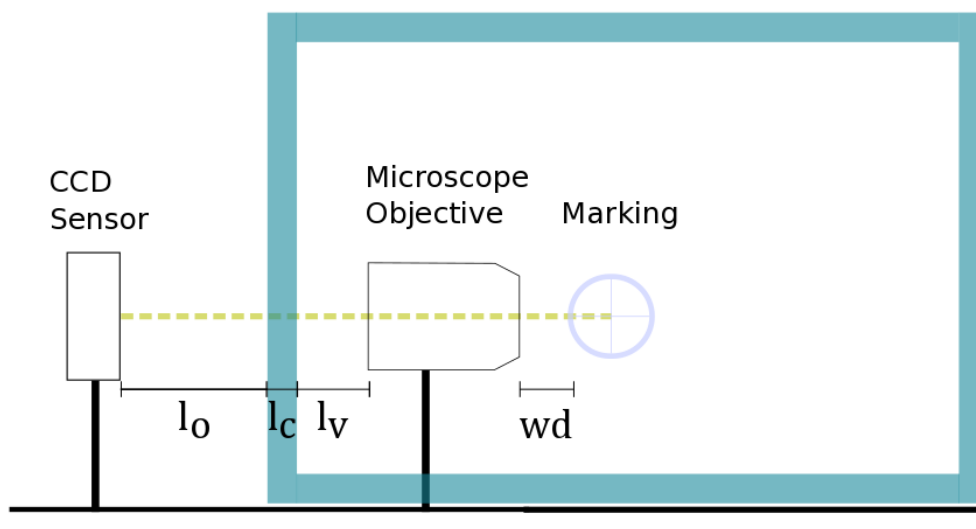


Abb. 12 optischer Aufbau

4.2 Aufbau

Der Aufbau ist anfällig auf Erschütterungen. Deshalb wurde der komplette Aufbau auf einem Optischen Tisch aufgebaut. Da ich die Messungen im Dunkeln durchführen muss um Streulicht zu vermeiden war das Laserlabor noch besser geeignet.

4.2.1 Kammer & Pumpe

Die Vakuumpumpe platzierte ich als erstes auf dem Tisch. Damit die Position der Vakuumpumpe nicht veränderte, schraubte ich Mehrere Posts auf den optischen Tisch. Dies konnte ich als Anschlag brauchen. Von der Seite klemmte ich die Kammer mit einem Winkel. Da die Kammer mit Gummiflüßen geliefert wird, unterlegte ich die Kammer mit Vier Distanzhalter. Um die Unebenheiten auszugleichen unterlegte ich die Kammer mit Unterlagscheiben.

Als Vakuumpumpe verwendete ich eine Membranpumpe, welche eine Saugleistung von $2.1\text{m}^3/\text{h}$ hat. Die Pumpe verfügt über einen KF16 Flansch. Um die Pumpe anzuschliessen verwendete ich einen KF Flex Schlauch. Den Schlauch schloss ich mit einem Rohradapter an das Serto Ventil an. Zum Belüften verwende ich das zweite Ventil, welches an der Kammer verbaut ist. Als Druckanzeige verwendete ich das Verbaute Manometer, eine genauere anzeige wurde für diesen Test nicht gebraucht.

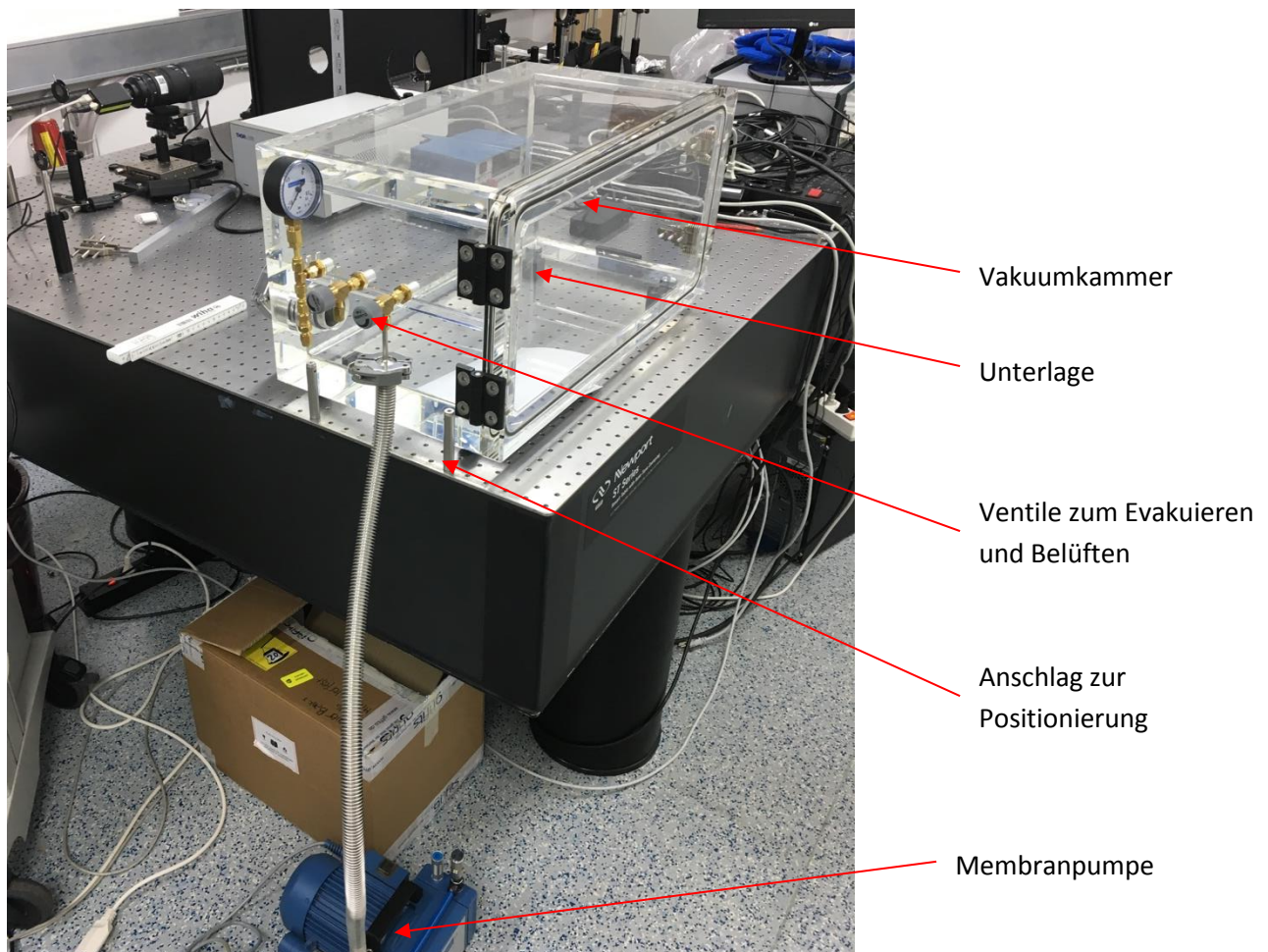


Abb. 13 Vakuumpumpe mit Kammer

4.2.2 Optische Aufbau

Der Aufbau auf der Aussenseite der Kammer besteht aus der Kamera sowie einer Beleuchtungsoptik. Zum Beleuchten wird eine LED verwendet. Durch die eingebaute Optik gibt diese einen gerichteten Lichtstrahl ab. Um den Strahl an die Richtige Stelle zu lenken, wird ein Beamsplitter verwendet. Dieser koppelt das Licht auf die Achse des Objektivs. Die Ausrichtung der LED ist nicht besonders heikel, da diese den gesamten Beamsplitter-Würfel ausleuchtet. Die Kamera welche unter einem Winkel von 90° zur LED steht befestigte ich auf einer Linearstage. Dies ermöglicht es mir den Abstand von Objektiv zum CCD einzustellen.

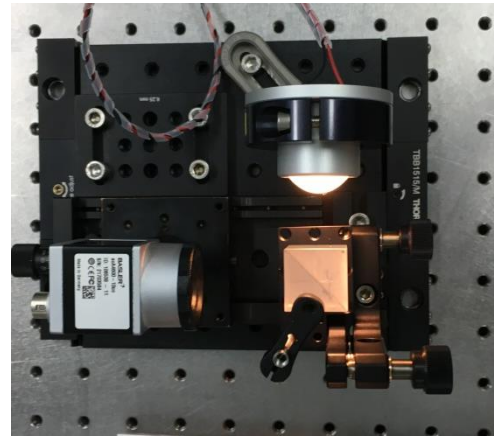


Abb. 14 Aufbau mit Kamera und LED

Es ist geplant die Messung an verschiedenen Positionen durchzuführen. Deshalb baute ich die Kamera und die LED auf einem kleinen Breadboard auf. Dadurch musste ich den Beamsplitter nur zu Beginn ausrichten. Danach kann der Aufbau als Ganzes verschoben werden. Das verwendete Breadboard hatte zudem die Möglichkeit den Aufbau über eine Stellschraube zu verschieben (Abb. 14). Dieser Aufbau ermöglicht es mir die Kamera in zwei Richtungen sehr genau zu positionieren. Die Position an welcher das Licht auf die Achse des Objektivs gekoppelt wird spielt für die Funktion keine Rolle. Um eine möglichst grosse Flexibilität zu erhalten platzierte ich den Beamsplitter jedoch so nah am Rand wie möglich. Mit der Linearstage konnte ich die Kamera dann direkt auf an die benötigte Position einstellen.

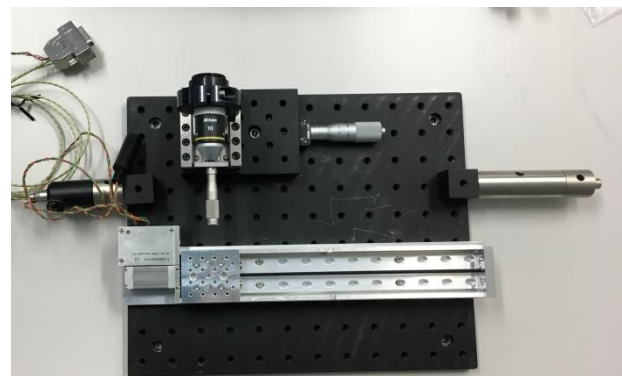


Abb. 15 Aufbau der in der Vakuumkammer

Im inneren der Kammer muss nur das Objektiv platziert werden. Damit eine scharfe Abbildung erzeugt werden kann, muss der Fokus eingestellt werden. Dazu wird eine Linearstage verwendet, welche das Objektiv entlang der optischen Achse bewegt. Zusätzlich wollte ich das Objektiv auch seitlich verschieben. Dies ist wichtig, damit es einfacher wird die Mitte des Kreuzes zu finden. Daher ergab sich einen Aufbau aus zwei orthogonalen Lineartischen.

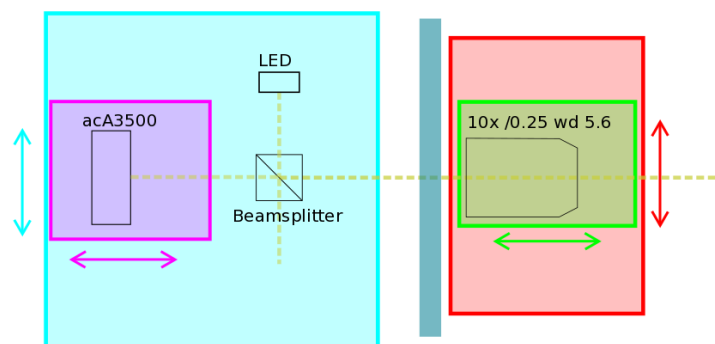


Abb. 16 Verstellmöglichkeiten Optiken

Im Inneren der Kammer befindet sich ein kleines Breadboard. Dieses wird mit zwei Optikhaltern eingeklemmt, damit die Position gleichbleibt. Die restlichen Komponenten werden alle auf dieses Breadboard geschraubt (Abb. 15). Damit die Länge von Objektiv zu Kamera eingehalten werden kann, platzierte ich das Objektiv so nah wie möglich an der Kammerwand. Den Adapter für die Zaber Linearstages platzierte ich gleich davor. Der Abstand wurde hauptsächlich durch die Stellschraube der Linearstages gegeben.

Als Markierung wird ein Fadenkreuz-Plättchen verwendet. Dieses wird eigentlich für Mikroskope verwendet um die Größe von Objekten zu schätzen (Abb. 17). Ich habe zwei Plättchen. Das eine hat nur ein Kreuz aufgezichnet. Dieses eignet sich gut um die Messungen zur Streuung zu machen. Ein zweites Plättchen hat zusätzlich eine Skala mit 100 μm Intervallen aufgedruckt. Dieses kann verwendet werden um die Vergrößerung zu messen. Bei den Plättchen muss beachtet werden, dass es eine reflektierende Seite und eine matte Seite gibt. Um einen genügend grossen Kontraste zu erhalten muss die reflektierende Seite verwendet werden.

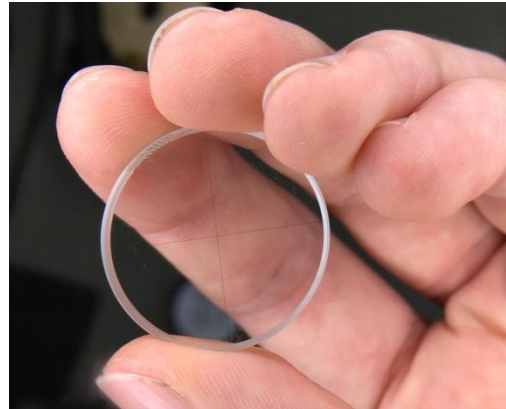


Abb. 17 Plättchen mit Markierung 30 μm

Das Fadenkreuz wird je nach Messung verschieden angebracht. Für die Messung ohne Quadrupole Magnet, wurde ein Rohr verwendet. Dies hat den Vorteil, dass die Länge schon relativ gut eingestellt werden kann. Das Plättchen wird mit Klebeband an das Rohr geklebt.

Um das Plättchen am Quadrupole Magnet zu befestigen, verwendete ich einen Optikhalter, welchen ich an den Magneten klemmte. Durch diesen Aufbau hatte ich ebenfalls die Möglichkeit das Plättchen etwas zu bewegen.

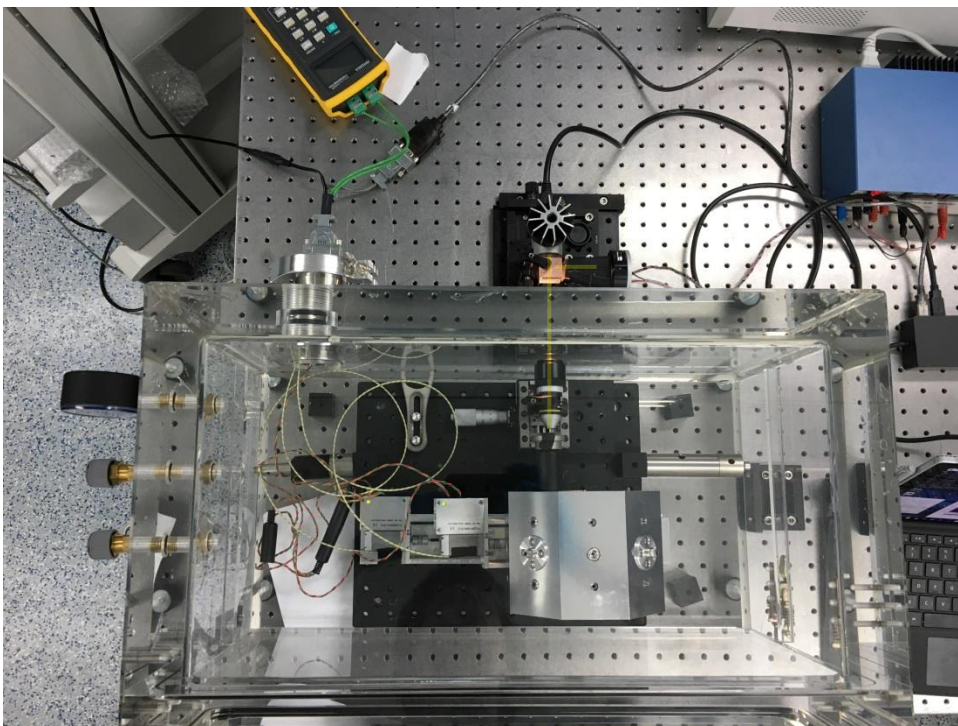


Abb. 18 Versuchsaufbau

4.3 Software

Die Software wurde in Python entwickelt. Diese Programmiersprache eignet sich sehr gut für das Projekt, da sowohl die Linearstages wie auch die Kamera einfach angesteuert werden können. Zusätzlich bietet Python viele Packages für Datenverarbeitung, Bildauswertung und Graph Erzeugung. Ein weiterer Vorteil von Python ist, dass der Code sehr verständlich ist. Dies ermöglicht es schnell Software zu entwickeln.

Die komplette Software ist in sieben Module aufgeteilt. Dabei ist jeweils ein in sich geschlossener Teil der Software zusammengefasst. Der Eintritt in die Software Erfolg über das Skript "ZaberControll.py". Dieses initialisiert das Programm und startet danach die grafische Oberfläche. Das GUI verteilt sich auf die Module: "GUI_main.py", "histogramm.py" und "imageanalysing.py". Die Restlichen Module liefern Bibliotheksfunktionen zur Positionsberechnung und der Motorsteuerung. Nachfolgend wird jedes Modul kurz vorgestellt.

4.3.1 ZaberContoll.py

In diesem Skript ist der einstiegspunkt in das Programm. Als erstes wird mit dem Serial Port des Motor Controllers verbunden. Die Software bietet die Möglichkeit einen Port auszuwählen, oder mit einem vordefinierten Port zu verbinden. Nach dem verbinden des Controllers werden die Stages neu nummeriert. Dadurch wird die Stage welche am nächsten beim Computer ist zur Nummer eins usw. Nach diesem Kommando können die Stags auf die Home-Position gefahren werden.

Wenn die Motoren die Home-Position erreicht haben, kann die Kamera initialisiert werden. Da der Kameraname je nach Port ändern kann, wird nach den verfügbaren Kameras gesucht und die erste in der Liste ausgewählt. Damit die Bilder der Kamera vergleichbar sind, wird das Format sowie die Framerate eingestellt.

Als nächstes wird ein Objekt vom Typ "FineTuning" erzeugt. Dieses erzeugt das Haupt GUI Fenster. Mit welchem die Motoren bewegt werden können. Mit dem nächsten Kommando wird dann in den Eventloop gewechselt. Dieser wird vom verwendeten GUI-Framework tkinter zu Verfügung gestellt.

Zuletzt befindet sich der Code um die Hardware korrekt zu entfernen. Dies ist wichtig, damit danach erneut zu den Geräten verbunden werden kann. Um dies zu garantieren werden mit einer try-exception Struktur alle Fehler abgefangen und das Programm wird richtig geschlossen.

4.3.2 GUI_Main.py

Im Modul sind zwei GUI-Klassen enthalten. Die Klassen wurden von der tkinter-Klasse(GUI-Framework) abgeleitet. Dadurch wird die Klasse direkt zu einem Fenster. In der Init-Methode wird das Fenster mit den Controls und einem Menü gefüllt. Das Fenster bietet die Möglichkeit den Magneten zu bewegen. Dafür ist für jede Richtung ein Button vorhanden(Abb. 19). Zusätzlich gibt es die Möglichkeit an die Home Position zu fahren oder an eine gespeicherte Position zu fahren. Das Menü bietet die Möglichkeit die Optionen zum bewegen des Magneten einzustellen. Über das Menü View können zusätzlich die weitem Fenster Eingblendet werden.

Die Klasse für die Kamera Preview ist ebenfalls im gleichen Modul enthalten. Diese verfügt neben einem Button zum an/abstellen der live Preview lediglich einen Graph. Wenn die Preview angeschaltet ist, wird ca. alle Sekunde ein neues Bild geladen. Diese sehr kleine Rate an Bildern lässt sich leider nicht verkleinern, da das Ansprechen der Kamera fast eine Sekunde braucht. Damit das Programm weiterhin Mousevents zulässt, wird die update-Methode mit einer Verzögerung von einigen ms gestartet. In dieser Zeit können die angefallenen Events abgearbeitet werden.

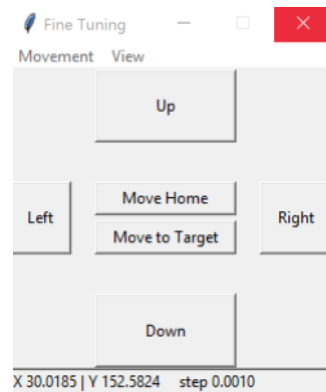


Abb. 19 Main GUI

4.3.3 Imageanalysing.py

In diesem Modul befindet sich die Klasse, welche die Messungen steuert. Diese verfügt über ein Menü über welches alle Messungen gestartet werden können. Im zweiten Menüpunkt können zudem einzelne Bilder aufgenommen und gespeichert werden. Diese Funktionen sind wichtig, um die Beleuchtung und die Ausrichtung vor der eigentlichen Messung zu überprüfen.

Die Messung erfolgt dann durch eine Kombination aus zwei Methoden. Dabei ist der zweite Teil jeweils die `captcher_frame()` Methode. Die erste Methode wird je nach verwendetem Messmodus gestartet. Diese Methode initialisiert beim ersten Durchlauf mehrere Attribute der Klasse, damit alle Informationen zur Länge und Art der Messung vorhanden sind. Danach wird die update Methode erneut aufgerufen. Durch die Attribute kann nach Ablauf der Auswertung wieder auf die richtige Messfunktion zurückgesprungen werden. Auch hier wird zwischen den Aufrufen der Methoden eine kleine Wartezeit eingebaut um andere Userevents abarbeiten zu können. Durch diese Verkettung kann je nach Messung eine andere Motorbewegung durchgeführt werden, der Code zum auswerten des Bildes ist jedoch immer der gleiche.

Der Code zum auswerten des Bildes befindet sich in der Methode `captcher_frame()`. Dies nimmt als erstes ein Bild mit der Kamera auf (Abb. 20). Da die Auswertung einige Zeit in Anspruch nimmt, wird das Bild sofort angezeigt. Im nächsten Schritt werden mehrere Schnitte parallel zu den Seiten gemacht.



Abb. 20 Raw Image

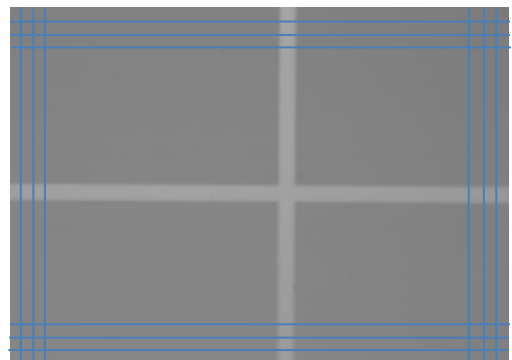


Abb. 21 Schnitte durchs Bild (je 50)

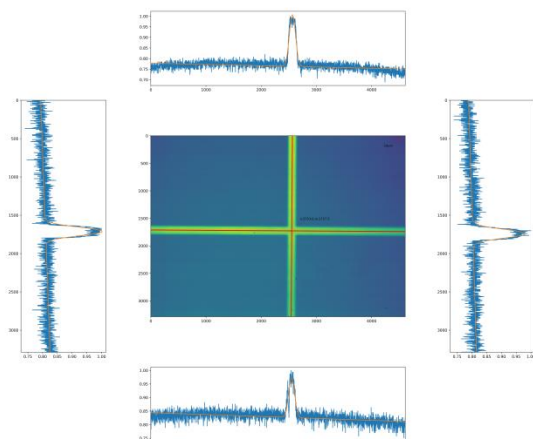


Abb. 22 Erkanntes Kreuz

Dabei werden von allen Seiten ausgehend je 50 Schnitte im Abstand von 5 Pixeln erstellt (Abb. 21). Da das Bild nicht farbig ist, entsprechen die Schnitte gleich der Intensitätsverteilung. Um die Mitte des Peaks zu finden, wird eine Gaußverteilung gefittet. Da das Bild jedoch nicht perfekt ist, wird nicht eine reine Gaußfunktion verwendet. Die Prototypfunktion hat neben der Gaußfunktion einen linearen Anteil addiert. Durch diese einfache Korrektur kann ein ungleichmäßig beleuchtetes Bild ausgeglichen werden. Zum Fitten der Prototypfunktion wird die Funktion `Scipy.optimize.curve_fit` verwendet. Diese hat als Rückgabewert die Parameter des Fitts. Das Zentrum des Gaußanteils entspricht gleich dem Zentrum der Linie. Gesamthaft habe ich so zwei Punktwolke mit je 100 Zentren. Durch diese Punkte kann nun je ein Polynom ersten Grades gelegt werden. Der Schnittpunkt der Funktionen entspricht dem Zentrum des Kreuzes. Das Verfahren stellte sich als sehr robust heraus. (Abb. 22)

Nachdem das Zentrum errechnet wurde, wird dieses als nächstes in eine Textdatei gespeichert. Zusätzlich werden das Originalbild und der Graph mit dem errechneten Zentrum gespeichert. Das Speichern geschieht bei jedem Messpunkt. Dadurch sind die Daten gesichert, wenn das Programm einen Fehler hat. Nach dem Speicher Vorgang wird die statistische Auswertung gestartet. Dazu wird die Funktion `update()` der Histogramm-Klasse aufgerufen. Zuletzt wird überprüft, ob die gewünschte Anzahl an Messungen erreicht wurde. Je nach dem wird dann erneut die Funktion zum Bewegen des Magneten aufgerufen.

4.3.4 Histogramm.py

In diesem Modul ist die Klasse definiert, um die statistische Auswertung einer Messung anzuzeigen. Angezeigt werden ein zweidimensionales Histogramm, sowie die Projektionen in X und Y Richtung. Die Klasse verfügt über lediglich zwei Methoden. Die eine dient zum manuellen Öffnen von Messungen. Die zweite enthält die Funktionalität, um eine Datei anzuzeigen. Dazu werden die Daten aus dem Textfeld in Arrays gelesen. Diese werden normiert und dann als Histogramme dargestellt. Dieser Teil dient nur als Vorschau und ist nicht gedacht, um die Messung direkt auszuwerten.

4.3.5 Zaber.py

In diesem Modul sind alle Funktionen zum Ansteuern der Zaber Linearstages enthalten. Eine Liste mit den implementierten Funktionen kann aus dem Kapitel 2.3.2 entnommen werden. Damit die verschiedenen Kommandos in die entsprechende Dezimalzahl umgewandelt werden können, wird eine Liste mit allen Paaren verwendet. Die restlichen Funktionen basieren alle samt auf den drei primitiven Funktionen zum Senden, Empfangen und Warten. Für die häufigen Kommandos wurden eigene Funktionen erstellt, um die Ansteuerung zu vereinfachen. Die einfachen Funktionen bleiben jedoch weiter ansprechbar. Dadurch wird ein Maximum an Flexibilität ermöglicht. Um eine Möglichkeit zum Notstoppen der Motoren baute ich eine Try-Except Struktur ein, welche den Befehl zum Schließen des Programms zum Stoppen der Bewegung braucht. Wenn jedoch keine Bewegung stattfindet, wird das Programm wie gewohnt beendet.

4.3.6 XY_POS.py

In diesem Modul sind alle Parameter der Stages definiert. Zusätzlich sind die Funktionen zum Umrechnen von Stage Position zu X, Y Koordinaten und umgekehrt definiert. Die Funktionen geben jeweils beide Werte als Tuple (gleichzeitig) zurück. Dies macht den Code kürzer, da beide Werte in einer Zeile berechnet werden können. Diese Funktionen wurden ein separates Modul zugewiesen, da die Funktionen ebenfalls zum Dimensionieren des Systems gebraucht werden.

4.3.7 Utilities.py

Dieses Modul enthält mehrere Funktionen, welche nicht speziell auf dieses Programm ausgelegt sind. Darin sind zwei GUI-Klassen enthalten, welche ein Fenster zur Dateneingabe erzeugen. Diese Möglichkeit zur Benutzereingabe wird an mehreren Stellen im Programm verwendet. Die Klassen sind so aufgebaut, dass Beschriftungen bei der Erzeugung des Objektes definiert werden können.

4.3.8 Anwendung

Hier wird eine kurze Anleitung gegeben, wie eine Messung gestartet werden kann. Nachdem das Skript ZaberControll.py gestartet wurde, öffnet sich ein Konsolenfenster. Dieses bietet dem User keine Eingabemöglichkeiten, jedoch sind in diesem Fenster viele Statusmeldungen und Fehler sichtbar. Nach dem initialisieren, was einige Zeit dauern kann, öffnet sich das GUI automatisch. Aus diesem Fenster kann über das View-Menü die andern Programmteile gestartet werden. Mit den Buttons im Fenster, kann der Magnet an die benötigte Position gefahren werden. Danach wird der Measurement-Tab geöffnet. Mit dem Menü können Testbilder gemacht werden. Diese werden sofort angezeigt. Wenn das Kreuz etwa mittig abgebildet wird, ist man bereit für die Messung. Bevor die Messung gestartet wird, empfiehlt es sich die verschiedenen Graphen so anzuordnen, dass diese alle sichtbar sind (Abb. 23). Über das Menü kann dann der gewünschte Messmodus gestartet werden. Diese werden im nächsten Kapitel genauer erklärt. In der Konsole kann der Fortschritt der Messung verfolgt werden. Nach der Messung kann die Software beendet werden.

Da die Software vorerst nur durch mich eingesetzt wird, wurde die Benutzerfreundlichkeit, sowie das Error Handling von Benutzereingaben auf ein Minimum begrenzt. Der Grund dafür ist, dass solche kleinen Verbesserungen viel Zeit in Anspruch nehmen, die Funktionalität des Programms jedoch nicht steigern.

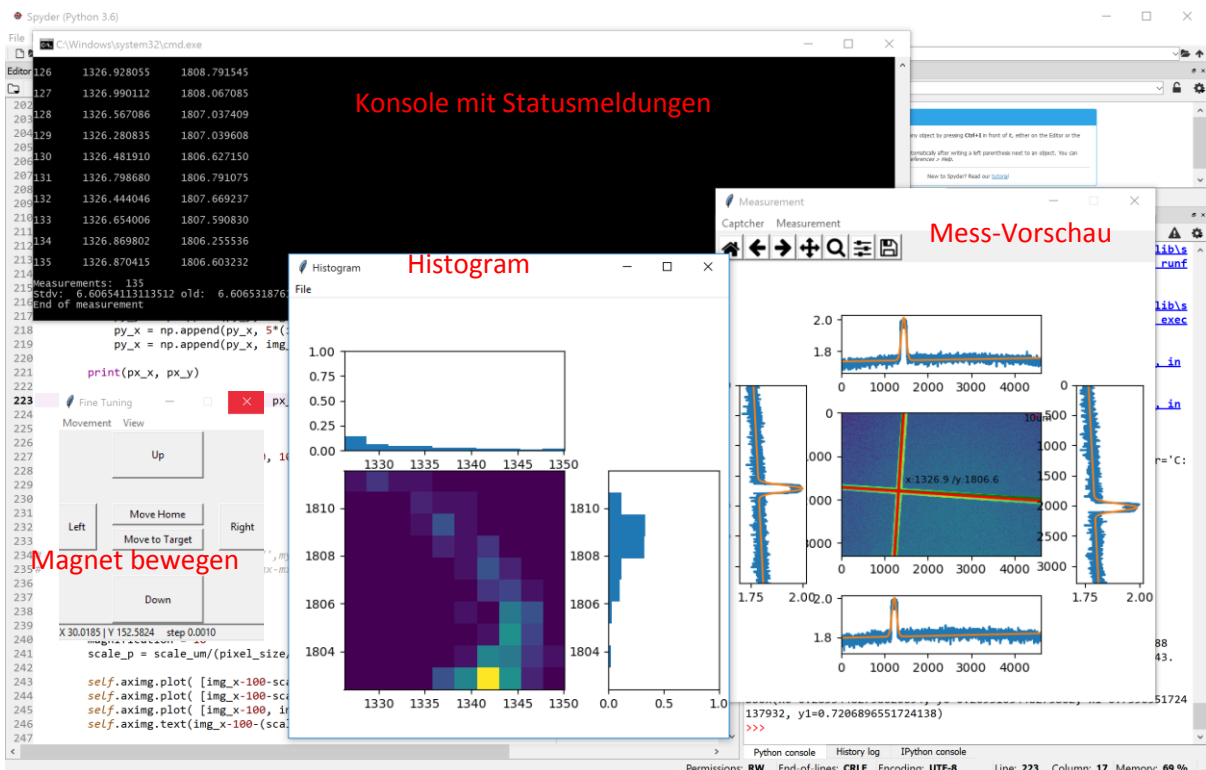


Abb. 23 Programm bei einer Messung

4.4 Messungen

Um die Stages zu testen, wurden verschiedene Messungen durchgeführt. Im Folgenden sind diese aufgelistet und erklärt. Zu jedem Messverfahren gibt es in der Software die entsprechende Funktion.

4.4.1 Stabilitätsmessung

Das Ziel dieses Messmodus ist es die Stabilität des Messsystems oder der Stages zu testen. Dazu wird das Zentrum des Kreuzes vermessen ohne die Stages zu bewegen. Dadurch kann die Streuung in der Messung gemessen werden, wenn das Kreuz fest am Breadboard befestigt ist. Wenn das Kreuz auf den Linearstages ist, kann deren Drift gemessen werden.

4.4.2 Home-Messung

Bei der Home Messung ist das Ziel, die Wiederholgenauigkeit des Endschalters zu testen. Dazu wird die Stage etwas von der Homeposition weggefahren. Danach wird mit dem Homebefehl diese wieder angefahren. Die Kamera ist auf den Nullpunkt ausgerichtet. Durch die Messung ergibt sich die Streuung des Homing Verfahrens. Dies ist die Grundlage für eine zuverlässige Positionierung.

4.4.3 Target-Messung

In diesem Messmodus wird die Stage an die Endposition gefahren und referenziert. Danach wird eine beliebige Position angefahren. An dieser wird mit dem Mikroskop Aufbau die Streuung gemessen. Mit dieser Messung kann ich in etwa die spätere Einsatzbewegung simulieren. Diese Messung braucht jedoch sehr viel Zeit, da der Magnet über grosse Distanzen bewegt wird.

4.4.4 Target-Target-Messung

Bei diesem Messmodus wird die Bewegung des Motors untersucht, wenn dieser zwischen zwei beliebigen Positionen hin und her pendelt. Dazwischen wird der Motor nicht neu referenziert. Mit dieser Messung kann man vor allem untersuchen wie häufig ein Zehlschritt 'verloren' geht.

4.4.5 Relativ-Messung

Mit diesem Modus wird gemessen wie gross ein relativer schritt wirklich ist. Dabei wird der Drift über das gesamte ignoriert und nur die Relative Distanz zwischen den zwei Punkten interessiert. Damit die Messung nachvollziehbar bleibt, werden sowohl Start als auch Endpunkt gespeichert. Die Differenz muss bei der Auswertung gebildet werden.

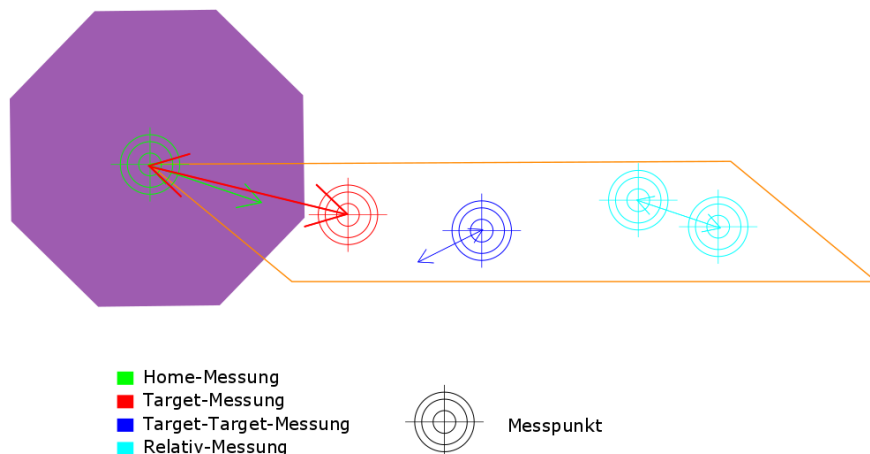


Abb. 24 Messverfahren, Pfeil zeigt Bewegung an

4.5 Auswertung

Die Messungen wurden jeweils mit 200 Punkten durchgeführt. Ein Test hat ergeben, dass die Streuung nach 200 Messungen nicht mehr stark zunimmt. Die Zahl ist so klein gewählt, da die Messungen schon bei 200 Messungen zwischen zwei und neun Stunden brauchten. Die Skala auf den Graphen ist die Distanz in μm zum Nullpunkt des CCD Sensors. Die Auswertung wurde mit dem Statistikprogramm R gemacht. Es wurde jeweils der Mittelwert berechnet sowie die Mittlere Quadratische Abweichung.

4.5.1 Masssystem

Um die gemessenen Daten zu verifizieren habe ich das Masssystem ebenfalls getestet. Wichtig für die Datenauswertung ist, zum einen, wie gross die Vergrößerung ist. Zum andern ist auch die Stabilität des Systems sehr wichtig. Um die Vergrößerung zu testen, verwendete ich ein zweites Mikroskop-Plättchen. Dieses hat eine Skala mit $100 \mu\text{m}$ schritten aufgedruckt. Mit der Kamera machte ich eine Aufnahme, von dem Plättchen. Anhand des Abstandes von zwei Strichen konnte dann die effektive Verstärkung berechnet werden. Dis ergab nicht wie ursprünglich berechnet 10 sondern 8.5. Diese Diskrepanz kommt zustande, da das objektiv nach dem ersten alignen etwas bewegt wurde. Zusätzlich ist es sehr schwierig die Distanz genau zu messen, da diese nur in mehreren Teilstrecken erfasst werden kann. Zusätzlich ist es schwierig die genaue Position des CCD und der Linsen innerhalb der Gehäuse zu erahnen. Dieser Unterschied in der Verstärkung ist nicht tragisch, da er beim umrechnen einfach berücksichtigt werden kann.

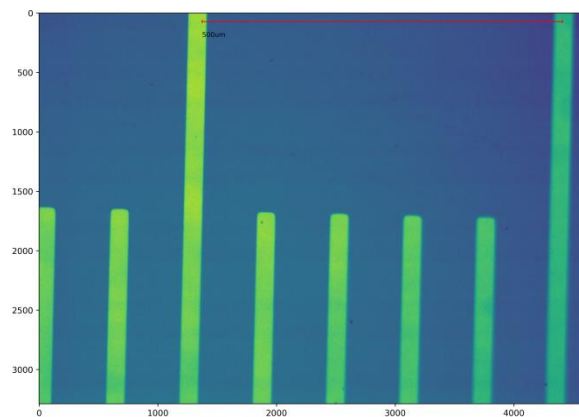


Abb. 25 Testbild mit 8.5x (Linie= $500 \mu\text{m}$)

Die zweite wichtige Grösse ist die Stabilität der Kamera selbst. Diese wurde gemessen, indem eine Stabilitätsmessung durchgeführt wurde. Dabei wurden wie bei den Anderen Messungen 200 Punkte gemessen. Das Ergebnis ist in Abb. 26 zu sehen.

Es ist ersichtlich, dass die Streuung deutlich unter einem Mikrometer bleibt. Der Mittlere Fehler erreicht die theoretische Auflösung. Dies ist möglich, da die Mittlungen an Stabilität gewonnen wird.

Messmodus	Stabilität
Anzahl Messungen	200
RMS X	$0.13 \mu\text{m}$
RMS Y	$0.08 \mu\text{m}$

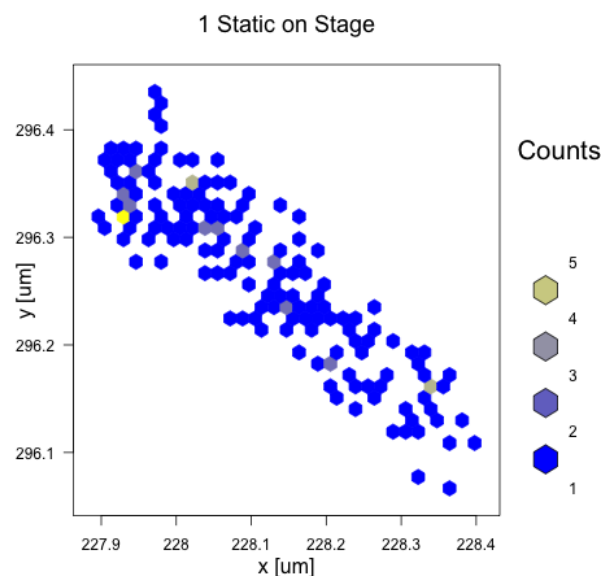


Abb. 26 Stabilitätsmessung

4.5.2 Home-Messungen.

Die Messung der Homegenauigkeit habe ich dreimal durchgeführt. Je einmal bei den Stages für sich alleine, und einmal zusammengebaut mit dem Magneten. Die Ergebnisse sind in den Graphen 27-29 dargestellt.

Verwendeter Aufbau	T-LSM200
Position	Home
Anzahl Messungen	200
RMS X	2.11 μm
RMS Y	0.46 μm

Die Verteilung in X-Richtung ist grösser, da die Stage sich in diese Richtung bewegt. Die Streuung in Y-Richtung ist das Spiel der Führung.

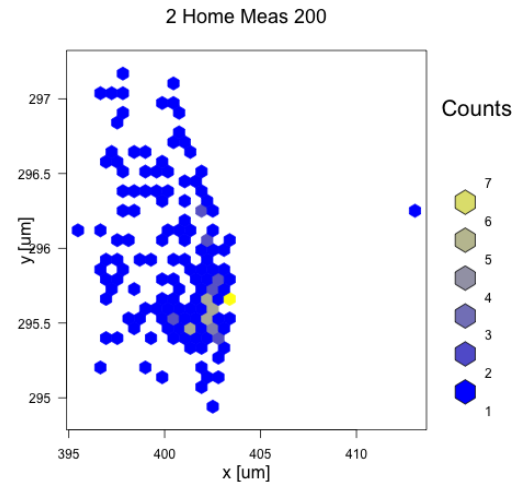


Abb. 27 Home-Messung T-LSM200

Verwendeter Aufbau	T-LSM100
Position	Home
Anzahl Messungen	200
RMS X	3.58 μm
RMS Y	0.59 μm

Die Messung wurde genau gleich durchgeführt wie die Messung mit der langen Stage. Die Streuung ist etwas grösser als bei dem ersten Versuch. Die Unterschiede sind vermutlich produktionsbedingt.

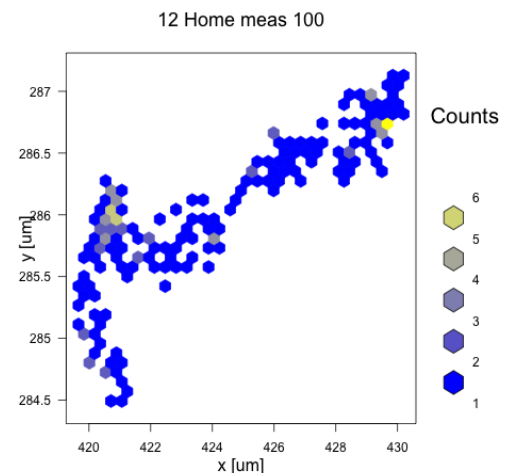


Abb. 28 Home-Messung T-LSM100

Verwendeter Aufbau	System
Position	Home
Anzahl Messungen	200
RMS X	6.72 μm
RMS Y	2.74 μm

Die Streuung mit dem Kompletten System ist deutlich grösser als bei den einzelnen Stages. Dies macht Sinn, da sich die Fehler summieren. Die Y-Streuung bleibt klein, da sich diese nur aus dem Spiel der Führung und dem Sinus des Stage Fehlers ergibt.

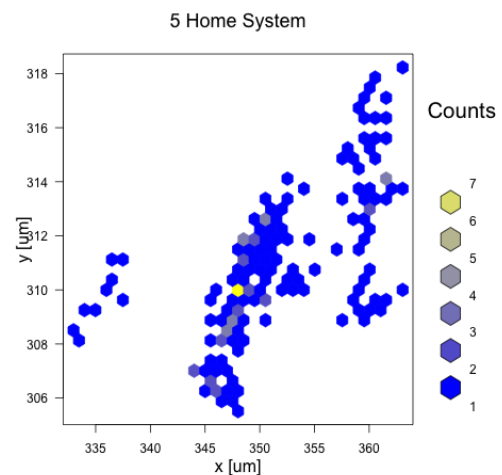


Abb. 29 Home-Messung System

4.5.3 Stabilitäts-Messung

Die Stabilität des kompletten Systems wurde einmal an der Homeposition gemessen. In Abbildung 30 sind die Punkte aufgezeichnet, welche gemessen wurden. Die Zahl neben jedem Punkt ist die Nummer des Punktes. Wie zu sehen ist, ist das System über die Zeit gedriftet. Dies lässt sich damit erklären, dass der Magnet welcher ca. 6 Kg ist das System stark belastet.

Die Nummern neben den Datenbanken sind die Nummer des Messpunktes. Es ist eine deutlich gerichtete Bewegung festzustellen.

Verwendeter Aufbau	System
Position	Home
Anzahl Messungen	200
Zeit	1h 40min
Driftgeschwindigkeit X	≈ 0.7 μm/h
Driftgeschwindigkeit Y	≈ 2.4 μm/h

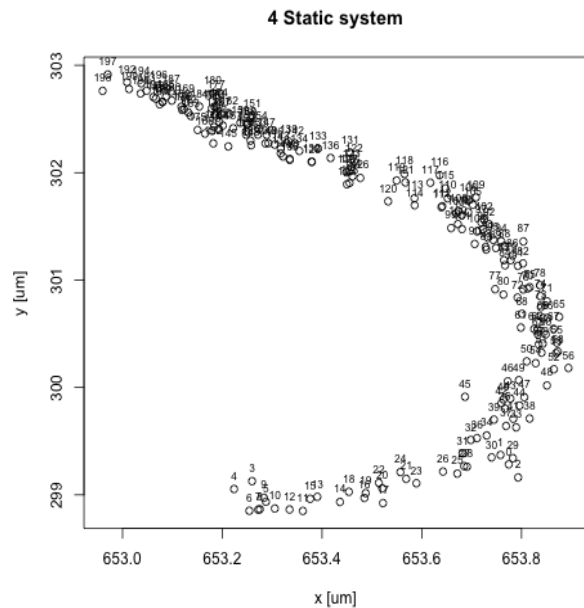


Abb. 30 Stabilität System

4.5.4 Target-Messungen

Bei diesen Messungen wurde jeweils eine Position angefahren, welche etwa der der späteren Einsatzposition entspricht. Im ersten Versuch wurde zwischen jeder Messung erneut am Endschalter referenziert. Dadurch ergab sich die Verteilung wie auf Abbildung 31.

Verwendeter Aufbau	System
Position	X206.2007 Y145.1001
Anzahl Messungen	200
RMS X	9.99 μm
RMS Y	6.25 μm

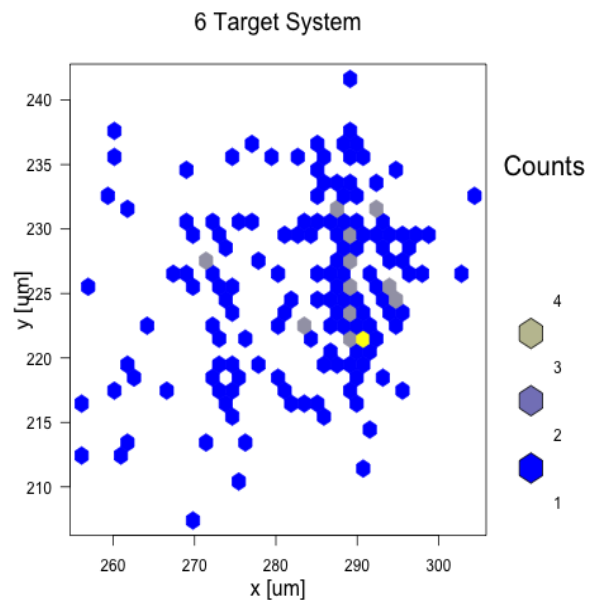


Abb. 31 Home-Target Messung

Im zweiten Versuch wurde zwischen zwei Punkten in der Mitte des Fahrweges gependelt. Wenn die Punkte im selben Punkt wie zuvor untersucht werden, ergibt sich eine leicht kleinere Streuung (Abb. 32). Dies kommt daher, dass bei dieser Messung nur +/-5mm Verfahren wurde.

Verwendeter Aufbau	System
Position 1	X206.2007 Y145.1001
Position 2	X211.2007 Y150.1001
Anzahl Messungen	200
RMS X	3.33 μm
RMS Y	5.79 μm

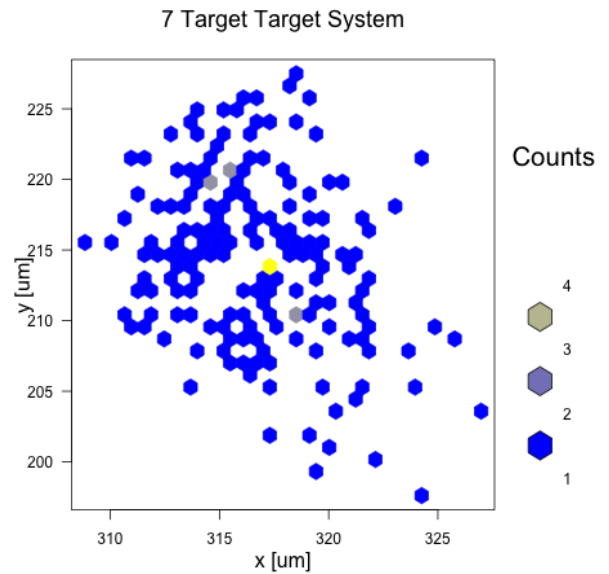


Abb. 32 Target-Target Messung

4.5.5 Relativ-Messungen

Bei diesen Messungen wurde die Relative Genauigkeit untersucht, welche erreicht werden kann, wenn man nur sehr kleine Strecken verfährt. Ich untersuchte, wie die Genauigkeit bei einer Schrittlänge von 1 μm aussieht. Bei der ersten Messreihe wurde der Magnet nur in X-Richtung bewegt (Abb. 33). Im zweiten Versuch wurde der Magnet nur in Y-Richtung bewegt (Abb. 34).

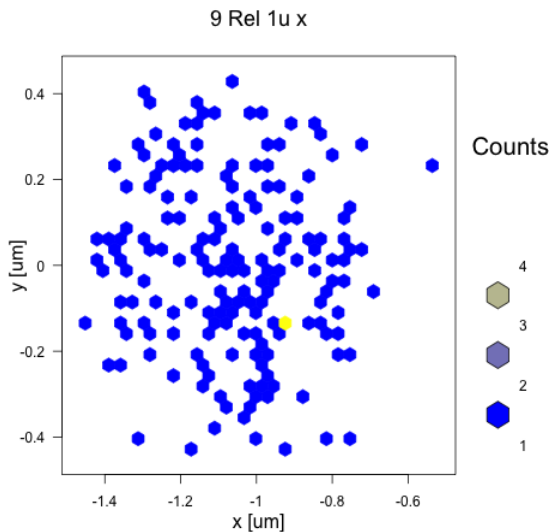


Abb. 33 Relativ Messung X

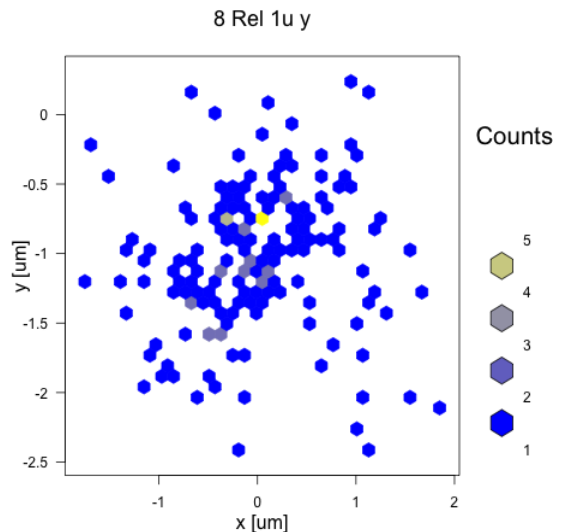


Abb. 34 Relativ Messung Y

Verwendeter Aufbau	System
Step	X 1 μm Y 0 μm
Anzahl Messungen	200
Mittelwert	X1.061 μm Y0.004 μm
RMS X	0.19 μm
RMS Y	0.20 μm

Verwendeter Aufbau	System
Step	X 0 μm Y 1 μm
Anzahl Messungen	200
Mittelwert	X0.038 μm Y1.029 μm
RMS X	0.66 μm
RMS Y	0.49 μm

4.5.6 Konklusion

Die Messungen haben ergeben, dass das Positionierungssystem den Magneten auf etwa $10\ \mu\text{m}$ genau positionieren kann. Dadurch wird die Verlangte Genauigkeit im Mikrometer Bereich nicht erreicht. Dies liegt zum einen daran, dass der Controller keinen Encoder verbaut hat und nur die Schritte zählt. Zudem wurde der Controller mit seinen Standardparametern verwendet. Durch optimieren der verschiedenen Parameter wie Strom, Microstepping Auflösung kann das Verhalten eventuell optimiert werden.

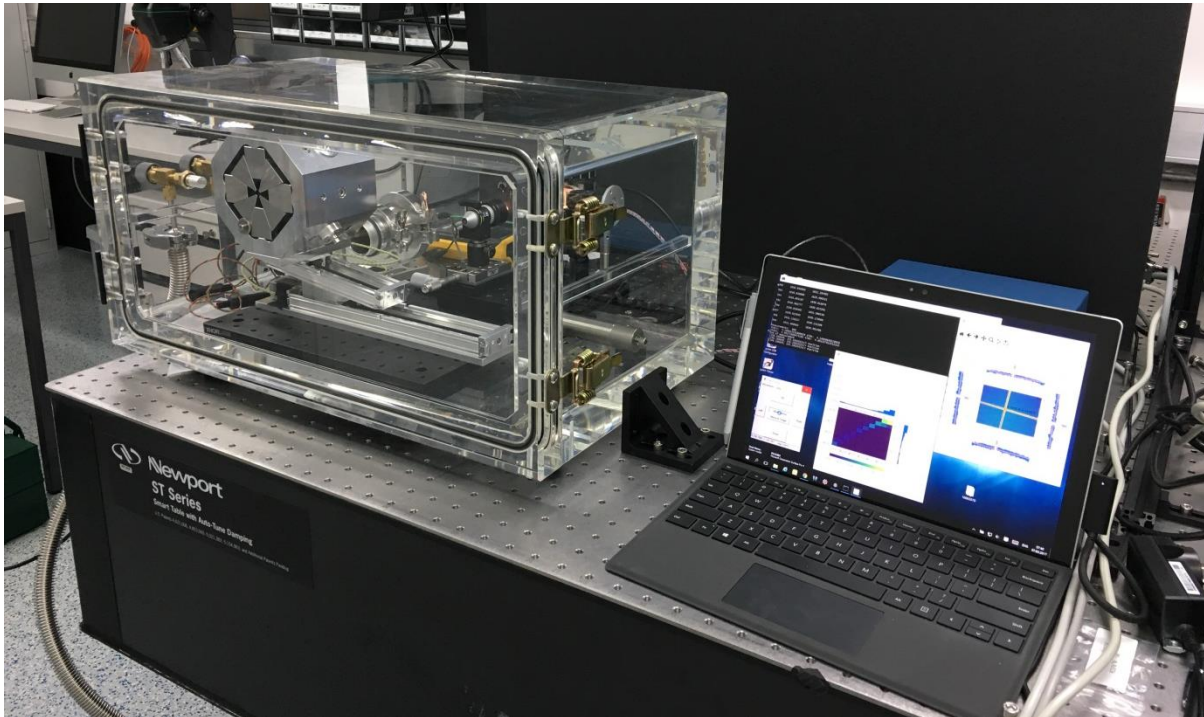


Abb. 35 Messaufbau

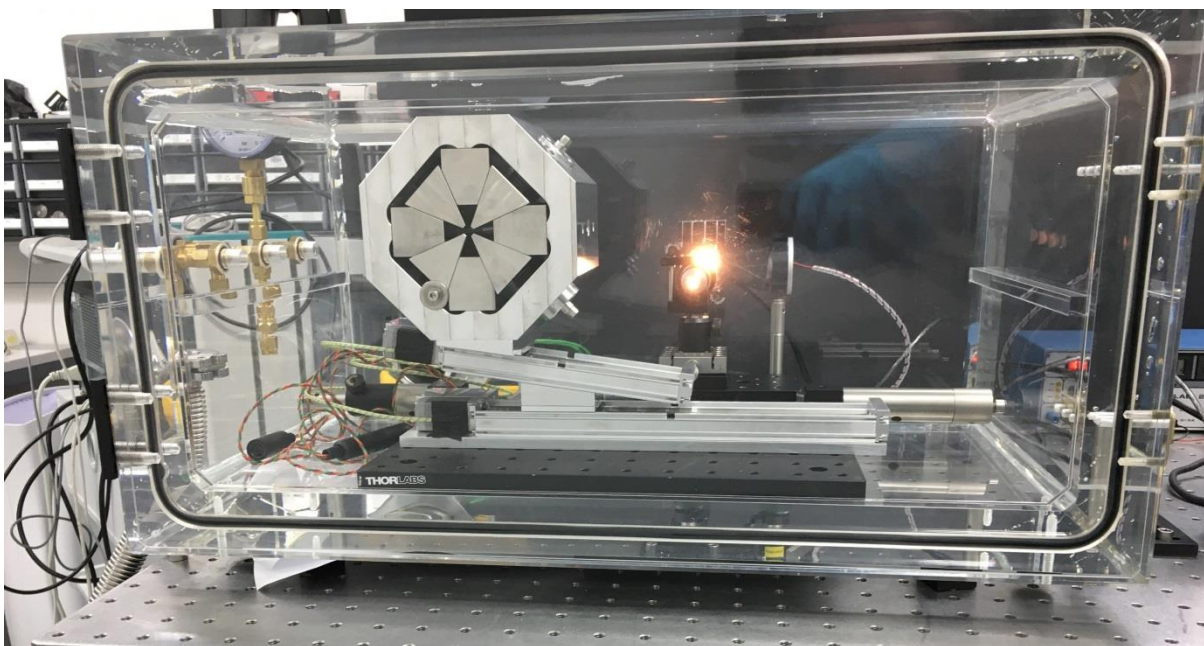


Abb. 36 Messaufbau

5 Auswertung der Arbeit

5.1 Software

Mit der entwickelten Software bin ich sehr zufrieden. Durch die Grafische Oberfläche ist das Messen sehr einfach und man sieht sofort, ob die Messung richtig läuft und hat eine Vorschau der gemessenen Daten. Die Software wurde erfolgreich verwendet um das Motorsystem zu vermessen.

Die Software erfüllt die verlangten Punkte:

- **Linearstages ansteuern**

Die Motoren können über ein Modul einfach angesprochen werden. Die häufigen Befehle sind als Funktionen vorhanden. Funktionen um auch speziellere Befehle zu senden sind ebenfalls vorhanden. Mit einem weiteren Modul ist es möglich die Position des Magneten direkt als X/Y Koordinaten anzugeben.

- **Kamera auslesen**

Die Kamer kann aus dem Programm ausgelesen werden. Die wichtigsten Parameter können geschrieben werden. Die Ansteuerung ist langsam, dies ist im für meine Anwendung jedoch nicht von Bedeutung.

- **Position des Kreuzes bestimmen**

Auf dem aufgenommenen Bild kann die Position des Kreuzes vollautomatisch erkannt werden. Der Algorithmus ist robust und kann Störungen verkraften.

- **Automatische Messungen**

Es stehen verschiedene Messmoden zu Verfügung, welche alle vollautomatisch ablaufen. Die Daten werden so exportierte, dass sie einfach weiterverarbeitet werden können. Um allfällige falsche Positionsbestimmungen vorzubeugen wird jeweils das Originalbild wie auch eine Version mit dem Fit gespeichert.

- **Benutzer Oberfläche**

Die Software verfügt über eine Grafische Oberfläche, welche dem Benutzer das Arbeiten erleichtert. Diese ist jedoch sehr rudimentär. Viele Benutzereingaben werden nicht komplett auf ihre Korrektheit überprüft. Momentan ist die Verwendung der Software durch andre Benutzer nicht geplant wodurch dies kein Problem darstellt.

5.2 Aufbau & Messungen

Mit dem Aufbau des Messsystems bin ich zufrieden. Ich konnte die die Messungen erfolgreich durchführen. Dies zeigt, dass der Versuchsaufbau geeignet war um die Messungen durchzuführen.

Folgende Vorgaben wurden erfüllt:

- **Messung im mit Sub um Genauigkeit.**

Die Messungen wurden mit einer Theoretischen Auflösung von $\Delta l = \frac{1.4}{8.5} = 0.165 \mu m$ durchgeführt. Durch die Streuung des Messaufbaus selbst ist effektiv eine Auflösung von $0.17 \mu m$ pro Pixel erreichbar. Durch die Mittelung beim Berechnen der Kreuzmitte kann jedoch mit Genauigkeiten im Sub-Pixel-Bereich gerechnet werden. Dadurch ist die Auflösung passend um Messungen im μm Bereich durchzuführen.

- **Flexibler Aufbau**

Durch die verwendeten Stages ist der Aufbau sehr Flexibel. Es kann innerhalb von einigen Minuten zwischen zwei Messarten umgebaut werden. Das Ausrichten auf das Kreuz gestaltet sich sehr einfach, da mehrere Lineartische verwendet wurden, die Kritischen Einstellungen vorzunehmen.

- **Messung im Vakuum**

Die Messungen wurden in einer Vakuumkammer durchgeführt. Der Druck ist jedoch deutlich höher als der später zu erwartende Enddruck. Dies gibt daher nur bedingt Aufschluss über die Vakuumverträglichkeit. Die zu erwartende Wärme Transport konnte jedoch besser untersucht werden als an Luft.

- **Statistische Auswertung**

Durch die vollautomatische Software war es möglich Messungen mit grosser Anzahl Punkte aufzunehmen. Aus Zeitgründen wurde jedoch ein relativ kleiner Wert für die Anzahl Messungen verwendet. Es sind jedoch genügend Messpunkte um statistische Aussagen zum Verhalten der Stages zu machen.

5.3 Probleme & Fehlerquellen

Schwierigkeiten & Probleme:

Zeit:

Ein grosses Problem war, dass die Messungen sehr viel Zeit beanspruchten. Die kürzeste Messung bei 200 Punkten dauerte 1.5 Stunden. Dies kann sich bei langen Verfahrdistanzen jedoch bis zu 9 Stunden verlängern. Dadurch war es nicht möglich mehr als 3 Messungen pro Tag zu machen.

Optiken Aufbauen:

Ich war mir das Arbeiten mit optischen aufbauten bereits aus früheren Praktika gewohnt. In diesem Labor hatte ich vor der IPA noch nie einen Aufbau durchgeführt. Ich musste feststellen, dass ich anfangs sehr viel Zeit brauchte mich im Labor zurechtzufinden.

Fehlerquellen:

Äussere Einflüsse:

Der Aufbau ist sicherlich anfällig auf Störungen aus der Umgebung wie Streulicht und Erschütterungen. Ich versuchte diese zu minimieren indem ich den Versuch im wenig frequentierten Optiklabor durchführte. Dort hatte ich die Möglichkeit den Aufbau auf dem optischen Tisch aufzubauen welcher sehr stabil steht. Zusätzlich wurde das Licht während den Messungen ausgeschaltet.

Thermische Prozesse:

Durch die Bewegung der Motoren entsteht Abwärme, welche das Verhalten des Systems beeinflussen kann. Die Temperatur wurde von Hand grob überwacht. Zwischen Ruhezustand und Verfahren wurde eine relativ konstante Temperaturdifferenz von 10°C gemessen.

Drift von Optikhaltern:

Alle Komponenten wurden an Optikhalten befestigt. Diese können über die Zeit etwas nachgeben. Besonders wenn diese mit der Stage ständig hin und her gefahren werden, was viele Vibrationen erzeugt. Dies ist schwierig zu überprüfen, da sich die Bewegungen auf wenige µm beschränken.

5.4 Ausblick

Die Messungen während der IPA gaben bereits einen guten Überblick zum Verhalten des Positionierung Systems. Dennoch können die Messungen zusätzlich optimiert werden, indem deutlich mehr Messpunkte ausgewertet werden. Dies kann helfen, die Probleme und oder Anforderungen des Systems genauer zu verstehen.

Der nächste Schritt wird jedoch sein das Positionierungssystem zu überarbeiten. Dabei stehen verschiedene Optionen offen. Die Stages werden mit Positionencoder nachgerüstet und die Ansteuerung der Motoren erfolgt dann geregelt mit genauen Sensoren. Die Alternative wäre, einen komplett anderen Aufbau zur Positionierung zu verwenden. Dieser könnte dann mit dem Bestehenden Aufbau ebenfalls ausgemessen werden.

5.5 Schlussfazit

Über die ganze IPA ziehe ich ein sehr positives Fazit. Ich konnte mich grösstenteils im Rahmen des Zeitplans bewegen, obwohl es einige Male etwas knapper wurde als geplant. Insbesondere das Abstimmen mit den langen Messungen brauchte mehr Flexibilität und Arbeitsstunden als ursprünglich gedacht.

Das Schreiben der Software klappte fast besser als ich gedacht hatte. Im Prozess kamen mir immer wieder weitere Ideen und Verbesserungen. Hier musste ich einige Male prioritätensetzen um den Zeitplan einhalten zu können.

Das Aufbauen des Versuchs brachte immer wieder kleinere Probleme. Diese konnte ich jedoch meistens gut umgehen und durch andere Teile kompensieren welche besser klappten als gedacht. Deutlich einfacher als ich im Voraus gedacht hatte war das Ausrichten der Kamer. Dies klappte immer innerhalb von einigen Minuten.

5.6 Danksagung

Ein grosses Dankeschön geht an meinen Fachvorgesetzten Rasmus Ischebeck und meinen Betreuer Eugenio Ferrari. Beide haben mich während der IPA begleitet und mir stets bereitwillig geholfen.

Zuletzt möchte ich mich bei meinem Experten Martin Klöckner und meinem Lehrmeister Tomislav Rebac bedanken welche stets grosses Interesse an meiner Arbeit zeigten.

6 Anhang

Datasheets:

- Zaber Linearstages T-LSM200A-SV2
- Basler acA4600 -10uc

Zeichnungen:

- Zusammenstellung ACHIP-PS-00

Programm:

- GUI_main.py
- histogramm.py
- imageanalysing.py
- utilities.py
- XY_POS.py
- zaber.py
- ZaberControll.py

Um die Übersicht zu verbessern wurden die Programme im beigelegten Dossier Gesammelt.

Messungen:

- 1 Stabilität Messsystem
- 2 Home Messung TLSM200
- 12 Home Messung TLSM100
- 5 Home Messung mit Magneten
- 4 Stabilität Messung mit Magneten
- 6 Target Messung
- 7 Target Target Messung
- 9 Relativ Messung X
- 8 Relativ Messung Y

Die gesamten Dokumente im Zusammenhang mit dem Positionierungssystem sind auf dem [BAB PH](#) Laufwerk sowie auf der [Switch Drive](#) hinterlegt.

Für weiter Informationen oder Fragen stehe ich zur Verfügung:

dominique.zehnder@psi.ch oder dominique.zehnder@hispeed.ch